

Team : Cyber 4

Mrunmayee Shirodkar : 2070

Divya Bhogle : 2043

Vrushali Walve : 2071

Manali Gawade : 2035

1. Initial Access

Image 1 : Overview

Initial Access TA0001

By: Mrunmayee Shirodkar
Cyber Security Intern

Short non-technical summary: Initial Access TA0001 refers to techniques that adversaries use to gain entry into target systems. These methods are the first step in a cyber attack, allowing attackers to establish a foothold in an organization's network.

What is Initial Access TA0001?

Initial Access TA001 involves the use of various techniques by attackers to gain an initial foothold in a target network. These methods serve as the gateway for launching cyber attacks, such as data exfiltration, ransomware, and espionage.

Key Targets

- Corporate Networks**: Internal systems and sensitive data within an organization's network
- Web Servers**: Public-facing servers hosting websites and applications
- Employees**: Individuals within an organization who might be susceptible to social engineering

Common Techniques

- Phishing**: Sending malicious emails with malware-infected attachments or links
- Exploiting Public-Facing Applications**: Attacking vulnerabilities in web applications that are accessible from the internet
- Drive-By Compromise**: Hosting malicious code on legitimate websites to infect visitors
- Valid Accounts**: Utilizing stolen or compromised user credentials to gain access
- Removable Media**: Using infected USB drives or other removable media to spread malware

Image 2 : Technical Details

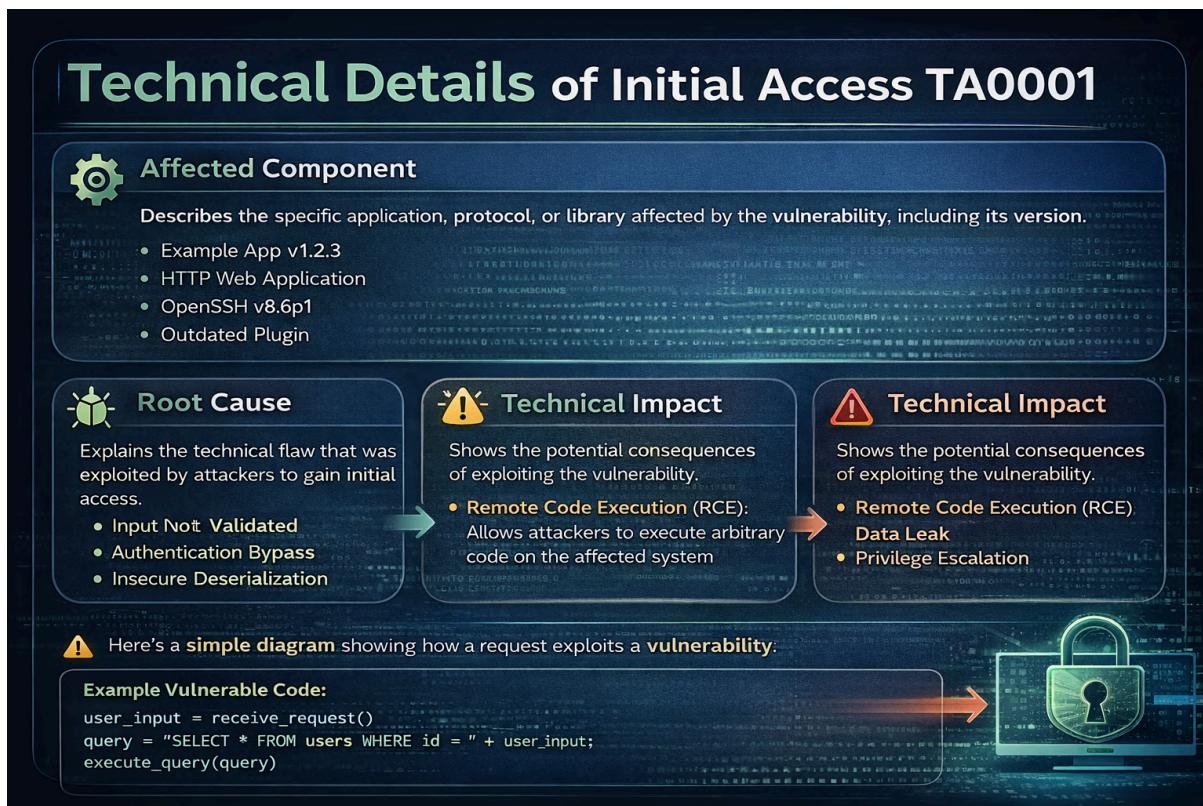
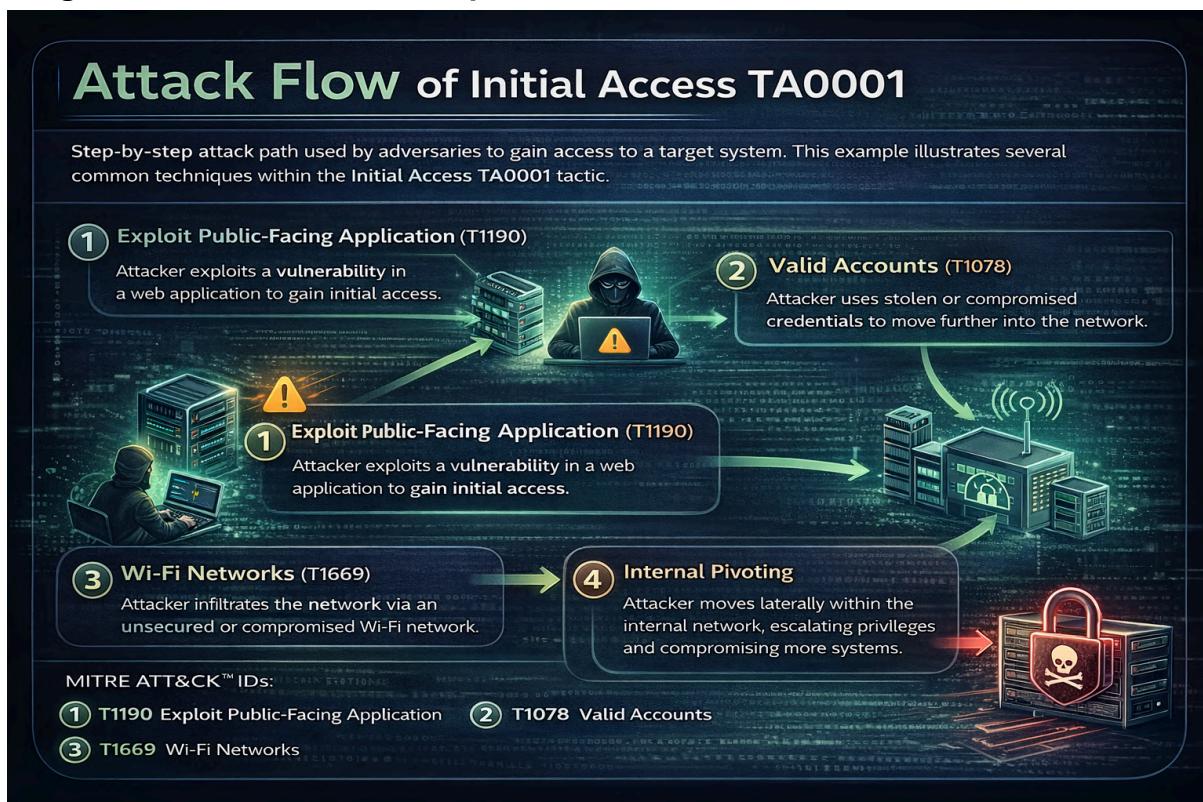


Image 3 : Attack Flow / Technique



2. Execution :

Image 1 : Overview

EXECUTION TA0002
Understanding the cyber threat of Execution TA0002

WHAT IS EXECUTION TA0002?

Execution (TA0002) is a technique used by cyber attackers to run malicious code on a target system. It allows hackers to take control by executing harmful programs or scripts.

Think of it as a way for attackers to make a computer do whatever they want, as if they have taken over the keyboard.

WHERE DOES IT APPEAR?

Common methods hackers use Execution TA0002 are common ways to use:

- Phishing Emails**: Clicking on malicious links or attachments.
- Malicious Files**: Opening infected documents or software.
- Command Line**: Directly executing commands in the terminal or command prompt.
- Startup Scripts**: Programs automatically running when a system starts.

Intern: Mrunmayee Shirodkar – Cyber Security Intern

Image 2 : Technical Details

TECHNICAL DETAILS (TT0002)
EXECUTION TA0002

Affected Component

<Vulnerable App>
Version 2.1.4
• <App Name>
• LibraryName (v1.2.3)
• <Protocol>

Root Cause

• Input not validated
• Unsafe deserialization
• Authentication bypass

Technical Impact

• Remote Code Execution (RCE)
• Data leak
• Privilege Escalation (PrivEsc)

Example Vulnerable Code or Library:

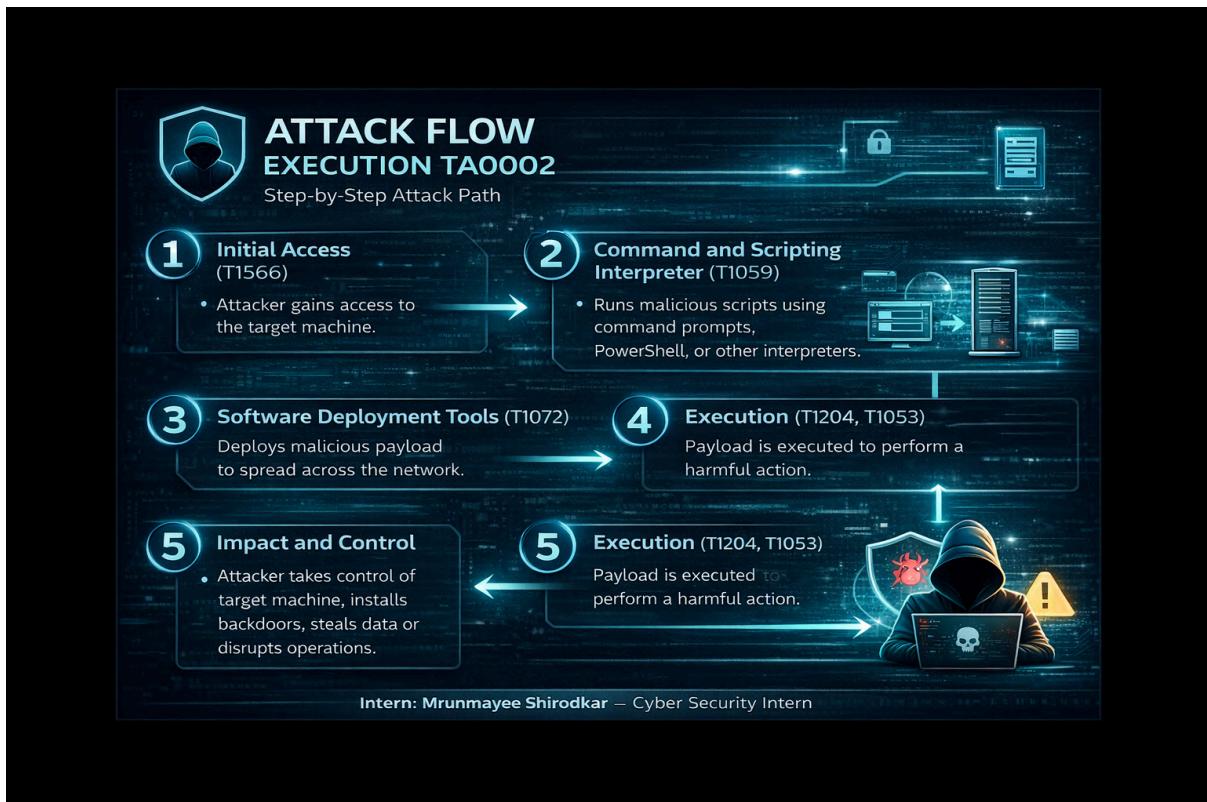
```
input = get_user_input();
deserialize(input)
// Insecure deserialization
```

Request → Vulnerable Code → Result

Request → Vulnerable Code → Result

Intern: Mrunmayee Shirodkar – Cyber Security Intern

Image 3 : Attack Flow / Technique



3. Persistence

Image 1 : Overview



Image 2 : Technical Details



Image 3 : Attack Flow / Technique



4. Privilege Escalation

Image 1 : Overview

Privilege Escalation (TA0004)

What is Privilege Escalation?

Privilege Escalation occurs when an attacker gains higher access rights than they are supposed to have. This allows them to perform actions that normal users shouldn't be able to do.

Where Does It Appear?

- **Privilege Escalation** vulnerabilities can exist in:

- Operating Systems
- Software Applications
- Misconfigured Systems

Types of Privilege Escalation

- Vertical Privilege Escalation**
 - Attacker elevates their privileges from a regular user to an admin (e.g., gaining root/admin access)
- Horizontal Privilege Escalation**
 - Attacker gains access to another user's account or resources at the same privilege level

Potential Consequences

- Full control over the system
- Access sensitive data
- Install malware or backdoors
- Disrupt or damage the system


Divya Bhogle
Cyber Security Intern

Image 2 : Technical details :

Privilege Escalation: Technical Details

Affected Component

System components that can be exploited:

- Operating Systems
- Software Applications
- Network Services

Root Cause

- Common flaws in code or configuration that can lead to:
- Weak permissions settings
- Unpatched software vulnerabilities
- Insecure coding practices

Technical Impact

- The exploitation of privilege escalation vulnerabilities can lead to:
- Gaining admin or root-level access
- Disabling security controls
- Accessing sensitive data or executing arbitrary code

Attack Flow Diagram

Attacker → REQUEST → VULNERABLE CODE → RESULT → Administrator

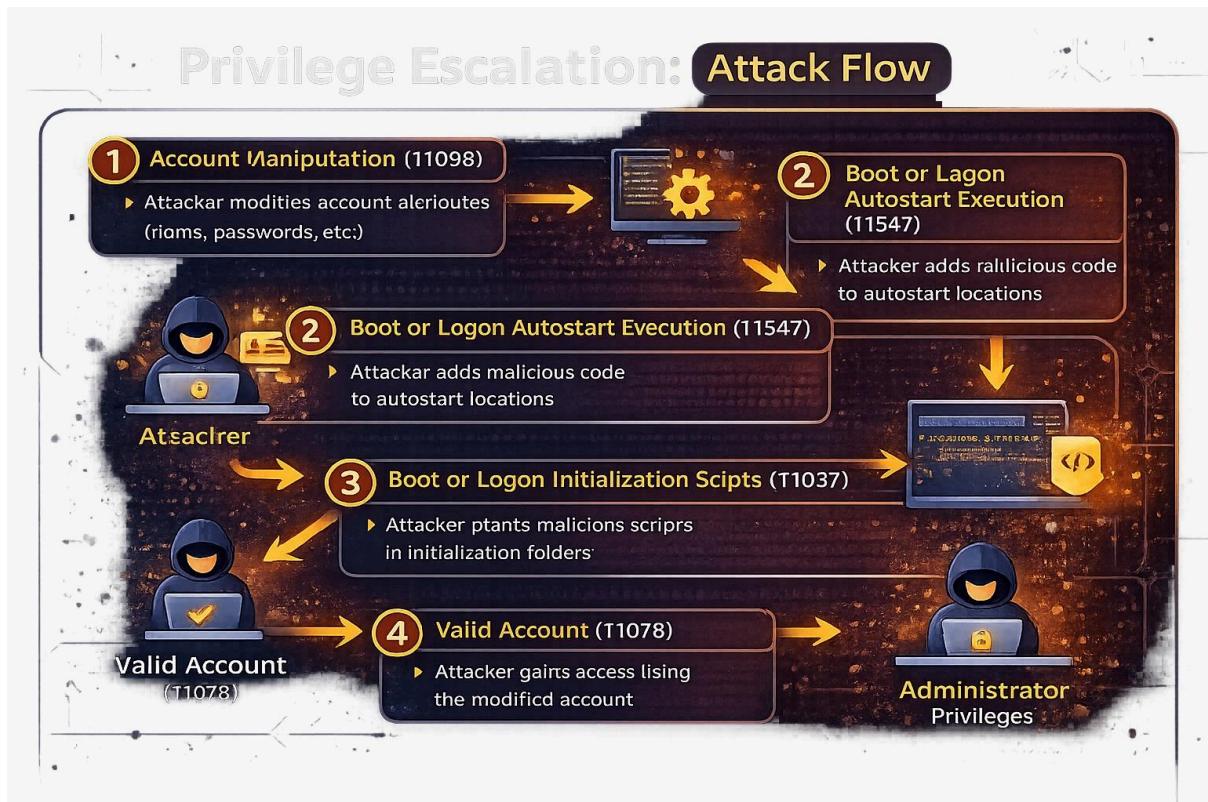
```
If (request.user == "admin") {  
    grant_admin_access();  
}
```

System Components for Escalation

System components that can be exploited to escalate privileges, such as:

- Operating Systems
- Software Applications
- Network Services

Image 3 : Attack Flow / Technique



5. Defense Evasion (TA0005)

Image 1 : Overview



Image 2 : Technical Details

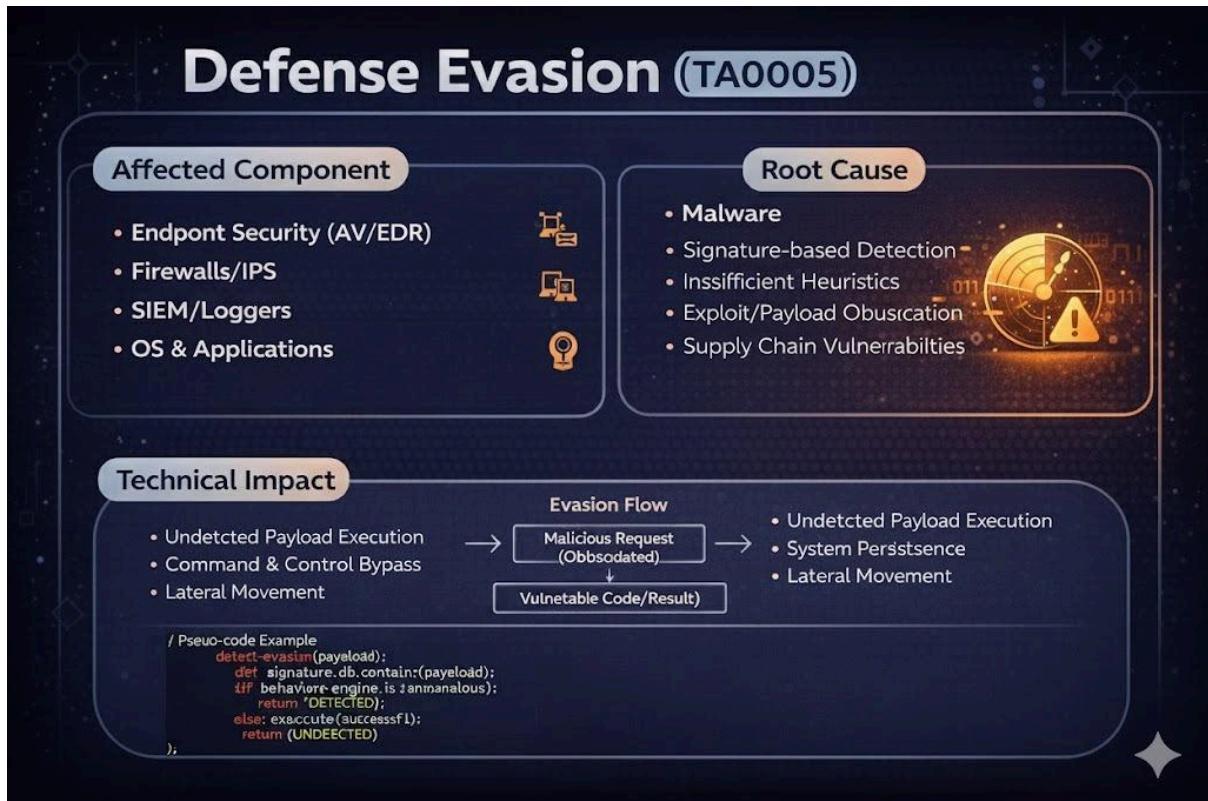
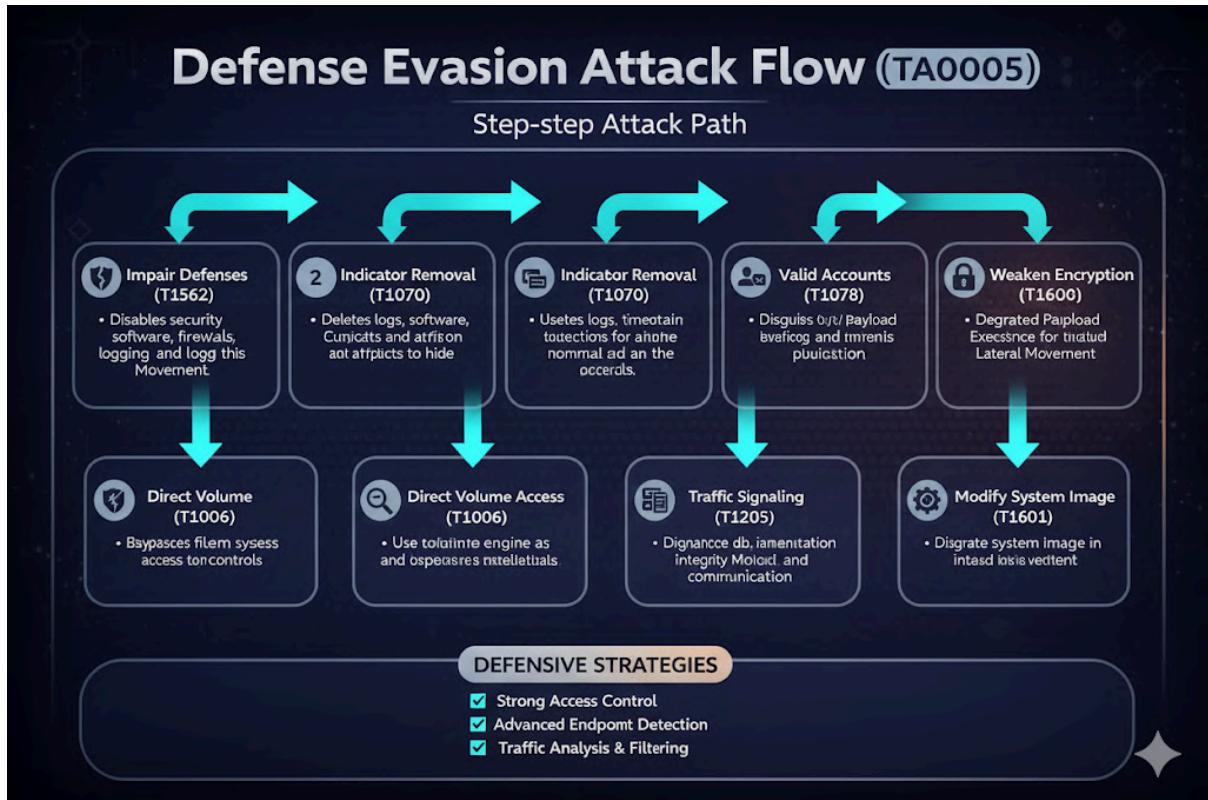


Image 3 : Attack Flow / Technique



6. Credential Access

Image 1 : Overview

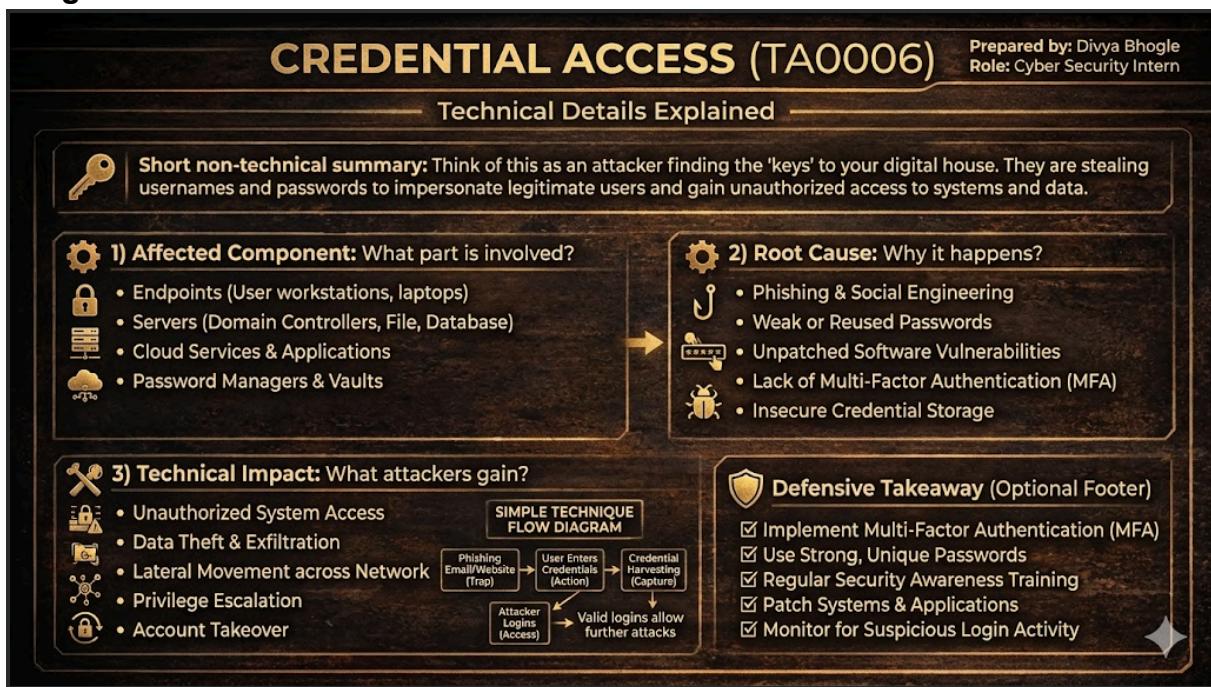


Image 2 : Technical Details

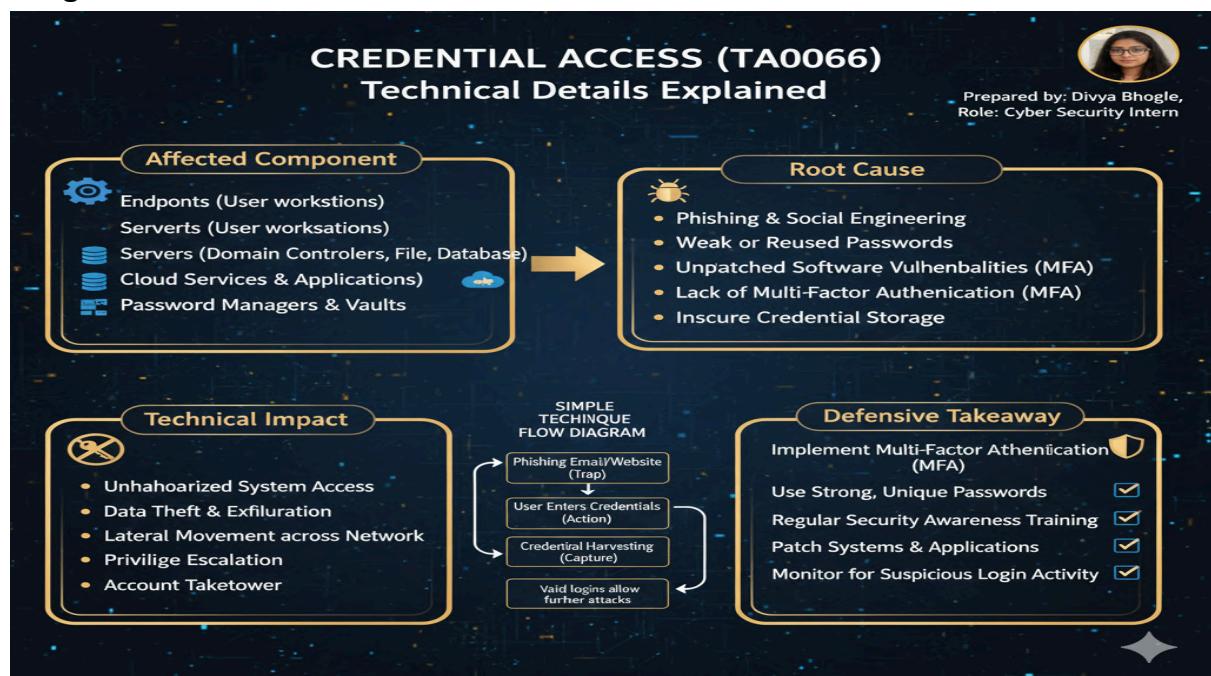
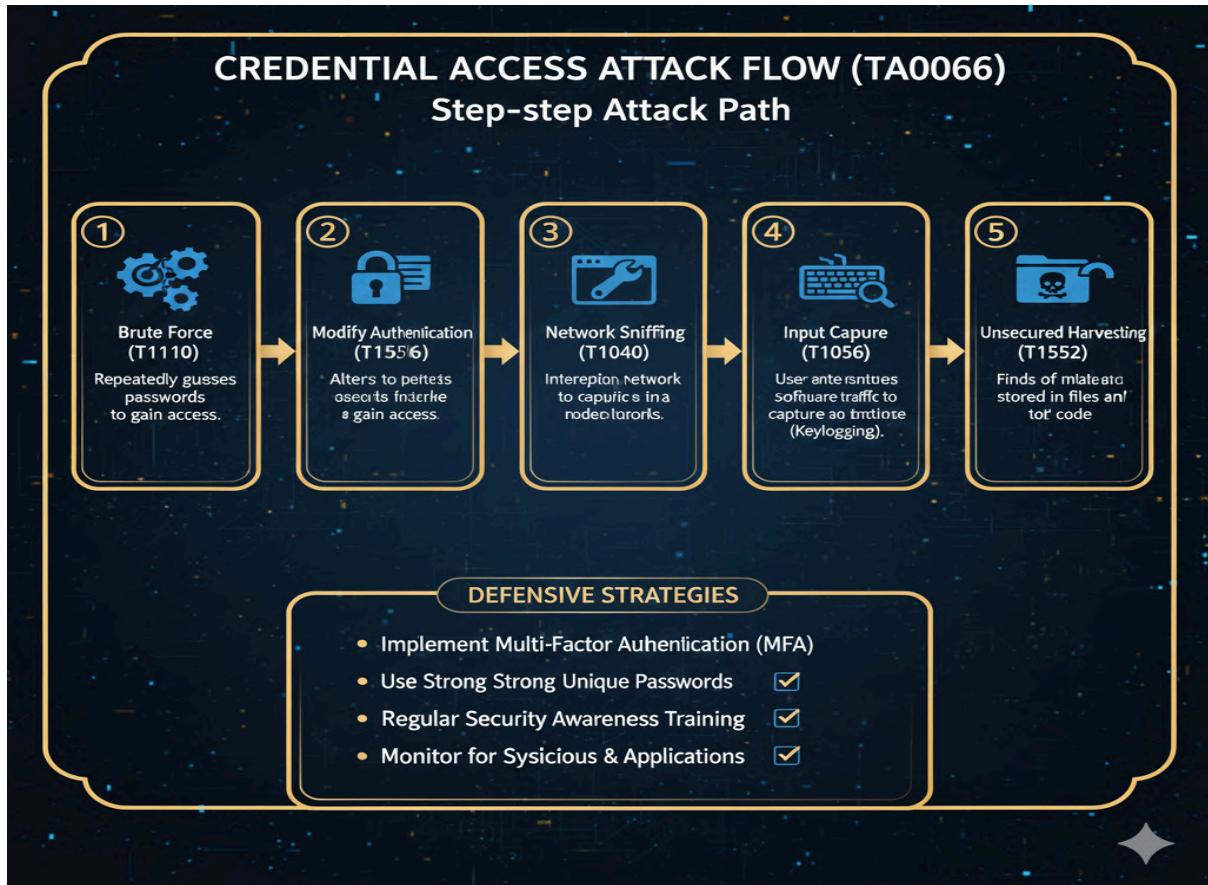


Image 3 : Attack Flow / Technique



7. Discovery (TA0007)

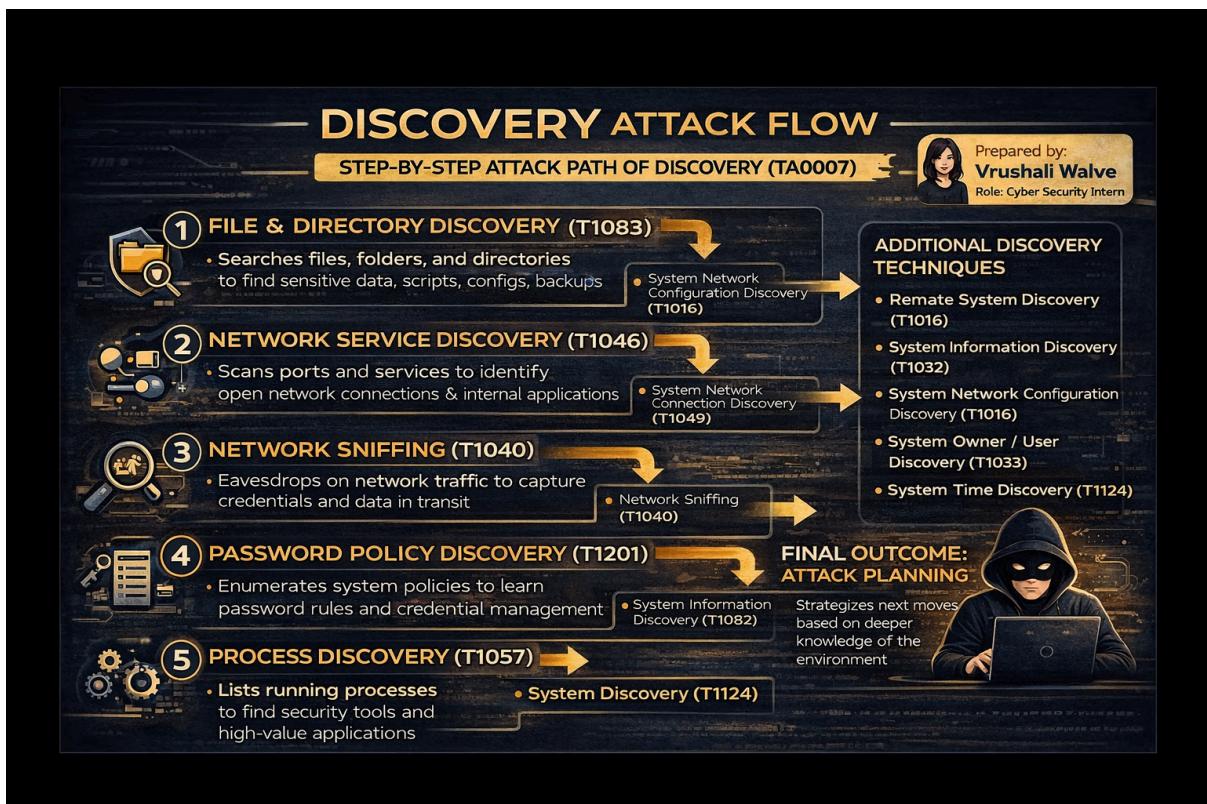
Image 1 : Overview



Image 2 : Technical Details



Image 3 : Attack Flow / Technique



8. Lateral Movement

Image 1 : Overview



Image 2 : Technical Details

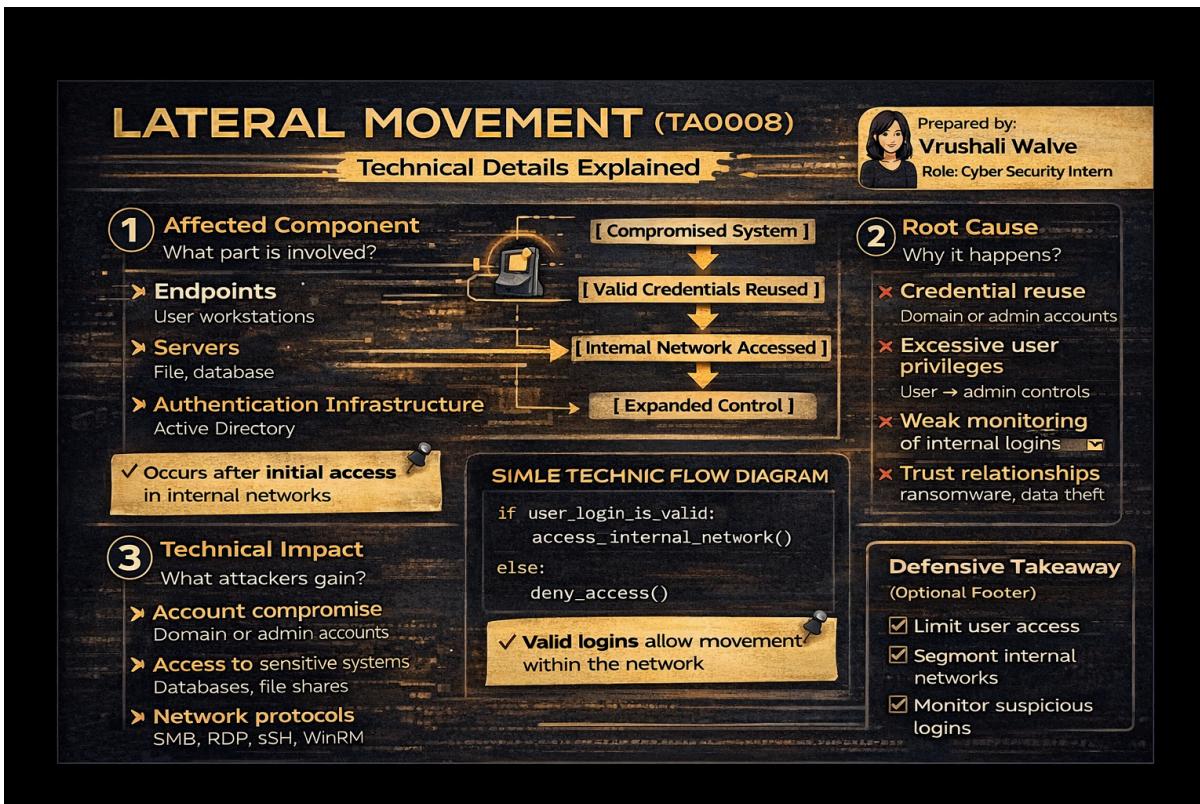
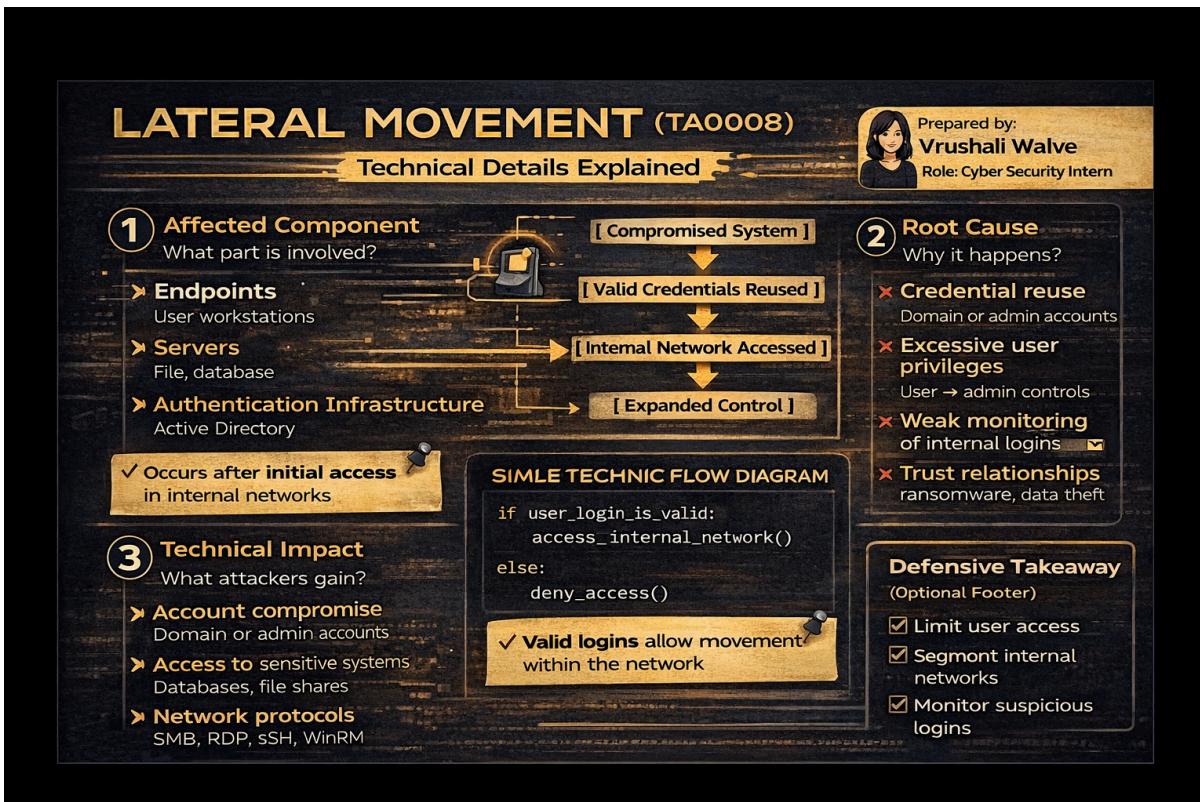


Image 3 : Attack Flow / Technique



9. Collection (TA0009)

Image 1 : Overview



Image 2 : Technical Details



Image 3 : Attack Flow / Technique



10. Command and Control

Image 1 : Overview

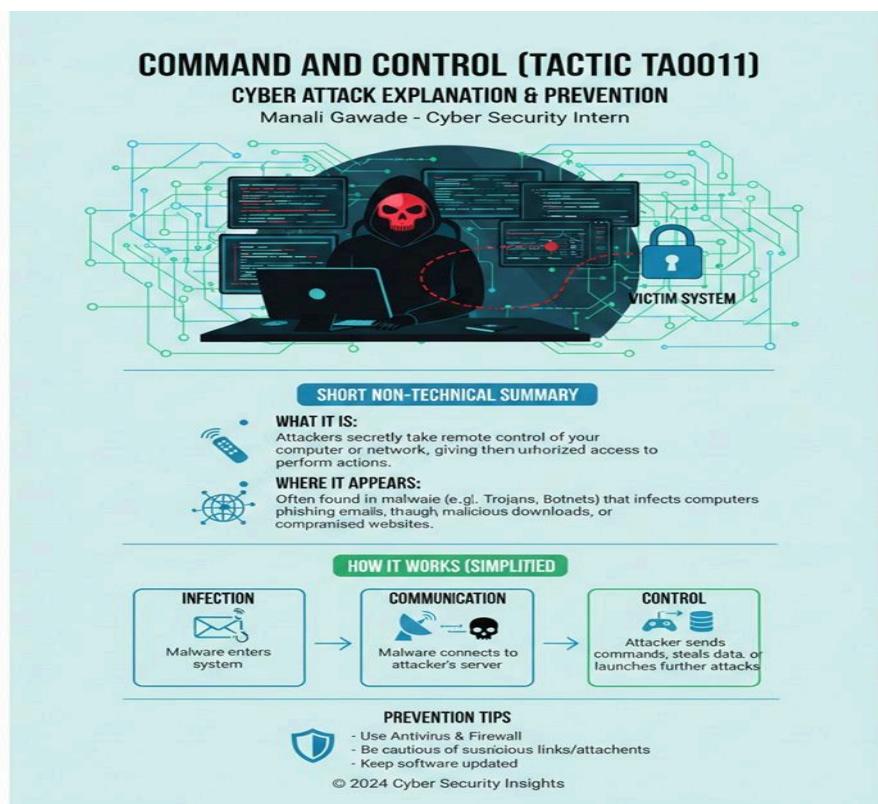


Image 2 : Technical Details

COMMAND AND CONTROL (TACTIC TA0011)

TECHNICAL DETAILS & PREVENTION

Manali Gawade - Cyber Security Intern

TECHNICAL EXPLANATION

AFFECTED COMPONENT

- Application: Custom Web Server (v.x)
- Protocol: HTTP/HTTPS(Custom API)
- Protocol: (Custom API)
- Library "json-parser.dll", application (1.2)
- Versions: All preceding 'preeder.dll' (1.2)
- Versions: All preceding v.3

ROOT CAUSE

- Insecure Deserialization & Lack of Input Validation. Maliciously crafted JSON object in ody exploitss explos taw in flaw s "json-parser.dll library, istect excution diwing artlinary code payload. No authentication sariilization checks thie payload.

TECHNICAL IMPACT

- Remote Dada Exfiltration
- Privilege Escalaten & Persistence

EXPLITATION FLOW (SIMPILIFIED)

The diagram illustrates a simplified exploitation flow:

- 1. MALICIOUS REQUEST**: A Client sends a POST /api/cmd HTTP/1.1 request to the target.com host. The Body contains malicious JSON: "obj = deserializ(sesLeoley) execute (a) send".
- VULNERABLE CODE**: The server processes the request using vulnerable code.
- 2. SYSTEM COMPROMISE**: The system is compromised, as shown by a terminal window displaying "C:\>> whoami nt authority\system".

© 2024 Cyber Security Insights

Image 3 : Attack Flow / Techniques

COMMAND AND CONTROL (TACTIC TA0011)

ATTACK FLOW & TECHNIQUES

Manali Gawade - Cyber Security Intern

STEP-BY-STEP ATTACK FLOW

A central illustration shows a hacker wearing a hooded mask with a skull, sitting at a computer. A dashed red arrow points from the computer screen to a 'VICTIM SYSTEM' icon, which is a computer with a padlock symbol.

STEP-BY-STEP ATTACK PATH

The diagram details the step-by-step attack path:

- 1. INITIAL COMPROMISE INGRESS TOOL TRANSFER**: PHISHING & MALICIOUS DOWNLOAD Installs C2 agent agent (T1105).
- 2. TRAFFIC SIGNALING (T1090) & HIDE INFRASTRUCTURE**: with normal traffic.
- 3. VULNERABLE CODE**: ENCRYPTED CHANNEL PROTOCOL TUNNELLING & C2 comms.
- 4. ENCRYPTED CHANNEL (T1032) & TEME (T1572) (T1572)**: Evacts C2 deection.
- 5. COMMAND SEND & EXfiltration (T1041)**: Further Attacks

© 2024 Cyber Security Insights

11. Exfiltration

Image 1 : Overview

COMMAND AND CONTROL
CYBER SECURITY VULNERABILITY EXPLAINED

SHORT NON-TECHNICAL SUMMARY

- Malicious actors take remote compromised systems or networks.
- They issue commands to steal data, disrupt operations, or launch further attacks.
- Often uses covert communication channels.

WHERE IT APPEARS

- IOT Devices (Cameras, Routers)
- Enterprise Networks
- Personal Computers
- Mobile Devices
- Critical Infrastructure Systems

IMPACT:
Data Theft, Downtime, Financial Loss, Reputation Damage.

INTERN'S NAME:
Manali Gawade
ROLE: Cyber Security Intern

Image 2 : Technical Details

COMMAND AND CONTROL
TECHNICAL DETAILS EXPLAINED

AFFECTED COMPONENTS

- Applications: Web Servers (Apache, Nginx <v1.20), Custom IoT Apps
- Protocols: Protocols
- DNS, HTTP/HTTPS, IRC, custom C2 libraries
- Libraries: libcurl 0, custom <7.8.0, IRC
- Libraries: custom C2 libraries
- Versions: Varies widely, often unpatched or off-the-shelf
- Critical Infrastructure Systems

ROOT CAUSE

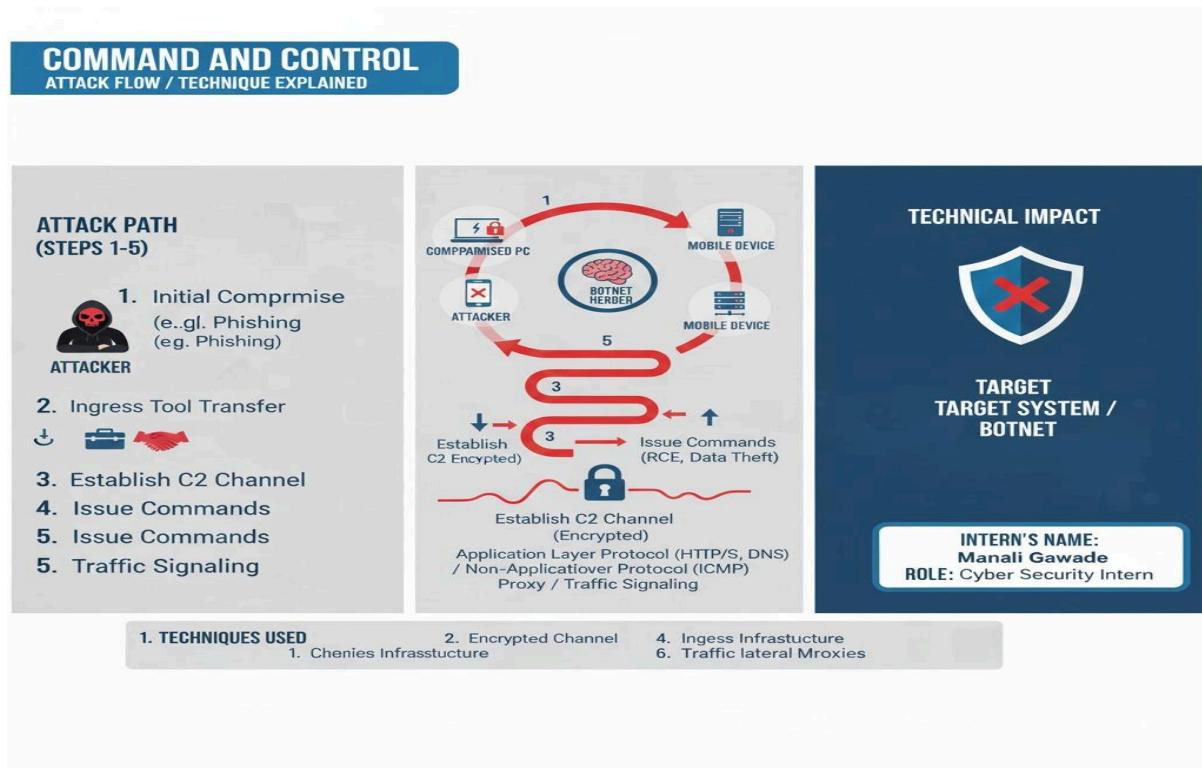
- Insecure Input Validation
- Authentication/Authorization Bypass
- Weak Encryption
- Weak Encryption/Obfuscation Vulnerabilities
- Exploited Unpatched Zero-Day
- Misconfigured Firewalls/Proxies

TECHNICAL IMPACT

- Remote Code Execution (RCE)
- Data Exfiltration
- Data Extraction/Leakage
- Denial of Service (DoS)
- Installation of Malware (e.g., Ransomware)
- Network Lateral Movement

INTERN'S NAME:
Manali Gawade
ROLE: Cyber Security Intern

Image 3 : Attack Flow / Techniques



12. Impact

Image 1 : Overview

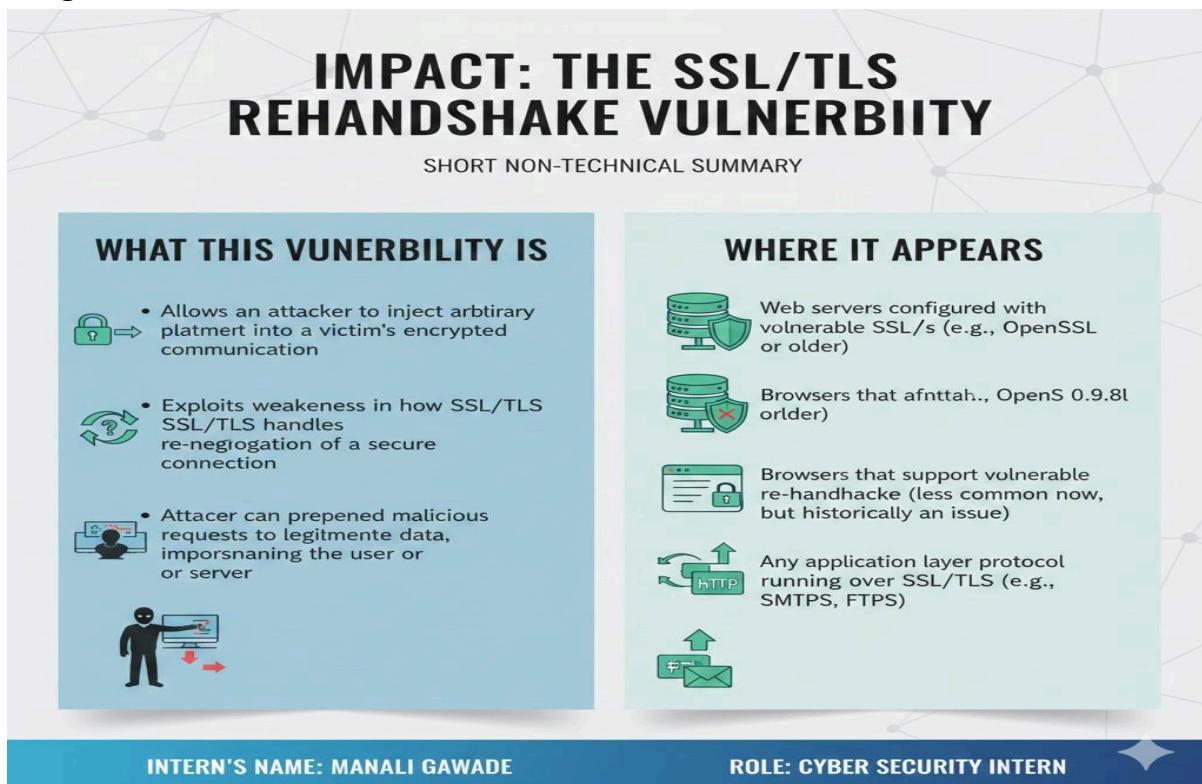


Image 2 : Technical Details

IMPACT: THE SSL/TLS REHANDSHAKE VULNERABILITY

TECHNICAL DETAILS

AFFECTED COMPONENTS & ROOT CAUSE

- Affected Components:**
 OpenSSL 0.9.8 through OpenSSL 1.0.0 through `srp1NSL` or later
- Root Cause:**
 OpenSSL 0.9.8, 0.9.8l, OpenSSL 1.0.0 through Mozilla NSS 3.12.x
- Weakness in SSL/TLS protocol design**
 where properly tie a no the new properly tie a to the ecurer allowing two platiereis data hs prevous injected data.



TECHNICAL IMPACT & MECHANISM



1. Attacker Handshakes (secure) Attacker inserts plaintext in new message and accept, unaware of an attacker injecting data.



PROTOCOL CONFUSION / DATA INJECTION
Allows attacker to inject arbitrary command/data into application layer protocols (e.g., HTTP headers, HTTPS headers, Auth Bypass, Data Leakage, or Data Loss, or other application-specific vulnerabilities).

INTERN'S NAME: MANALI GAWADE ROLE: CYBER SECURITY INTERN

Image 3 : Attack Flow / Techniques

IMPACT: THE SSL/TLS REHANDSHAKE VULNERABILITY

ATTACK FLOW / TECHNIQUE

STEP-STEP ATTACK PATH (1-5)

- 1. ATTACKER ESTABLISHES CONNECTION**
 - Attacker initiates a secure SSL/TLS handshake with own vulnerable server.
- 2. VICTIM CONNECTED INITIATED**
 - Legitimate client establishes new handshake with the chosen vulnerable server.
- 3. REHANDSHAKE SERVER**
 - Attacker initiates SSL/TLS handshake messages with victim's secure session.
- 4. INJECTION & CONFUSION**
 - Server incorrectly associates new attack's message with the legitimate client's secure connection.
- 5. EXPLOITATION**
 - Disk Wipe
 - Firmware Corruption
 - Inhibit System Recovery
 - System Shutdown/Reboot

INTERN'S NAME: MANALI GAWADE ROLE: CYBER SECURITY INTERN