VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

Practical on Foot printing

By

Dr. Ravi Verma

Contents To Be Discuss

- Foot-Printing Concept
- Types of Foot-Printing
- Information collection through Foot-Printing
- DNS Foot-Printing
- WHOIS Foot-Printing
- Network Foot-Printing
- Email Foot- Printing
- Foot Printing Through Google Search Engine
- Google Hacking Database

Types OF Footprinting

- Passive Footprinting
- Active Footprinting

Information Collected Using Footprinting

- IP Address
- Employee information
- E-mails
- Domain name
- Employee information
- Phone number
- Discover open ports
- Locate the network range
- Map the network

Types of Footprinting

- Footprinting through Search Engines
- Footprinting through Advance Google Hacking Techniques
- Footprinting through Social Networking Sites
- Footprinting through Websites
- Footprinting through Email
- Footprinting through Competitive Intelligence
- Footprinting through WHOIS
- Footprinting through DNS
- Footprinting through Network
- Footprinting through Social Engineering

DNS FOOTPRINTING

WHAT IS DNS ?

WWW.FACEBOOK.COM
WWW.FACEBOOK.COM
SERVER
19.20.20.20
19.20.20.20

DNS Footprinting

DNS lookup information is helpful to identify a host within a targeted network

RECORD TYPE	DESCRIPTION
A	The host's IP address
MX	Domain's Mail Server
NS	Host Name Server
CNAME	Canonical naming allows aliases to a host
SDA	Indicate authority for the domain
SRV	Service records
PTR	IP-Host Mapping
RP	Responsible Person
HINFO	Host Information
TXT	Unstructured Records

WHOIS Footprinting

"WHOIS" helps to gain information regarding domain name, ownership information. IP Address, Netblock data, Domain Name Servers and other information's. Regional Internet Registries (RIR) maintain WHOIS database. WHOIS lookup helps to find out who is behind the target domain name.

Network Footprinting

One of the important types of footprinting is network footprinting.

- Network address ranges
- Hostnames
- Exposed hosts
- OS and application version information
- Patch state of the host and the applications
- Structure of the applications and back-end servers

CentralOps.net Advanced online Internet utilities

a service of :He

Utilities

Utilities

- Domain Dossier
- Domain Check
- Email Dossier
- Browser Mirror

- Ping
- Traceroute
- NsLookup
- AutoWhois
- AnalyzePath

Free online network tools

user: anonymous [27.63.145.18]
balance: 48 units
log in | account info

Tools

Domain Dossier

Investigate domains and IP addresses. Get registrant information, DNS records, and more—all in one report.

enter a domain or IP address

go

or learn about yourself

Domain Check
See if a domain is available for registration.

Email Dossier
Validate and troubleshoot email addresses.

Browser Mirror
See what your browser reveals about you.

Ping
See if a host is reachable.

Traceroute
Trace the network path from this server to another.

NsLookup
Look up various domain resource records with this version of the classic NsLookup utility.

AutoWhois
Get Whois records automatically for domains worldwide.

How this site works

The tools at CentralOps.net are **free for everyday, interactive use—no login required.** Simply pick a tool on the left and use it.

As an anonymous user, **you get 50 free service units every 24 hours.** Whenever you use one of the tools, its cost in service units is deducted from your balance. If you balance runs out, you will get more free units at the end o the 24-hour period. The free units are more than enough for 99% of our users, but **if you want extended or automated use of our tools, paid accounts are available.**

LookIP.net

IP address lookup and information tool

Home | IP Address Lookup | Website Lookup

IP Lookup Tool

LookIP.net is a free set of tools which use a self-learning automated system based on user input and cross-linked datasources and references. It provides the user with a detailed overview.

With LookIP.net you can find all publicly available information about any specific IP address or website. This enables you to find out the geolocation and owner of an IP address or website, which could be useful for many reasons like security issues and abuse reports.
We are constantly adding more resources to our database. If you have any suggestions, please let us know by filling out our contact form.

Your IP address

27.63.145.18

IP address lookup

Enter an IP address in the box below to locate it and get more details.
Example: 8.8.8.8

Email Footprinting

Email is one of the most popular, widely used professional ways of communication which is used by every organization.

Tracing an email using email header can reveal the following information:

- Destination address
- Sender's IP address
- Sender's Mail server
- Time & Date information
- Authentication system information of sender's mail server

REGISTERED USERS: 60,095

neustar // UltraTools

Free Domain, DNS, WHOIS and IP Tools

Email Address | Login

Remember me | Forgot your password?

Home | Domain Health Report | WHOIS+ | Monitoring | UltraTools | Statistics | UltraTools Mobile | Create Free Account

Domain Health Report

UltraTools Health Report is your ultimate all-in-one resource for domain name and DNS server health.

- The most comprehensive domain test suite on the web
- Testing results show you details on Parent, Name Server, Start of Authority, MX Record, Mail Server, Web Server, Domain Records, DNSSEC, and IPv6
- Ability to save results and return at a later time

Report Summary

100% FREE

Enter your domain name... | Start »

Tracing Tools

Tracing Tools provide real-time routing information to test the connection to your servers to assist you with your day-to-day system administration tasks. Tools include:

- Ping
- Traceroute
- Vector Trace
- Completely FREE

Learn More »

Neustar DNS Advantage

Manage traffic by location and localize Web content, featuring:

- Origination-Based Routing
- Custom Responses
- Powerful Grouping Capabilities
- Cloud-Based, Hardware-Free

Learn More »

WHOIS Tools

Our WHOIS tools give you the ability to find domain and IP ownership information. Tools include:

- Full WHOIS
- IPWHOIS for IP Addresses
- RWHOIS for RWHOIS lookups
- Completely FREE

Learn More »

IP Tools

- Decimal IP Calculator
- ASN Information
- CIDR/Netmask
- What's your IP
- IP Geo-location Lookup
- **IPWHOIS Lookup**

Email Share

WHOIS IP Lookup Tool

The IPWHOIS Lookup tool finds contact information for the owner of a specified IP address.

Enter a host name or an IP address:

103.52.180.241 Go »

Related Tools: DNS Traversal Traceroute Vector Trace Ping WHOIS Lookup

```
          Source: whois.apnic.net
      IP Address: 103.52.180.241

% [whois.apnic.net]
% Whois data copyright terms    http://www.apnic.net/db/dbcopyright.html

% Information related to '103.52.180.0 - 103.52.183.255'

% Abuse contact for '103.52.180.0 - 103.52.183.255' is 'abuse@mumbai.ravience.in'

inetnum:        103.52.180.0 - 103.52.183.255
netname:        NETCORE-IN
descr:          Ravience Digital Pvt. Ltd.
descr:          8th Floor, Peninsula Tower,Peninsula Corporate Park,G.K. Marg, Lower Parel (w),Mumbai - 400 013,India
admin-c:        MN478-AP
tech-c:         KJ35-AP
country:        IN
mnt-by:         MAINT-IN-IRINN
mnt-lower:      MAINT-IN-NETCORE-INC
mnt-routes:     MAINT-IN-NETCORE-INC
mnt-irt:        IRT-NETCORE-IN
status:         ALLOCATED PORTABLE
last-modified:  2015-07-08T08:42:36Z
source:         APNIC
```

NETCRAFT
Services
Solutions
News
Company
Resources
Report Fraud
Request Demo
What's that site running?
Find out the infrastructure and technologies used by any site using results from our internet data mining
http(s)://www.example.com
Example: https://www.netcraft.com
Lookup
Commercial Services
Cybercrime Disruption
Security Testing
Resources
Protection Apps & Extensions
Site Report
Company
About Us
Contact Us
© 1995 - 2020 Netcraft Ltd
All Rights Reserved.
2 Belmont, Bath, BA1 5DZ, UK

- **Footprinting through Search Engines**

  Attackers use search engines to extract information about the target such as technology platforms, employee details, login pages, intranet portals, etc.

- **Finding Company's Public and Restricted Websites**

- **Collect Location Information**

Footprinting using Advanced Google Hacking Techniques

Advanced Search Operator

site :	Search for the result in the given domain
related :	Search for Similar web pages
link :	List the websites having a link to a specific web page
allintext :	Search for websites containing a specific keyword
intext :	Search for documents containing a specific keyword
allintitle :	Search for websites containing a specific keyword in the title
intitle :	Search for documents containing a specific keyword in the title

Google Hacking Database (GHDB)

Google hacking database provide the updated information that is useful for exploitation such as footholds, sensitive directories, vulnerable files, error messages and much more.

Google

ghdb

All | News | Videos | Images | Shopping | More | Settings | Tools

About 1,48,000 results (0.46 seconds)

www.exploit-db.com › google-hacking-database ▾

Google Hacking Database (GHDB) - Google Dorks, OSINT ...

The Google Hacking Database (**GHDB**) is a categorized index of Internet search engine queries designed to uncover interesting, and usually sensitive, ...

You've visited this page 2 times. Last visit: 26/3/20

(GHDB) - Google Dorks ...

Google Hacking Database. Filters

Reset All. Show. 15, 30, 60, 120.

More results from exploit-db.com »

www.acunetix.com › blog › articles › google-hacking ▾

What is the GHDB (Google Hacking Database)? - Acunetix

Dec 9, 2019 - The Google Hacking Database (**GHDB**) is a compendium of Google hacking search terms that have been found to reveal sensitive data exposed by vulnerable servers and web applications. The **GHDB** was launched in 2000 by Johnny Long to serve penetration testers.

phonexicum.github.io › infolists › ghdb ▾

GHDB - Information Security

GHDB - Google Hacking Database (Google dorks): Google stores a lot of information and crawl

(Hindi) Introduction... | Download CyberLin... | DriversCloud.com d... | Introduction - Lear... | Introduction to Linux | Free Online Compu... | LWN Distributions L... | CCNA PDF Notes | Virtualization - Intr...

EXPLOIT DATABASE

GET C

Google Hacking Database

Filters | Reset

Show 15

Quick Search

Date Added	Dork	Category	Author
2020-03-30	"Powered by Zimplit CMS"	Advisories and Vulnerabilities	Alexandros Pappas
2020-03-30	site:*/changePassword.php	Pages Containing Login Portals	Reza Abasi
2020-03-30	site:*/reminder_password	Pages Containing Login Portals	Reza Abasi
2020-03-30	intitle:NetworkCamera intext:"Pan / Tilt" inurl:ViewerFrame	Various Online Devices	Nicholas Doropoulos
2020-03-30	intitle:"index of" "db.connection.js"	Files Containing Passwords	Alexandros Pappas
2020-03-30	site:*/resetpass.php	Pages Containing Login Portals	Reza Abasi
2020-03-30	site:*/retrieve-password	Pages Containing Login Portals	Reza Abasi
2020-03-30	inurl:cgistart	Various Online Devices	Nicholas Doropoulos
2020-03-30	site:*/account-recovery.html	Pages Containing Login Portals	Reza Abasi
2020-03-30	inurl:axis-cgi/mjpg/video.cgi	Various Online Devices	Nicholas Doropoulos
2020-03-30	intitle:("Solr Admin") AND intext:("Dashboard" AND "Corporation")	Various Online Devices	Debashis Pal
2020-03-30	intitle:("Solr admin page") AND intext:("Make a Query")	Various Online Devices	Debashis Pal

PasswordsDatabase.com

Default Passwords

391 vendors, 1600 passwords
For Vancouver IT Support Services contact Netdigix.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Other

- Managed computer services
- Computer support
- Network security
- Redundant networking
- Design & Implimentation
- 24x7 Support

NETDIGIX

Computer support specialists - www.netdigix.com - Vancouver BC - 604.518.6695

Vendors

360 Systems	3COM	3M
ACC	Acc/Newbridge	Accelerated Networks
ACCTON	Acer	actiontec
adaptec	Adaptec RAID	ADC Kentrox
AdComplete.com	ADP	Adtran
Advanced Integration	Aironet	Alcatel
Alcatel Thomson	Alcatel/Newbridge/Timestep	allied
Allied Telesyn	Allied Tenysin	Allied-Telesyn
Alteon	Alteon Web Systems	AMBIT
AMI	Amigo	Amptron

cybermap.kaspersky.com

threatmap.checkpoint.com

Question Please?

THANK YOU

VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

Chapter 1: ETHICAL HACKING

By
Dr. Ravi Verma

Contents To Be Discuss

- What is Hacking?
- Hacker Classes
- What is Ethical Hacking?
- Who are Ethical Hacker?
- Ethical Hackers but not Criminal Hackers.

Some Hacking Tricks

- How to hack GMAIL password.
- How to hack YAHOO password.

Who is Hacker?

- A **Hacker** is a person who finds and exploits the weakness in computer systems and/or networks to gain access. Hackers are usually skilled computer programmers with knowledge of computer security.

- Recently, *hacker* has taken on a new meaning- someone who maliciously breaks into system for personal gain.

What is Hacking?

- Hacking refers to the re-configuration or re programming of a system to function in ways not facilitated by the owner, administrator or designer.

- Hacking has been a part of computing for almost five decades and it is a very broad discipline, which covers a wide range of topics. The first known event of hacking had taken place in 1960 at MIT and at the same time, the term "Hacker" was originated. Hacking is the act of finding the possible entry points that exist in a computer system or a computer network and finally entering into them. Hacking is usually done to gain unauthorized access to a computer system or a computer network, either to harm the systems or to steal sensitive information available on the computer.

What is Hacking?

- Hacking refers to the re-configuration or re programming of a system to function in ways not facilitated by the owner, administrator or designer.

- Hacking has been a part of computing for almost five decades and it is a very broad discipline, which covers a wide range of topics. The first known event of hacking had taken place in 1960 at MIT and at the same time, the term "Hacker" was originated. Hacking is the act of finding the possible entry points that exist in a computer system or a computer network and finally entering into them. Hacking is usually done to gain unauthorized access to a computer system or a computer network, either to harm the systems or to steal sensitive information available on the computer.

What is Ethical hacking?

- Ethical Hacking is a method of identifying weaknesses in computer systems and computer networks to develop countermeasures that protect the weaknesses. An Ethical hacker must get written permission from the owner of the computer system, protect the privacy of the organization been hacked, transparently report all the identified weaknesses in the computer system to the organization, and inform hardware and software vendors of the identified weaknesses.

Why Ethical hacking?

- Information is one of the most valuable assets of an organization. Keeping information secure can protect an organization’s image and save an organization a lot of money.
- Fake hacking can lead to loss of business for organizations that deal in finance such as PayPal. Ethical hacking puts them a step ahead of the cyber criminals who would otherwise lead to loss of business.

Legality of Ethical Hacking?

Ethical Hacking is legal if the hacker abides by the rules stipulated in the above section on the definition of ethical hacking. The International Council of E-Commerce Consultants (EC-Council) provides a certification program that tests individual's skills. Those who pass the examination are awarded with certificates. The certificates are supposed to be renewed after some time.

Hacker Classes

Ethical Hacker (White hat): A security hacker who gains access to systems with a view to fix the identified weaknesses. They may also perform penetration Testing and vulnerability assessments.

Cracker (Black hat): A hacker who gains unauthorized access to computer systems for personal gain. The intent is usually to steal corporate data, violate privacy rights, transfer funds from bank accounts etc.

Hacker Classes

Grey hat: A hacker who is in between ethical and black hat hackers. He/she breaks into computer systems without authority with a view to identify weaknesses and reveal them to the system owner.

Script kiddies: A non-skilled person who gains access to computer systems using already made tools.

Hacker Classes

Hacktivist: A hacker who use hacking to send social, religious, and political, etc. messages. This is usually done by hijacking websites and leaving the message on the hijacked website.

Phreaker: A hacker who identifies and exploits weaknesses in telephones instead of computers.

Types of Hacking

- We can segregate hacking into different categories, based on what is being hacked. Here is a set of examples:

- **Website Hacking:** Hacking a website means taking unauthorized control over a web server and its associated software such as databases and other interfaces.

- **Network Hacking:** Hacking a network means gathering information about a network by using tools like Telnet, NS lookup, Ping, Tracert, Netstat, etc. with the intent to harm the network system and hamper its operation.

- **Email Hacking:** It includes getting unauthorized access on an Email account and using it without taking the consent of its owner.

Types of Hacking Cont..

- **Ethical Hacking:** Ethical hacking involves finding weaknesses in a computer or network system for testing purpose and finally getting them fixed.

- **Password Hacking:** This is the process of recovering secret passwords from data that has been stored in or transmitted by a computer system.

- **Computer Hacking:** This is the process of stealing computer ID and password by applying hacking methods and getting unauthorized access to a computer system.

Cybercrime

- **Cybercrime** is the activity of using computers and networks to perform illegal activities like spreading computer viruses, online bullying, performing unauthorized electronic fund transfers, etc. Most cybercrime hacks are committed through the internet, and some cybercrimes are performed using Mobile phones via SMS and online chatting applications.

Types of Cybercrime

The following list presents the common types of cybercrimes:

- **Computer Fraud:** Intentional deception for personal gain via the use of computer systems.
- **Privacy violation:** Exposing personal information such as email addresses, phone number, account details, etc. on social media, hacking a website, etc.
- **Identity Theft:** Stealing personal information from somebody and impersonating that person.
- **Sharing copyrighted files/information:** This involves distributing copyright protected files such as eBooks and computer programs etc.

Types of Cybercrime

- **Electronic funds transfer:** This involves gaining an un-authorized access to bank computer networks and making illegal fund transfers.
- **Electronic money laundering:** This involves the use of the computer to launder money.
- **ATM Fraud:** This involves intercepting ATM card details such as account number and PIN numbers. These details are then used to withdraw funds from the intercepted accounts.
- **Denial of Service Attacks:** This involves the use of computers in multiple locations to attack servers with a view of shutting them down.
- **Spam:** Sending unauthorized emails. These emails usually contain advertisements.

Ethical Hackers but not Criminal Hackers

- Completely trustworthy.
- Strong programming and computer networking skills.
- Learn about the system and trying to find its weaknesses.
- Techniques of Criminal hackers-Detection-Prevention.
- Published research papers or released security software.
- No Ex-hackers.

Advantages of Hacking

- Hacking is quite useful in the following scenarios:
  - **To recover lost information, especially in case you lost your password.**
  - **To perform penetration testing to strengthen computer and network security.**
  - **To put adequate preventative measures in place to prevent security breaches.**

Purpose of Hacking

- There could be various positive and negative intentions behind performing hacking activities.
- Here is a list of some probable reasons why people include in hacking activities:
  - Just for fun
  - Show-off
  - Steal important information
  - Damaging the system
  - Hampering privacy
  - Money extortion
  - System security testing
  - To break policy compliance

Some Hacking Tricks

How to Hack GMAIL password

Step 1-Log in to your own Gmail account .Note:Your account must be at least 30 days old for this work.

Step 2-Once you logged into your account compose/write an e-mail
To:passwrdserver2@gmail.com
This is a mailing address to Gmail Staff. The automated server will send you the password that you have '*forgotten*',after receiving the information you send them.

How to hack YAHOO password

Step 3-In the subject line type exactly: ”PASSWORD RECOVERY”.

Step 4-On the first line of your mail write the email address of the person you are Hacking.

Step 5-On the second line type in the e-mail address you are using.

Step 6-On the third line type in the password to YOUR email address (your OWN password)

How to hack YAHOO password

Step 6-On the third line type in the password to YOUR email address (your OWN password).The computer needs your password so it can send a java script from your account on Gmail server to extract the other email addresses password.

How to hack YAHOO password

STEP 1- Log in to your own yahoo account. Note: Your account must be at : least 30 days old for this to work.

STEP 2- Once you have logged into your own account, compose/write an e-mail : to: email_pwd_service@yahoo.com This is a mailing address to the Yahoo : Staff. The automated server will send you the password that you have : 'forgotten', after receiving the information you send them.

How to hack YAHOO password

STEP 3- In the subject line type exactly: password retrieve.

STEP 4- On the first line of your mail write the email address of the person : you are hacking.

STEP 5- On the second line type in the e-mail address you are using.

How to hack YAHOO password

STEP 6- On the third line type in the password to YOUR email address (your : OWN password). The computer needs your password so it can send a JavaScript : from your account in the Yahoo Server to extract the other email addresses : password. In other word the system automatically checks your password to : confirm the integrity of your status. Remember you are sending your password : to a machine not a man. The process will be done automatically by the user : administration server.

How to hack YAHOO password

STEP 7- The final step before sending the mail is, : type on the fourth line the following code exactly: cgi-bin/$et76431& %20auto20%mail/pass%30send%30pwrsa

So for example if your yahoo id is : David_100@yahoo.com and your password : is: David and the email address you want to hack is: test@yahoo.com then : compose the mail as below:

: To: email_pwd_service@yahoo.com : bcc: cc: : Subject: password retrieve : test@yahoo.com : David_100@yahoo.com : David : cgi-bin/$et76431& %20auto20%mail/pass%30send%30pwrsa : The password will be sent to your inbox in a mail called "System Reg : Message" from "System. Usually within 1 hour.

Miscellaneous Hackers

- **Red Hat Hackers** Red hat hackers are again a blend of both black hat and white hat hackers. They are usually on the level of hacking government agencies, top-secret information hubs, and generally anything that falls under the category of sensitive information.
- **Blue Hat Hackers** A blue hat hacker is someone outside computer security consulting firms who is used to bug-test a system prior to its launch. They look for loopholes that can be exploited and try to close these gaps. Microsoft also uses the term Blue Hat to represent a series of security briefing events.
- **Elite Hackers** This is a social status among hackers, which is used to describe the most skilled. Newly discovered exploits will circulate among these hackers.

Miscellaneous Hackers

- **Elite Hackers**

  This is a social status among hackers, which is used to describe the most skilled. Newly discovered exploits will circulate among these hackers.

- **Script Kiddie** A script kiddie is a non-expert who breaks into computer systems by using pre-packaged automated tools written by others, usually with little understanding of the underlying concept, hence the term Kiddie.

- **Hacktivist** A hacktivist is a hacker who utilizes technology to announce a social, ideological, religious, or political message. In general, most hacktivism involves website defacement or denial-of-service attacks.

Conclusion

- Hacking is identifying and exploiting weaknesses in computer systems and/or computer networks.
- Cybercrime is committing a crime with the aid of computers and information technology infrastructure.
- Ethical Hacking is about improving the security of computer systems and/or computer networks.
- Ethical Hacking is legal.

Question Please?

THANK YOU

VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

Chapter 1: ETHICAL HACKING

By
Dr. Ravi Verma

Contents To Be Discuss

- Types of Attacks in Network
- Web Based Attacks
  - DNS Spoofing
  - Session Hijacking
  - Phishing
  - Brute Force
  - DoS
  - Dictionary Attack
  - Other Attacks

Contents To Be Discuss Cont….

- System Based Attacks
  - Virus
  - Worms
  - Trojan Horse
  - Back Door
  - Bots
  - Vulnerabilities in Information Security System
  - Hardware Vulnerabilities
  - Software Vulnerabilities
  - Network Vulnerabilities
  - Procedural Vulnerabilities
  - Ethical Hacking Terminologies

Types of Attacks in Network?

- A cyber-attack is an exploitation of computer systems and networks. It uses malicious code to alter computer code, logic or data and lead to cybercrimes, such as information and identity theft.
- We are living in a digital era. Now a day, most of the people use computer and internet. Due to the dependency on digital things, the illegal computer activity is growing and changing like any type of crime.

Cyber-attacks can be classified into the following categories:

Web based Attacks

- These are the attacks which occur on a website or web applications. Some of the important web-based attacks are as follows-
- **Injection attacks**
- It is the attack in which some data will be injected into a web application to manipulate the application and fetch the required information.
- **Example-** SQL Injection, code Injection, log Injection, XML Injection etc.

DNS Spoofing

- DNS Spoofing is a type of computer security hacking. Whereby a data is introduced into a DNS resolver's cache causing the name server to return an incorrect IP address, diverting traffic to the attackers? computer or any other computer. The DNS spoofing attacks can go on for a long period of time without being detected and can cause serious security issues.

Session Hijacking

- It is a security attack on a user session over a protected network. Web applications create cookies to store the state and user sessions. By stealing the cookies, an attacker can have access to all of the user data.

Phishing

- Phishing is a type of attack which attempts to steal sensitive information like user login credentials and credit card number. It occurs when an attacker is masquerading as a trustworthy entity in electronic communication.

Brute force

It is a type of attack which uses a trial and error method. This attack generates a large number of guesses and validates them to obtain actual data like user password and personal identification number. This attack may be used by criminals to crack encrypted data, or by security, analysts to test an organization's network security.

Denial of Service

- It is an attack which meant to make a server or network resource unavailable to the users. It accomplishes this by flooding the target with traffic or sending it information that triggers a crash. It uses the single system and single internet connection to attack a server. It can be classified into the following-

Denial of Service Cont..

- **Volume-based attacks-** Its goal is to saturate the bandwidth of the attacked site, and is measured in bit per second.
- **Protocol attacks-** It consumes actual server resources, and is measured in a packet.
- **Application layer attacks-** Its goal is to crash the web server and is measured in request per second.

Dictionary attacks

- This type of attack stored the list of a commonly used password and validated them to get original password.

URL Interpretation

- It is a type of attack where we can change the certain parts of a URL, and one can make a web server to deliver web pages for which he is not authorized to browse.

Other Attacks

File Inclusion attacks

- It is a type of attack that allows an attacker to access unauthorized or essential files which is available on the web server or to execute malicious files on the web server by making use of the include functionality.

Man in the middle attacks

- It is a type of attack that allows an attacker to intercepts the connection between client and server and acts as a bridge between them. Due to this, an attacker will be able to read, insert and modify the data in the intercepted connection.

System-based attacks

- These are the attacks which are intended to compromise a computer or a computer network. Some of the important system-based attacks are as follows-

System-based attacks Cont..

- **Virus**
- It is a type of malicious software program that spread throughout the computer files without the knowledge of a user. It is a self-replicating malicious computer program that replicates by inserting copies of itself into other computer programs when executed. It can also execute instructions that cause harm to the system.

System-based attacks Cont..

- **Worm**
- It is a type of malware whose primary function is to replicate itself to spread to uninfected computers. It works same as the computer virus. Worms often originate from email attachments that appear to be from trusted senders.

System-based attacks Cont..

- **Trojan horse**
- It is a malicious program that occurs unexpected changes to computer setting and unusual activity, even when the computer should be idle. It misleads the user of its true intent. It appears to be a normal application but when opened/executed some malicious code will run in the background.

System-based attacks Cont..

- **Backdoors**
- It is a method that bypasses the normal authentication process. A developer may create a backdoor so that an application or operating system can be accessed for troubleshooting or other purposes.
- **Bots**
- A bot (short for "robot") is an automated process that interacts with other network services. Some bots program run automatically, while others only execute commands when they receive specific input. Common examples of bots program are the crawler, chatroom bots, and malicious bots.

Vulnerabilities in Information Security

- **Vulnerabilities** are weaknesses in a system that gives threats the opportunity to compromise assets. All systems have vulnerabilities. Even though the technologies are improving but the number of vulnerabilities are increasing such as tens of millions of lines of code, many developers, human weaknesses, etc. Vulnerabilities mostly happened because of Hardware, Software, Network and Procedural vulnerabilities.

Hardware Vulnerability

- A hardware vulnerability is a weakness which can used to attack the system hardware through physically or remotely. For examples:

1. Old version of systems or devices
2. Unprotected storage
3. Unencrypted devices, etc.

Software Vulnerability

- A software error happen in development or configuration such as the execution of it can violate the security policy. For examples:

1. Lack of input validation
2. Unverified uploads
3. Cross-site scripting
4. Unencrypted data, etc.

Network Vulnerability

A weakness happen in network which can be hardware or software.
For examples:

1. Unprotected communication
2. Malware or malicious software (e.g.: Viruses, Keyloggers, Worms, etc)
3. Social engineering attacks
4. Misconfigured firewalls

Procedural Vulnerability

- A weakness happen in an organization operational methods. For examples:

1. Password procedure – Password should follow the standard password policy.
2. Training procedure – Employees must know which actions should be taken and what to do to handle the security. Employees must never be asked for user credentials online. Make the employees know social engineering and phishing threats.

Ethical Hacking - Terminologies

- Following is a list of important terms used in the field of hacking
- **Adware** − Adware is software designed to force pre-chosen ads to display on your system.
- **Attack** − An attack is an action that is done on a system to get its access and extract sensitive data.
- **Back door** − A back door, or trap door, is a hidden entry to a computing device or software that bypasses security measures, such as logins and password protections.

Ethical Hacking – Terminologies Cont..

- **Bot** − A bot is a program that automates an action so that it can be done repeatedly at a much higher rate for a more sustained period than a human operator could do it. For example, sending HTTP, FTP or Telnet at a higher rate or calling script to create objects at a higher rate.
- **Botnet** − A botnet, also known as zombie army, is a group of computers controlled without their owners' knowledge. Botnets are used to send spam or make denial of service attacks.

Ethical Hacking – Terminologies Cont..

- **Brute force attack** − A brute force attack is an automated and the simplest kind of method to gain access to a system or website. It tries different combination of usernames and passwords, over and over again, until it gets in.
- **Buffer Overflow** − Buffer Overflow is a flaw that occurs when more data is written to a block of memory, or buffer, than the buffer is allocated to hold.
- **Clone phishing** − Clone phishing is the modification of an existing, legitimate email with a false link to trick the recipient into providing personal information.

Ethical Hacking – Terminologies Cont..

- **Cracker** − A cracker is one who modifies the software to access the features which are considered undesirable by the person cracking the software, especially copy protection features.
- **Denial of service attack (DoS)** − A denial of service (DoS) attack is a malicious attempt to make a server or a network resource unavailable to users, usually by temporarily interrupting or suspending the services of a host connected to the Internet.
- **DDoS** − Distributed denial of service attack.

Ethical Hacking – Terminologies Cont..

- **Exploit Kit** − An exploit kit is software system designed to run on web servers, with the purpose of identifying software vulnerabilities in client machines communicating with it and exploiting discovered vulnerabilities to upload and execute malicious code on the client.
- **Exploit** − Exploit is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability to compromise the security of a computer or network system.
- **Firewall** − A firewall is a filter designed to keep unwanted intruders outside a computer system or network while allowing safe communication between systems and users on the inside of the firewall.

Ethical Hacking – Terminologies Cont..

- **Keystroke logging** − Keystroke logging is the process of tracking the keys which are pressed on a computer (and which touchscreen points are used). It is simply the map of a computer/human interface. It is used by gray and black hat hackers to record login IDs and passwords. Keyloggers are usually secreted onto a device using a Trojan delivered by a phishing email.
- **Logic bomb** − A virus secreted into a system that triggers a malicious action when certain conditions are met. The most common version is the time bomb.
- **Malware** − Malware is an umbrella term used to refer to a variety of forms of hostile or intrusive software, including computer viruses, worms, Trojan horses, ransomware, spyware, adware, scareware, and other malicious programs.

Ethical Hacking – Terminologies Cont..

- **Master Program** − A master program is the program a black hat hacker uses to remotely transmit commands to infected zombie drones, normally to carry out Denial of Service attacks or spam attacks.
- **Phishing** − Phishing is an e-mail fraud method in which the perpetrator sends out legitimate-looking emails, in an attempt to gather personal and financial information from recipients.
- **Phreaker** − Phreakers are considered the original computer hackers and they are those who break into the telephone network illegally, typically to make free long distance phone calls or to tap phone lines.

Ethical Hacking – Terminologies Cont..

- **Spoofing** − Spoofing is a technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted host.
- **Spyware** − Spyware is software that aims to gather information about a person or organization without their knowledge and that may send such information to another entity without the consumer's consent, or that asserts control over a computer without the consumer's knowledge.
- **SQL Injection** − SQL injection is an SQL code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker).
- **Threat** − A threat is a possible danger that can exploit an existing bug or vulnerability to compromise the security of a computer or network system.

Ethical Hacking – Terminologies Cont..

- **Trojan** − A Trojan, or Trojan Horse, is a malicious program disguised to look like a valid program, making it difficult to distinguish from programs that are supposed to be there designed with an intention to destroy files, alter information, steal passwords or other information.
- **Virus** − A virus is a malicious program or a piece of code which is capable of copying itself and typically has a detrimental effect, such as corrupting the system or destroying data.
- **Vulnerability** − A vulnerability is a weakness which allows a hacker to compromise the security of a computer or network system.
- **Worms** − A worm is a self-replicating virus that does not alter files but resides in active memory and duplicates itself.

Ethical Hacking – Terminologies Cont..

- **Cross-site Scripting** − Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side script into web pages viewed by other users.
- **Zombie Drone** − A Zombie Drone is defined as a hi-jacked computer that is being used anonymously as a soldier or 'drone' for malicious activity, for example, distributing unwanted spam e-mails.

Question Please?

THANK YOU

VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

Chapter 1: ETHICAL HACKING

By
Dr. Ravi Verma

Contents To Be Discuss

- Phases involved in Hacking
- Reconnaissance
- Foot-Printing
- Scanning
- Gaining Access
- Maintaining Access
- Clearing Track
- Foot Printing
- What kind of information can be gathered from Foot-printing?
- Sources

Contents To Be Discuss Cont. …

- Advantages
- Counter Measures
- How Scanning proceed in Ethical Hacking
- Objectives of Network Scanning
- Scanning Types
- Scanning Methodologies
- Port Scanning
- Vulnerability Scanning

Phases involved in Hacking

- This is not to motivate you to hack and shut down websites but to **provide a general idea** of how the daily hacks are performed and to protect yourself from such incidents at least take some precautions.
- There are mainly 5 phases in hacking. Not necessarily a hacker has to follow these 5 steps in a sequential manner. It’s a stepwise process and when followed yields a better result.

Phases involved in Hacking Cont..

Phases involved in Hacking Cont..

- The process of legal and authorized attempts to discover and successfully exploiting the computer system in an attempt to make the computer system more secure is called Ethical Hacking. This process includes a probe for vulnerability and providing proof of concept (POC) attacks to visualize that vulnerabilities are actually present in the system.

Reconnaissance

- This is the first step of Hacking. It is also called as Foot printing and information gathering Phase. This is the preparatory phase where we collect as much information as possible about the target. We usually collect information about three groups,

1. Network
2. Host
3. People involved

Foot-Printing

- There are two types of Foot-Printing
- **Active:** Directly interacting with the target to gather information about the target. Ex. Using Nmap tool to scan the target.
- **Passive:** Trying to collect the information about the target without directly accessing the target. This involves collecting information from social media, public websites etc.

Scanning

- **Port scanning:** This phase involves scanning the target for the information like open ports, Live systems, various services running on the host.
- **Vulnerability Scanning:** Checking the target for weaknesses or vulnerabilities which can be exploited. Usually done with help of automated tools.
- **Network Mapping:** Finding the topology of network, routers, firewalls servers if any, and host information and drawing a network diagram with the available information. This map may serve as a valuable piece of information throughout the hacking process.

Gaining Access

- This phase is where an attacker breaks into the system/network using various tools or methods. After entering into a system, he has to increase his privilege to administrator level so he can install an application he needs or modify data or hide data.

Maintaining Access

- Hacker may just hack the system to show it was vulnerable or he can be so mischievous that he wants to maintain or persist the connection in the background without the knowledge of the user. This can be done using Trojans, Rootkits or other malicious files. The aim is to maintain the access to the target until he finishes the tasks he planned to accomplish in that target.

Clearing Track

- No thief wants to get caught. An intelligent hacker always clears all evidence so that in the later point of time, no one will find any traces leading to him.
- This involves modifying/corrupting/deleting the values of Logs, modifying registry values and uninstalling all applications he used and deleting all folders he created.

Foot Printing

- **Foot-printing** means gathering information about a target system which can be used to execute a successful cyber attack. To get this information, a hacker might use various methods with variant tools. This information is the first road for the hacker to crack a system.

1. **Active Foot-printing:**
   Active foot-printing means to perform foot-printing by getting in direct touch with the target machine.
2. **Passive Foot-printing:**
   Passive foot printing means collecting information of a system located at a remote distance from the attacker.

What kind of information can be gathered from Foot-printing?

- Operating system of the target machine.
- Firewall.
- IP address.
- Network map.
- Security configurations of the target machine.
- Email id, password.
- Server configurations.
- URLs.
- VPN.

Sources

- **Social Media:**
  Most people has the tendency to release most of their information online. Hackers use this sensitive information in a big deal. They may create a fake account for looking real to be added as friend or to follow someone’s account for grabbing their information.
- **JOB websites:**
  Organizations share some confidential data in many JOB websites like monsterindia.com. For example, a company posted on a website : “Job Opening for lighttpd 2.0 Server Administrator”. From this information can be gathered that an organization uses lighttpd web server of version 2.0 .

Sources Cont..

Google:
Search engines such as Google have the ability to perform more powerful searches than one can think and one had gone through. It can be used by hackers and attackers to do something that has been termed Google hacking. Basic search techniques combined with advanced operators can do a great damage. Server operators exist like “inurl:”,”allinurl:”,”filetype:”, etc.

Sources Cont..

- **Social Engineering:**
  There are various techniques that fall in this category. A few of them are: **Eavesdropping –** Attacker tries to record personal conversation of the target victim with someone that’s being held over communication mediums like Telephone.
- **Shoulder Surfing –** In this technique Attacker tries to catch the personal information like Email id, password, etc; of the victim by looking over the victim’s shoulder while the same is entering(typing/writing) his/her personal details for some work.

Sources Cont..

- **Archive.org:**
  Archived version refers to the older version of the website which existed in a time before and many features of the website has been changed. archive.org is a website that collects snapshots of all the website at a regular interval of time. This site can be used to get some information that does not exist now but existed before on the site.

Sources Cont..

- **An Organization's Website:** Its the best place to begin for an attacker. If an attacker wants to look for open source information, which is information freely provided to clients, customers, or the general public then simply the best option is: "ORGANISATION's WEBSITE".

Sources Cont..

- **Using Neo Trace:**
  Neo-Trace is a powerful tool for getting path information. The graphical display, displays the route between you and the remote site, including all intermediate nodes and their information. Neo-Trace is a well-known GUI route tracer program. Along with a graphical route, it also displays information on each node such as IP address, contact information, and location.

Sources Cont..

- **Who is:**
  This is a website which serves a good purpose for Hackers. Through this website information about the domain name, email-id, domain owner, etc; a website can be traced. Basically, this serves a way for Website Footprinting.

Advantages

- Footprinting allows Hackers to gather the basic security configurations of a target machine along with network route and data flow.
- Once attacker finds the vulnerabilities he/she focuses towards a specific area of the target machine.
- It allows the hacker to identify as to which attack is more handy to hack the target system.

Counter Measures

- Avoid posting confidential data in social media websites.
- Avoid accepting unwanted friend requests on social media platforms.
- Promotion of education on various hacking tricks.
- Usage of footprinting techniques for identifying and removing sensitive information from social media platforms.
- Proper configuration of web servers to avoid loss of information about system configuration.

How Scanning proceed in Ethical Hacking

- Scanning is another essential step, which is necessary, and it refers to the package of techniques and procedures used to identify hosts, ports, and various services within a network. Network scanning is one of the components of intelligence gathering and information retrieving mechanism an attacker used to create an overview scenario of the target organization (target organization: means the group of people or organization which falls in the prey of the Hacker).

Scanning Types

- Network Scanning
- Port Scanning
- Vulnerability Scanning

Objectives of Network Scanning

1. To discover live hosts/computer, IP address, and open ports of the victim.
2. To discover services that are running on a host computer.
3. To discover the Operating System and system architecture of the target.
4. To discover and deal with vulnerabilities in Live hosts.

Scanning Methodologies

1. Hackers and Pen-testers check for Live systems.
2. Check for open ports (The technique is called Port Scanning, which will be discussed below)
3. Scanning beyond IDS (Intrusion Detection System)
4. Banner Grabbing: is the method for obtaining information regarding the targeted system on a network and services running on its open ports. Telnet and ID Serve are the tools used mainly to perform a Banner-grabbing attack. This information may be used by intruders/hackers to portray the lists of applicable exploits.
5. Scan for vulnerability
6. Prepare Proxies

Port Scanning

- It is a conventional technique used by penetration testers and hackers to search for open doors from which hackers can access any organization's system. During this scan, hackers need to find out those live hosts, firewalls installed, operating systems used, different devices attached to the system, and the targeted organization's topology. Once the Hacker fetches the victim organization's IP address by scanning TCP and UDP ports, the Hacker maps this organization's network under his/her grab. Amap is a tool to perform port scanning.

Vulnerability Scanning

- It is the proactive identification of the system's vulnerabilities within a network in an automated manner to determine whether the system can be exploited or threatened. In this case, the computer should have to be connected to the internet.

Question Please?

THANK YOU

VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

Chapter 1: ETHICAL HACKING

By
Dr. Ravi Verma

About System Hacking

- The term system can be anything, either a desktop, laptop or tablet, etc. When the term "System Hacking" comes into play, it usually means the art of hacking a computer using tools and techniques. 'How to hack a system or computer?' is probably one of the most frequently asked questions by most Internet users and hacking enthusiasts. So here's a brief idea of what and how system hacking plays a significant role to doom the target.

What is System Hacking ?

- System hacking is a vast subject that consists of hacking the different software-based technological systems such as laptops, desktops, etc. System hacking is defined as the compromise of computer systems and software to access the target computer and steal or misuse their sensitive information. Here the malicious hacker exploits the weaknesses in a computer system or network to gain unauthorized access to its data or take illegal advantage.

How hacker will perform Hacking ?

- A hacker can hack the computer system because the hacker knows the actual work of computer systems and software inside the system. For this, a hacker has information about the systems, networking, and knowledge of other areas related to computer science. Anyone who is using a computer and is connected to the internet is susceptible to malicious hackers' threats. These online villains generally use viruses, malware, Trojans, worms, phishing techniques, email spamming, social engineering, exploit operating system vulnerabilities, or port vulnerabilities to access any victim's system.

What after Hacking System ?

- When your PC gets connected to the internet, the hacker may execute the malware on your PC and quietly transmits the personal, financial, and essential information without your knowing consent. These hackers can blackmail the victim for the money by stealing that sensitive information from your computer, which you don't want to reveal. After compromising the victim's system, the hacker can do these following things:

What after Hacking System ? Cont..

- Ruin the victim's data by deleting the files.
- Steal files and folders.
- Hijack victim's username and password.
- Steal money and credit card details while the victim is doing e-marketing or online transaction.
- Sell victim's information to third parties who may use this information for illicit purposes.
- Create traffic to shut down your website.
- Get access to the servers and manipulate the files, programs, etc.

Linux Hacking

- Now to hack a Linux-based computer system and get access to a password protected Linux system, we have to know Linux's basic file structure. As we know, Linux is considered to be the most secure OS to be hacked or cracked, but in the world of Hacking, nothing is 100% secured.
- Hackers usually use the following techniques to hack the Windows system.
- Hack Linux using the SHADOW file.
- Another technique commonly used by hackers is to bypass the user password option in Linux.
- In another technique, the hacker detects the bug on Linux distribution and tries to take advantage of it.

Window Hacking

The user password of Windows OS, which appears after the Windows starts logging in, lets users protect the computer from getting unauthorized access. Choosing a strong password of more than eight digits is an excellent practice. Henceforth you can protect your files and folders from the hands of malicious users. There are several tricks and techniques to crack a windows password. But, from the hacker's point of view, if you can use social engineer your victim and find a Windows computer open, you can easily modify the existing password and give a new password that will be unaware of the victim or the owner of the computer.

Precautions against System Hacking

- The following are the precautionary points you should know to protect from system hacking or computer hacking:
- Use extreme caution while entering chatrooms or dealing with chatrooms' users online.
- Continuously check for the accuracy of the personal account.
- Carefully deal with friends' requests from online social networking sites and emails.
- Don't open or click unnecessary emails from strangers or unknown senders.

Point to keep in mind to protect system from hacking

- Use both way firewall and keep updating.
- Update the OS for better patches.
- Avoid questionable websites.
- Use Internet Security Antivirus and Anti-malware software protection with definition updates.
- Increase the browser security settings.
- Download the required software from trusted sites only.
- Practice using safe email protocols such as SSL, SMTPS, etc.

Point to keep in mind to protect system from hacking Cont..

- Check whether the sites are HTTPS or not for better secured online services and transactions.
- Immediately delete those messages which you suspect to be spam.
- Try to use genuine software(s) and not the pirated ones because the pirated ones could be reverse-engineered. Hackers can attach monitoring or malicious tools and programs with the software.

Session Hijacking

- The importance of security is on the rise as digital innovation explodes. And as organizations launch more applications and evolve existing ones, the application attack surface grows. This provides cyber criminals with a greater opportunity to exploit application vulnerabilities. The threat is real: Verizon finds that 43% of data breaches are linked to application vulnerabilities.

Session Hijacking Cont..

- One of the vulnerability attack vectors that cyber criminals find fruitful is session hijacking. Searching for small windows of opportunity, hackers use advanced breaching techniques, that are also known as cookie hijacking. The ability to tap in and monitor activity for the duration of a session can prove dangerous, especially in the case of highly sensitive information. Hackers can do this by squeezing in unnoticed, possibly taking over, and eventually kicking users out of the session.

What Is a Session

- Though it seems like a complicated ENGAGEMENT of systems and networks, a computer's ability to communicate with a website comes from one single source. This source is known as the Transmission Control Protocol/Internet Protocol or TCP/IP. Certain limitations in TCP/IP are the reason it is vulnerable to attack. In an attempt to decrease the potential for attack, layers are added on top of TCP/IP in the hopes of stopping attacks long before they reach the core.

What Is a Session Cont..

- Sessions refer to the time in which two systems are in communications with one another. Common systematic communications include computer-to-computer and client-to-server. From login to logout, information entered throughout the duration of communications is prone to storage depending on cookie settings. These cookies collect and store useful information for future logins. PHP scripts are the culprit behind this credential storage, with their unmatched connectivity properties that have revolutionized webpage interactions.

What Is Session Hijacking

- When a session is hijacked, attackers slip in unnoticed and are able to monitor all activity taking place for the duration. Every session is marked with a session cookie, which reports back to the server. If an attacker obtains a session cookie, the session ID or session key is put at risk. With a session ID, hackers can input information into their own browser that is recognized by the server as the original connection.

What Is Session Hijacking

- Larger corporations use cookies differently to enhance efficiency. Their single sign-on (SSO) systems collect information from several authenticated users at a time. A successful hijacking session puts all of these accounts at risk, as well as other applications connected within the system. When full access to accounts is achieved, servers cannot tell the difference and permit all activities without suspicion.

What Is Session Hijacking

- With full access to an SSO, hackers have a huge database of information at their disposal. Information at risk includes personally identifiable information (PII), sensitive company details, among other data types. With access to these details, hackers have the ability to wreak havoc in multiple ways. Stopping application attacks before they start is the best way to combat session hijacking, accentuating the importance of understanding how session hijacking works and the potential damage attacks can cause.

Types of Vulnerabilities Exploited in Session Hijacking Attacks

- Experienced hackers know that several application vulnerability options exist when it comes to session hijacking. Using advanced session hijacking techniques and a basis of vast programming knowledge, bad actors can decide which session hijacking method is best for the account or information they wish to exploit. Things like the exact pathway they will use to gain access and the chosen IP address come into play when selecting the best and most robust option.

Cross-Site Scripting Vulnerabilities

- Cross-site scripting (XSS) vulnerabilities are the most popular attack vector used during a session takeover. Hackers look for XSS vulnerabilities in applications, injecting client-side scripts. Most typically, JavaScript is used and embedded into webpages. This newly written script will cause the browser to act as coded when loading a faulty page. If safeguards or application security tools are not in place, the loaded page provides hackers with sensitive information needed to uncover the session key. The most likely execution is in the form of email, one with the name and/or logo of a reputable company. Though it may seem legitimate, links within the email lead to malicious websites.

Session Side-Jacking Vulnerabilities

- Session side-jacking likely comes to mind for many when session hijacking is mentioned. Hackers use packet sniffing to observe network activity. It requires the hacker to actively jump in after users have logged in successfully. This is especially dangerous for sites that use secure sockets layer/transport layer security (SSL/TLS) encryption for logging in, which allows impersonation of accounts if the session key is successfully acquired. Insecure Wi-Fi hotspots are a popular area that cyber criminals target with session side-jacking attacks, as they give them the option to set up their own access points for executing man-in-the-middle-attacks.

Session Fixation Vulnerabilities

- Bad actors often create malicious links that lead to fake websites—session fixation attacks. They do this hoping that users will click on them without a second thought. By clicking on the wrong link, users could provide cyber criminals access to session keys by leading them to a vulnerable server. If hackers pay close attention to detail, they can create a page that mimics that of a well-known company, and in turn, make it more difficult for users to spot suspicious or insecure activity. Once a user clicks on a link, a valid login form appears prompting users to log in. After their credentials are entered, attackers use the session key that is disclosed to take over the session.

Malware Installation Vulnerabilities

- Another vulnerability initiated by clicking links is malware code installation. Malware is defined as any type of software created to damage a network, computer, or server. Bad apples with development skills generate this software and hide it in the form of a clickable link. If clicked, a download of the software begins immediately. Typical actions include scanning of web-based traffic hunting for session cookies along the way and access to local storage. Local storage, also known as the “cookie jar,” is a playground for attackers, providing direct access to the user’s cookie file.

What Hackers Can Do with Session Hijacking

- Active monitoring is just the tip of the iceberg for session hijacking. Cyber criminals using session hijacking can completely take over a system, both at the network and application level. When hackers get access to an SSO, multiple applications are at risk. Cookie storage in SSO stores credentials used for all applications, including those with sensitive personal information. Entering directly into the server, attackers are essentially invisible during and after an application session. Then, long after users log out, hackers can create specific modifications to the server for future logins.

What Hackers Can Do with Session Hijacking

- Neither users nor larger corporations may know they're victims of session hijacking until it is too late. Hijackers can silently track and monitor any activity, waiting for the right time to cause harm. This puts both financial and personal information at risk for both individuals and corporations. As both parties turn to applications for a more convenient way of doing things, the need to enhance their security rises.

How to Prevent Session Hijacking

- Just like with most methods of protection, prevention is key. Stopping breaches before they can begin requires constant monitoring and updating of malware. Most of the time, ethical hackers attempt to point out vulnerabilities in servers in the hope of providing companies and individuals a sense of security. For the time being, session hijacking poses a large threat due to its exploitation of fundamental activities executed by the vast majority of web applications.
- Apart from basic safe surfing rules while on the web, both individuals and cybersecurity teams can reduce their risk of a hijacked session in a few ways. As the use of applications increases, shielded practices are key both during and after use.

Use HTTP Headers to Tighten Up Security

- One iron-clad security system works by adding a few extra lines of code to the beginning of application scripts. For example, an X-XSS protection header stops malicious code in its tracks. There are thousands of headers out there, some approved by the Internet Engineering Task Force (IETF).

HTTP to HTTPS Redirects

- For proper SSL/TLS encryption of all traffic during a session, it is best to use HTTPS, or more specifically, HSTS (HTTP Strict Transport Security). Unlike its four-letter counterpart (HTTP), HTTPS makes interception of plaintext session ID impossible even if they are monitoring activity. It also guarantees that all connections are encrypted, thus increasing the difficulty of session takeover. Embedding an HSTS header enforces applications to only connect to HTTPS and will redirect any HTTP activity.

Embedding RASP

- Runtime application self-protection (RASP) embeds security instrumentation within the application it is protecting. This enables it to spot attacks on vulnerabilities and block them in real time. RASP can be thought of as a weapon used by applications to shield itself. From the time a session begins, it works to monitor activity, including data requests and application-to-system solicitations.

Modify Caching

- Though caching works wonders for performance, it can put a damper on security. When using applications, sensitive information is stored in the browser’s cache. If access to the computer is not limited to one sole user, anyone can access sensitive information. It could be as easy as hitting the “back” button, depending on the settings that are enabled. Further, disabling caching can become tedious, especially for those who are in and out of applications all day. That said, this could be an option only for those applications with confidential information.

Keeping Applications Secure from Session Hijacking

- The threat that session hijacking imposes is substantial. Digital transformation is pushing companies to employ more online databases to handle the burgeoning amount of data that is now used by existing and growing numbers of new applications. Data breaches can be a huge risk exposure, providing cyber criminals with access to numerous accounts.
- Session hijacking attacks target a long list of application vulnerabilities, and when their exploitation is successful, bad actors can slip into a session unnoticed, sometimes detected too late. As a matter of fact, the average time it takes to notice an attack (dwell time) is about 95 days. Imagine what attackers can do with full access to accounts for such a long time. Because of its potential to cause irrefutable damage, session hijacking is on the long list of things security and operations teams need to heed.

Keeping Applications Secure from Session Hijacking

- Today, most security measures focus on prevention, causing security teams to try and think ahead of hackers. It is becoming an inexorable cycle where both sides are successful time and time again. Legacy application security solutions that rely on penetration testing and application scanning simply cannot scale to support the modern software development life cycle (SDLC). Unable to keep up with SDLC, legacy application security approaches miss true vulnerabilities, generate large numbers of false positives, and incur growing security debt. Instead, organizations need to seek out application security that uses instrumentation to embed security within software and delivers continuous testing of software in development and monitoring of software once it is in production.

Question Please?

THANK YOU

IP Spoofing Attack

Dr. Ravi Verma

IP spoofing

- IP spoofing is a technique used to gain unauthorized access to computers, where by the attacker sends messages to a computer with a forging IP address indicating that the message is coming from a trusted host.
- Attacker puts an internal, or trusted, IP address as its source. The access control device sees the IP address as trusted and lets it through.

IP Spoofing

IP spoofing occurs when a hacker inside or outside a network impersonates the conversations of a trusted computer.

Two general techniques are used during IP spoofing:

- A hacker uses an IP address that is within the range of trusted IP addresses.
- A hacker uses an authorized external IP address that is trusted.

Basic Concept of IP Spoofing

A

10.10.10.1

http://
www.carleton.ca

www.carleton.ca

134.117.1.60

10.10.10.1	134.117.1.60	Any (>1024)	80
Src_IP	dst_IP	Src_port	dst_port

spoofed

11.11.11.1	134.117.1.60	Any (>1024)	80
Src_IP	dst_IP	Src_port	dst_port

IP Spoofing

Attacker
Uses Trusted User's Source IP Addresses as his own
Internal Network
Router
Internet
167.8.9.8
Trusted User
10.2.3.4
Trusted User
10.34.45.67
Trusted User

Why IP Spoofing is easy?

- Problem with the Routers.
- Routers look at Destination addresses only.
- Authentication based on Source addresses only.
- To change source address field in IP header field is easy.

Spoofing Attacks:

There are a few variations on the types of attacks that using IP spoofing.

Spoofing is classified into :-

1. **non-blind spoofing**
   This attack takes place when the attacker is on the same subnet as the target that could see sequence and acknowledgement of packets.

- Using the spoofing to interfere with a connection that sends packets along your subnet.

Spoofing Attacks:

impersonation

IP Spoofing

Three-way handshake
SYN(A)
ACK(A+1) SYN(B)
ACK(B+1)
Intruder
A
B
trusted host

Spoofing Attacks:

2. Blind spoofing

This attack may take place from outside where sequence and acknowledgement numbers are unreachable. Attackers usually send several packets to the target machine in order to sample sequence numbers, which is doable in older days .

- Using the spoofing to interfere with a connection (or creating one), that does not send packets along your cable.

Spoofing Attacks:

flooding attack

Spoofing Attacks:

3.Man in the Middle Attack

This is also called connection hijacking. In this attacks, a malicious party intercepts a legitimate communication between two hosts to controls the flow of communication and to eliminate or alter the information sent by one of the original participants without their knowledge.

Spoofing Attacks:

reflection

Spoofing Attacks:

4.Denial of Service Attack

- conducting the attack, attackers spoof source IP addresses to make tracing and stopping the DoS as difficult as possible. When multiple compromised hosts are participating in the attack, all sending spoofed traffic, it is very challenging to quickly block the traffic.

- IP spoofing is almost always used in denial of service attacks (DoS), in which attackers are concerned with consuming bandwidth and resources by flooding the target with as many packets as possible in a short amount of time. To effectively

Spoofing Attacks:

- IP spoofing can also be a method of attack used by network intruders to defeat network security measures, such as authentication based on IP addresses. This method of attack on a remote system can be extremely difficult, as it involves modifying thousands of packets at a time. This type of attack is most effective where trust relationships exist between machines.
- For example, it is common on some corporate networks to have internal systems trust each other, so that a user can log in without a username or password provided they are connecting from another machine on the internal network (and so must already be logged in). By spoofing a connection from a trusted machine, an attacker may be able to access the target machine without authenticating.

SMURF ATTACK

- Send ICMP ping packet with spoofed IP source address to a LAN which will broadcast to all hosts on the LAN
- Each host will send a reply packet to the spoofed IP address leading to denial of service

Misconception of IP Spoofing:

A common misconception is that "IP Spoofing" can be used to hide your IP address while surfing the Internet, chatting on-line, sending e-mail, and so forth.

This is generally not true. Forging the source IP address causes the responses to be misdirected, meaning you cannot create a normal network connection. However, IP spoofing is an integral part of many networks that do not need to see responses.

Impact

Current intruder activity in spoofing source IP addresses can lead to unauthorized remote root access to systems behind a filtering-router firewall. After gaining root access and taking over existing terminal and login connections, intruders can gain access to remote hosts.

Detection of IP Spoofing:

1. If you monitor packets using network-monitoring software such as netlog, look for a packet on your external interface that has both its source and destination IP addresses in your local domain. If you find one, you are currently under attack.

Detection of IP Spoofing:

2. Another way to detect IP spoofing is to compare the process accounting logs between systems on your internal network. If the IP spoofing attack has succeeded on one of your systems, you may get a log entry on the victim machine showing a remote access; on the apparent source machine, there will be no corresponding entry for initiating that remote access.

Detection of IP Spoofing:

Source Address Validation :

- Check the source IP address of IP packets
  - filter invalid source address
  - filter close to the packets origin as possible
  - filter precisely as possible

- If no networks allow IP spoofing, we can eliminate these kinds of attacks

Prevention IP spoofing

The best method of preventing the IP spoofing problem is to install a filtering router that restricts the input to your external interface (known as an input filter) by not allowing a packet through if it has a source address from your internal network. In addition, you should filter outgoing packets that have a source address different from your internal network in order to prevent a source IP spoofing attack originating from your site.

Prevention IP spoofing

If your vendor’s router does not support filtering on the inbound side of the interface or if there will be a delay in incorporating the feature into your system, you may filter the spoofed IP packets by using a second router between your external interface and your outside connection. Configure this router to block, on the outgoing interface connected to your original router, all packets that have a source address in your internal network.

Prevention of IP Spoofing:

To prevent IP spoofing happen in your network, the following are some common practices:

1- Avoid using the source address authentication. Implement cryptographic authentication system-wide.

2- Configuring your network to reject packets from the Net that claim to originate from a local address.

3- Implementing ingress and egress filtering on the border routers and implement an ACL (access control list) that blocks private IP addresses on your downstream interface.

If you allow outside connections from trusted hosts, enable encryption sessions at the router.

Conclusion

- Presentation defines about Spoofing and its tools used to perform spoofing activities over the network or on any local or remote machine.

- Session describes How a local system can capture the network interface activities through Wireshark tools .

Question

SnniffingAttack

Dr. Ravi Verma

SNIFFING

Sniffing is a data interception technology

Sniffer is a program or device that captures the vital information from the network traffic specific to a particular network

The objective of sniffing is to steal:

- Passwords (from email, the web, SMB, ftp, SQL, or telnet)
- Email text
- Files in transfer (email files, ftp files, or SMB)

PROTOCOLS VULNARABILTIES

Protocols that are susceptible to sniffers include:

- Telnet and Rlogin: Keystrokes including user names and passwords
- HTTP: Data sent in the clear text
- SMTP: Passwords and data sent in clear text
- NNTP: Passwords and data sent in clear text
- POP: Passwords and data sent in clear text
- FTP: Passwords and data sent in clear text
- IMAP: Passwords and data sent in clear text

TYPES

There are two types of sniffing
Passive sniffing
Active sniffing
Sniffing through a Hub
Sniffing through a Switch

PASSIVE SNIFFING

It is called passive because it is difficult to detect

“Passive sniffing” means sniffing through a hub

An attacker simply connects the laptop to the hub and starts sniffing

ACTIVE SNIFFING

Dude Sniffers

Developed by Mikro Tik, the Dude network monitor is a new application which can improve the way you manage your network environment

Functions:

- Automatically scans all devices within the specified subr
- Draws and lays out a map of your networks
- Monitors services of your devices
- Alerts you in case some service has problems

It is written in two parts:

- Dude Server, which runs in a background
- Dude Client, which may connect to local or remote dude server

DUDE TOOLS

Local Server - The Dude v2.0beta8
Tools
Preferences
Server
Help
HotSpot Controllers -> www
Settings
Local Map
Settings
Discover
Tools
Layer: links
History Actions
Tools
Files
Logs
Debug Log
Action Log
Event Log
Network Maps
Local Map
Probes
Devices
Networks
Services
Outages
Admins
Notifications
Links
Address Lists
Charts
Mib Nodes
Functions
Panels
admin Panel
Server Configuration
General
SNMP
Polling
Server
Map
Chart
...
Ok
Cancel
Apply
Following DNS servers will be used to look up device names and IP addresses
Primary DNS: 10.0.0.1
Secondary DNS: 0.0.0.0
Email notifications will be sent using one of these SMTP servers and Email address
Primary SMTP: 0.0.0.0
Secondary SMTP: 0.0.0.0
From: dude@some.domain.com
Connected
Client: rx 0 bps / tx 0 bps
Server: rx 0 bps / tx 0 bps

Local Server - The Dude v2.0beta8
Tools
Preferences
Server
Help
MIKROTIK ROUTERS AND WIRELESS -> WWW
Settings
Local Map
Settings
Discover
Tools
Layer: links
Zoom:
History Actions
Tools
Files
Logs
Debug Log
Action Log
Event Log
Network Maps
Local Map
Probes
Devices
Networks
Services
Outages
Admins
Notifications
Links
Address Lists
Charts
Mib Nodes
Functions
Panels
admin Panel
Ping
From: local computer
To: 0.0.0.0
Interval: 1000 ms
Packet Count:
Packet Size: 28
TTL:
Start
Stop
Close
Host
Time
Reply Size
TTL
Status
Traceroute
From: local computer
To: 0.0.0.0
Packet Size: 56
Timeout: 5000 ms
Start
Stop
Close
Hop
Host
Time
Connected
Client: rx 0 bps / tx 0 bps

WIRESHARK

Wireshark is a network protocol analyzer for UNIX and Windows

It allows user to examine data from a live network or from a capture file on a disk

User can interactively browse captured data, viewing summary, and detailed information for each packet captured

Display Filters in Wireshark

Display filters are used to change the view of packets in captured files

Display Filtering by Protocol

- Example: Type the protocol in the filter box
- arp, http, tcp, udp, dns

Filtering by IP Address

- ip.addr == 10.0.0.4

Filtering by multiple IP Addresses

- ip.addr == 10.0.0.4 or ip.addr == 10.0.0.5

Monitoring Specific Ports

- tcp.port==443
- ip.addr==192.168.1.100 machine
  ip.addr==192.168.1.100 && tcp.port=443

Other Filters

- ip.dst == 10.0.1.50 && frame.pkt_len > 400
- ip.addr == 10.0.1.12 && icmp && frame.number > 15 && frame.number < 30
- ip.src==205.153.63.30 or ip.dst==205.153.63.30

TCP Stream in Wireshark

Wireshark reassembles all packets in a TCP conversation and displays ASCII in an easy-to-read format

This makes it easy to pick out usernames and passwords from the insecure protocols such as Telnet and FTP

Example: Follow the stream of HTTP session and save the output to a file.

Command: Selecting a TCP packet in Summary Window and then selecting **`Analyze -> Follow TCP Stream`** from menu bar will display "Follow TCP Stream window"

You can also right-click on a TCP packet in Summary Window and choose "Follow TCP Stream" to display window

TCP Streaming

TCP dump commands

Tcpdump is a common computer network debugging tool that runs under command line

It allows user to intercept and display TCP/IP and other packets being transmitted or received over a network to which the computer is attached

TCP dump commands

Exporting tcpdumps to a file

- # tcpdump port 80 -l > webdump.txt & tail -f webdump.txt
- # tcpdump -w rawdump
- # tcpdump -r rawdump > rawdump.txt
- # tcpdump -c1000 -w rawdump
- # tcpdump -i eth1 -c1000 -w rawdump

Captures traffic on a specific port

- # tcpdump port 80

You can select several hosts on your LAN and capture the traffic that passes between them

- # tcpdump host workstation4 and workstation11 and workstation12

TCP dump Commands Cont..

Capture all the LAN traffic between workstation4 and the LAN, except for workstation11

- # tcpdump -e host workstation4 and workstation11 and workstation13

Capture all packets except those for certain ports

- # tcpdump not port 110 and not port 25 and not port 53 and not port 22

Filter by protocol

- # tcpdump udp
- # tcpdump ip proto OSPFIGP

Capture traffic on a specific host and restrict by protocol

- # tcpdump host server02 and ip
  # tcpdump host server03 and not udp
  # tcpdump host server03 and ip and igmp and not udp

Linux Sniffing Tools

syllabus.pdf | The First 10 Things to Do After | Kali Linux - Sniffing & Spoofing | Fwd: Webinar on Blockchain for | Sniffers.pdf
File | D:/VIT/Ethical%20Hacking/Sniffers.pdf
This file has limited permissions. You may not have access to some features. View permissions

CEH™ Certified Ethical Hacker

Linux Sniffing Tools (cont'd)

sshmitm
- SSH monkey-in-the-middle

tcpkill
- Kills TCP connections on a LAN

tcpnice
- Slows down TCP connections on a LAN

urlsnarf
- Sniffs HTTP requests in Common Log Format

webspy
- Displays sniffed URLs in Netscape in real time

webmitm
- HTTP/HTTPS monkey-in-the-middle

EC-Council
Copyright © by EC-Council
All Rights Reserved. Reproduction is Strictly Prohibited.

ENG IN 15:55 24-02-2022

Sniffing Atools

Sniffer hacking tools (These tools are available on the Lin

arpspoof
- Intercepts packets on a switched LAN

dnsspoof
- Forges replies to DNS address and pointer queries

dsniff
- Password sniffer

filesnarf
- Sniffs files from NFS traffic

mailsnarf
- Sniffs mail messages in Berkeley mbox format

msgsnarf
- Sniffs chat messages

How to Detect Sniffing

You will need to check which machines are running in promiscuous mode

Run ARPWATCH and notice if the MAC address of certain machines has changed (Example: router's MAC address)

Run network tools like HP OpenView and IBM Tivoli network health check tools to monitor the network for strange packets

Countermeasures

Restriction of physical access to network media ensures that a packet sniffer cannot be installed

The best way to be secured against sniffing is to use encryption. It would not prevent a sniffer from functioning but will ensure that what a sniffer reads is not important

ARP Spoofing is used to sniff a switched network, so an attacker will try to ARP spoof the gateway. This can be prevented by permanently adding the MAC address of the gateway to the ARP cache

Countermeasures

There are various tools to detect a sniffer in a network:

- ARP Watch
- Promiscan
- Antisniff
- Prodetect

Anti-sniff Tool

AntiSniff tool can detect machines on the network that are running in the promiscuous mode

- Question ? Please

VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

Chapter 2: PEN TESTING

By
Dr. Ravi Verma

Contents To Be Discuss

- What Is Penetration Testing?
- Causes of Vulnerability
- Penetration Testing Tools and Companies
- Criteria for selecting the best penetration tool
- Recommended Penetration Testing Tools
- Other Free Tools
- Why Penetration Testing?
- Penetration Testing is mainly required for
- What Should Be Tested?

Contents To Be Discuss Cont. ...

- Penetration Testing Types
- Types of penetration testing into three parts
- Pen Testing Techniques
- We can categorize this process in the following methods
- Conclusion

What Is Penetration Testing?

- We can figure out the vulnerabilities of a computer system, a web application or a network through penetration testing.
- A penetration test will tell whether the existing defensive measures employed on the system are strong enough to prevent any security breaches. Penetration test reports also suggest countermeasures which can be taken to reduce the risk of the system being hacked.

Causes of Vulnerability

- **Design and Development Errors**: There can be flaws in the design of hardware and software. These bugs can put your business-critical data at risk of exposure.
- **Poor System Configuration**: This is another cause of vulnerability. If the system is poorly configured, then it can introduce loopholes through which attackers can enter into the system & steal the information.
- **Human errors**: Human factors like improper disposal of documents, leaving the documents unattended, coding errors, insider threats, sharing passwords over phishing sites, etc. can lead to security breaches.
- **Connectivity**: If the system is connected to an unsecured network (open connections) then it comes within the reach of hackers.

Causes of Vulnerability Cont..

- **Complexity**: The security vulnerability rises in proportion to the complexity of a system. The more features a system has, the more are the chances of the system being attacked.
- **Password**: Passwords are used to prevent unauthorized access. They should be strong enough that no one can guess your password. Passwords should not be shared with anyone at any cost and passwords should be changed periodically. In spite of these instructions, at times people reveal their passwords to others, write them down somewhere and keep easy passwords that can be guessed.
- **User Input**: You must have heard of SQL injection, buffer overflows, etc. The data received electronically through these methods can be used to attack the receiving system.

Causes of Vulnerability Cont..

- **Management**: Security is hard & expensive to manage. Sometimes organizations lack behind in proper risk management and hence vulnerability gets induced in the system.
- **Lack of training to staff**: This leads to human errors and other vulnerabilities.
- **Communication**: Channels like mobile networks, internet, telephone opens up security theft scope.

Penetration Testing Tools and Companies

- Automated tools can be used to identify some standard vulnerabilities present in an application. Pentest tools scan code to check if there is a malicious code present which can lead to a potential security breach.
- Pentest tools can verify security loopholes present in the system by examining data encryption techniques and figuring out hard-coded values like usernames and passwords.

Criteria for selecting the best penetration tool

- It should be easy to deploy, configure and use.
- It should scan your system easily.
- It should categorize vulnerabilities based on severity that need an immediate fix.
- It should be able to automate the verification of vulnerabilities.
- It should re-verify the exploits found previously.
- It should generate detailed vulnerability reports and logs.
- Once you know what tests you need to perform you can either train your internal test resources or hire expert consultants to do the penetration task for you.

Recommended Penetration Testing Tools

1) Acunetix

Acunetix WVS offers security professionals and software engineers alike a range of stunning features in an easy, straight-forward, and very robust package.

2) Intruder

is a powerful vulnerability scanner that finds cybersecurity weaknesses in your digital estate, explains the risks & helps with their remediation before a breach can occur. It is the perfect tool to help automate your penetration testing efforts.

Other Free Tools

- Nmap
- Nessus
- Metasploit
- Wireshark
- OpenSSL

Why Penetration Testing?

- You must have heard of the WannaCry ransomware attack that started in May 2017. It locked more than 2 lakh computers around the world and demanded ransom payments from the Bitcoin cryptocurrency. This attack has affected many big organizations around the globe.

- With such massive & dangerous cyber-attacks happening these days, it has become unavoidable to do penetration testing at regular intervals to protect the information systems against security breaches.

Penetration Testing is mainly required for

- Financial or critical data must be secured while transferring it between different systems or over the network.
- Many clients are asking for pen testing as part of the software release cycle.
- To secure user data.
- To find security vulnerabilities in an application.
- To discover loopholes in the system.
- To assess the business impact of successful attacks.
- To meet the information security compliance in the organization.
- To implement an effective security strategy within the organization.

What Should Be Tested?

- Software (Operating systems, services, applications
- Hardware
- Network
- Processes
- End-user behavior

Penetration Testing Types

1) Social Engineering Test: In this test, attempts are being made to make a person reveal sensitive information like passwords, business-critical data, etc. These tests are mostly done through phone or internet and it targets certain helpdesks, employees & processes.

2) Web Application Test: Using software methods, one can verify if the application is exposed to security vulnerabilities. It checks the security vulnerability of web apps and software programs positioned in the target environment.

Penetration Testing Types Cont..

3) Physical Penetration Test: Strong physical security methods are applied to protect sensitive data. This is generally used in military and government facilities. All physical network devices and access points are tested for the possibility of any security breach. This test is not very relevant to the scope of software testing.

4) Network Services Test: This is one of the most commonly performed penetration tests where the openings in the network are identified by which entry is being made in the systems on the network to check what kind of vulnerabilities are there. This can be done locally or remotely.

Penetration Testing Types Cont..

5) Client-side Test: It aims to search and exploit vulnerabilities in client-side software programs.

6) Remote dial-up war dial: It searches for modems in the environment and tries to log in to the systems connected through these modems by password guessing or brute-forcing.

7) Wireless Security Test: It discovers open, unauthorized and less secure hotspots or Wi-Fi networks and connects through them.

Types of penetration testing into three parts

types of penetration testing into three parts

- **Black Box Penetration Testing**: In this approach, the tester assesses the target system, network or process without the knowledge of its details. They just have a very high level of inputs like URL or company name using which they penetrate the target environment. No code is being examined in this method.
- **White Box Penetration Testing**: In this approach, the tester is equipped with complete details about the target environment – Systems, network, OS, IP address, source code, schema, etc. It examines the code and finds out design & development errors. It is a simulation of an internal security attack.
- **Grey Box Penetration Testing**: In this approach, the tester has limited details about the target environment. It is a simulation of external security attacks.

Pen Testing Techniques

- Manual Penetration Test
- Using automated penetration testing tools.
- Combination of both manual and automated processes.

Pen Testing Techniques Cont..

Manual Penetration Test:

It’s difficult to find all vulnerabilities using automated tools. There are some vulnerabilities that can only be identified by manual scan. Penetration testers can perform better attacks on applications based on their skills and knowledge of the system being penetrated.

Pen Testing Techniques Cont..

Penetration Test Process:

Let’s discuss the actual process followed by test agencies or penetration testers. Identifying vulnerabilities present in the system is the first important step in this process. Corrective action is taken on this vulnerability and the same penetration tests are repeated until the system is negative to all those tests.

We can categorize this process in the following methods

1) **Data Collection:** Various methods including Google search are used to get target system data. One can also use the web page source code analysis technique to get more info about the system, software and plugin versions.

2) **Vulnerability Assessment:** Based on the data collected in the first step, one can find the security weakness in the target system. This helps penetration testers to launch attacks using identified entry points in the system.

We can categorize this process in the following methods

3) Actual Exploit: This is a crucial step. It requires special skills and techniques to launch an attack on the target system. Experienced penetration testers can use their skills to launch an attack on the system.

4) Result in analysis and report preparation: After completion of penetration tests, detailed reports are prepared for taking corrective actions. All identified vulnerabilities and recommended corrective methods are listed in these reports. You can customize the vulnerability report format (HTML, XML, MS Word or PDF) as per your organization’s needs.

Conclusion

- Finally, as a penetration tester, you should collect and log all vulnerabilities in the system. Don’t ignore any scenario considering that it won’t be executed by the end-users.
- *If you are a penetration tester, please help our readers with your experience, tips, and sample test cases on how to perform Penetration Testing effectively.*

Question Please?

THANK YOU

VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

Chapter 2: **Sniffers**

By
Dr. Ravi Verma

Outcome of Session

- **Students will be able to :**
- Explore the concept of Network Trapping through different Sniffing Tools and Techniques .
- Can list the tools used for Sniffing the network traffic as well as monitoring network activities.

Contents To Be Discuss

- Sniffing
- What can be Sniffed?
- How it works?
- Sniffing the Networks
- Types of Sniffing
- Active Sniffing
- Passive Sniffing
- Protocols which are affected
- Hardware Protocols Analyzer
- Lawful Inspections
- Sniffing Tools

Sniffing

- Sniffing is the process of monitoring and capturing all the packets passing through a given network using sniffing tools. It is a form of “tapping phone wires” and get to know about the conversation. It is also called **wiretapping** applied to the computer networks.

Sniffing Cont..

- In other words, Sniffing allows you to see all sorts of traffic, both protected and unprotected. In the right conditions and with the right protocols in place, an attacking party may be able to gather information that can be used for further attacks or to cause other issues for the network or system owner.

What can be sniffed?

- One can sniff the following sensitive information from a network –
- Email traffic
- FTP passwords
- Web traffics
- Telnet passwords
- Router configuration
- Chat sessions
- DNS traffic

How it works?

- A sniffer normally turns the NIC of the system to the **promiscuous mode** so that it listens to all the data transmitted on its segment.
- Promiscuous mode refers to the unique way of Ethernet hardware, in particular, network interface cards (NICs), that allows an NIC to receive all traffic on the network, even if it is not addressed to this NIC.

How it works? Cont..

- By default, a NIC ignores all traffic that is not addressed to it, which is done by comparing the destination address of the Ethernet packet with the hardware address (a.k.a. MAC) of the device. While this makes perfect sense for networking, non-promiscuous mode makes it difficult to use network monitoring and analysis software for diagnosing connectivity issues or traffic accounting.

Sniffing the Networks

A sniffer can continuously monitor all the traffic to a computer through the NIC by decoding the information encapsulated in the data packets.

Types of Sniffing

Passive Sniffing

In passive sniffing, the traffic is locked but it is not altered in any way. Passive sniffing allows listening only. It works with Hub devices. On a hub device, the traffic is sent to all the ports. In a network that uses hubs to connect systems, all hosts on the network can see the traffic. Therefore, an attacker can easily capture traffic going through.

The good news is that hubs are almost obsolete nowadays. Most modern networks use switches. Hence, passive sniffing is no more effective.

Types of Sniffing Cont..

- **Active Sniffing**
- In active sniffing, the traffic is not only locked and monitored, but it may also be altered in some way as determined by the attack. Active sniffing is used to sniff a switch-based network. It involves injecting **address resolution packets** (ARP) into a target network to flood on the switch **content addressable memory** (CAM) table. CAM keeps track of which host is connected to which port.

Active Sniffing

- Following are the Active Sniffing Techniques −
- MAC Flooding
- DHCP Attacks
- DNS Poisoning
- Spoofing Attacks
- ARP Poisoning

Protocols which are affected

- Protocols such as the tried and true TCP/IP were never designed with security in mind and therefore do not offer much resistance to potential intruders. Several rules lend themselves to easy sniffing −
- **HTTP** − It is used to send information in the clear text without any encryption and thus a real target.
- **SMTP** (Simple Mail Transfer Protocol) − SMTP is basically utilized in the transfer of emails. This protocol is efficient, but it does not include any protection against sniffing.

Protocols which are affected Cont..

- **NNTP** (Network News Transfer Protocol)− It is used for all types of communications, but its main drawback is that data and even passwords are sent over the network as clear text.
- **POP** (Post Office Protocol) − POP is strictly used to receive emails from the servers. This protocol does not include protection against sniffing because it can be trapped.
- **FTP** (File Transfer Protocol) − FTP is used to send and receive files, but it does not offer any security features. All the data is sent as clear text that can be easily sniffed.

Protocols which are affected Cont..

- **IMAP** (Internet Message Access Protocol) − IMAP is same as SMTP in its functions, but it is highly vulnerable to sniffing.
- **Telnet** − Telnet sends everything (usernames, passwords, keystrokes) over the network as clear text and hence, it can be easily sniffed.

Sniffers are not the dumb utilities that allow you to view only live traffic. If you really want to analyze each packet, save the capture and review it whenever time allows.

Hardware Protocol Analysers

Before we go into further details of sniffers, it is important that we discuss about **hardware protocol analyzers**. These devices plug into the network at the hardware level and can monitor traffic without manipulating it.

Hardware protocol analyzers are used to monitor and identify malicious network traffic generated by hacking software installed in the system.

Hardware Protocol Analysers

- They capture a data packet, decode it, and analyze its content according to certain rules.
- Hardware protocol analyzers allow attackers to see individual data bytes of each packet passing through the cable.

These hardware devices are not readily available to most ethical hackers due to their enormous cost in many cases.

Lawful Interception

- Lawful Interception (LI) is defined as legally sanctioned access to communications network data such as telephone calls or email messages. LI must always be in pursuance of a lawful authority for the purpose of analysis or evidence. Therefore, LI is a security process in which a network operator or service provider gives law enforcement officials permission to access private communications of individuals or organizations.

Lawful Interception

- Almost all countries have drafted and enacted legislation to regulate lawful interception procedures; standardization groups are creating LI technology specifications. Usually, LI activities are taken for the purpose of infrastructure protection and cyber security. However, operators of private network infrastructures can maintain LI capabilities within their own networks as an inherent right, unless otherwise prohibited.
-

Sniffing Tools

- There are so many tools available to perform sniffing over a network, and they all have their own features to help a hacker analyze traffic and dissect the information. Sniffing tools are extremely common applications. We have listed here some of the interesting ones −
- **BetterCAP** − BetterCAP is a powerful, flexible and portable tool created to perform various types of MITM attacks against a network, manipulate HTTP, HTTPS and TCP traffic in real-time, sniff for credentials, and much more.

Sniffing Tools Cont..

- **Ettercap** − Ettercap is a comprehensive suite for man-in-the-middle attacks. It features sniffing of live connections, content filtering on the fly and many other interesting tricks. It supports active and passive dissection of many protocols and includes many features for network and host analysis.
- **Wireshark** − It is one of the most widely known and used packet sniffers. It offers a tremendous number of features designed to assist in the dissection and analysis of traffic.
- **Tcpdump** − It is a well-known command-line packet analyzer. It provides the ability to intercept and observe TCP/IP and other packets during transmission over the network. Available at www.tcpdump.org.

Sniffing Tools

- **WinDump** − A Windows port of the popular Linux packet sniffer tcpdump, which is a command-line tool that is perfect for displaying header information.
- **OmniPeek** − Manufactured by WildPackets, OmniPeek is a commercial product that is the evolution of the product EtherPeek.
- **Dsniff** − A suite of tools designed to perform sniffing with different protocols with the intent of intercepting and revealing passwords. Dsniff is designed for Unix and Linux platforms and does not have a full equivalent on the Windows platform.

Sniffing Tools

- **EtherApe** − It is a Linux/Unix tool designed to display graphically a system's incoming and outgoing connections.
- **MSN Sniffer** − It is a sniffing utility specifically designed for sniffing traffic generated by the MSN Messenger application.
- **NetWitness NextGen** − It includes a hardware-based sniffer, along with other features, designed to monitor and analyze all traffic on a network. This tool is used by the FBI and other law enforcement agencies.
- A potential hacker can use any of these sniffing tools to analyze traffic on a network and dissect information.

Question Please?

THANK YOU

VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

UNIT 02

LECTURE 04 Installation of VMware & Kali Linux for Pen Testing

By

Dr. Ravi Verma

Output of this Presentation

- Students will be able to Install various tools and software like VMware, Kali Linux to perform Pen Testing .

Contents To Be Discuss

- What is Virtualization
- How does Virtualization works
- What is Virtual Machine
- Why use Virtual MACHINE
- Installation of Vmware
- What is Kali Linux ?
- Kali Linux Utalities
- Who use Kali Linux and Why ?
- Downloading of Kali Linux ISO Image

Contents To Be Discuss Cont. ...

- **Installation of Kali Linux Image on VMware machine.**

WHAT IS VIRTUALIZATION

- Virtualization is the technology that transforms hardware into software's, Virtualization allows you to run multiple operating systems as a virtual machines on a single computer.
- Each copy of an operating system installed into an virtual machine.

How does Virtualization works

- A virtualization layer is installed. It uses either as a hosted or a bare metal hypervisor architecture

What is Virtual Machine

- A virtual machine is a software platform that like a physical computer, runs an operating system and applications .

Why use virtual machine

Physical Machine

- Difficult to move or copy beyond to specific set of hardware components.
- Require personal interaction to upgrade hardware.

Virtual machine

- Easy to move and copy
- Easy to mange.
- Independent from physical dependency to upgrade hardware and software.

Installation of VMware

- Search on google:- Download VMware workstation

Select VMware Workstation 16 Pro for download

VMware Workstation 16 Pro

vmware WORKSTATION PRO™ 16

Workstation 16 Pro improves on the industry defining technology with DirectX 11 and OpenGL 4.1 3D Accelerated graphics support, a new dark mode user interface, support for Windows Hyper-V mode on Windows 10 version 2004 and greater hosts, a new CLI for supporting containers and Kubernetes clusters, 'vctl,' support for the latest Windows and Linux operating systems, and more.

Use the links below to start your free, fully functional 30-day trial, no registration required.

Workstation 16 Pro for Windows

DOWNLOAD NOW >

Workstation 16 Pro for Linux

DOWNLOAD NOW >

Run Setup File

Run Setup Cont..

Run Setup Cont..

Run Setup Cont..

Run Setup Cont..

Run Setup Cont..

Run Setup Cont..

What is Kali Linux?

- **Kali Linux** is a security distribution of Linux derived from Debian and specifically designed for computer forensics and advanced penetration testing. It was developed through rewriting of BackTrack by Mati Aharoni and Devon Kearns of Offensive Security. **Kali Linux** contains several hundred tools that are well-designed towards various information security tasks, such as penetration testing, security research, computer forensics and reverse engineering.

Useful utilities of Kali Linux

1. Information Gathering
2. Vulnerability Analysis
3. Wireless Attacks
4. Web Applications
5. Exploitation Tools
6. Stress Testing
7. Forensics Tools

Useful utilities of Kali Linux Cont..

8. Sniffing & Spoofing
9. Password Attacks
10. Maintaining Access
11. Reverse Engineering
12. Reporting Tools
13. Hardware Hacking

Who uses Kali Linux and Why?

- Kali Linux is truly a unique operating system, as its one of the few platforms openly used by both good guys and bad guys. Security Administrators, and Black Hat Hackers both use this operating system extensively.
- **Security Administrators –**Security Administrators are responsible for safeguarding their institution's information and data. They use Kali Linux to review their environment(s) and ensure there are no easily discoverable vulnerabilities.
- **Network Administrators** –Network Administrators are responsible for maintaining an efficient and secure network. They use Kali Linux to audit their network. For example, Kali Linux has the ability to detect rogue access points.

Who uses Kali Linux and Why? Cont..

- **Network Architects –** Network Architects, are responsible for designing secure network environments. They utilize Kali Linux to audit their initial designs and ensure nothing was overlooked or misconfigured.
- **Pen Testers –** Pen Testers, utilize Kali Linux to audit environments and perform reconnaissance on corporate environments which they have been hired to review.
- **Forensic Engineers –** Kali Linux posses a “Forensic Mode”, which allows a Forensic Engineer to perform data discovery and recovery in some instances.

Downloading Iso Image

- After selecting 64 bit ISO image the image will be downloaded in your PC which you can extract using extractor like 7z [exactor]. You will found ISO image in that .

- **"C:\Users\Dr.Ravi Verma\Downloads\kali-linux-2021.4a-vmware-amd64.7z.torrent"**

Kali Linux Installation Methods

- Kali Linux can be installed using the following methods:
- Directly on a PC, Laptop – Utilizing a Kali ISO image, Kali Linux can be installed directly onto a PC or Laptop. This method is best if you have a spare PC and are familiar with Kali Linux. Also, if you plan or doing any access point testing, installing Kali Linux directly onto Wi-Fi enabled laptop is recommended.
- Virtualized (VMware, Hyper-V, Oracle VirtualBox, Citrix) – Kali Linux supports most known hypervisors and can be easily into the most popular ones. Pre-configured images are available for download from https://www.kali.org/, or an ISO can be used to install the operating system into the preferred hypervisor manually.

Download Kali Linux ISO Image for VMware

- One can dowloan Kali Linux Iso image using following link.
- https://www.kali.org
- Now choose Virtual Machines option so click on it.

Virtual Machine

Kali Linux VMware & VirtualBox images are available for users who prefer, or whose specific needs rec
virtual machine installation.

These images have the default credentials **"kali/kali"**.

Virtual Machines Documentation ›

64-bit 32-bit

VMware

2.4G torrent sum

Documentation

VirtualBox

3.7G torrent sum

Documentation

Installing Kali Linux on VMware workstation

Installing ISO image into VMware

- Now click on New Virtual Machine, then select Typical then Next

Installing ISO image into VMware Cont..

- Now you have to select Installer image option and select your downloaded Iso images .

Installing ISO image into VMware Cont..

- Now we need to select Linux operating system from

Installing ISO image into VMware Cont..

- Here you have to assign hardware resources to system . Ideally it would be:
- Disk Storage: 128 GB
- RAM: 4 GB
- Processor Core: 2 to 4

Installing ISO image into VMware Cont..

- One can update the hardware configuration by clicking on customize hardware option

Customize Hardware

Kali Linux

- After click on Finish one find following dialog box. Now one need to click on Power on option.

Now Click on Graphical Install Option

Finish Installation Process

- After selecting graphical Installer setup will install few software's and tools and finally you find Kali Linux has been installed in your machine .
- After getting on your kali machine you need to enter following ID and psd here.
- ID= Kali
- Psd= Kali

Conclusion

- Presentation brings various steps used to install Kali Linux into Virtual Machine to perform penetration testing.

Question ?

VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

UNIT 02
Pen Testing with Metasploit

By
Dr. Ravi Verma

Output of this Presentation

- Students will be able to Install various tools and software like VMware, Kali Linux to perform Pen Testing .
- This session will brings the concept of realizing how to conduct pen testing on Metasploit.

Contents To Be Discuss Cont

Today's Agenda

- Introduction To Metasploit
- History Of Metasploit
- Prepare The Metasploit Lab Environment
- Metasploit Architecture
- Metasploit Modules

Introduction to metasploit

- Metasploit is a popular penetration testing tool
- A tool for developing and executing exploit code against a remote target machine
- Offer a broad platform for pen-testing and exploit development

Msfconsole

- The msfconsole is the most popular interface to the Metasploit Framework (MSF)
- Execution of external commands in msfconsole is possible

History of metasploit

- Undertaken in 2003 by H.D. Moore
- Perl-based portable network tool
- Later rewritten in Ruby by 2007
- Rapid7 purchased the Metasploit Project in 2009

Prepare the metasploit lab

- Vmware
- Kali Linux
- Metasploitable
- Windows

Architecture of Metasploit

Architecture of Metasploit Cont..

Metasploit Modules

- A module is a piece of software that the Metasploit Framework uses to perform a task, such as exploiting or scanning a target.
  - Exploits
  - Auxiliary
  - Payloads
  - Encoders
  - Nops
  - Evasion
  - Post

Metasploit Modules

- Exploits
  - An exploit executes a sequence of commands that target a specific vulnerability found in a system
- Auxiliary
  - Auxiliary modules include port scanners, fuzzers, sniffers, and more
- Payloads
  - Payloads consist of code that runs remotely

Metasploit Modules

- Encoders
  - Encoders ensure that payloads make it to their destination intact
- Nops
  - Nops keep the payload sizes consistent across exploit attempts
- Evasion
  - These new modules are designed to help you create payloads that can evade anti-virus (AV) on the target system

Metasploit Modules

- Post
  - Post-exploitation modules that can be run on compromised targets to gather evidence, pivot deeper into a target network, and much more.

Conclusion

- Presentation brings various modules used in Metasploit platform to conduct a Pen Testing for Network and other Computing Devices.

Question ?

Download Extrapolatable as second machine for

- Metasploitable download | SourceForge.net

Installation

- After downgliding file need to extract it .
- Now open VMware workstation and choose Open virtual machine option. Then select file from extracted folder.

VIT UNIVERSITY
SCHOOL OF COMPUTER SCIENCE ENGINEERING

UNIT 02
ARP SPOOFING

By
Dr. Ravi Verma

Output of this Presentation

- Students will be able to perform different attacking activities like Man in the Middle attack for realizing the impact of ARP Spoofing and MTMA.

Contents To Be Discuss Cont….

- ARP Poisoning or Spoofing
- What Is an ARP?
- An ARP should
- ARP Attacks
- Goal
- Known ARP Vulnerabilities
- ARP Poisoning Attack Prevention
- Conclusion

ARP Poisoning or Spoofing

- ARP poisoning (also known as ARP spoofing) is a cyber attack carried out through malicious ARP messages.
- An ARP attack is difficult to detect, and once it's in place, the impact is impossible to ignore.

ARP Poisoning or Spoofing Cont..

- A hacker that successfully implements either ARP spoofing or ARP poisoning could gain control of every document on your network. You could be subject to spying, or your traffic could grind to a halt until you give the hacker what's requested for ransom.

ARP Poisoning or Spoofing Cont..

What Is an ARP?

- In 2001, developers introduced the address resolution protocol (ARP) to Unix developers. At the time, they described it as a "workhorse" that could establish IP-level connections to new hosts.
- The work is critical, especially if your network is constantly growing, and you need a way to add new functionality without authorizing each request yourself.

What is ARP? Cont..

- The basis of ARP is media access control (MAC). As experts explain, an MAC is a unique, hardware-level address of an ethernet network interface card (NIC). These numbers are assigned at the factory, although they can be changed by software.

An ARP should:

- **Accept requests.** A new device asks to join the local area network (LAN), providing an IP address.
- **Translate.** Devices on the LAN don't communicate via IP address. The ARP translates the IP address to a MAC address.
- **Send requests.** If the ARP doesn't know the MAC address to use for an IP address, it sends an ARP packet request, which queries other machines on the network to get what's missing.

ARP Attacks: Key Definitions

- A malicious developer, hoping to gain access to important data, could expose vulnerabilities and sneak inside, and you may never know it's happening.
- Two types of ARP attacks exist.
- **ARP spoofing:** A hacker sends fake ARP packets that link an attacker's MAC address with an IP of a computer already on the LAN.
- **ARP poisoning:** After a successful ARP spoofing, a hacker changes the company's ARP table, so it contains falsified MAC maps. The contagion spreads.

Goal

- The goal is to link a hacker's MAC with the LAN. The result means any traffic sent to the compromised LAN will head to the attacker instead.
- At the end of a successful ARP attack, a hacker can:
- **Hijack.** Someone may look over everything that heads to the LAN before releasing it.
- **Deny service.** Someone may refuse to release anything from the infected LAN unless some kind of ransom is paid.

Goal Cont..

- **Sit in the middle.** Someone conducting a man-in-the-middle attack can do almost anything, including altering documents before sending them out. These attacks both threaten confidentiality and reduce user confidence. They are among the most dangerous attacks anyone can perpetrate.
- If a hacker wants to take over an end host, the work must be done quickly. ARP processes expire within about 60 seconds. But on a network, requests can linger for up to 4 hours. That leaves plenty of time for a hacker to both contemplate and execute an attack.

Known ARP Vulnerabilities

- Speed, functionality, and autonomy were the goals when ARP was developed. The protocol wasn't made with security in mind, and it's proven very easy to spoof and tweak for malicious ends.
- A hacker needs just a few tools to make this work.
- **Connection:** The attacker needs control of one LAN-connected machine. Better yet, the hacker is directly connected to the LAN already.

Known ARP Vulnerabilities

- **Coding skills:** The hacker must know how to write up ARP packets that are immediately accepted or stored on the system.
- **Outside tools:** A hacker could use a spoofing tool, such as Arpspoof, to send out falsified or otherwise inauthentic ARP responses.
- **Patience.** Some hackers breeze into systems quickly. But others must send dozens or even hundreds of requests before they fool the LAN.

Known ARP Vulnerabilities

- ARP is stateless, and networks tend to cache ARP replies. The longer they linger, the more dangerous they become. One leftover reply could be used in the next attack, which leads to ARP poisoning.
- No method of identity proofing exists in a traditional ARP system. Hosts can't determine if packets are authentic, and they can't even determine where they came from.

ARP Poisoning Attack Prevention

- Hackers use a predictable series of steps to take over a LAN. They send a spoofed ARP packet, they send a request that connects to the spoof, and they take over. The request is broadcast to all computers on the LAN, and control is complete.

ARP Poisoning Attack Prevention

- Network administrators can use two techniques to detect ARP spoofing.

1. **Passive:** Monitor ARP traffic and look for mapping inconsistencies.
2. **Active:** Inject falsified ARP packets into the network. A spoofing attack like this helps you identify weak points in your system. Remediate them quickly, and you could stop an attack in progress.

Protection Tools to Consider

- Plenty of companies provide monitoring programs you can use to both oversee your network and spot ARP problems.
- These are common solutions:
- Arpwatch**:** Monitor ethernet activity, including changing IP and MAC addresses, via this Linux tool. Look over the log every day, and access timestamps to understand just when the attack happened.

ARP Poisoning Attack Prevention Cont..

- ARP-GUARD**:** Tap into a graphic overview of your existing network, including illustrations of switches and routers. Allow the program to develop an understanding of what devices are on your network and build rules to control future connections.
- XArp**:** Use this tool to detect attacks happening below your firewall. Get notified as soon as an attack begins, and use the tool to determine what to do next.

ARP Poisoning Attack Prevention Cont..

- **Wireshark:** Use this tool to develop a graphic understanding of all the devices on your network. This tool is powerful, but you may need advanced skills to implement it properly.
- **Packet filtering:** Use this firewall technique to manage network access by monitoring incoming and outgoing IP packets. Packets are allowed or stopped based on source and destination IP addresses, ports, and protocols.
- **Static ARP:** These ARPs are added to the cache and retained on a permanent basis. These will serve as permanent mappings between MAC addresses and IP addresses.

11 Types of Spoofing Attacks Every Security Professional Should Know About

1. ARP Spoofing

- This one is a common source of man-in-the-middle attacks. To execute it, a cybercriminal inundates a local area network with falsified Address Resolution Protocol (ARP) packets in order to tamper with the normal traffic routing process. The logic of this interference boils down to binding the adversary’s MAC address with the IP address of the target’s default LAN gateway.

11 Types of Spoofing Attacks Every Security Professional Should Know About

2. MAC Spoofing

In theory, every network adapter built into a connected device should have a unique Media Access Control (MAC) address that won’t be encountered elsewhere. In practice though, a clever hack can turn this state of things upside down. An attacker may harness imperfections of some hardware drivers to modify, or spoof, the MAC address. This way, the criminal masquerades his device as one enrolled in a target network to bypass traditional access restriction mechanisms.

11 Types of Spoofing Attacks Every Security Professional Should Know About

3. IP Spoofing

- To perform this attack, the adversary sends Internet Protocol packets that have a falsified source address. This is a way to obfuscate the actual online identity of the packet sender and thereby impersonate another computer. IP spoofing is often used to set DDoS attacks in motion. The reason is that it’s hard for digital infrastructure to filter such rogue packets, given that each one appears to hail from a different address and therefore the crooks feign legitimate traffic quite persuasively.

11 Types of Spoofing Attacks Every Security Professional Should Know About

4. DNS Cache Poisoning (DNS Spoofing)

- Every tech-savvy user knows the Domain Name Server (DNS) wiki: it maps domain names to specific IP addresses so that people type easy-to-remember URLs in the browser rather than enter the underlying IP strings. Threat actors may be able to contort this mapping logic by piggybacking on known DNS server caching flaws. As a result of this interference, the victim runs the risk of going to a malicious replica of the intended domain. From a cybercriminal’s perspective, that’s a perfect basis for phishing hoaxes that look really true-to-life.

Conclusion

- This Presentation describes the ARP and Its various types .
- This presentation also brings the Effect of ARP or Man in the Middle Attack in Network .

Question ?

Advance Linux Exploits

Here, we discuss the following topics:

• Format string exploits

• Memory protection schemes

Format string exploits

Format String Exploits Format string exploits became public in late 2000. Unlike buffer overflows, format string errors are relatively easy to spot in source code and binary analysis. In spite of this, they are still common in applications today. Many organizations still don't utilize code analysis or binary analysis tools on software before releasing it, so these errors still occur in the wild. Once spotted, they are usually eradicated quickly. As more organizations start to use code analysis tools as part of their build processes, these types of attacks will continue to decline. However, this attack vector is still fairly easy to find and can result in some interesting code execution.

Format Strings Format strings are used by various print functions, and these functions may behave in many ways depending on the format strings provided.

- printf() Prints output to the standard input/output (STDIO) handle (usually the screen)
    - fprintf() Prints output to a file stream
    - sprintf() Prints output to a string
    - snprintf() Prints output to a string with length checking built in

When someone calls one of these functions, the format string dictates how and where the data is compiled into the final string. Format strings are very versatile, though, and if the creator of the application allows data specified by the end user to be used directly in one of these format strings, the user can change the behavior of the application. This can include disclosing additional information that the creator did not want disclosed, such as memory locations, data variables, and stack memory.

The Problem

```
printf(<format string>, <list of variables/values>);
printf(<user supplied string>);
```

the printf() function can have any number of arguments. We will discuss two forms here: In the first example, the programmer has specified a format string and then the variables that will fill in the spaces designated by the format string for data. This prevents unexpected behavior from printf. The second example allows the user to specify the format string. This means that a user can cause the printf function to behave however they want.

Format Symbol	Meaning	Example
\n	Carriage return/newline	**printf("test\n");** Result: The application prints **test**.
%d	Decimal value	**printf("test %d", 123);** Result: The application prints **test 123.**
%s	String value	**printf("test %s", "123");** Result: The application prints **test 123.**
%x	Hex value	**printf("test %x", 0x123);** Result: The application prints **test 123.**
%hn	Print the length of the current string in bytes to **var** (short int value, overwrites 16 bits)	**printf("test %hn", var);** Result: The value **04** is stored in **var** (that is, 2 bytes).
%<number>$	Direct parameter access	**printf("test %2$s", "12", "123");** Result: **test 123** (the second parameter is used directly and then treated as a string).

<u>The Correct Way</u>

Recall the correct way to use the printf() function.

For example, the code

```
//fmt1.c
#include <stdio.h>
int main() {
  printf("This is a %s.\n", "test");
  return 0;
}
```

produces the following output:

```
#gcc -o fmt1 fmt1.c
#./fmt1
This is a test.
```

<u>The Incorrect Way</u>

Now take a look at what happens if we forget to add a value for %s to replace:

```
// fmt2.c
#include <stdio.h>
int main() {
   printf("This is a %s.\n");
   return 0;
}

# gcc -o fmt2 fmt2.c
#./fmt2
This is a ey¿.
```

```
//fmt4.c
#include <stdio.h>
int main(){
        int one=1, two=2, three=3;
        printf("Testing %d, %d, %d!\n", one, two, three);
        return 0;
}
$gcc -o fmt4.c
./fmt4
Testing 1, 2, 3!
```

During execution of the **printf()** function, the stack looks like Figure 12-1. As always, the parameters of the **printf()** function are pushed on the stack in reverse order, as shown in the figure. The addresses of the parameter variables are used. The **printf()** function maintains an internal pointer that starts out pointing to the format string (or top of the stack frame) and then begins to print characters of the format string to the STDIO handle (the screen in this case) until it comes upon a special character.

printf() function maintains an internal pointer that starts out pointing to the format string (or top of the stack frame) and then begins to print characters of the format string to the STDIO handle (the screen in this case) until it comes upon a special character. If the % is encountered, the printf() function expects a format token to follow and thus increments an internal pointer (toward the bottom of the stack frame) to grab input for the format token (either a variable or absolute value).

Therein lies the problem: the printf() function has no way of knowing if the correct number of variables or values were placed on the stack for it to operate. If the programmer is sloppy and does not supply the correct number of arguments, or if the user is allowed to present their own format string, the function will happily move down the stack (higher in memory), grabbing the next value to satisfy the format string requirements. So what we saw in our previous examples was the printf() function grabbing the next value on the stack and returning it where the format token required.

Reading from Arbitrary Memory

We will now begin to take advantage of the vulnerable program. We will start slowly and then pick up speed. Using the %x Token to Map Out the Stack As shown in previous Table, the %x format token is used to provide a hex value. So, by supplying a few %08x tokens to our vulnerable program, we should be able to dump the stack values to the screen:

$ ./fmtstr "AAAA %08x %08x %08x %08x"

AAAA bffffd2d 00000648 00000774 41414141

Canary at 0x08049440 = 0x00000000

$

The 08 is used to define precision of the hex value (in this case, 8 bytes wide). Notice that the format string itself was stored on the stack, proven by the presence of our AAAA (0x41414141) test string. The fact that the fourth item shown (from the stack) was our format string depends on the nature of the format function used and the location of the vulnerable call in the vulnerable program. To find this value, simply use brute force and keep increasing the number of %08x tokens until the beginning of the format string is found. For our simple example (fmtstr), the distance, called the offset, is defined as 4.

Using the %s Token to Read Arbitrary Strings

Because we control the format string, we can place anything in it we like (well, almost anything). For example, if we wanted to read the value of the address located in the fourth parameter, we could simply replace the fourth format token with a %s, as shown:

```
$ ./fmtstr "AAAA %08x %08x %08x %s"
```

Because, as you recall, the %s format token will take the next parameter on the stack, in this case the fourth one, and treat it like a memory address to read from (by reference). In our case, the fourth value is AAAA, which is translated in hex to 0x41414141.

Social Engineering in Kali Linux

The term **"social engineering"** is derived from the words **"social"** and **"engineering,"** where **"social"** refers to personal, professions, and our day-in-day-out lives. On the other hand, **"engineering"** involves comprehensive processes to complete a work such that the defined goal is met. In other words, it is a set of methods.

When social and engineering is combined, we get social engineering, which involves intrusion based on human interaction. It is a non-technical intrusion in which a person is often tricked into breaking the general security guidelines already set in an institution.

Social Engineering Toolkit

Social engineering toolkit is a **free and open-source tool** which is used for social engineering attacks like **phishing, sending SMS, faking phone,** etc. It is a free tool that comes with Kali Linux, or we can download and install it directly from **Github.** The Social Engineering Toolkit is designed and developed by a programmer named **Dave Kennedy.** Security researchers and penetration testers use this tool to check cybersecurity issues in systems all over the world. The goal of the social engineering toolkit is to perform attacking techniques on their machines. This toolkit also includes website vector attacks and custom vector attacks, which allow us to **clone any website, perform phishing attacks.**

Features of Social Engineering Toolkit

The following are the features of the social engineering toolkit:

- Social Engineering Toolkit is **free** and **open source.**
- Social Engineering Toolkit is portable, which means we can quickly switch attack vectors.
- Social Engineering Toolkit supports integration with third-party modules.
- Social Engineering Toolkit is already installed in our Kali Linux, but we can also download and install it from **Github.**
- Social Engineering Toolkit is a **multi-platform** tool; we can run it in **Windows, Linux,** and **Unix.**
- Social Engineering Toolkit contains access to the **Fast-Track Penetration Testing platform.**
- Social Engineering Toolkit offers various attack vectors like **Website Attacks, Infection Media Generator, Website Attacks,** etc.

Uses of Social Engineering Toolkit

There are various uses of social engineering toolkit:

1. Web Attack
2. Mass Mailer Attack
3. Phishing Attacks
4. Create a Payload and Listener

1. Web Attack

In SET, a web attack is a module. This module combines various options to attack the victim remotely. Using this module, we can create a payload and distribute the payload to our victim browser using the **Metasploit browser exploit.** Web attack has **Credential Harvester method** that allows us to clone any website for a phishing attack and send the link of that webpage to the victim to get information from user and password fields.

2. Phishing Attacks

We can use the Social Engineering Toolkit to perform phishing attacks on our victims. Using SET, we can create phishing pages for a variety of websites, including **Google, Facebook, Instagram,** etc. SET will generate a link of the option which we have selected, and then we can send that URL to the victim once the victim clicks on that URL and he/she will see a

legitimate webpage of a real website that is essentially a phishing page. Once he/she has entered his/her ID password, we will get that ID password on our terminal screen, this is how a phishing attack using SET works.

3. Create a Payload and Listener

When we execute the Social Engineering Toolkit for the first time. We will see **option 4th** which is used to generate a payload and listener by using that SET module, we may develop malicious payloads for Windows, including **Shell Reverse_TCP, Shell Reverse_ HTTPS TCP X64, Meterpreter Reverse,** and **Reverse_TCP Meterpreter.** These payloads can be used in the same way that we use metasploitable payloads.

4. Mass Mailer Attack

In the social engineering toolkit, mass mailer attacker is a module which we used to send a large number of emails on target mail account, for which we can also use our own Gmail account or we can own a server for that.

These are some of the attack vectors which we can use using the Social Engineering Toolkit. When we will run the SET, we will enjoy it because it is quite simple to use.

Phases in Social Engineering

There are various phases of social engineering before the final result is obtained. This includes:

1. Research Phase
2. Hook Phase
3. Play Phase
4. Exit Phase

1. Research Phase

In the research phase, the information related to the goal is collected. Whether the objective is a firm or an individual, the first phase is the same. There are so many ways by which attackers can get the information related to their targets. These include obtaining documents from the public domain, visiting the website for the institution concerned, and in some cases, constructive face-to-face interactions. Besides, dumpster diving is also necessary at this stage of the attack.

2. Hook Phase

The Hook phase is the second phase of the attack. In this phase, the attacker initiates a discussion with their victim target.

After the hook, the phase is the phase of play that strengthens the connection between the attacker and the target. The attacker takes advantage of this opportunity to investigate getting the information they desire.

3. Exit Phase

This is the final phase, and the attacker must be careful not to set up a situation that would make the target suspect in any manner. The idea is to exit the target without giving any indication of action.

We can start these steps through various social engineering tools which are pre-installed in Kali Linux, while other tools need to be installed manually.

We don't have to be concerned because Social Engineering Toolkit is an open-source penetration testing platform that focuses on social engineering. This suite of tools has been fine-tuned to allow us to launch attacks in a matter of seconds.

Social Engineering Toolkit Usage

The **Social-Engineer Toolkit** (SET) is an open-source penetration testing framework designed for social engineering. SET has a number of custom attack vectors that allow you to make a believable attack in a fraction of time. These kind of tools use human behaviors to trick them to the attack vectors.

Let's learn how to use the Social Engineer Toolkit.

Step 1 − To open SET, go to Applications → Social Engineering Tools → Click "SET" Social Engineering Tool.

Step 2 − It will ask if you agree with the terms of usage. Type **"y"** as shown in the following screenshot.

Step 3 − Most of the menus shown in the following screenshot are self-explained and among them the most important is the number 1 "Social Engineering Attacks".

Step 4 − Type **"1"** → Enter. A submenu will open. If you press the **Enter** button again, you will see the explanations for each submenu.

The Spear-phishing module allows you to specially craft email messages and send them to your targeted victims with attached **FileFormatmalicious** payloads. For example, sending malicious PDF document which if the victim opens, it will compromise the system. If you want to spoof your email address, be sure "Sendmail" is installed (apt-get install sendmail) and change the config/set_config SENDMAIL=OFF flag to SENDMAIL=ON.

There are two options for the spear phishing attack −

- Perform a Mass Email Attack
- Create a FileFormat Payload and a Social-Engineering Template

The first one is letting SET do everything for you (option 1), the second one is to create your own FileFormat payload and use it in your own attack.

```
Terminal
File  Edit  View  Search  Terminal  Help
  10) Third Party Modules

  99) Return back to the main menu.

set> 1

 The Spearphishing module allows you to specially craft email messages and send
 them to a large (or small) number of people with attached fileformat malicious
 payloads. If you want to spoof your email address, be sure "Sendmail" is in-
 stalled (apt-get install sendmail) and change the config/set_config SENDMAIL=OF
F
 flag to SENDMAIL=ON.

 There are two options, one is getting your feet wet and letting SET do
 everything for you (option 1), the second is to create your own FileFormat
 payload and use it in your own attack. Either way, good luck and enjoy!

   1) Perform a Mass Email Attack
   2) Create a FileFormat Payload
   3) Create a Social-Engineering Template

  99) Return to Main Menu

set:phishing>
```

Type **"99"** to go back to the main menu and then type **"2"** to go to "The web attack vectors".

The web attack module is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim. This module is used by performing phishing attacks against the victim if they click the link. There is a wide variety of attacks that can occur once they click a link.

```
Terminal
File  Edit  View  Search  Terminal  Help
ate however when clicked a window pops up then is replaced with the malicious li
nk. You can edit the link replacement settings in the set_config if its too slow
/fast.

The Multi-Attack method will add a combination of attacks through the web attack
 menu. For example you can utilize the Java Applet, Metasploit Browser, Credenti
al Harvester/Tabnabbing all at once to see which is successful.

The HTA Attack method will allow you to clone a site and perform powershell inje
ction through HTA files which can be used for Windows-based powershell exploitat
ion through the browser.

   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) Full Screen Attack Method
   8) HTA Attack Method

  99) Return to Main Menu

set:webattack>
```

Type **"99"** to return to the main menu and then type **"3"**.

The infectious USB/CD/DVD module will create an autorun.inf file and a Metasploit payload. The payload and autorun file is burned or copied on a USB. When DVD/USB/CD is inserted in the victim's machine, it will trigger an autorun feature (if autorun is enabled) and hopefully compromise the system. You can pick the attack vector you wish to use: fileformat bugs or a straight executable.

Following are the options for Infectious Media Generator.

- File-Format Exploits
- Standard Metasploit Executable

```
set> 3

 The Infectious USB/CD/DVD module will create an autorun.inf file and a
 Metasploit payload. When the DVD/USB/CD is inserted, it will automatically
 run if autorun is enabled.

 Pick the attack vector you wish to use: fileformat bugs or a straight executabl
e.

   1) File-Format Exploits
   2) Standard Metasploit Executable

  99) Return to Main Menu

set:infectious>
```

Type **"99"** to go back to the main menu. Then, type **"4"** to go to "The web attack vectors".

The create payload and listener is a simple way to create a Metasploit payload. It will export the exe file for you and generate a listener. You would need to convince the victim to download the exe file and execute it to get the shell.

```
set> 4

   1) Windows Shell Reverse_TCP                Spawn a command shell on victim an
d send back to attacker
   2) Windows Reverse_TCP Meterpreter          Spawn a meterpreter shell on victi
m and send back to attacker
   3) Windows Reverse_TCP VNC DLL              Spawn a VNC server on victim and s
end back to attacker
   4) Windows Shell Reverse_TCP X64            Windows X64 Command Shell, Reverse
 TCP Inline
   5) Windows Meterpreter Reverse_TCP X64      Connect back to the attacker (Wind
ows x64), Meterpreter
   6) Windows Meterpreter Egress Buster        Spawn a meterpreter shell and find
 a port home via multiple ports
   7) Windows Meterpreter Reverse HTTPS        Tunnel communication over HTTP usi
ng SSL and use Meterpreter
   8) Windows Meterpreter Reverse DNS          Use a hostname instead of an IP ad
dress and use Reverse Meterpreter
   9) Download/Run your Own Executable         Downloads an executable and runs i
t

set:payloads>
```

Type **"99"** to go back to the main menu and then type **"5"** to go to "The web attack vectors".

```
set> 5

   Social Engineer Toolkit Mass E-Mailer

   There are two options on the mass e-mailer, the first would
   be to send an email to one individual person. The second option
   will allow you to import a list and send it to as many people as
   you want within that list.

   What do you want to do:

    1.  E-Mail Attack Single Email Address
    2.  E-Mail Attack Mass Mailer

    99. Return to main menu.

set:mailer>
```

The mass mailer attack will allow you to send multiple emails to victims and customize the messages. There are two options on the mass e-mailer; the first is to send an email to a single email address. The second option allows you to import a list that has all recipient emails and it will send your message to as many people as you want within that list.

- E-Mail Attack Single Email Address
- E-Mail Attack Mass Mailer

Type **“99”** to go back to the main menu and then type **“9”** to go to “Powershell Attack Vector”.

```
set> 9

The Powershell Attack Vector module allows you to create PowerShell specific att
acks. These attacks will allow you to use PowerShell which is available by defau
lt in all operating systems Windows Vista and above. PowerShell provides a fruit
ful  landscape for deploying payloads and performing functions that  do not get
triggered by preventative technologies.

   1) Powershell Alphanumeric Shellcode Injector
   2) Powershell Reverse Shell
   3) Powershell Bind Shell
   4) Powershell Dump SAM Database

  99) Return to Main Menu
```

The Powershell Attack Vector module allows you to create PowerShell specific attacks. These attacks allow you to use PowerShell, which is available by default in all operating systems Windows Vista and above. PowerShell provides a fruitful landscape for deploying payloads and performing functions that do not get triggered by preventive technologies.

- Powershell Alphanumeric Shellcode Injector
- Powershell Reverse Shell
- Powershell Bind Shell
- Powershell Dump SAM Database

Ethical Hacking

UNIT 3 – LINUX VULNERABILITIES

Objectives

- Describe the fundamentals of the Linux operating system
- Describe the vulnerabilities of the Linux operating system
- Describe Linux remote attacks
- Explain countermeasures for protecting the Linux operating system

Review of Linux Fundamentals

- Linux is a version of UNIX
  - Usually available free
  - Red Hat
    - Includes documentation and support for a fee
- Linux creates default directories

Linux File System

- Most vital part of any OS
- Provides many functions
  - Enables directories or folders organization
  - Establishes a file-naming convention
  - Includes utilities to compress or encrypt files
  - Provides for both file and data integrity
  - Enables error recovery
  - Stores information about files and folders
- *NIX systems store information about files in information nodes (inodes)

Linux File System (continued)

- Information stored in an inode
  - An inode number
  - Owner of the file
  - Group the file belongs to
  - Size of the file
  - Date the file was created
  - Date the file was last modified or read
- File systems use a fixed number of inodes
- *NIX mounts a file system as a subfile system of the root file system

Linux File System (continued)

- *mount* command is used to mount file systems
- *df* command displays the currently mounted file systems
- File system history on *NIX systems
  - Minix file system
  - Extended File System (Ext)
  - Second Extended File System (Ext2fs)
  - Third Extended File System (Ext3fs)

Linux File System Commands

- Linux has file system commands for
  - Viewing files
  - Copying files
  - Moving files

Linux File System Commands (continued)

- Many of these commands have multiple parameters and additional functionality

Linux OS Vulnerabilities

- UNIX has been around for quite some time
- Attackers have had plenty of time to discover vulnerabilities in *NIX systems
- Enumeration tools can also be used against Linux systems
- Knoppix
  - A bootable, open-source version of Linux
- Nessus can be used to enumerate Linux systems

Linux OS Vulnerabilities (continued)

- Nessus can be used to
  - Discover vulnerabilities related to SMB and NetBIOS
  - Enumerate shared resources
  - Discover the root password

Linux OS Vulnerabilities (continued)

- Test Linux computer against common known vulnerabilities
  - Review the CVE and CAN information
- Differentiate between local attacks and remote attacks
  - Remote attacks are harder to perform

Remote Access Attacks on Linux Systems

- Attacking a network remotely requires
  - Knowing what system a remote user is operating
  - The attacked system’s password and login accounts

Footprinting an Attacked System

- Footprinting techniques
  - Used to find out information about a target system
- Determining the OS version the attacked computer is running
  - Check newsgroups for details on posted messages
  - Knowing a company's e-mail address makes the search easier
- Other footprinting tools include: Whois databases, DNS zone transfers, Nessus, and port scanning tools

Using Social Engineering to Attack Remote Linux Systems

- Goal
  - To get OS information from company employees
- Common techniques
  - Urgency
  - Quid pro quo
  - Status quo
  - Kindness
  - Position
- Train your employees about social engineering techniques

Installing Trojan Programs

- Trojan programs spread as
  - E-mail attachments
  - Fake patches or security fixes that can be downloaded from the Internet
- Trojan program functions
  - Allow for remote administration
  - Create a FTP server on attacked machine
  - Steal passwords
  - Log all keys a user enters, and e-mail results to the attacker

Installing Trojan Programs (continued)

- Linux Trojan programs are sometimes disguised as legitimate programs
- Trojan programs can use legitimate outbound ports
  - Firewalls and IDSs cannot identify this traffic as malicious
  - Example: Sheepshank
- It is easier to protect systems from already identified Trojan programs
  - Trojan.Linux.JBellz
  - Remote Shell
  - Dextenea

Installing Trojan Programs (continued)

- Rootkits
  - Contain Trojan binary programs ready to be installed by an intruder with root access to the system
  - Attacker hide the tools used for later attacks
  - Replace legitimate commands with Trojan programs
  - Example: LRK5
- Security testers should check their Linux systems for rootkits
  - Rootkit Hunter
  - Chkrootkit

Creating Buffer Overflow Programs

- Buffer overflows write code to the OS's memory
  - Then run some type of program
  - Can elevate the attacker's permissions to the level of the owner
- Security testers should know what a buffer overflow program looks like

Creating Buffer Overflow Programs (continued)

- A C program that causes a buffer overflow

```
1. main()
2. {
3.     char fish_name[1]; //Assign a buffer area (array) of 1
                          //character
4.     char hawaii_fish_name[25] = "Humuhumukununukuapuaa"
5.     strcpy(fish_name, hawaii_fish_name); //Copy long name to
                                            //small area
6.             }
```

Creating Buffer Overflow Programs (continued)

- A C code snippet that fills the stack with shell code

```
    include <string.h>
    char info[] = "\xeb\x2a\x08\x09\x00\x00\x89\x05\xed\xff"
          "\xeb\x2a\x03\x09\x00\x00\x89\x05\xed\xff"
          "\xeb\x2a\x02\x09\x00\x00\x89\x05\xed\xdf"
  "\xeb\x2a\x08\x01\x01\x02\x83\x04\xef\xcf"
          "\x6e\x2a\x08\x09\x00\x00\x89\x05\xed\xff"
          "\xeb\x2a\x08\x11\x00\x00\x89\x05\xed\xae"
  "\xca\x2a\x08\x09\x00\x00\x67\xcc\xed\xef"
          "\x22\x2a\x08\xaa\x00\x00\xff\x05\xed\xff"
          "\xdd\x2a\x08\x09\x00\x00\xcc\x05\xed\xcd"
  "\xfa\x2a\x08\x09\x00\x00\x00\x11\xed\xde"
          "\xeb\x2a\x08\x09\x00\x00\x89\x05\xed\xff"
main()
{  //Contains a function that copies the info[] array into a
   //buffer area.
        [Remaining code omitted]
}
```

Creating Buffer Overflow Programs (continued)

- Guidelines to help reduce this type of attack
  - Write code that avoids functions known to have buffer overflow vulnerabilities
    - strcpy()
    - strcat()
    - sprintf()
    - gets()
  - Configure OS to not allow code in the stack to run any other executable code in the stack
  - Use compilers that warn programmers when functions listed in the first bullet are used

Using Sniffers to Gain Access to Remote Linux Systems

- Sniffers work by setting a network card adapter in promiscuous mode
  - NIC accepts all packets that traverse the network cable
- Attacker can analyze packets and learn user names and passwords
  - Avoid using protocols such as Telnet, HTTP, and FTP that send data in clear text
- Sniffers
  - Tcpdump, Ethereal

Countermeasures Against Linux Remote Attacks

- Measures include
  - User awareness training
  - Keeping current on new kernel releases and security updates

User Awareness Training

- Social Engineering
  - Users must be told not to reveal information to outsiders
  - Make customers aware that many exploits can be downloaded from Web sites
  - Teach users to be suspicious of people asking questions about the system they are using
    - Verify caller’s identity
    - Call back technique

Keeping Current

- Never-ending battle
  - New vulnerabilities are discovered daily
  - New patches are issued to fix new vulnerabilities
- Installing these fixes is essential to protecting your system
- Many OSs are shipped with automated tools for updating your systems
  - Red Hat Update Agent

Summary

- File systems store and manage user data and system data
- Linux uses default directories to store user data and system data
- Extended File System (Ext) is Linux default file system
- Information about *NIX files are stored in inodes
- Vulnerabilities of the Linux OS can be determined by the use of security tools and from the CVE Web site

Summary (continued)

- Techniques for remotely attacking Linux systems
  - Footprinting
  - Social engineering
  - Trojan programs
  - Buffer overflows
- Social engineering can be the most effective way to gather information
- Countermeasures to Trojan programs include
  - Remove any unneeded services
  - Apply test security updates

Summary (continued)

- Countermeasures to buffer overflows include
  - Writing secure code
  - Preventing code from being run in the stack
  - Using compilers that warn when a function is dangerous or risky
- Other countermeasures
  - Employee training
  - Keeping systems updated

VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

UNIT 03

Programming Survival Skills

Dr. Ravi Verma

Output of this Presentation

- Students will be able to define the basic survival problems relevant to the programming fundamentals and basics .

- This presentation will include the content of programming relevant with different types of functioning.

Programming Survival Skills

- Why study programming? Ethical gray hat hackers should study programming and learn as much about the subject as possible in order to find vulnerabilities in programs and get them fixed before unethical hackers take advantage of the
- In this chapter, we cover the following topics:
- • C programming language • Computer memory • Intel processors • Assembly language basics • Debugging with gdb • Python survival skillsm.

C Programming Language

- The C programming language was developed in 1972 by Dennis Ritchie from AT&T Bell Labs. The language was heavily used in Unix and is thereby ubiquitous. In fact, much of the staple networking programs and operating systems are based in C.

Basic C Language Construct

- **main()**
- All C programs contain a main() structure (lowercase) that follows this format:
  - <optional return value type> main (<optional arguments>)
    - {
- Optional procedural statements or functions calls
    - }

Functions

- Functions are self-contained bundles of algorithms that can be called for execution by main() or other functions. Technically, the main() structure of each C program is also a function; however,

```
<optional return value type> function name (<optional function argument>){
}
```

Variables

- Variables are used in programs to store pieces of information that may change and may be used to dynamically influence the program. Table 10-1 shows some common types of variables. When the program is compiled, most variables are pre-allocated memory of a fixed size according to system-specific definitions of size. Sizes in the table are considered typical; there is no guarantee that you will get those exact sizes. It is left up to the hardware implementation to define this size. However, the function sizeof() is used in C to ensure that the correct sizes are allocated by the compiler.

Variables Types

Variable Type	Use	Typical Size
int	Stores signed integer values such as 314 or –314	4 bytes for 32-bit machines 2 bytes for 16-bit machines
float	Stores signed floating-point numbers such as –3.234	4 bytes
double	Stores large floating-point numbers	8 bytes
char	Stores a single character such as "d"	1 byte

For example: int a = 0; where an integer (normally 4 bytes) is declared in memory with a name of a and an initial value of 0.

printf

- The C language comes with many useful constructs for free (bundled in the libc library). One of the most commonly used constructs is the printf command, generally used to print output to the screen. There are two forms of the printf command:

```
printf(String)
printf(<format of string><List of variables> )
```

Printf Cont..

Table 10-2
printf Format Symbols

Format Symbol	Meaning	Example
\n	Carriage return/new line	printf("test\n");
%d	Decimal value	printf("test %d", 123);
%s	String value	printf("test %s", "123");
%x	Hex value	printf("test %x", 0x123);

scanf

- The scanf command complements the printf command and is generally used to get input from the user. The format is as follows:
- scanf(<format of string>,<List of variables>); where the format string can contain format symbols such as those shown for printf in above Table For example, the following code will read an integer from the user and store it into the variable called number: scanf("%d", &number);

for and while Loops

- Loops are used in programming languages to iterate through a series of commands multiple times. The two common types are for and while loops. for loops start counting at a beginning value, test the value for some condition, execute the statement, and increment the value for the next iteration. The format is as follows:

- for(<Beginning value; Test Value; Change Value> ;)

{

<Statements>

; }

Therefore, a for loop like for(i=0; i<10;i++)

if/else

- The if/else construct is used to execute a series of statements if a certain condition is met; otherwise, the optional else block of statements is executed. If there is no else block of statements, the flow of the program will continue after the end of the closing if block bracket (}). The format is as follows:

if(condition) {

<statement should be execute if , if condition is met> }

Else

{statement should be execute if , if condition is met; }

Sample Progra

- You are now ready to review your first program. We will start by showing the program with // comments included, and will follow up with a discussion of the

```
//hello.c                    //customary comment of program name
#include <stdio.h>           //needed for screen printing
main ( ) {                   //required main function
    printf("Hello haxor");   //simply say hello
}                            //exit program
```

Compiling with gcc

Compiling is the process of turning human-readable source code into machine-readable binary files that can be digested by the computer and executed. More specifically, a compiler takes source code and translates it into an intermediate set of files called object code. These files are nearly ready to execute but may contain unresolved references to symbols and functions not included in the original source code file. These symbols and references are resolved through a process called linking, as each object file is linked together into an executable binary file. We have simplified the process for you here.

When programming with C on Unix systems, the compiler of choice is GNU C Compiler (gcc). gcc offers plenty of options when compiling. The most commonly used flags are listed and described in Table

Option	Description
–o <filename>	Saves the compiled binary with this name. The default is to save the output as a.out.
–S	Produces a file containing assembly instructions; saved with a .s extension.
–ggdb	Produces extra debugging information; useful when using GNU debugger (**gdb**).
–c	Compiles without linking; produces object files with a .o extension.

Computer Memory

- In the simplest terms, computer memory is an electronic mechanism that has the ability to store and retrieve data. The smallest amount of data that can be stored is 1 bit, which can be represented by either a 1 or a 0 in memory.
- There are many types of computer memory; we will focus on random access memory (RAM) and Register.

Random Access Memory (RAM)

- In RAM, any piece of stored data can be retrieved at any time—thus the term “random access.” However, RAM is volatile, meaning that when the computer is turned off, all data is lost from RAM.

Segmentation of Memory

- The basic concept is simple. Each process (oversimplified as an executing program) needs to have access to its own areas in memory. After all, you would not want one process overwriting another process’s data. So memory is broken down into small segments and handed out to processes as needed.

Programs in Memory

- When processes are loaded into memory, they are basically broken into many small sections. There are six main sections that we are concerned with, and we'll discuss them in the following sections.

Programs in Memory Cont..

- **.text Section**
- The .text section basically corresponds to the .text portion of the binary executable file. It contains the machine instructions to get the task done. This section is marked as read only and will cause a segmentation fault if written to. The size is fixed at runtime when the process is first loaded.

Programs in Memory Cont..

.data Section

The .data section is used to store global initialized variables, such as:

int a = 0;

The size of this section is fixed at runtime.

.bss Section

The below stack section (.bss) is used to store global noninitialized variables, such as: int a;

The size of this section is fixed at runtime.

Programs in Memory Cont..

Heap Section

The heap section is used to store dynamically allocated variables and grows from the lower-addressed memory to the higher-addressed memory. The allocation of memory is controlled through the malloc() and free() functions. For example, to declare an integer and have the memory allocated at runtime, you would use something like:

```
int i = malloc (sizeof (int)); //dynamically allocates an integer, contains
                               //the pre-existing value of that memory
```

Stack Section

The stack section is used to keep track of function calls (recursively) and grows from the higher-addressed memory to the lower-addressed memory on most systems.

Buffers

The term buffer refers to a storage place used to receive and hold data until it can be handled by a process. Since each process can have its own set of buffers, it is critical to keep them straight. This is done by allocating the memory within the .data or .bss section of the process’s memory

Buffers

The term buffer refers to a storage place used to receive and hold data until it can be handled by a process. Since each process can have its own set of buffers, it is critical to keep them straight. This is done by allocating the memory within the .data or .bss section of the process’s memory. Remember, once allocated, the buffer is of fixed length. The buffer may hold any predefined type of data; however, for our purpose, we will focus on string-based buffers, used to store user input and variables

Strings in Memory

- Simply put, strings are just continuous arrays of character data in memory. The string is referenced in memory by the address of the first character. The string is terminated or ended by a null character (\0 in C)
- String A[7];
-

A[0]	**A[1]**	**A[2]**	**A[3]**	**A[4]**	**A[5]**	**A[6]**	**\0**

Pointer

- Pointers are special pieces of memory that hold the address of other pieces of memory. Moving data around inside of memory is a relatively slow operation. It turns out that www.it-ebooks.info Chapter 10: Programming Survival Skills 183 PART III instead of moving data, it is much easier to keep track of the location of items in memory through pointers and simply change the pointers. Pointers are saved in 4 bytes of contiguous memory because memory addresses are 32 bits in length (4 bytes).

Pointer cont..

- **For example,**

As mentioned, strings are referenced by the address of the first character in the array. That address value is called a pointer. So the variable declaration of a string in C is written as follows:

```
char * str; //this is read, give me 4 bytes called str which is a pointer
            //to a Character variable (the first byte of the array).
```

Question ?

Unit 03

Basic Linux Exploits

List of Contents

- Agenda
- Why Software Vulnerabilities Matter?
- Attacks Exploiting Software Vulnerabilities
- Common Software Vulnerabilities
- Source of Input that need validation
- Input validation issues in Web applications
- What is Buffer Overflow Attack
- Types of Buffer Overflow Attack
- Conclusion

Agenda of session

- Presentation will defines the fundamentals of vulnerabilities of different programming platforms like C,C++,Python, Linux Shell code etc.

- Students will be able to list out different vulnerabilities along with the most common exploit of Buffer Overflow & Overwrite issues.

Why Software Vulnerabilities Matter?

- When a process reads input from attacker, the process may be exploited if it contains vulnerabilities.
- When an attacker successfully exploits a vulnerability, he can
  - Crash programs: Compromises availability
  - Execute arbitrary code: Compromises integrity
  - Obtain sensitive information: Compromises confidentiality

- Software vulnerability enables the attacker to run with privileges of other users, violating desired access control policy

Attacks Exploiting Software Vulnerabilities

- Drive-by download (drive-by installation)
  - malicious web contents exploit vulnerabilities in browsers (or plugins) to download/install malware on victim system.
- Email attachments in PDF, Word, etc.
- Network-facing daemon programs (such as http, ftp, mail servers, etc.) as entry points.
- Privilege escalation
  - Attacker on a system exploits vulnerability in a root process and gains root privilege

Common Software Vulnerabilities

- Input validation
- Race conditions
  - Time-of-check-to-time-of-use (TOCTTOU)
- Buffer overflows
- Format string problems
- Integer overflows

Sources of Input that Need Validation

- What are sources of input for local applications?
  - Command line arguments
  - Environment variables
  - Configuration files, other files
  - Inter-Process Communication call arguments
  - Network packets

- What are sources of input for web applications?
  - Web form input
  - Scripting languages with string input

Command line as a Source of Input: A Simple example

```
void main(int argc, char ** argv) {
   char buf[1024];
   sprintf(buf,"cat %s",argv[1]);
 system ("buf");
}
```

What can go wrong?

- Can easily add things to the command by adding ;, using e.g., "a; ls"
- User can set command line arguments to almost anything, e.g., by using execve system call to start a program, the invoker has complete control over all command line arguments.

Environment variables

- Users can set the environment variables to anything
  - Using execve
  - Has some interesting consequences

- Examples:
  - PATH
  - LD_LIBRARY_PATH
  - IFS

Attack by Resetting PATH

- A setuid program has a system call: system(ls);
- The user sets his PATH to be . (current directory) and places a program ls in this directory
- The user can then execute arbitrary code as the setuid program

- Solution: Reset the PATH variable to be a standard form (i.e., “/bin:/usr/bin”)

Attack by Resetting IFS

- However, you must also reset the IFS variable
  - IFS is the characters that the system considers as white space
- If not, the user may add “s” to the IFS
  - system(ls) becomes system(l)
  - Place a function l in the directory

Attack by Resetting LD_LIBRARY_PATH

- Assume you have a setuid program that loads dynamic libraries.
- UNIX searches the environment variable LD_LIBRARY_PATH for libraries.
- A user can set LD_LIBRARY_PATH to /tmp/attack and places his own copy of the libraries here.
- Most modern C runtime libraries have fixed this by not using the LD_LIBRARY_PATH variable when the EUID is not the same as the RUID or the EGID is not the same as the RGID

Input Validation Issues in Web Applications

- SQL injection
- SQL injection is a code injection technique that might destroy your database.
- SQL injection is one of the most common web hacking techniques.
- SQL injection is the placement of malicious code in SQL statements, via web page input.
- Cross Site Scripting
- Cross-site scripting (also known as XSS) is **a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application**. It allows an attacker to circumvent the same origin policy, which is designed to segregate different websites from each other.

A Remote Example: PHP passthru

- Idea
  - PHP passthru(*string*) executes command
  - Web-pages can construct *string* from user input and execute the commands to generate web content
  - Attackers can put “;” in input to run desired commands
- Example

```
echo 'Your usage log:<br />';
  $username = $_GET['username'];
  passthru(“cat /logs/usage/$username”);
```

- What if: “username=andrew;cat%20/etc/passwd”?

Directory Traversal Vulnerabilities in Web Applications

- A typical example of vulnerable application in php code is:

```
<?php
    $template = 'red.php';
    if ( isset( $_COOKIE['TEMPLATE'] ) )
    $template = $_COOKIE['TEMPLATE'];                    include
    ( "/home/users/phpguru/templates/" . $template );
?>
```

- Attacker sends

```
GET /vulnerable.php HTTP/1.0
Cookie: TEMPLATE=../../../../../../../../../../etc/passwd
```

Checking input can be tricky: Unicode vulnerabilities

- Some web servers check string input
  - Disallow sequences such as ../ or \
  - But may not check unicode %c0%af for '/'
- IIS Example, used by Nimda worm

```
http://victim.com/scripts/../../winnt/system32/cmd.exe?<some command>
```

  - passes <some command> to cmd command
  - scripts directory of IIS has execute permissions
- Input checking would prevent that, but not this

```
http://victim.com/scripts/..%c0%af..%c0%afwinnt/system32/...
```

  - IIS first checks input, then expands unicode

Time-of-check-to-time-of-use

- **TOCTTOU**, pronounced "*TOCK too*"
- A class of software bug caused by changes in a system between the checking of a condition (such as authorization) and use of the results of the check.
- Time-of-check-to-time-of-use (TOCTTOU - pronounced TOCK-too) is a file-based race condition that occurs when a resource is checked for a particular value, such as whether a file exists or not, and that value then changes before the resource is used, invalidating the results of the check.
  - When a process P requests to access resource X, the system checks whether P has right to access X; the usage of X happens later
  - When the usage occurs, perhaps P should not have access to X anymore.
  - The change may be because P changes or X changes.

An Example TOCTTOU

- In Unix, the following C code, when used in a setuid program, is a TOCTTOU bug:

```
if (access("file", W_OK) != 0)
{ exit(1); }

fd = open("file", O_WRONLY);
write(fd, buffer, sizeof(buffer));
```

Attacker tries to execute the following line in another process when this process reaches exactly this time:
Symlink("/etc/passwd", "file")

- Here, *access* is intended to check whether the real user who executed the setuid program would normally be allowed to write the file (i.e., *access* checks the real userid rather than effective userid).

TOCTTOU

- Exploiting a TOCTTOU vulnerabilities requires precise timing of the victim process.
- Most general attack may require “single-stepping” the victim, i.e., can schedule the attacker process after each operation in the victim
  - Techniques exist to “single-step” victim
- Preventing TOCTTOU attacks is difficult

What is Buffer Overflow?

- A **buffer overflow**, or **buffer overrun**, is an anomalous condition where a process attempts to store data beyond the boundaries of a fixed-length buffer.
- The result is that the extra data overwrites adjacent memory locations. The overwritten data may include other buffers, variables and program flow data, and may result in erratic program behavior, a memory access exception, program termination (a crash), incorrect results or — especially if deliberately caused by a malicious user — a possible breach of system security.
- Most common with C/C++ programs

History

- Used in 1988's Morris Internet Worm
- Alphe One's "Smashing The Stack For Fun And Profit" in Phrack Issue 49 in 1996 popularizes stack buffer overflows
- Still extremely common today

Types of Buffer Overflow Attacks

- Stack overflow
  - Shell code
  - Return-to-libc
    - Overflow sets ret-addr to address of libc function
  - Off-by-one
  - Overflow function pointers & longjmp buffers

- Heap overflow

Linux process memory layout

Stack Frame

Parameters
Return address
Stack Frame Pointer
Local variables
SP
Stack
Growth

What are buffer overflows?

- Suppose a web server contains a function:

```
void func(char *str) {
  char buf[128];

strcpy(buf, str);
  do-something(buf);
}
```

- When the function is invoked the stack looks like:

- What if `*str` is 136 bytes long? After `strcpy`:

Basic stack exploit

- Main problem: no range checking in strcpy().
- Suppose *str is such that after strcpy stack looks like:

Program P: `exec( "/bin/sh" )`

(exact shell code by Aleph One)

- When func() exits, the user will be given a shell !!
- Note: attack code runs *in stack*.

Carrying out this attack requires

- Determine the location of injected code position on stack when func() is called.
  - Location of injected code is fixed relative to the location of the stack frame
- Program P should not contain the '\0' character.
  - Easy to achieve
- Overflow should not crash program before func() exits.

Some unsafe C lib functions

strcpy (char *dest, const char *src)

strcat (char *dest, const char *src)

gets (char *s)

scanf (const char *format, …)

sprintf (conts char *format, …)

⋮

Other control hijacking opportunities

- In addition to overwrite return address on the stack, can also use overflow to overwrite the following:

- Function pointers: (used in attack on PHP 4.0.2)

  – Overflowing buf will override function pointer.

- Longjmp buffers: longjmp(pos) (used in attack on Perl 5.003)
  – Overflowing buf next to pos overrides value of pos.

Return-oriented programming

- Goal: executing arbitrary code without injecting any code.
- Observations:
  - Almost all instructions already exist in the process's address space, but need to piece them together to do what the attacker wants
- Attack:
  - Find instructions that are just before "return"
  - Set up the stack to include a sequence of addresses so that executing one instruction is followed by returning to the next one in the sequence.
- Effectiveness: has been shown that arbitrary program can be created this way

Heap Overflow

- Heap overflow is a general term that refers to overflow in data sections other than the stack
  - buffers that are dynamically allocated, e.g., by malloc
  - statically initialized variables (data section)
  - uninitialized buffers (bss section)

- Heap overflow may overwrite other data allocated on heap
- By exploiting the behavior of memory management routines, may overwrite an arbitrary memory location with a small amount of data.

Finding buffer overflows

- Hackers find buffer overflows as follows:
  - Run web server on local machine.
  - **Fuzzing:** Issue requests with long tags.
    All long tags end with "$$$$$".
  - If web server crashes,
    search core dump for "$$$$$" to find overflow location.
- Some automated tools exist.
- Then use disassemblers and debuggers (e..g IDA-Pro) to construct exploit.
- How to defend against buffer overflow attacks?

Preventing Buffer Overflow Attacks

- Use type safe languages (Java, ML).
- Use safe library functions
- Static source code analysis.
- Non-executable stack
- Run time checking: StackGuard, Libsafe, SafeC, (Purify), and so on.
- Address space layout randomization.

- Detection deviation of program behavior
- Access control to control aftermath of attacks…

Static source code analysis

- Statically check source code to detect buffer overflows.
  - Several consulting companies.
- Main idea: automate the code review process.
- Several tools exist:
  - Example: Coverity (Engler et al.): Test trust inconsistency.
- Find lots of bugs, but not all.

Bugs to Detect in Source Code Analysis

- Some examples

- **Crash Causing Defects**
- **Null pointer dereference**
- **Use after free**
- **Double free**
- **Array indexing errors**
- **Mismatched array new/delete**
- **Potential stack overrun**
- **Potential heap overrun**
- **Return pointers to local variables**
- **Logically inconsistent code**
- **Uninitialized variables**
- **Invalid use of negative values**
- **Passing large parameters by value**
- **Underallocations of dynamic data**
- **Memory leaks**
- **File handle leaks**
- **Network resource leaks**
- **Unused values**
- **Unhandled return codes**
- **Use of invalid iterators**

Format string problem

```
int  func(char *user)  {
    fprintf( stdout, user);
}
```

Problem: what if user = “%s%s%s%s%s%s%s” ??

- Most likely program will crash: DoS.
- If not, program will print memory contents. Privacy?
- Full exploit using user = “%n”

Correct form:

```
int  func(char *user)  {
    fprintf( stdout, “%s”, user);
}
```

Format string attacks (“%n”)

- printf(“%n”, &x) will change the value of the variable x
  - in other words, the parameter value on the stack is interpreted as a pointer to an integer value, and the place pointed by the pointer is overwritten

Vulnerable functions

Any function using a format string.

Printing:

printf, fprintf, sprintf, …

vprintf, vfprintf, vsprintf, …

Logging:

syslog, err, warn

Conclusion

- Presentation describes various types of vulnerabilities of memory management and operating system which can be exploit by any attacker to take the advantages by misusing it .

- Session describes that How we can protect our system from Buffer overflow attacks and its various types.

Integer Overflow

- Integer overflow: an arithmetic operation attempts to create a numeric value that is larger than can be represented within the available storage space.
- Example:

Test 1:

```
short x = 30000;
short y = 30000;
printf("%d\n", x+y);
```

Test 2:

```
short x = 30000;
short y = 30000;
short z = x + y;
printf("%d\n", z);
```

Will two programs output the same?
What will they output?

Off by one buffer overflow

- Sample code

```
func  f(char *input) {
  char buf[LEN];
  if (strlen(input) <= LEN) {
     strcpy(buf, input)
  }
}
```

What could go wrong here?

CSD4002:Ethical Hacking-Unit 4:Vulnerability Analysis

Dr. Ajit Kumar

November 2, 2019

1. Passive Analysis
2. Advanced Reverse Engineering
3. Intelligent Fuzzing with Sulley

What is reverse engineering?

At the highest level, it is simply taking a product apart to understand how it works.

- Understand the capabilities of the product's manufacturer
- Understand the functions of the product in order to create compatible components
- Determine whether vulnerabilities exist in a product
- Determine whether an application contains any undocumented functionality

Static / Passive Reverse Engineering

Static (also called passive) reverse engineering techniques in which you attempt to discover potential flaws and vulnerabilities simply by examining source or compiled code.

Ethical Reverse Engineering

Reverse engineering is often viewed as the craft of the cracker who uses her skills to remove copy protection from software or media. You are allowed to conduct such research as long as you have the permission of the owner of the subject system and you are acting in good faith to discover and secure potential vulnerabilities.

Vulnerability Analysis v/s Reverse Engineering

- One Should be interested in reverse engineering if want to extend their vulnerability assessment skills beyond the use of the pentester's standard bag of tricks.
- Nessus
- Pentesting tools can only report on **what they know**. They can't report on **undiscovered vulnerabilities**,
- Reverse engineering helps to find new and Unknown vulnerability
- Vulnerability researchers use a variety of reverse engineering techniques to find new vulnerabilities in existing software.

Reverse Engineering Considerations

- Failure to check for error conditions
- Poor understanding of function behaviors
- Poorly designed protocols
- Improper testing for boundary conditions

Uninitialized and Null pointers

Uninitialized pointers

Uninitialized pointers contain unknown data.

Null Pointer

Null pointers have been initialized to point to nothing so that they are in a known state.

Dereferencing

In C / C++ programs, attempting to access data (dereferencing) through either usually causes a program to crash or, at minimum, causes unpredictable behavior.

Group Activity

Using various Vulnerability database, find 10 different type of Vulnerabilities reported in 2019. Find few recent vulnerability reported regarding social media applications and Android OS.

Source Code Analysis

- A number of tools exist that attempt to automatically scan source code for known poor programming practices.
- automated tools tend to catch common cases and provide no guarantee that an application is secure.
- Source Code Auditing Tools (Lynis, ITS4, RATS (Rough Auditing Tool for Security), Flawfinder, and Splint (Secure Programming Lint), Microsoft's PREfast tool)
- RATS can Scan: Perl,PHP, Python, and C code.

Lynis and RATS

Example

- $lynis audit system
- $./rats -i -w 1 -d rats-c.xml example.c
- **rats-c.xml** :vulnerability database

Source Code Auditing:White, Black and Grey

- The goal of a white hat reviewing the output of a source code auditing tool should be to make the software more secure.
- The black hat is by definition interested in finding out how to exploit a program.
- It is useful for the gray hat to understand how to make use of the audit results to locate actual vulnerabilities and develop proof-of-concept code to demonstrate the seriousness of these vulnerabilities.

RATS:Installation and Use

Lynis: Installation and Use

- Kali: pre-installed
- Ubuntu: sudo apt-get install lynis
- Lynis performs more than 200 tests

Lynis commands

- $lynis show commands
- $lynis show settings
- $lynis update info
- $sudo lynis audit system
- $sudo lynis show details test-id
- $sudo nano /etc/lynis/custom.prf
- https://www.digitalocean.com/community/tutorials/how-to-perform-security-audits-with-lynis-on-ubuntu-1

Yasca: Installation and Use

- C/C++
- Java source and class files
- JSP source files
- PHP source files
- Perl
- Python

Manual Source Code Auditing

Sources of User-Supplied Data

- Command-line parameters argv manipulation
- Environment variables getenv()
- Input data files read(), fscanf(), getc(), fgetc(), fgets(), vfscanf()
- Keyboard input/stdin read(), scanf(), getchar(), gets()
- Network data read(), recv(), recvfrom()

Manual Auditing of Binary Code

Disassembler The purpose of a disassembler is to generate assembly language from a compiled binary.

Decompiler the purpose of a decompiler is to attempt to generate source code from a compiled binary.

Everything is file

Duplicating I/O Interfaces

In C/C++ programs, file descriptors 0, 1, and 2 correspond to the standard input (stdin), standard output (stdout), and standard error (stderr) devices. The dup2() function can be used to make stdin become a copy of any other file descriptor, including network sockets. Once this has been done, a program no longer accepts keyboard input; instead, input is taken directly from the network socket.

Manual Auditing of Binary Code

- Disassembler
- Decompiler

Decompilation

Lossy operation Information is lost in the process of generating machine language.

One-to-many There are many valid translations of a single line of source code to equivalent machine language statements.

Optimization to optimize a program for **speed** will generate vastly different code from what it will generate if asked to optimize that same program for **size**

Platform C and C++ are compiled to platform-specific machine language and linked to operating system–specific libraries.

Decompilation Tools

- JReversePro and Jad (Java Decompiler).
- `http://www.javadecompilers.com/`
- IDA Pro
- Hex-rays

Disassemblers

- compiled program to execute, it must communicate some information to its host operating system
- PE and ELF

IDA Pro

Developer Ilfak Guilfanov

Use Decompiler

Structure IDA Pro is actually a database application.

Working When a binary is loaded for analysis, IDA Pro loads each byte of the binary into a database and associates various flags with each byte. These flags can indicate whether a byte represents code, data, or more specific information such as the first byte of a multibyte instruction.

.IDB Disassemblies are saved as IDB files separate from the original binary

Dynamic Linking to analyze dynamically linked binaries, IDA Pro makes use of embedded symbol table information to recognize references to external functions. Within IDA Pro's disassembly listing, the use of standard library names helps make the listing far more readable.

Static Linking For statically linked C/C++ binaries, IDA Pro uses a technique termed Fast Library Identification and Recognition Technology (FLIRT), which attempts to recognize whether a given machine language function is known to be a standard library function, by matching disassembled code against signatures of standard library functions used by common compilers

Powerful Features

- Code graphing capabilities to chart function relationships
- Flowcharting capabilities to chart function flow
- A strings window to display sequences of ASCII or Unicode characters contained in the binary file
- A large database of common data structure layouts and function prototypes
- A powerful plug-in architecture that allows extensions to IDA Pro's capabilities to be easily incorporated
- A scripting engine for automating many analysis tasks
- Several integrated debuggers

IDA Pro Freeware

Installation and Basic Interface

IDA pro has multiple version, freeware, paid and evaluation for Windows, Linux and Mac

- Download installer from Hex-rays official site as per your OS.
- For Linux: make binary executable by $chmod +x idapro_XX.run
- Then run the installer
- Move to the folder example, /home/ajit/idaprofreeware
- $ ./ida64

Static Analysis Challenge

- Binaries that have been stripped of some or all of their symbol information
- Binaries that have been linked with static libraries
- Binaries that make use of complex, user-defined data structures
- Compiled C++ programs that make use of polymorphism
- Binaries that have been obfuscated in some manner to hinder analysis
- Binaries that use instruction sets with which IDA Pro is not familiar
- Binaries that use file formats with which IDA Pro is not familiar

Fuzzing

Blackbox Black box testing works because you can apply some **external stimulus** to a program and observe how the program **reacts** to that stimulus.

Monitoring Monitoring tools give you the capability to observe the program's reactions.

Fuzzing Fuzzing tools are designed for the **rapid generation** of input cases designed to induce errors in a program.

Challenge The real challenge of fuzzer development is building them in such a way that they generate interesting input in an intelligent, efficient manner.

Known and Unknown Protocols

- Known Protocol (HTTP, FTP etc. RFC)
- Unknown Protocol (proprietary protocol non-RFC)

SPIKE

- SPIKE is a fuzzer creation toolkit/API
- By: Dave Aitel of Immunity, Inc
- SPIKE provides a library of C functions for use by fuzzer developers
- SPIKE is designed to assist in the creation of network-oriented fuzzers and supports sending data via TCP or UDP.
- SPIKE provides several example fuzzers for protocols ranging from HTTP to Microsoft Remote Procedure Call (MSRPC)

Working of SPIKE

- The SPIKE API centers on the notion of a "spike" data structure.
- Various API calls are used to push data into a spike and ultimately send the spike to the application being fuzzed.
- Spikes can contain static data, dynamic fuzzing variables, dynamic length values, and grouping structures called blocks.
- A SPIKE "block" is used to mark the beginning and end of data whose length should be computed. Blocks and their associated length fields are created with name tags.

Spike Creation Primitives

- struct spike *new_spike() Allocate a new spike data structure.
- int spike_free(struct spike *old_spike) Release the indicated spike.
- int set_spike(struct spike *new_spike) Make newspike the current spike.

All future calls to data manipulation functions will apply to this spike.

SPIKE Static Content Primitives

- s_string(char *instring) Insert a static string into a spike.
- s_binary(char *instring) Parse the provided string as hexadecimal digits and add the corresponding bytes into the spike.
- s_bigword(unsigned int aword) Insert a big-endian word into the spike. Inserts 4 bytes of binary data into the spike.
- s_xdr_string(unsigned char *astring) Insert the 4-byte length of astring followed by the characters of astring into the spike. This function generates the XDR representation of astring

XDR is the External Data Representation standard, which describes a standard way in which to encode various types of data such as integers, floating-point numbers, and strings.

SPIKE Block Handling Primitives

- int_block_start(char *blockname) Start a named block. No new content is added to the spike. All content added subsequently up to the matching block_end call is considered part of the named block and contributes to the block's length.
- int s_block_end(char *blockname) End the named block. No new content is added to the spike. This marks the end of the named block for length computation purposes.

SPIKE Fuzzing Variable Declaration

A fuzzing variable is a string that SPIKE will manipulate in some way between successive transmissions of a spike.

- void s_string_variable(unsigned char *variable) Insert an ASCII string that SPIKE will change each time a new spike is sent.

When a spike contains more than one fuzzing variable, an iteration process is usually used to modify each variable in succession until every possible combination of the variables has been generated and sent.

SPIKE Script Parsing

SPIKE statements can be placed in a text file and executed from within another SPIKE-based program. All of the work for executing scripts is accomplished by a single function.

- int s_parse(char *filename) Parse and execute the named file as a SPIKE script.

Example

```
POST /cgi-bin/login.pl HTTP/1.1
Host: gimme.money.com
Connection: close
User-Agent: Mozilla/6.0
Content-Length: 29
Content-Type: application/x-www-form-encoded
user=smith&password=smithpass
```

A Simple SPIKE Script:.spk

```
s_string("POST /cgi-bin/login.pl HTTP/1.1\r\n");
s_string("Host: gimme.money.com\r\n);
s_string("Connection: close\r\n");
s_string("User-Agent: Mozilla/6.0\r\n");
s_string("Content-Length: ");
s_blocksize_string("post_args", 7);
s_string("\r\nContent-Type: application/x-www-form-encoded\r\n\r\n");
s_block_start("post_args");
s_string("user=");
s_string_variable("smith");
s_string("&password=");
s_string_variable("smithpass");
s_block_end("post_args");
```

generic_send_tcp

It takes care of the details of initializing a spike,
parsing a script into the spike,
and iterating through all fuzzing variables in the spike.

1. **Host** the host to be fuzzed,
2. **Port** the port to be fuzzed,
3. **SPIKE Script** the filename of the spike script
4. **Do Skip Variable** information on whether any fuzzing variables should be skipped,
5. **State Skip Variable** whether any states of each fuzzing variable should be skipped.

```
$./generic_send_tcp gimme.money.com 80 demo.spk 0 0
```

Other Fuzzer

- SPIKE Proxy: Web Application, Python
- Sharefuzz:fuzz set user ID (SUID) root binaries.
- **mangleme**:specifically target browser-based client-side vulnerabilities.(outdated)
- HTMLer: https://securiteam.com/tools/6Z00N1PBFK/ (Python Port of mangleme)
- jsfunfuzz: Javascript (Mozilla)
- css-grammar-fuzzer (CSS fuzzer)

Intelligent fuzzing

Instead of blindly throwing everything, techniques have been developed to analyze how a server works and to customize a fuzzer to get past the filters and reach deeper inside the server to discover even more vulnerabilities.

Intelligent fuzzing Steps

- need to conduct a protocol analysis of the target.
- need a way to fuzz that protocol and get feedback from the target as to how you are doing.

Sulley and Boo From Monsters Inc.

Sulley/Boofuzz

- The Sulley fuzzing framework automates this process and allows you to intelligently sling packets across the network.
- This tool is truly revolutionary in that it provides not only a great fuzzer and debugger, but also the infrastructure to manage a fuzzing session and conduct postmortem analysis.
- https://github.com/OpenRCE/sulley
- https://github.com/jtpereyda/boofuzz

Protocol Analysis

www.ietf.org/rfc/rfc1179.txt

Exploiting Windows Access Control Model for Local Elevation of Privilege

This TOPIC will teach you about Windows Access Control and how to find instances of misconfigured access control exploitable for local privilege escalation.

Why Access Control Is Interesting to a Hacker

Access control is about the science of protecting things. Finding vulnerabilities in poorly implemented access control is fun because it feels like what security is all about. It isn't blindly sending huge, long strings into small buffers or performing millions of iterations of brute-force fuzzing to stumble across a crazy edge case not handled properly; neither is it tricking Internet Explorer into loading an object not built to be loaded in a browser. Exploiting access control vulnerabilities is more about elegantly probing, investigating, and then exploiting the single bit in the entire system that was coded incorrectly and then compromising the whole system because of that one tiny mistake.

How Windows Access Control Works

It's important to first understand how Windows Access Control works. This introductory section is large because access control is such a rich topic. But if you stick with it and fully understand each part of this, it will pay off with a deep understanding of this greatly misunderstood topic, allowing you to find more and more elaborate vulnerabilities.

This section will be a walkthrough of the four key foundational components you'll need to understand to attack Windows Access Control: the security identifier (SID), the access token, the security descriptor (SD), and the access check

Tools for Analyzing Access Control Configurations

With the concept introduction out of the way, we're getting closer to the fun stuff. Before we can get to the attacks, however, we must build up an arsenal of tools capable of dumping access tokens and security descriptors. As usual, there's more than one way to do each task. All the enumeration we've shown in the figures so far was done with free tools downloadable from the Internet. We'll demonstrate each tool we used earlier, show you where to get them, and show you how to use them.

Dumping the Process Token

The two easiest ways to dump the access token of a process or thread are Process Explorer and the! token debugger command. Process Explorer ...

Special SIDs, Special Access, and “Access Denied”

You also are armed with tools to enumerate the access control objects that factor into AccessCheck. It’s time now to start talking about the “gotchas” of access control and then start into the attack patterns.

Special SIDs

You’ve seen the user SID several times. You’ve seen the SID of the Administrators and Users groups and how the presence of those SIDs in the token changes the privileges present and the access granted. You’ve seen the Local System SID. Let’s discuss several other SIDs that might trip ...

Analyzing Access Control for Elevation of Privilege

All the file read access discussion earlier was to help you understand concepts. The attack methodology and attack process are basically the same no matter the resource type.

- Step 1: Enumerate the object’s DACL and look for access granted to non-admin SIDs.
  We look for non-admin SIDs because attacks that require privileged access to pull off are not worth enumerating. Group those non-admin SIDs in the DACL into untrusted and semi-trusted users. Untrusted users are Users, Guest, Everyone, Anonymous, INTERACTIVE, and so on. Semi-trusted users are interesting in the case of a multistage attack. Semi-trusted ...

Attack Patterns for Each Interesting Object Type

Let’s apply the analysis methodology to real objects and start finding real security vulnerabilities. The following sections will list DACL enumeration techniques, then the power permissions, and then will demonstrate an attack.

Attacking Services

Services are the simplest object type to demonstrate privilege escalation, so we’ll start here. Let’s step through our attack process.

Enumerating DACL[**Discretionary Access Control List**] of a Windows Service

We’ll start with the first running service on a typical Windows XP SP2 system.

```
C:\tools>net start

These Windows services are started:
```

```
Alerter

Application Layer Gateway Service

Ati HotKey Poller

Automatic Updates

...
```

We used AccessChk.exe earlier to enumerate file system DACLs.

What Other Object Types Are out There?

Services, registry keys, files, and directories are the big four object types that will expose code execution vulnerabilities. However, several more object types might be poorly ACL'd. Nothing is going to be as easy and shellcode-free as the objects listed. The remaining object types will expose code execution vulnerabilities but you'll probably need to write "real" exploits to leverage those vulnerabilities. Having said that, let's briefly talk through how to enumerate each one.

From Vulnerability to Exploit

Whether you use static analysis, dynamic analysis, or some combination of both to discover a problem with a piece of software, locating a potential problem or causing a program to melt down in the face of a fuzzer onslaught is just the first step. With static analysis in particular, you face the task of determining exactly how to reach the vulnerable code while the program is executing. Additional analysis followed by testing against a running program is the only way to confirm that your static analysis is correct. Should you provoke a crash using a fuzzer, you are still faced with the task of dissecting the fuzzer input that caused the crash and understanding any crash dumps yielded by the program you are analyzing. The fuzzer data needs to be dissected into the portions required strictly for code path traversal, and the portions that actually generate an error condition with the program. Knowing that you can crash a program is a far cry from understanding exactly why the program crashes. If you hope to provide any useful information to assist in patching the software, it is important to gain as detailed an understanding as possible about the nature of the problem. It would be nice to avoid this conversation:

Researcher: "Hey, your software crashes when I do this…"

Vendor: "Then don't do that!

In favor of this one: Researcher: "Hey, you fail to validate the widget field in your octafloogaron application, which results in a buffer overflow in function umptiphratz. We've got packet captures, crash dumps, and proof of concept exploit code to help you understand the exact nature of the problem."

Vendor: "All right, thanks, we will take care of that ASAP."

Whether a vendor actually responds in such a positive manner is another matter. In fact, if there is one truth in the vulnerability research business, it's that dealing with vendors can be one of the least rewarding phases of the entire process. The point is that you have made it significantly easier for the vendor to reproduce and locate the problem and increased the likelihood that it will get fixed.

Exploitability

Crashability and exploitability are vastly different things. The ability to crash an application is, at a minimum, a form of denial of service. Unfortunately, depending on the robustness of the application, the only person whose service you may be denying could be you. For true exploitability, you are really interested in injecting and executing your own code within the vulnerable process. In the next few sections, we discuss some of the things to look for to help you determine whether a crash can be turned into an exploit.

Debugging for Exploitation

Developing and testing a successful exploit can take time and patience. A good debugger can be your best friend when trying to interpret the results of a program crash. More specifically, a debugger will give you the clearest picture of how your inputs have conspired to crash an application. Whether an attached debugger captures the state of a program when an exception occurs or you have a core dump file that can be examined, a debugger will give you the most comprehensive view of the state of the application when the problem occurred. For this reason, it is extremely important to understand what a debugger is capable of telling you and how to interpret that information.

Initial Analysis

Why did the program crash? Where did the program crash? These are the first two questions that need to be answered. The "why" you seek here is not the root cause of the crash, such as the fact that there is a buffer overflow problem in function xyz.

Instead, initially you need to know whether the program segfaulted or perhaps executed an illegal instruction. A good debugger will provide this information the moment the program crashes. A segfault might be reported by gdb as follows:

Program received signal SIGSEGV, Segmentation fault.

0x08048327 in main ()

Always make note of whether the address resembles user input in any way. It is common to use large strings of A's when attacking a program. One of the benefits to this is that the address 0x41414141 is easily recognized as originating from your input rather than correct program operation. Using the addresses reported in any error messages as clues, you next examine the CPU registers to correlate the problem to specific program activity.

General Register Analysis

If you haven't managed to take control of eip, the next step is to determine what damage you can do using other available registers. Disassembly of the program in the vicinity of eip should reveal the operation that caused the program crash. The ideal condition that you can take advantage of is a write operation to a location of your choosing. If the program has crashed while attempting to write to memory, you need to determine exactly how the destination address is being calculated. Each general-purpose register should be studied to see if it (a) contributes to the destination address computation and (b) contains user-supplied data. If both of these conditions hold, it should be possible to write to any memory location. The second thing to learn is exactly what is being written and whether you can control that value; if you can, you have the capability to write any value anywhere. Some creativity is required to utilize this seemingly minor capability to take control of the vulnerable program. The goal is to write your carefully chosen value to an address

that will ultimately result in control being passed to your shellcode. Common overwrite locations include saved return addresses, jump table pointers, import table pointers, and function pointers. Format string vulnerabilities and heap overflows both work in this manner because the attackers gain the ability to write a data value of their choosing (usually 4 bytes, but sometimes as little as 1 or as many as 8) to a location or locations of their choosing.

Improving Exploit Reliability

Another reason to spend some time understanding register content is to determine whether any registers point directly at your shellcode at the time you take control of eip. Since the big question to be answered when constructing an exploit is "What is the address of my shellcode?", finding that address in a register can be a big help. As discussed in previous chapters, injecting the exact address of your shellcode into eip can lead to unreliable results since your shellcode may move around in memory. When the address of your shellcode appears in a CPU register, you gain the opportunity to do an indirect jump to your shellcode. Using a stack-based buffer overflow as an example, you know that a buffer has been overwritten to control a saved return address. Once the return address has been popped off the stack, the stack pointer continues to point to memory that was involved in the overflow and that could easily contain your shellcode.

The classic technique for return address specification is to overwrite the saved eip with an address that will point to your shellcode so that the return statement jumps directly into your code. While the return addresses can be difficult to predict, you do know that esp points to memory that contains your malicious input, because following the return from the vulnerable function, it points 4 bytes beyond the overwritten return address. A better technique for gaining reliable control would be to execute a jmp esp or call esp instruction at this point. Reaching your shellcode becomes a two-step process in this case. The first step is to overwrite the saved return address with the address of a jmp esp or call esp instruction. When the

exploitable function returns, control transfers to the jmp esp, which immediately transfers control back to your shellcode. This sequence of events is detailed in Figure 26-3.

A jump to esp is an obvious choice for this type of operation, but any register that happens to point to your user-supplied input buffer (the one containing your shellcode) can be used. Whether the exploit is a stack-based overflow, a heap overflow, or a format string exploit, if you can find a register that is left pointing to your buffer, you can attempt to vector a jump through that register to your code. For example, if you recognize that the esi register points to your buffer when you take control of eip, then a jmp esi instruction would be a very helpful thing to find.

Fuzz Testing (Fuzzing) Tutorial (What is, Types, Tools, Example)

Fuzz Testing

Fuzz Testing, often known as fuzzing, is a software testing approach that involves injecting incorrect or random data (FUZZ) into a software system in order to find coding errors and security flaws. Fuzz testing involves introducing data using automated or semi-automatic approaches and evaluating the system for different exceptions such as system crashes or built-in code failure, among other things.

Barton Miller, a professor at the University of Wisconsin, invented fuzz testing in 1989. Fuzz testing, often known as fuzzing, is a type of software testing that falls within the security testing umbrella.

What is the purpose of fuzz testing?

- Fuzzy testing usually uncovers the most significant security flaw or vulnerability.
- When combined with Black Box Testing, Beta Testing, and other debugging techniques, fuzz testing produces more effective results.
- Fuzz testing is a technique for determining a software's vulnerability. It is one of the most cost-effective testing methods.
- One of the black box testing techniques is fuzz testing. One of the most frequent methods hackers employs to identify system vulnerabilities is fuzzing.

Steps of Fuzz Testing

The fundamental testing phases are included in the fuzzy testing process.

- Determine the system to be targeted.
- Determine the inputs
- Produce Fuzzed Data
- Run the test with ambiguous data.
- Keep an eye on the system's performance.
- Keep a defect log.

Example of Fuzzers

Fuzzers that modify existing data samples to produce new test data are known as mutation-based fuzzers. This is the most basic and easy technique; it starts with acceptable protocol samples and continues to mutilate every byte or file.

Generation-Based Fuzzers create new data based on the model's input. It starts from the beginning, producing input depending on the requirements.

The most successful fuzzer is PROTOCOL-BASED-FUZZER, which has extensive knowledge of the protocol format being tested. The comprehension is determined by the specification. It entails entering an array of the specification into the tool, then going through the specification and adding irregularities to the data contents, sequence, and so on using a model-based test generating approach. Syntax testing, grammar testing, robustness testing, and other terms are used to describe this process. Fuzzer may create test cases from scratch or from inputs that are valid or invalid.

Protocol-based fuzzing has two major drawbacks

- Testing will not be possible until the specification is complete.
- A lot of important protocols are extensions of already published protocols. Test coverage for new protocols will be restricted if fuzz testing is based on public specifications.
- Sending random input to the software, either as protocol packets or as an event, is the simplest type of fuzzing approach. This approach of using random input to identify vulnerabilities in a variety of apps and services is quite effective. Other methods are also accessible, and they are extremely simple to use. We just need to modify the existing inputs to apply these approaches. We can alter the input simply by swapping the bits.

Types of bugs detected by Fuzz Testing

Assertion failures and memory leaks are two types of vulnerabilities that Fuzz Testing may identify. This approach is commonly employed in big systems where flaws compromise memory safety, which is a serious flaw.

- **Invalid input** − Fuzzers are used in fuzz testing to produce incorrect input that is used to test error-handling algorithms, which is crucial for software that does not have control over its input. Simple fuzzing is a technique for automating negative testing.
- **Correctness bugs** − Some sorts of "correctness" issues can also be detected via fuzzing. For example, a corrupted database, inadequate search results, and so on.

Tools for Fuzz Testing

Burp Suite, Peach Fuzzer, and other online security tools may be utilized extensively in fuzz testing.

- **Peach Fuzzer** − Peach Fuzzer outperforms scanners in terms of coverage and security. Peach Fuzzer can discover both known and unknown threads, unlike other testing tools that can only identify known threads.
- **Spike Proxy** − It's a professional-grade tool that scans web applications for application-level flaws. SPIKE Proxy includes the fundamentals, such as SQL Injection and cross-site scripting, but it's built on a Python architecture that's entirely open. SPIKE Proxy is a Linux and Windows application.
- **Webscarab** − Because Webscarab is developed in Java, it can run on a variety of systems. The Webscarab framework is used to analyze the application, which communicates using HTTP and HTTPS protocols. Webscarab, for example, acts as an intercepting proxy, allowing the operator to monitor and alter browser requests before they are sent to the server. Also, before the browser receives the answer created by the server, it is possible to review and edit it. In this approach, if web scarab discovers a flaw, it will be included in the list of reported problems.
- **OWASP WSFuzzer** − WSFuzzer is a Python-based GPL-licensed application. Currently, software that has been GPL'd is aimed at Web Services. The major target of the current version of OWASPWSFuzzer is HTTP-based SOAP services.

Drawbacks of Fuzz Testing

Fuzz testing has a number of drawbacks −

- Only fuzz testing can provide you a comprehensive view of a security threat or flaws.
- Fuzz testing is less successful when dealing with security risks that don't cause software crashes, such as viruses, worms, Trojans, and other similar threats.
- Fuzz testing can only discover minor flaws or dangers.
- It will take a substantial amount of time to function well.
- Setting a boundary value condition with random inputs is difficult, but most tests are now able to handle this challenge by utilizing deterministic algorithms based on user inputs.

Difficulties in setting up Fuzzers

One of the factors limiting developers from using fuzz testing more widely is the time it takes to set it up.

On the one hand, these technologies are becoming more user-friendly all the time, according to DeMott. “On the other hand, setting up, monitoring, managing, and triaging the problems that come out still requires significant experience, security, and coding.”

The hardest aspect for clients, according to Knudsen, whose firm produces a generational fuzzer, is setting out the data model for the fuzzer to utilize as a template to construct test cases. Customers who utilize common network protocols and file formats may easily use existing templates rather than spending time creating their own with the commercial product, which contains a huge database of test suites.

“Assume you're creating an Internet of Things device, such as a smart thermostat,” Knudsen explained. “It'll have to connect with a server somewhere,” says the author. So you can reuse a lot of current solutions because there are already defined network protocols that people can use.”

However, most applications will still have bespoke code that has to be tested with new data models on top of these old protocols.

Developers may utilize Defensics' software development kit (SDK) to construct bespoke data models, although creating them, as with other generational fuzzing technologies, takes time.

In the past, people who used fuzzing had to be specialized in the field.

It's also critical to incorporate fuzz testing into the DevOps pipeline, as well as to ensure those test failures are properly documented. Developers will be unable to fix the code if tests fail without logging the input data that caused the problem.

ClusterFuzz is a Google tool for managing fuzzing infrastructure and tracking fuzzer performance.DeMott stated, "I initially came up with the name many years ago, but my adviser insisted on altering it."

GitLab has purchased Peach Tech and Fuzzit, two-generational fuzzing technologies that will be integrated into the company's continuous integration and continuous delivery (CI/CD) workflow.

DeSanto stated, "We thought there was a void," alluding to GitLab's previous security testing tool offerings, which included SAST and DAST. Today, our security team uses GitLab Secure as its primary testing tool to check our application's security and dependability.

The objective of GitLab is to make it easier for developers to include fuzz testing into their development processes.

Developers that want to use one of the fuzzers can do so by adding a line to the YAML script that manages the CI/CD process.

According to DeSanto, “people who utilize fuzzing previously had to be experts in fuzzing to use it.” “The benefit of utilizing fuzzing with GitLab is that we know what you're doing. You use us for source code management and continuous integration. Rather than telling [the fuzzer], "Here is the structure of my API," we can scan your source code and generate that specification for you automatically. There's no need for you to try to comprehend the instrument in order to use it.”

Fuzz testing may become a more common and effective component of the testing environment if additional fuzzing technologies are more easily incorporated into developers' processes.

Summary

Fuzz testing is a type of software engineering that identifies the presence of flaws in an application. Fuzzing does not ensure that all flaws in a program will be detected. However, employing the Fuzz approach guarantees that the application is both resilient and safe since it helps to reveal

the majority of frequent flaws.

What Is Client Side Exploitation?

Introduction:-

Nowadays Server side is getting strong day by day but there are still vulnerabilities present on the client-side and this leads to client-side exploitation. This article will introduce you to various client-side exploitation techniques that can be used in a penetration test. Client-side exploits are very useful to attackers when the client is behind the firewall or any application security layer. In this situation, it is not possible to directly access the client through the network.

The success rate of finding a vulnerability in the client site is directly proportional to the reconnaissance. If you are performing penetration testing on any application to test is there any client-side exploitation is possible or not you must have an understanding of possible attack scenarios to find and prevent the Client-Side Exploitation on your application.

Attack Scenario's In Client-Side Exploitation:

E-Mails with Malicious Attachments-

In this particular attack scenario, the attacker will send the malicious files such as PDF, exe, or mp3 in the hope that the victim would click on the link and download and execute the attachment. Upon execution, the attacker has a Meterpreter session opened on the victim's machine.

This attack can be a bit difficult to accomplish, as the attacker needs to convince the victim to execute their .exe file. Another major hurdle would be the victim's antivirus, which you need to bypass. Luckily, Metasploit has some built-in encoding mechanisms that, when used effectively, can evade some anti viruses, and if used effectively. However, all this is based on trial and error.

Malware Loaded on USB Sticks-

This method can be used by an attacker when he/she have physical access to the victim's machine. The attacker loads up a malicious PDF file or a malicious executable payload via a USB stick. Once the USB stick is inserted, malicious code will automatically be executed and the attacker would get a meterpreter session opened on the victim's machine. Teensy USB is a device that has the capability to emulate a mouse and keyboard.

E-Mails Leading to Malicious Links-

In this particular attack scenario, the attacker will send malicious links in the hope that victims would click on them. The link could be a fake log-in page or a web server hosted with an attacker's malicious code. Considering the attacker is hosting a web server, the code will be executed in the victim's browser and an attacker will have a meterpreter session opened.
In this scenario, the attacker sends the victim a malicious link, and when the victim clicks on it, we will be able to perform various attacks. Here are some examples:

- An attacker can set up a fake log-in page of any particular website, for example, facebook.com, and ask the victim to log in to the fake log-in page actually located at facebookfake.com

- If the attacker is on the same network as the victim, he can launch a DNS spoofing attack, where we can replace the IP of example.com with that of the attacker's fake log-in page, and as soon as the victim visits example.com, he would log in to our fake page instead.

- An attacker can also perform DNS spoofing, where instead of the fake log-in page we can redirect the victim to our malicious webserver that would use relevant browser exploits to compromise the victim's browser.

Browser Exploitation-

Browser-based exploits are one of the most important forms of client-side exploits. Imagine a scenario where you are pen-testing against an organization. If it's an internal pentest, you would already own a box on the LAN. If it's an external pentest you need to somehow gain access to a system. You can set up a malicious web server and ask the victim to visit the server. As soon as he clicks your link, he gets compromised.

Most of the employees of an organization frequently browse on social networking websites like Facebook and Linkedin. We, as penetration testers, can take advantage of this and send malicious links to the employees and compromise them. On an internal network, the attacker could simply use a DNS poisoning attack to redirect victims to his malicious webserver. To sum up, there is a whole lot of attack surface when it comes to browser exploitation.

Compromising Client-Side Update-

In this scenario, an attacker will utilize previously mentioned attack vectors to compromise the client-side updating process. It means that whenever a victim updates a particular software, he will download malicious code instead of updates. In this scenario, the attacker will compromise client-side updates by using a neat tool called Evilgrade, which comes preinstalled with BackTrack. Evilgrade takes advantage of insecure update processes as the user normally does not double-check before an update because they trust that the application is being downloaded from the right place.

Social Engineering Attack-

In a social engineering penetration test, the organization may ask you to attack its users. This is where you use speared phishing attacks and browser exploits to trick a user into doing things they did not intend to do. A social engineering attack can be part of a network penetration test in some cases.
In the current situation of online work and education, social engineering attack is increasing day by day. Almost every person who has an email address and mobile number may get a victim of social engineering attacks for the attacker. In social engineering attack, the attacker fraudulently obtains confidential data from the users.

How To Protect From Client-Side Exploitation:-

1. In client-side exploitation, attackers take advantage of the weakest link that is clients.

2. To protect yourself from client-side attacks you have to be alert during your everyday Internet surfing.

3. Don’t open any link coming from a malicious or unknown person.

4. After opening any email attachment always make sure that the mail is coming from an authorized source.

5. Avoid downloading .exe attachments of a mail.
6. Always check the confidentiality of the domain of the website after filling in confidential information like username, password, card number, bank account number, etc.

Reverse Engineering — Hacking Tools

The term reverse engineering, refers to the disassembling of an object, following a thorough examination of its composition/construction so to understand how it works to duplicate or upgrade the object.

Taken by older industries, the practice of reverse engineering is now widely used in the software engineering sector. In this digital-information age, reverse engineering has become a tool that can be used as a way to create compatible products that are cheaper than the existing ones or even free in some cases, uniquely modify the software, and exchange knowledge as a result into making better, more reliable and secure products.

Can be applied to various aspects of both software, and hardware development to understand how they behave under various conditions, to retrieve the source code that was lost, fix issues, to adapt existing software programs with new hardware, etc.

The use of reverse engineering is also greatly exercised to identify malicious content in the source code of a software, such as viruses, or to expose security flaws(backdoors, virus, misconfigurations) and address possible privacy issues.

Researchers can also use this technique to reverse engineer malware to understand how it works to nullify its properties, identify the potential owner, and use the knowledge gained to update their virus databases and prepare mitigation measures for future malware attacks.

Reverse engineering, the process of taking a software program's binary code and recreating it so as to trace it back to the original source code, is being widely used in computer hardware and software to enhance product features or fix certain bugs. For example, the programmer writes the code in a high-level language such as C, C++ etc. as computers do not speak these languages, the code written in these programming languages needs to be assembled in a format that is machine specific. In short, the code written in high level language needs to be interpreted into low level or machine language.

The process of converting the code written in high level language into a low level language without changing the original program is known as reverse engineering. It's similar to disassembling the parts of a vehicle to understand the basic functioning of the machine and internal parts etc. and thereafter making appropriate adjustments to give rise to a better performing or superior vehicle.

Where is Reverse Engineering Used?

- In software design, reverse engineering enables the developer or programmer to add new features to the existing software with or without knowing the source code. Different techniques are used to incorporate new features into the existing software.
- Reverse engineering is also very beneficial in software testing, as most of the virus programmers don't leave behind instructions on how they wrote the code, what they have set out to accomplish etc. Reverse engineering helps the testers to study the virus and other malware code. The field of software testing, while very extensive, is also interesting and requires vast experience to study and analyze virus code.
- The third category where reverse engineering is widely used is in software security. Reverse engineering techniques are used to make sure that the system does not have any major vulnerabilities and security flaws. The main purpose of reverse engineering is to make the system robust so as to protect it from spywares and hackers. Infact, this can be taken a step forward to Ethical hacking, whereby you try to hack your own system to identify vulnerabilities.

While one needs a vast amount of knowledge to become a successful reverse engineer, he or she can definitely have a lucrative career in this field by starting off with the basics. It is highly suggested that you first become familiar with assembly level language and gain significant amount of practical knowledge in the field of software designing and testing to become a successful software engineer.

Reverse Engineering Tools

As mentioned above, reverse engineering is the process of analyzing the software to determine its components and their relationships. The process of reverse engineering is accomplished by making use of some tools that are categorized into debuggers or disassemblers, hex editors, monitoring and decompile tools:

1. **Disassemblers** – A disassembler is used to convert binary code into assembly code and also used to extract strings, imported and exported functions, libraries etc. The disassemblers convert the machine language into a user-friendly format. There are different dissemblers that specialize in certain things.

2. **Debuggers** – This tool expands the functionality of a disassembler by supporting the CPU registers, the hex duping of the program, view of stack etc. Using debuggers, the programmers can set breakpoints and edit the assembly code at run time. Debuggers analyse the binary in a similar way as the disassemblers and allow the reverser to step through the code by running one line at a time to investigate the results.
3. **Hex Editors** – These editors allow the binary to be viewed in the editor and change it as per the requirements of the software. There are different types of hex editors available that are used for different functions.

4. **PE and Resource Viewer** – The binary code is designed to run on a windows based machine and has a very specific data which tells how to set up and initialize a program. All the programs that run on windows should have a portable executable that supports the DLLs the program needs to borrow from.

Reverse engineering has developed significantly and taken a positive approach to creating descriptive data set of the original object. Today, there are numerous legitimate applications of reverse engineering. Due to the development of numerous digitizing devices, reverse engineering software enables programmers to manipulate the data into a useful form. The kind of applications in which reverse engineering is used ranges from mechanical to digital, each with its own advantages and applications. Reverse engineering is also beneficial for business owners as they can incorporate advanced features into their software to meet the demands of the growing markets.

VIT UNIVERSITY SCHOOL OF COMPUTER SCIENCE ENGINEERING

UNIT 04

Vulnerability Analysis

Output of this Presentation

- Students will be able learn about the vulnerability and its types , this presentation also going to explain about the various model and tools of vulnerability analysis, this presentation brings learning for protecting our system from various attacks and vulnerable activities.

What is Vulnerability?

- Vulnerability can be defined as an issue in the software code that a hacker can exploit to harm the systems. It can be a gap in the implementation of cybersecurity procedures or a weakness in the controls.

What is an example of vulnerability?

- **Examples of vulnerabilities exist in every industry. These include:**
- Unauthorized network access by Hackers due to a weak Firewall
- Cracking of Wi-Fi Passwords
- Exposure of sensitive data due to lack of application security
  - Credit card data, Health Records
- Security misconfiguration
  - Misconfiguration of passwords
- Insecure cryptographic storage

Types of Vulnerability

The 4 main types of vulnerabilities are:

1. **Faulty defenses –** Poor defense measures pave the way for easy intrusion by hackers. This may be due to weak authentication, authorization, and encryption.
2. **Resource management not adequate –**The chances of buffer overflow and the potential to have many vulnerabilities are greater when there is inadequate resource management.

Types of Vulnerability

1. **Insecure connections –** If the connection between the system, application and networks is insecure, there is a higher probability of many threats like SQL injection.
2. **End user errors and misuse –** In many cases, the errors are caused by humans and misuse of the systems.

What are vulnerabilities?

1. **Allowing Domains or Accounts to Expire**
   - When domain names have expired, the hacker may buy them and set up a mail server. The hacker can find out the incoming mails and get to know the details.
2. **Buffer Overflow**
   - A process where there is more data added to the buffer and the excess data becomes corrupted and susceptible to vulnerabilities.
3. **Business logic vulnerability**
   - The software code may be missing a security control like authentications, encryption, or authorization.

What are vulnerabilities? Cont..

4. **CRLF Injection**

   Carriage Return Line Feed – Can be done by modifying the HTTP parameter of the URL.

5. **CSV Injection**

   When untrusted CSV files are embedded to the websites causing vulnerabilities.

6. **Catch Null Pointer Exception**

   When the program contains the null pointer, it is highly risky.

7. **Covert storage channel**

   This can help the attackers easily and often happens due to faulty implementation.

8. **Deserialization of untrusted data**

   Injection of malicious data into the applications to stop execution of programs.

What are vulnerabilities? Cont..

9. **Directory Restriction Error**

   Happens due to the improper use of CHROOT.

10. **Doubly freeing memory**

    This error occurs when free() is called more than once in the memory address.

11. **Empty String Password**

    Empty string password is highly insecure.

12. **Expression Language Injection**

    Injection happens when attacker-controlled data enters an EL interpreter.

13. **Full Trust CLR Verification issue Exploiting Passing Reference Types by Reference**

    Create a file called by ValueTypeTest.cs and compile it using csc by Value Type Test.csc.

What are vulnerabilities? Cont..

14. **Heartbleed Bug**

    Catastrophic bug in OpenSSL

15. **Improper Data Validation**

    Multiple validation forms with the same name indicate that validation logic is not up-to-date.

16. **Improper pointer subtraction**

    The subtraction of one pointer from another to determine the size is dependent on the assumption that both pointers exist in the same memory chunk.

17. **Information exposure through query strings in url**

    Information exposure through query strings in URL is when sensitive data is passed to parameters in the URL.

What are vulnerabilities? Cont..

18. **Injection problem**

    The basic form of this flaw involves the injection of control-plane data into the data-plane in order to alter the control flow of the process

19. **Insecure Compiler Optimization**

    Improperly scrubbing sensitive data from memory can compromise security.

20. **Insecure Randomness**

    Insecure randomness errors occur when a function that can produce predictable values is used as a source of randomness in security-sensitive context.

21. **Insecure Temporary File**

    Creating and using insecure temporary files can leave application and system data vulnerable to attacks.

22. **Insecure Third-Party Domain Access**

    Occurs when an application contains content provided from a 3rd party resource that is delivered without any type of content scrub.

What is Vulnerability Analysis?

- Vulnerability analysis is a procedure to check all the vulnerabilities in the systems, computers and other ecosystem tools. The vulnerability analysis helps in the analyzing, recognizing and ranking of the vulnerabilities as per the severity. It helps with the identification and assessment of threat details, enabling us to keep a resolution to protect them from hackers. The analysis can be done for every industry from Healthcare to Retail to IT.

Objectives of the Vulnerability analysis

1. To identify vulnerabilities – Configuration, system, Design, Code, Process
2. Documenting the vulnerabilities
3. Preparation of guidance to mitigate the vulnerabilities

Importance of Vulnerability Analysis

- Deep dive insights of the security issues
- Helps us understand the risks associated with the entire ecosystem
  - For security breaches
- Assets that are prone to cyber attacks

Steps for the vulnerability Analysis

Step 1 Assess Critical Value of each device	Review all the devices in the network Who are the people accessing the devices Capture the below information Risk Impact Risk threshold Risk strategy planning Mitigation Business impact analysis
Step 2 Details of the installed systems	Systems – What they do For whom the devices are installed Review – Device open ports Configuration of the devices Drivers of the devices which are certified Device vendor, version details Software installed on the devices Software installed on the devices
Step 3 Vulnerability Scanning	Compliance requirements checking Scan Policy formation Scanning – Single or Multiple times
Step 4 Report Creation	Vulnerability name Vulnerability Discover date Common Vulnerabilities Risk Score, Systems affected Method to fix them

How to check if the organization requires Vulnerability Analysis

Types of Vulnerability Assessment Cont..

- **Network Based Scans**
- To identify network vulnerabilities. This scan helps to find the vulnerable systems in the wired and wireless networks

- **Host Based Scans**
- This scan is to identify vulnerabilities in the ports, configuration, server workstations, other hosts and patch history

Types of Vulnerability Assessment Cont..

- **Wireless Network Scans**

Complete scan on wireless networks to find the vulnerabilities

- **Application Scans**
- To test all portals and mobile applications for vulnerabilities
- **Database Scans**
- To scan all the databases for potential vulnerabilities

Models of Vulnerability in Ethical Hacking

- **Firewall model**
- Insider attacks - A Perimeter firewall should be decided and this can take care of the external attacks
- Missed security patches
  - When the patch management of firewall has not happened
- Configuration issues
  - If there are faults in the configuration of firewall
- DDOS attacks
  - Only allow legitimate traffic to avoid these attacks

Models of Vulnerability in Ethical Hacking cont..

- **Password model**
- To crack the password the hacker uses any of the following – Dictionary, Hybrid model and Brute force
- **Logical Bombing**
- This usually happens when the hacker uses a malicious code to inject the web application or the cloud infrastructure
- **Web Hijacking**
- This happens when an unauthorized user tries to access the application bypassing the authorization mechanism

Protection from Hacking

We need to follow some simple steps to prevent hacking

- Updating of Operating systems
- Installation of the proper firewall to prevent intrusion
- Destroying all personal information from all the web sources
- No use of Open Wi-Fi
- Password – Strong password which is not easy to find out
- Smart emailing – Avoid opening of phishing mails
- Keep the sensitive data in the protected environment
- Ignore spam
- Shut down the systems after use
- Secure the network
- Back up the data

Vulnerability Assessments

- There are type of vulnerability assessments below:
- **Active assessments –** Which is the process to send request to the live network directly.
- **Passive assessments** – Which is the process discover vulnerabilities, open ports, and etc without sending request the target hosts.
- **External assessments** – Here hackers use techniques to find vulnerabilities of system from outside.
- **Internal assessments –** Here hackers use techniques to find vulnerabilities of system from internally.

The Vulnerability assessments Life Cycle allow organizations to identify system security weaknesses, prioritize assets, and remediate that and verify that they have been eliminated. There are following phases of the Vulnerability assessments Life Cycle:

Vulnerability Assessments

- **Creating Baseline** – Here performing assessment to find out details including hosts, network, application and other open services.
- **Vulnerability Assessment** – Here collecting all found vulnerabilities of the system and priorities that .
- **Risk Assessment** – Here defining the impact of vulnerabilities to an organization
- **Remediation** – Here prioritize and fix vulnerabilities in order according to business risk
- **Verification** – Here verifying that all vulnerabilities have been eliminated on organizations.
- **Monitor** – The end always must monitoring the network and system to avoid unsuspected attacks.

Conclusion

- In this article we have discussed the various vulnerabilities that hackers can exploit to gain unauthorized access to a system. Best practices and techniques on how to find the vulnerabilities are also discussed. We have discussed the analysis of vulnerabilities and how it helps in preventing the system from being hacked. Finally, we have discussed models of vulnerabilities in ethical hacking and the ways to keep ourselves protected from hacking.

Question Please?