Threat Intel Report on

MITRE ATT&CK® Framework macOS Platform

by

Team Data Wizards

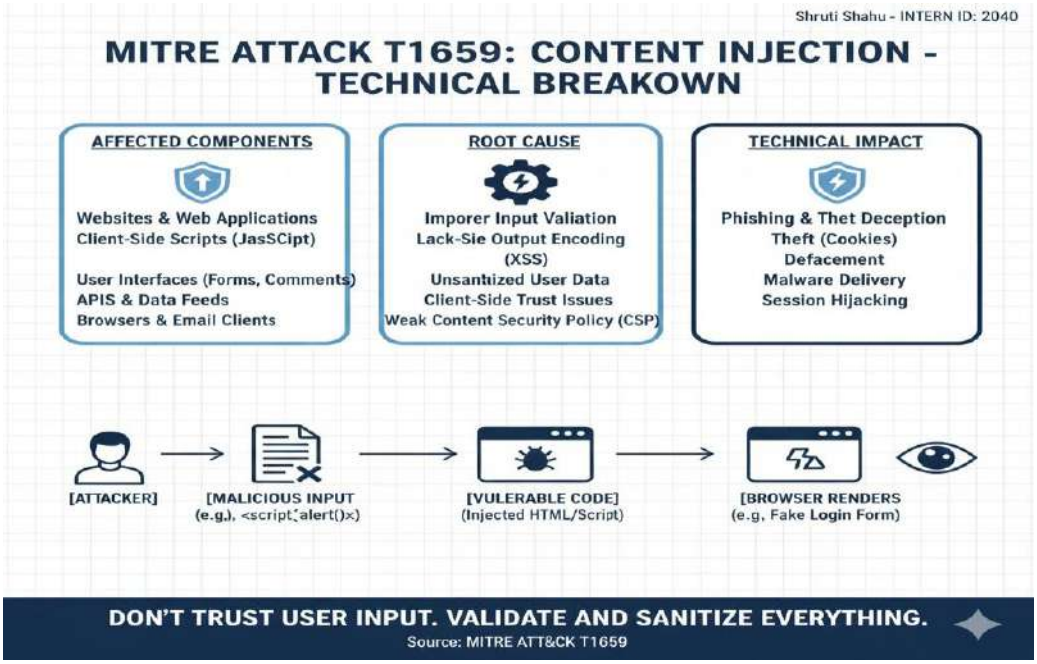
(Shruti Shahu - 2040, Sanika Raul - 2041,

Riya Kadam – 2044, Vibhuti Naik - 2046)

Initial Access

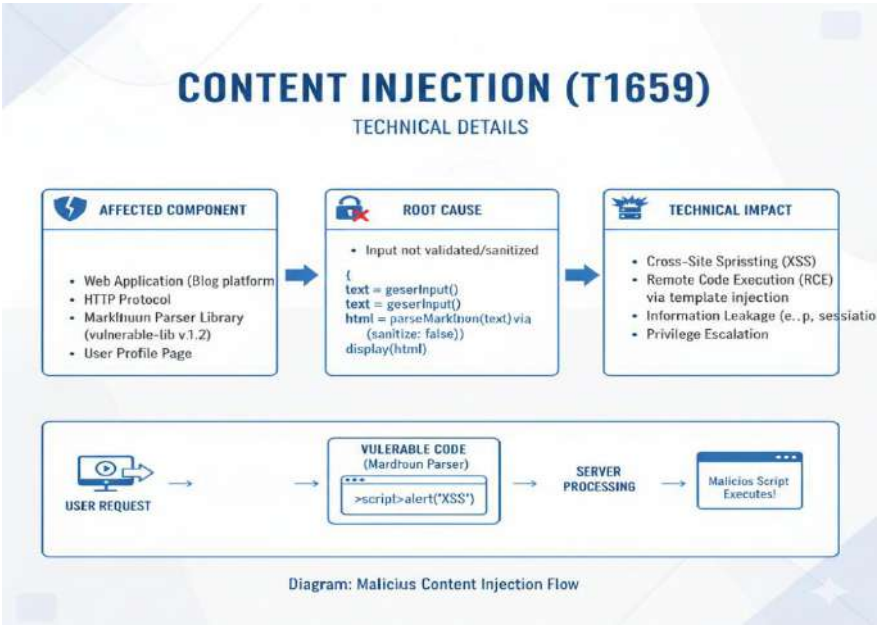
Content Injection(T1659)

Overview



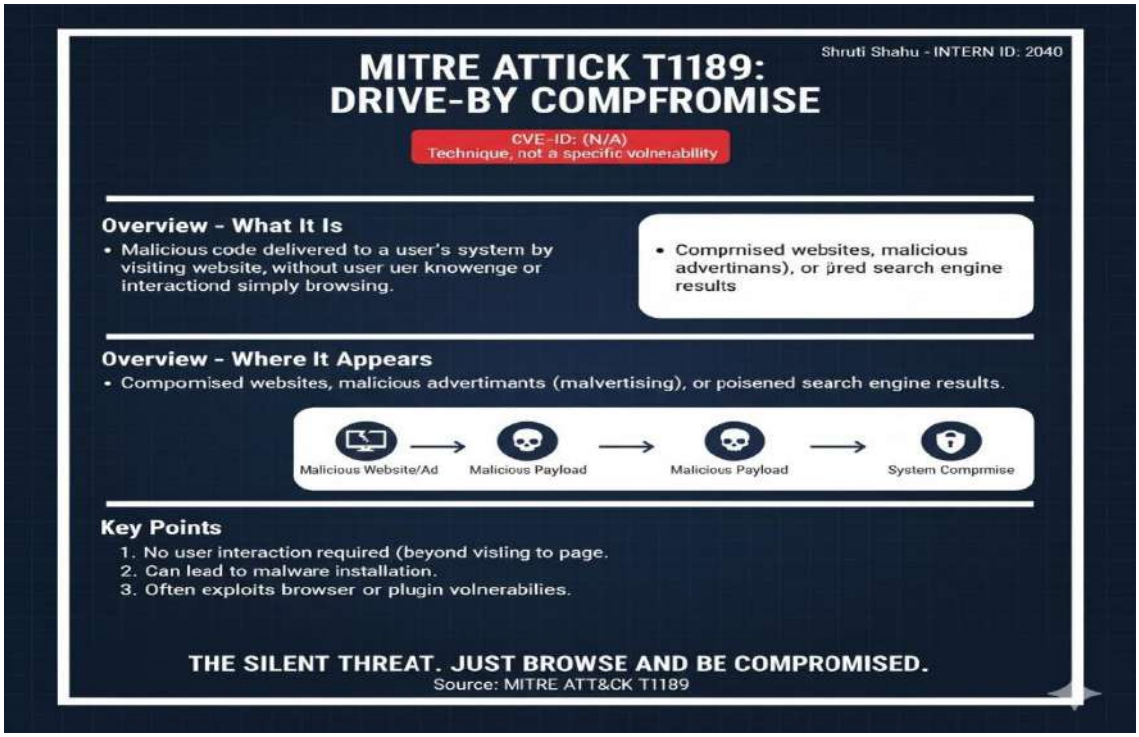
Technical Details

Attack Flow / Technique



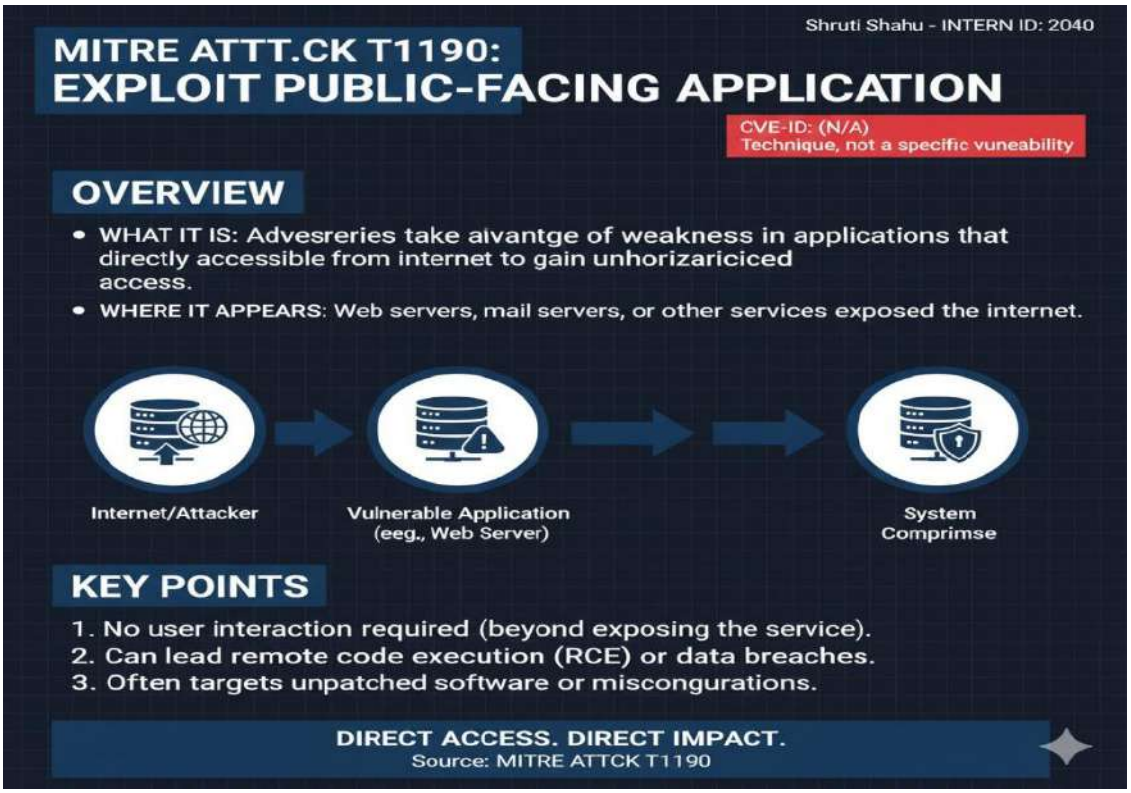
Drive-By Compromise(T1189)

Overview



Exploit Public-Facing Application(T1190)

Overview

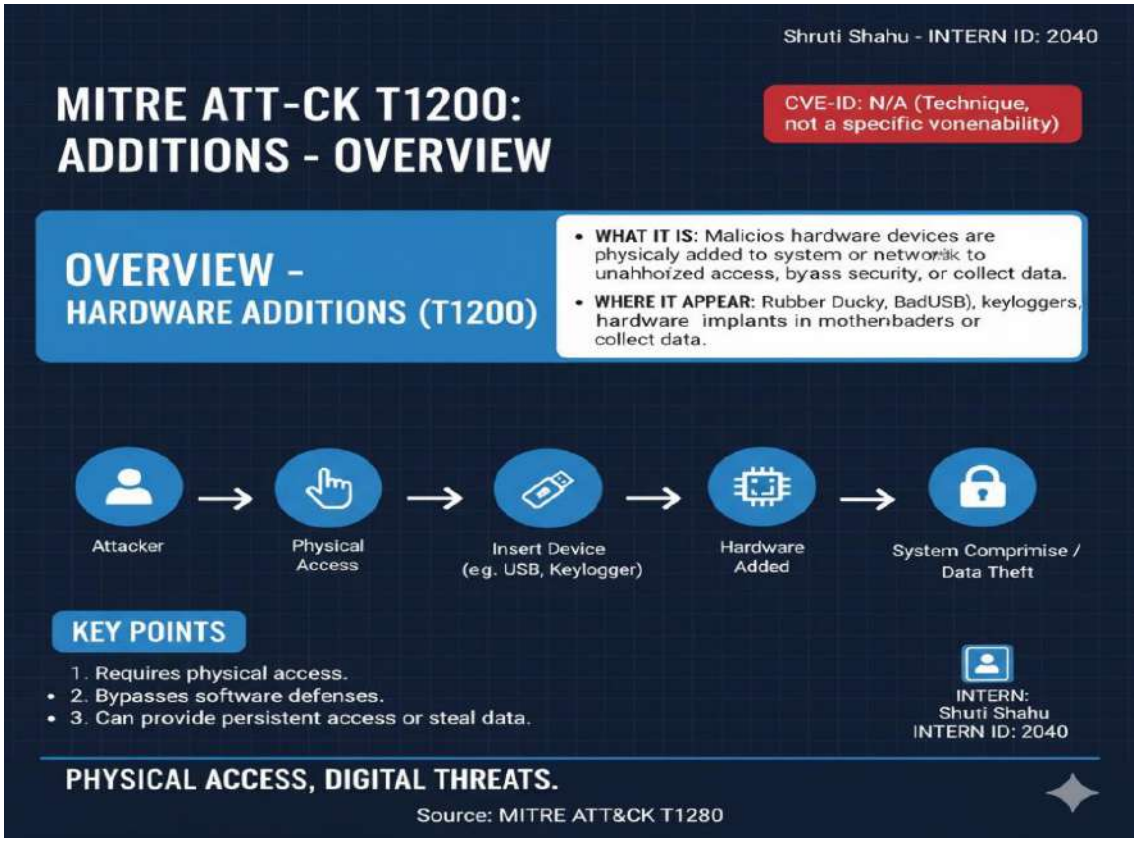


External Remote Services

Overview

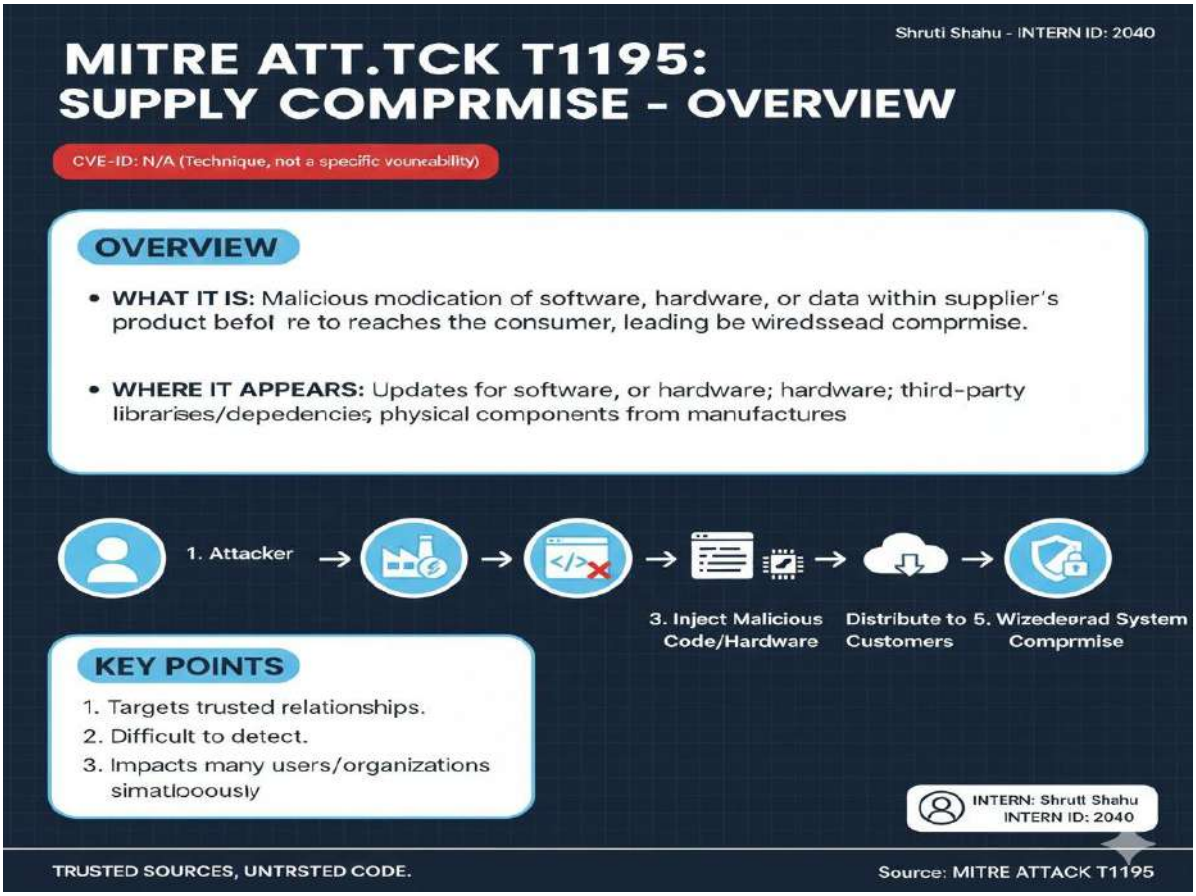
Additions-Overview(T1200)

Overview



Supply Comprmise(T1195)

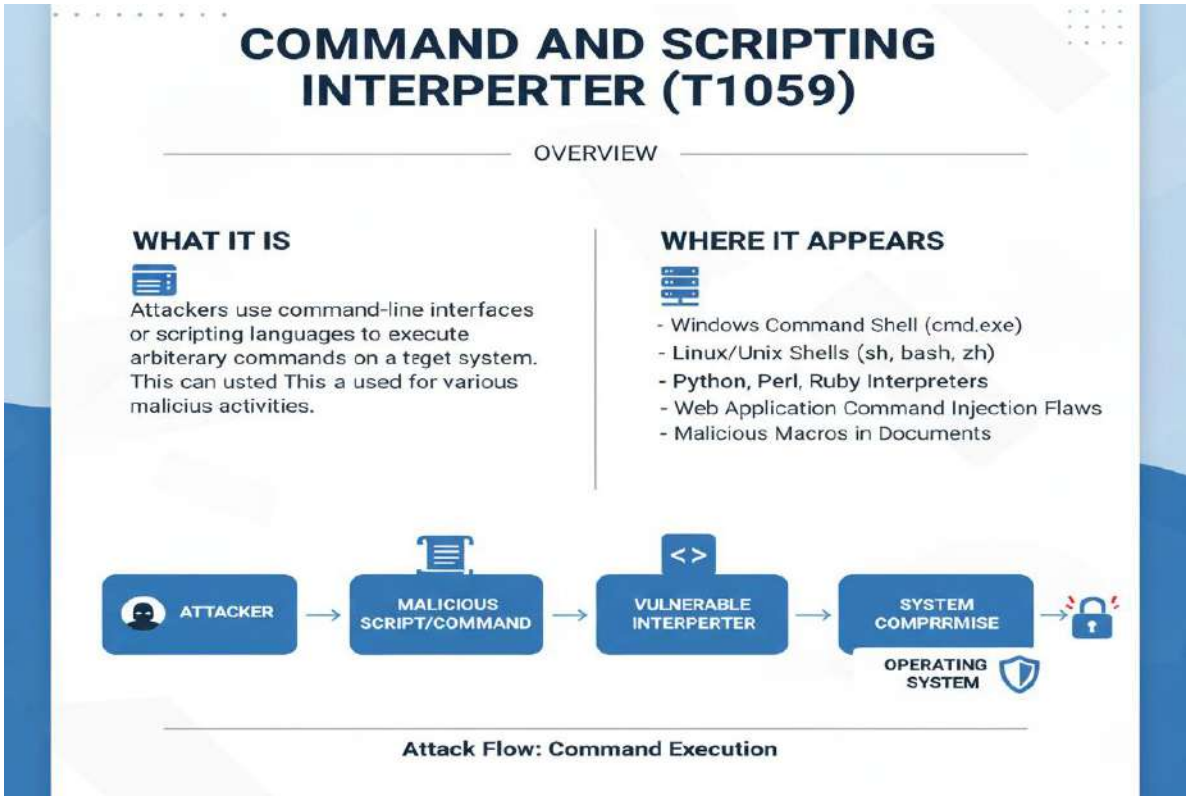
Overview



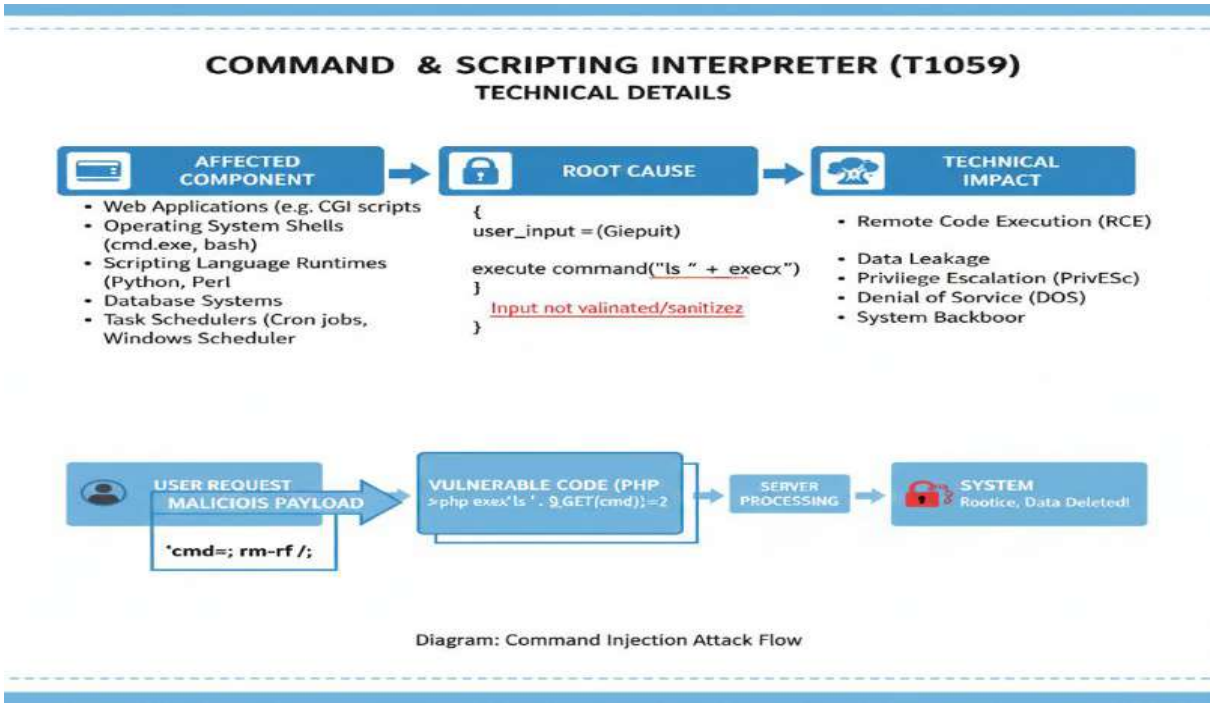
Excution

Command and Scripting Interpreter(T1059)

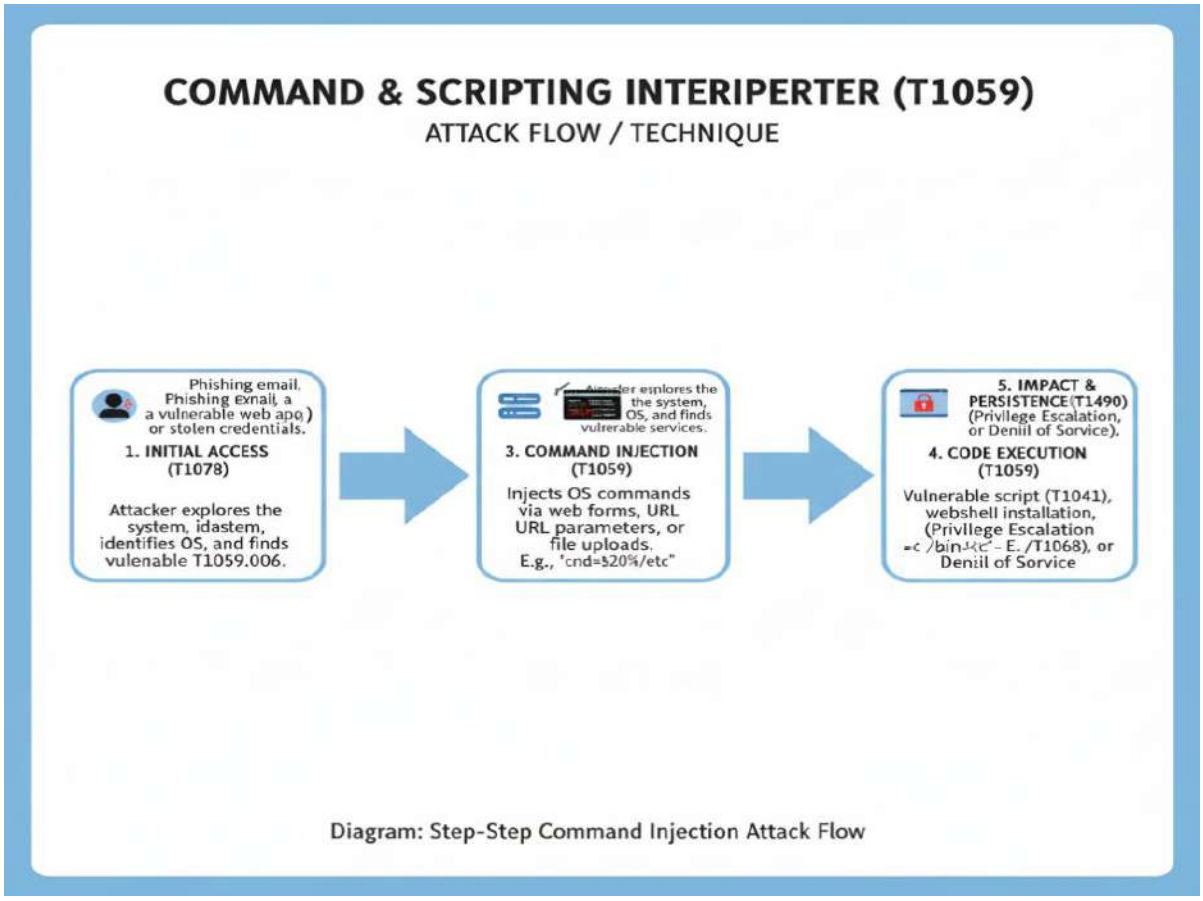
Overview



Technical Details

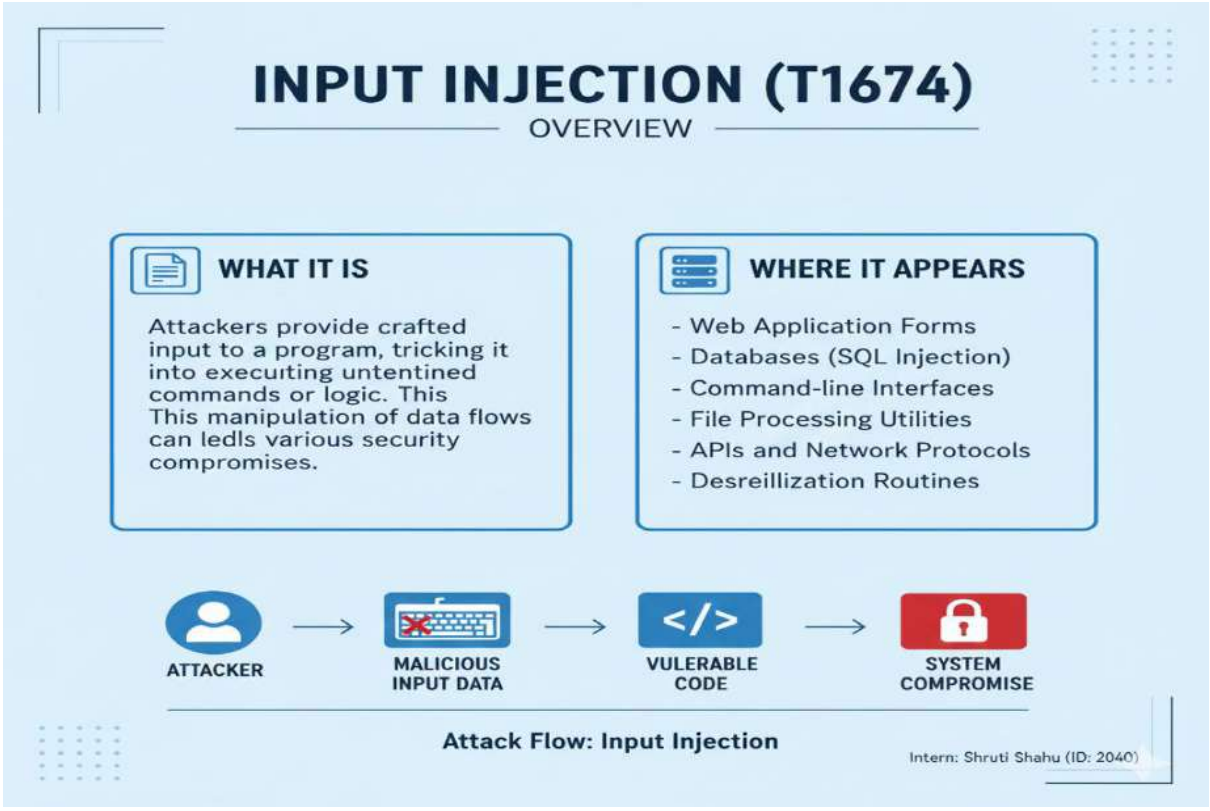


Attack Flow / Technique



Input Injection(T1674)

Overview



Technical Details

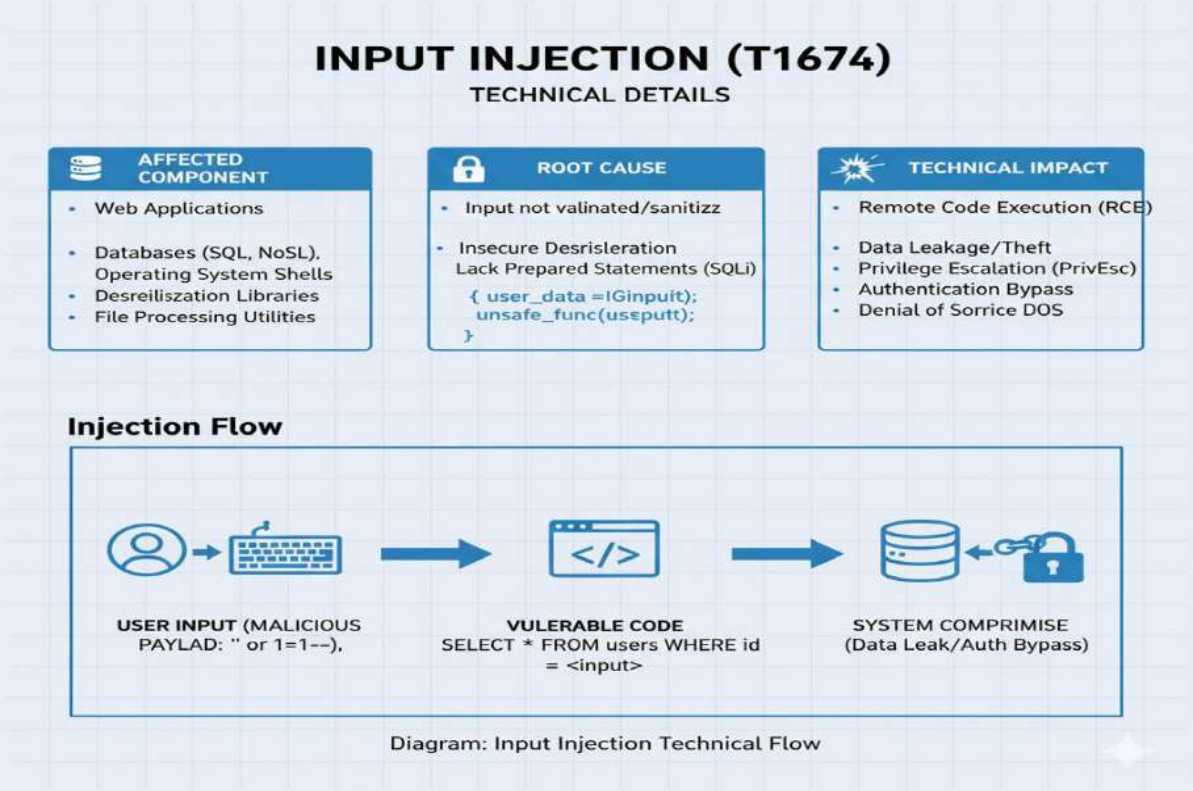


Diagram: Input Injection Technical Flow

Attack Flow / Technique

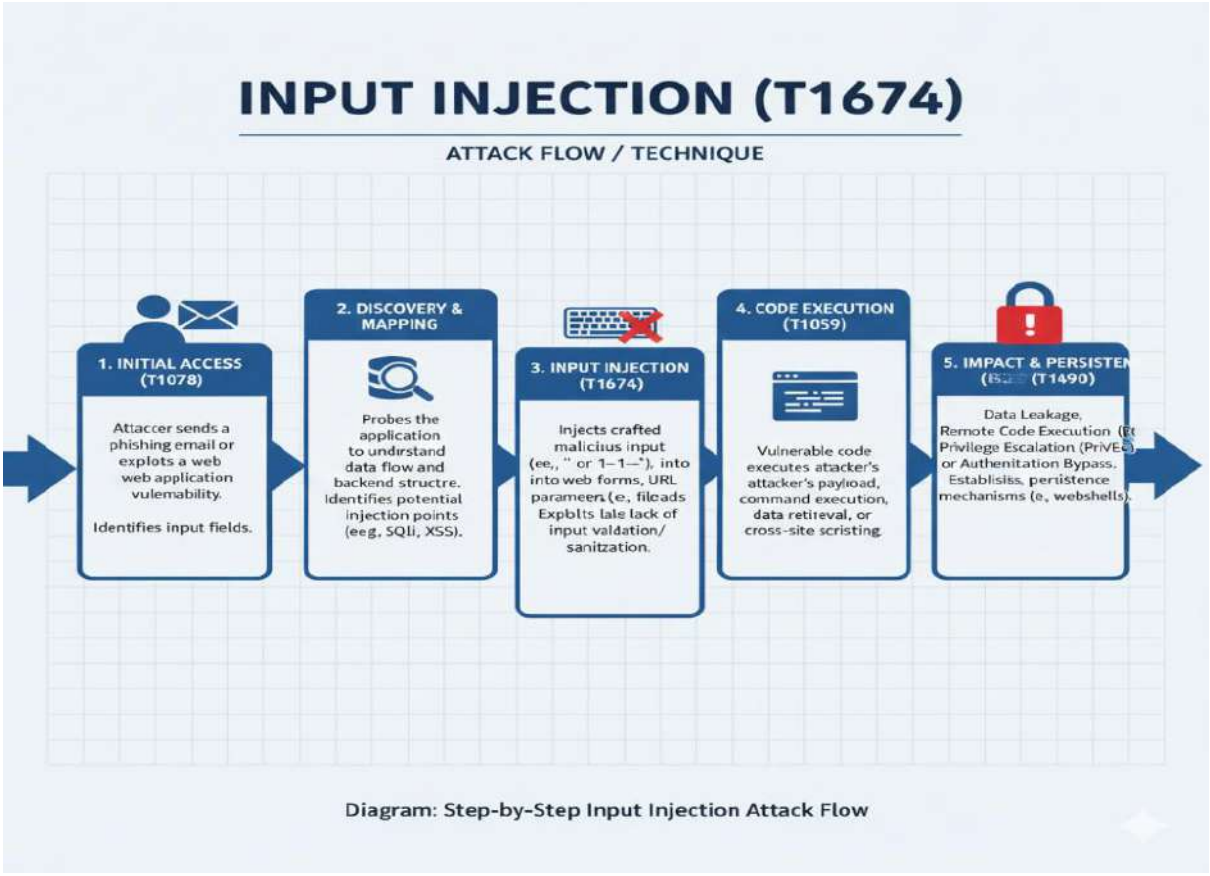
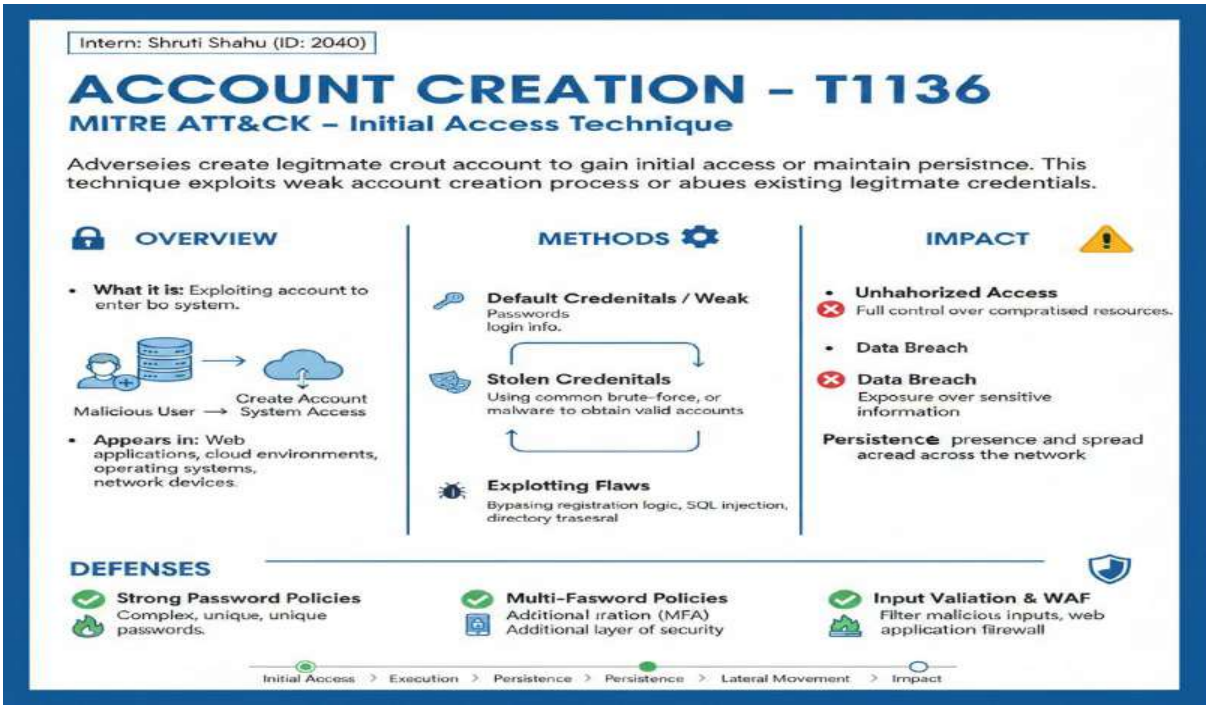


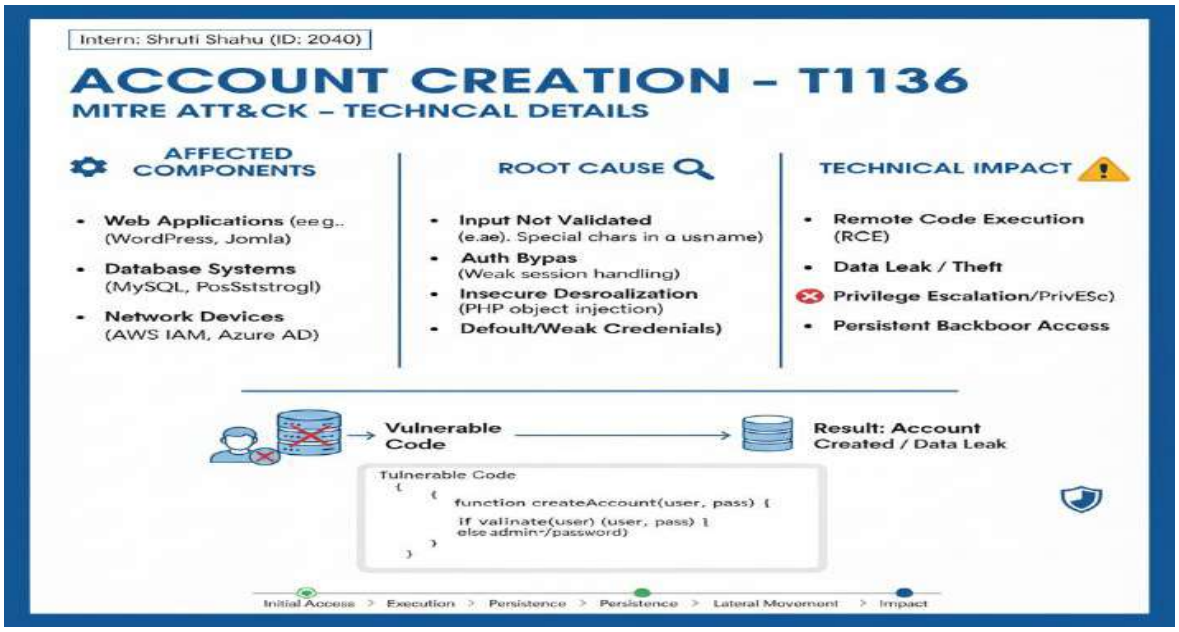
Diagram: Step-by-Step Input Injection Attack Flow

Create Account(T1136)

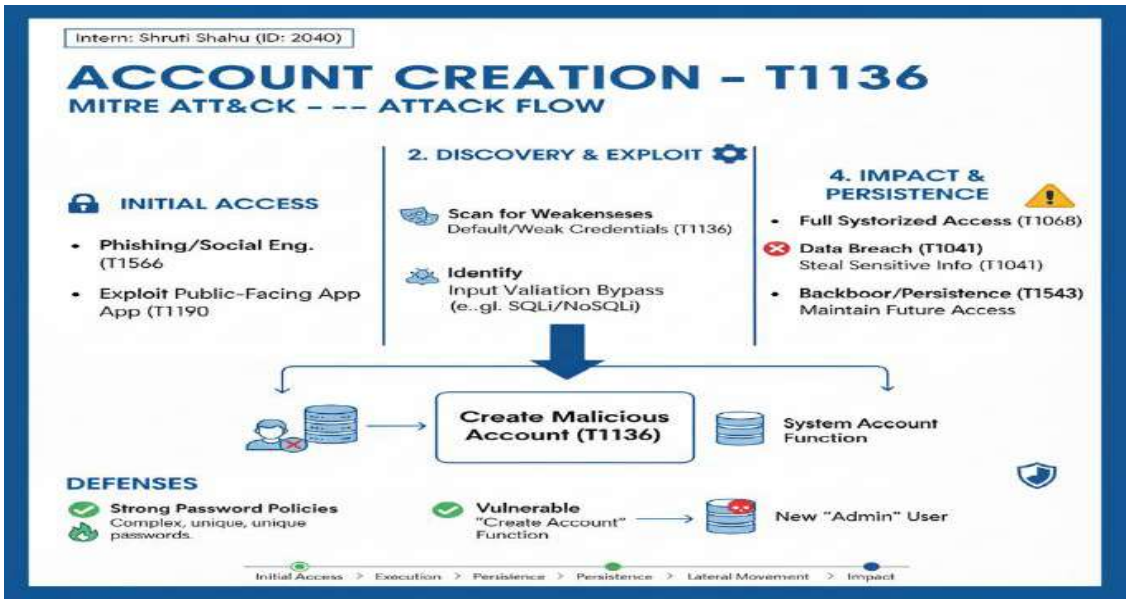
Overview



Technical Details

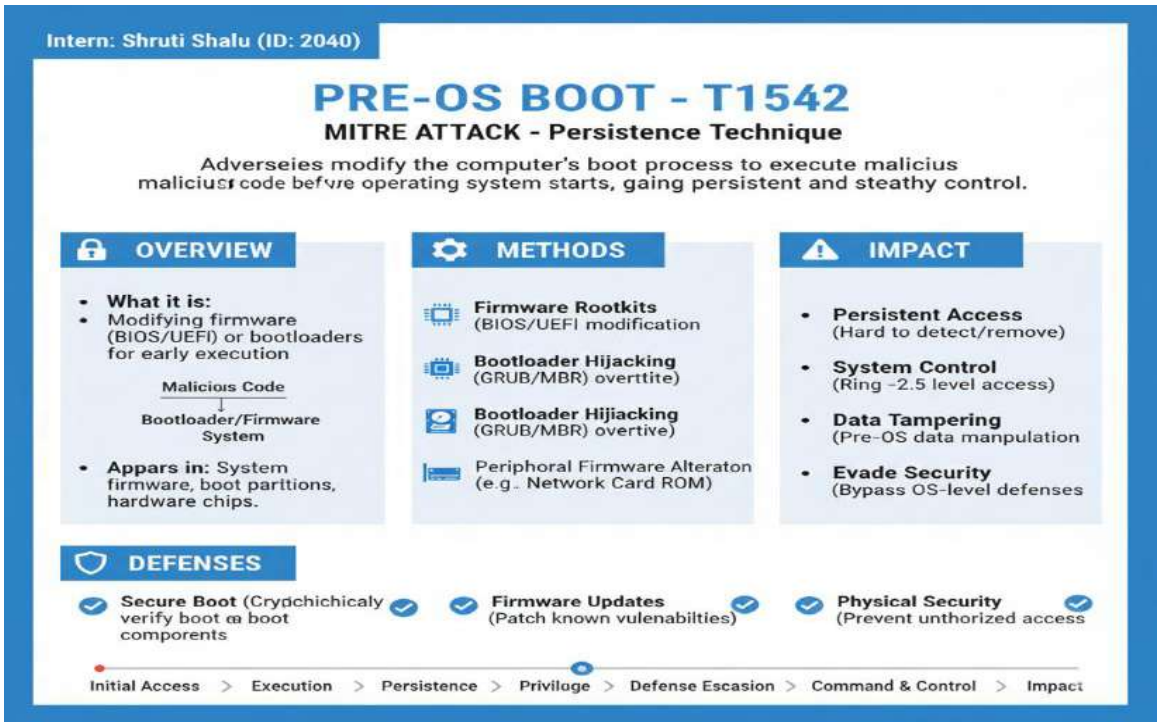


Attack Flow / Techniques

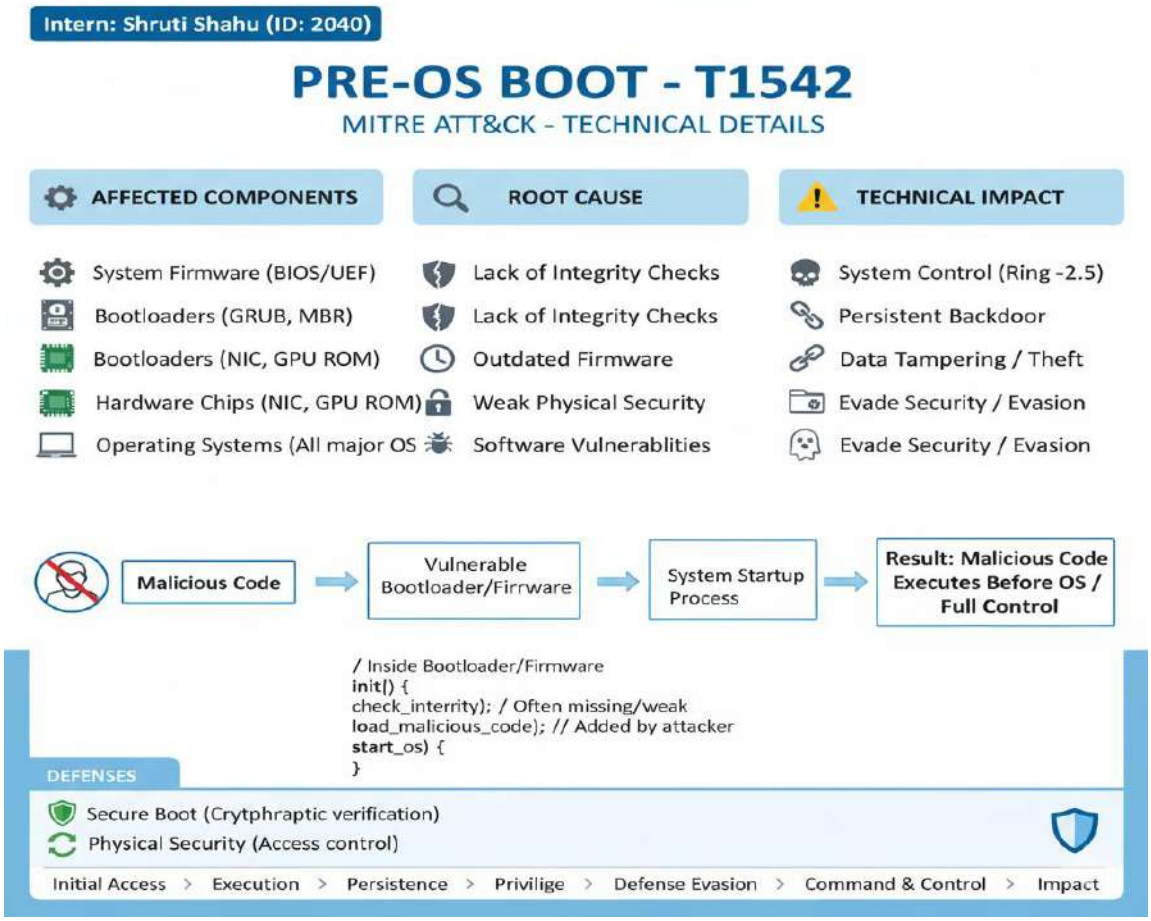


Pre-OS Boot(T1542)

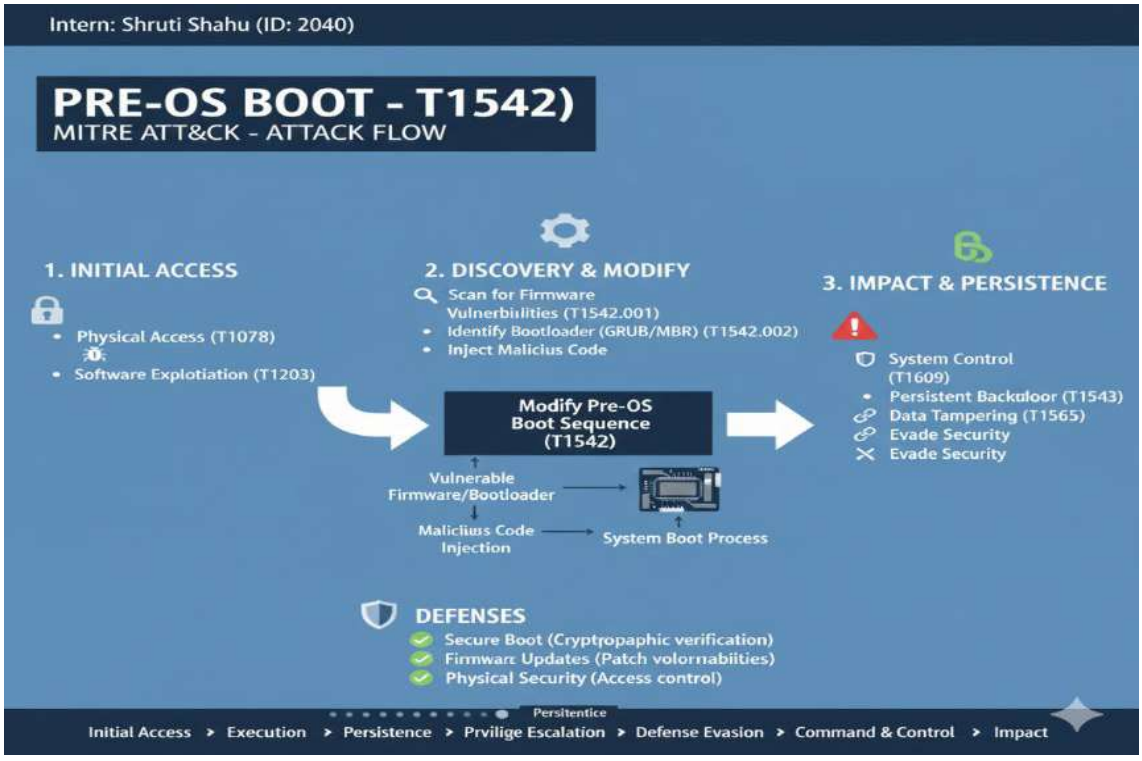
Overview



Technical Details



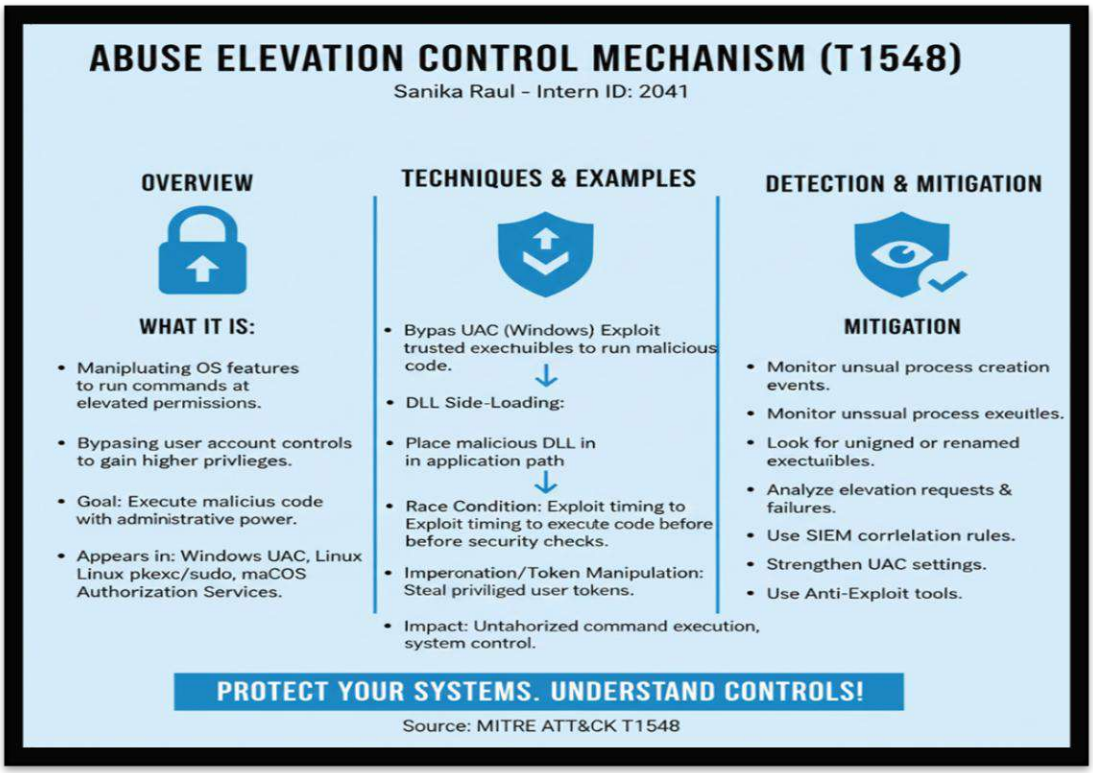
Attack Flow / Techniques



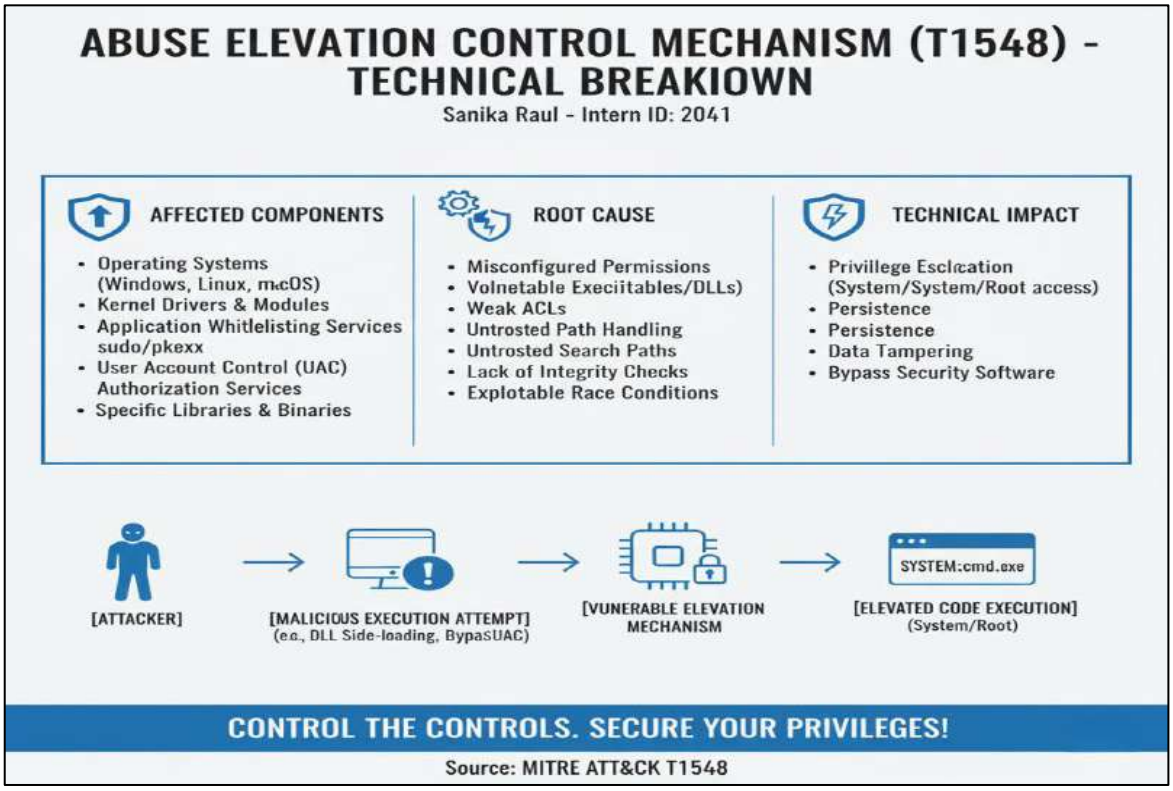
Privilege Escalation

Abuse Elevation Control Mechanism (T1548)

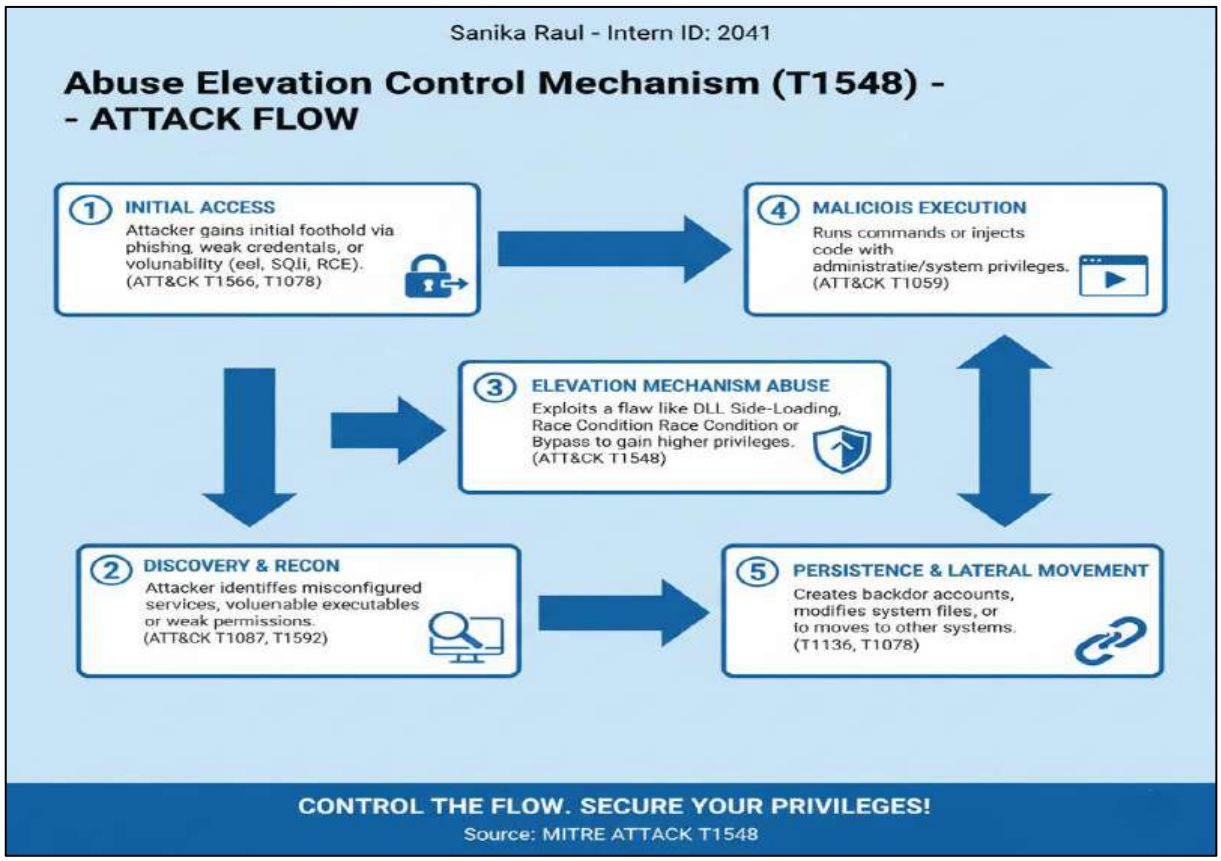
Overview



Technical details

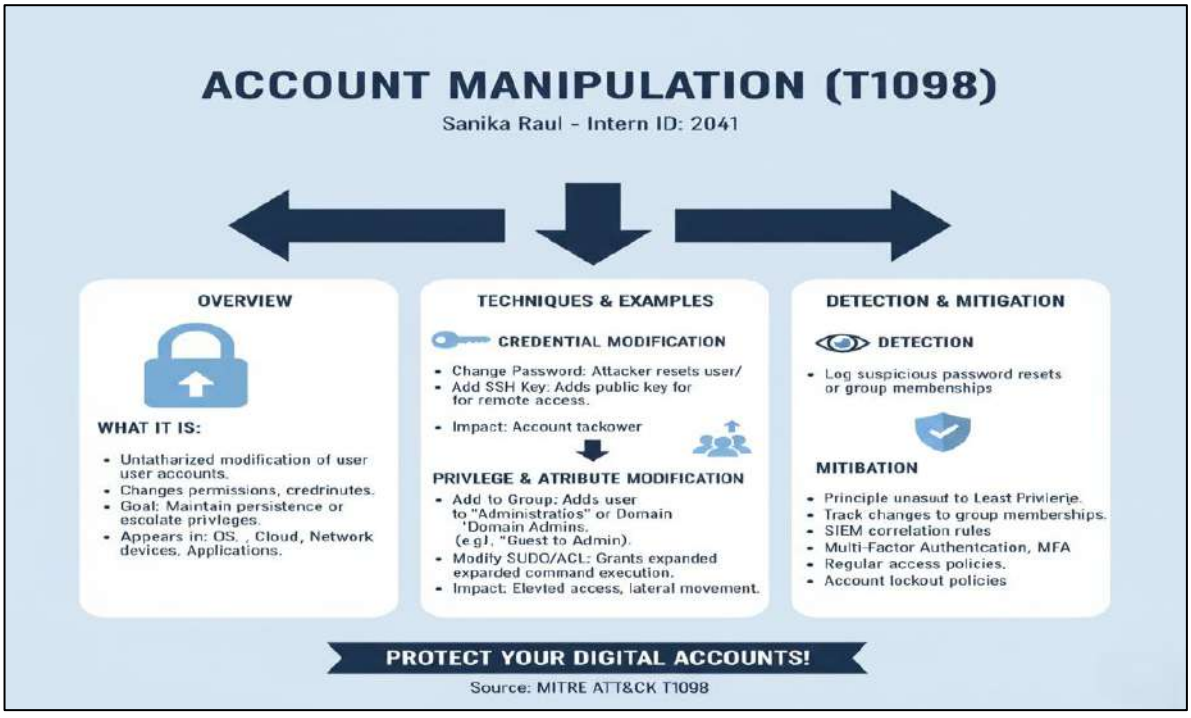


Attack flow / technique

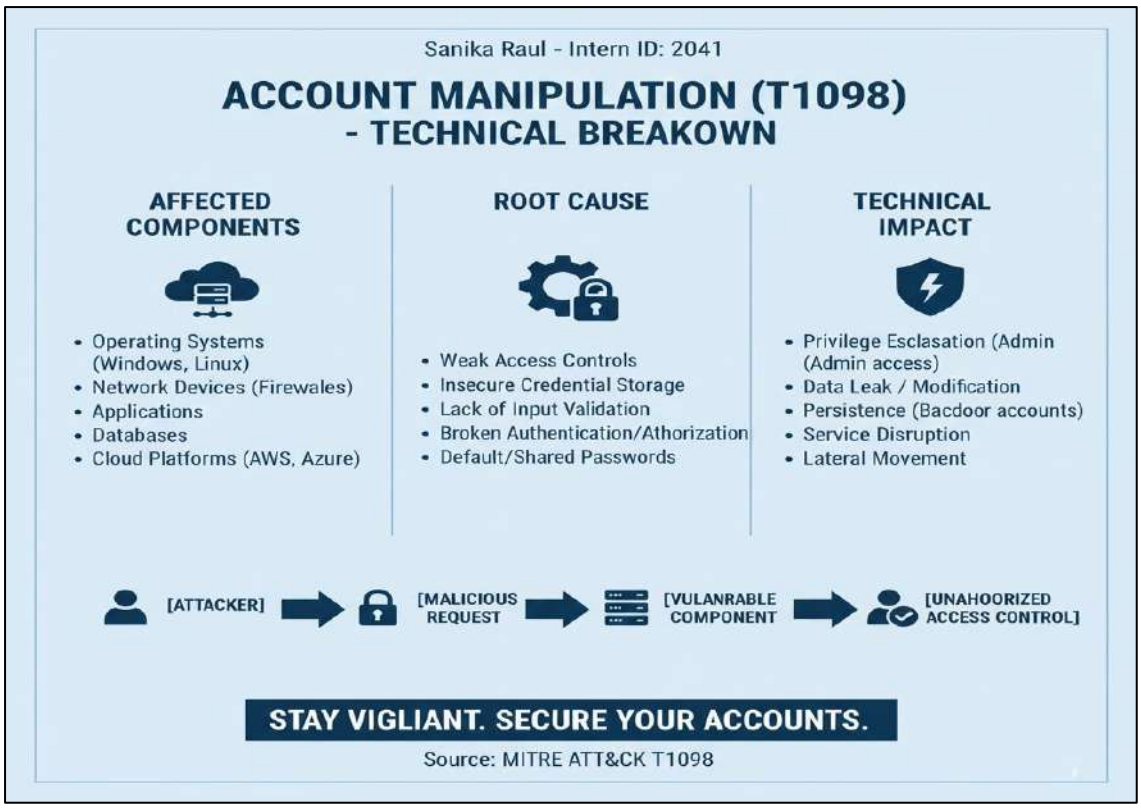


Account Manipulation (T1098)

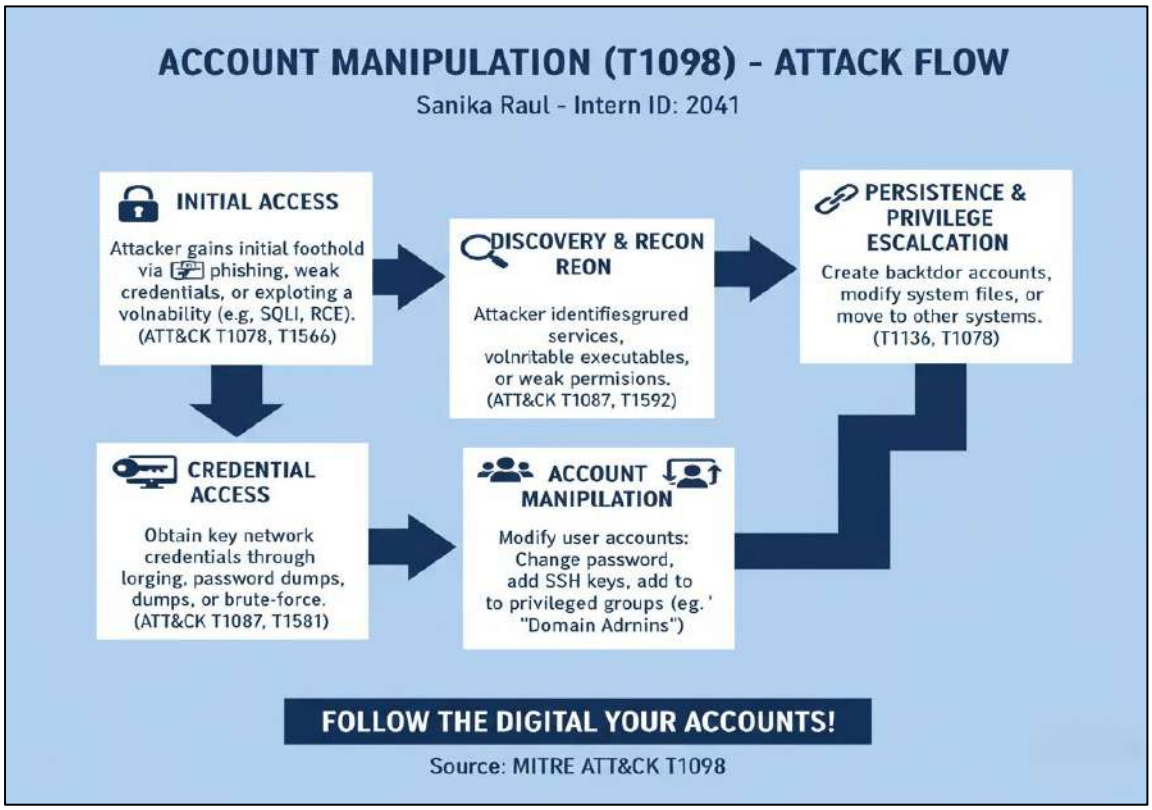
Overview



Technical details

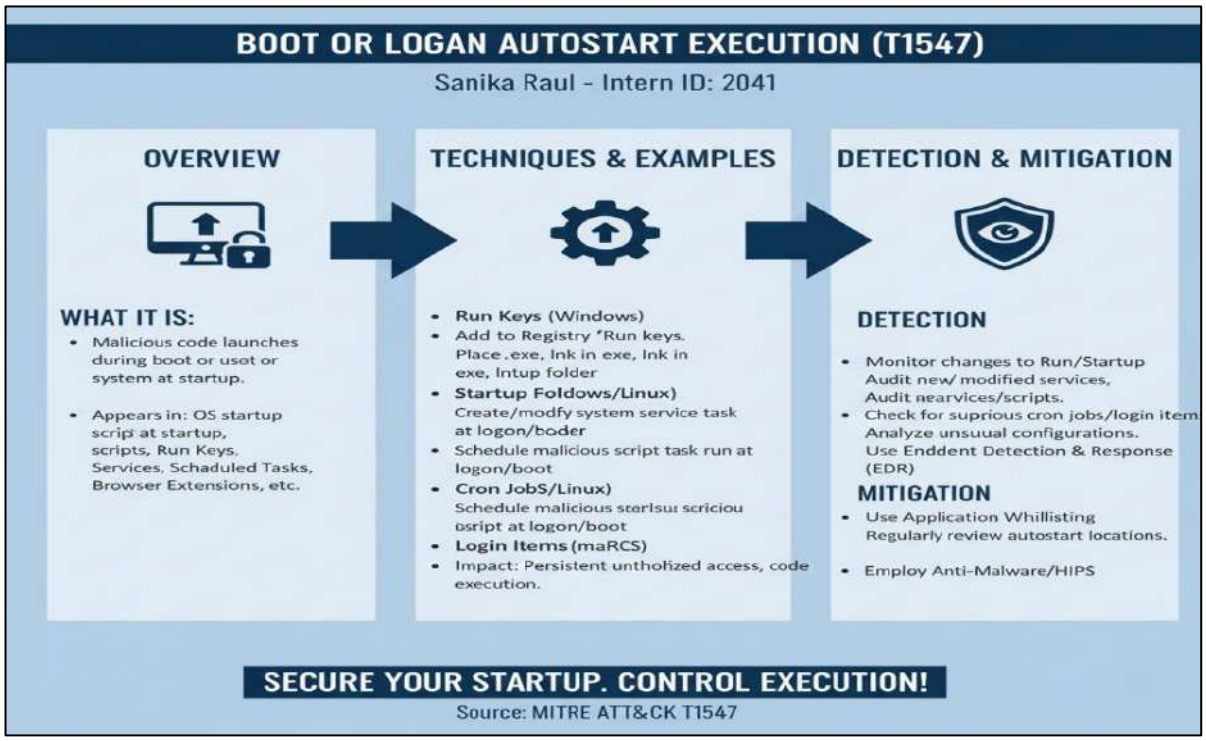


Attack flow / technique

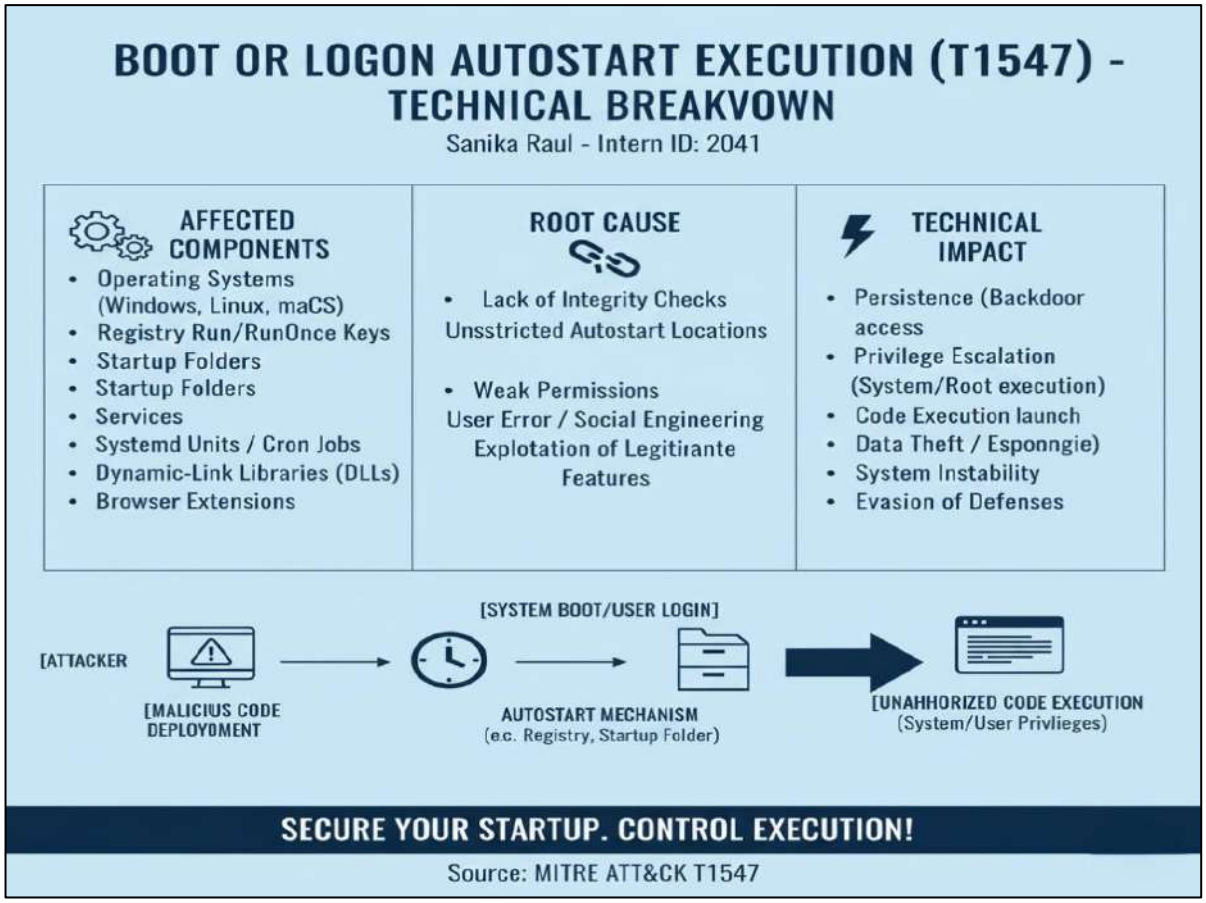


Boot or Logon Autostart Execution (T1547)

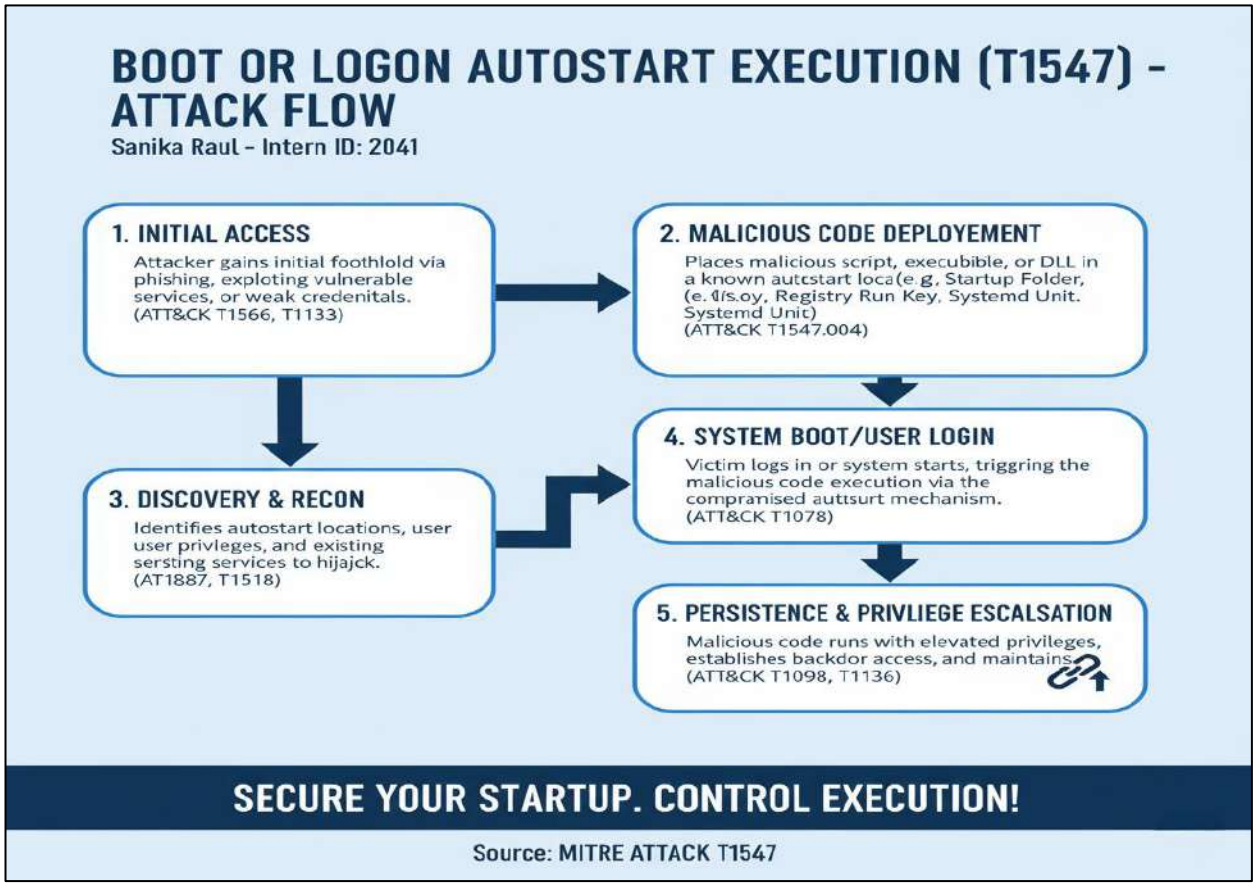
Overview



Technical details

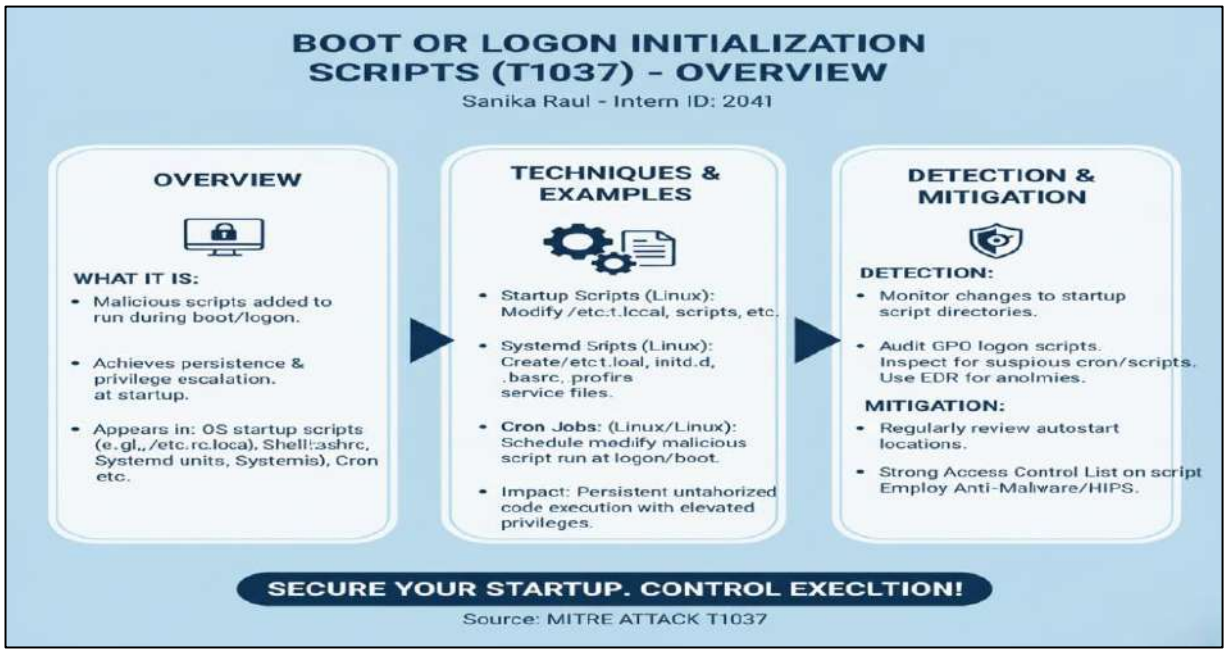


Attack flow / technique

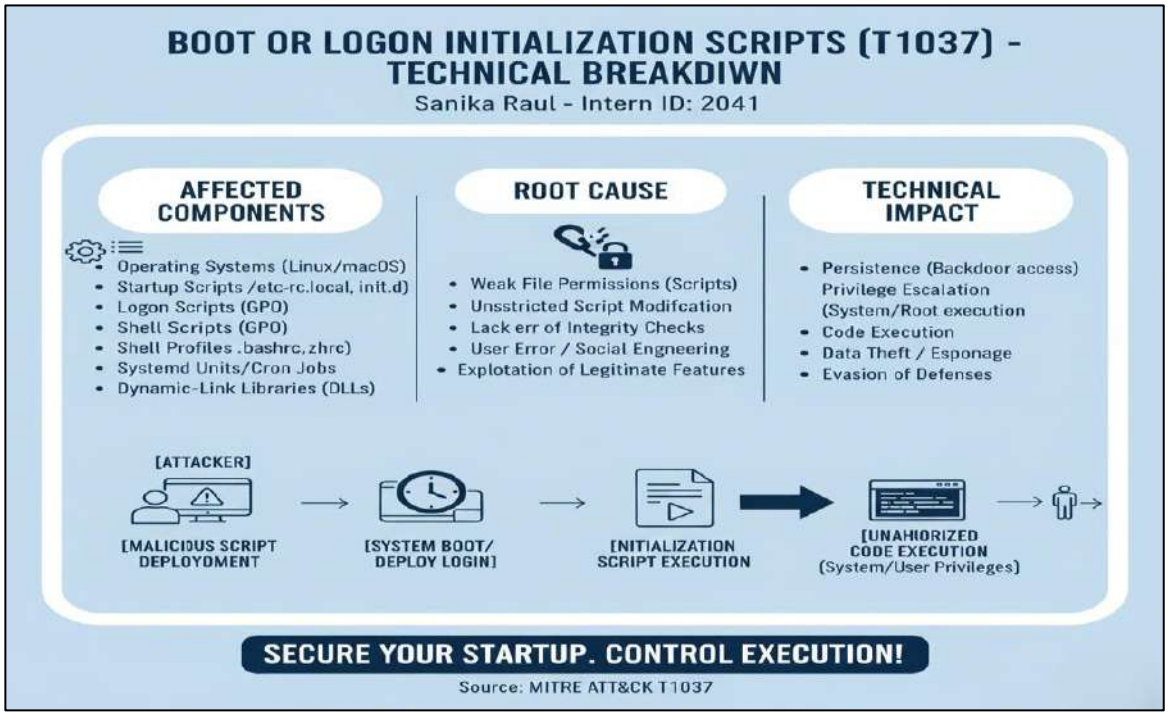


Boot or Logon Initialization Scripts (T1037)

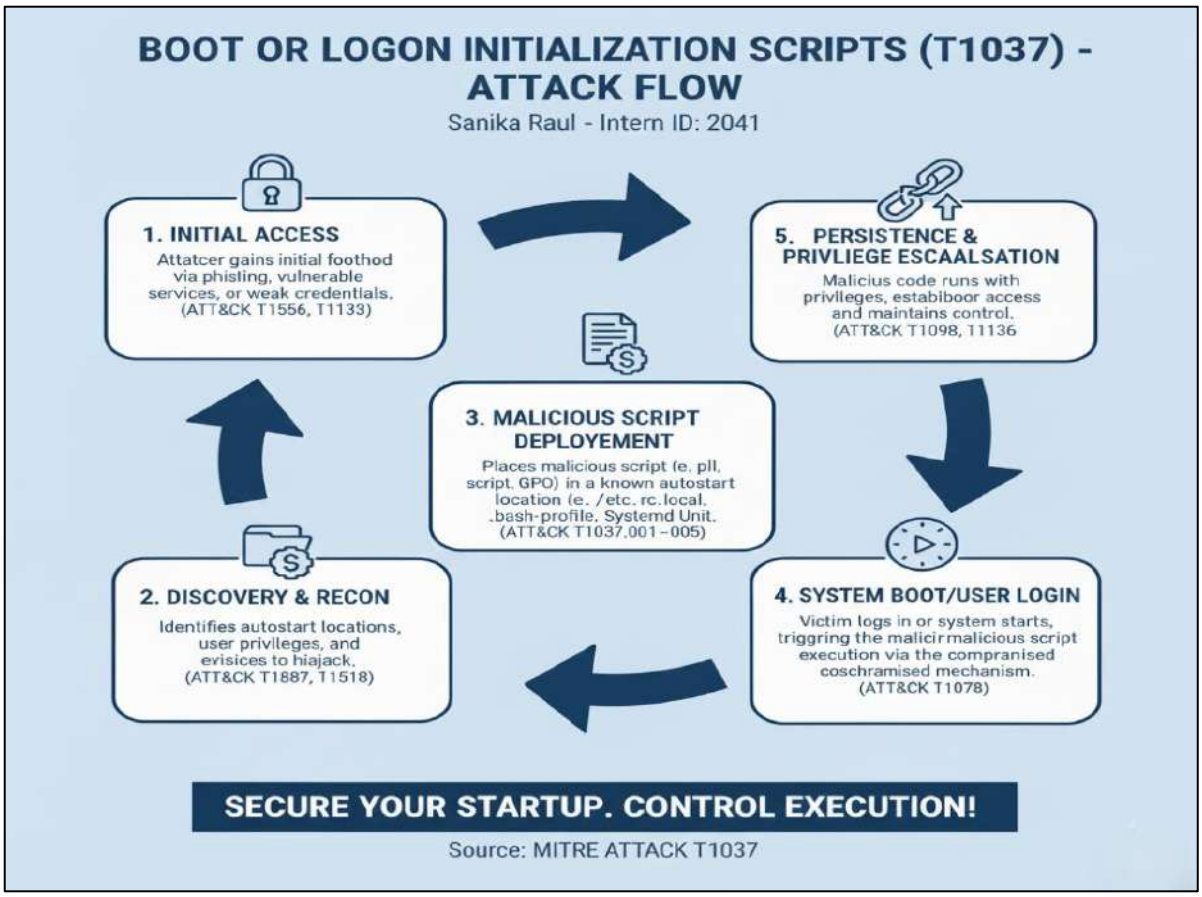
Overview



Technical details

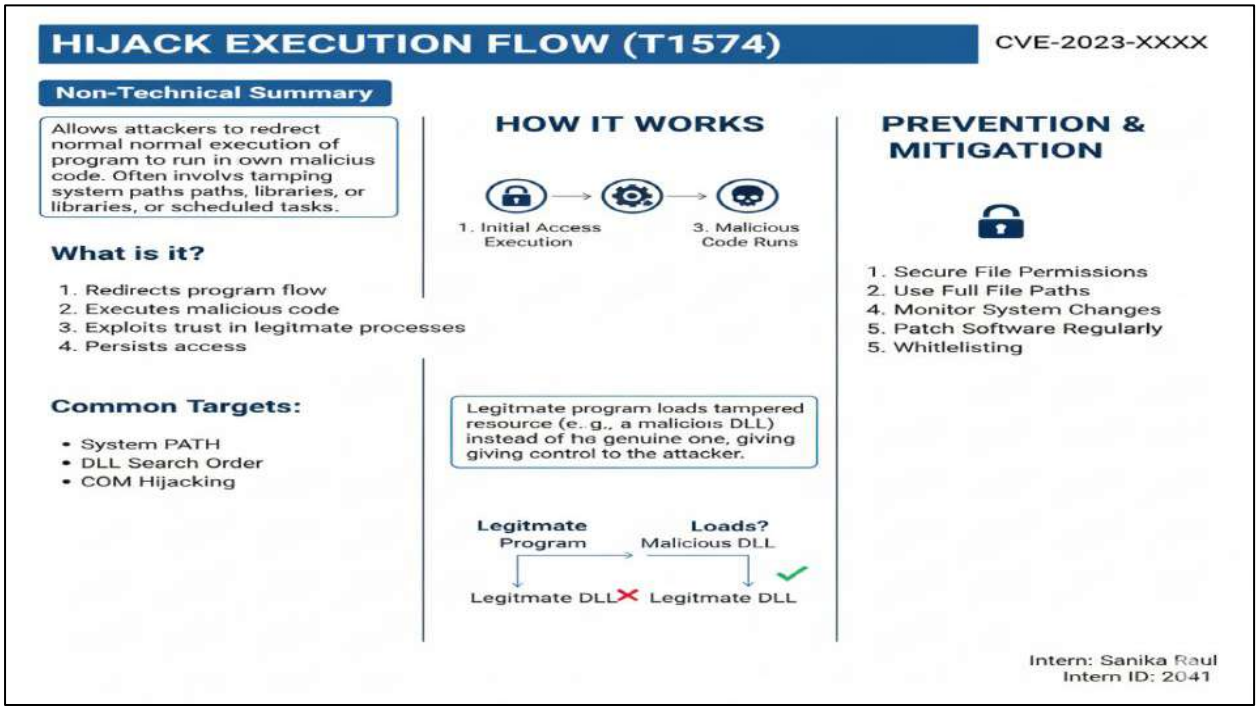


Attack flow / technique

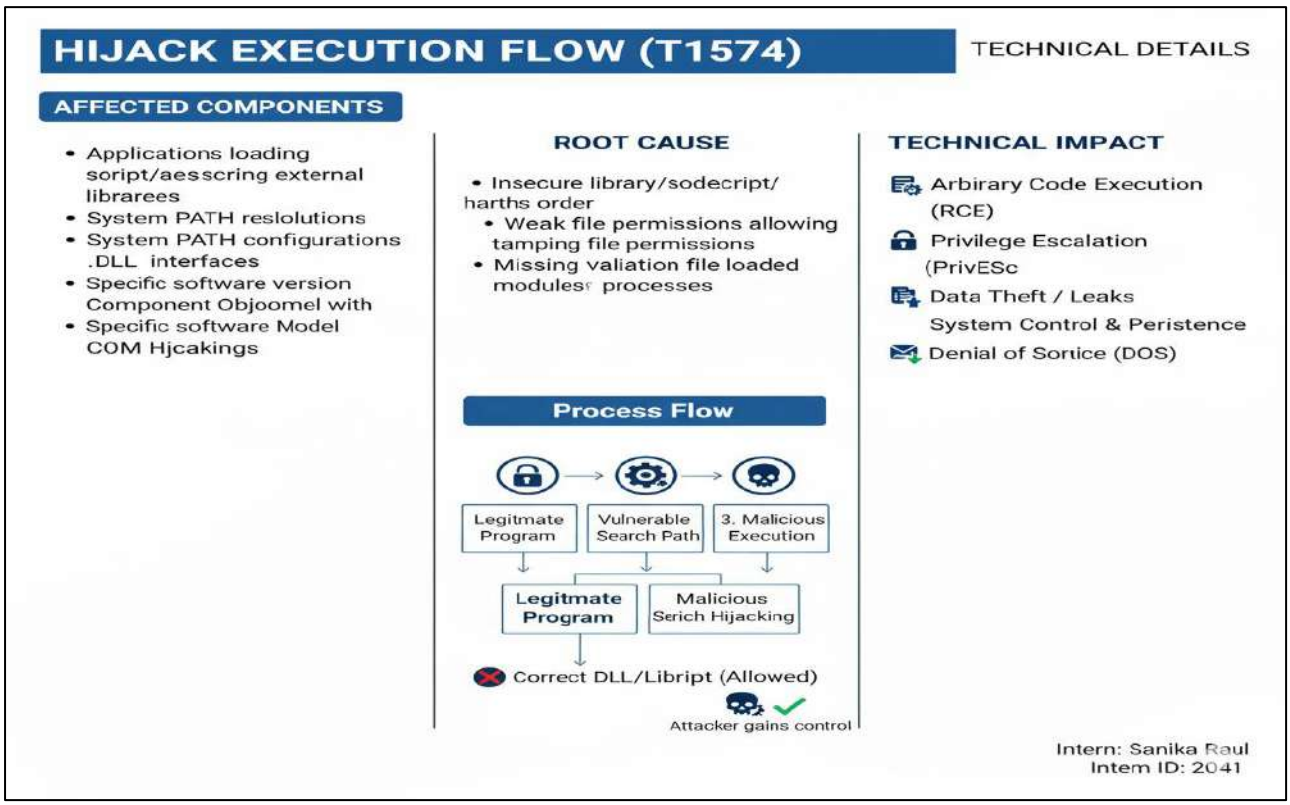


Hijack Execution Flow (T1574)

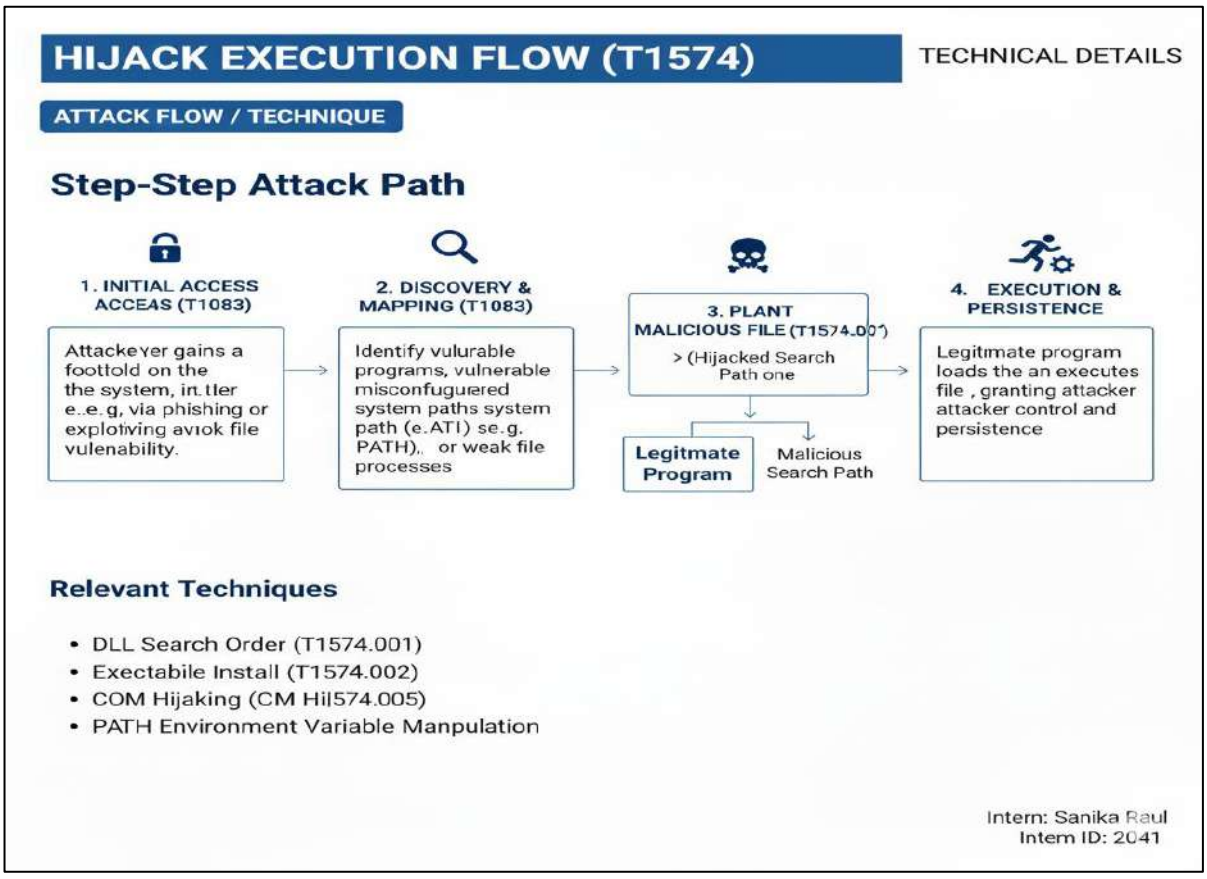
Overview



Technical details

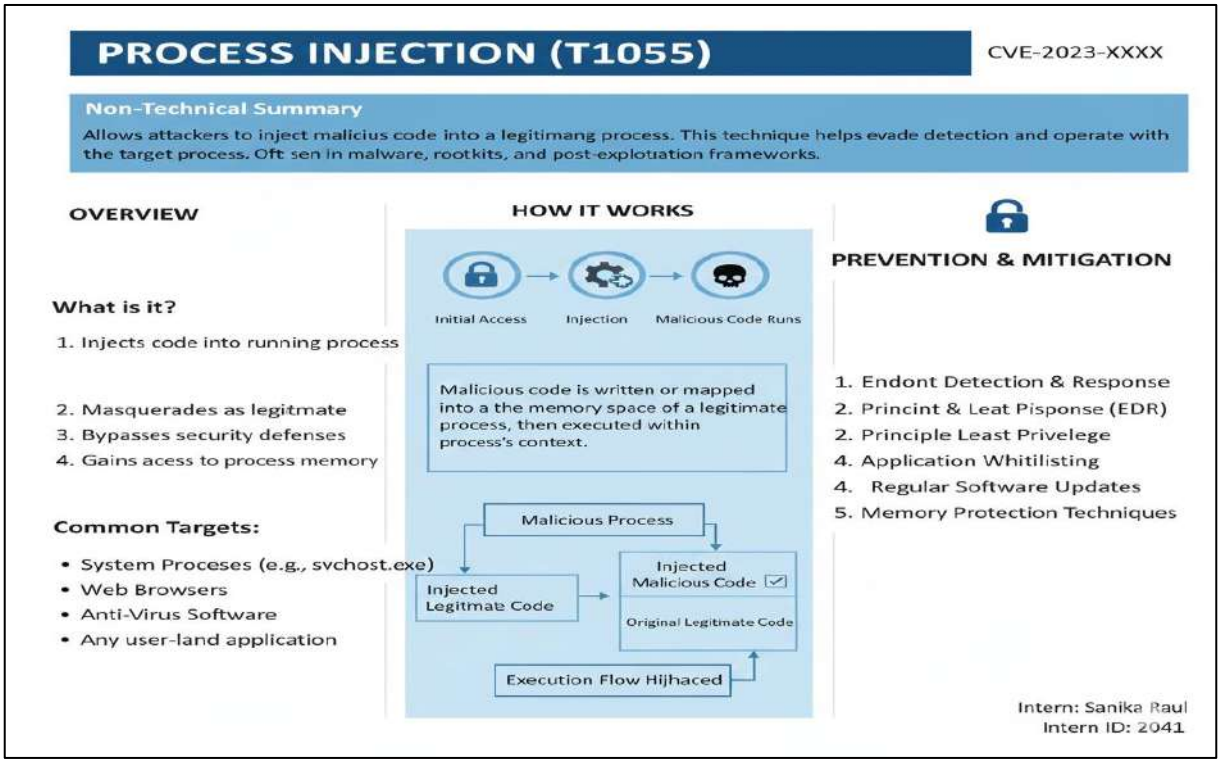


Attack flow / technique

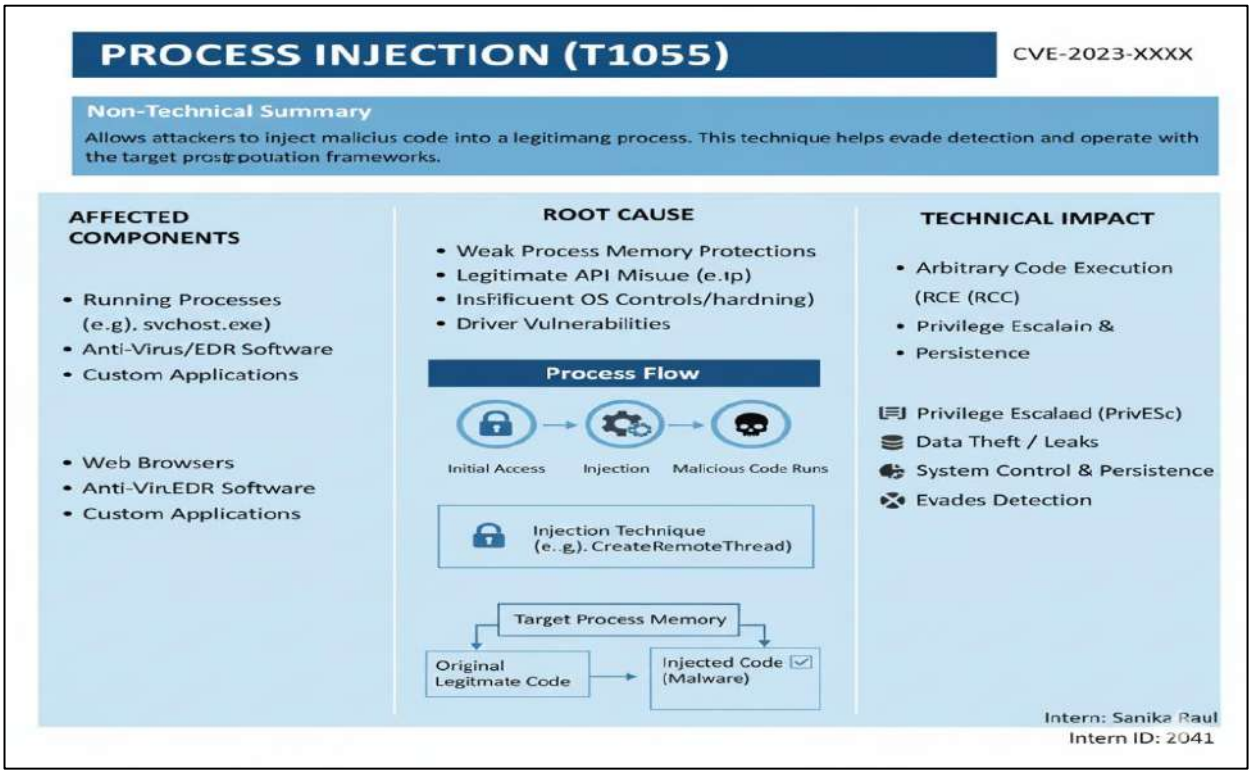


Process Injection (T1055)

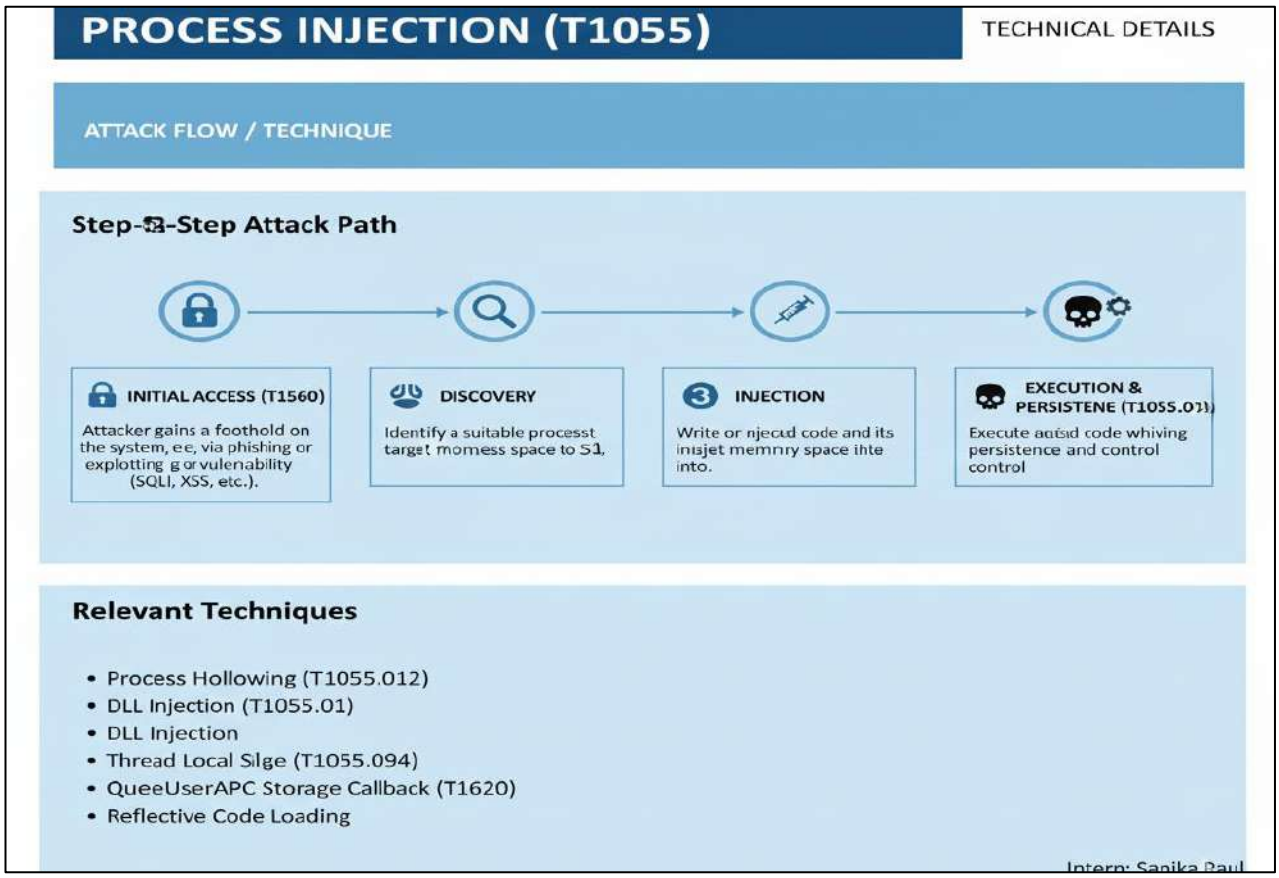
Overview



Technical details

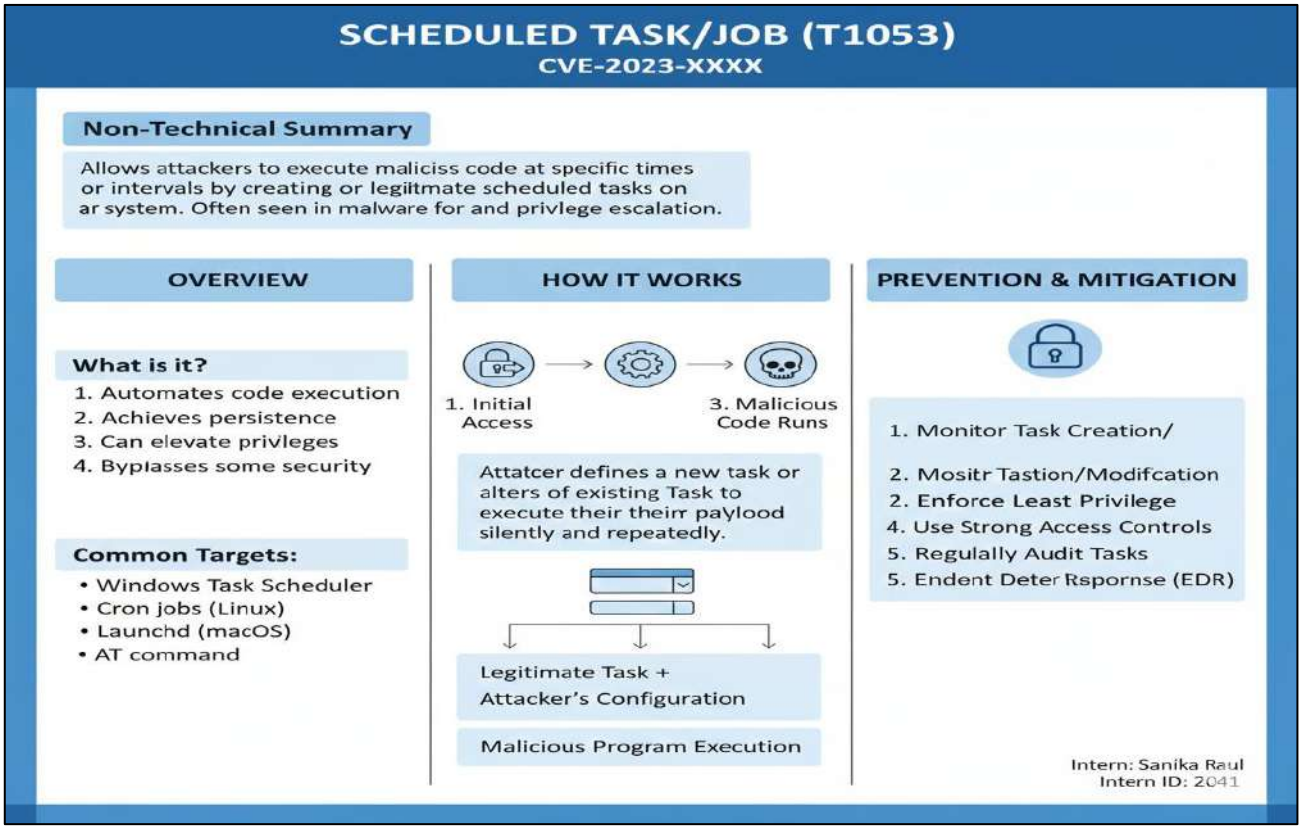


Attack flow / technique

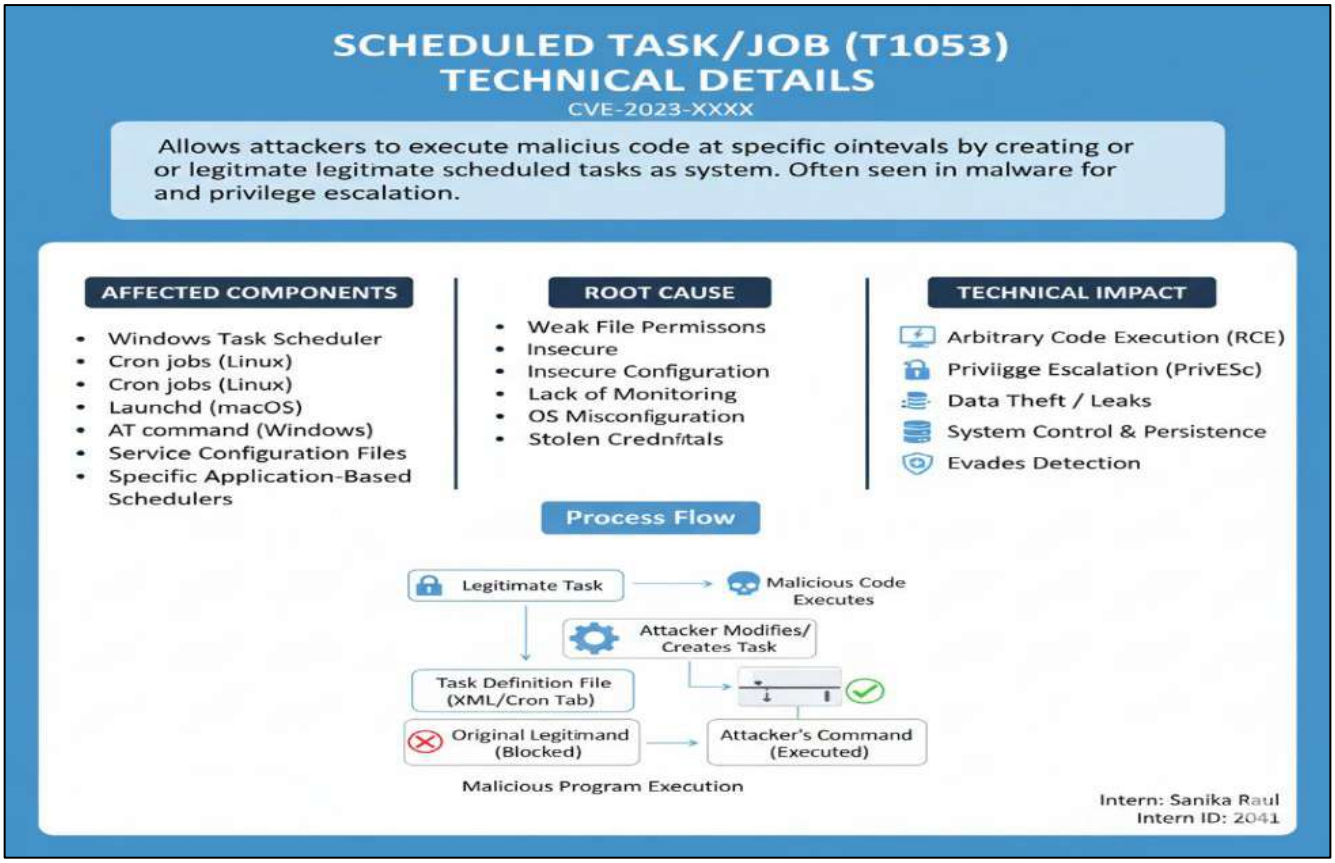


Scheduled Task/Job (T1053)

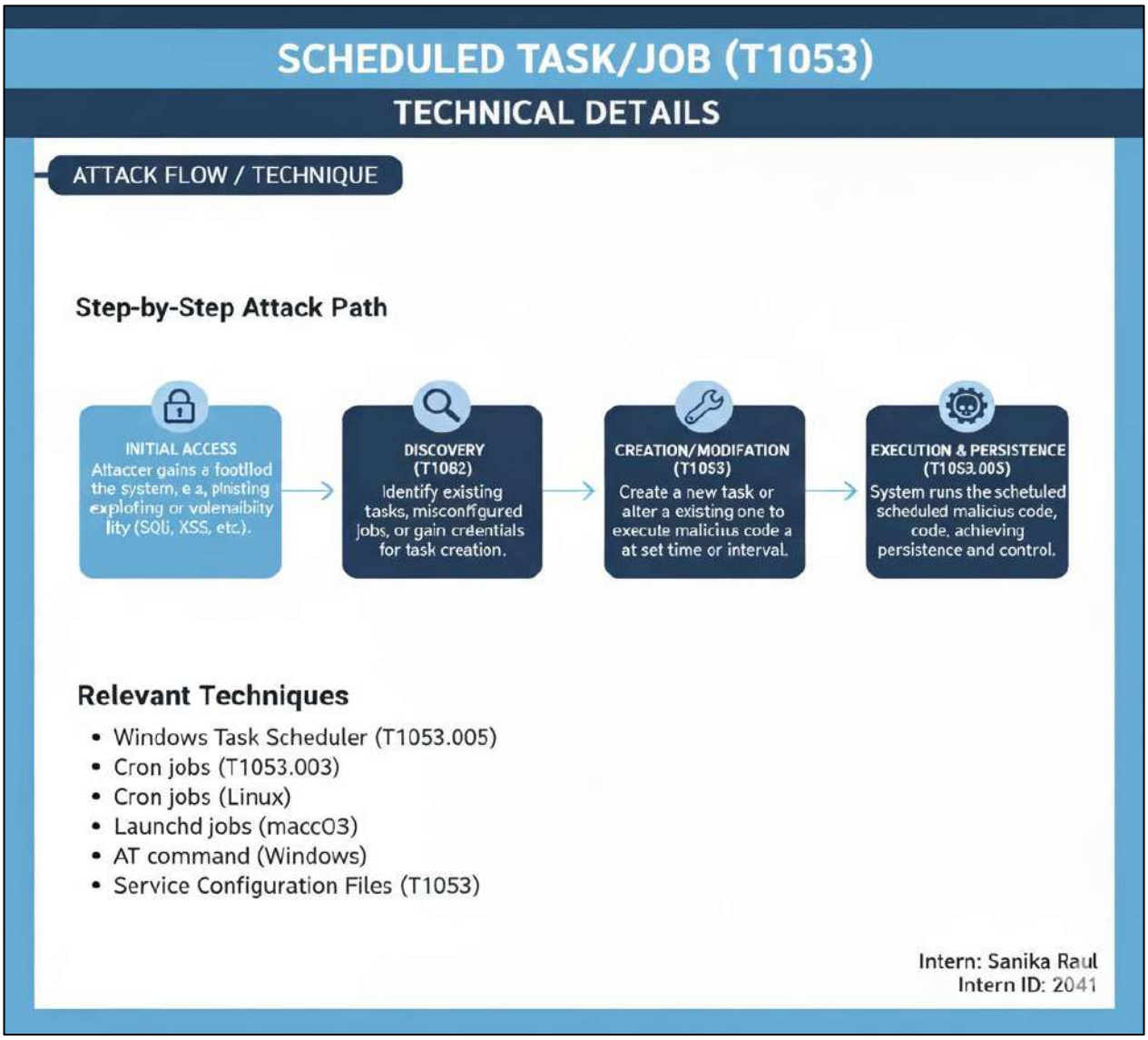
Overview



Technical details

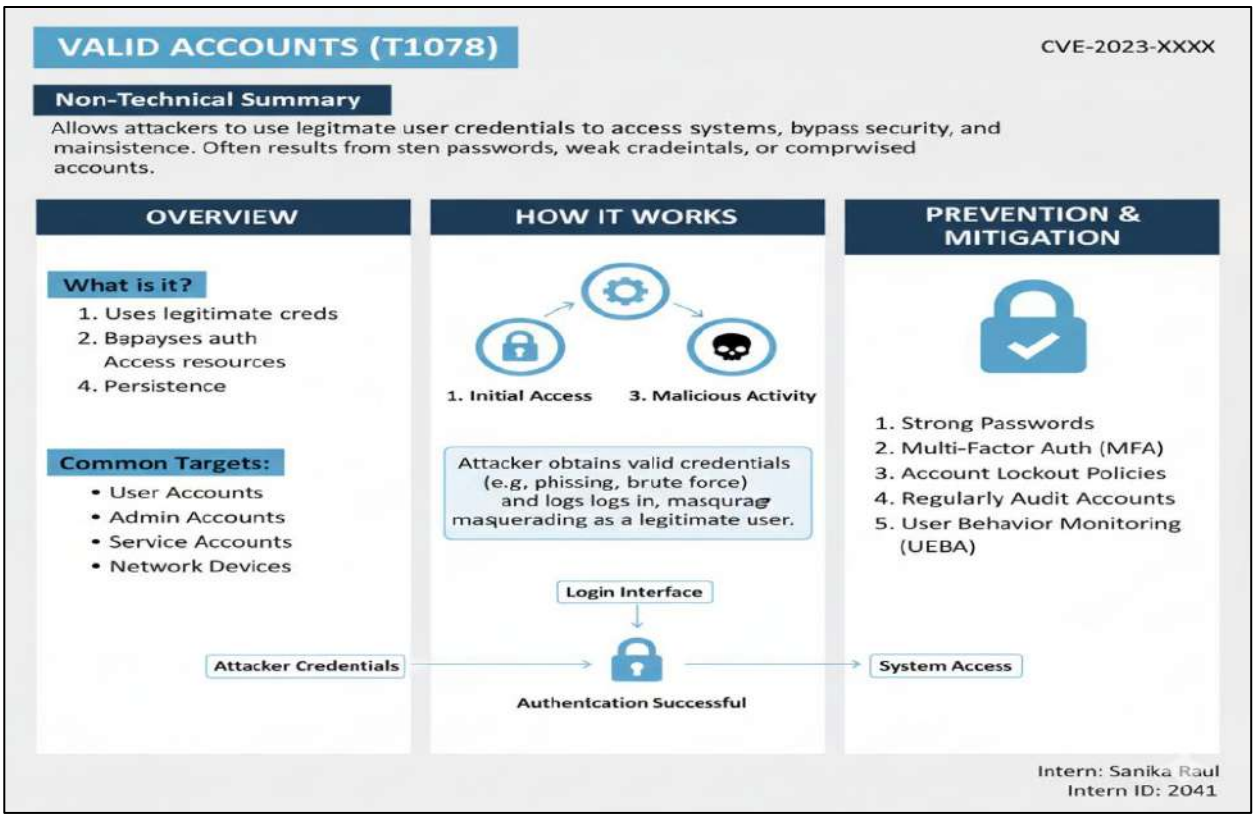


Attack flow / technique

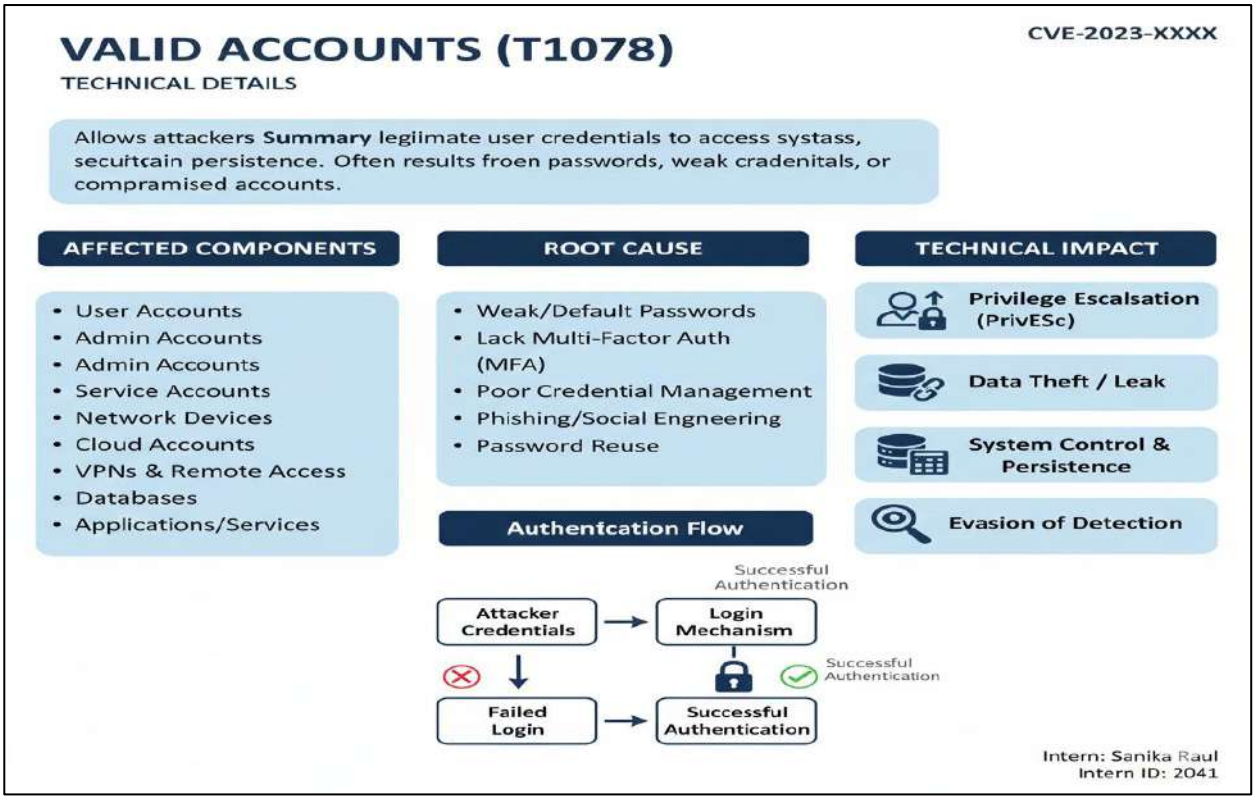


Valid Accounts (T1078)

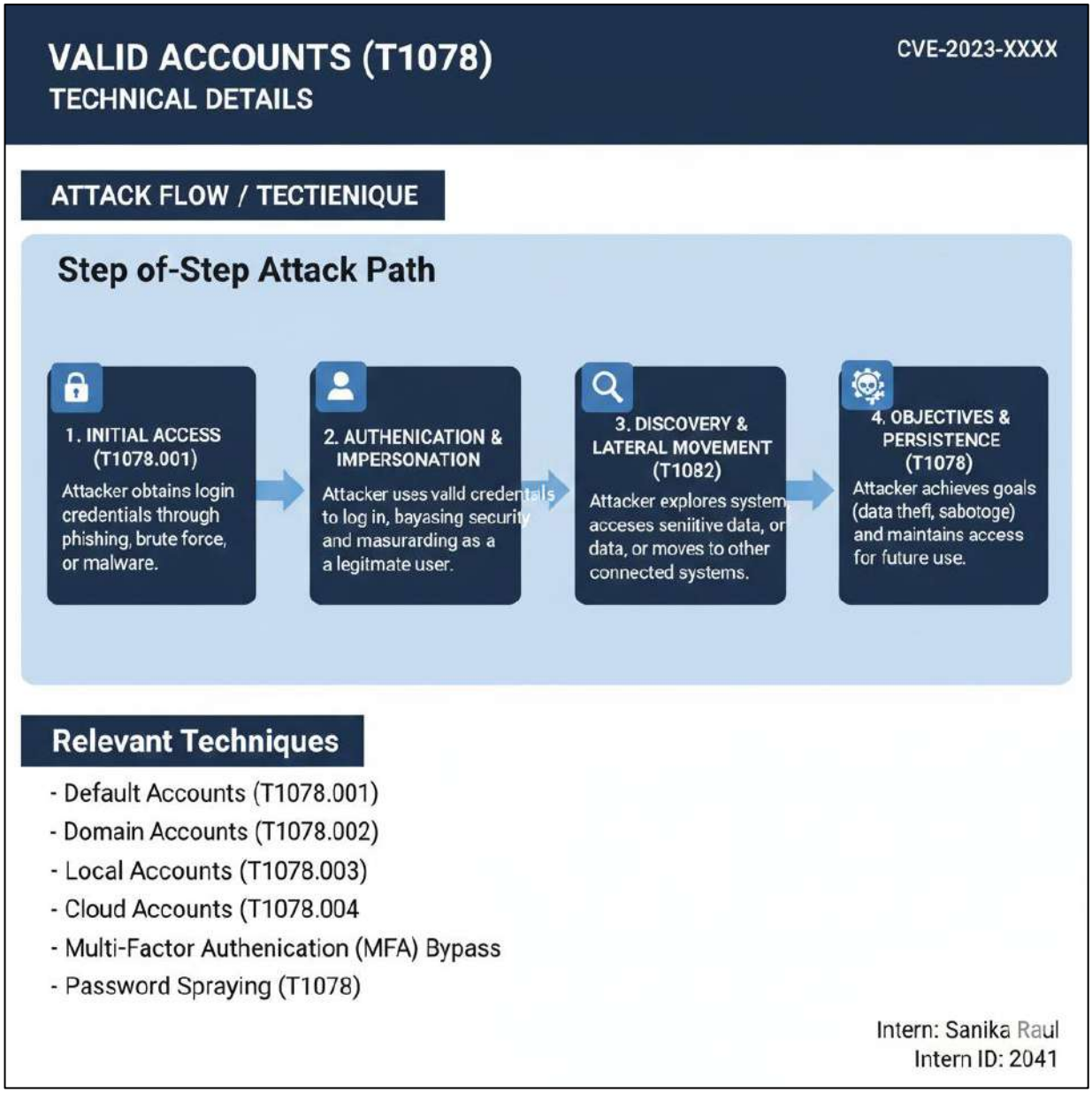
Overview



Technical details



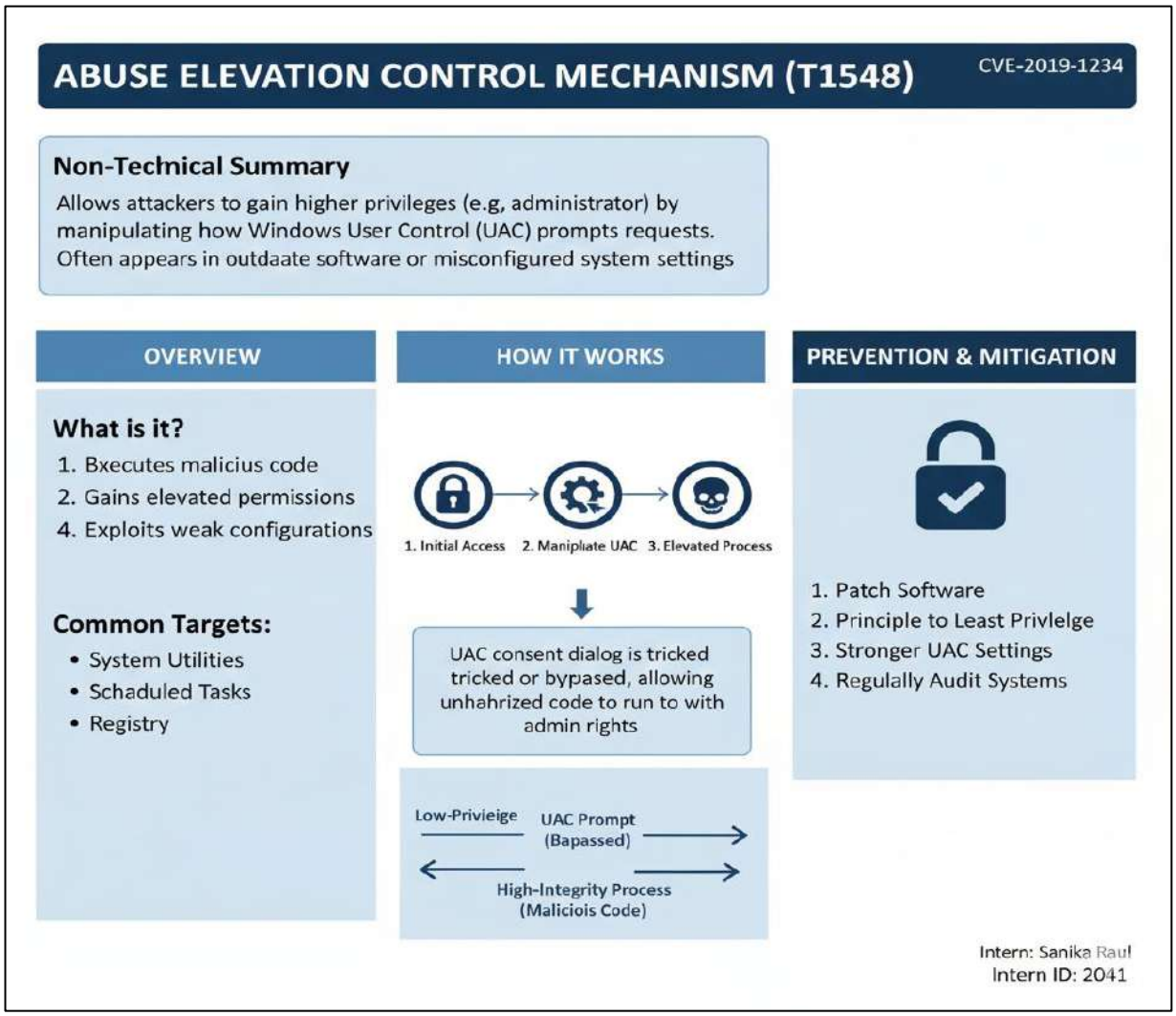
Attack flow / technique



Defense Evasion

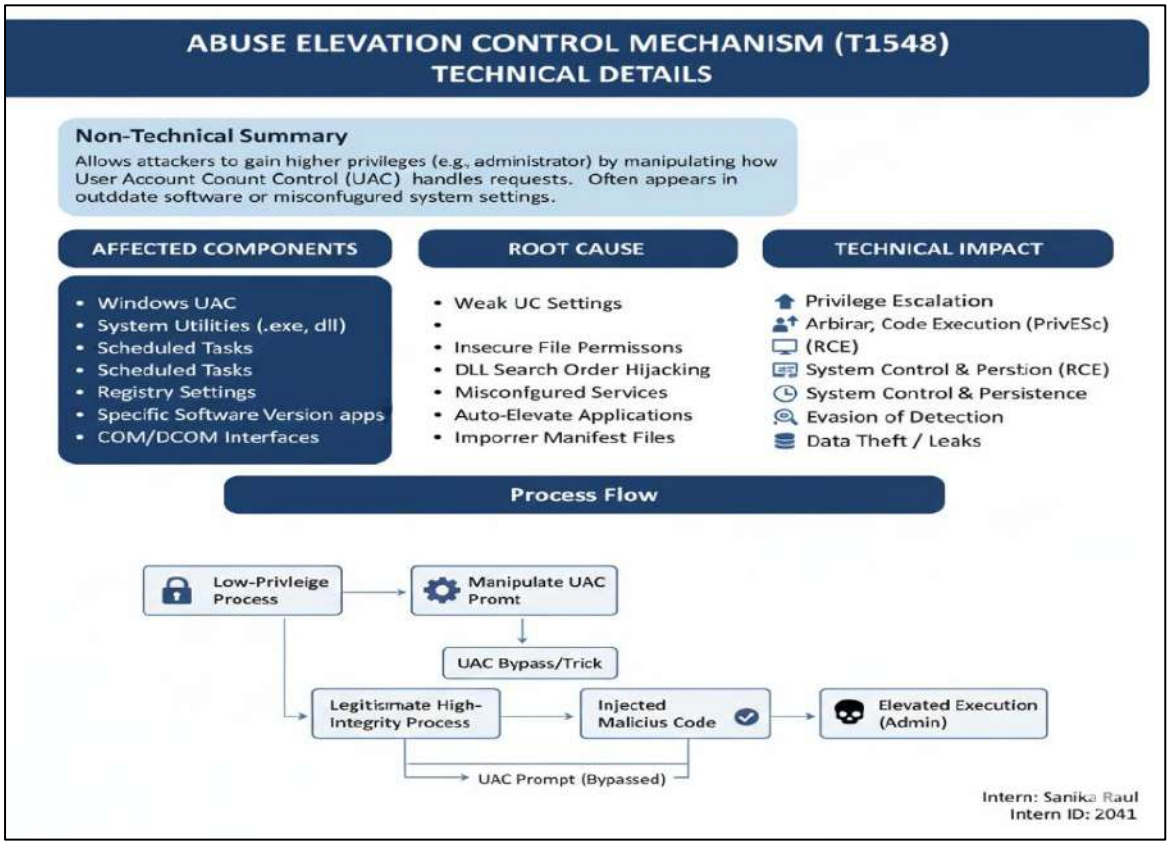
Abuse Elevation Control Mechanism (T1548)

Overview



Technical details

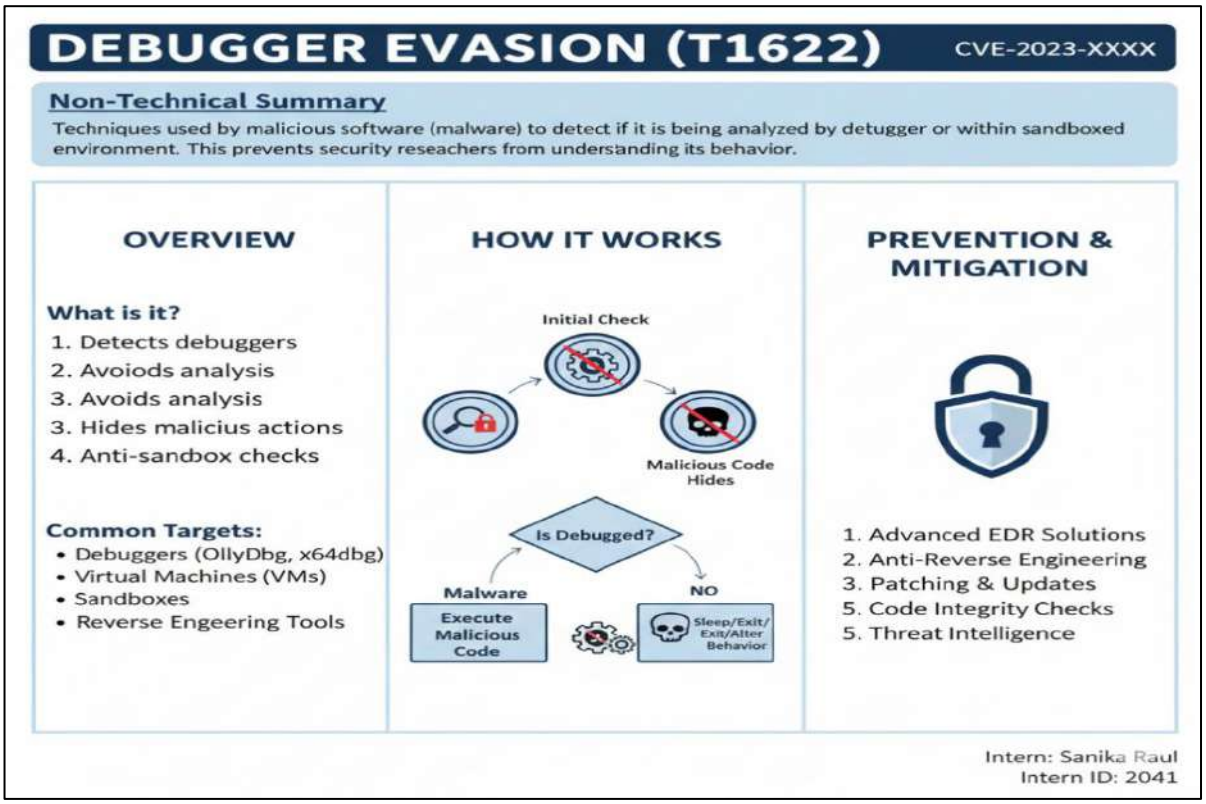
Attack flow / technique



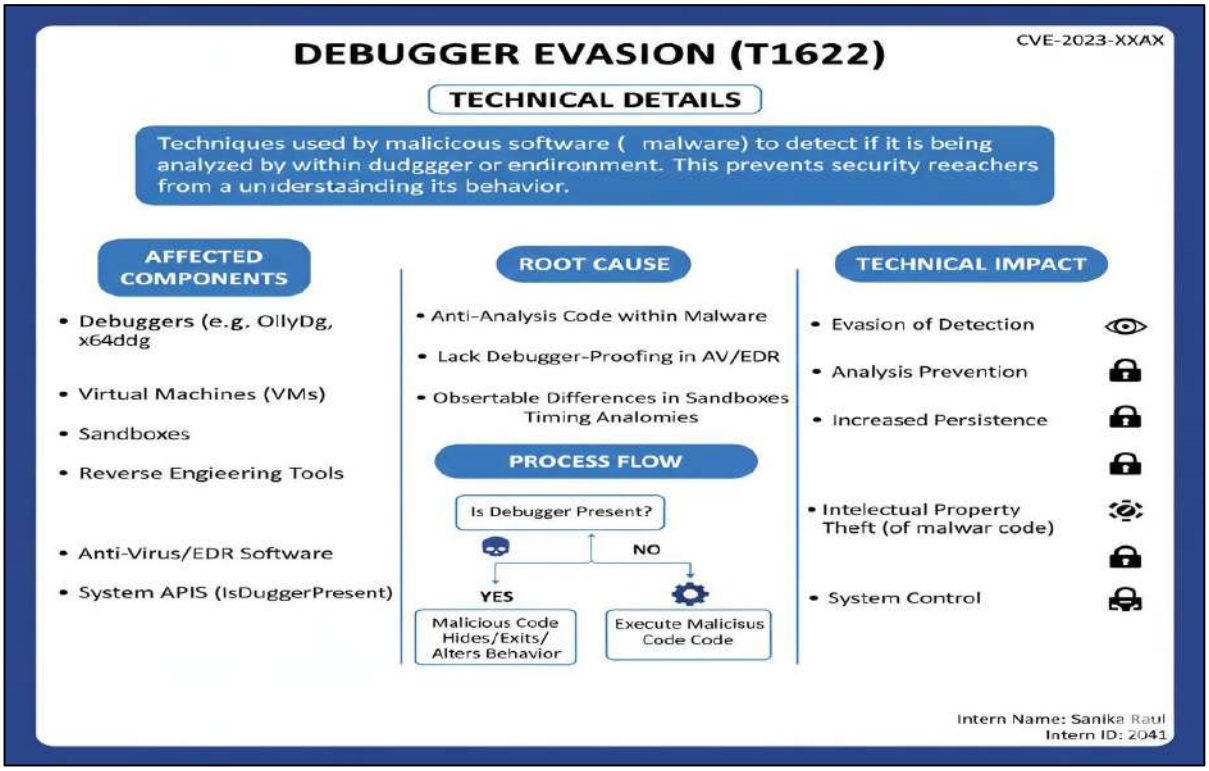
Attack flow / technique

Debugger Evasion (T1622)

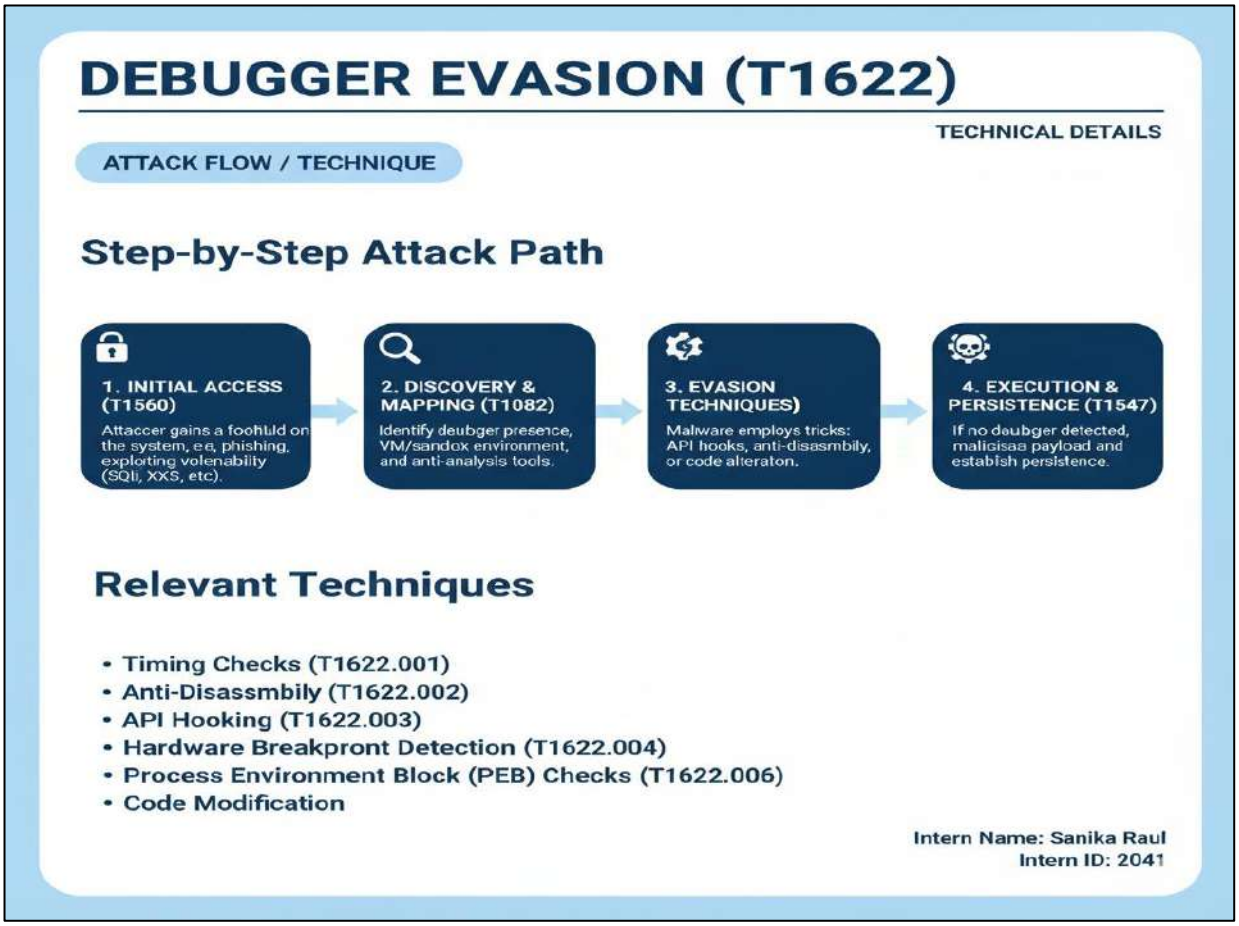
Overview



Technical details

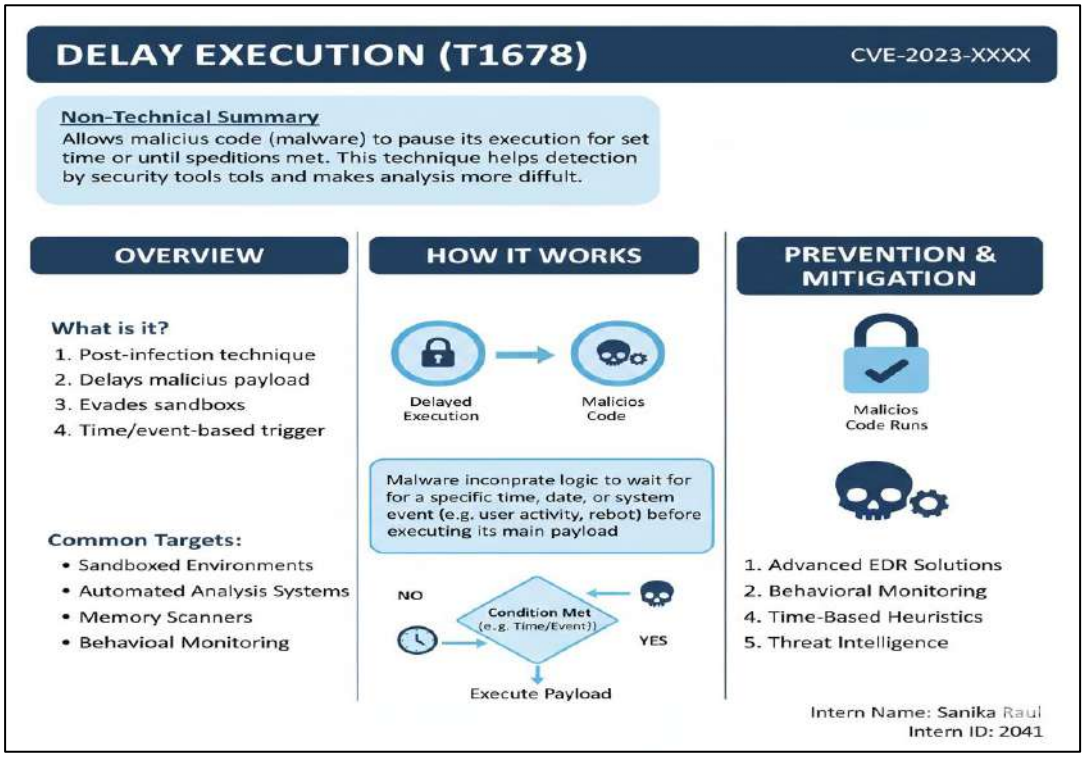


Attack flow / technique

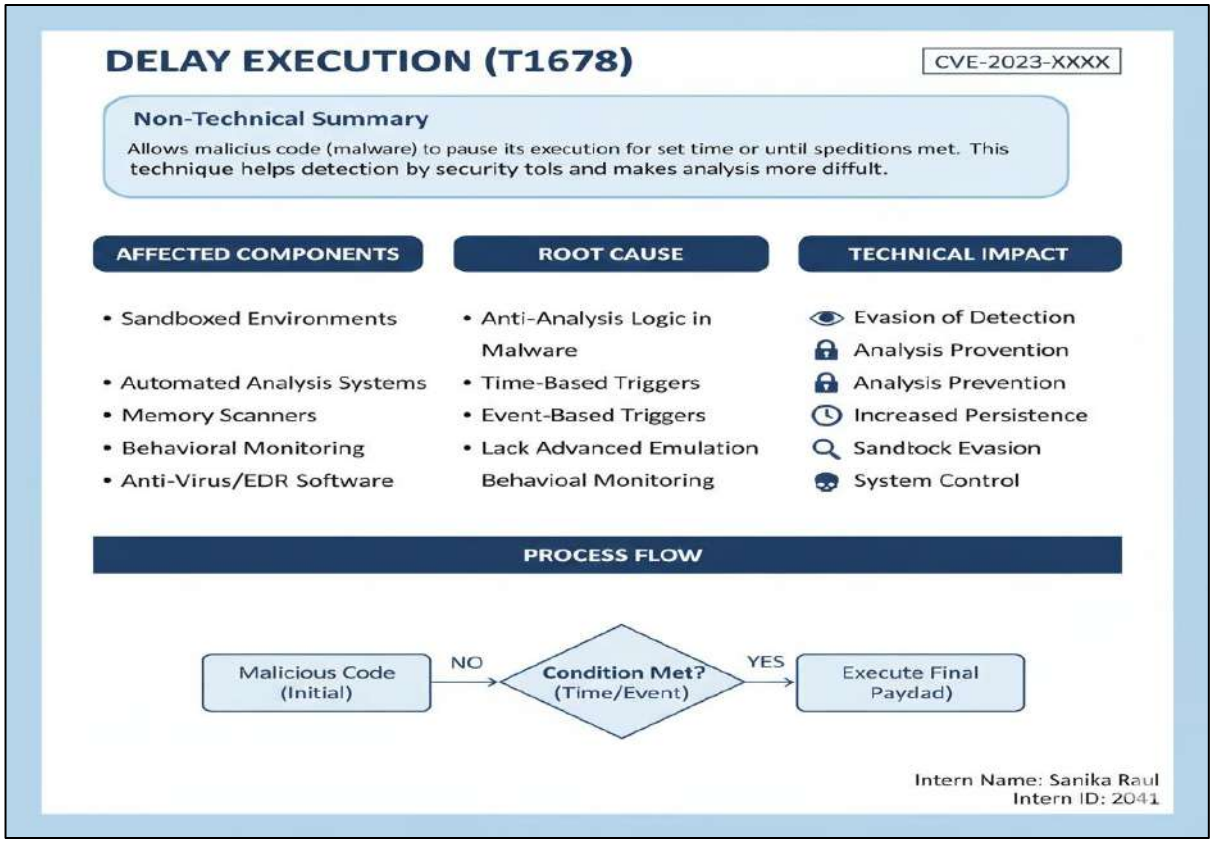


Delay Execution (T1678)

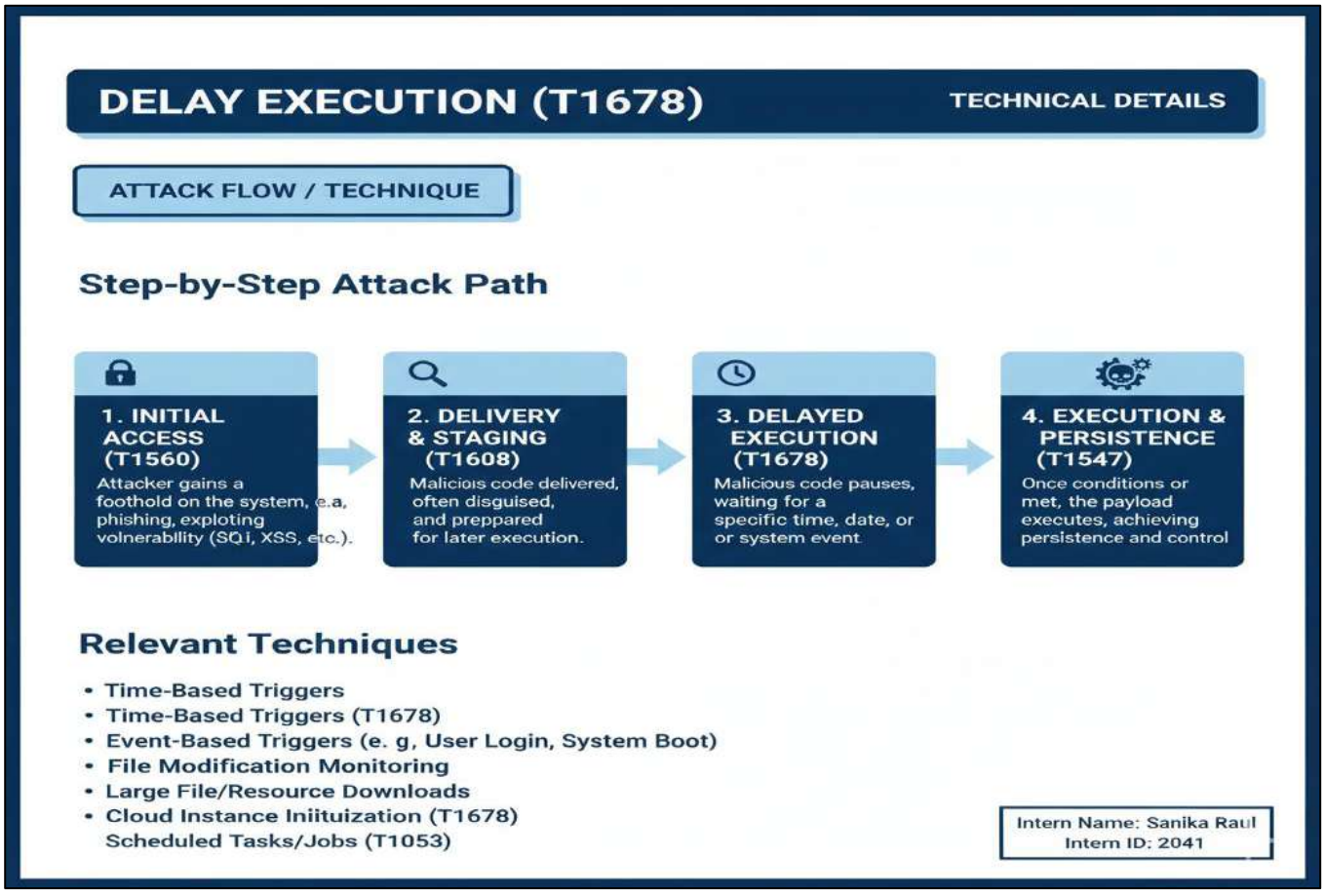
Overview



Technical details

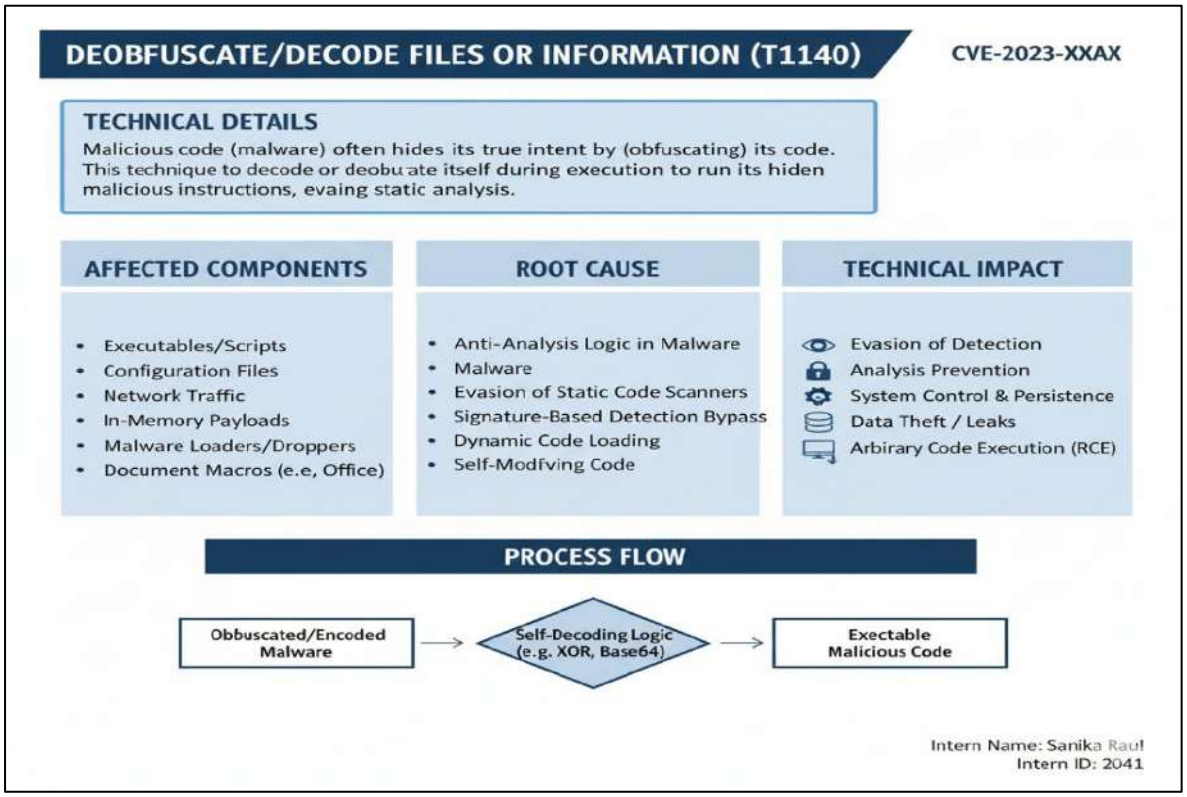


Attack flow / technique

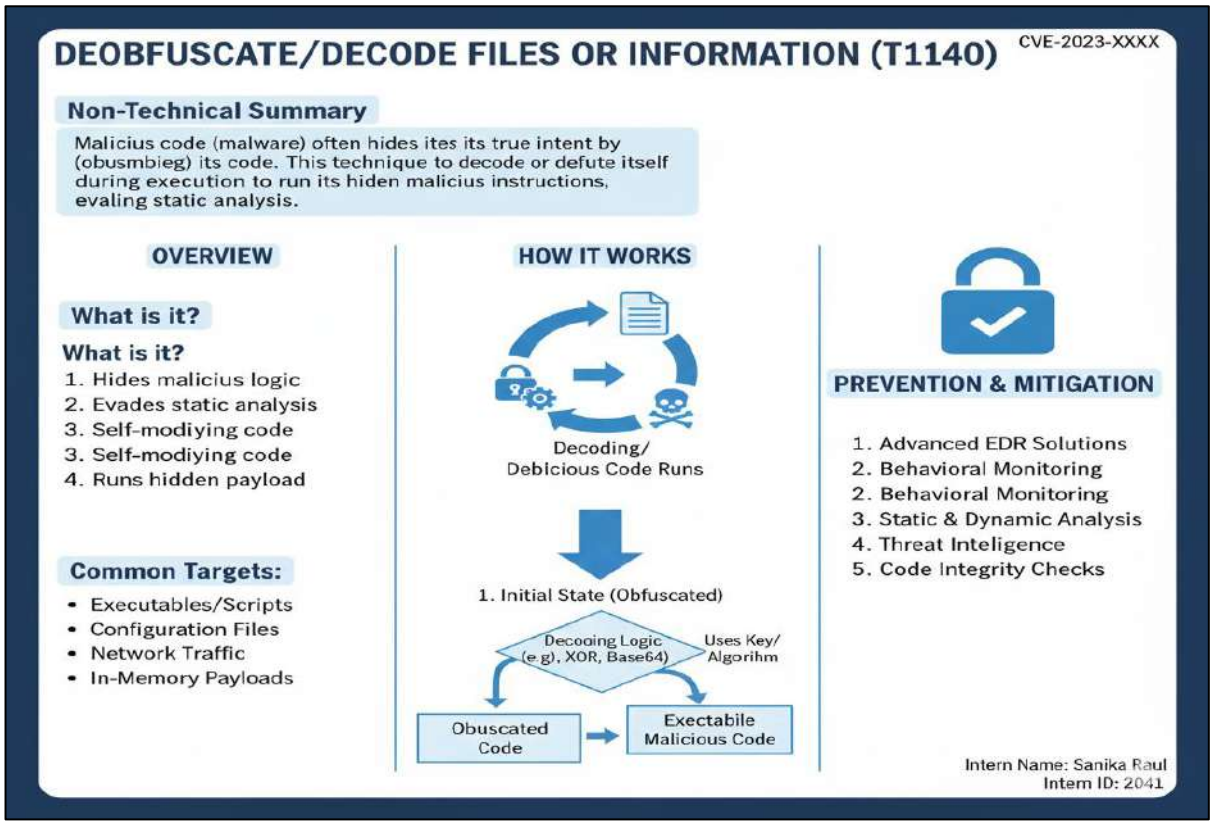


Deobfuscate/Decode Files or Information (T1140)

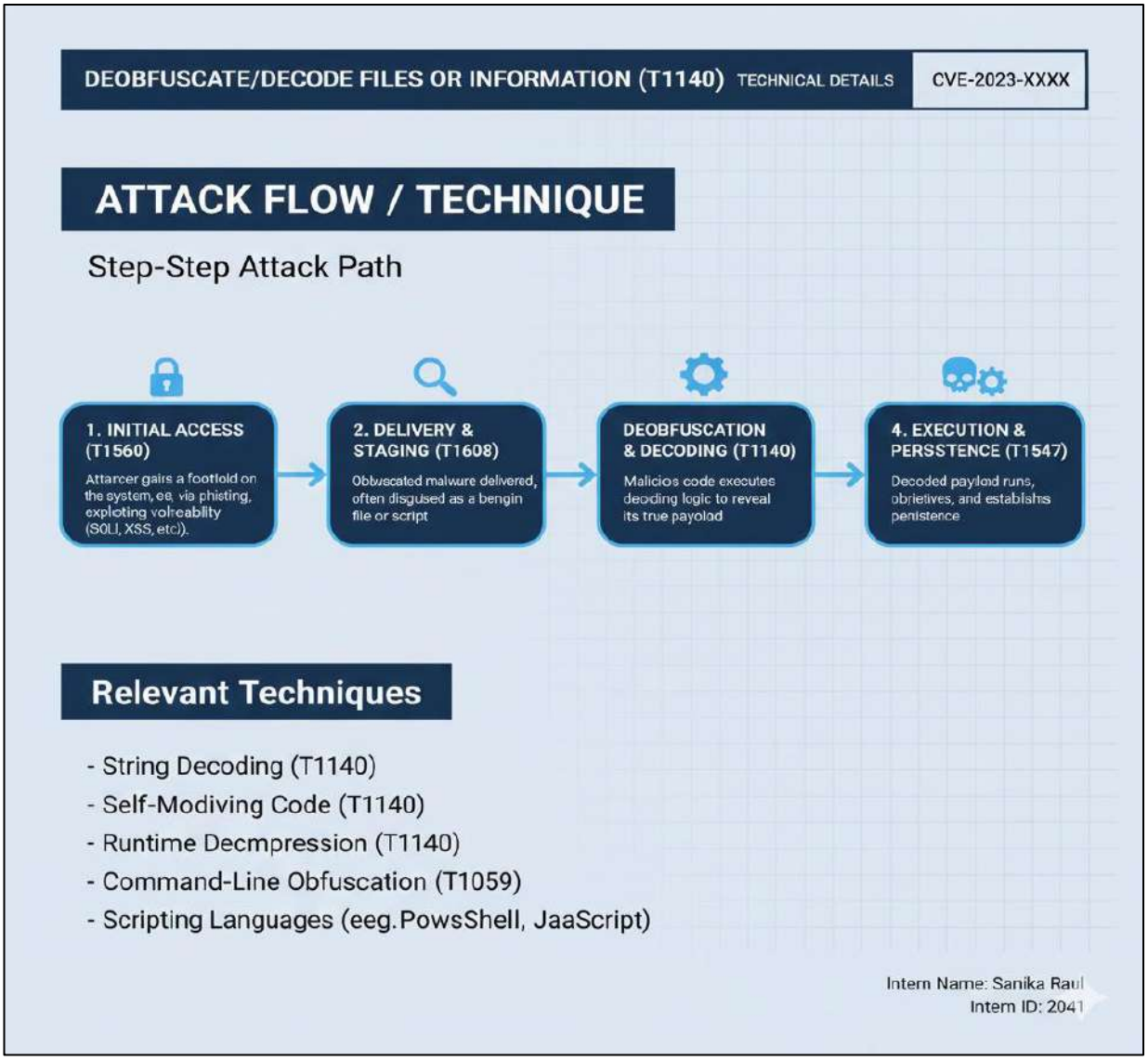
Overview



Technical details

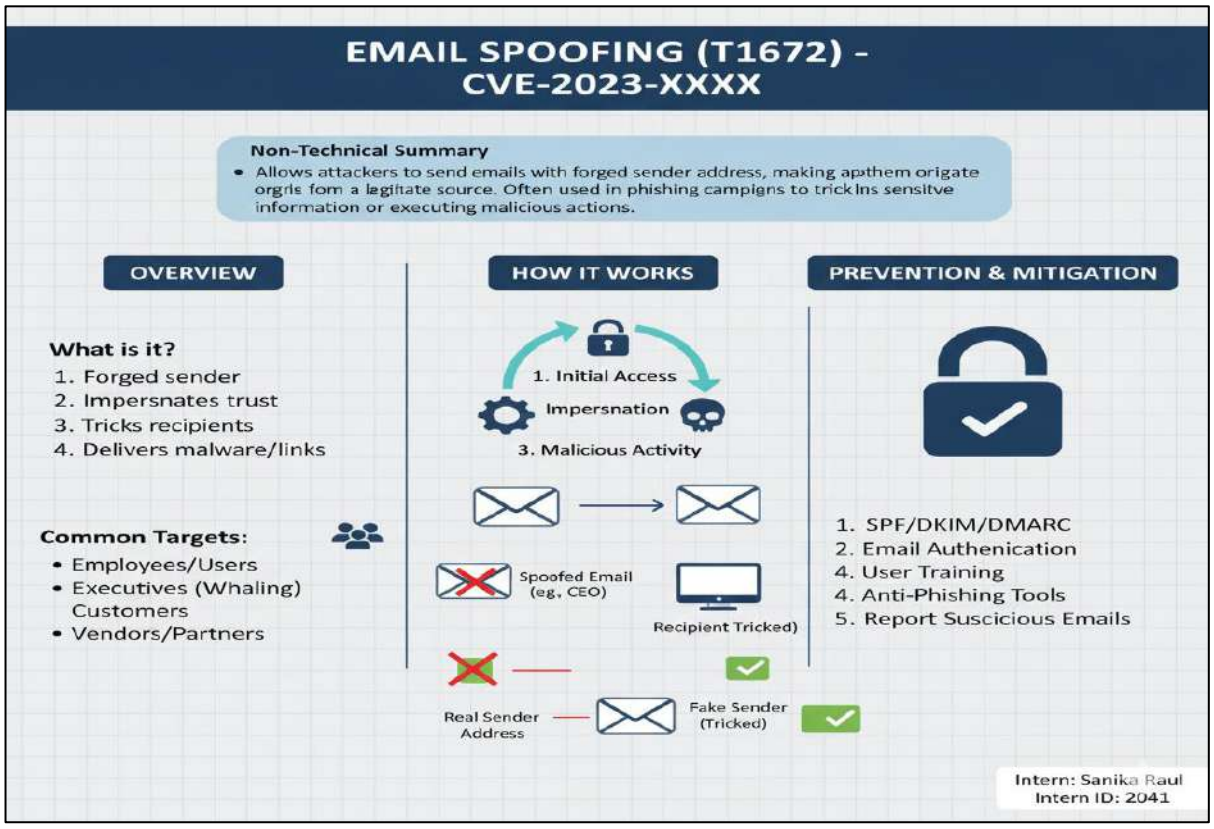


Attack flow / technique

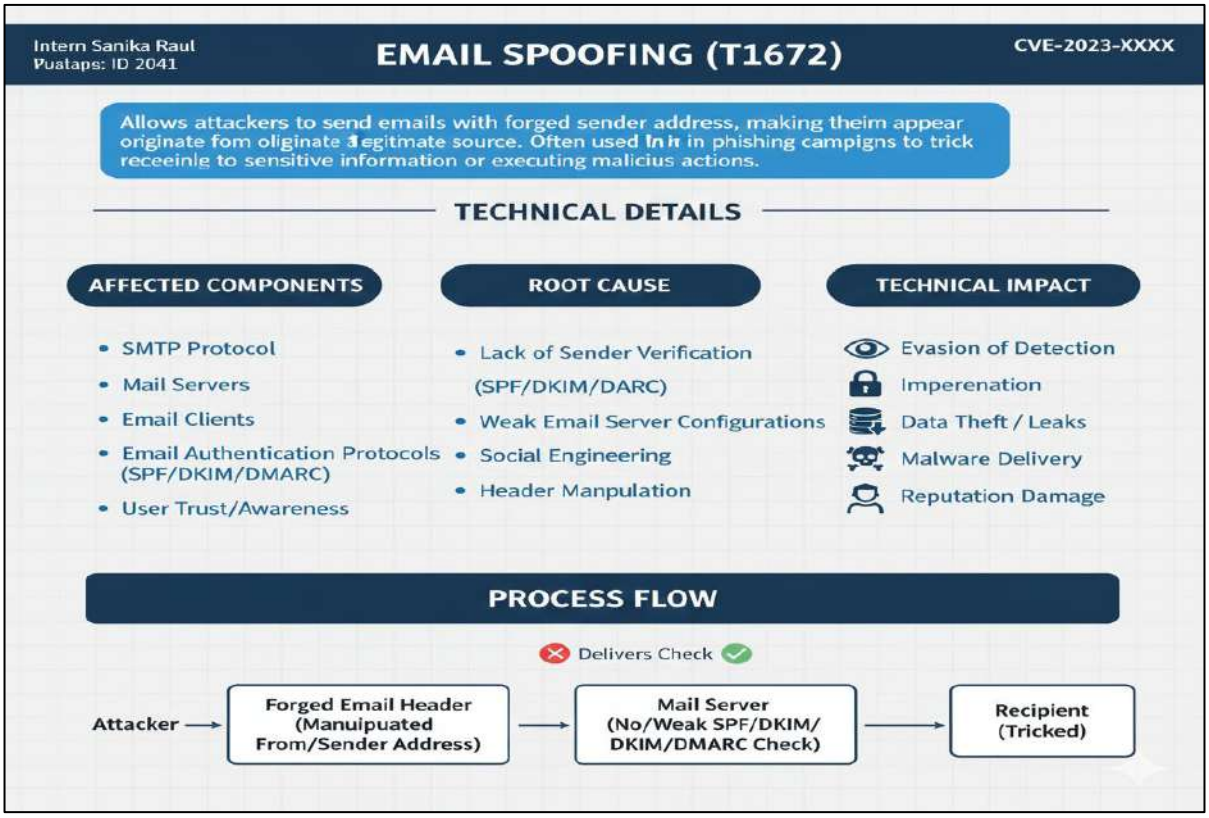


Email Spoofing (T1672)

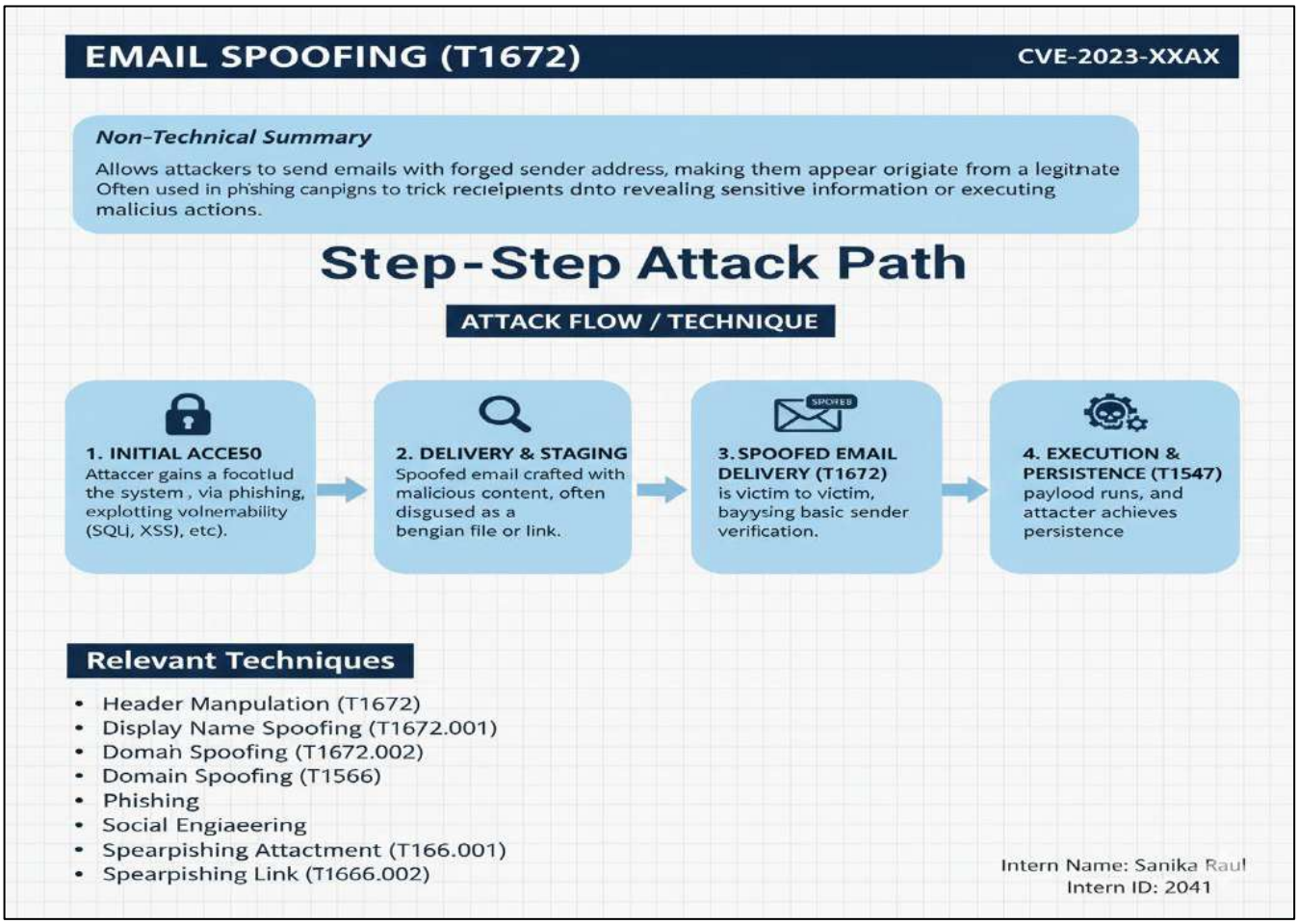
Overview



Technical details

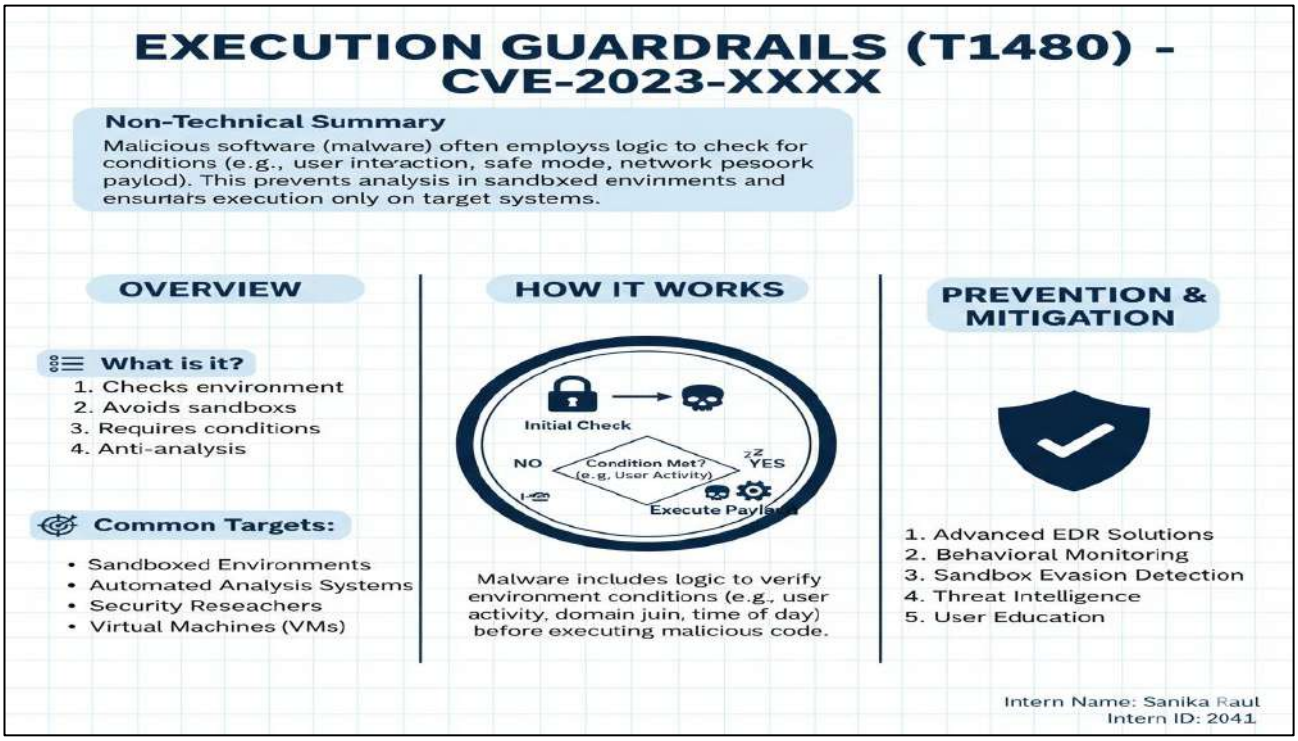


Attack flow / technique

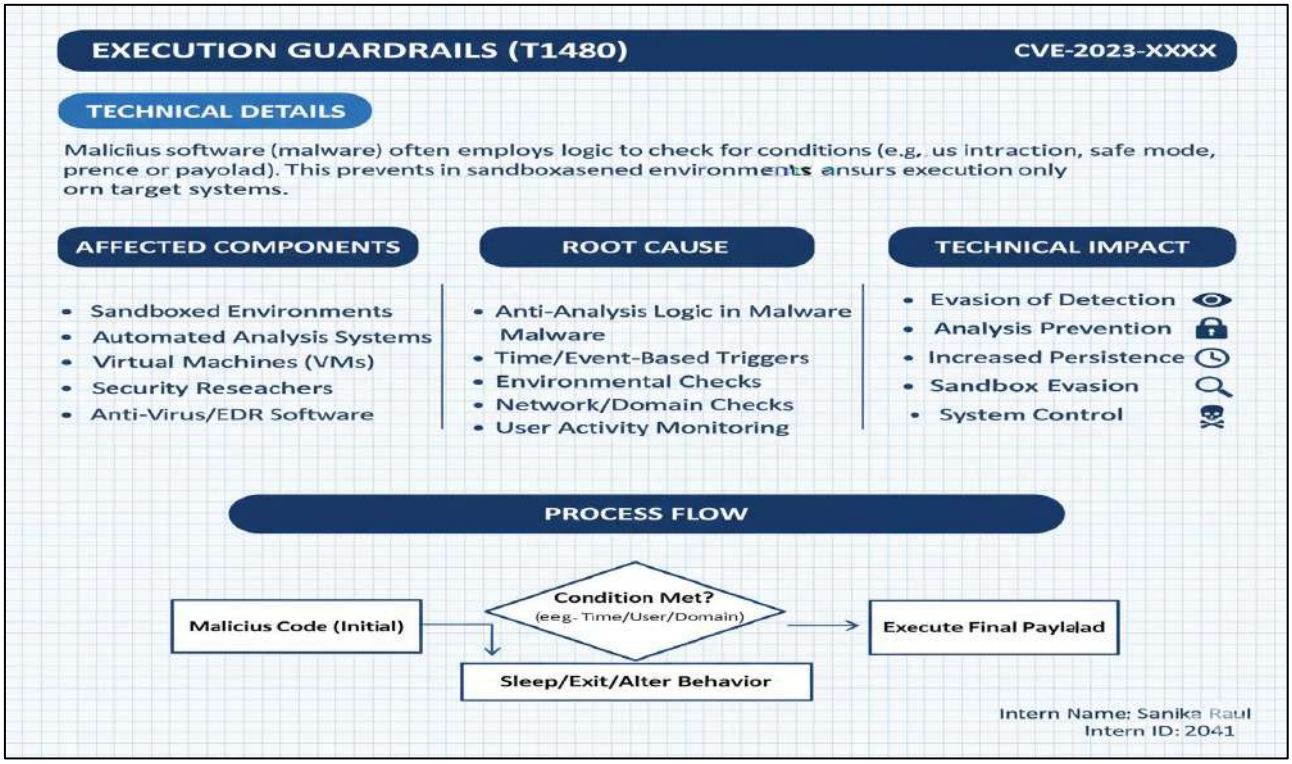


Execution Guardrails (T1480)

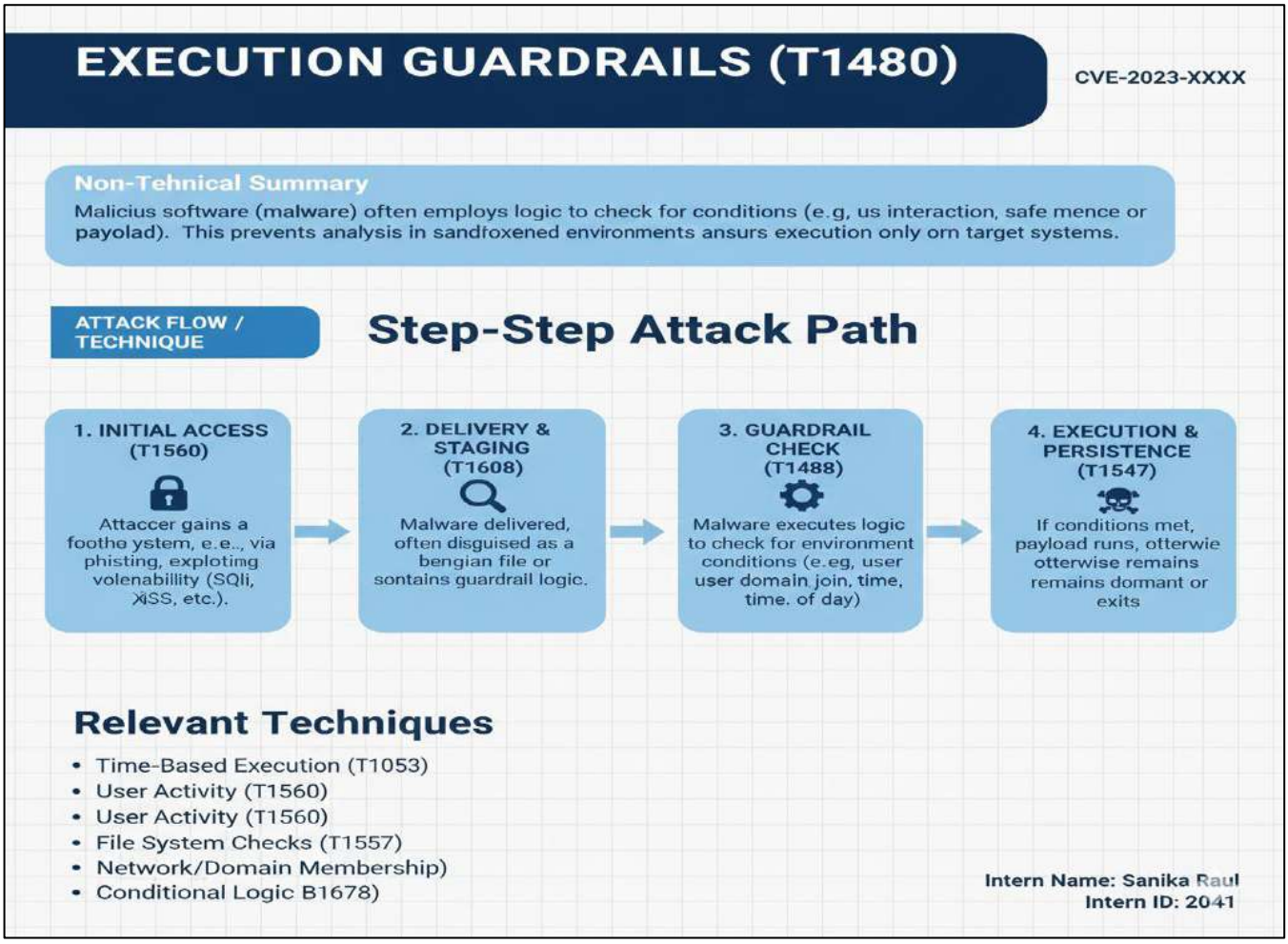
Overview



Technical details

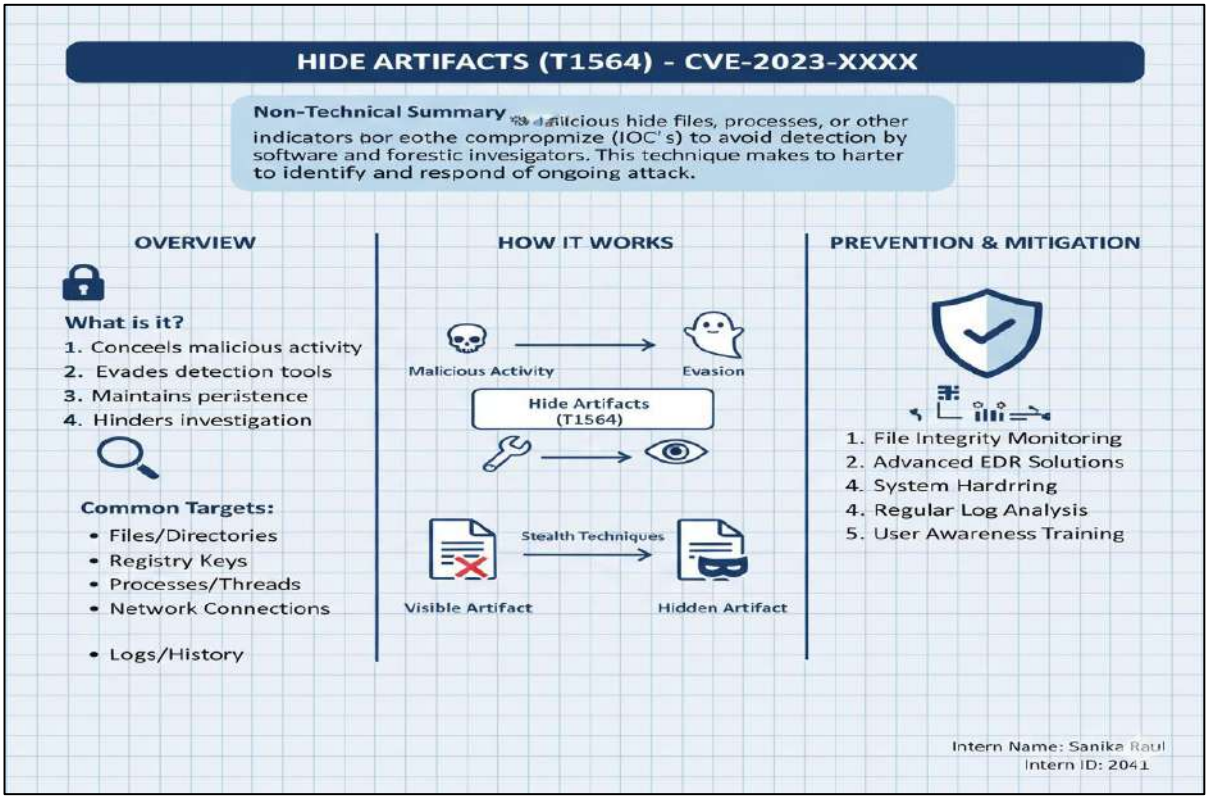


Attack flow / technique

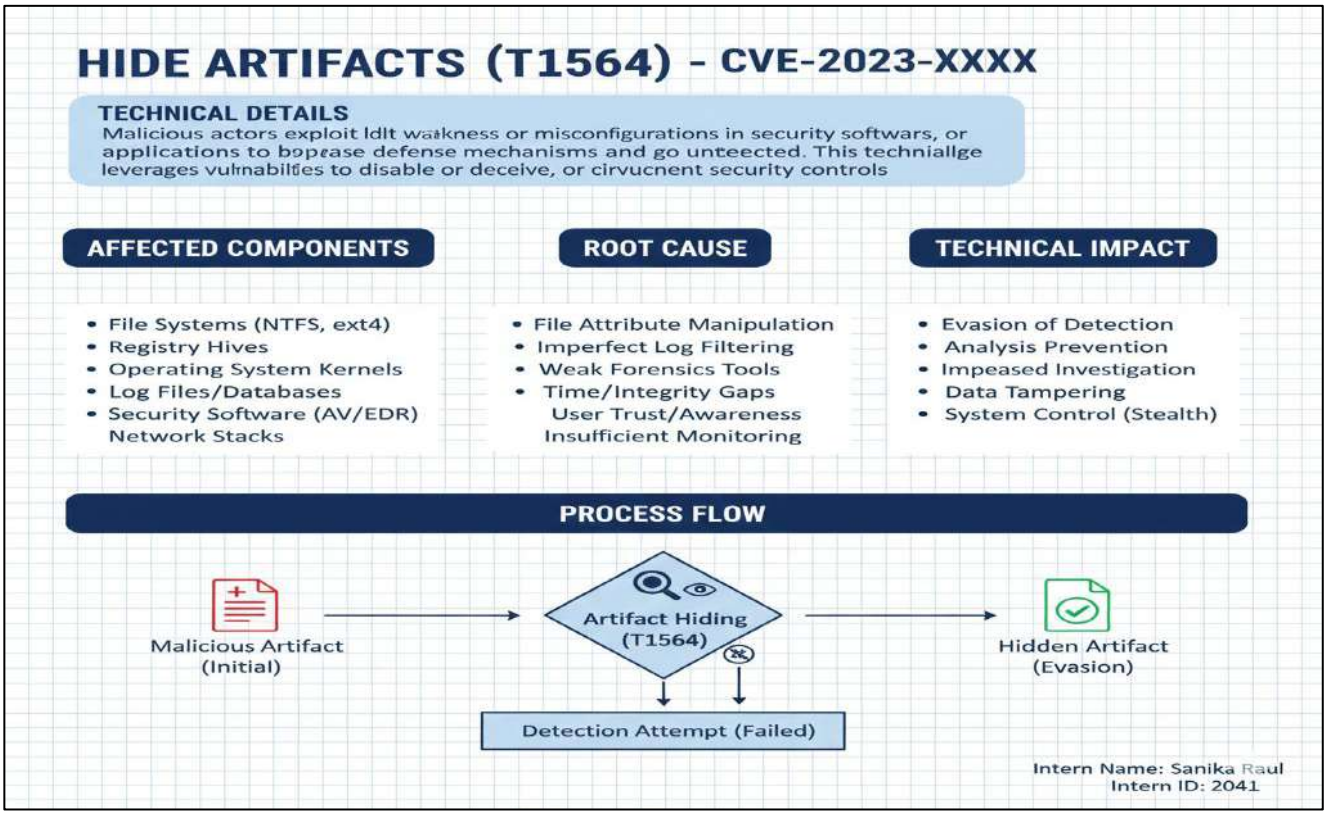


Hide Artifacts (T1564)

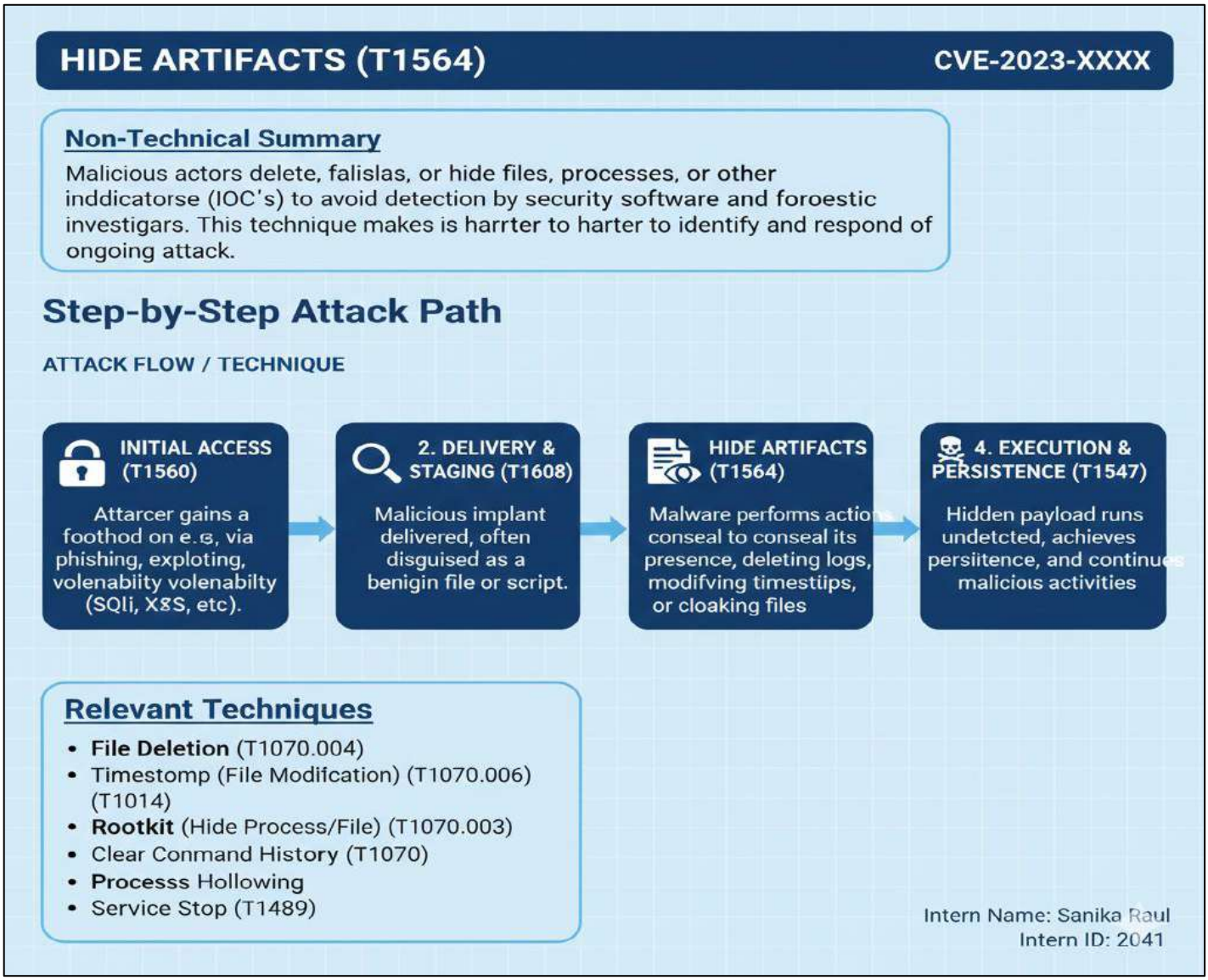
Overview



Technical details



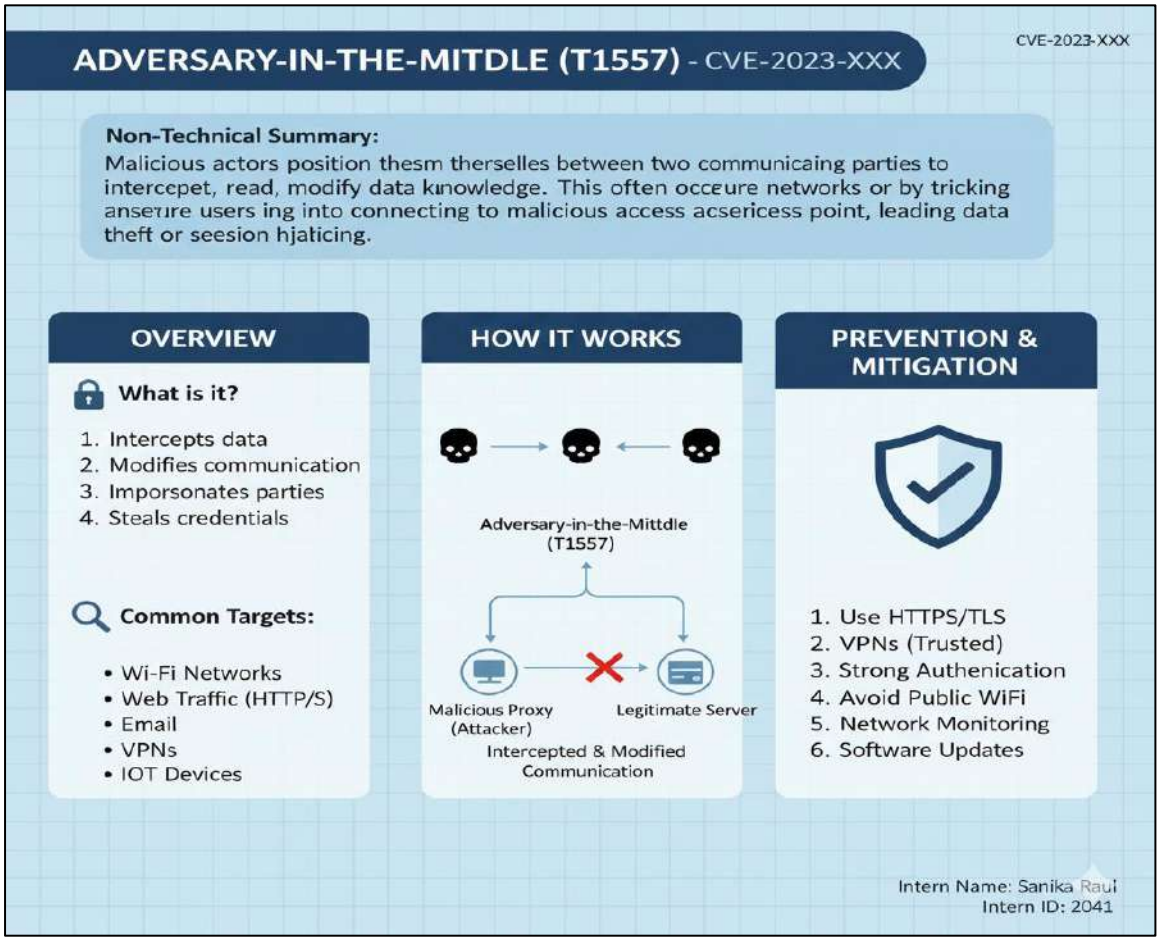
Attack flow / technique



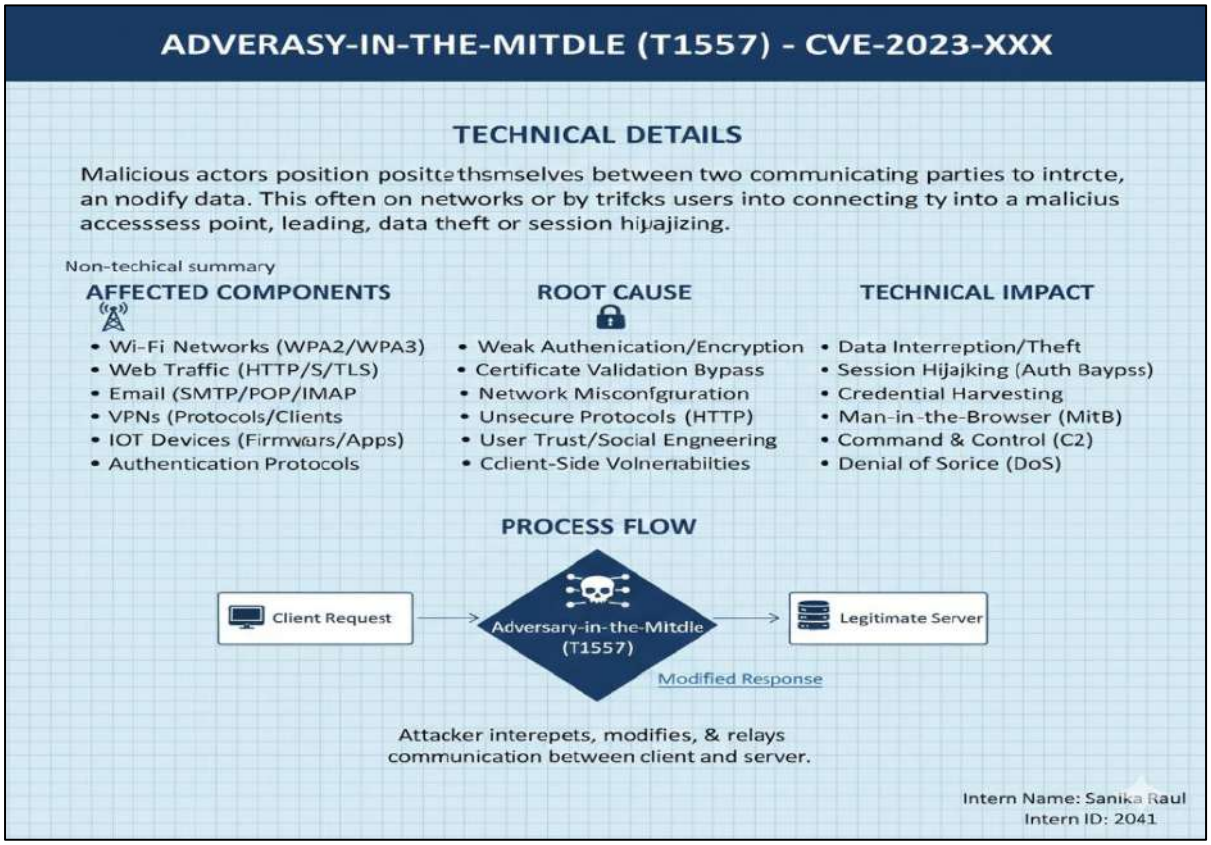
Credential Access

Adversary-in-the-Middle (T1557)

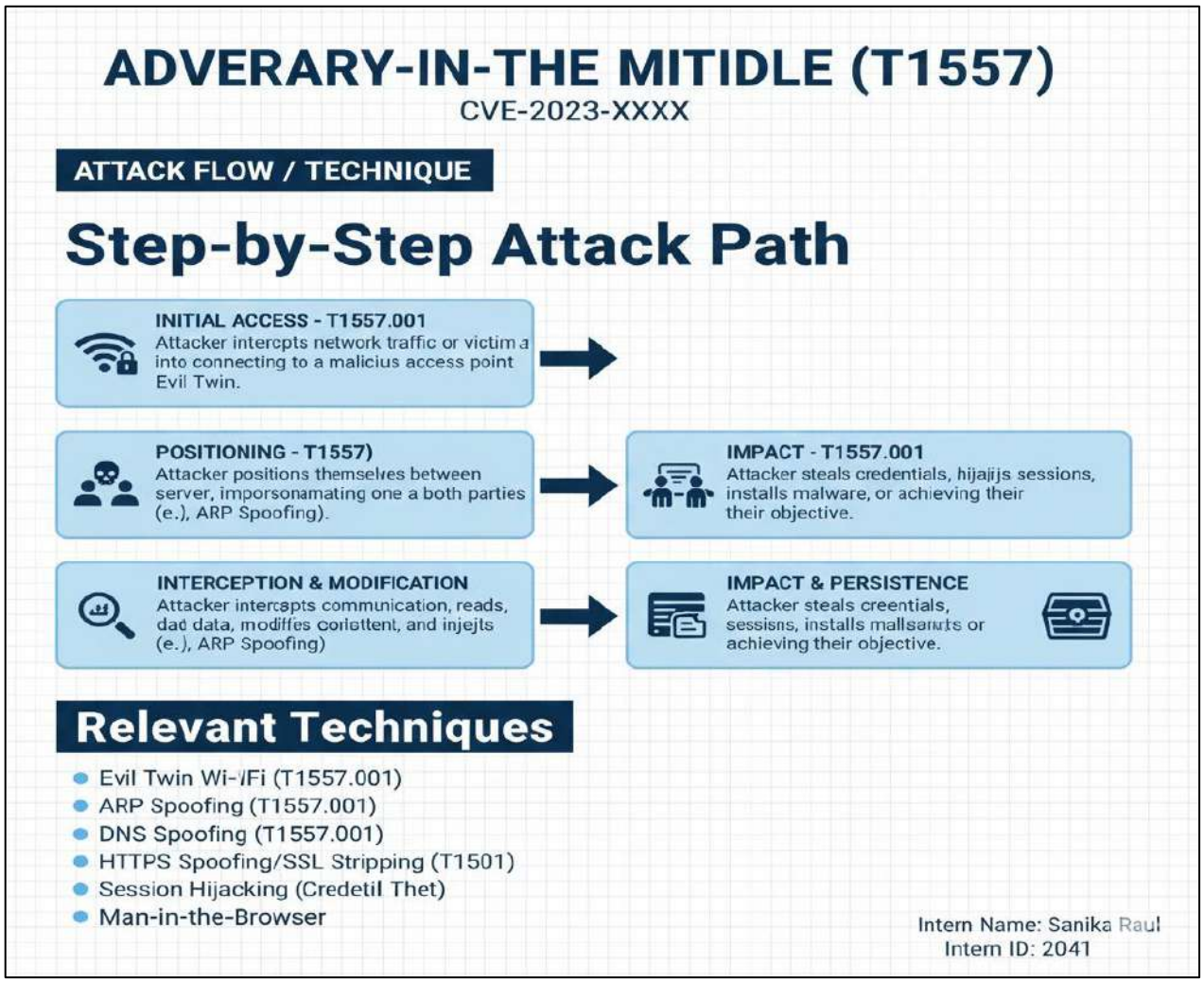
Overview



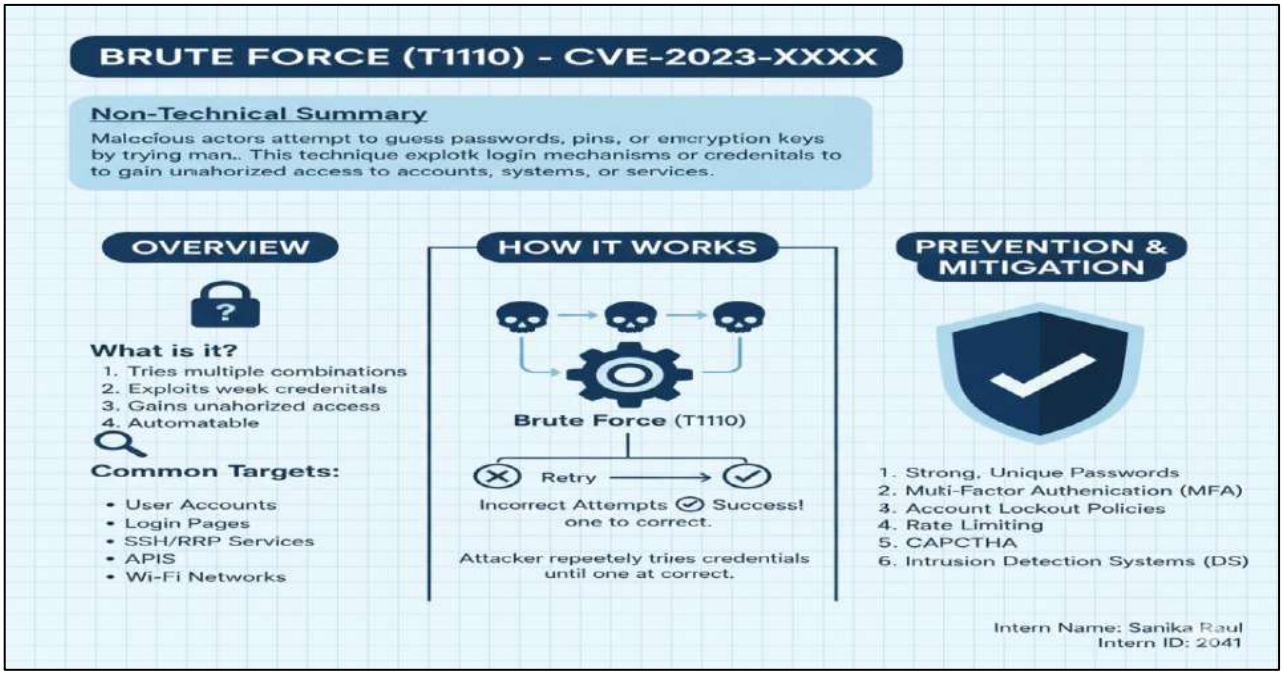
Technical details



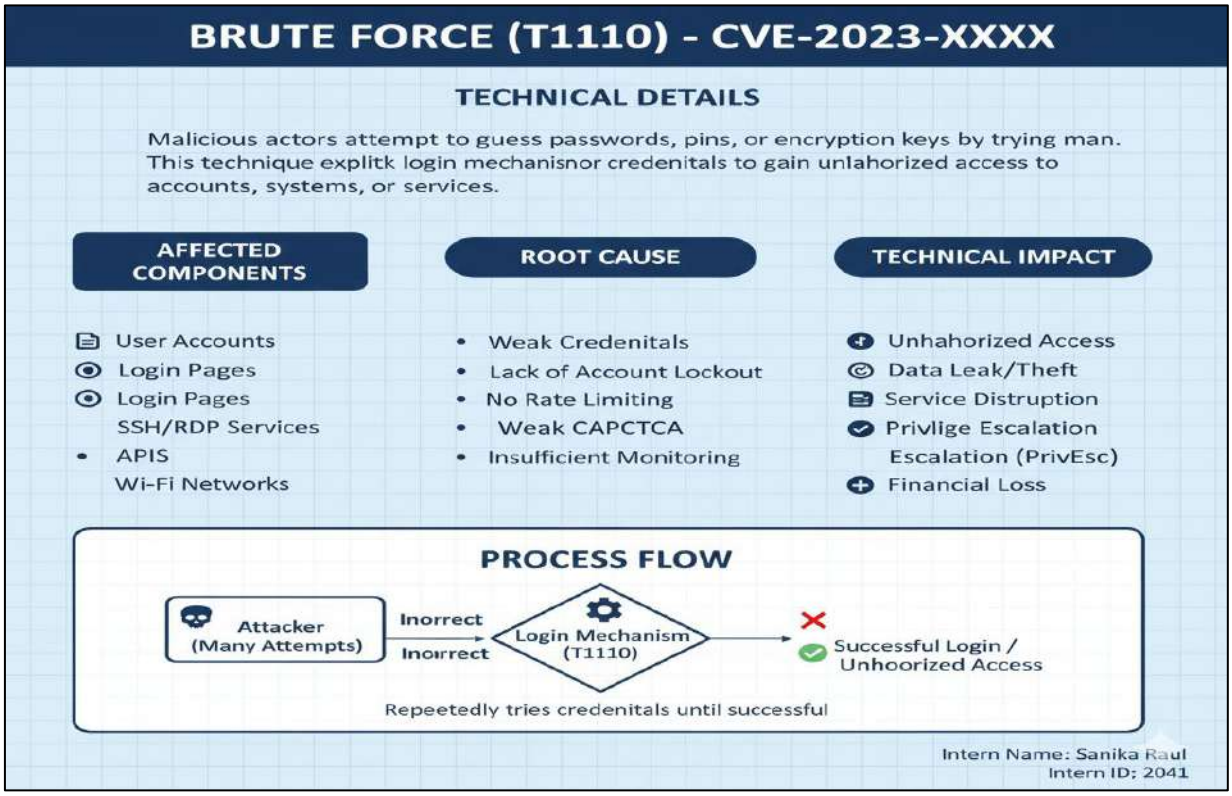
Attack flow / technique



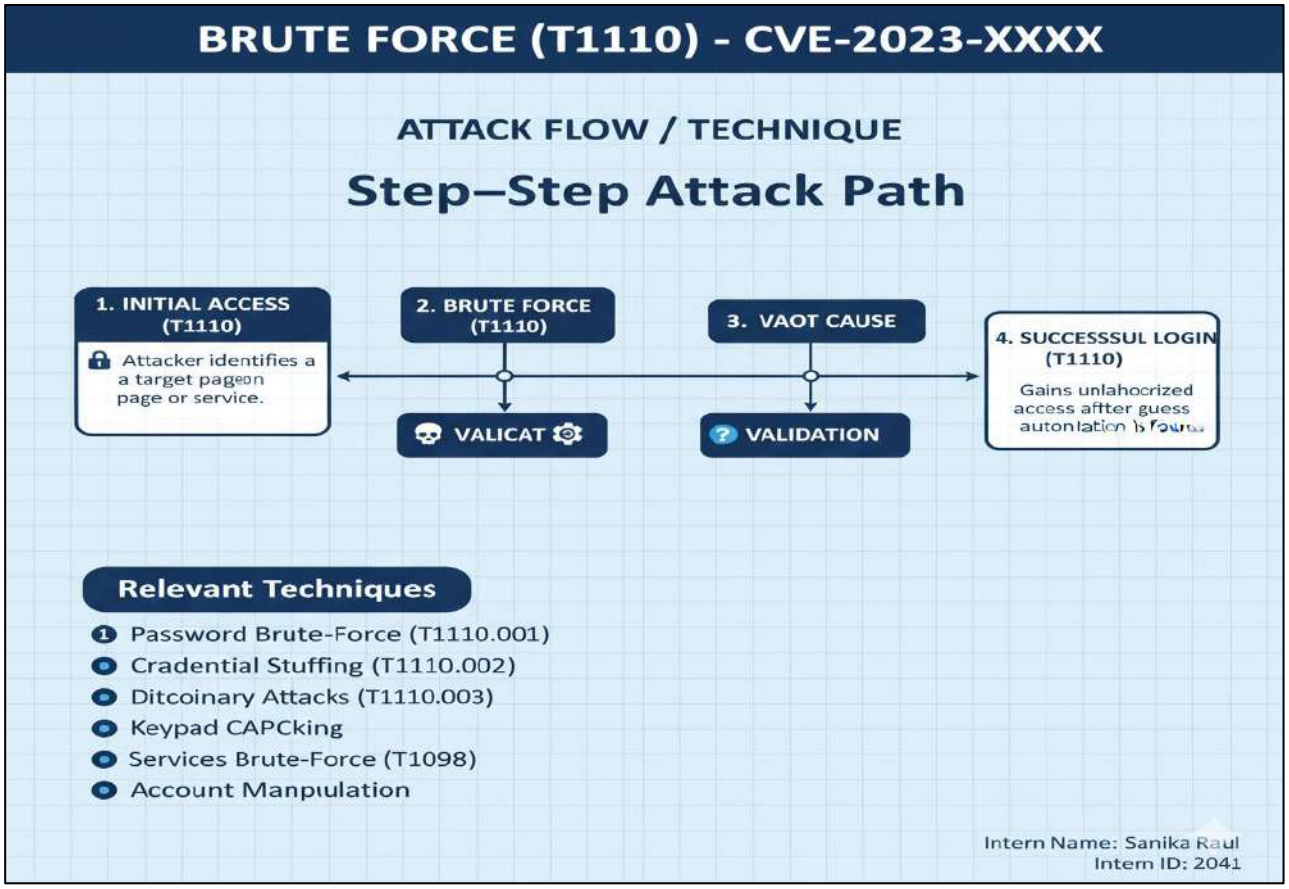
Brute Force (T1110)



Technical details

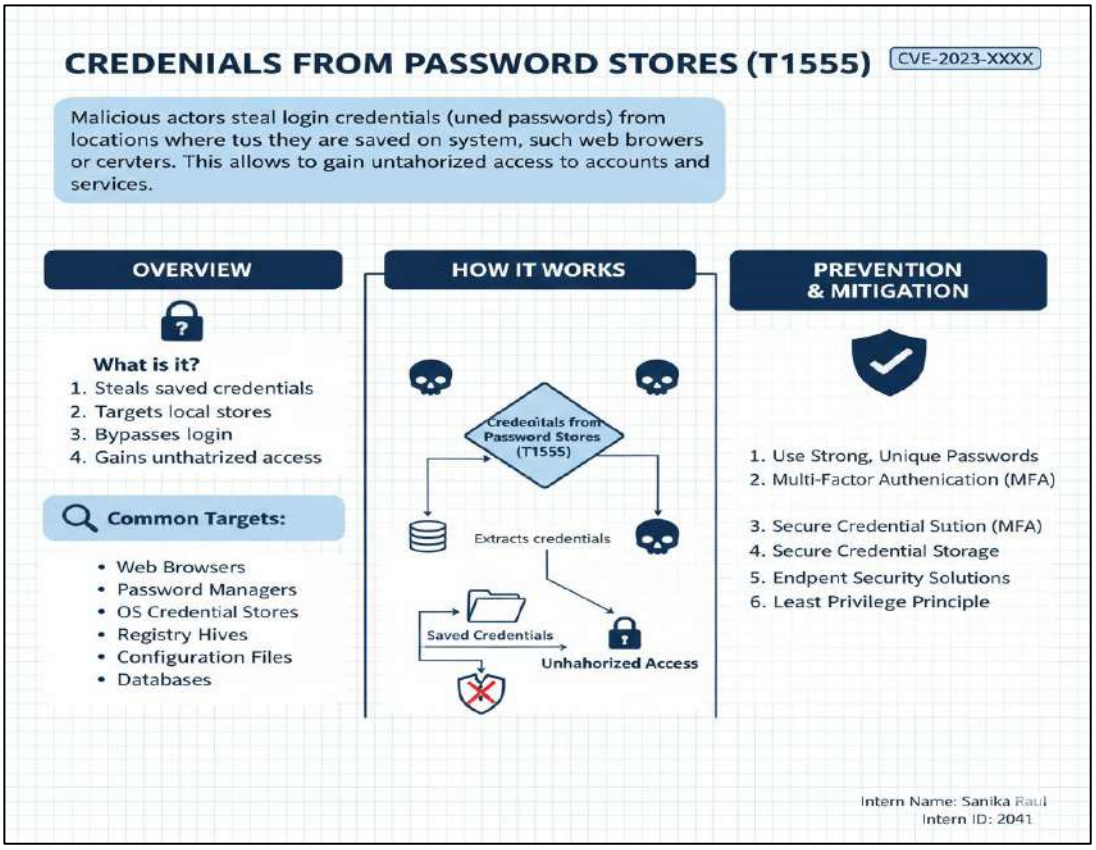


Attack flow / technique

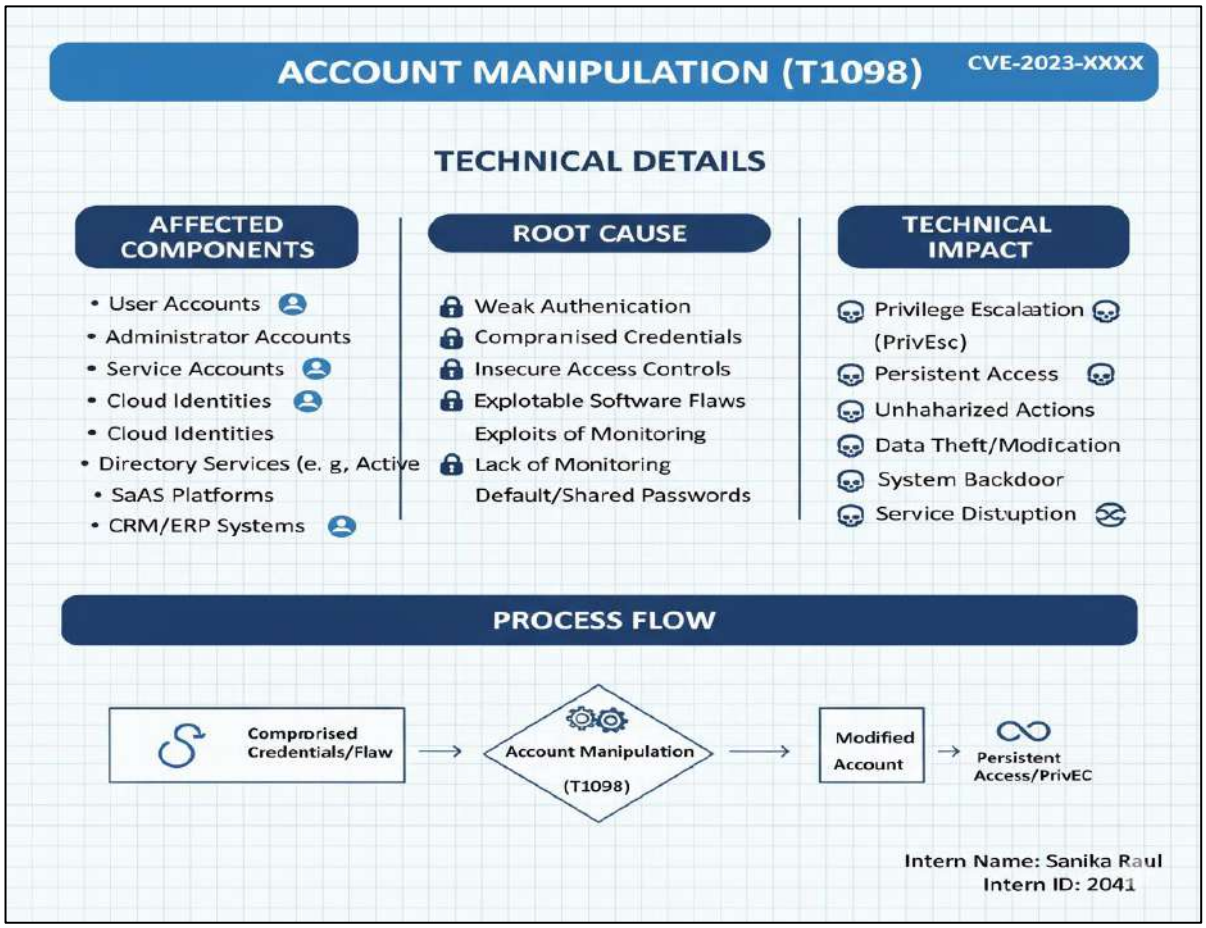


Credentials from Password Stores (T1555)

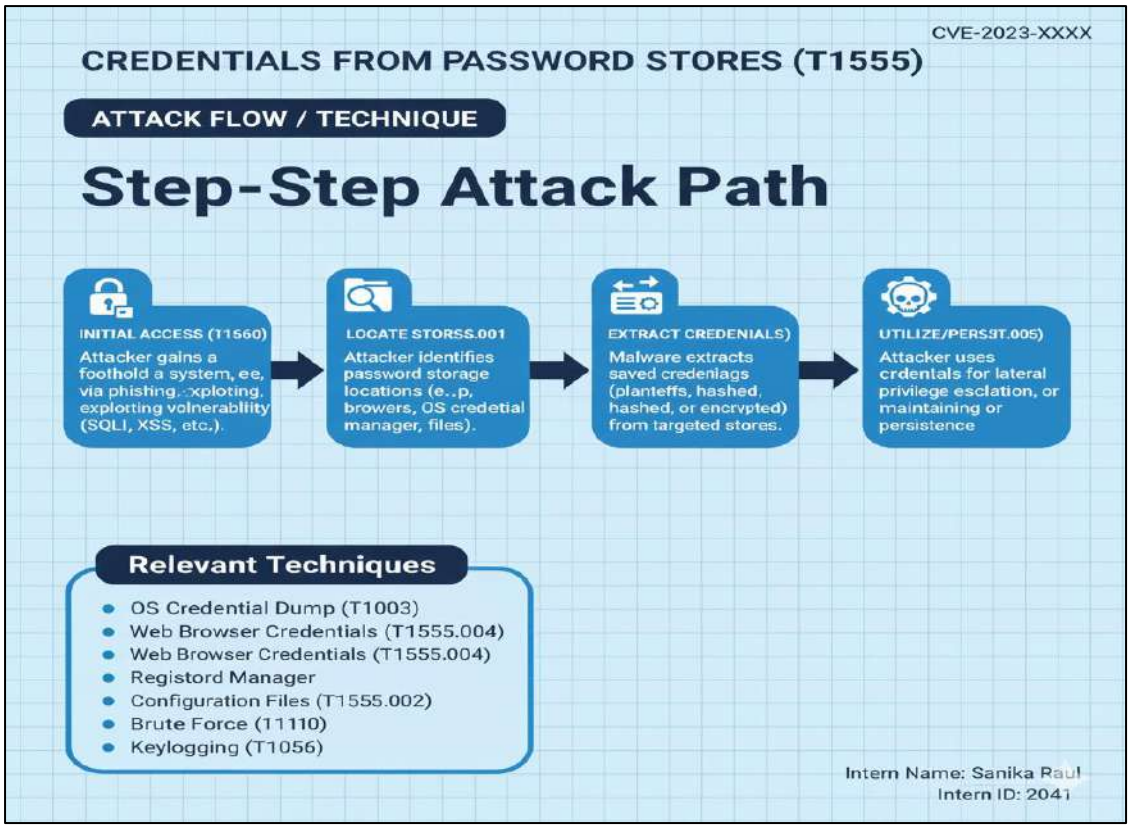
Overview



Technical details

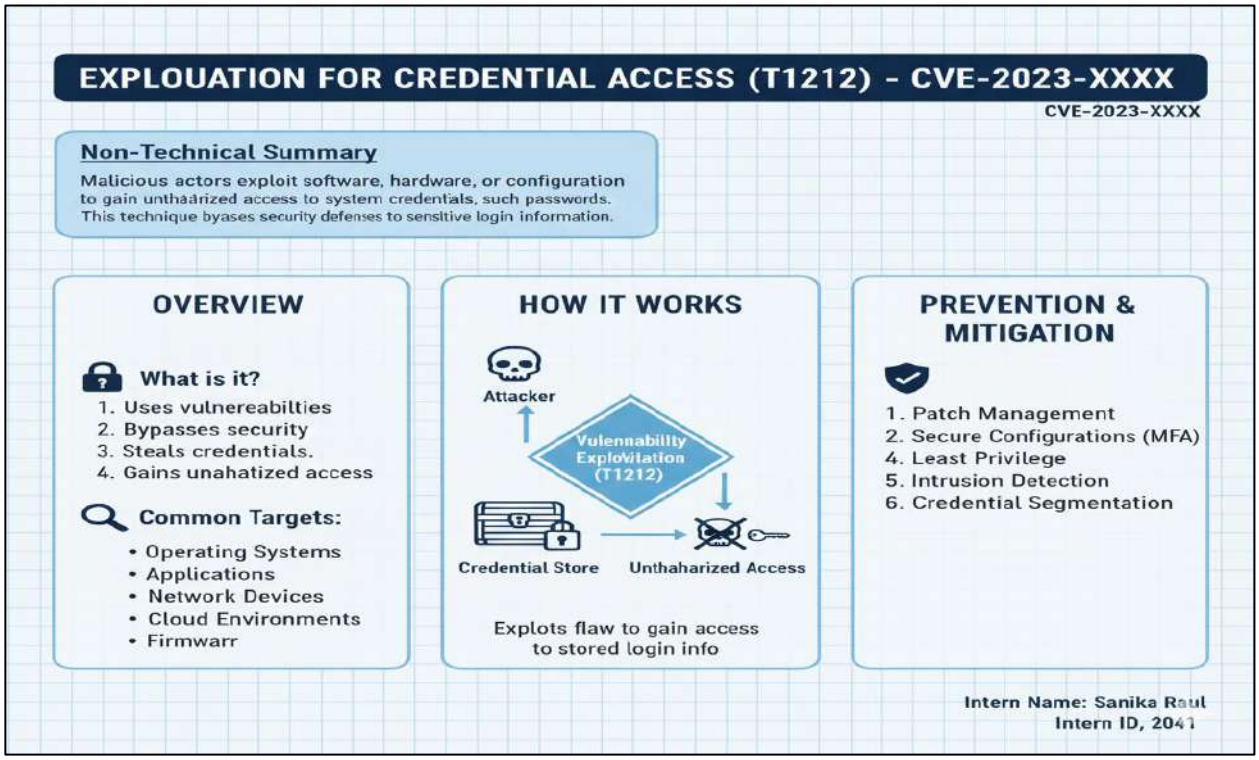


Attack flow / technique

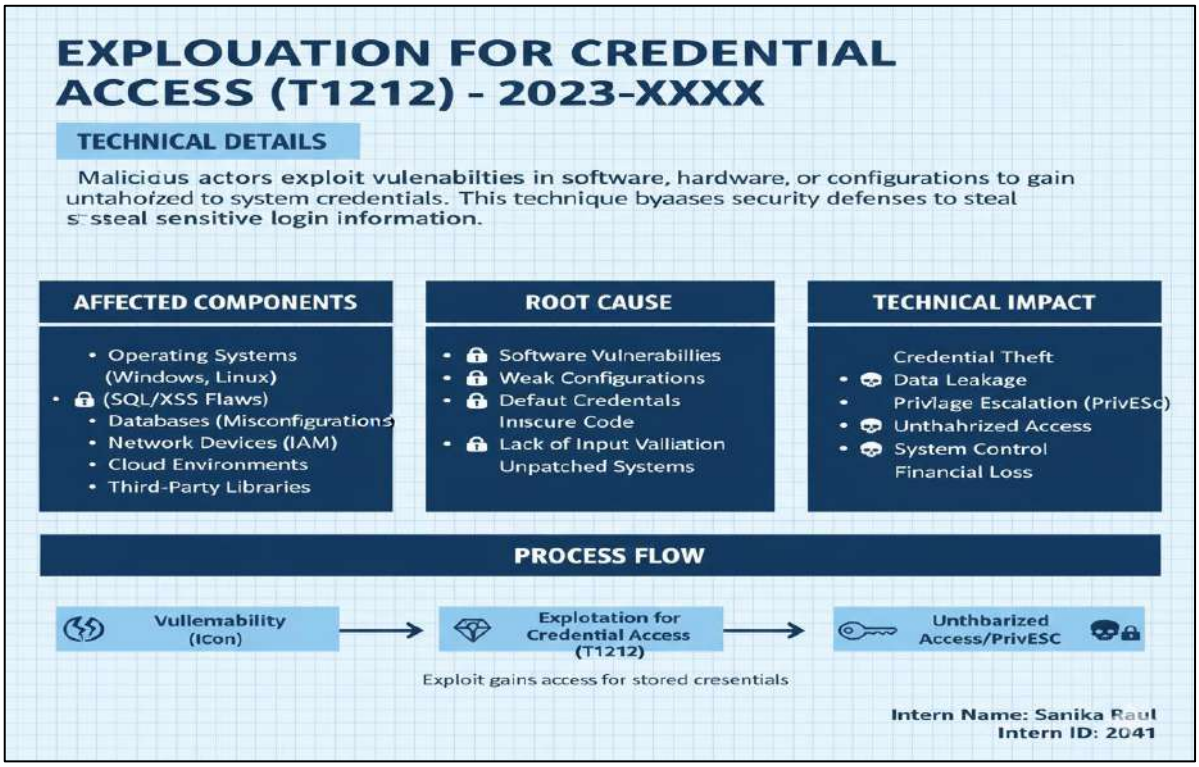


Exploitation for Credential Access (T1212)

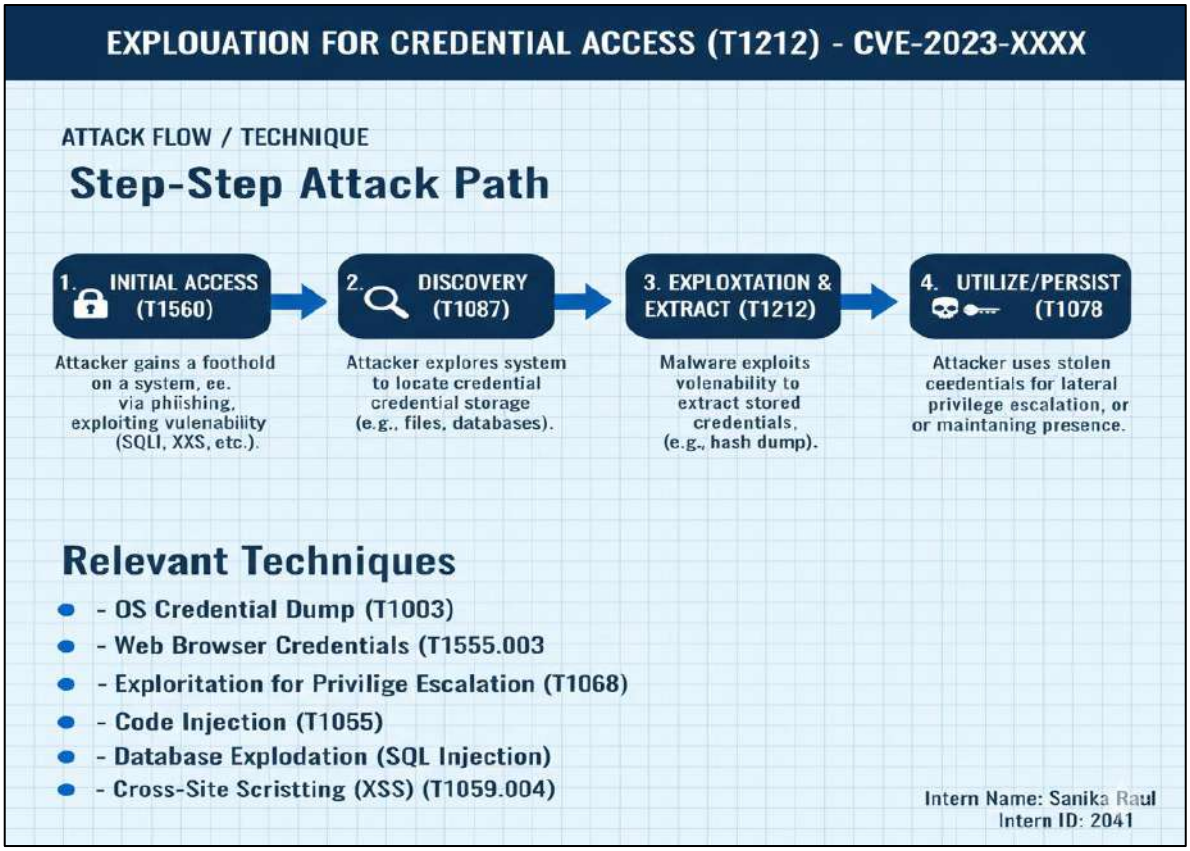
Overview



Technical details

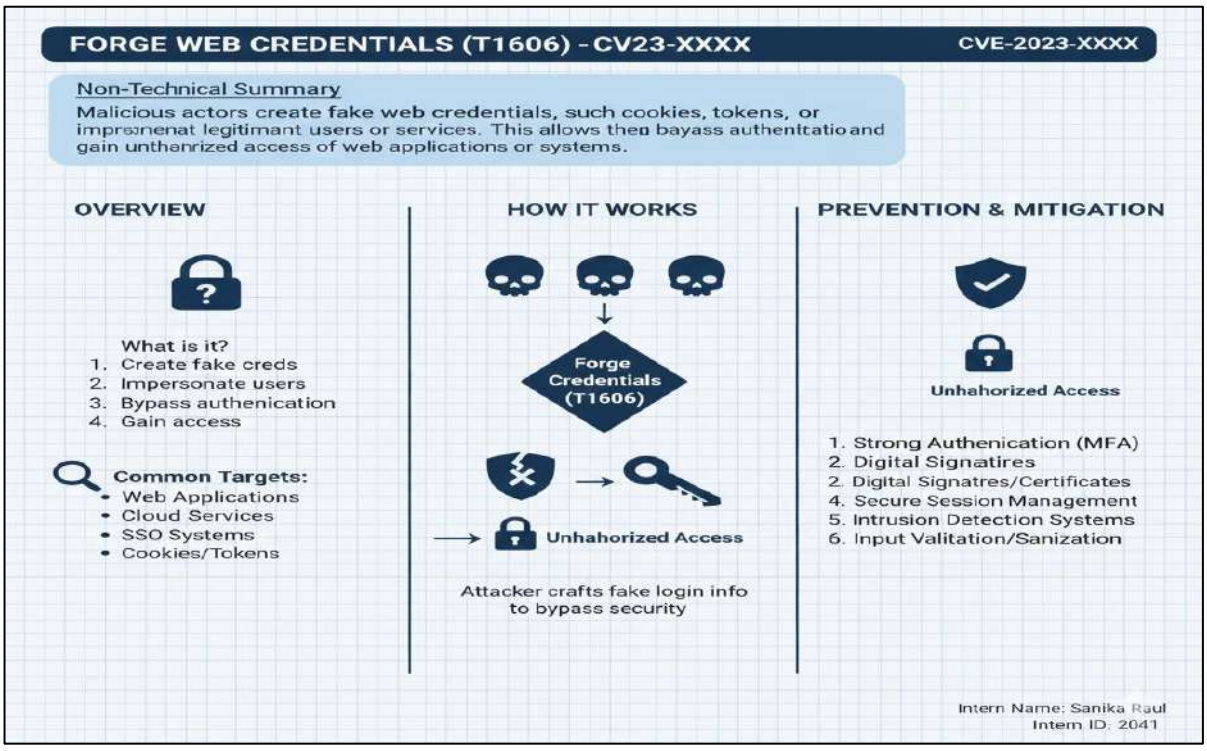


Attack flow / technique

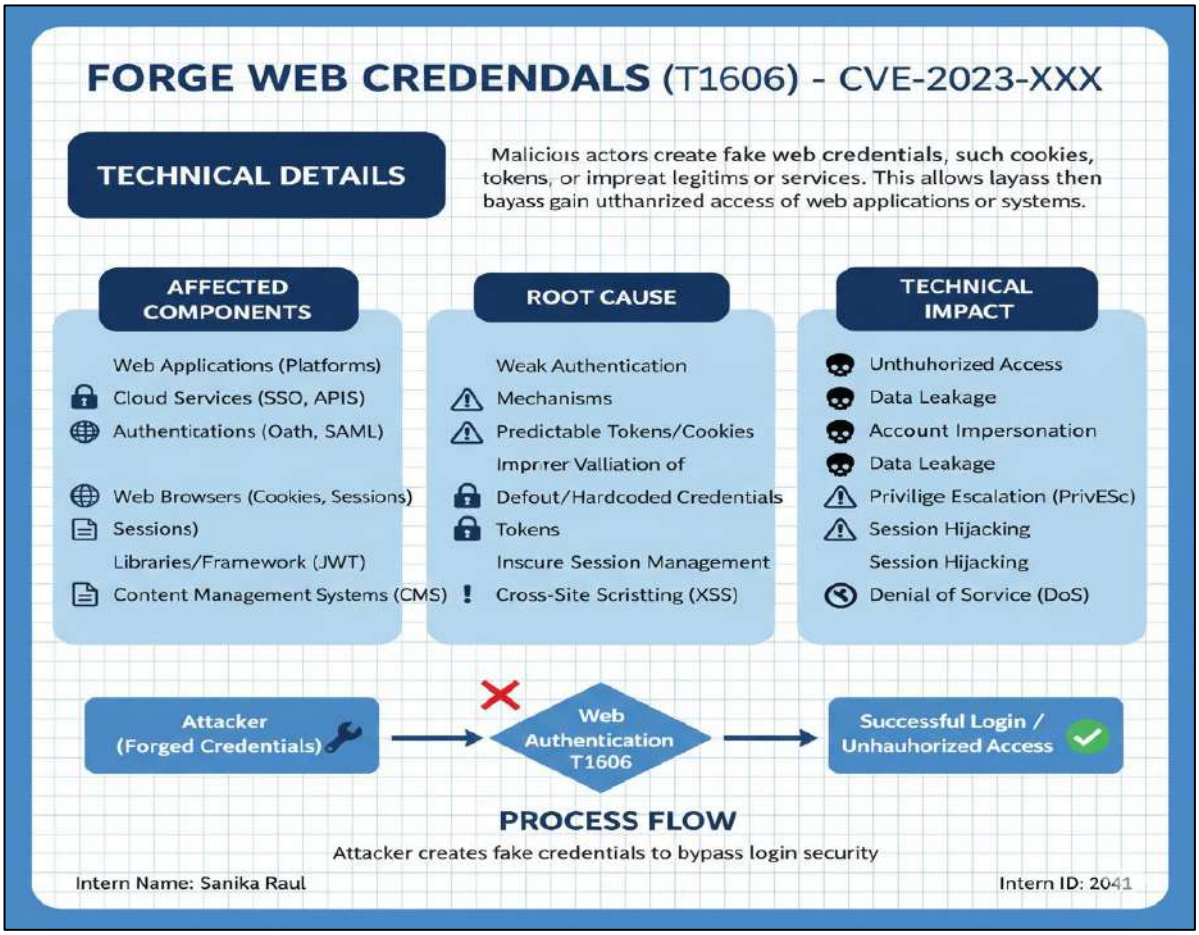


Forge Web Credentials (T1606)

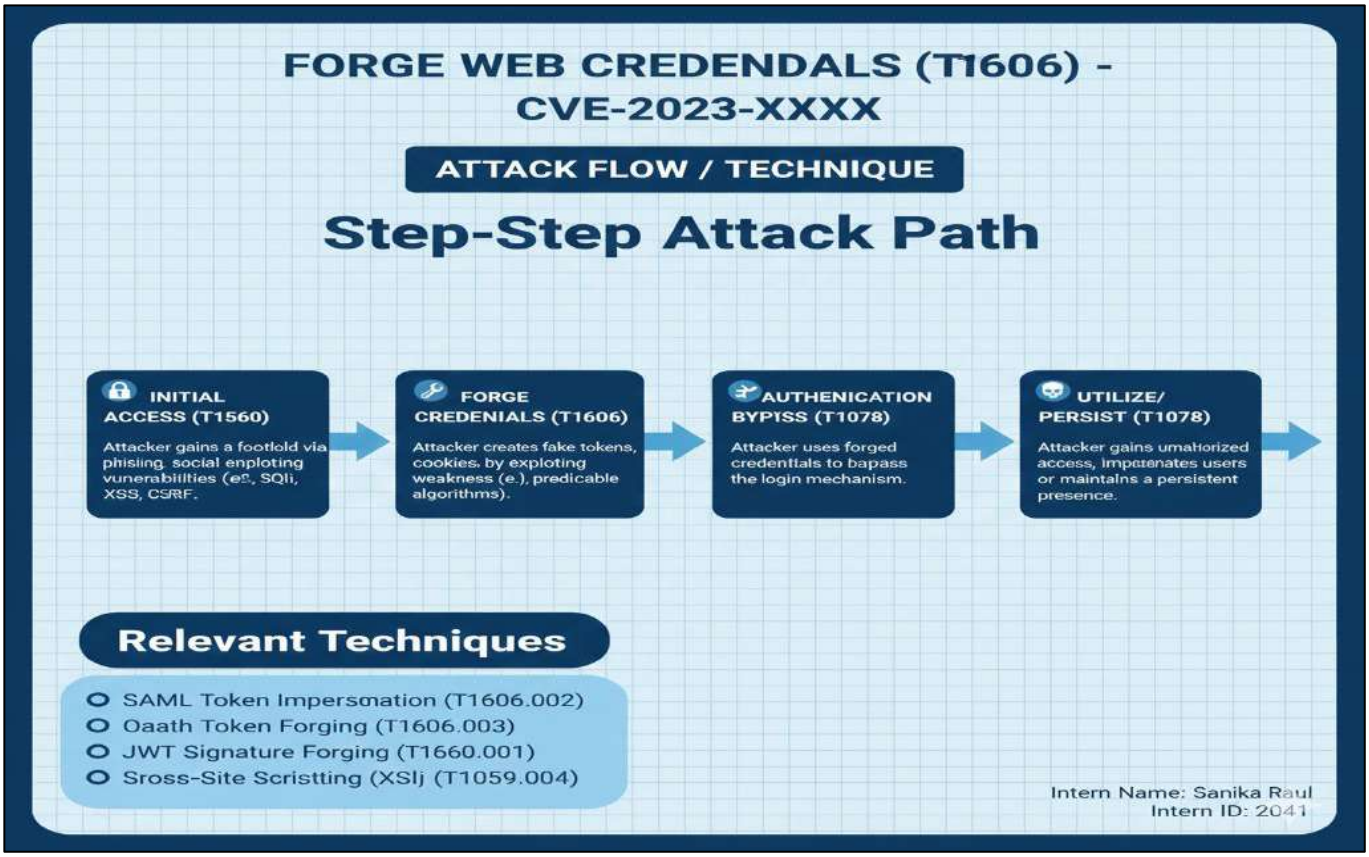
Overview



Technical details

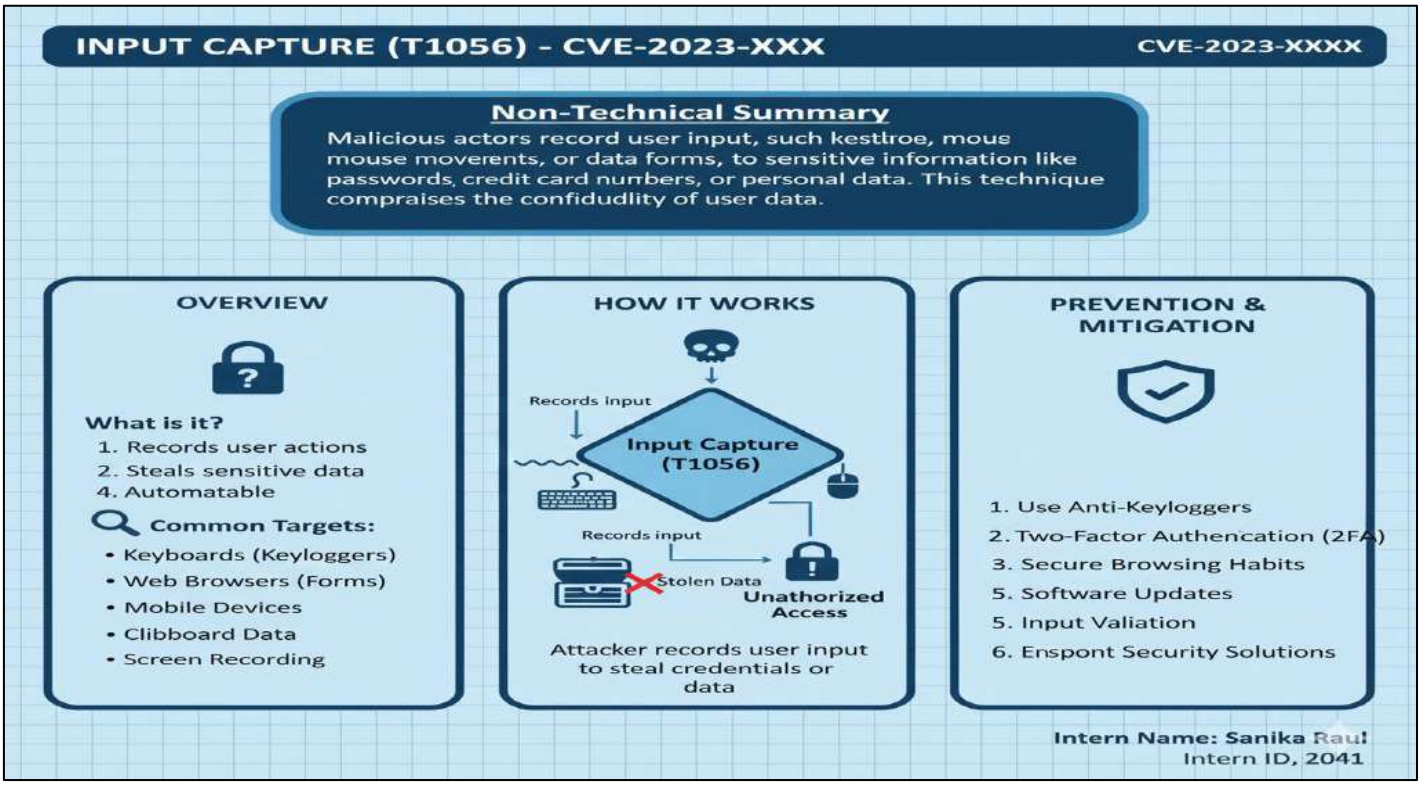


Attack flow / technique

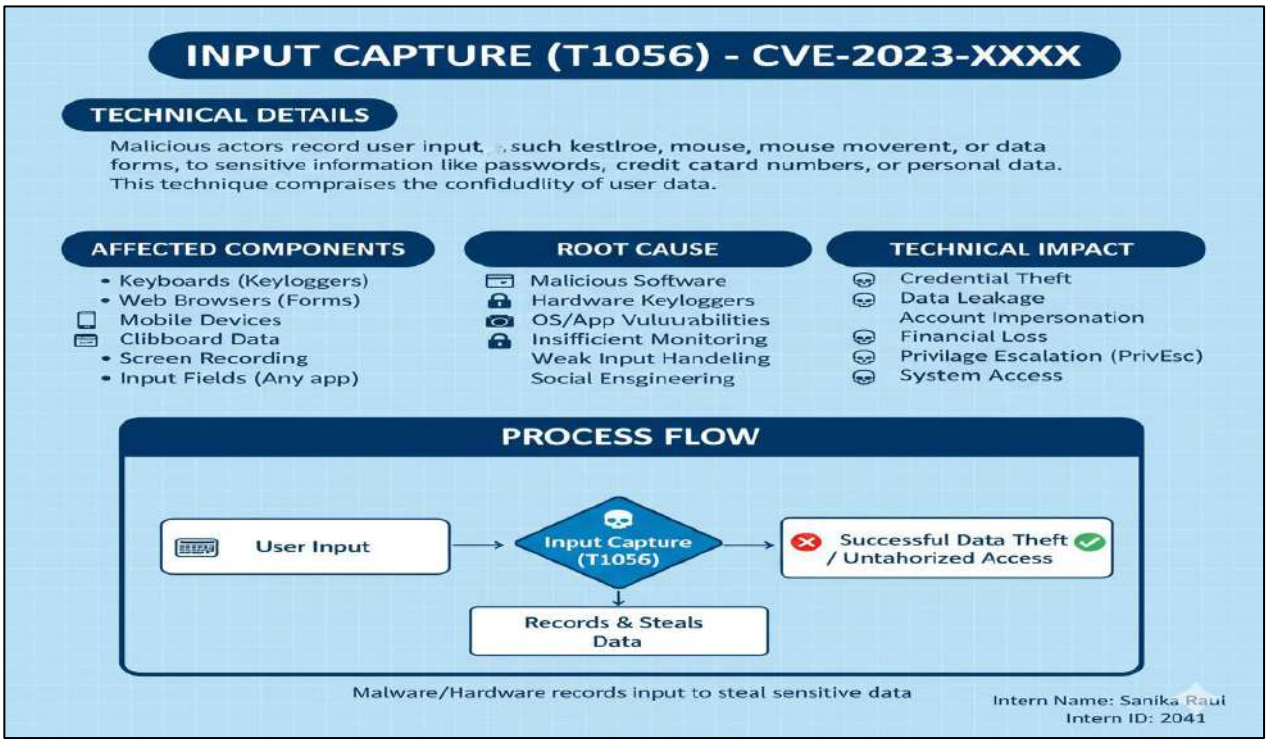


Input Capture (T1056)

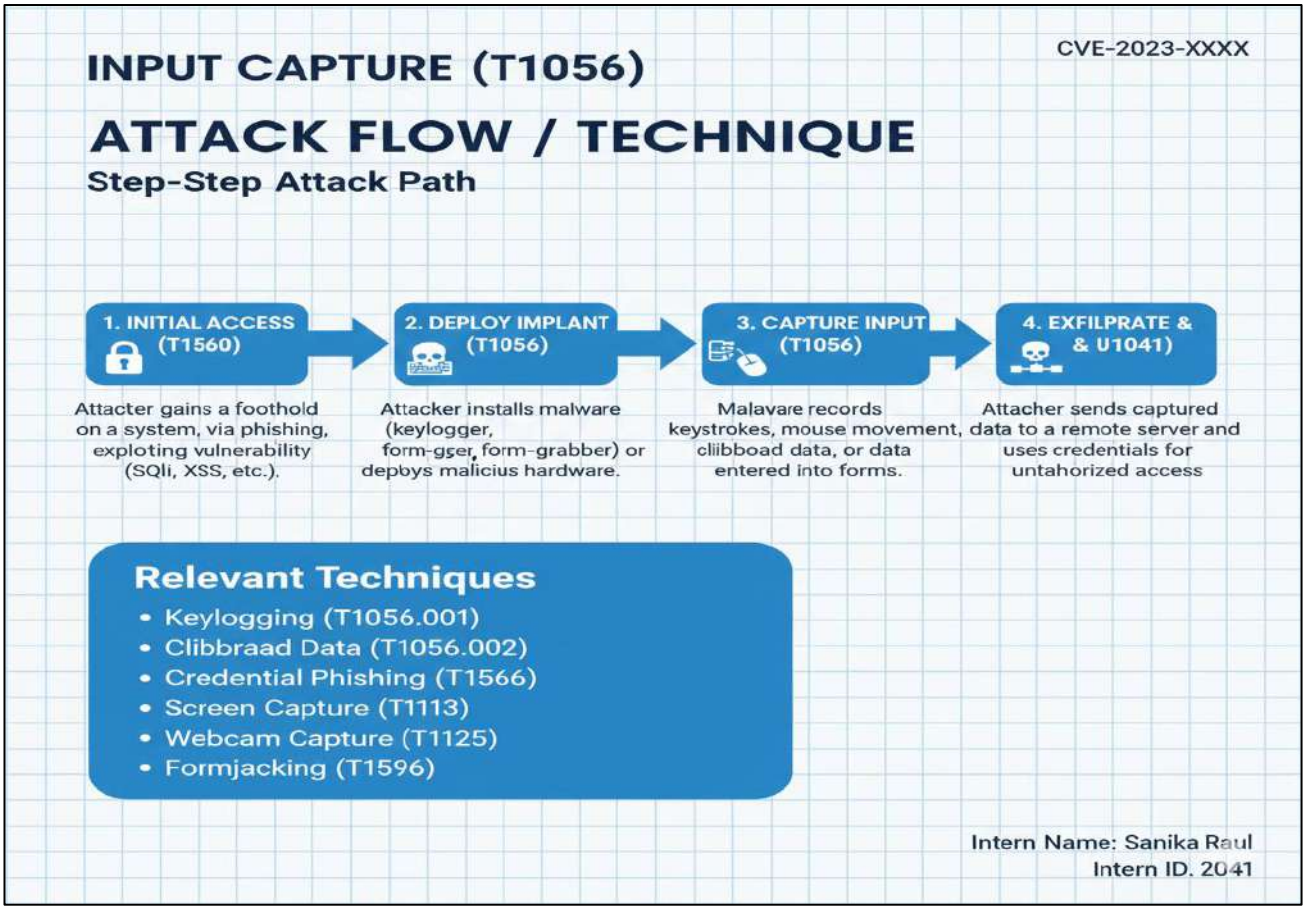
Overview



Technical details

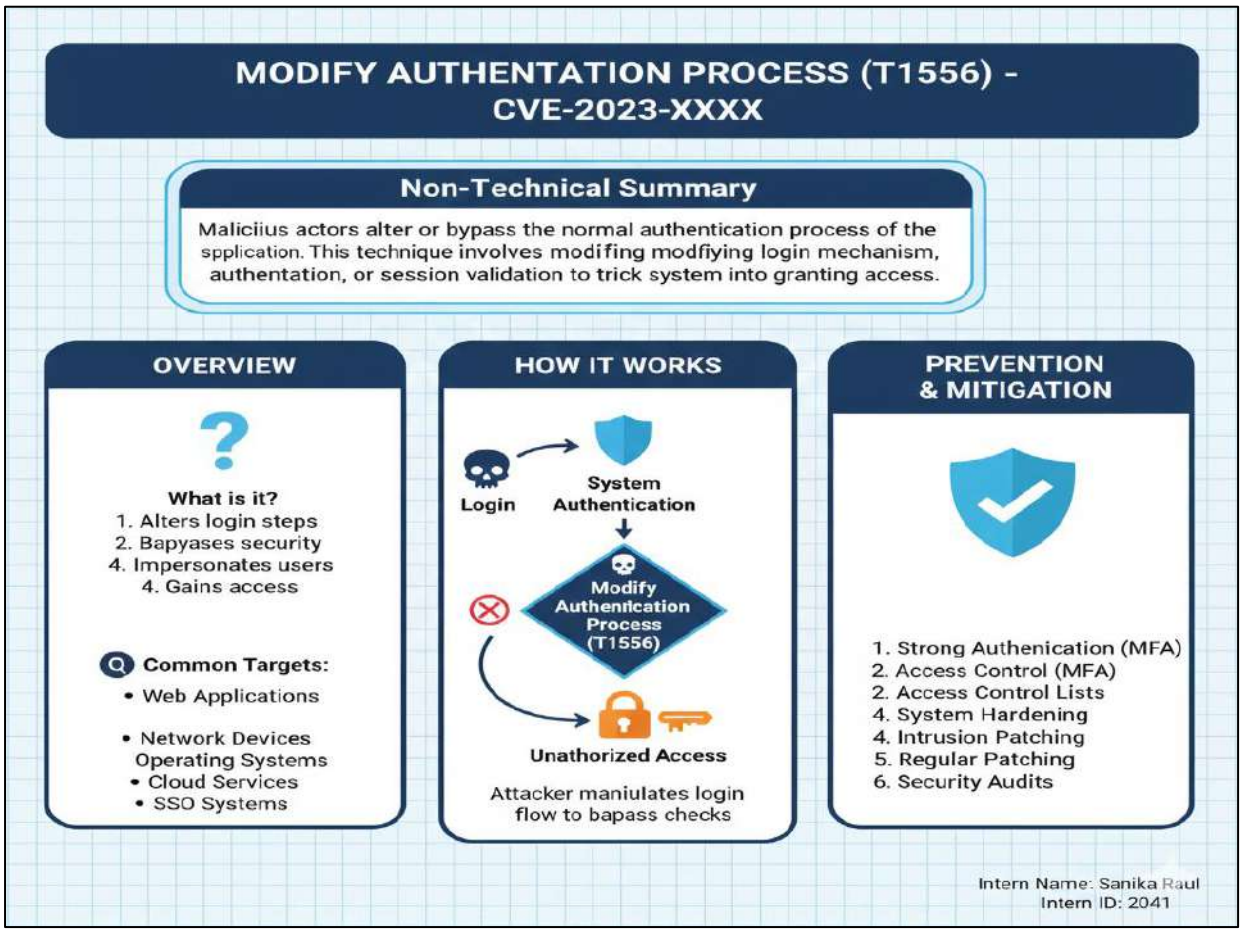


Attack flow / technique



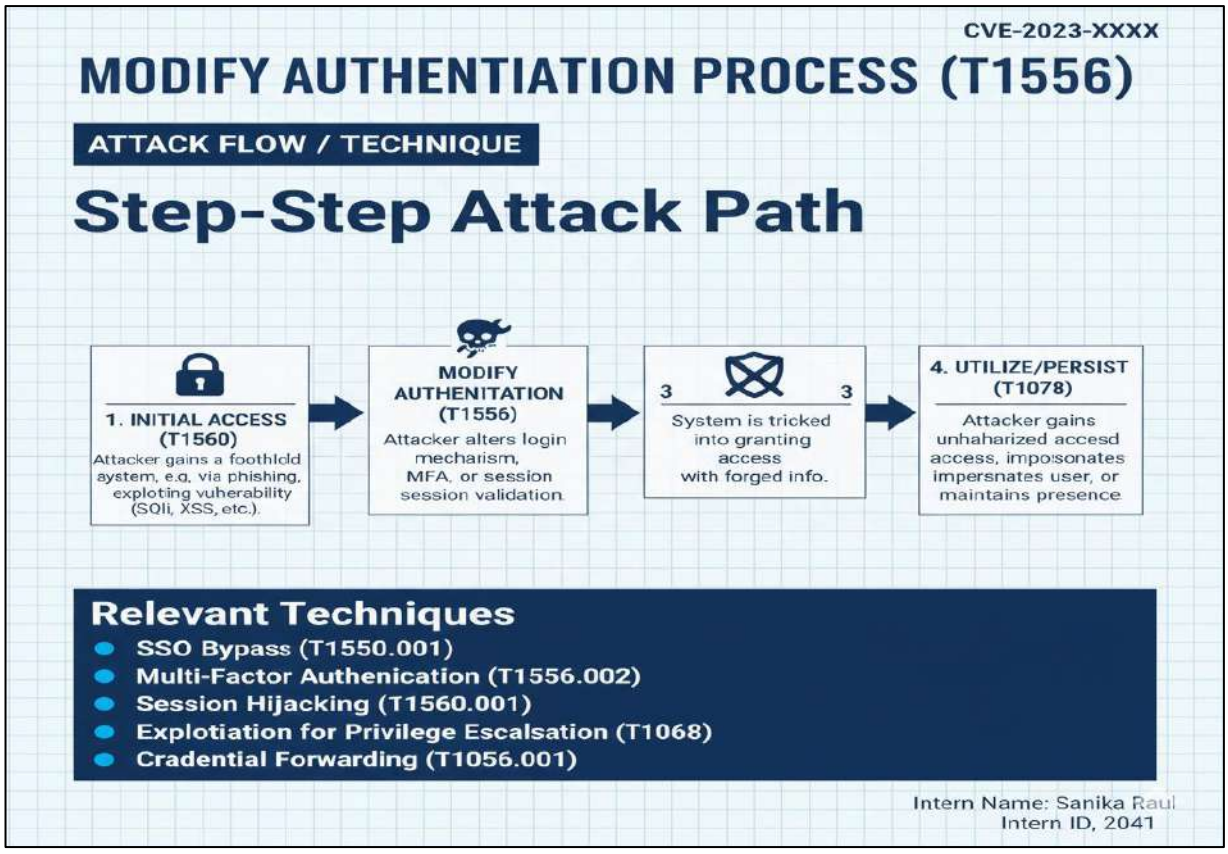
Modify Authentication Process (T1556)

Overview



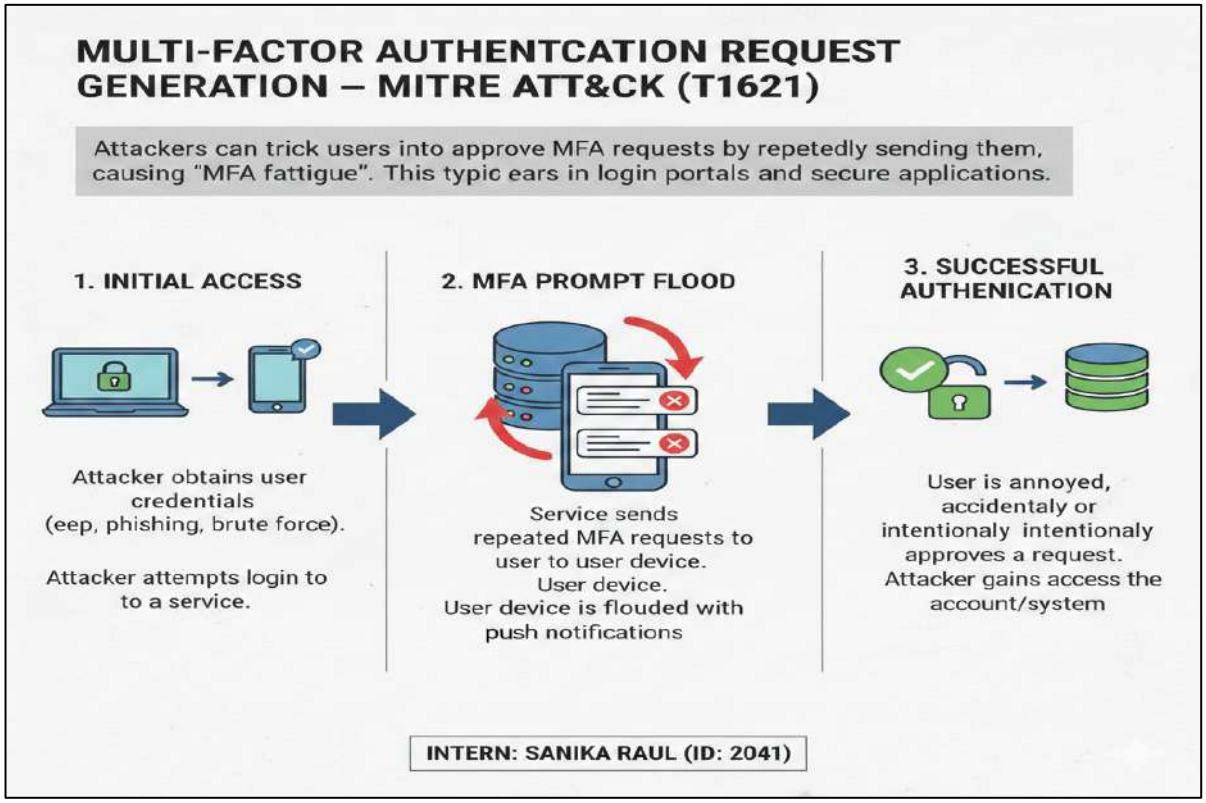
Technical details

Attack flow / technique

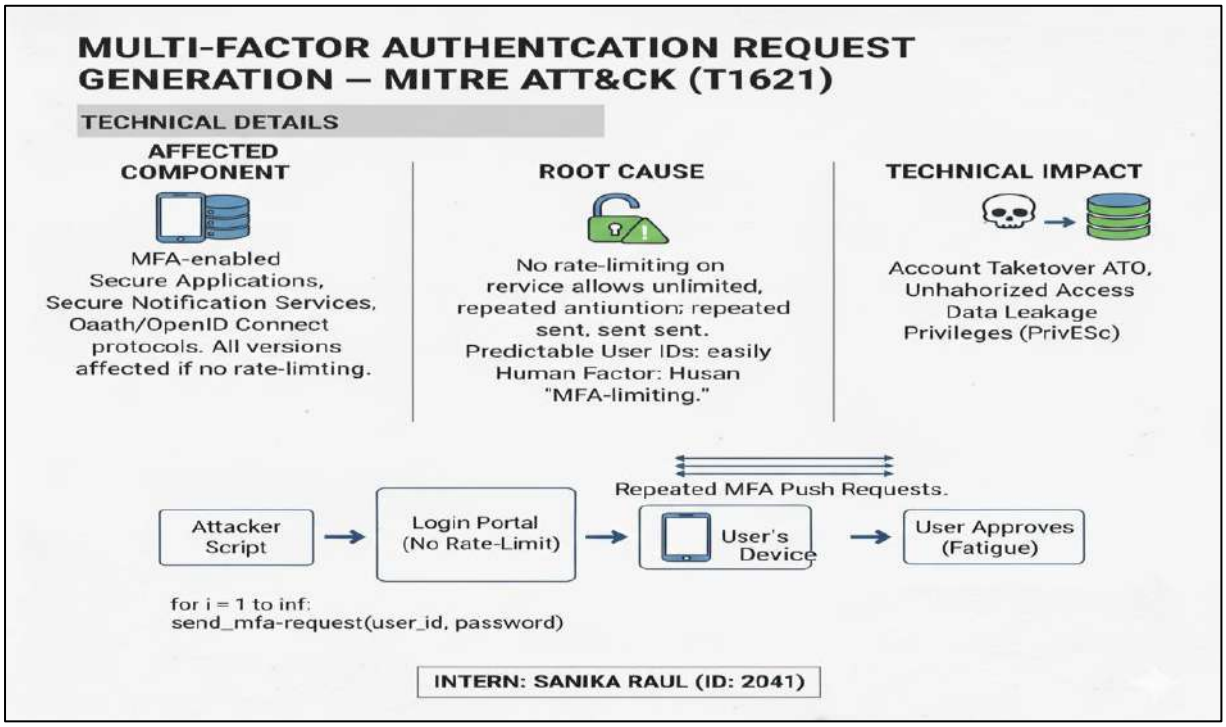


Multi-Factor Authentication Request Generation (T1621)

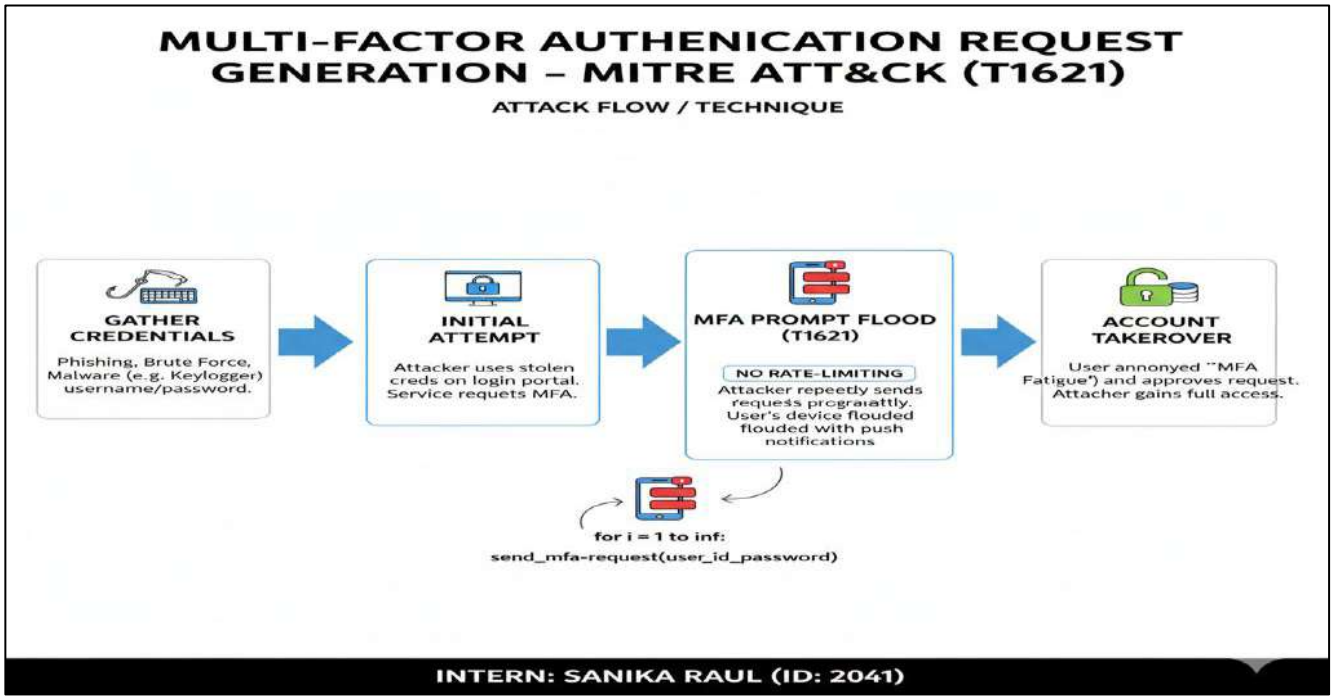
Overview



Technical details

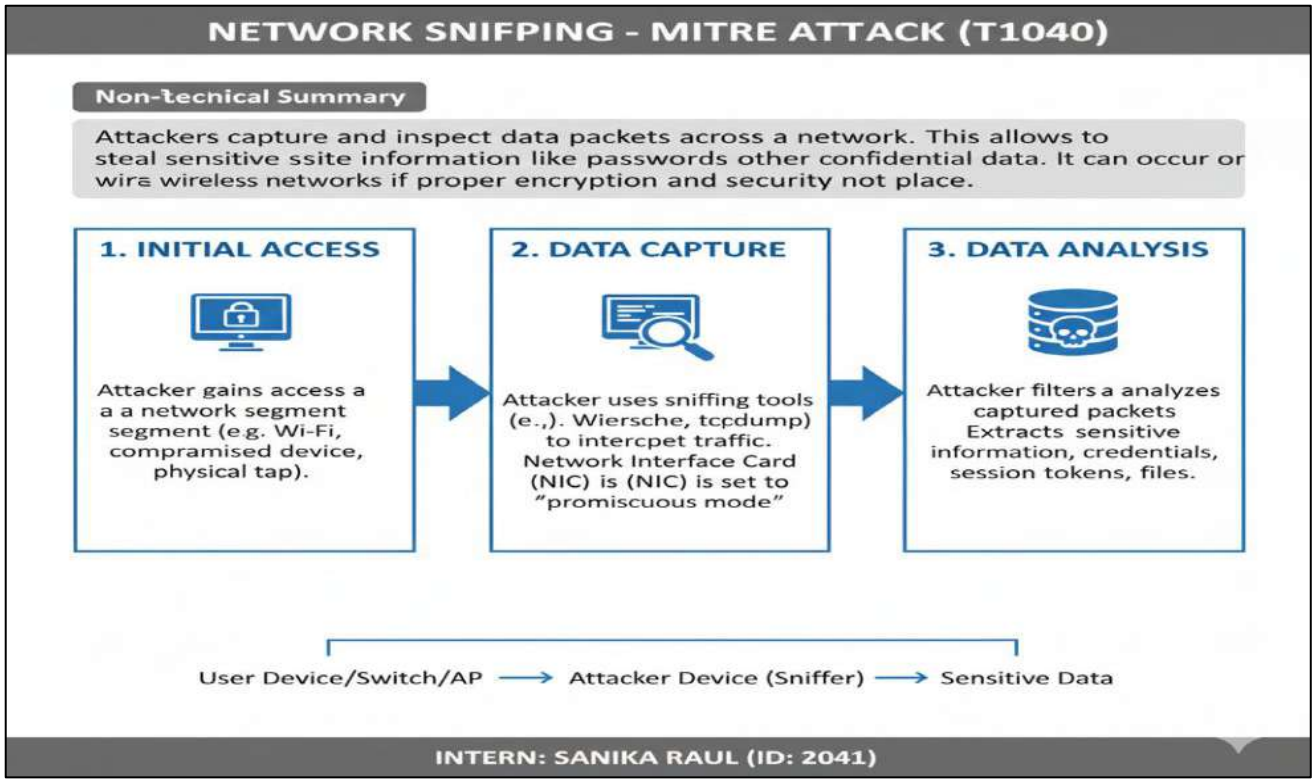


Attack flow / technique

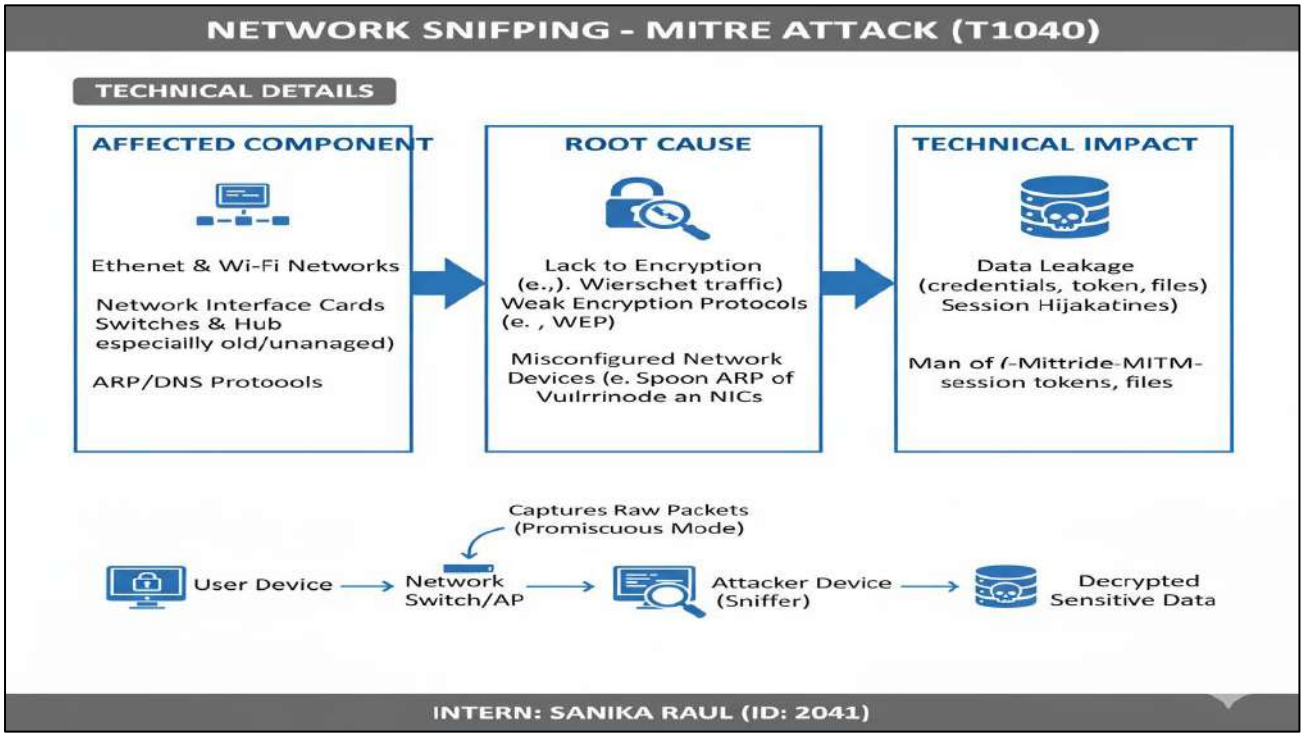


Network Sniffing (T1040)

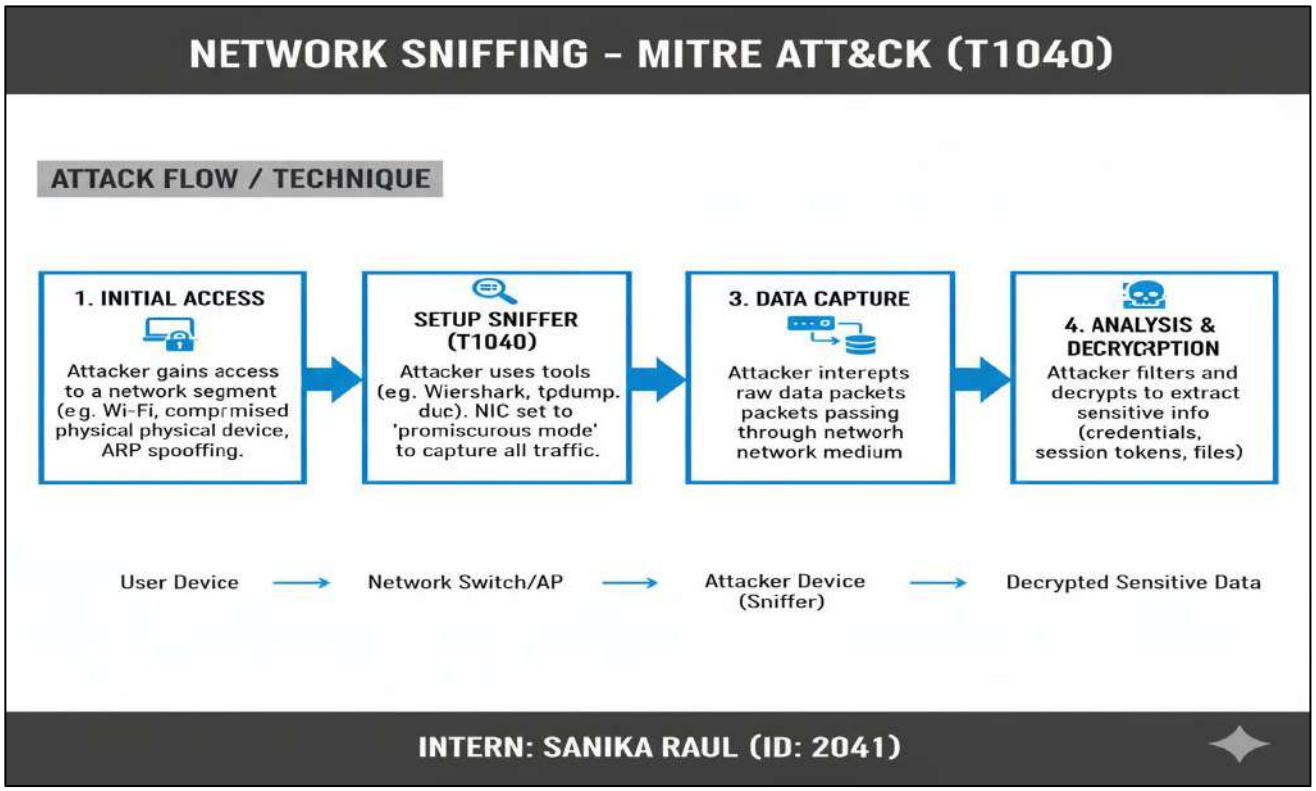
overview



Technical details

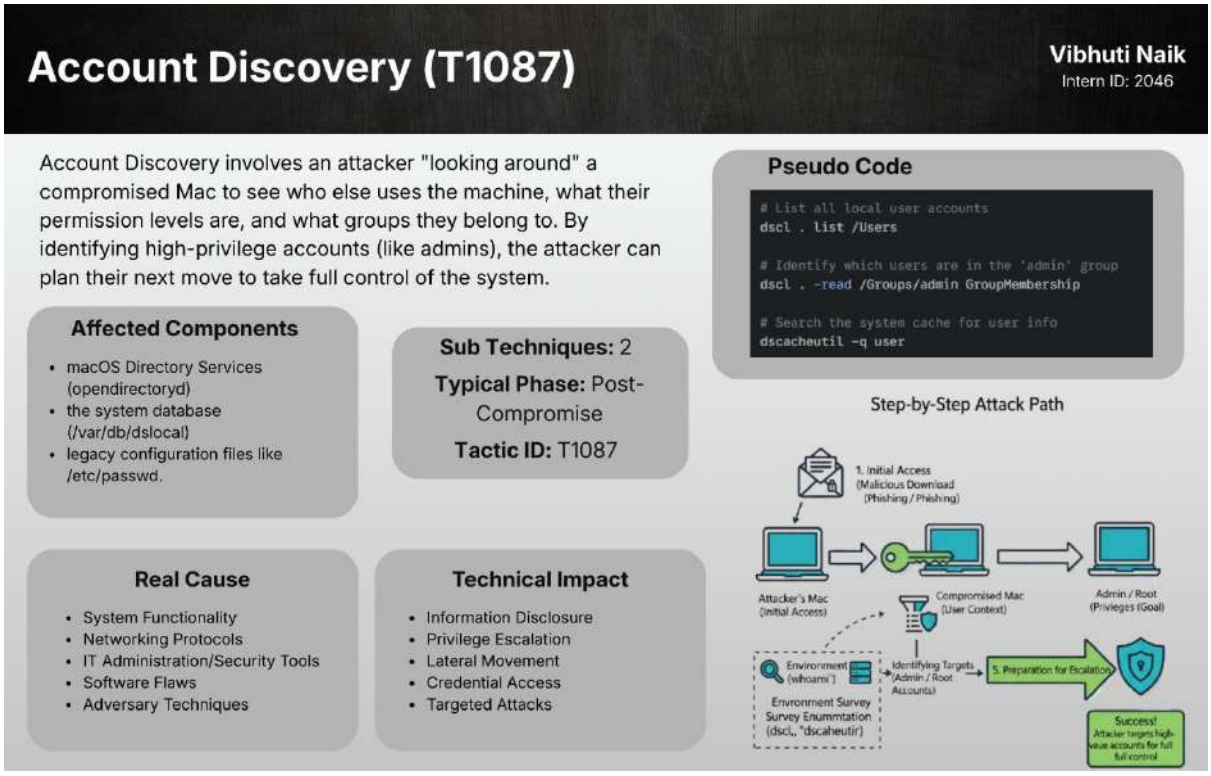


Attack flow / technique



Discovery

Account Discovery



Application Window Discovery

Application Window Discovery (T1010)

Vibhuti Naik
Intern ID: 2046

An attacker who has already breached a Mac scans the graphical interface to see which applications are currently open and what their window titles are. This "situational awareness" helps them decide if the machine belongs to a high-value target (like a CFO) and which apps to target for screen capturing or data theft.

Pseudo Code

```
-- Tells the system to report every open window name
tell application "System Events"
    get name of every window of (every process whose background only is false)
end tell
```

Affected Components

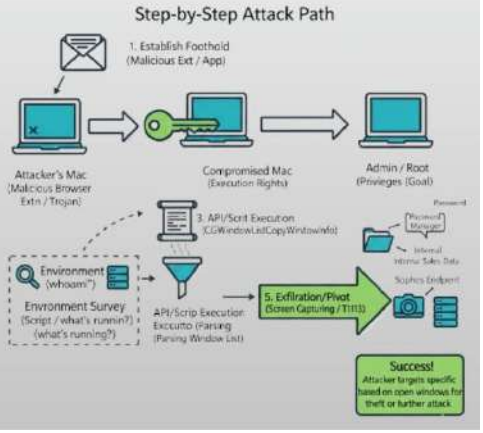
- App/Framework: Core Graphics (Quartz), AppKit.
- APIs: CGWindowListCopyWindowInfo, NSRunningApplication.
- Scripting: AppleScript (osascript).

Sub Techniques: 0

Typical Phase: Post-Compromise

Tactic ID: T1010

Step-by-Step Attack Path



Real Cause

- Abuse of Native System Features
- Bypassing Security Measures
- Living off the Land (LotL) Binaries
- User Override

Technical Impact

- Sensitive Information Gathering
- Security Software Discovery
- Targeted Attacks and Deception
- Abuse of Native Features
- Facilitating Sandbox Escape

Browser Information Discovery

Browser Information Discovery (T1217)

Vibhuti Naik
Intern ID: 2046

An attacker who has already breached a Mac scans the system for browser-related files. These files are goldmines of information; bookmarks can reveal internal company dashboards, history can reveal sensitive interests or sites, and cookies can sometimes be stolen to hijack active sessions.

Pseudo Code

```
# Locate Safari bookmarks (stored as a property list)
find ~/Library/Safari -name "Bookmarks.plist"

# Locate Chrome history (a SQLite database)
ls ~/Library/Application\ Support/Google/Chrome/Default/History

# Locate Firefox 'places' database (history + bookmarks)
find ~/Library/Application\ Support/Firefox/Profiles -name "places.sqlite"
```

Affected Components

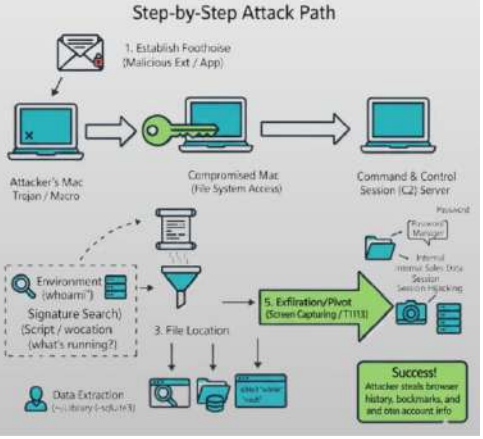
- Browsers: Safari, Google Chrome, Mozilla Firefox, Microsoft Edge, Brave, and Arc.
- Storage Files: Bookmarks (JSON/Plist), History (SQLite), Cookies (SQLite/Binary), and Login Data (encrypted databases).

Sub Techniques: 0

Typical Phase: Post-Compromise

Tactic ID: T1217

Step-by-Step Attack Path



Real Cause

- System Design and Data Storage
- Malware and Browser Hijackers
- Exploitation of Vulnerabilities
- Abuse of System Features

Technical Impact

- Credential Theft
- System Information Disclosure
- Man-in-the-Middle (MitM) Attacks
- System Information Disclosure
- Follow-on Attacks
- Persistent Access

Debugger Evasion

Debugger Evasion (T1622)

Vibhuti Naik
Intern ID: 2046

This is a "hide-and-seek" defense mechanism used by malware. When a security researcher or an automated system tries to open a suspicious file in a "debugger", the malware detects it is being watched. It then changes its behavior-either by crashing, displaying fake messages, or simply refusing to run-to prevent the researcher from figuring out how it works.

Affected Components

- System Calls: ptrace (Process Trace).
- Libraries: System.framework, Kernel (Mach-O headers).
- Security Tools: lldb (the standard macOS debugger), GDB, and sandboxes.

Sub Techniques: 0

Typical Phase: Post-Compromise

Tactic ID: T1622

Real Cause

- Process and Environment Checks
- Timing Checks
- Exception Handling
- Obfuscation
- Direct System Calls
- Lesser Known Languages

Technical Impact

- Concealment of Malicious Logic
- Bypass of Security Controls
- Persistent Threats
- Defence Evasion
- Detection Bypassing

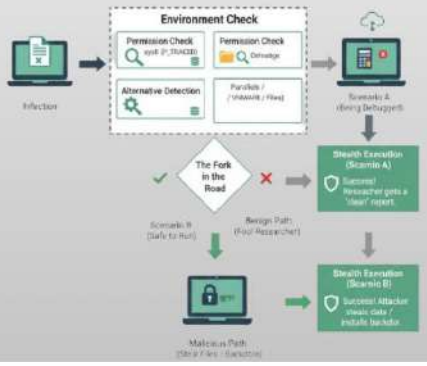
Pseudo Code

```
#include <sys/types.h>
#include <sys/ptrace.h>

int main() {
    // This tells the macOS kernel: "Do not let any debugger attach to
    // If a debugger is already attached, the program will immediately
    ptrace(PT_DENY_ATTACH, 0, 0, 0);

    // If we reach here, no debugger is attached.
    execute_malware_payload();
    return 0;
}
```

Step-by-Step Attack Path



Device Driver Discovery

Device Driver Discovery (T1652)

Vibhuti Naik
Intern ID: 2046

This is a technique where an attacker scans the Mac to see what hardware and "device drivers" (software that talks to hardware) are installed. By knowing which drivers are present—such as those for specialized audio interfaces, VPNs, or printers—the attacker can look for specific vulnerabilities in those drivers to crash the system or gain deep, "Kernel-level" control.

Affected Components

- Subsystem: I/O Kit (macOS framework for hardware/drivers).
- Components: Kernel Extensions (.kext) and System Extensions (.systemextension).
- Utilities: system_profiler, kextstat, ioreg.

Sub Techniques: 0

Typical Phase: Post-Compromise

Tactic ID: T1652

Real Cause

- Hardware Communication
- System Boot Process
- Driver Matching
- Enumerate Local Device Drivers
- Flaws in specific Drivers
- macOS Architecture

Technical Impact

- Vulnerability Identification
- Privilege Escalations
- Security Software Evasion
- Malicious Driver Deployment
- Denial of Service
- Persistence and Evasion of Controls

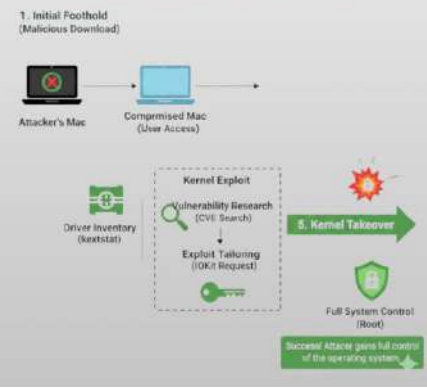
Pseudo Code

```
# List all loaded kernel extensions, filtering for non-Apple developers
kextstat | grep -v "com.apple"

# Use the I/O Kit Registry explorer to find specific hardware details
ioreg -l -p IODeviceTree

# Get a detailed report of all installed software drivers
system_profiler SPExtensionsDataType
```

Step - Step Attack Path



File and Directory Discovery

File & Directory Discovery (T1083)

Vibhuti Naik
Intern ID: 2046

Once an attacker gains access to a Mac, they don't always know where the "treasure" is. They use this technique to browse through folders—much like a thief searching drawers—to find things like saved passwords, browser history, proprietary source code, or internal company documents.

Affected Components

- **Component:** macOS Terminal / Shell (zsh/bash).
- **Impacted Versions:** All versions of macOS (Sequoia, Sonoma, Ventura, etc.).
- **Protocols/Libraries:** Standard POSIX APIs and System Utilities.

Sub Techniques: 0

Typical Phase: Post-Compromise

Tactic ID: T1083

Pseudo Code

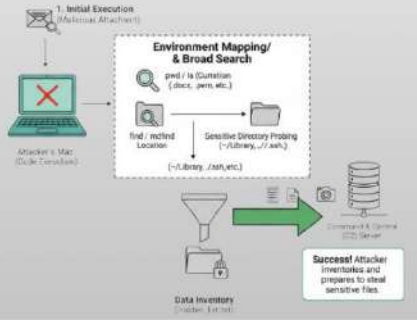
```
# 1. Who am I and where am I?
whoami && pwd

# 2. List all files, including hidden ones
ls -laR /Users/$(whoami)/Documents

# 3. Search for sensitive file types (passwords, keys)
find / -name "*.pdf" -or -name "*.txt" | grep -i "password"

# 4. Check for mounted network drives
df -h
```

Step-by-Step Attack Path



Real Cause

- Weak access controls
- No file monitoring
- Excessive permissions
- No encryption
- Insecure Design

Technical Impact

- Targeted Data Collection
- Credential Theft
- Privilege Escalation
- Lateral Movement
- File Modification & Corruption
- Establishing Persistence

Process Discovery

Process Discovery (T1057)

Vibhuti Naik
Intern ID: 2046

Once an attacker gets inside a Mac, they need to know what else is running. They check the "Process List" to see if there are antivirus tools they should avoid, or if specific apps like Slack, Chrome, or Microsoft Excel are open. This helps them understand the user's role and plan their next step (like stealing passwords).

Affected Components

- System Utilities: ps, top, pgrep.
- APIs: libproc (Process Information Library), sysctl.
- Frameworks: AppleScript (System Events).

Sub Techniques: 0

Typical Phase: Post-Compromise

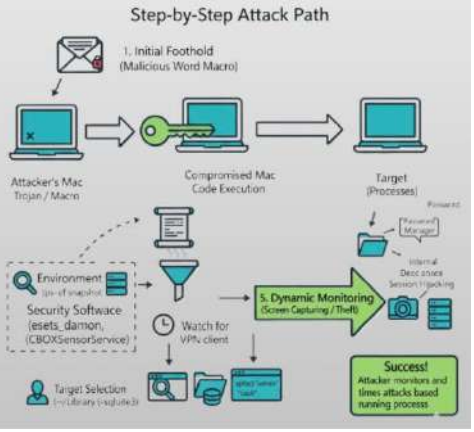
Tactic ID: T1057

Pseudo Code

```
# List all running processes with user and command details
ps aux

# Specifically look for security software (e.g., CrowdStrike or SentinelOne)
ps aux | grep -E "sensor|agent|falcon"
```

Step-by-Step Attack Path



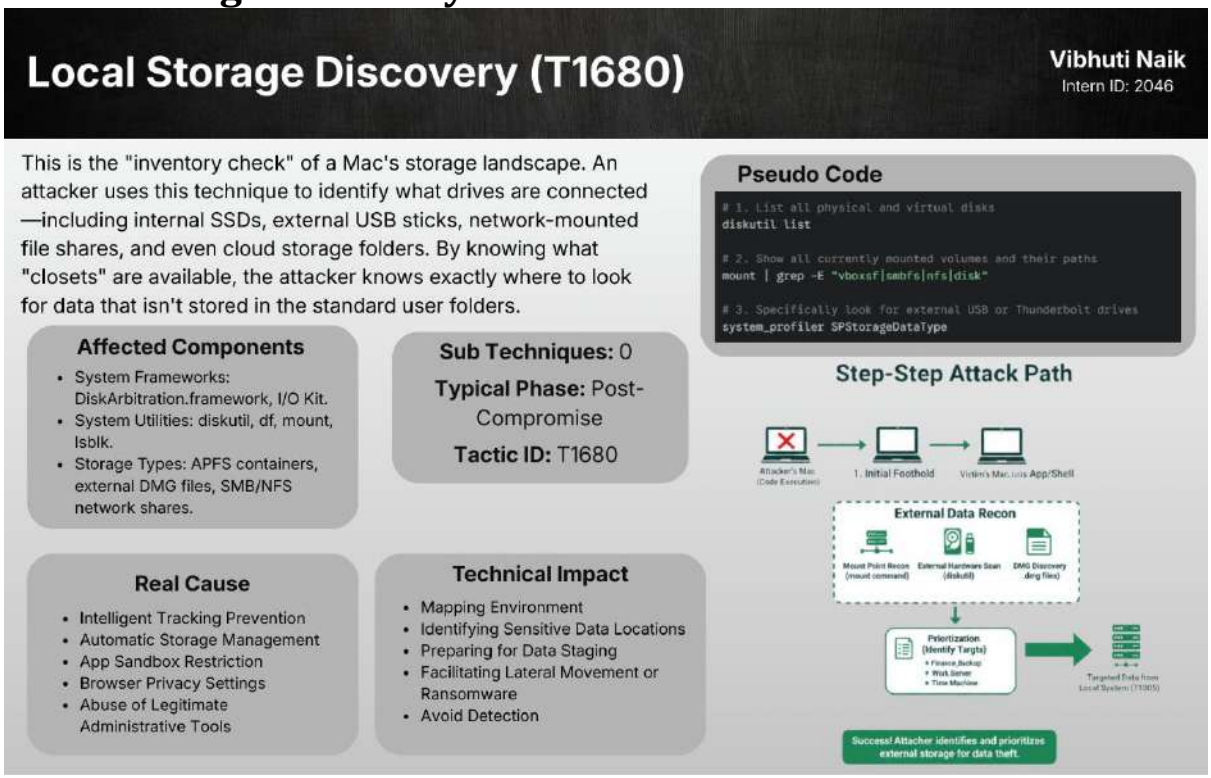
Real Cause

- Legitimate System Monitoring
- Security Tools/Audits
- Misconfigured Monitoring Software
- Malware Activity

Technical Impact

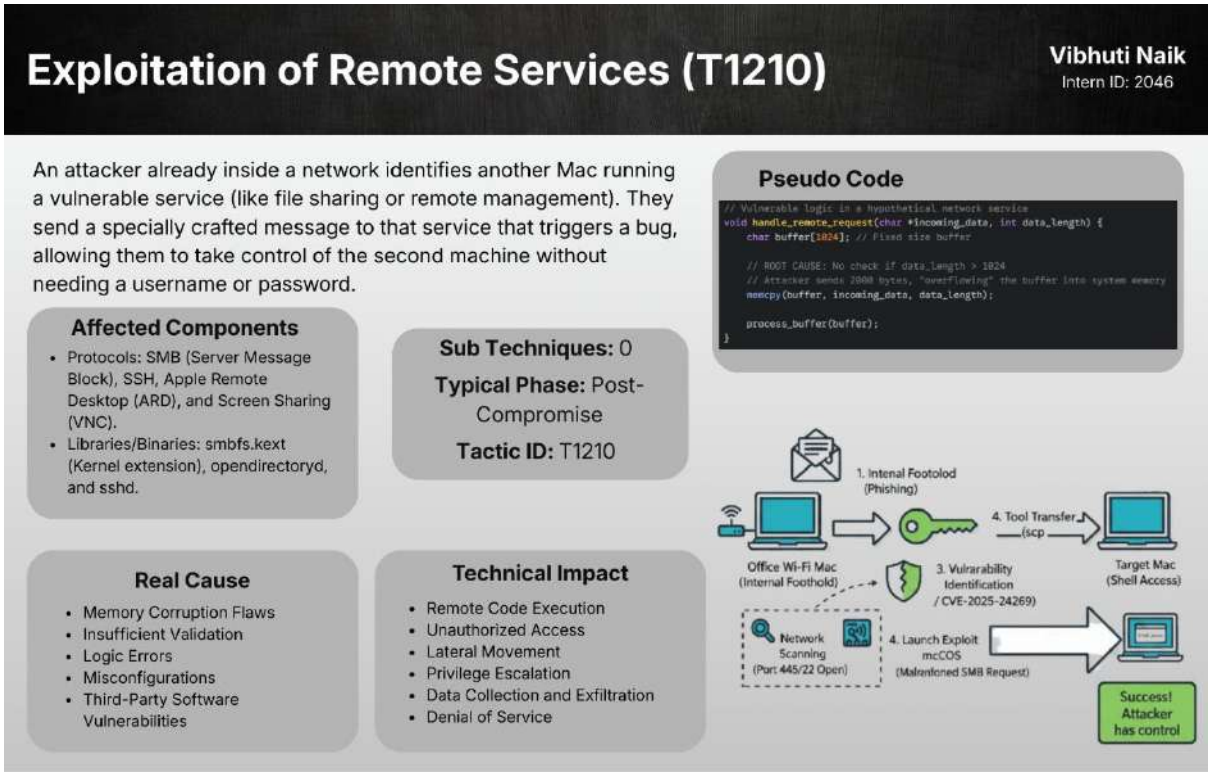
- System Resource Consumption
- Data Privacy and Security
- Network Usage
- Compatibility and Integration
- Automation Blueprint

Local Storage Discovery



Lateral Movement

Exploitation of Remote Services



Internal Spearphishing

Internal Spearphishing (T1534)

Vibhuti Naik
Intern ID: 2046

An attacker who has already stolen one employee's login uses that person's "trusted" identity to send malicious links or files to other employees. Because the message comes from a real colleague or the internal IT department, the new victims are much more likely to click and infect their own Macs.

Affected Components

- Apps: Apple Mail, Microsoft Outlook, Slack, Microsoft Teams, Zoom.
- System Services: macOS Transparency, Consent, and Control (TCC) prompts, which the victim might be tricked into approving.

Sub Techniques: 0

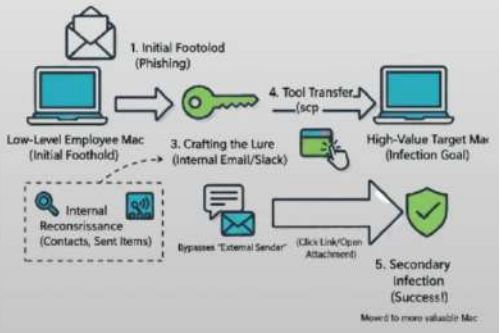
Typical Phase: Post-Compromise

Tactic ID: T1534

Pseudo Code

```
# Attacker script running on Compromised Mac A
# It searches the internal directory and sends an automated 'phish'
# disguised as a "New Security Policy" PDF.

send_internal_mail(
    sender="legit-finance-peer@company.com",
    recipient="it-admin@company.com",
    subject="Urgent: Updated Q1 Payroll Doc",
    attachment="Payroll_Update.dmg"  # Actually a malicious disk image
)
```



Real Cause

- External Spearphishing
- Malware Installation
- Credential Theft
- Social Engineering
- User Execution

Technical Impact

- Remote Code Execution
- Unauthorized Access
- Privilege Escalation
- Data Collection
- Denial of Service
- System Instability

Lateral Tool Transfer

Lateral Tool Transfer (T1570)

Vibhuti Naik
Intern ID: 2046

This is the digital equivalent of an intruder entering a building through one window and then moving their bag of burglary tools into other rooms. On macOS, attackers use built-in file transfer utilities to spread malware, backdoors, or "living-off-the-land" scripts from an initially infected Mac to others on the same Wi-Fi.

Affected Components

- Native Utilities: scp (Secure Copy), rsync, sftp, curl.
- Protocols: SSH (Secure Shell), SMB (Server Message Block/File Sharing), Apple Archive.

Sub Techniques: 0

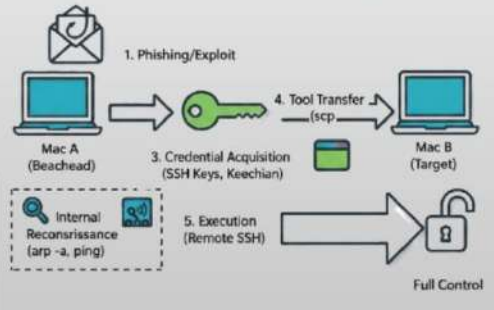
Typical Phase: Post-Compromise

Tactic ID: T1570

Pseudo Code

```
# Using scp to push a malicious script to a coworker's Mac
scp ./malicious_tool user@192.168.1.50:/tmp/

# Using rsync to sync an entire folder of hacking tools to a server
rsync -avz ./tools_folder/ admin@internal-server.local:/Users/Shared/
```



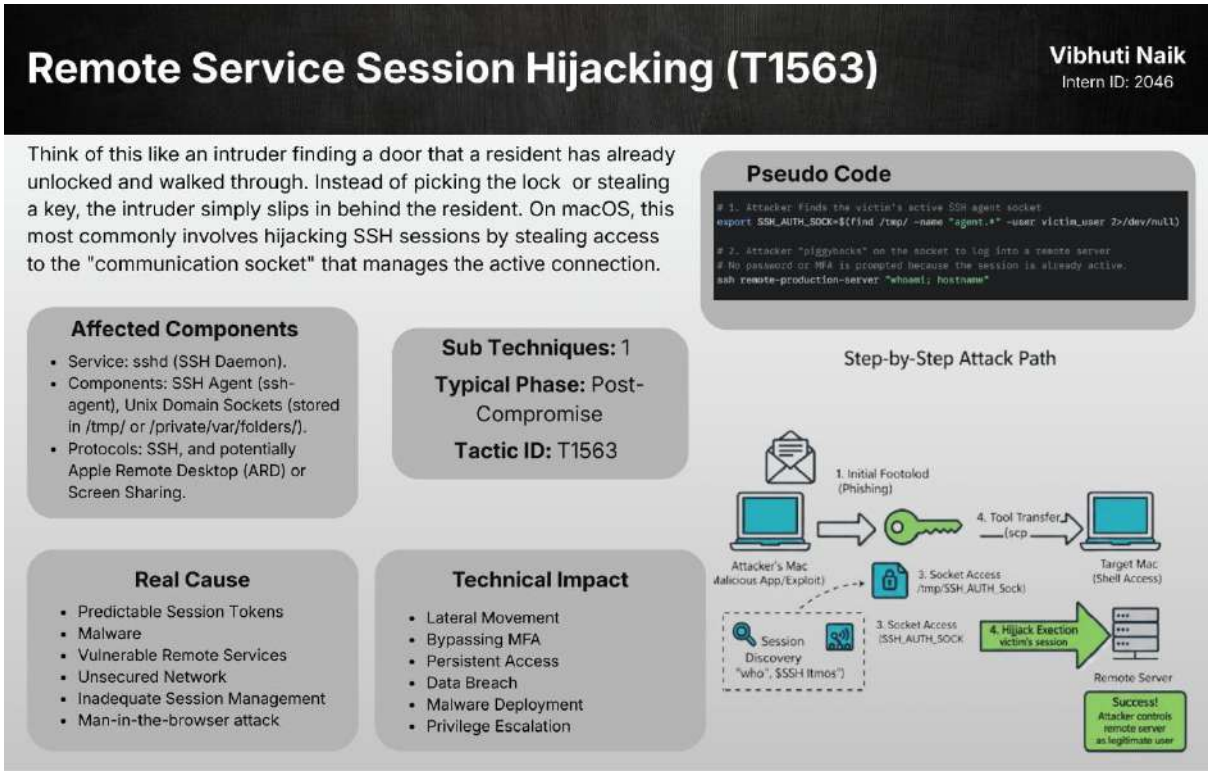
Real Cause

- Credential Theft and Abuse
- Misconfiguration of Remote Services
- Abuse of Native Tools
- Software Vulnerabilities
- Inadequate Access

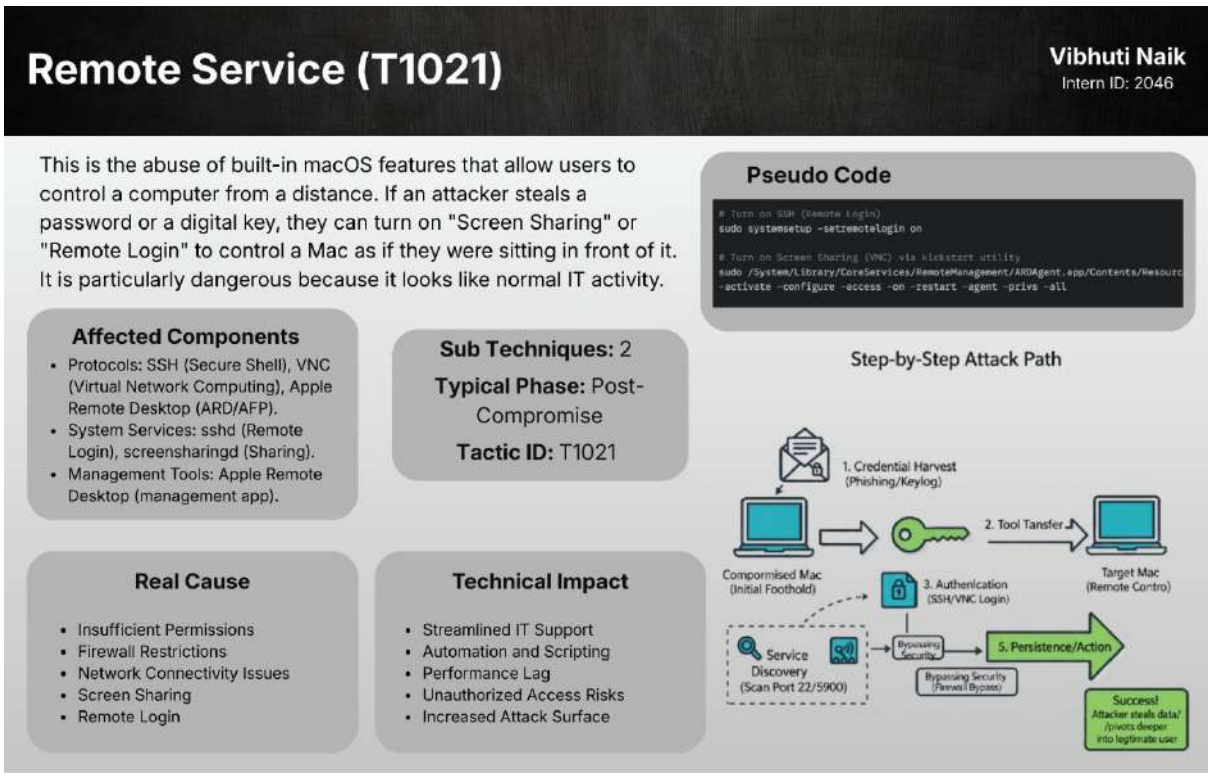
Technical Impact

- Widespread Network Infiltration
- Persistence and Evasion
- Credential Theft
- Data Exfiltration
- Automation of Attacks
- Privilege Escalation

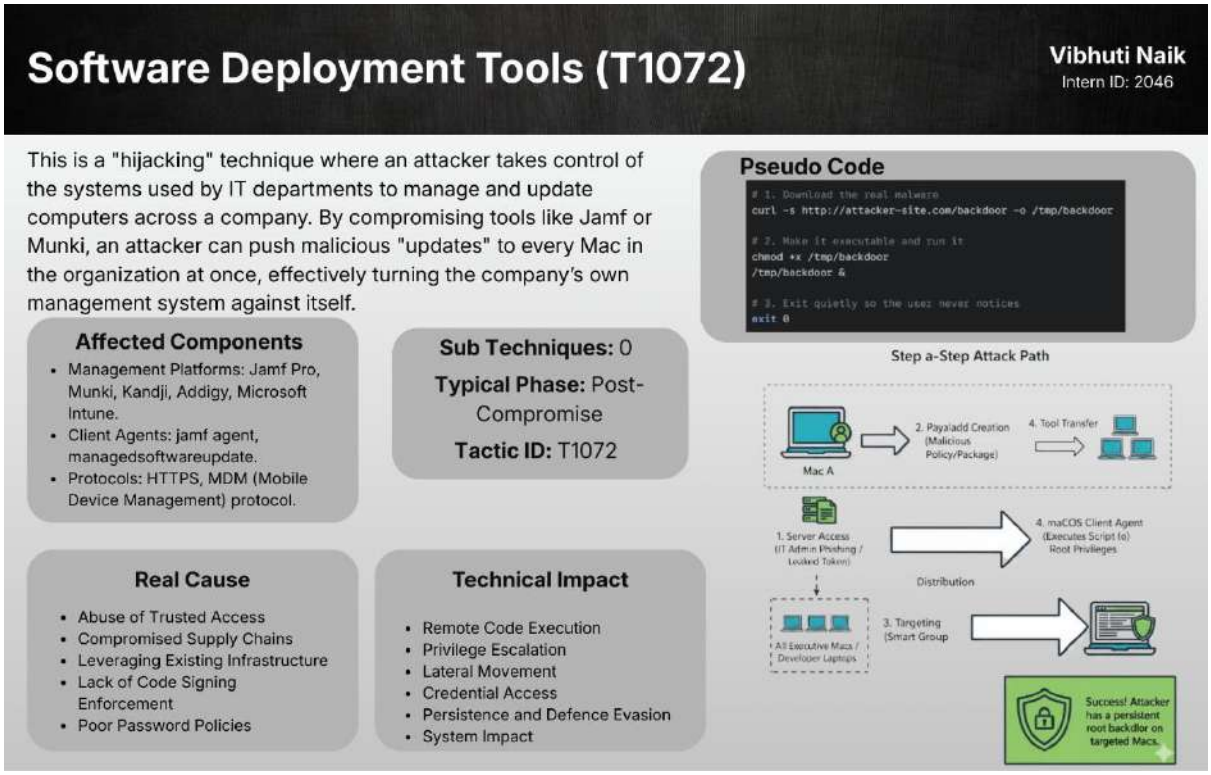
Remote Service Session Hijacking (1)



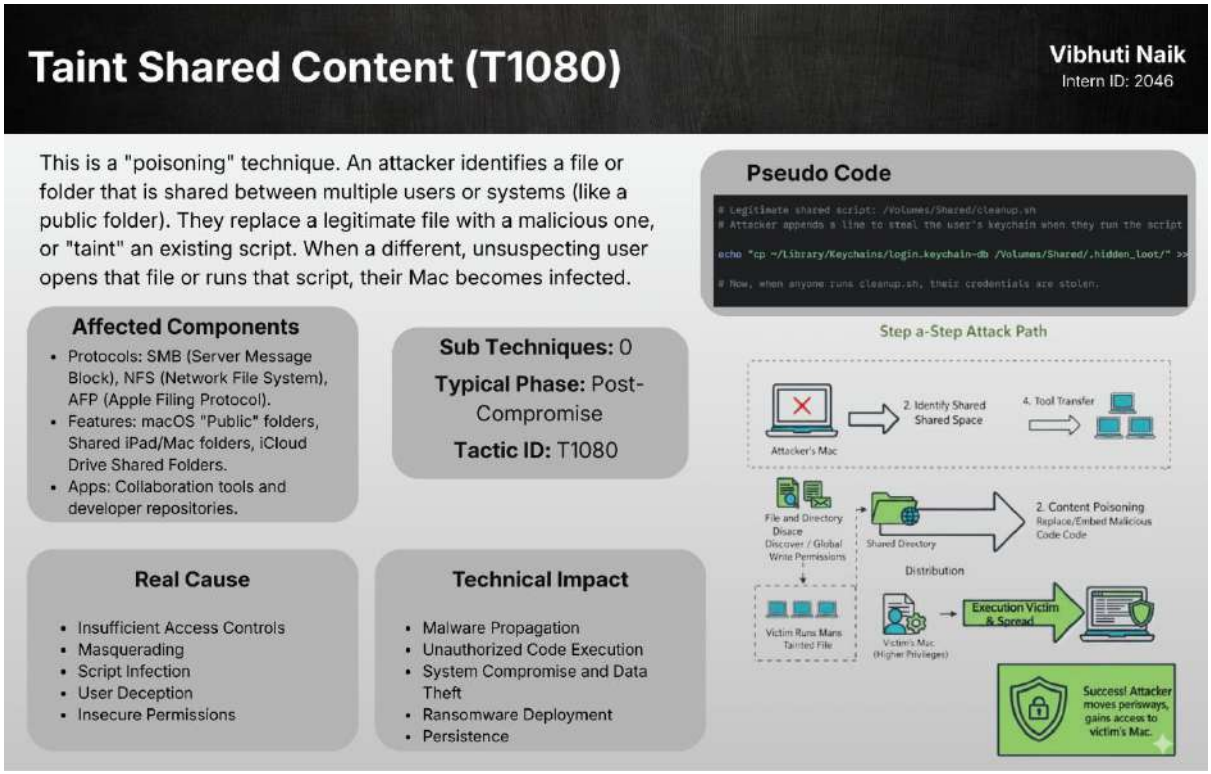
Remote Services (2)



Software Deployment Tools



Taint Shared Content



Collection

Adversary-in-the-Middle (2)

Adversary-in-the-Middle (T1557)

Vibhuti Naik
Intern ID: 2046

An attacker positions themselves between a Mac and the service it is communicating with. By "sitting in the middle," the attacker can transparently intercept, read, and even modify the data being sent back and forth without the user realizing it. On macOS, this often happens on public Wi-Fi or through malicious local network.

Affected Components

- Protocols: HTTP/HTTPS, DNS, SMB, and LDAP.
- System Services: mDNSResponder (Bonjour/multicast DNS), networkd.
- Apps: Any browser (Safari, Chrome) or app that communicates over the network without strict certificate pinning.

Sub Techniques: 2

Typical Phase: Post-Compromise

Tactic ID: T1557

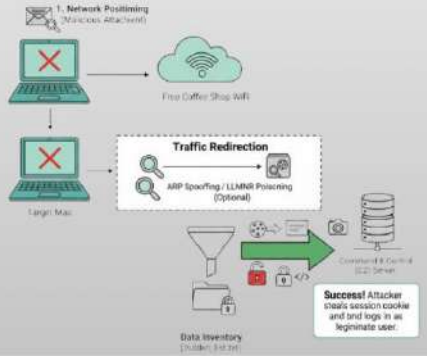
Pseudo Code

```
# Attacker tells the Mac: "I am the Router"
arpspoof -i en0 -t [Target_Mac_IP] [Router_IP]

# Attacker tells the Router: "I am the Target Mac"
arpspoof -i en0 -t [Router_IP] [Target_Mac_IP]

# All traffic now flows through the Attacker's machine
```

Step-by-Step Attack Path



Real Cause

- Phishing and Malware
- Unsecured Wi-Fi Network
- Network Protocol Exploitation
- Software Vulnerabilities
- DNS Spoofing
- ARP Poisoning

Technical Impact

- MFA Bypass
- Credential Harvesting
- Session Hijacking
- Eavesdropping
- Data Manipulation
- Malware Injection

Archive Collected Data (3)

Archive Collected Data (T1560)

Vibhuti Naik
Intern ID: 2046

This is a "packaging" stage of a digital robbery. After an attacker has found sensitive files (like photos, documents, or passwords) on a Mac, they compress them into a single file (like a .zip or .tar). This makes the data smaller and easier to steal quickly without triggering network alarms that look for many small file transfers.

Affected Components

- Native Utilities: /usr/bin/zip, /usr/bin/tar, /usr/bin/ditto.
- Third-Party Apps: 7-Zip, Keka, or custom Python/Go scripts.
- Libraries: libarchive, zlib.

Sub Techniques: 3

Typical Phase: Post-Compromise

Tactic ID: T1560

Pseudo Code

```
# Attacker uses the built-in 'zip' tool with a password (-P)
# to bundle the user's Documents and SSH keys into a hidden file.
zip -r -P "ComplexPassword123" ~/.hidden_archive.zip ~/Documents ~/.ssh

# Alternatively, using 'ditto' to create a compressed CPIO archive
ditto -c -k --sequesterRsrc --keepParent ~/Library/Keychains /tmp/backup.zip
```

Step-Step Attack Path

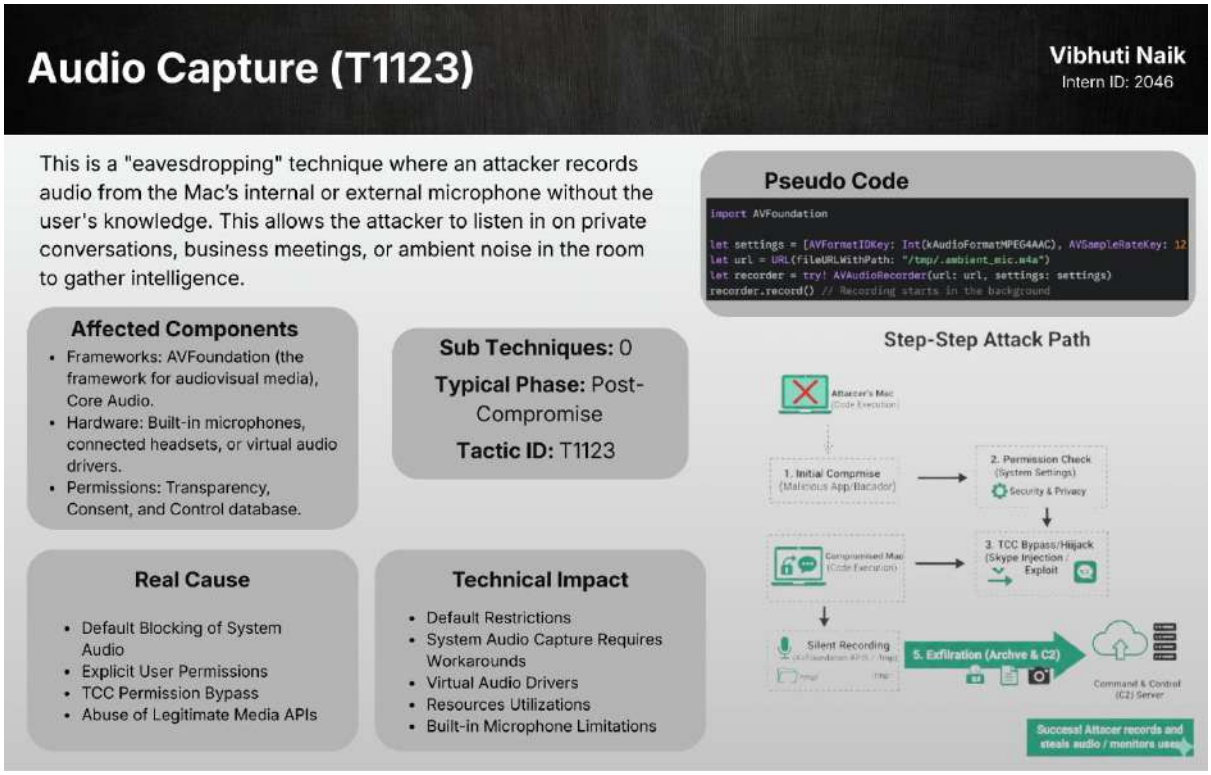
Real Cause

- System & App Cache
- Time Machine Local Snapshots
- System Log Files
- Disk Images and Plugins
- Sensitive Files
- Credential and Browser Data

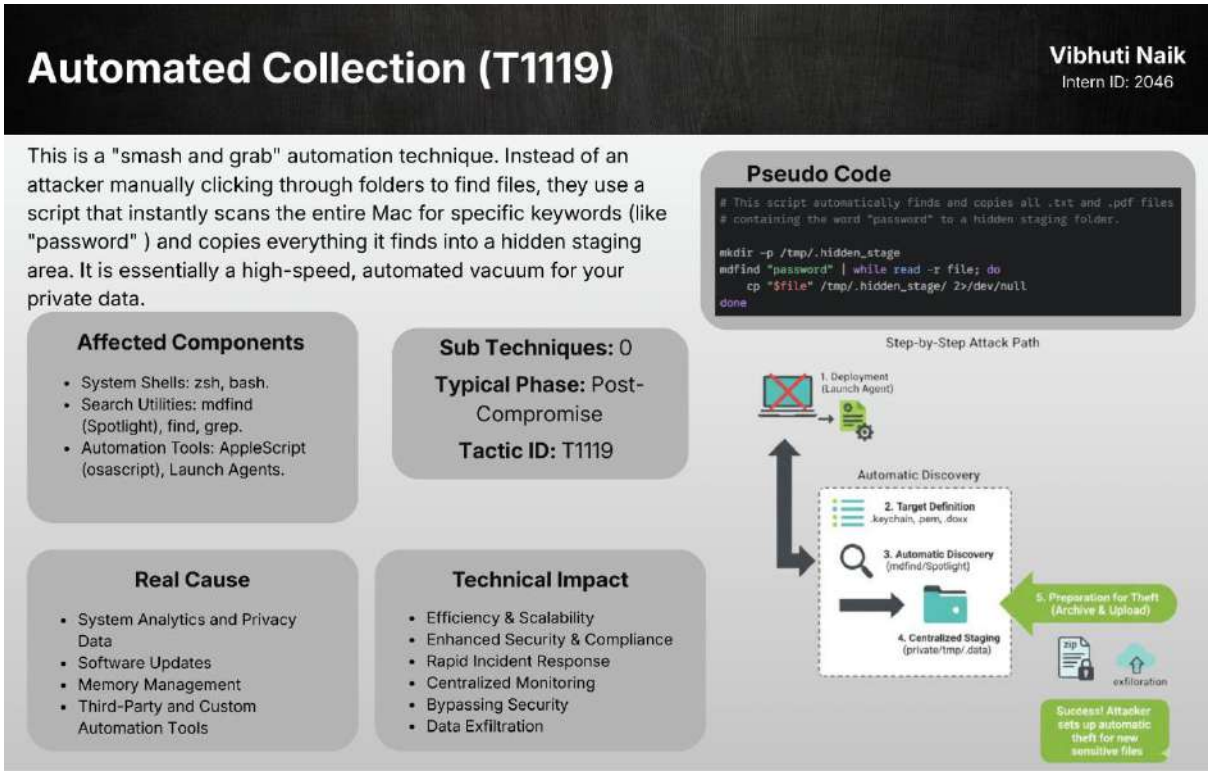
Technical Impact

- Improved System Performance
- Optimized Storage Management
- Enhanced Data Security
- Compliance and Integrity
- Quicker Backups
- Data Leaks

Audio Capture



Automated Collection



Clipboard Data

Clipboard Data (T1115)

Vibhuti Naik
Intern ID: 2046

This is a "digital eavesdropping" technique where an attacker monitors or steals the information you copy and paste. Since users frequently copy sensitive items like passwords from a manager, credit card numbers, or private messages, this provides an attacker with high-value data without needing to search through files

Affected Components

- System Service: pboard (The macOS pasteboard server).
- Frameworks: AppKit (NSPasteboard class).
- Features: Universal Clipboard (copying on iPhone, pasting on Mac).

Sub Techniques: 0

Typical Phase: Post-Compromise

Tactic ID: T1115

Real Cause

- Design for Interoperability
- historical Technical Debt
- User Productivity Needs
- Universal Clipboard Feature
- Privacy Prompts

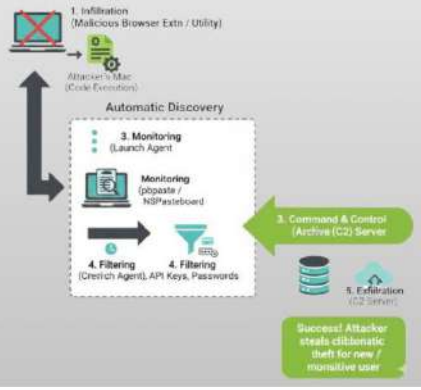
Technical Impact

- Universal Access for Applications
- Rich Data Formats
- Clipboards Snooping
- Pastejacking
- Malware Exploitations

Pseudo Code

```
# Simple bash script to monitor and log clipboard changes every 5 sec
last_clip=""
while true; do
    current_clip=$(pbpaste)
    if [ "$current_clip" != "$last_clip" ]; then
        echo "$(date): $current_clip" >> /tmp/.log_clipboard.txt
        last_clip=$current_clip
    fi
    sleep 5
done
```

Step-by-Step Attack Path



Data from Information Repositories (1)

Data from Information Repositories (T1213)

Vibhuti Naik
Intern ID: 2046

This is a "knowledge theft" technique. Once an attacker is inside a Mac, they look for internal company knowledge bases, wikis, or project management tools that are often left logged in or have cached data. These repositories are "gold mines" because they contain sensitive architectural diagrams, network passwords, onboarding documents, and private API keys.

Affected Components

- Apps: Confluence, Jira, Notion, Evernote, Slack (archived channels), Microsoft OneNote.
- Local Storage: Browser caches, ~/Library/Application Support/ folders, and local database files.
- Protocols: HTTP/HTTPS

Sub Techniques: 1

Typical Phase: Post-Compromise

Tactic ID: T1213

Real Cause

- Social Engineering
- TCC Evasion
- Bypassing Gatekeeper
- Use of Scripting
- Lack of Inherent Mitigation
- Insecure of Encryption

Technical Impact

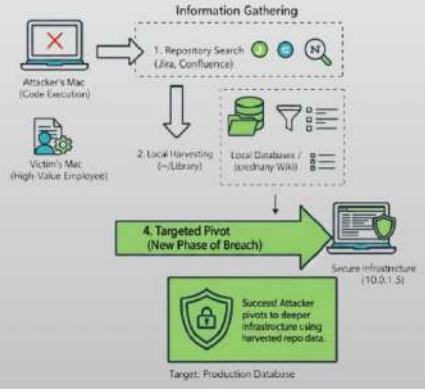
- Sensitive Data Exposure
- Credential Theft
- Transmitted Data Manipulations
- Privilege Escalation
- Defence Evasion
- Lateral Movement

Pseudo Code

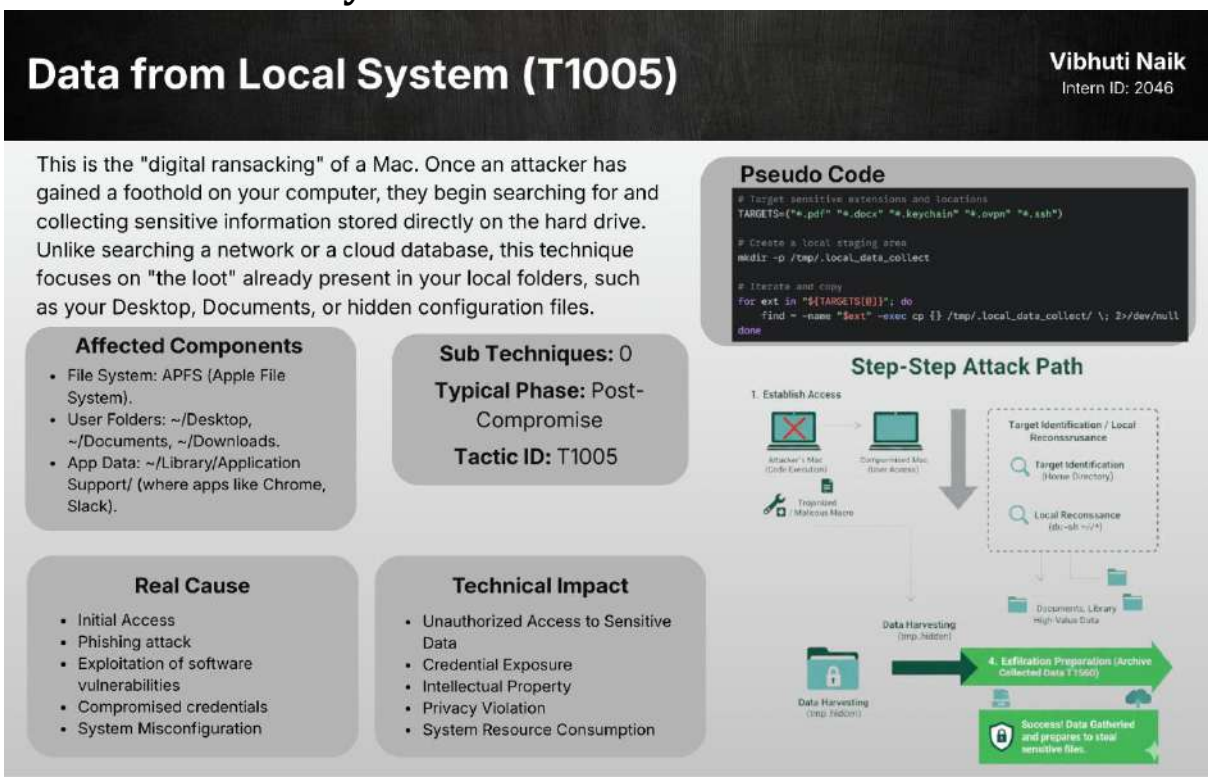
```
# Locate the local database for a repository app
DB_PATH="~/Library/Application Support/Notion/Local Storage/leveldb"

# An attacker uses a tool to dump the strings/text from these files
# to find keywords like "password" or "AWS_SECRET"
grep -rEi "password|secret|key|internal" "$DB_PATH"
```

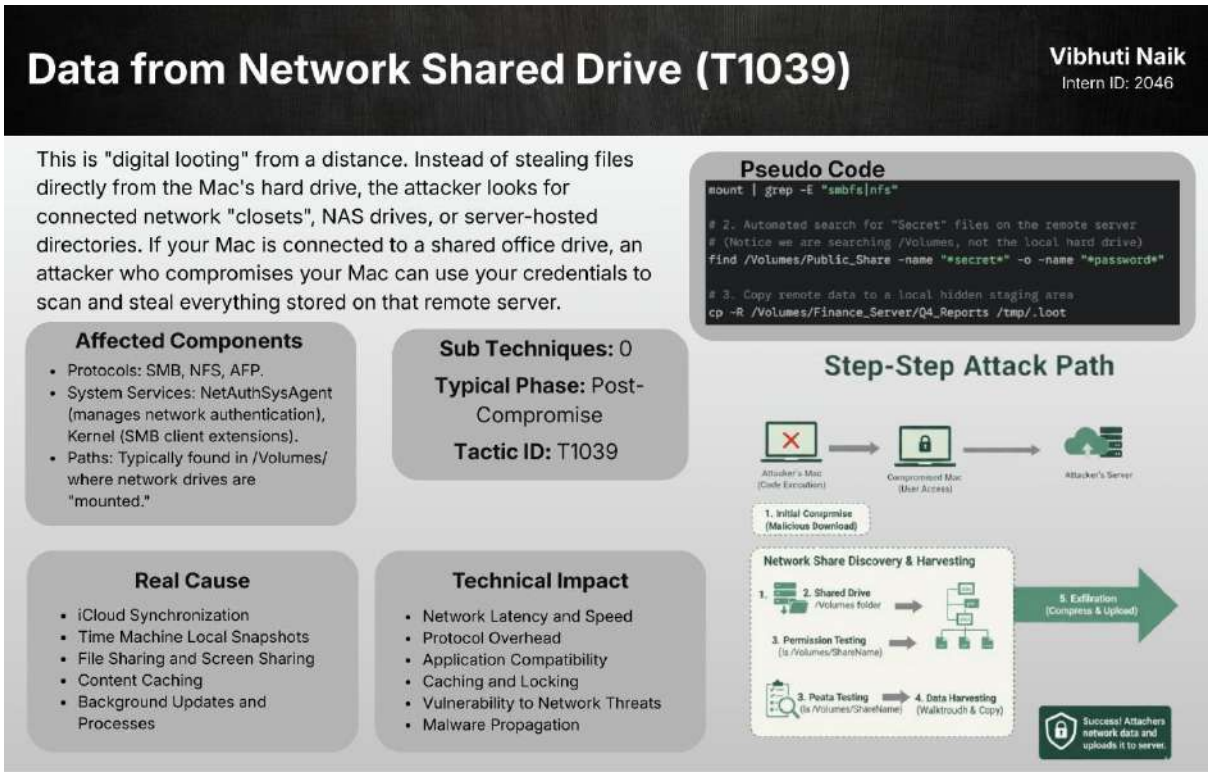
Step a-Step Attack Path



Data from Local System



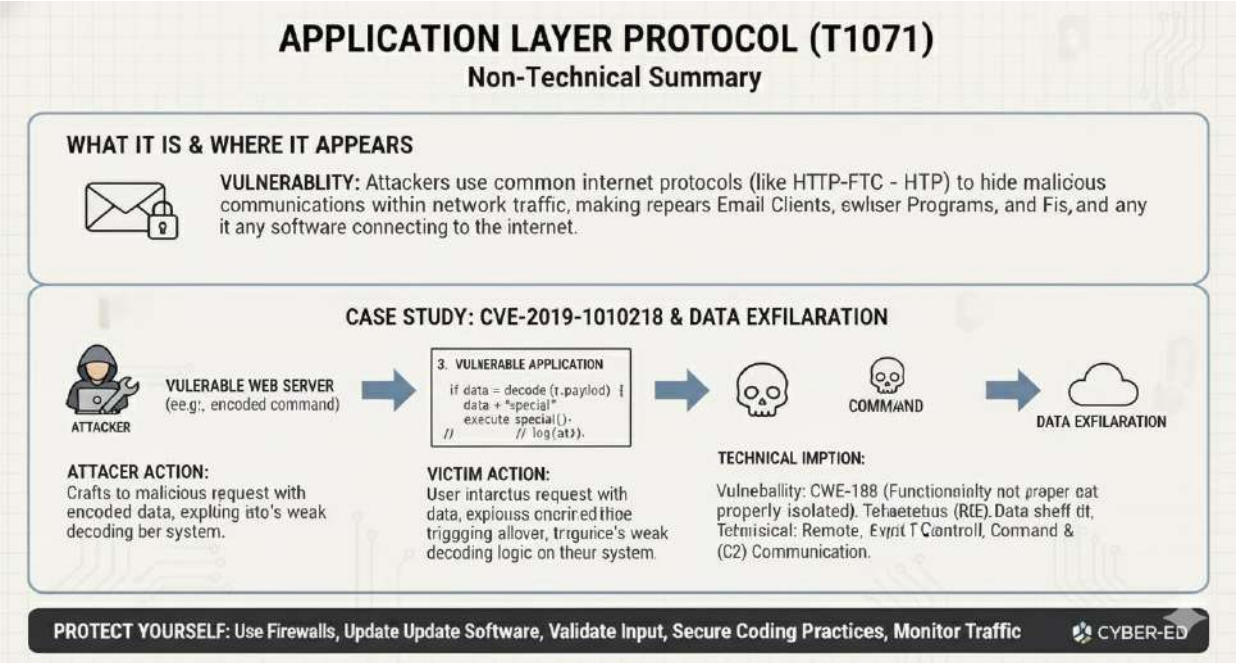
Data from Network Shared Drive



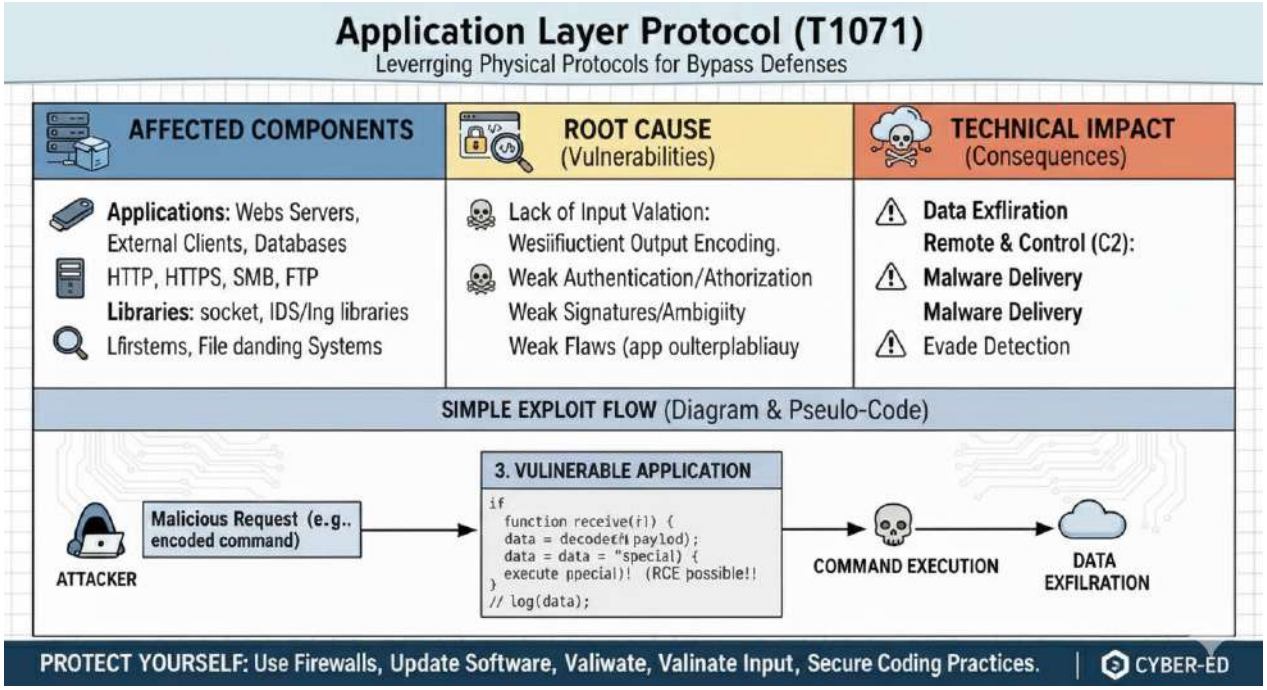
Command And Control :

Application Layer Protocol (T1071) :

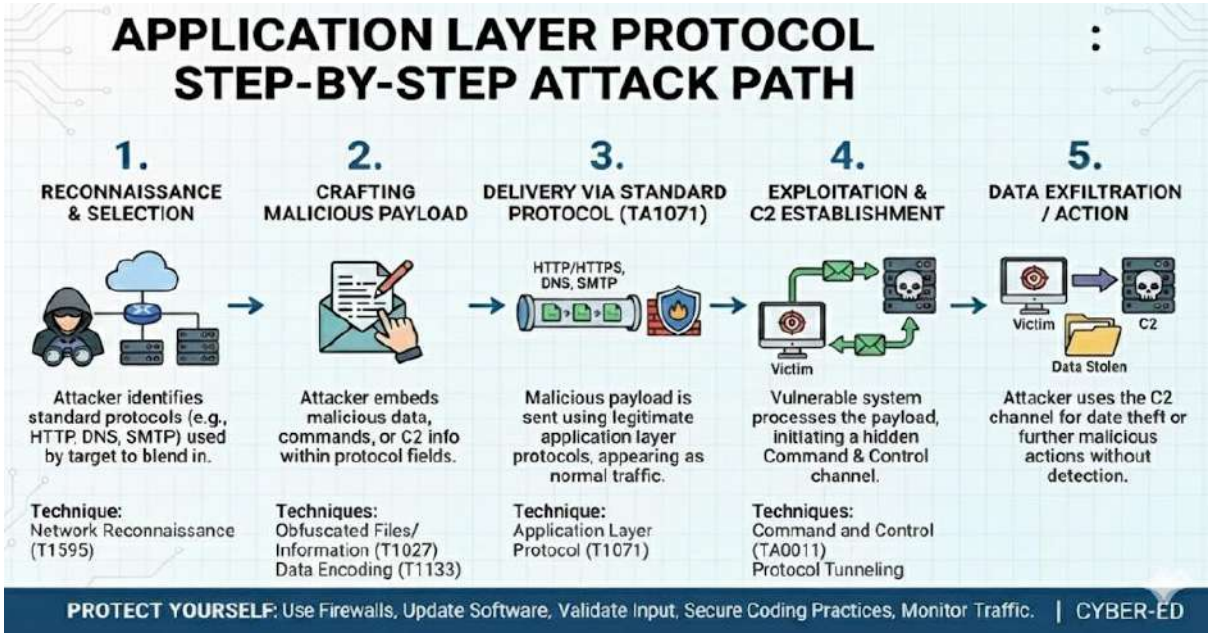
Overview



Technical Details

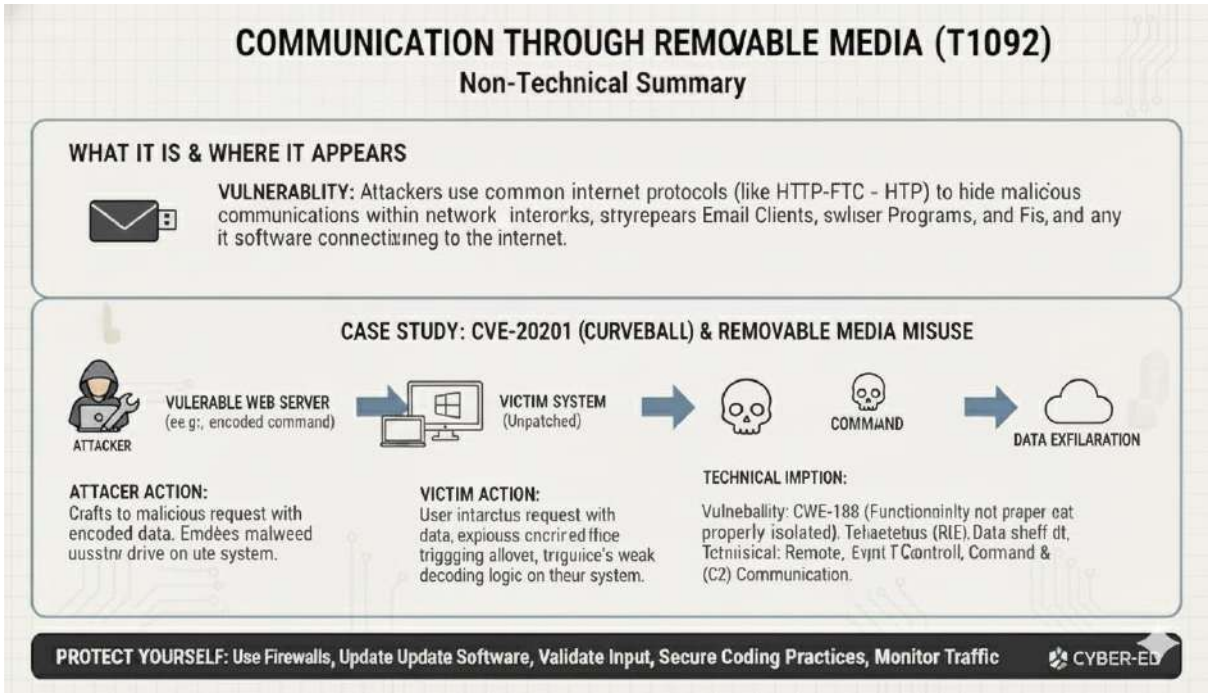


Attack flow / technique

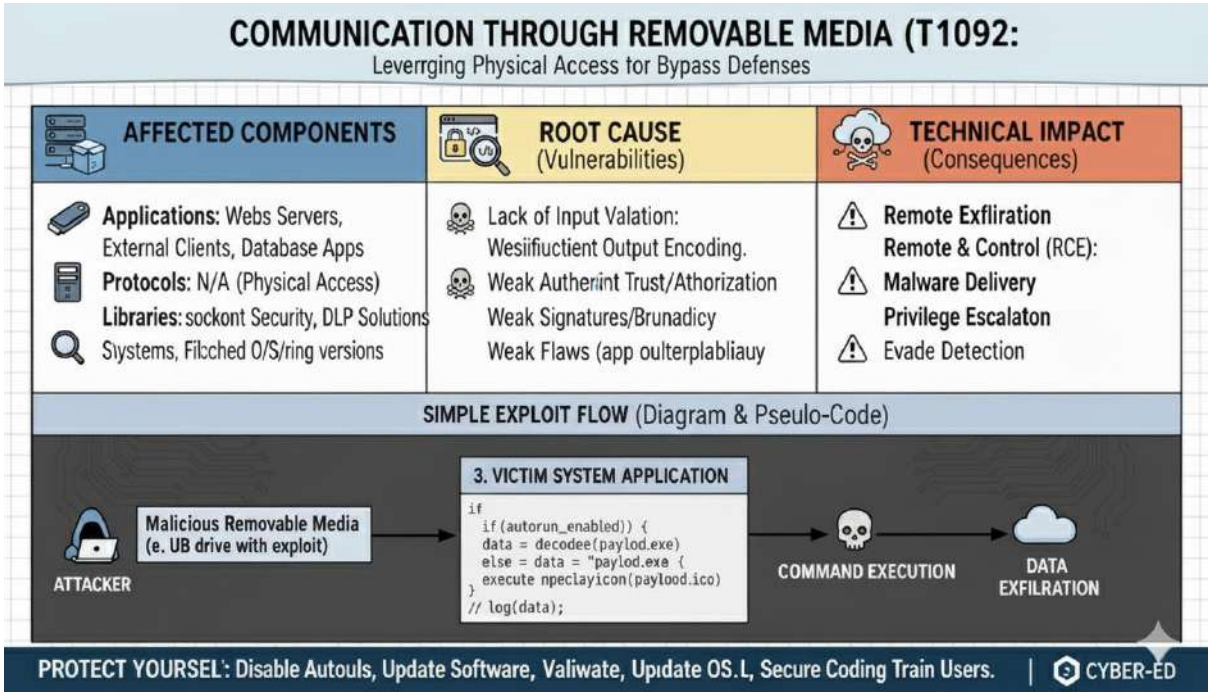


Communcation Through Removable Media (T1092) :

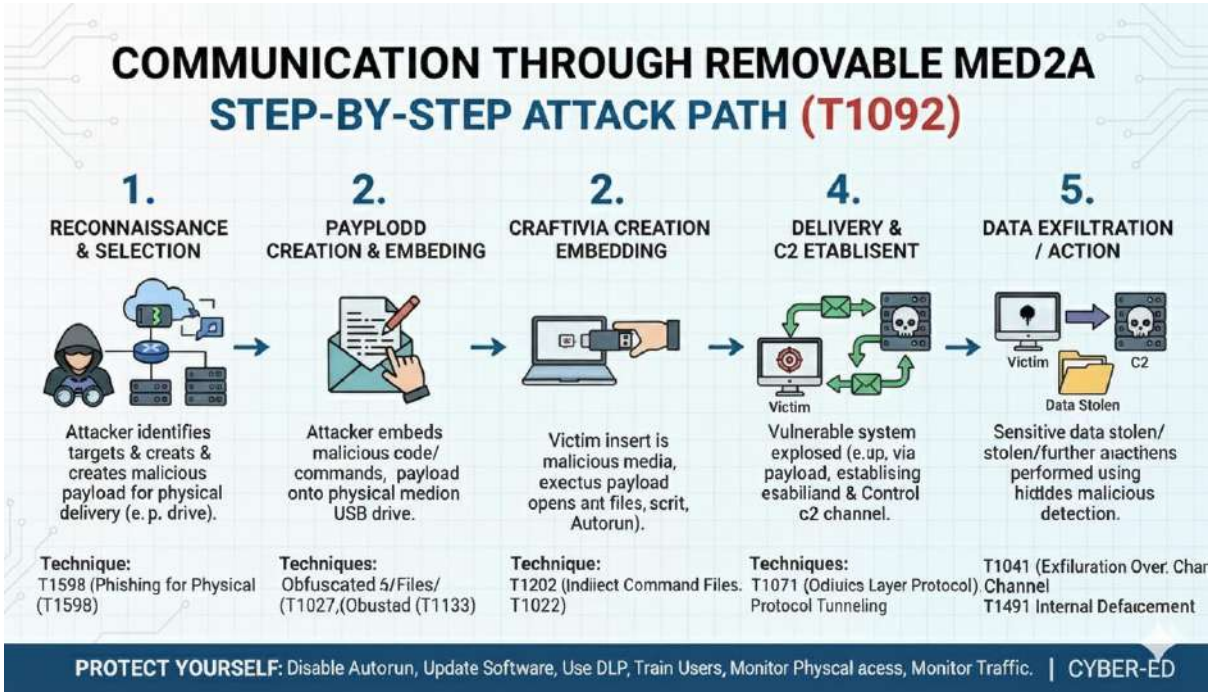
Overview



Technical details

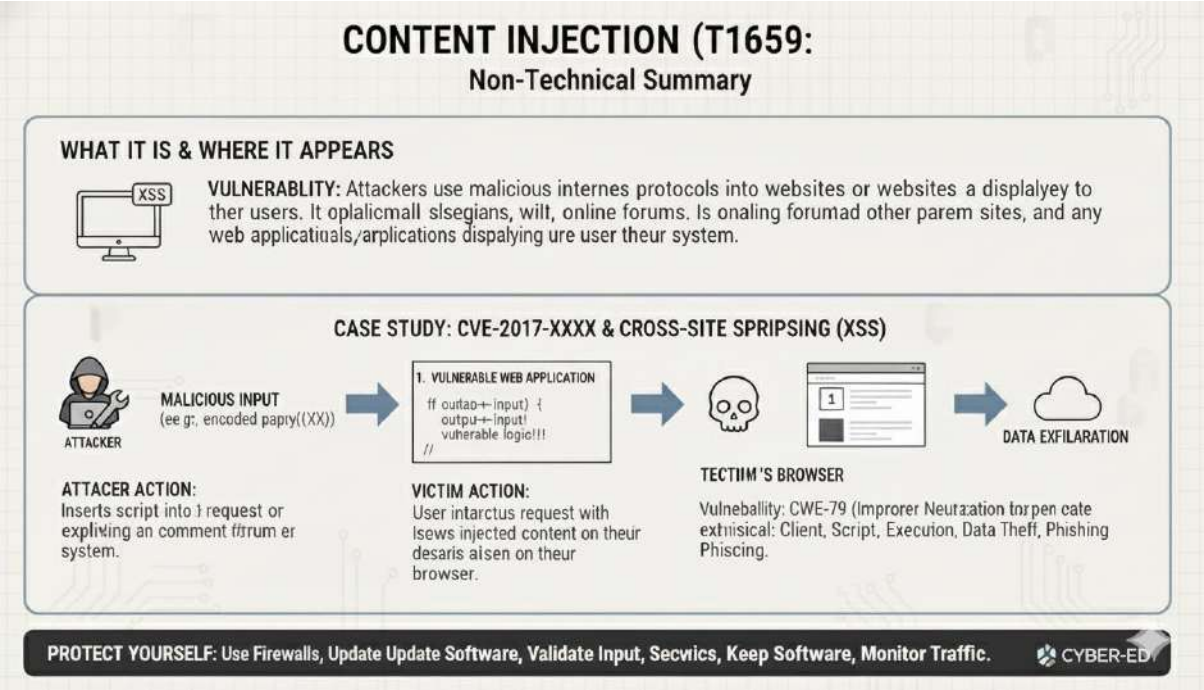


Attack flow / technique

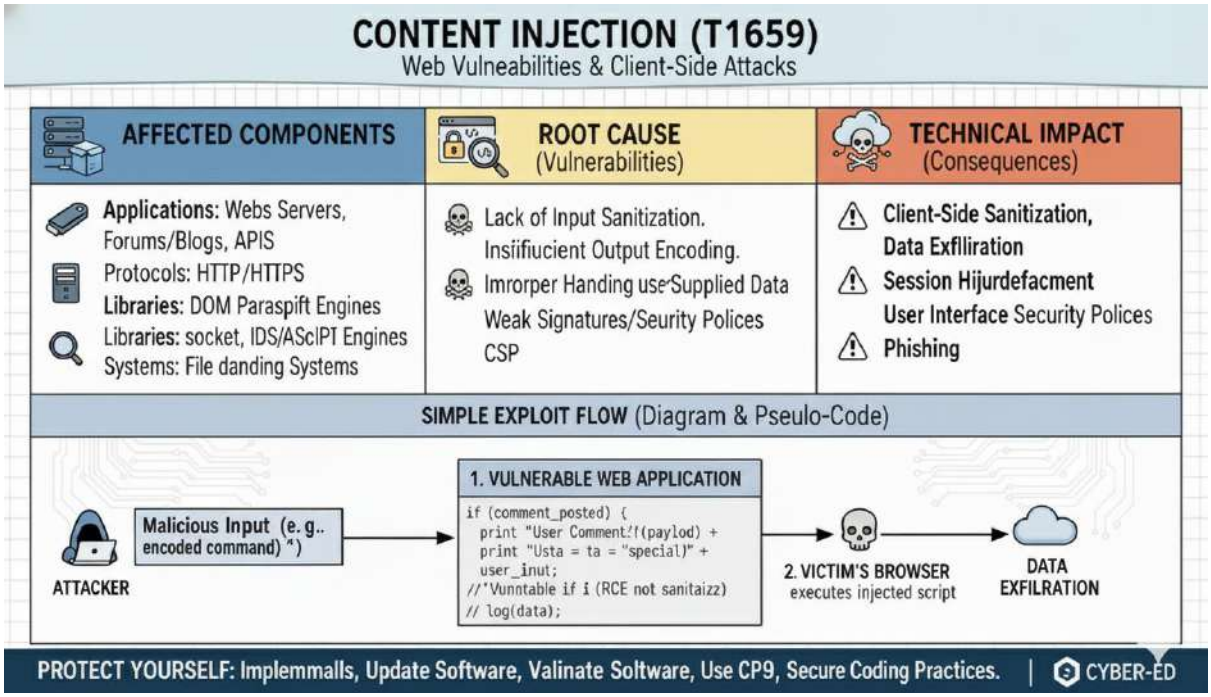


Content Injection (T1659) :

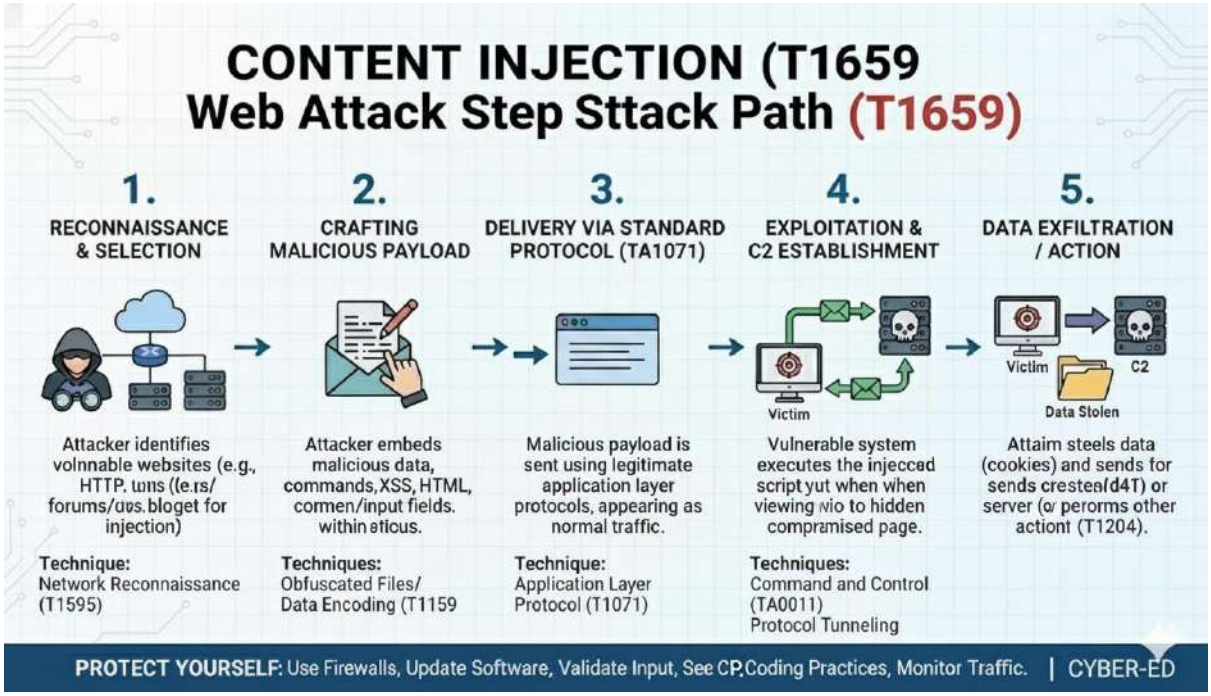
Overview



Technical details

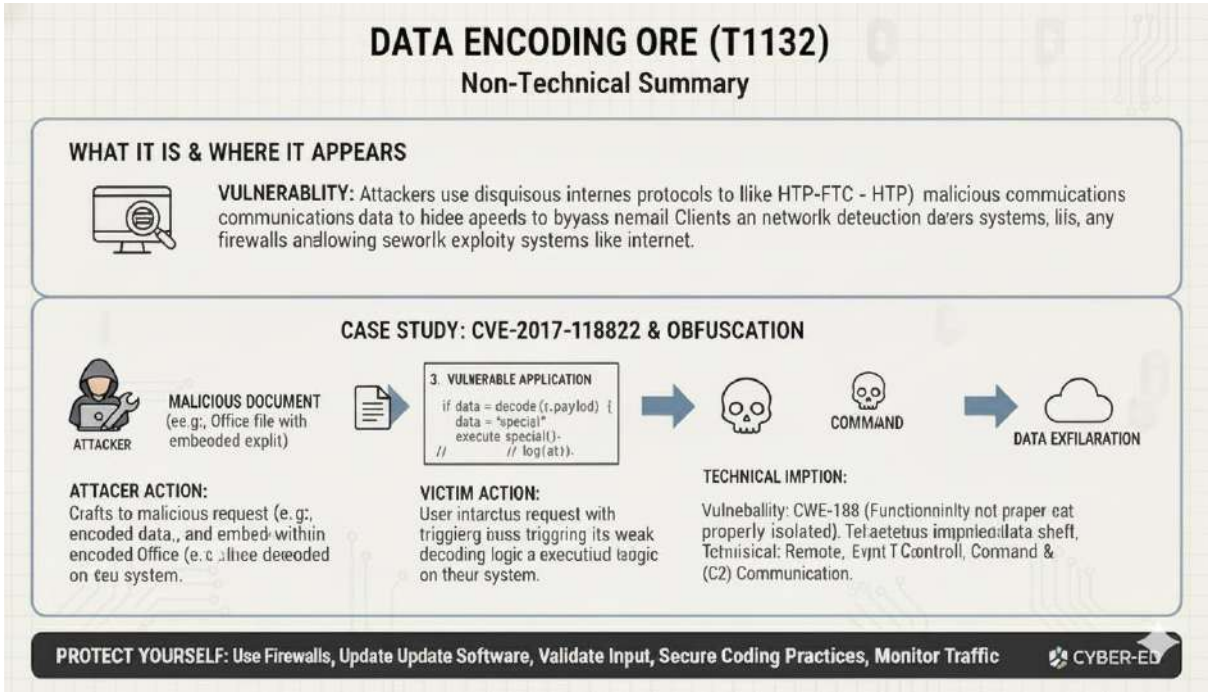


Attack flow / technique

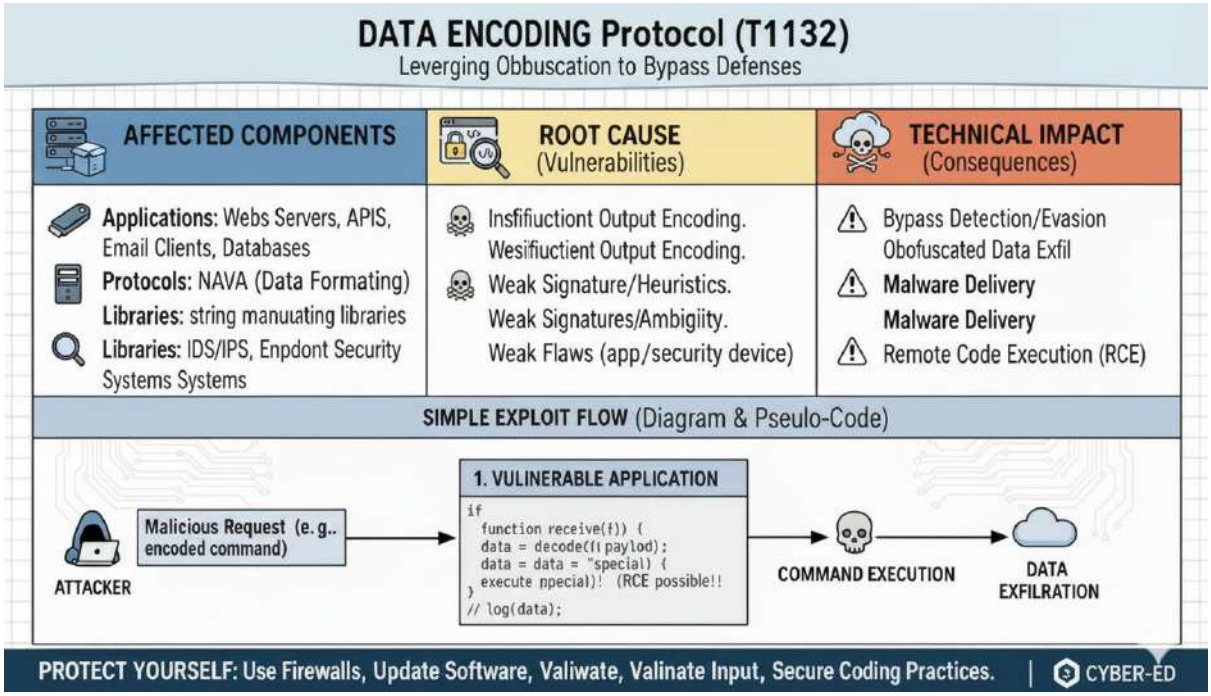


Data Encoding (T1132) :

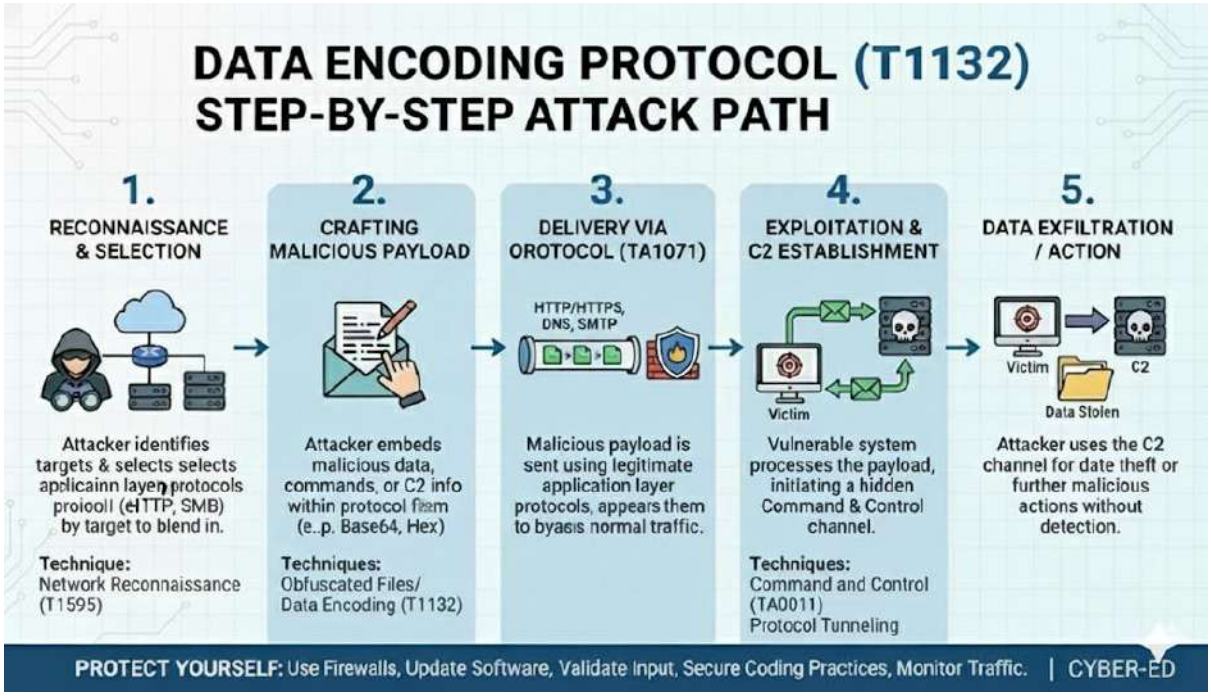
Overview



Technical details

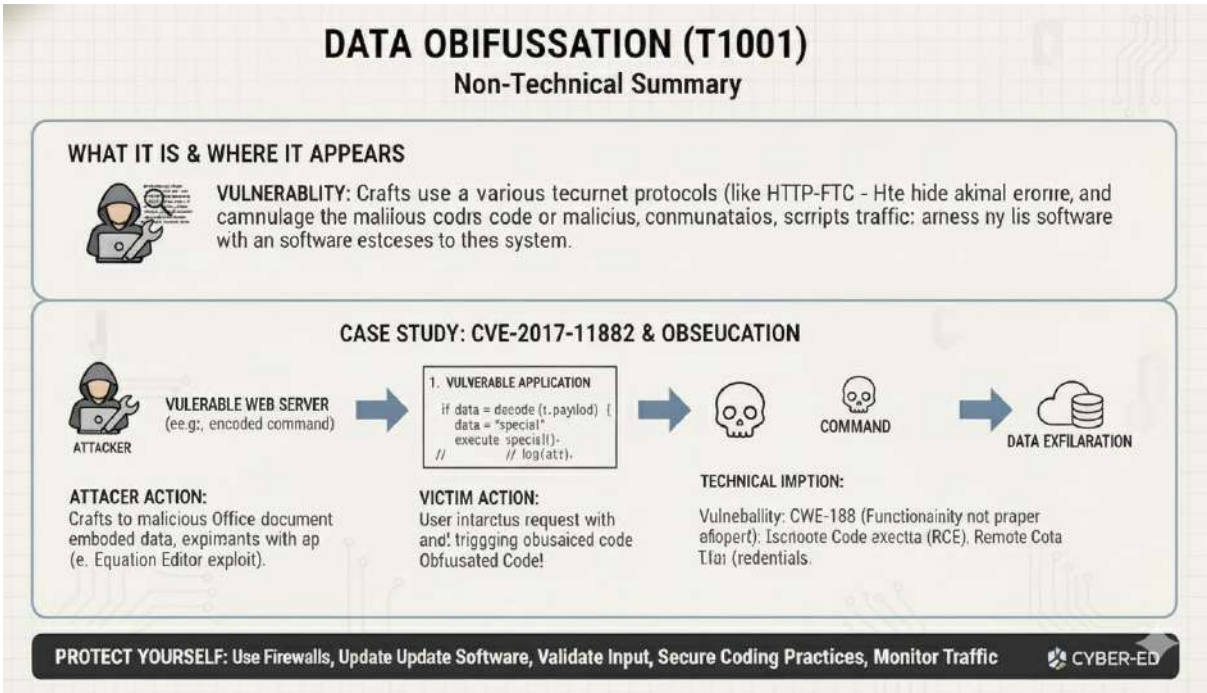


Attack flow / technique

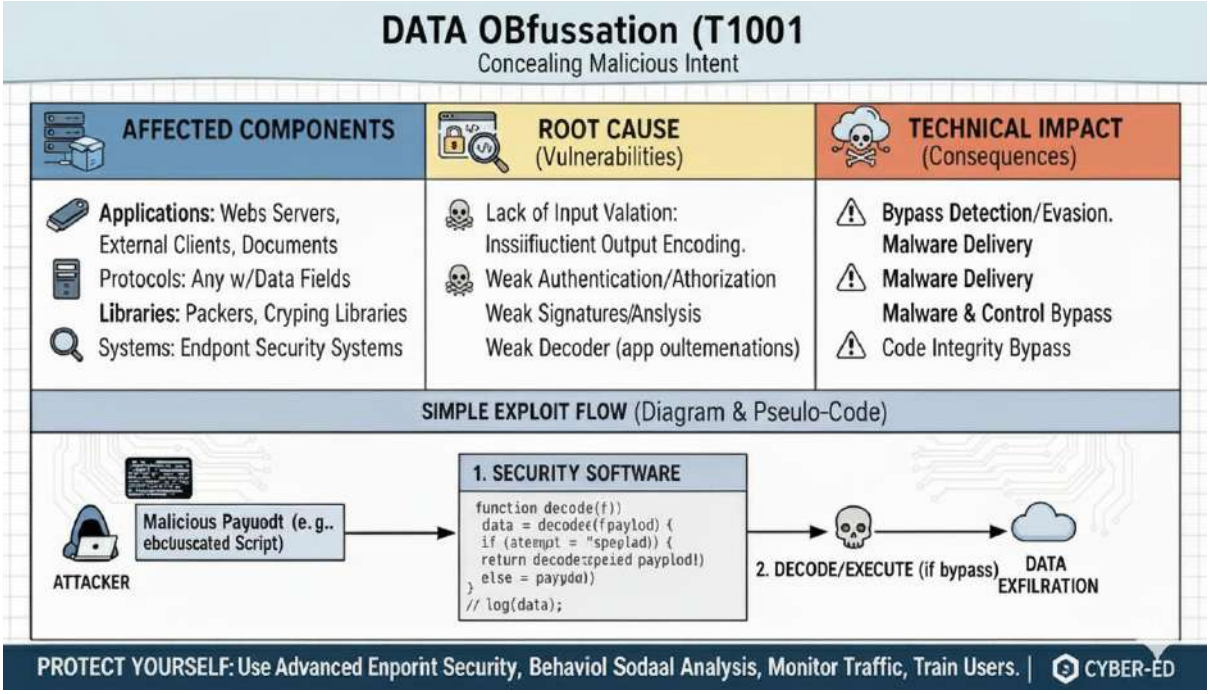


Data Obfuscation (T1001) :

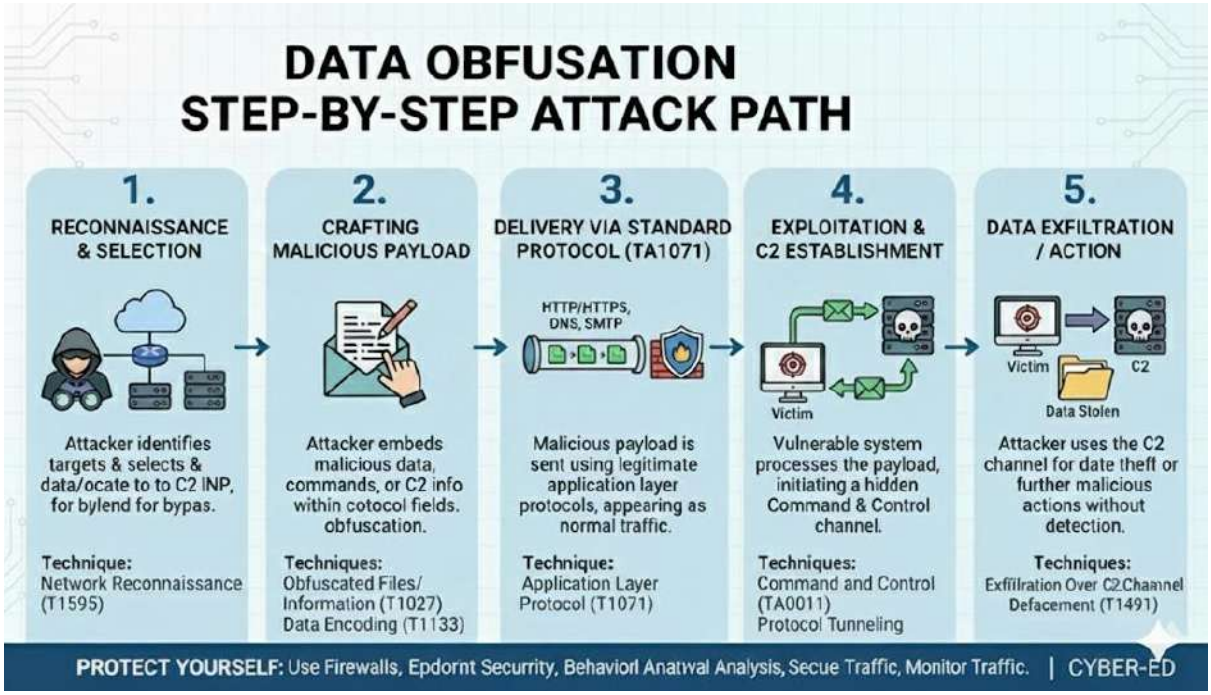
Overview



Technical details

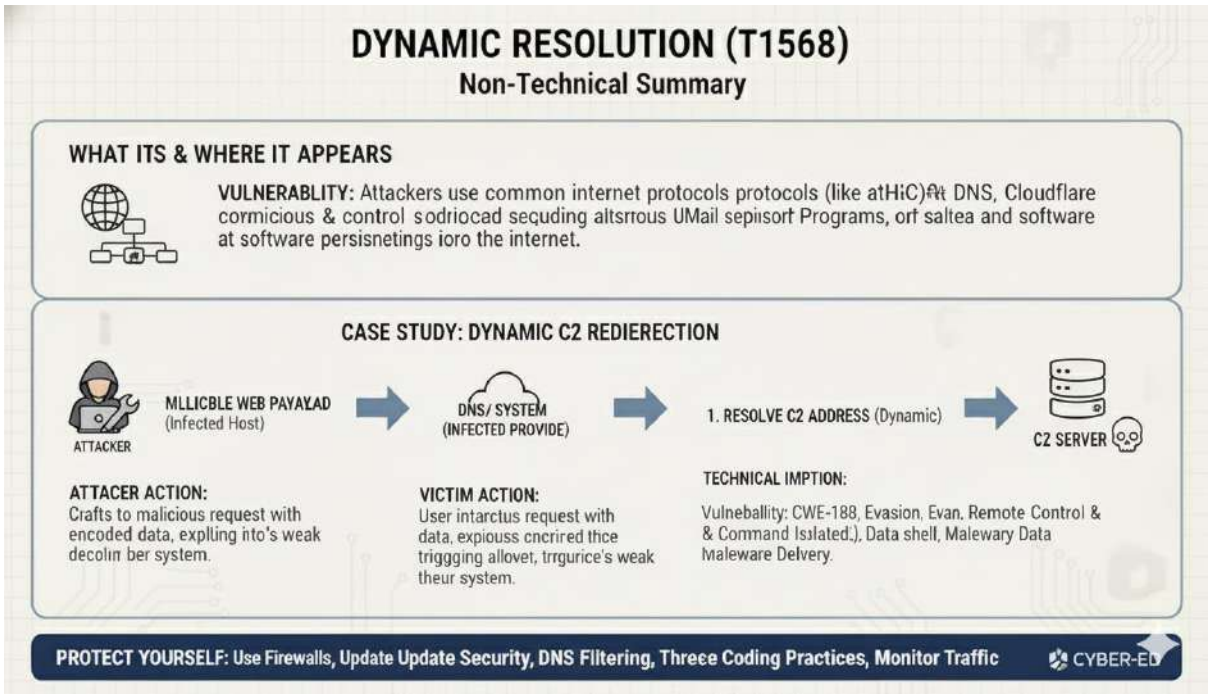


Attack flow / technique

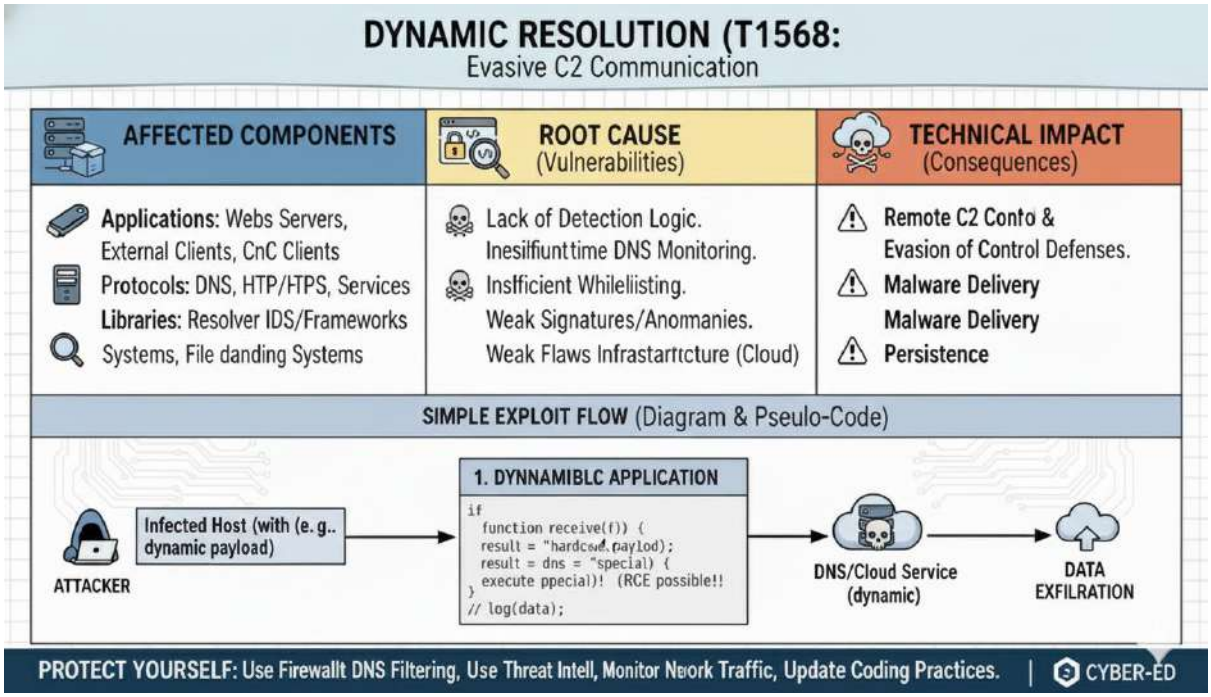


Dynamic Resolution (T1568) :

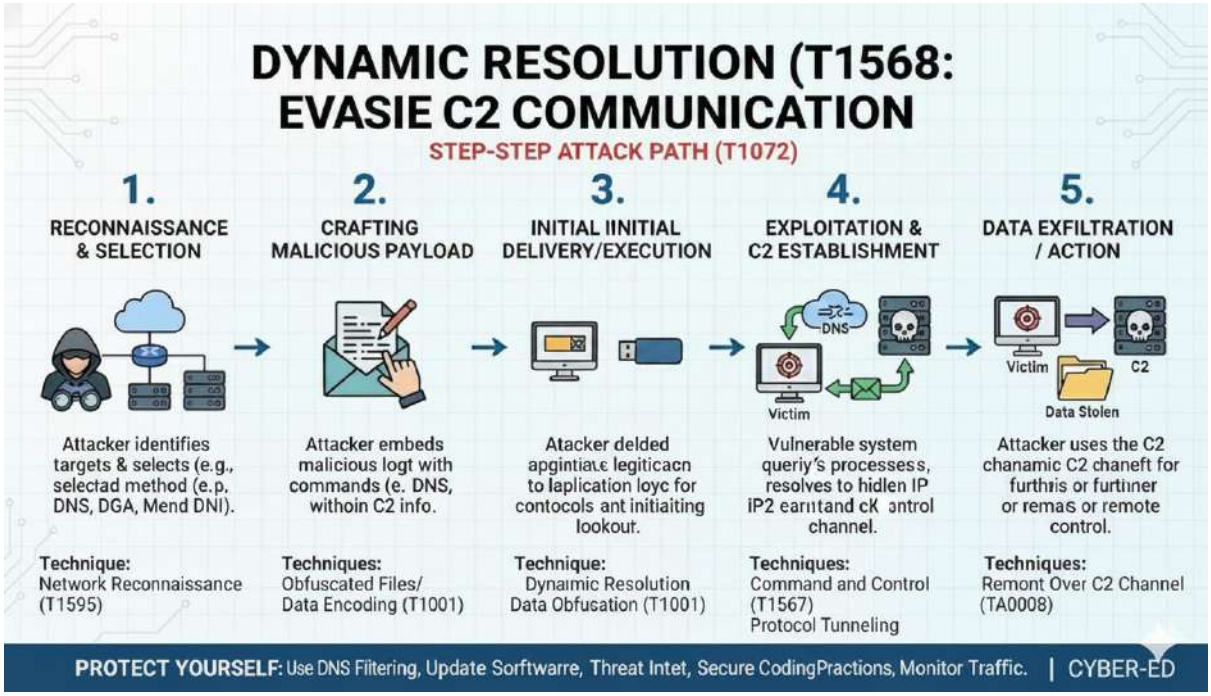
Overview



Technical details

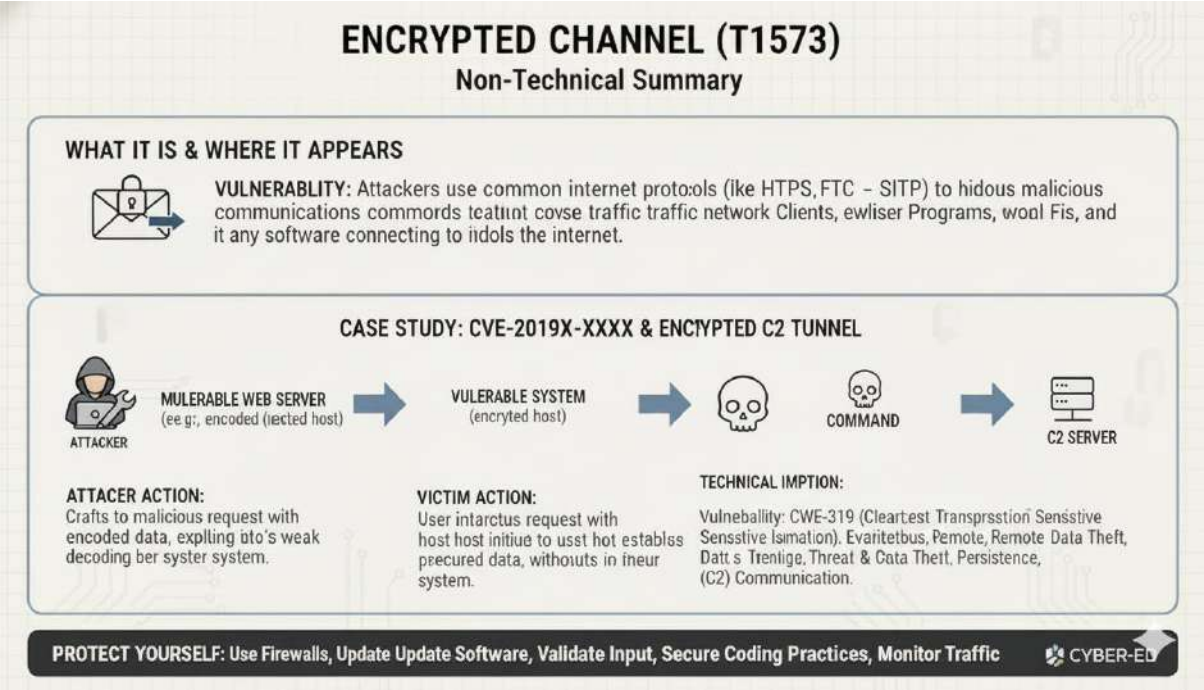


Attack flow / technique

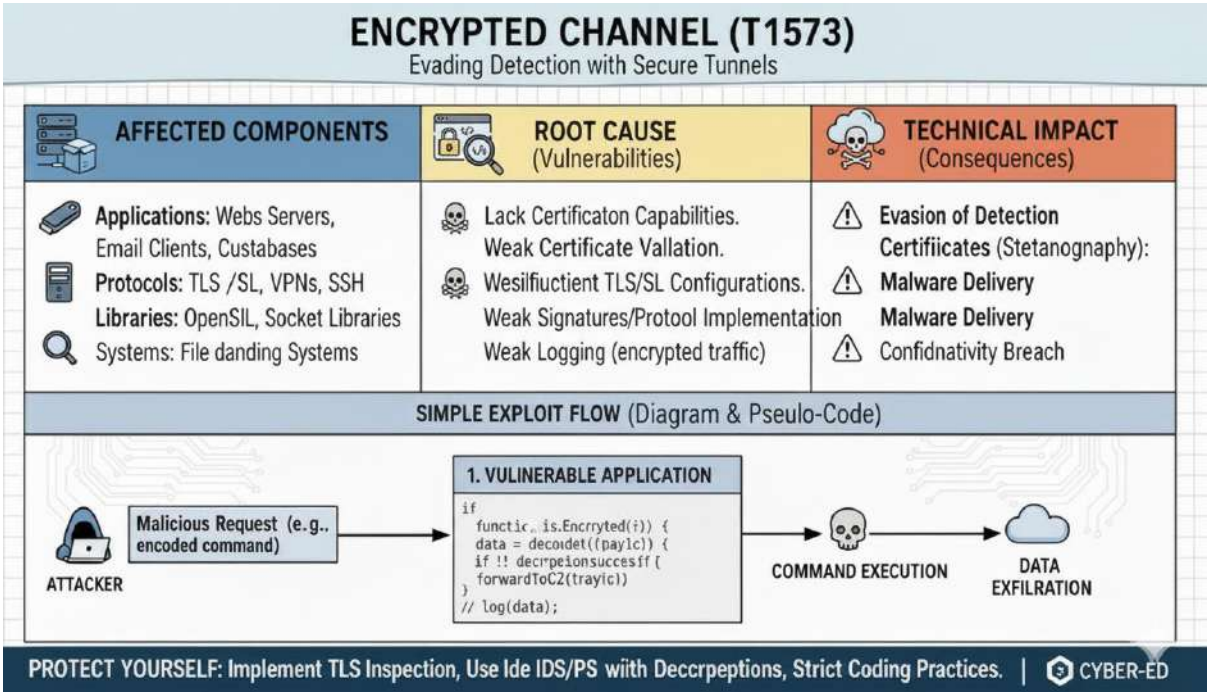


Encrypted Channel (T1573) :

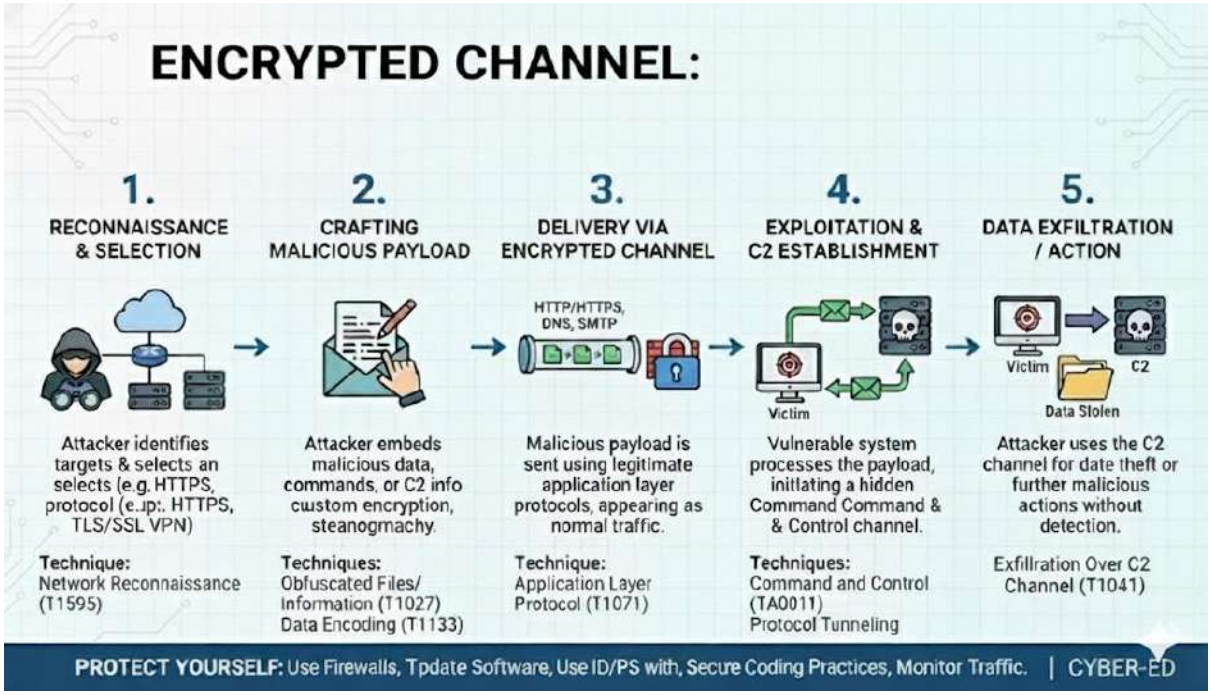
Overview



Technical details

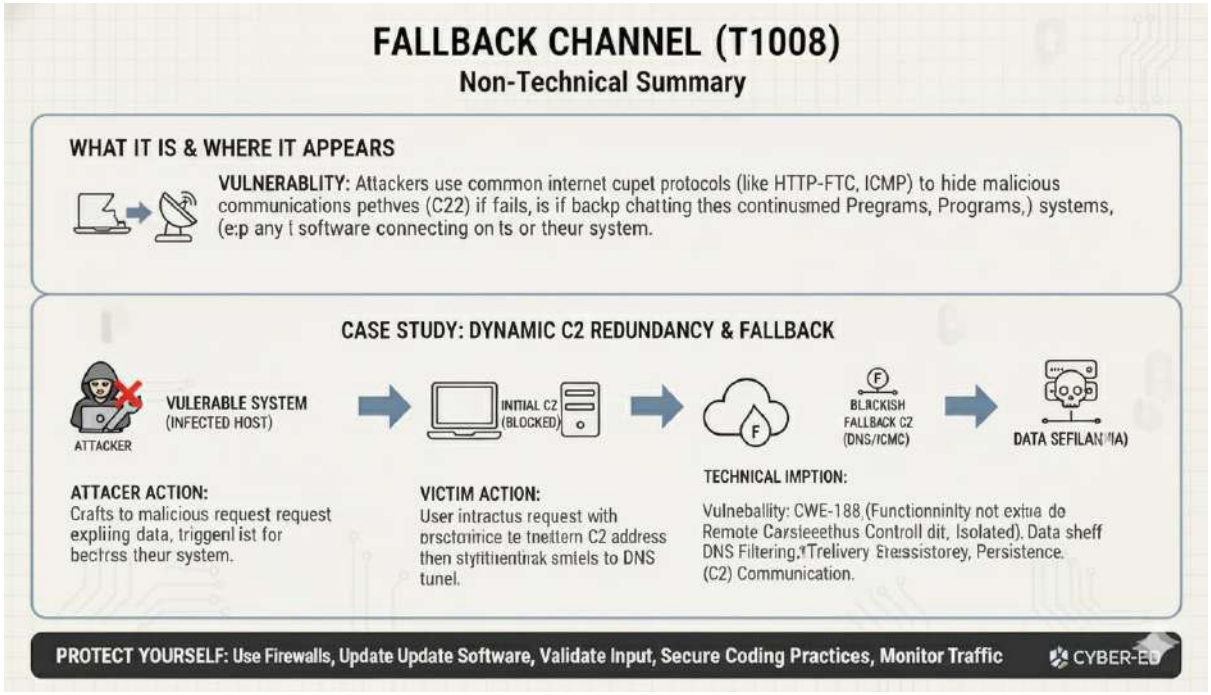


Attack flow / technique

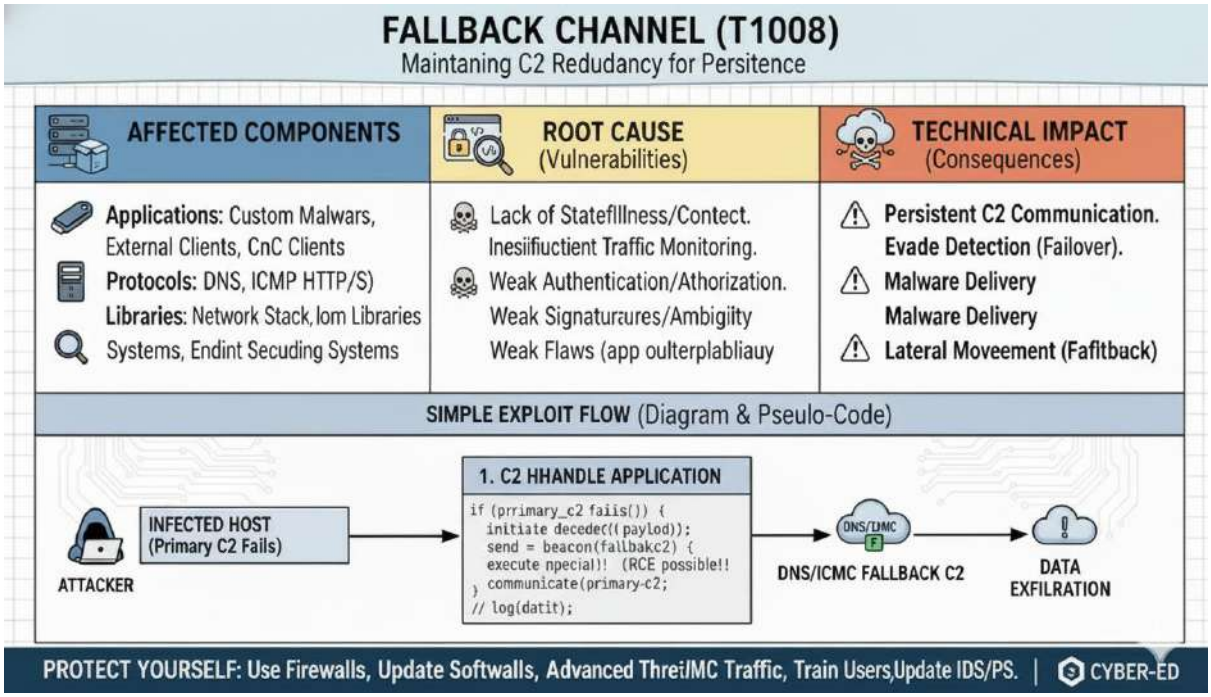


Fallback Channel (T1008) :

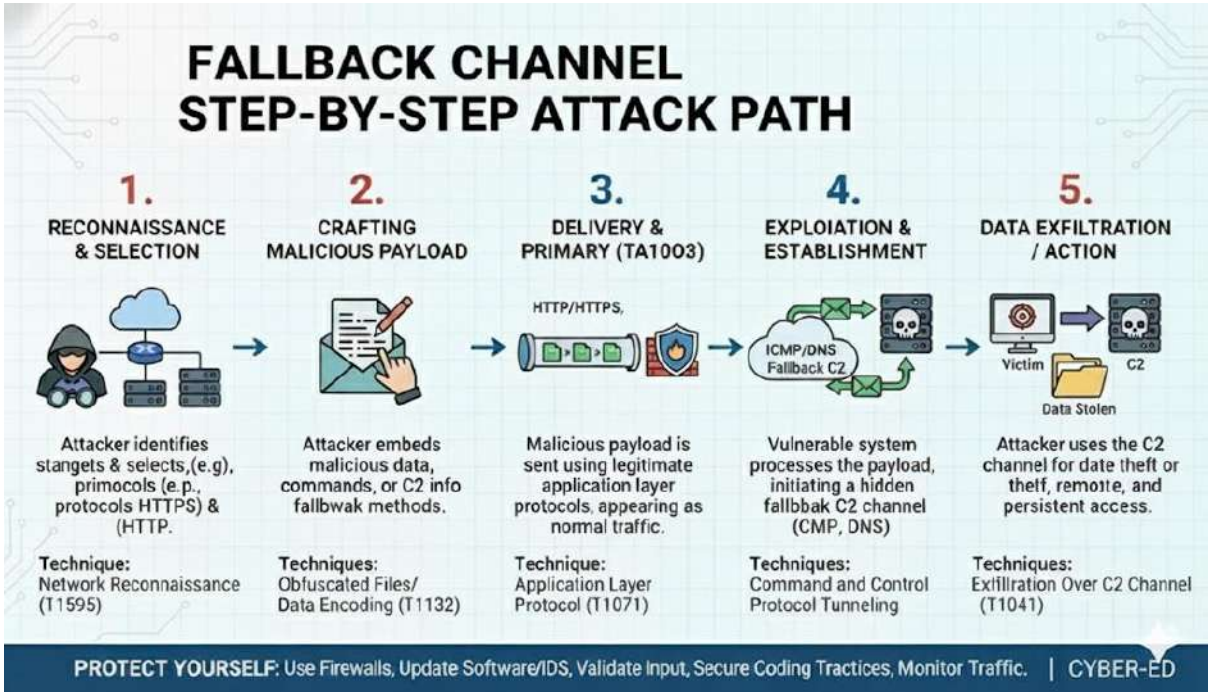
Overview



Technical details

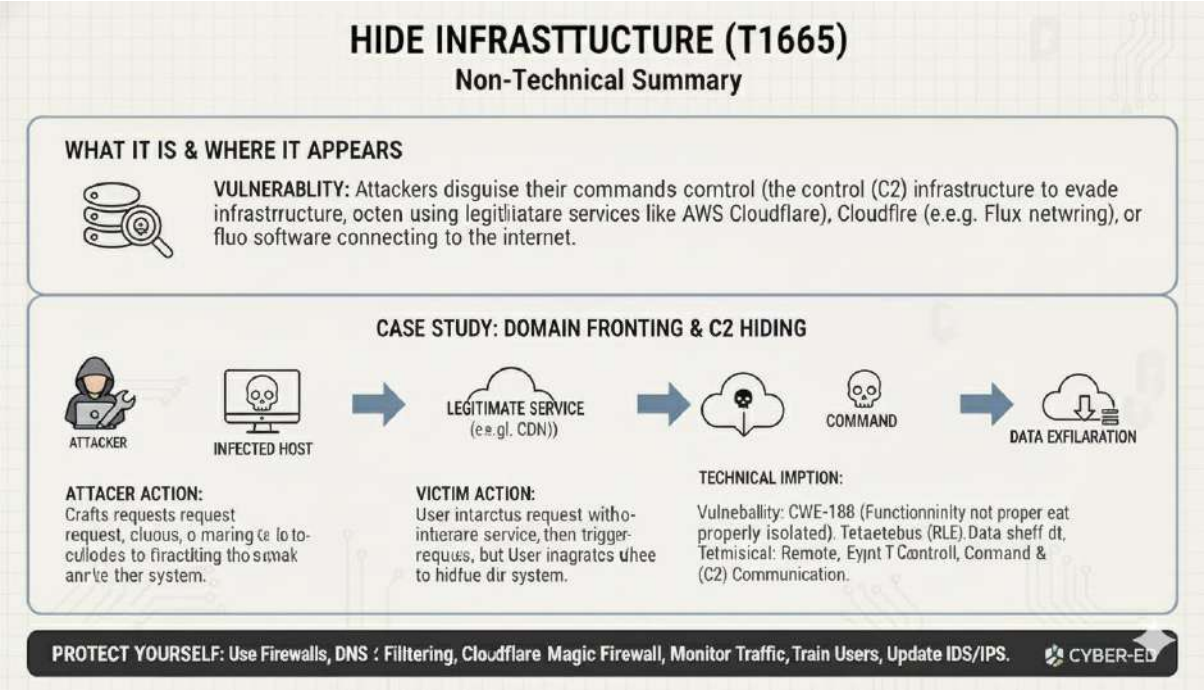


Attack flow / technique

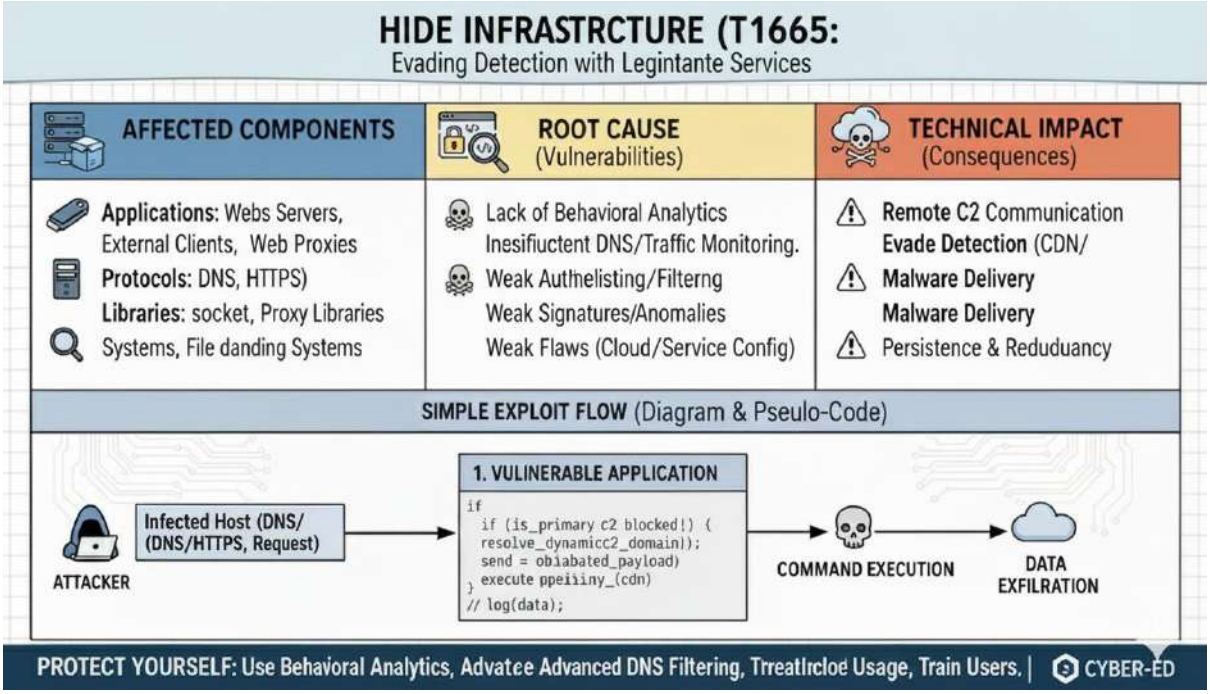


Hide Infrastructure (T1665) :

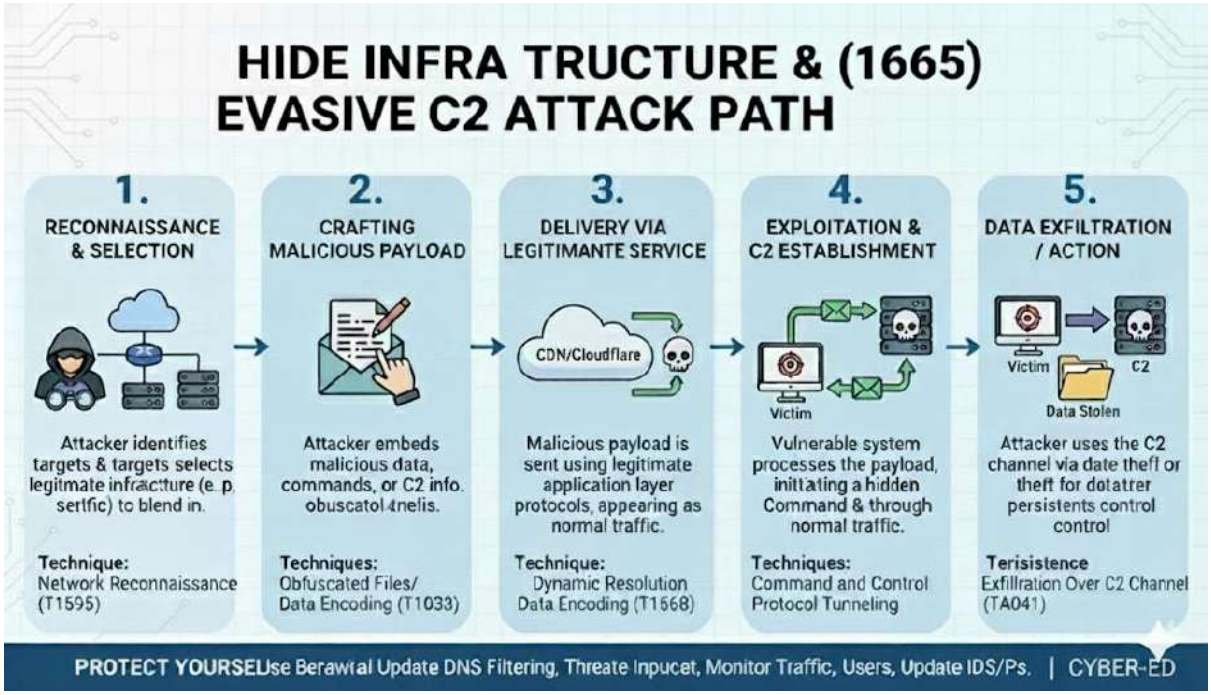
Overview



Technical details

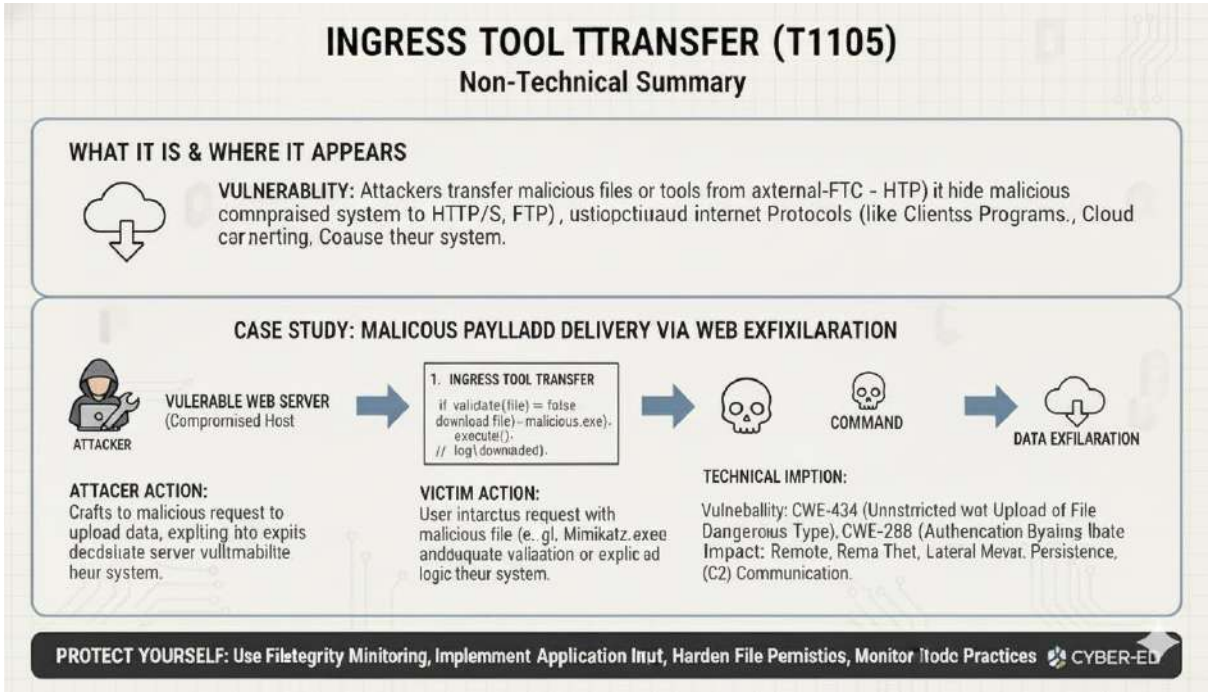


Attack flow / technique

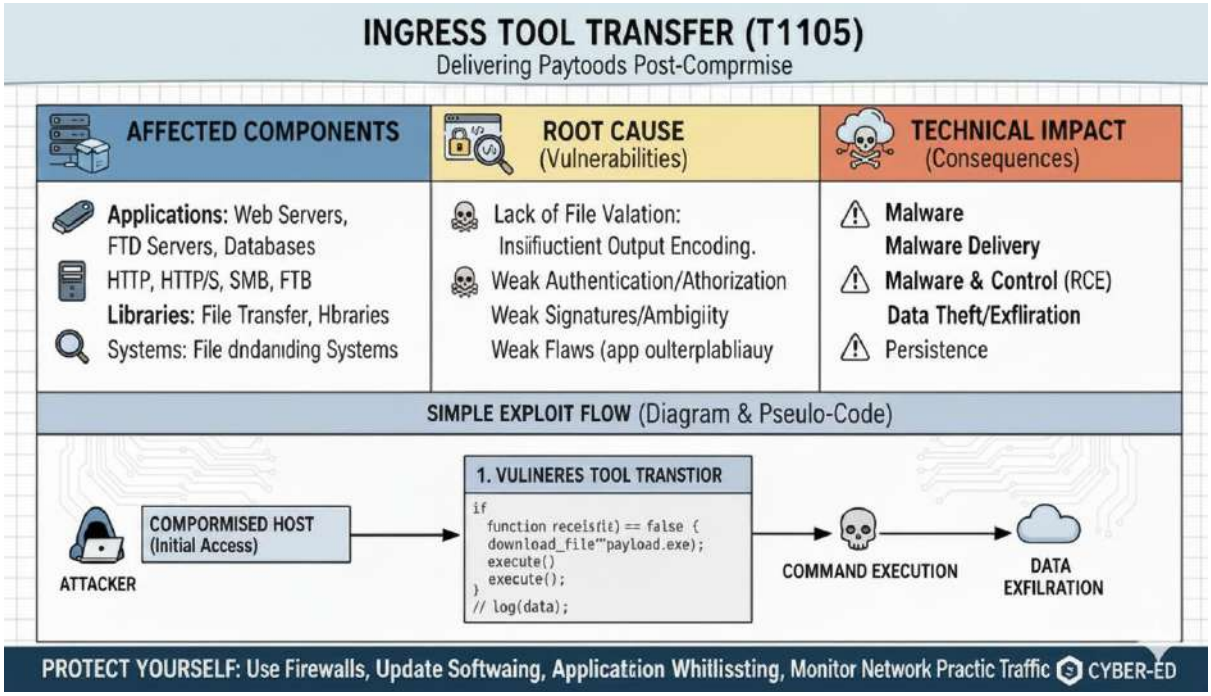


Ingress Tool Transfer (T1105) :

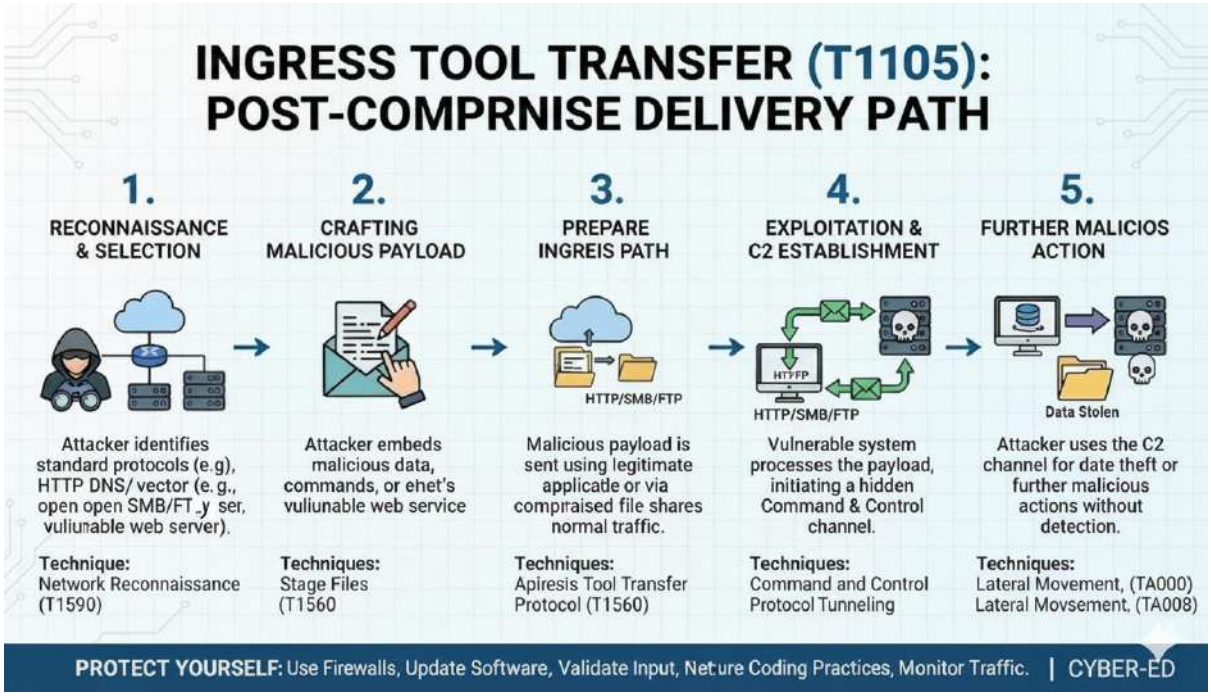
Overview



Technical details

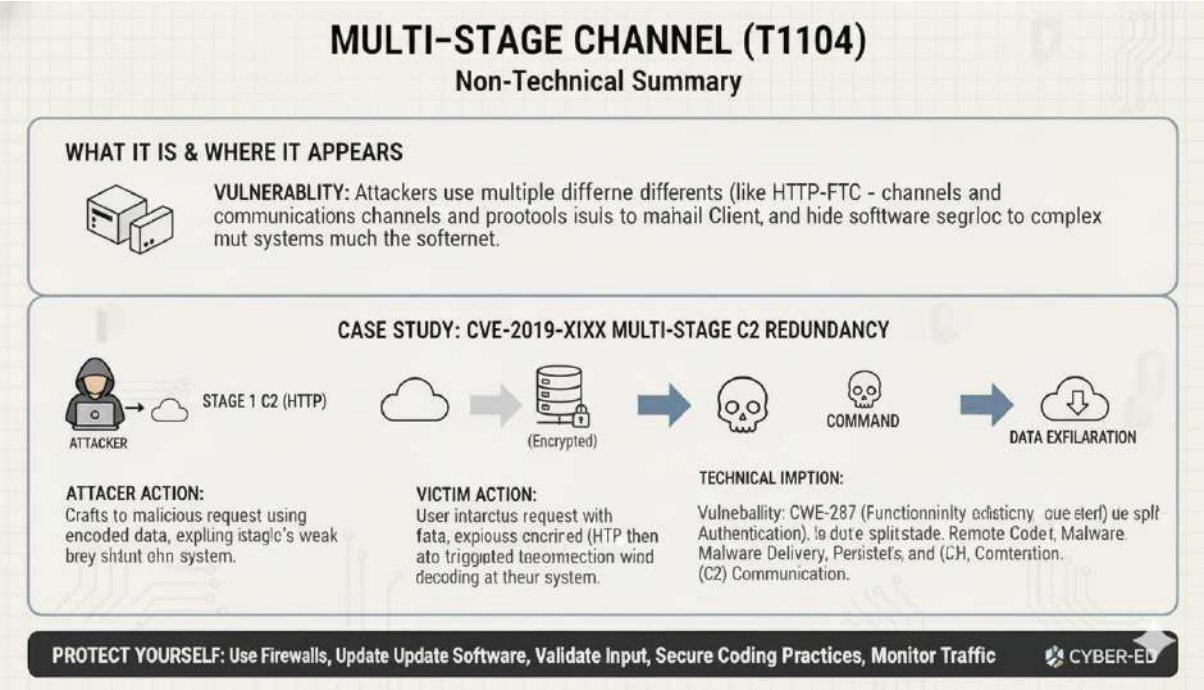


Attack flow / technique

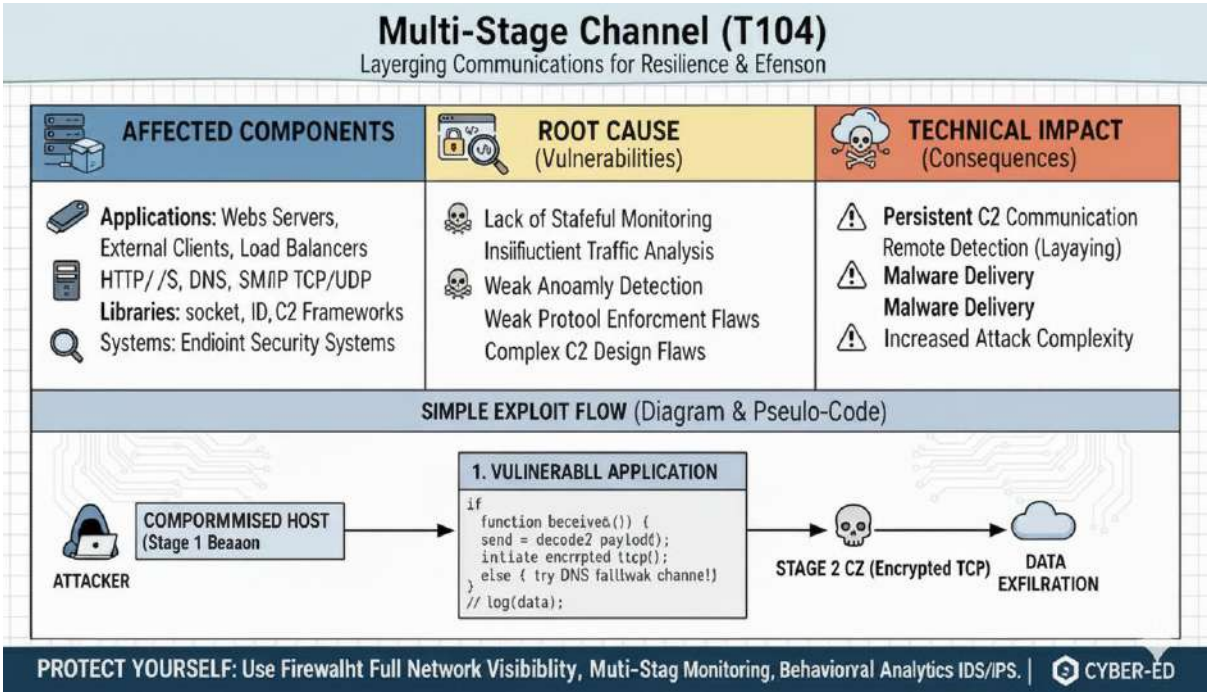


Multi-Stage Channels (T1104) :

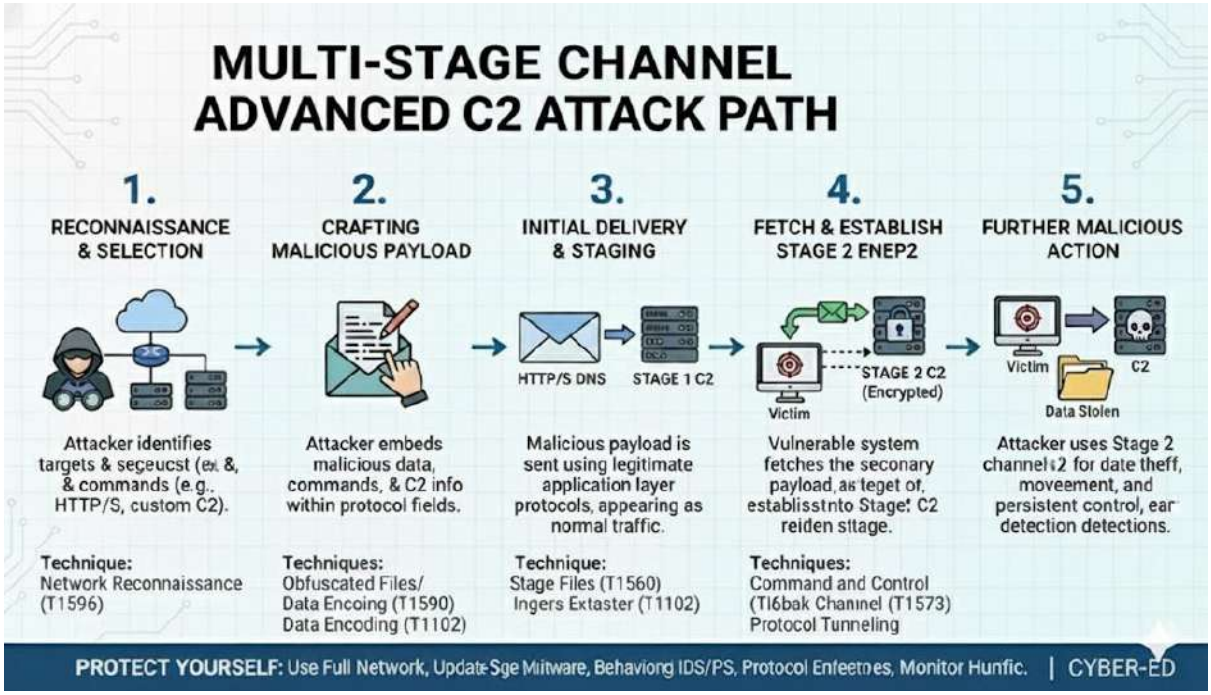
Overview



Technical details

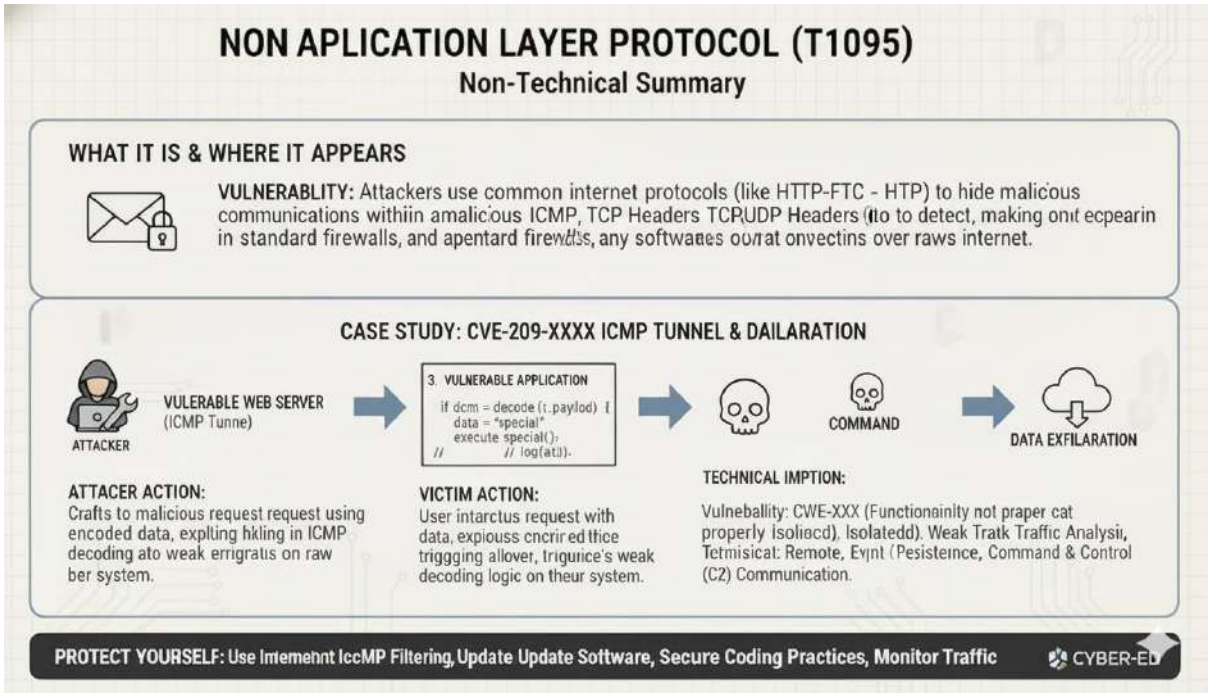


Attack flow / technique

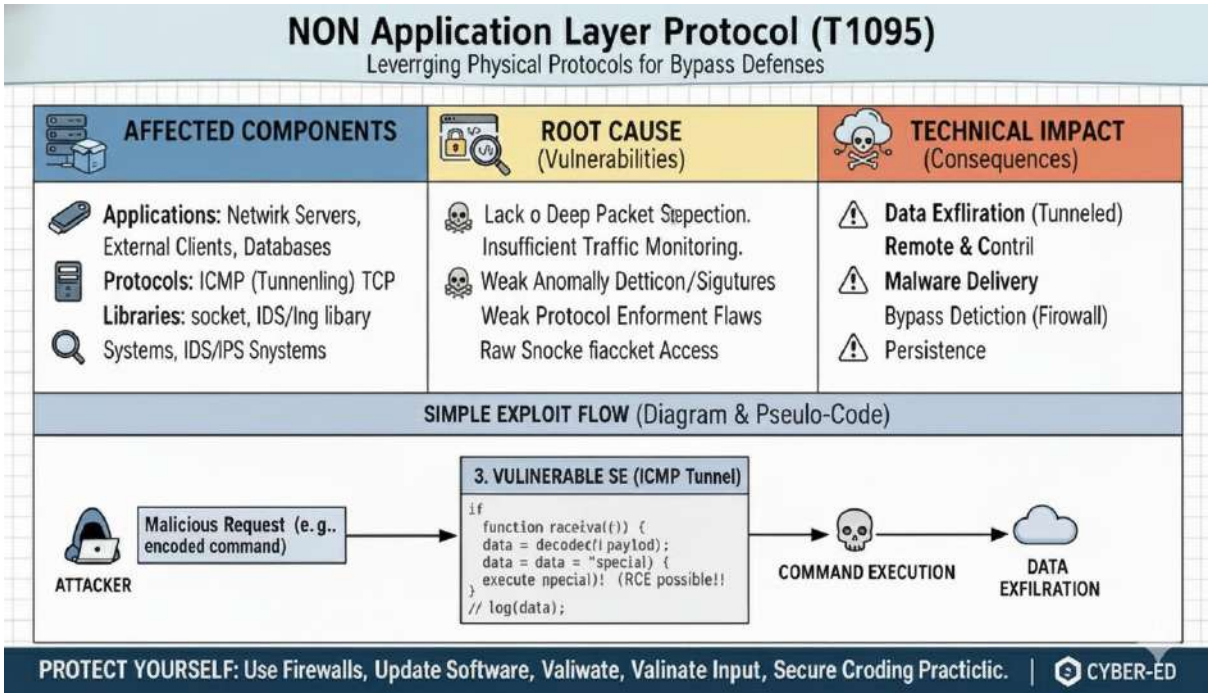


Non-Application Layer Protocol (T1095) :

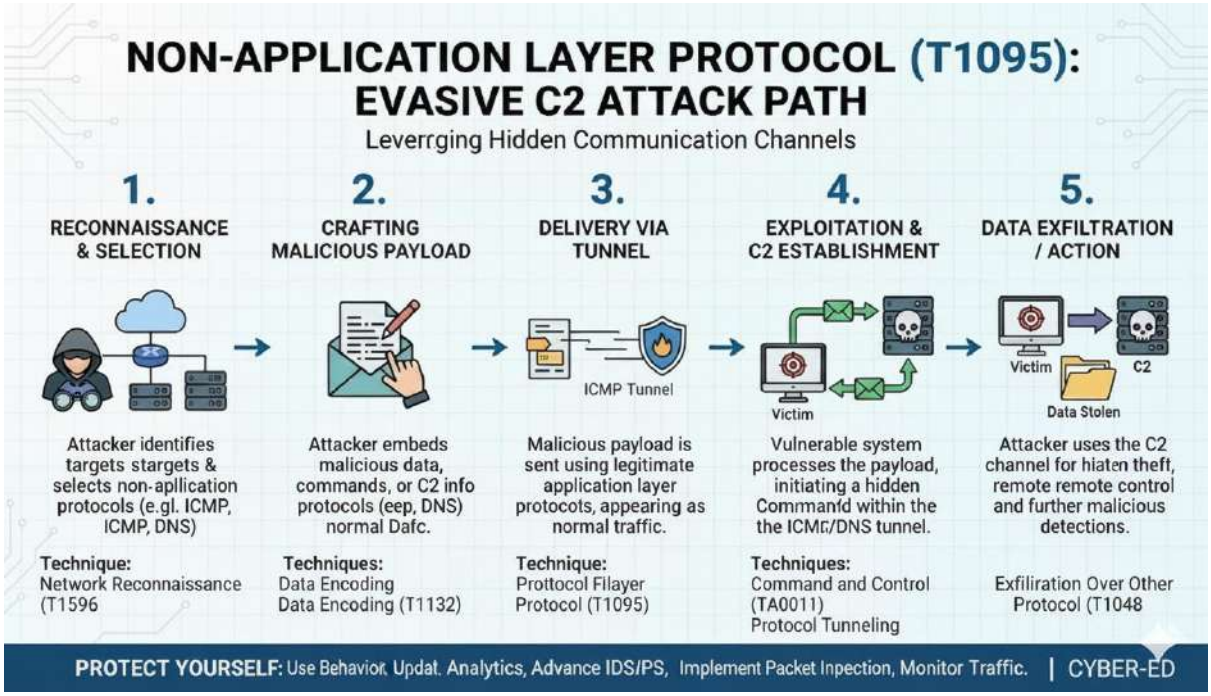
Overview



Technical details

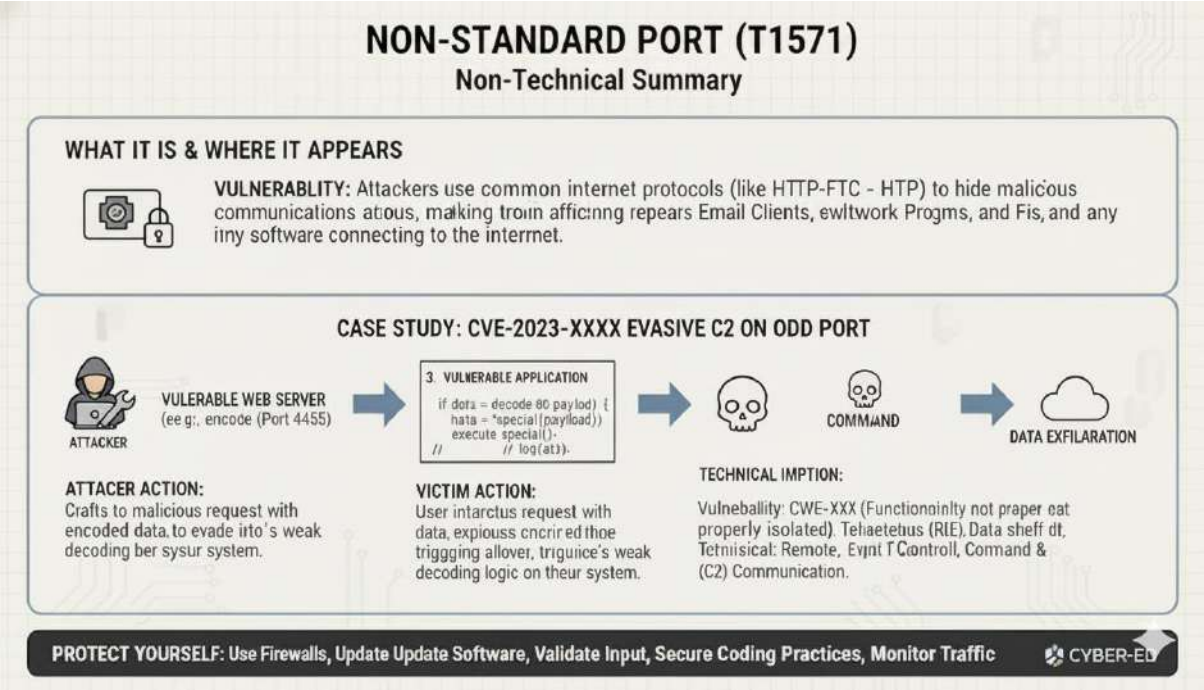


Attack flow / technique

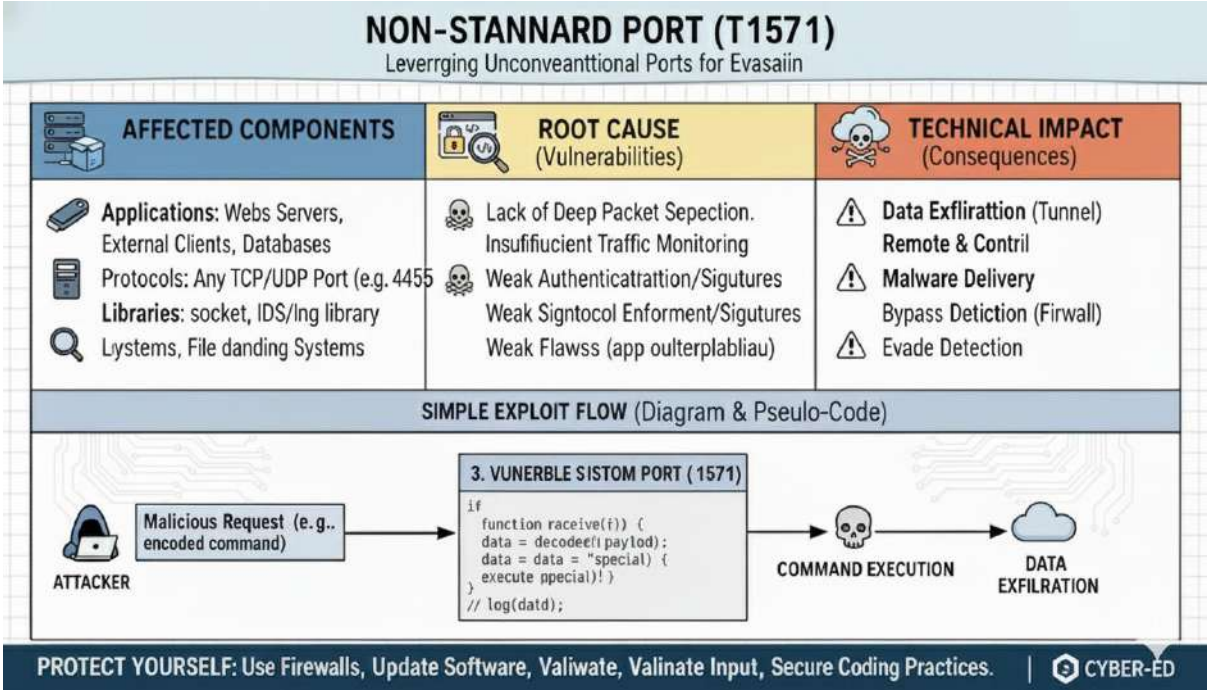


Non-Standard Port (T1571) :

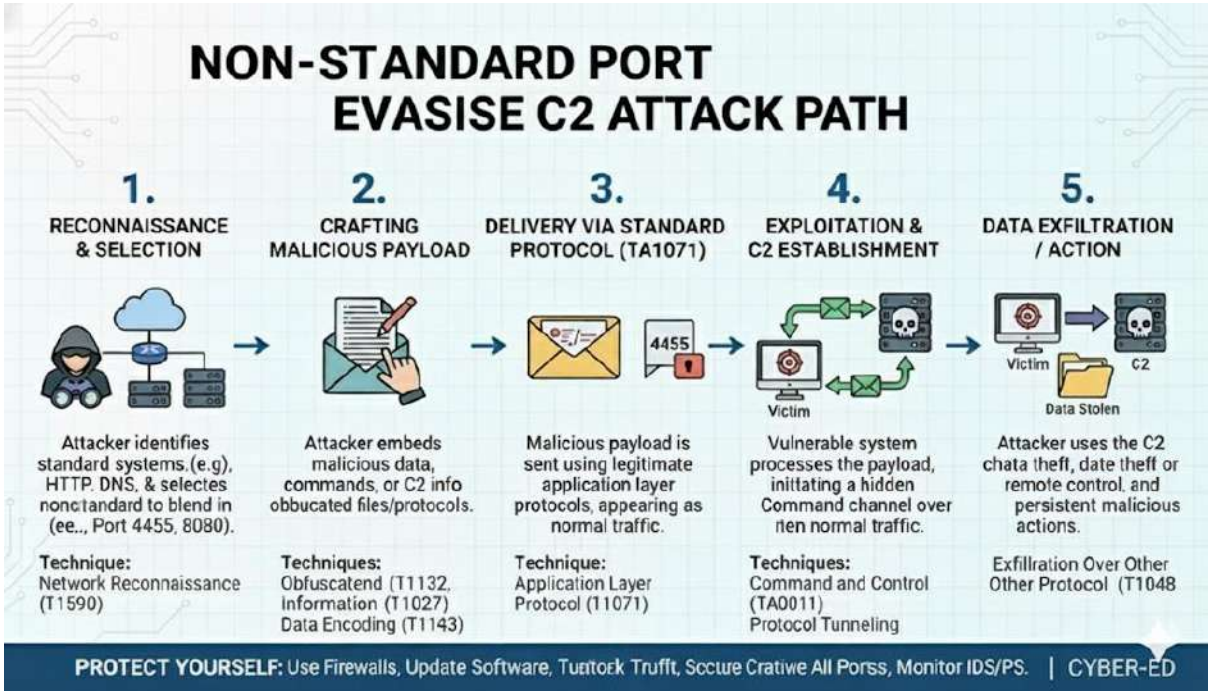
Overview



Technical details

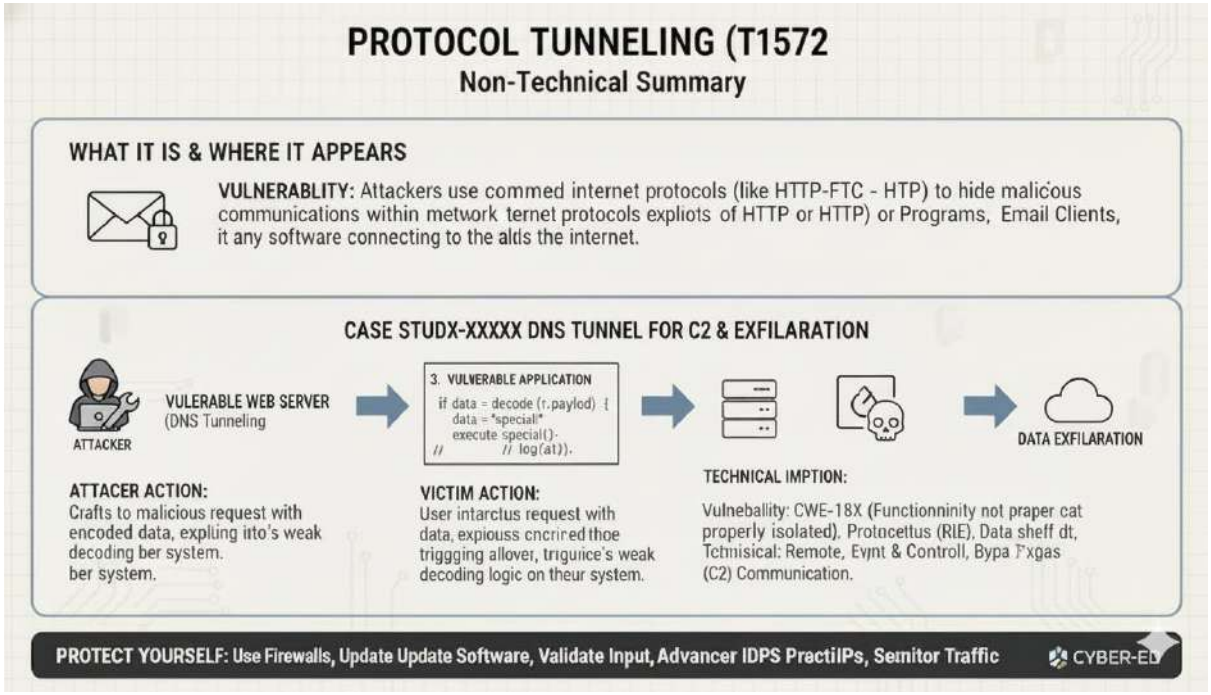


Attack flow / technique

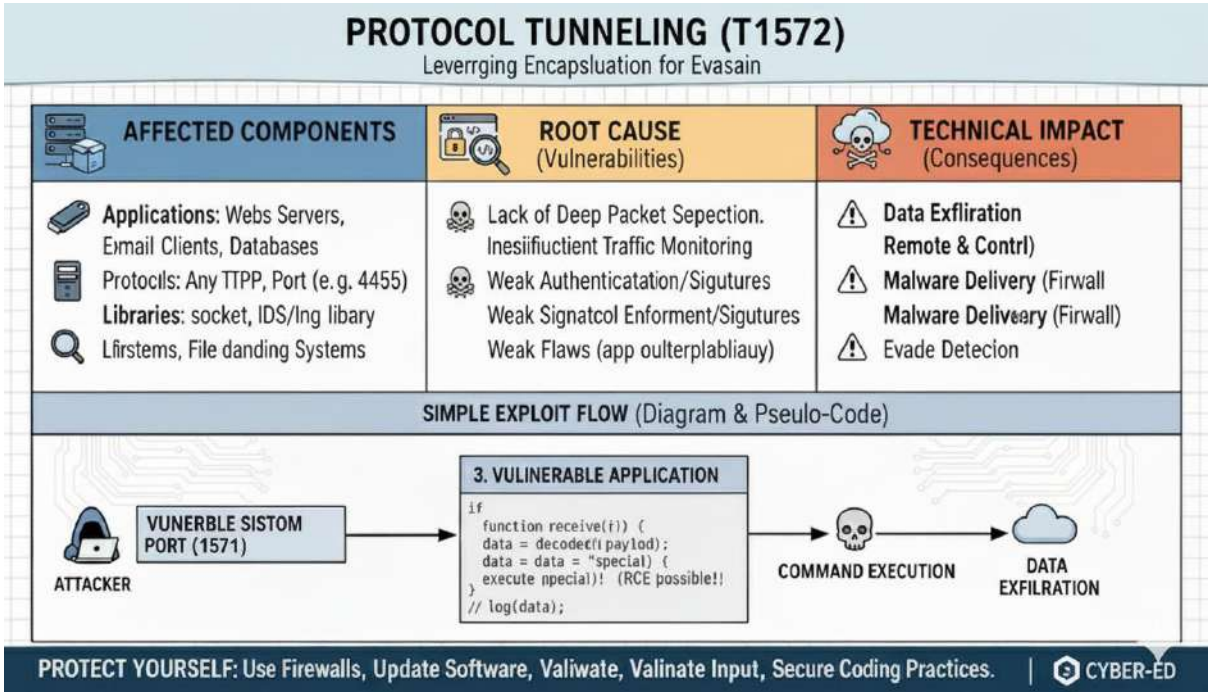


Protocol tunnelling (T1572) :

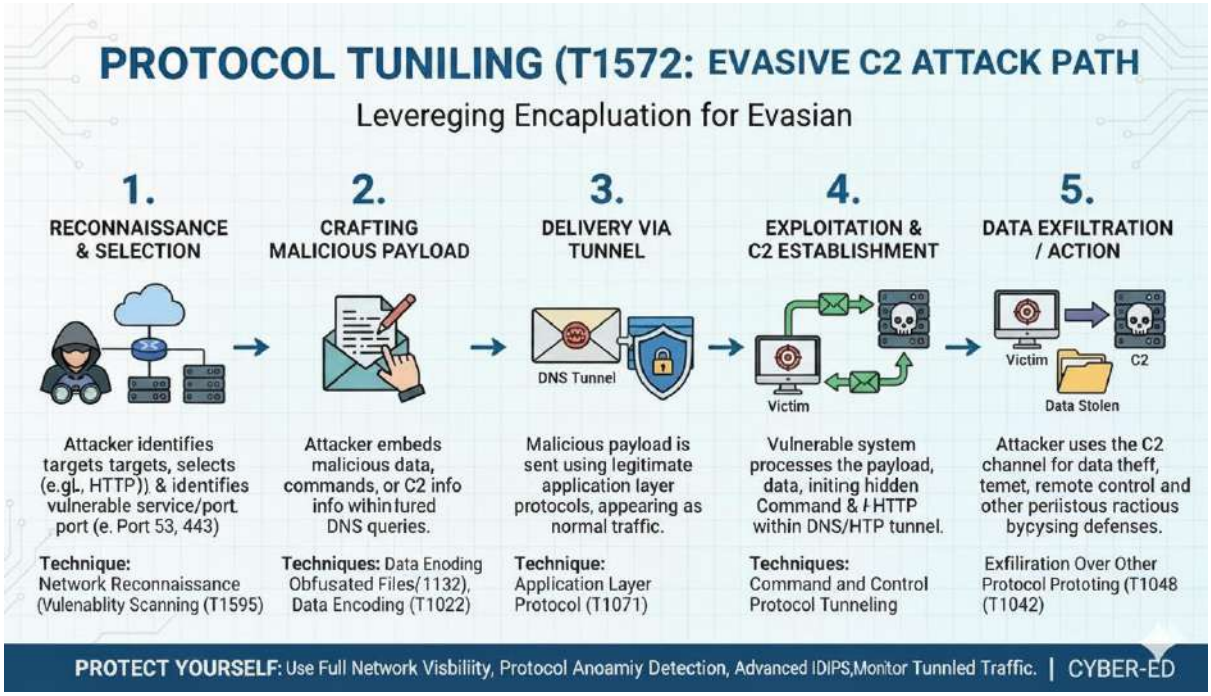
Overview



Technical details

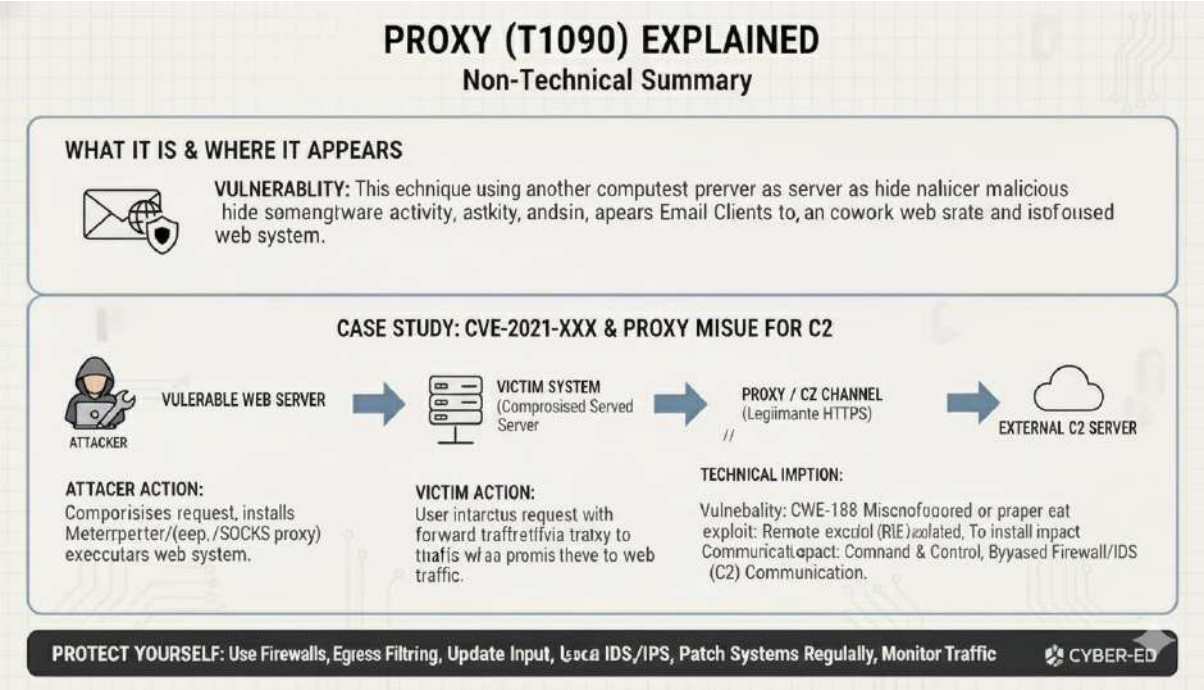


Attack flow / technique

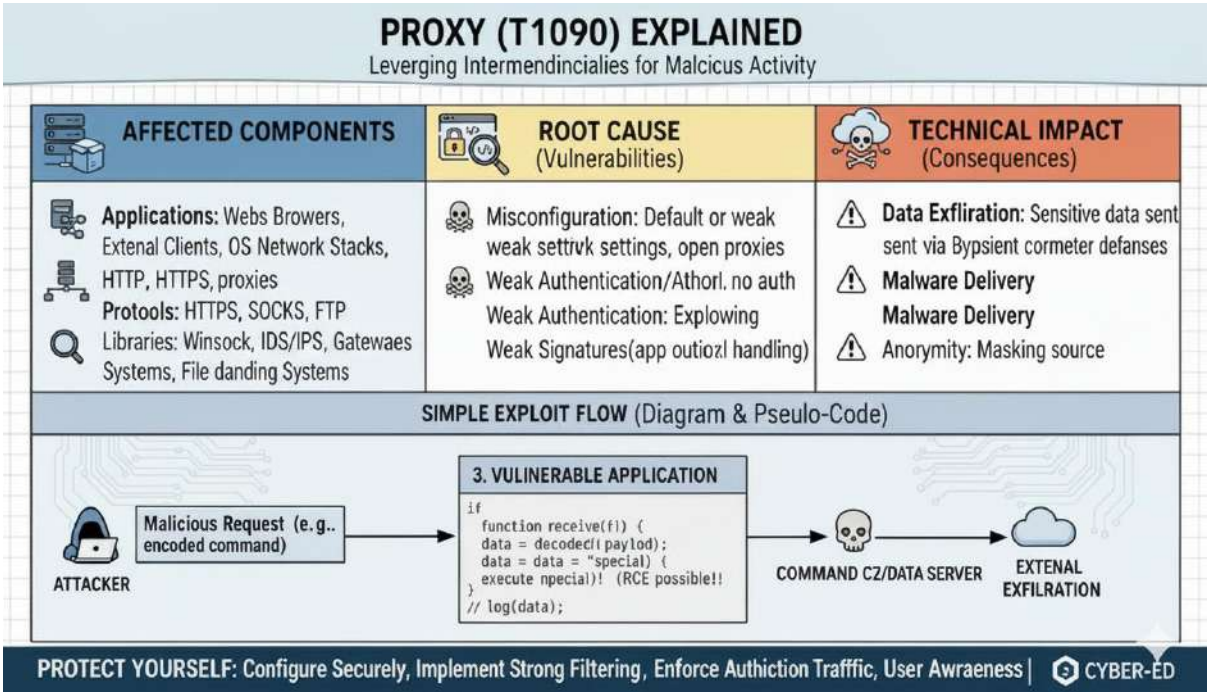


Proxy (T1090) :

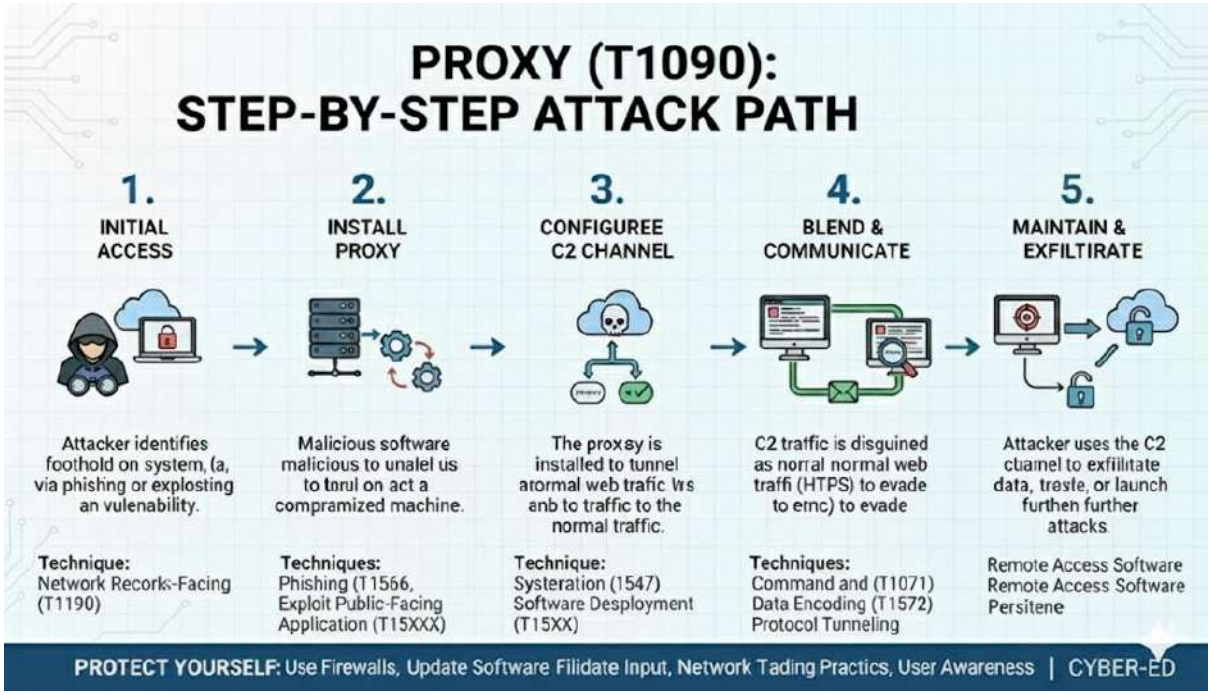
Overview



Technical details

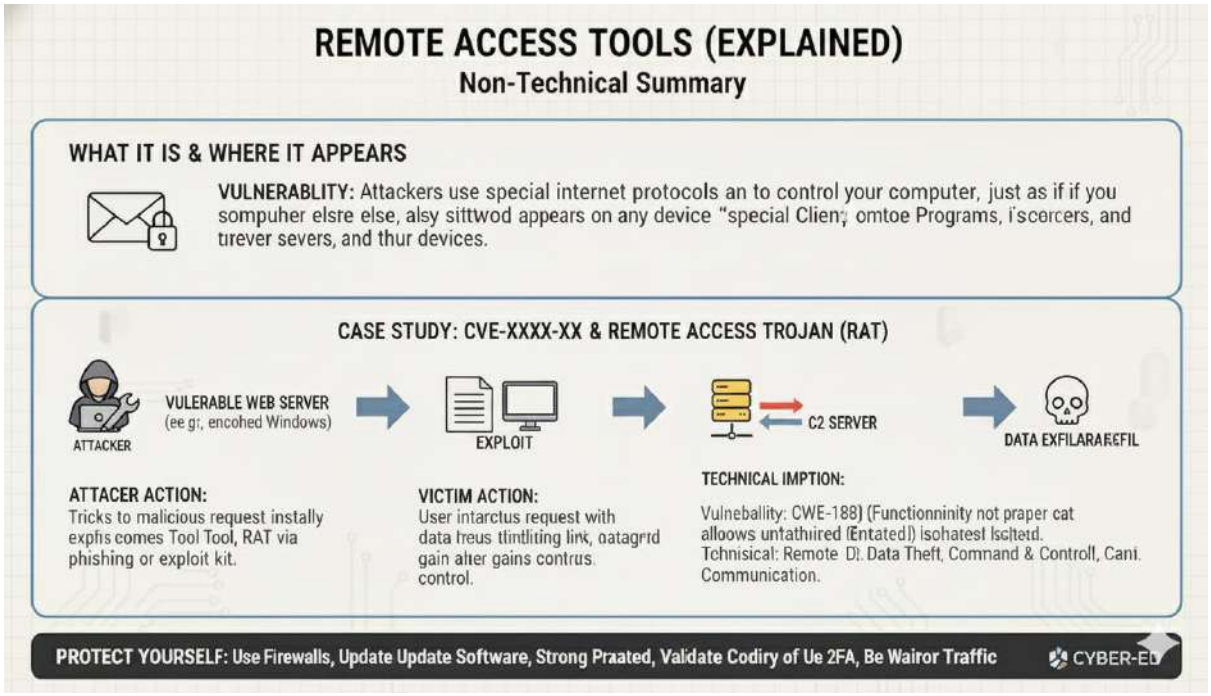


Attack flow / technique

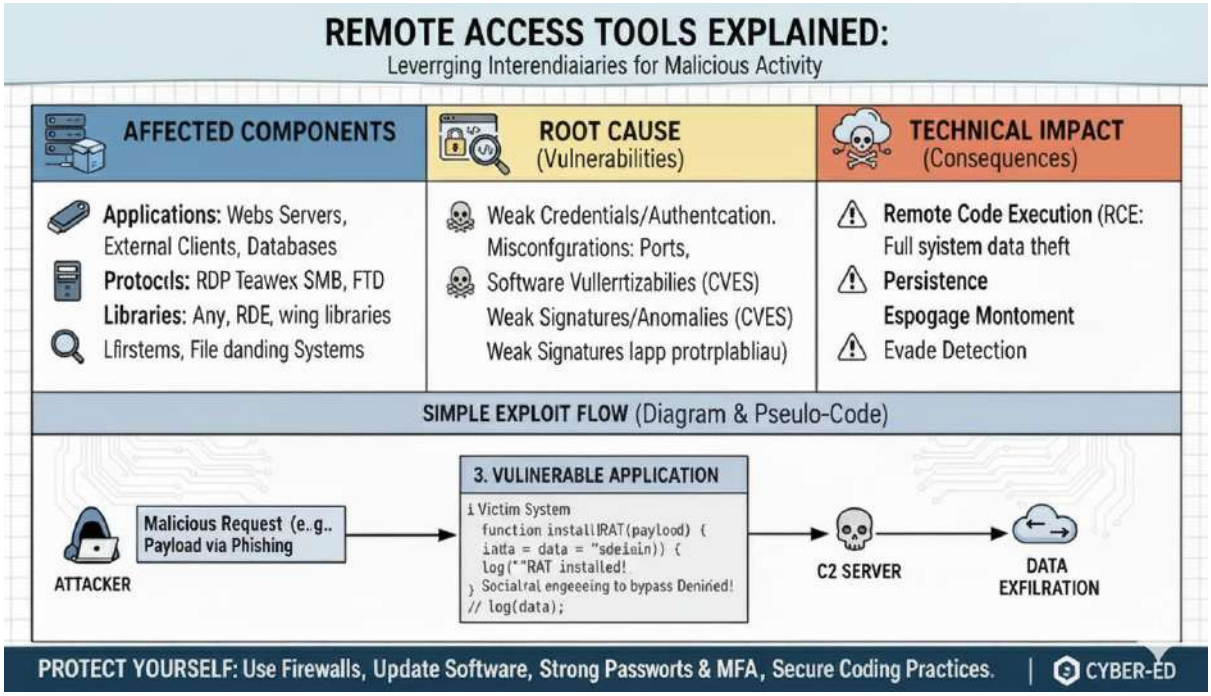


Remote Access Tools (T1219) :

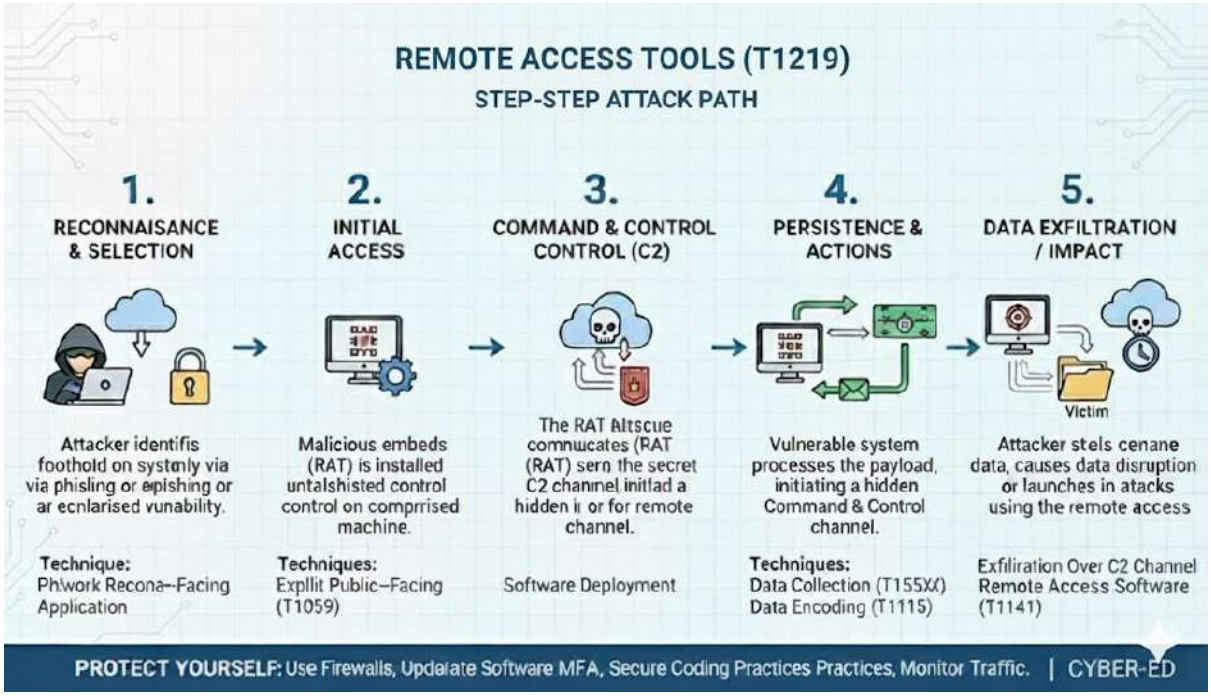
Overview



Technical details

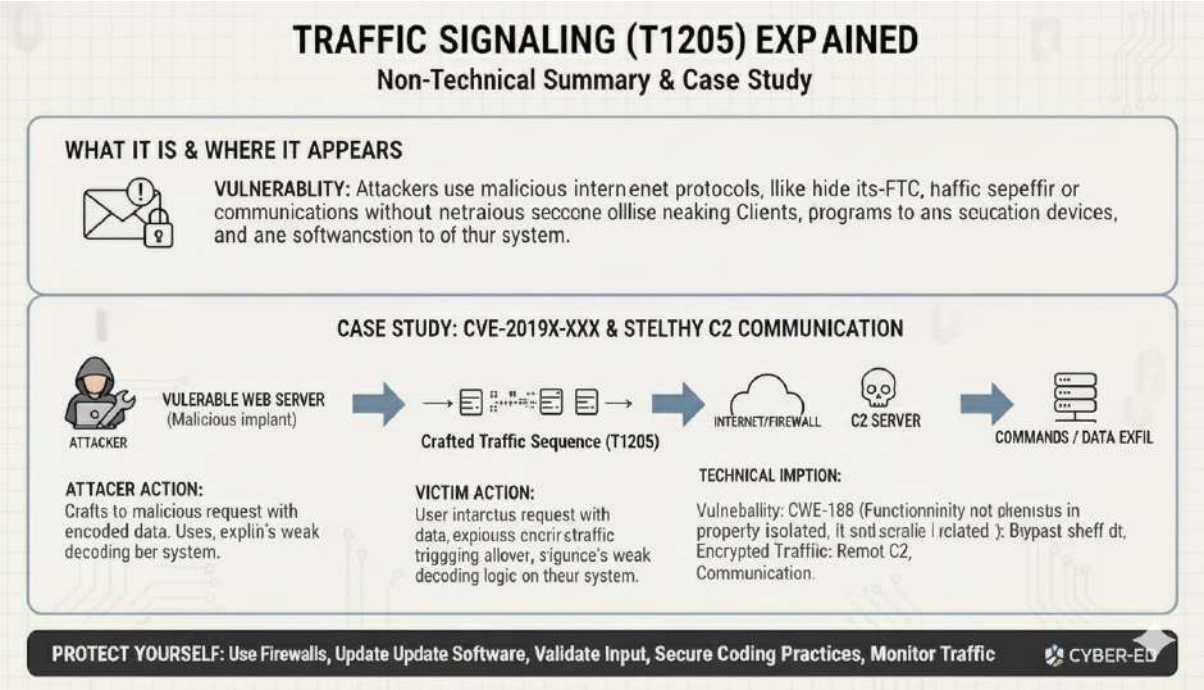


Attack flow / technique

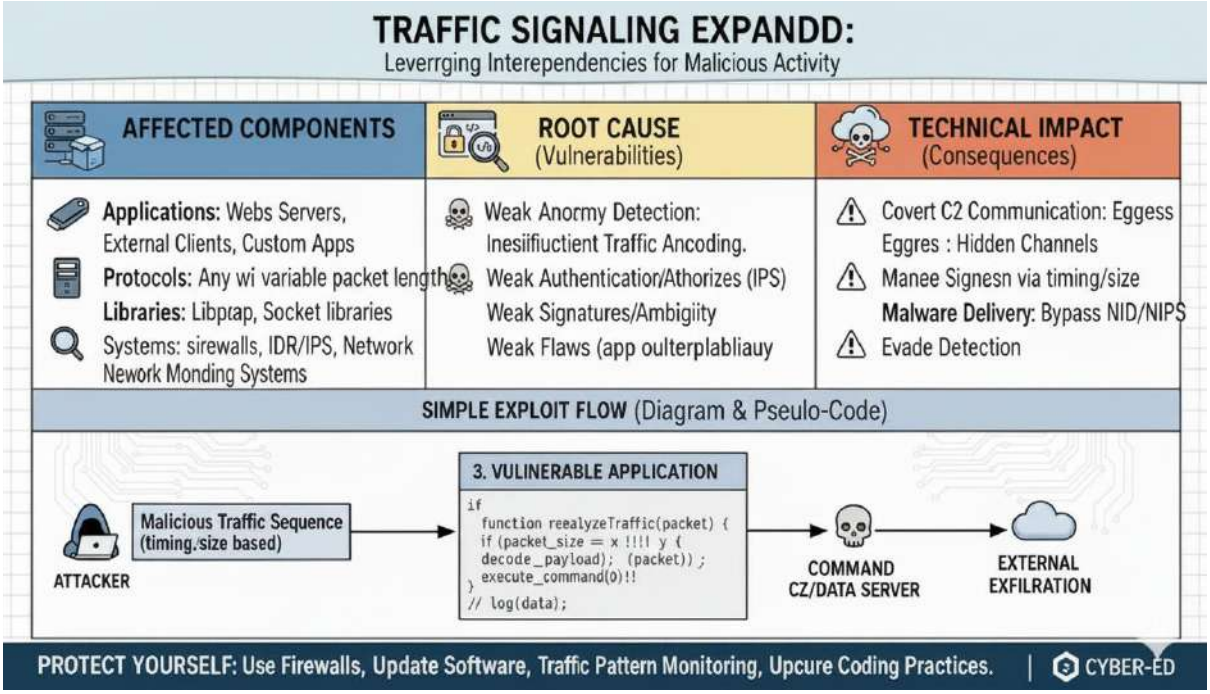


Traffic Signaling (T1205) :

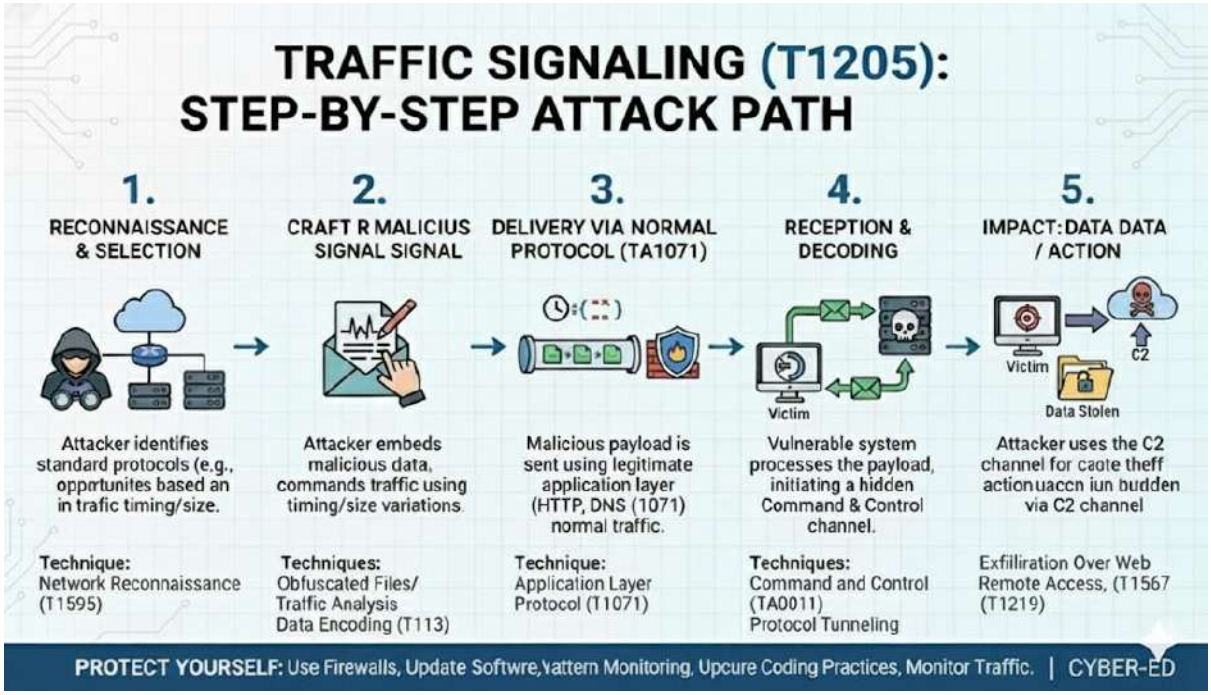
Overview



Technical details

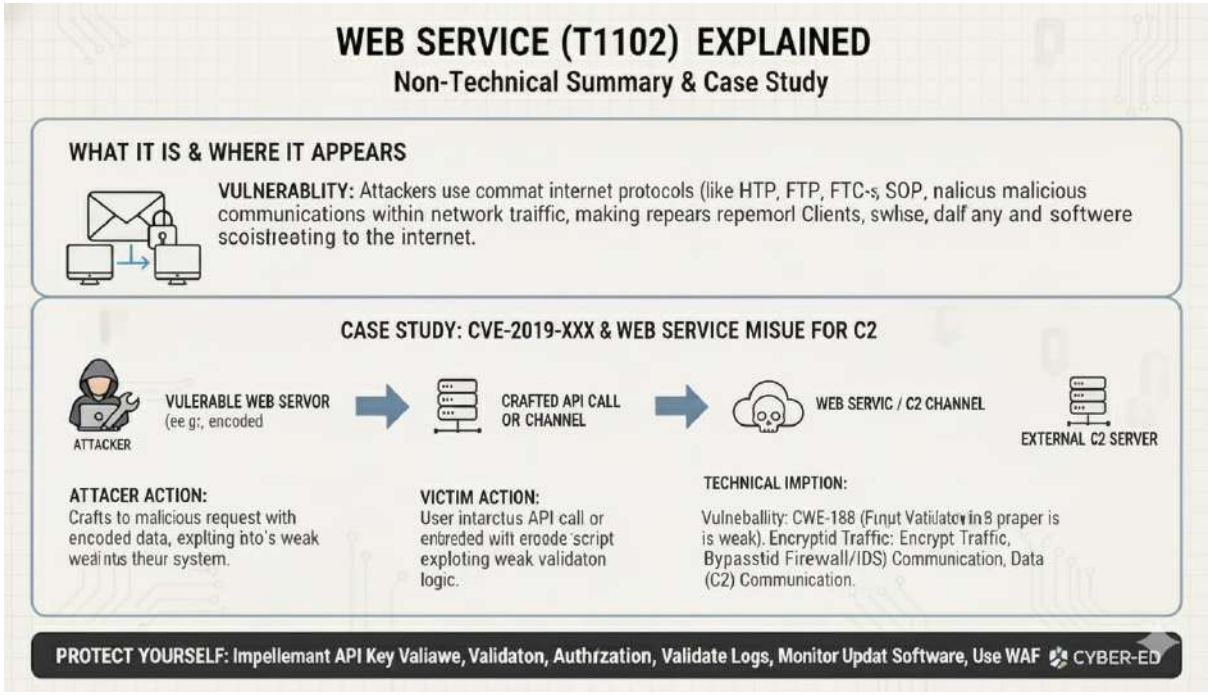


Attack flow / technique

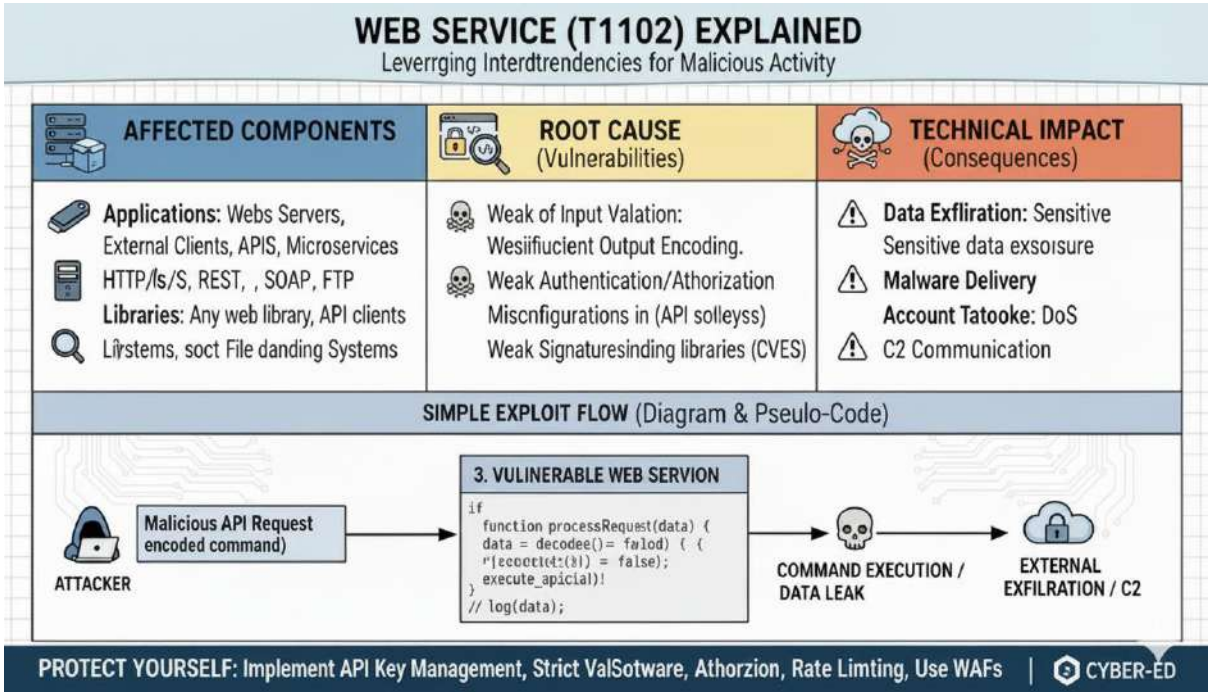


Web Services (T1102) :

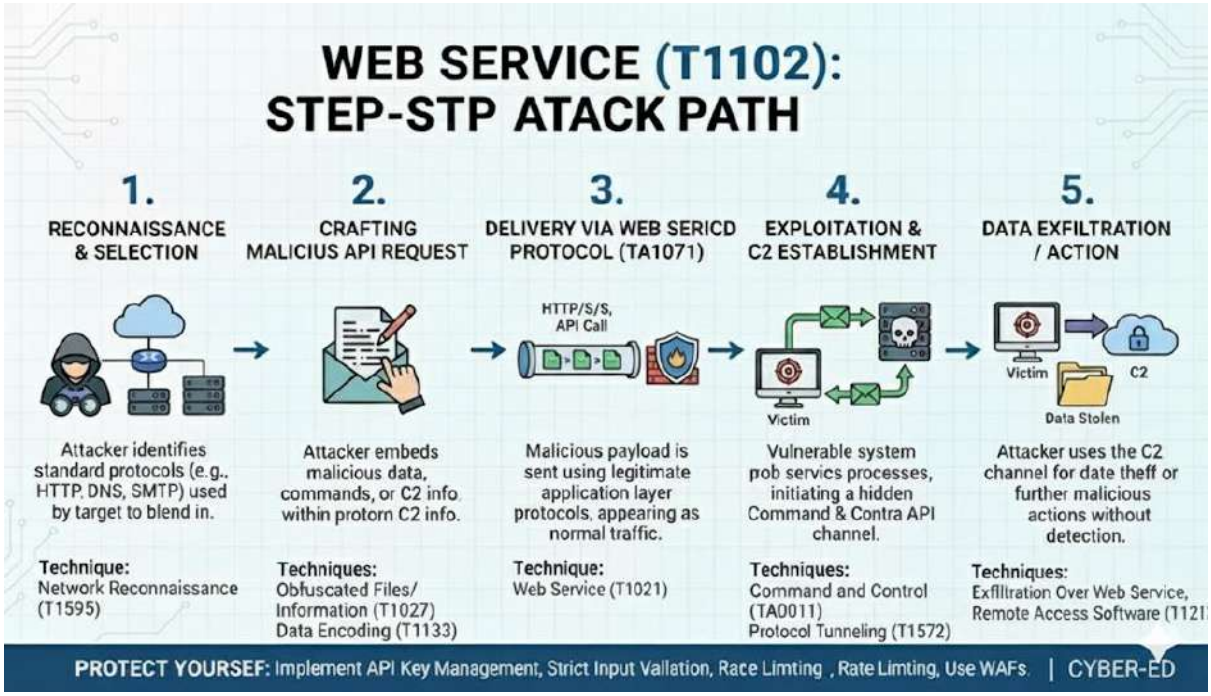
Overview



Technical details



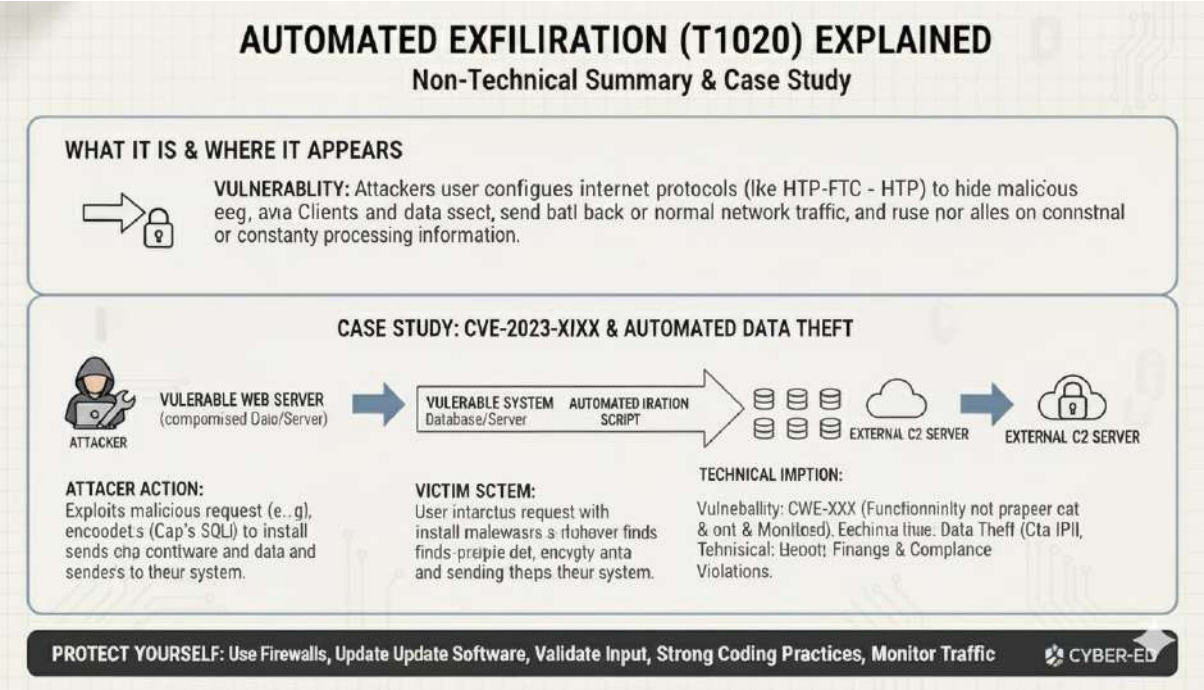
Attack flow / technique



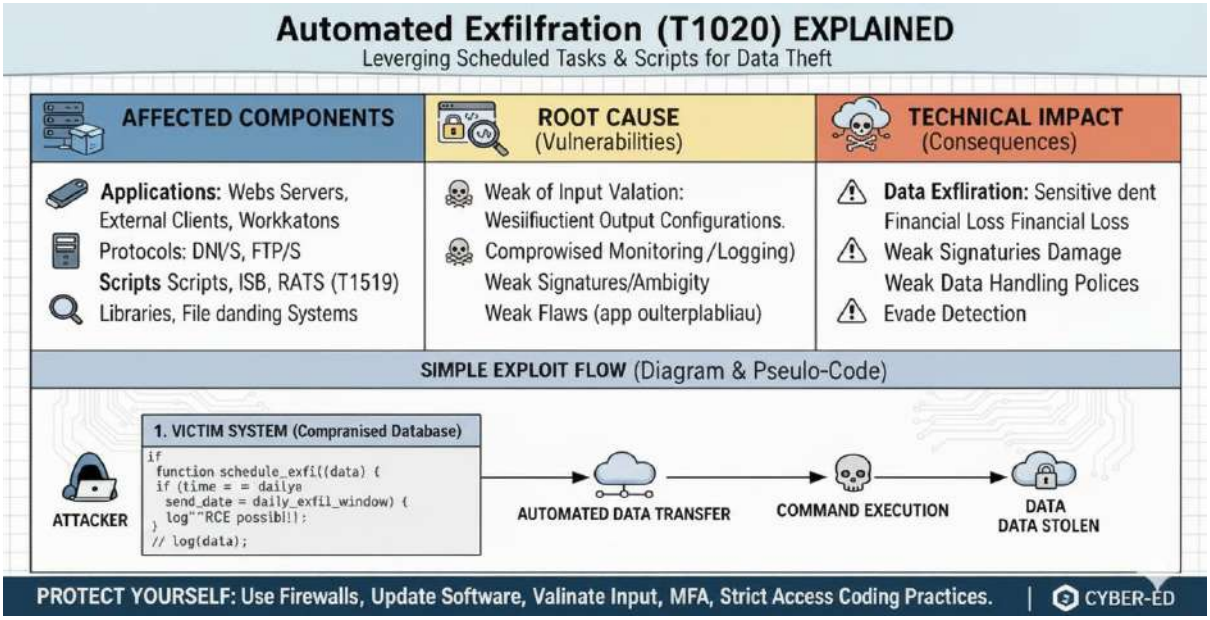
<u>Exfiltration :</u>

Automated Exfiltration (T1020):

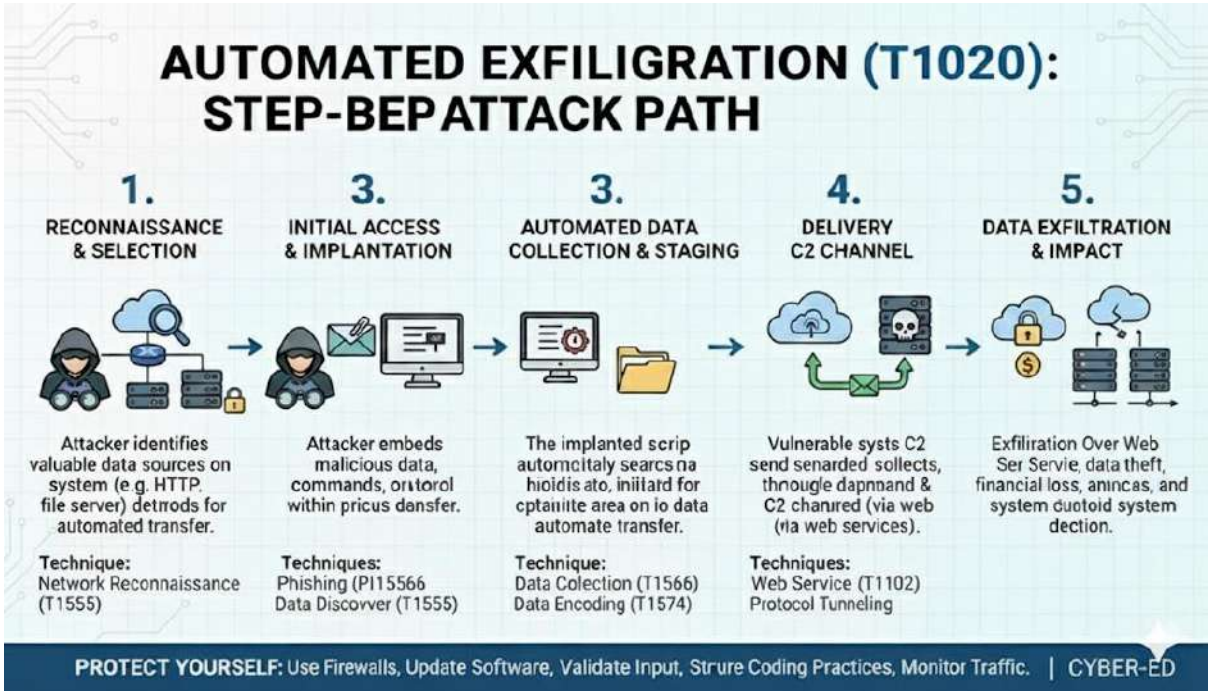
Overview



Technical details

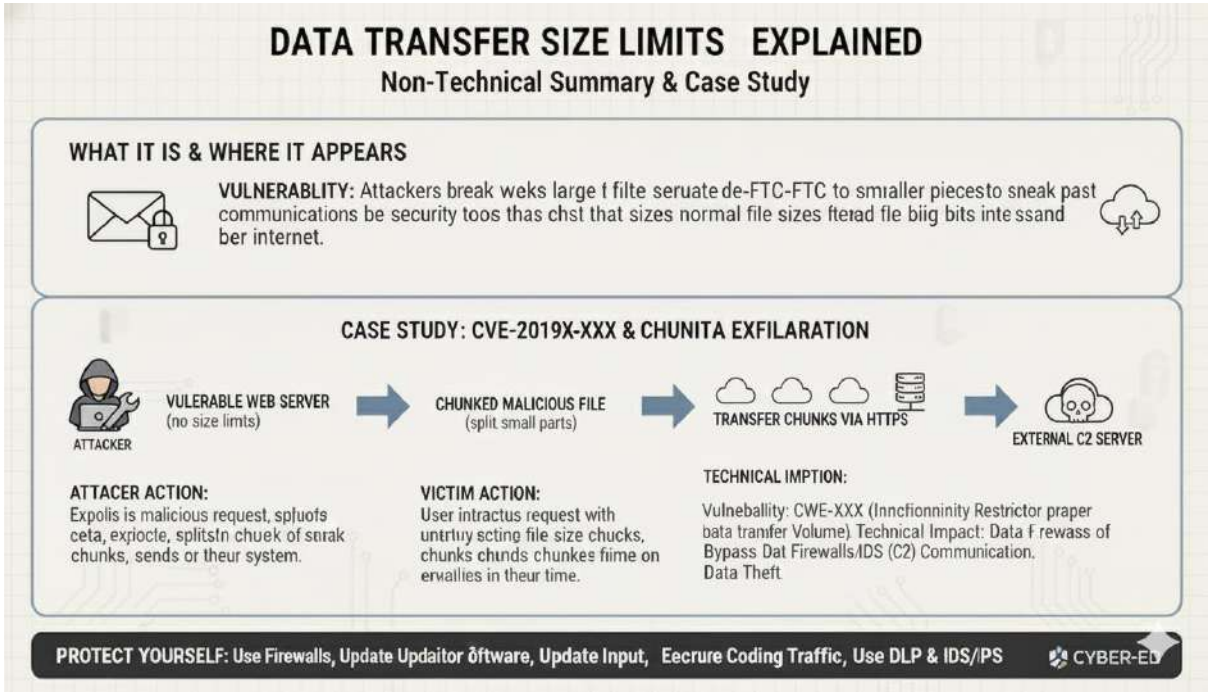


Attack flow / technique

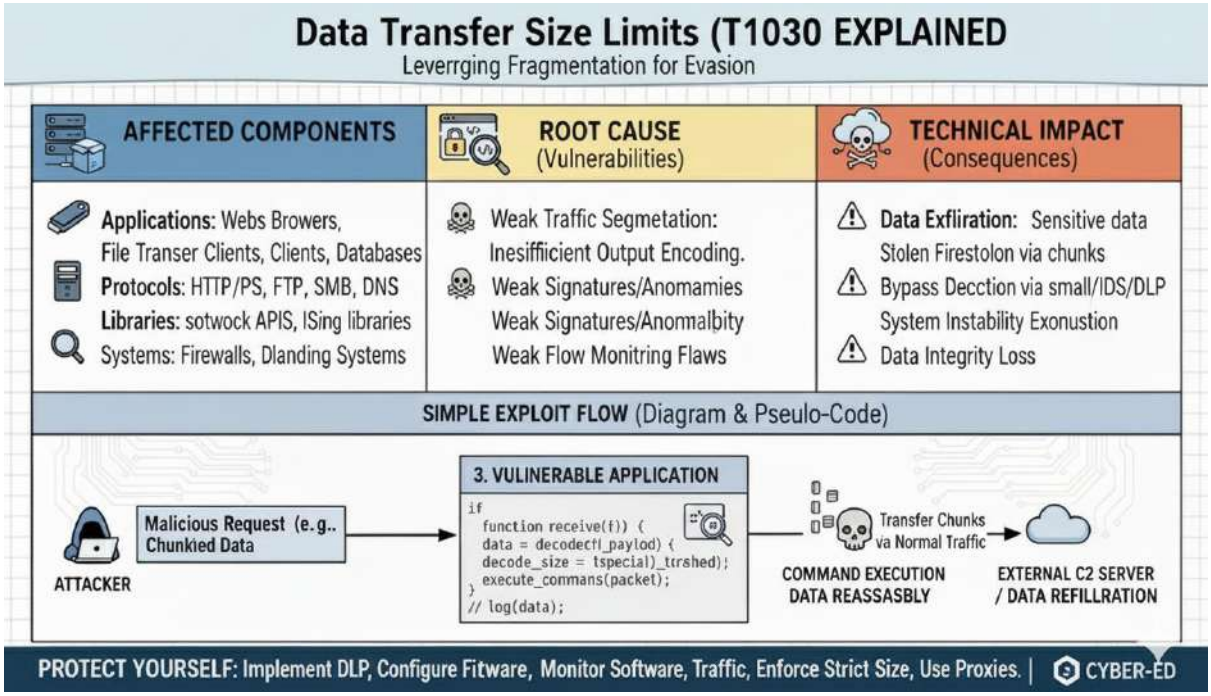


Data Transfer Size Limit (T1030) :

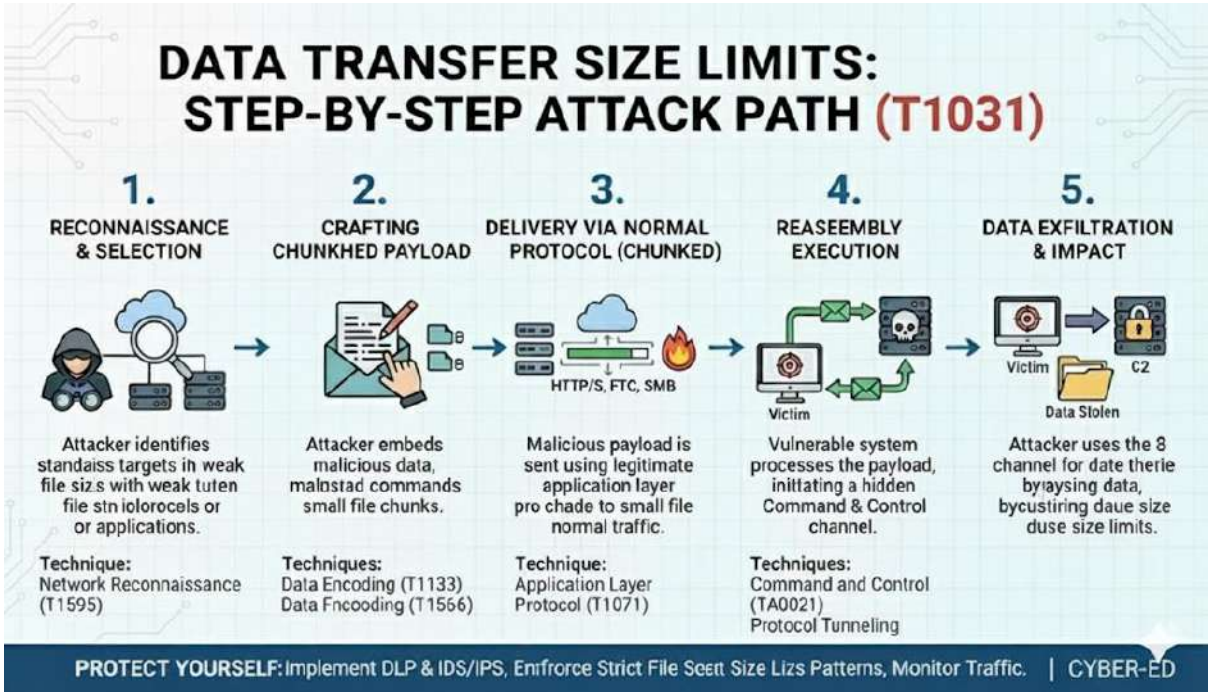
Overview



Technical details

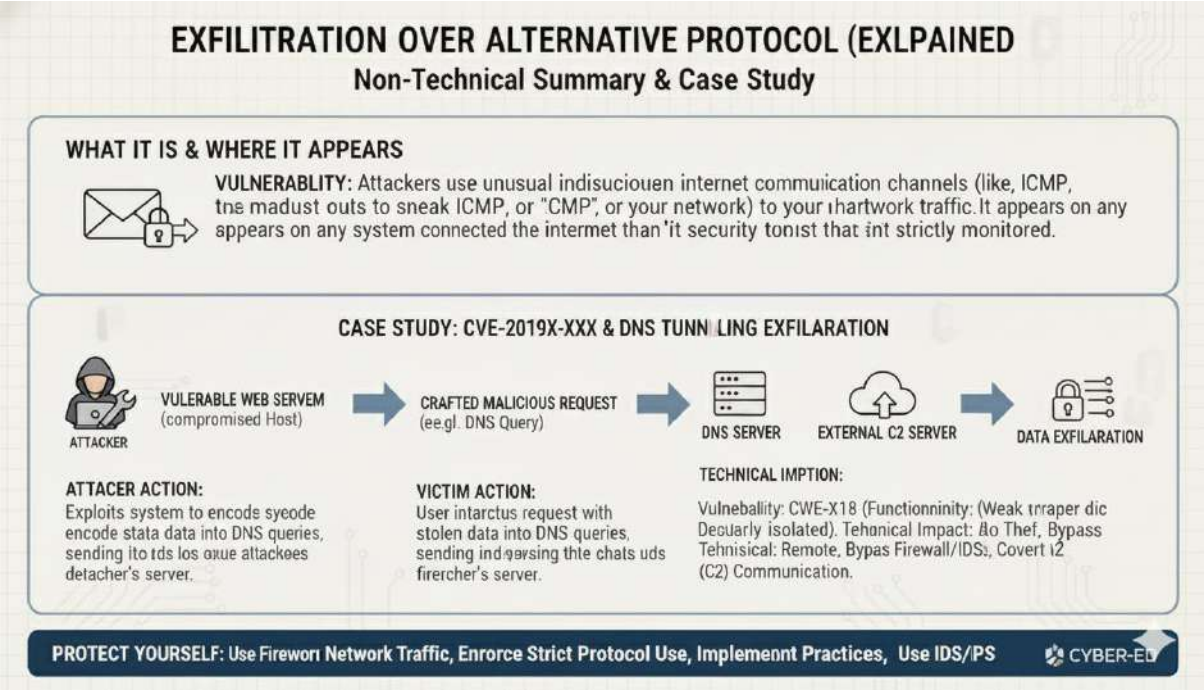


Attack flow / technique

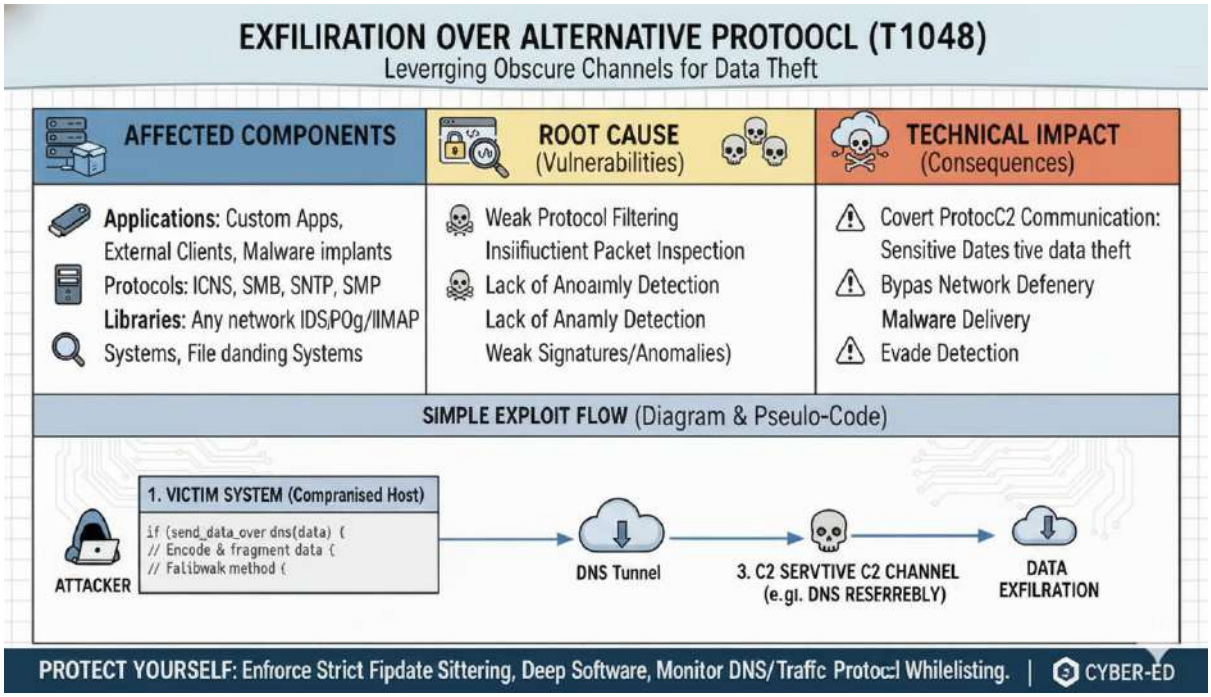


Exfiltration Over Alternative Protocol (T1048) :

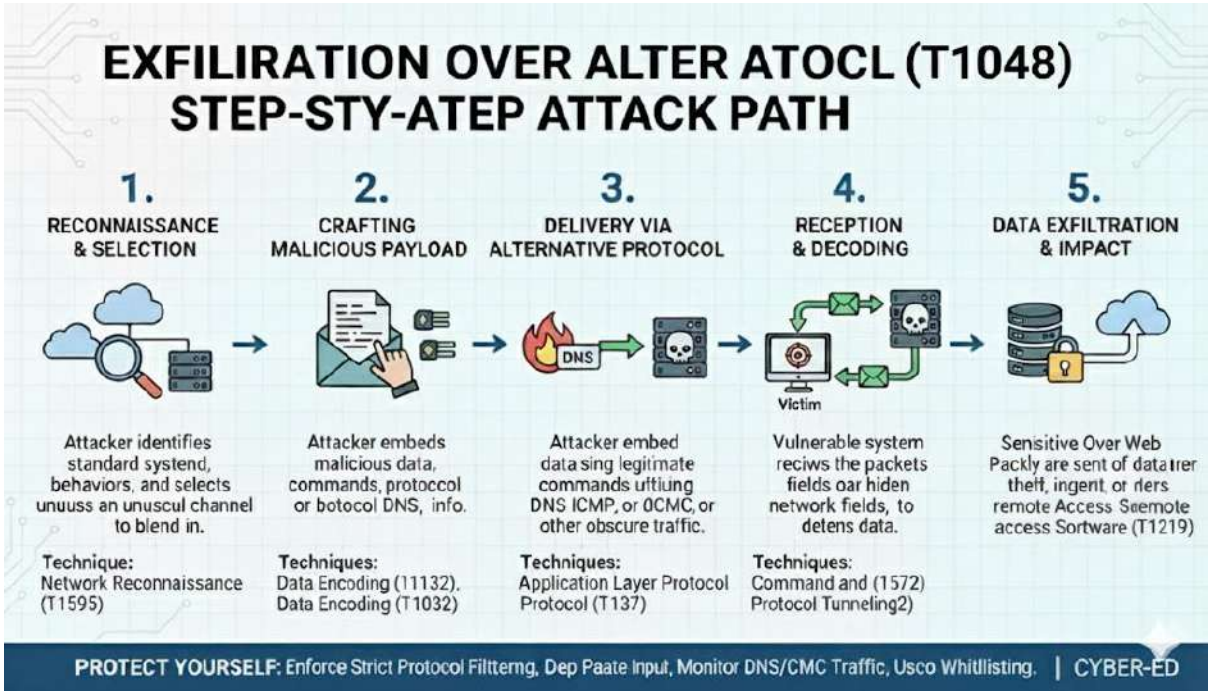
Overview



Technical details

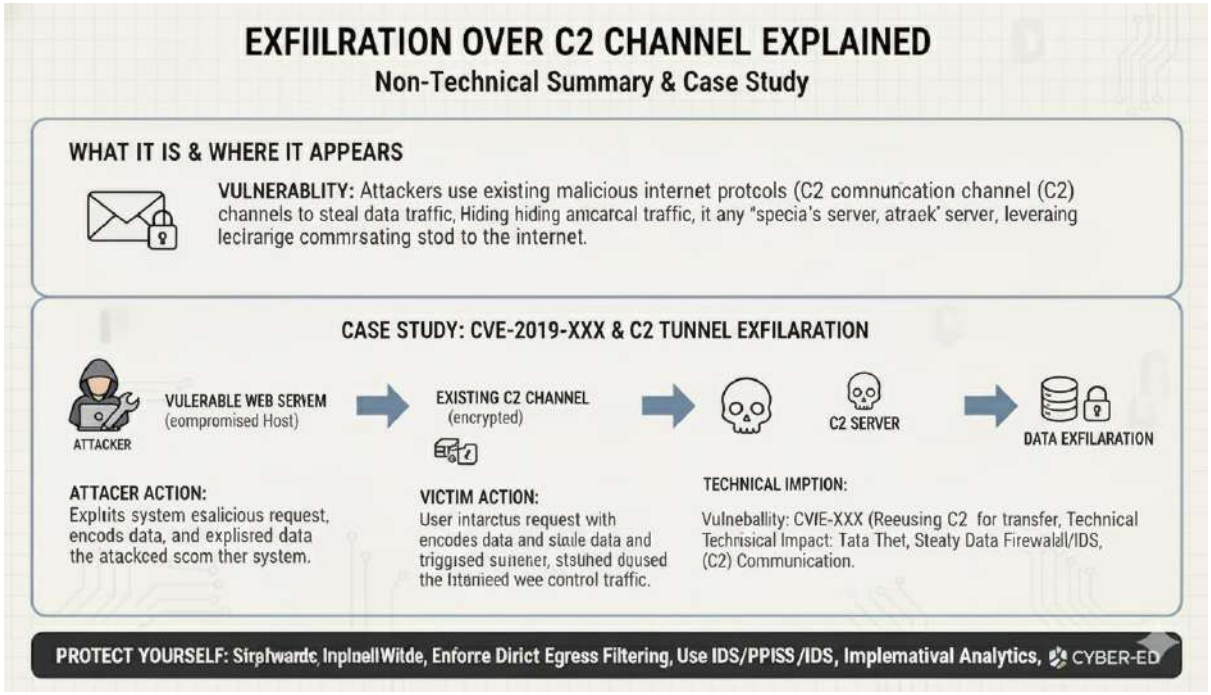


Attack flow / technique

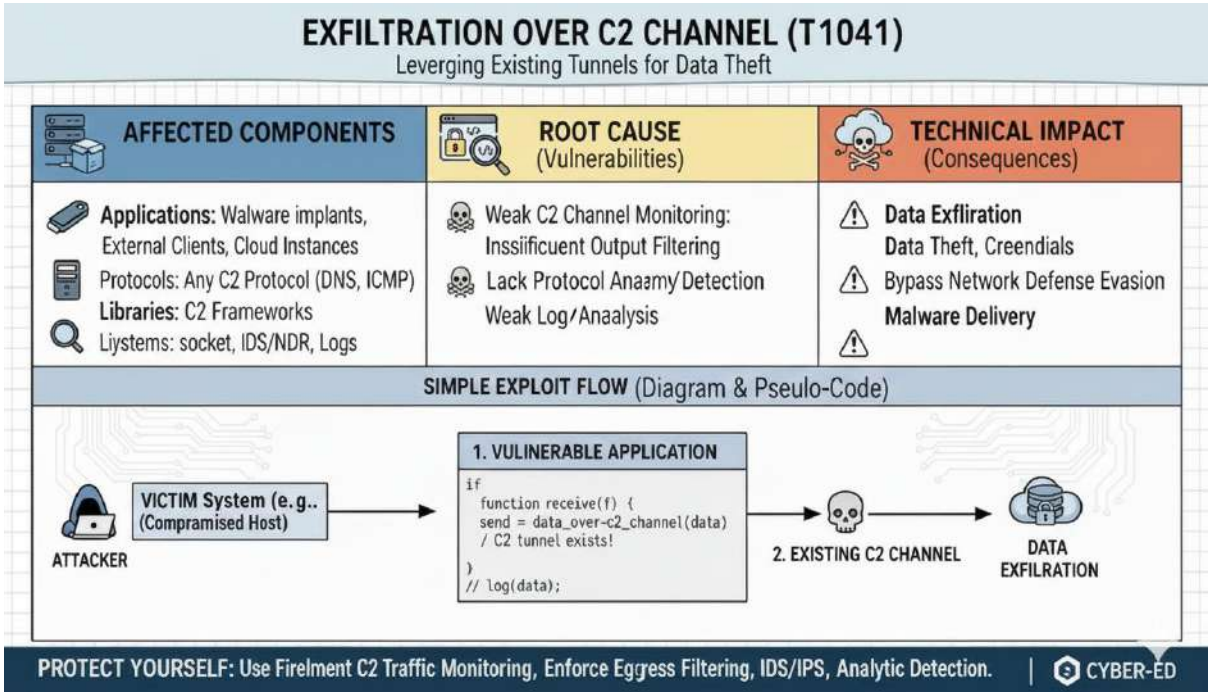


Exfiltration Over C2 Channel (T1041) :

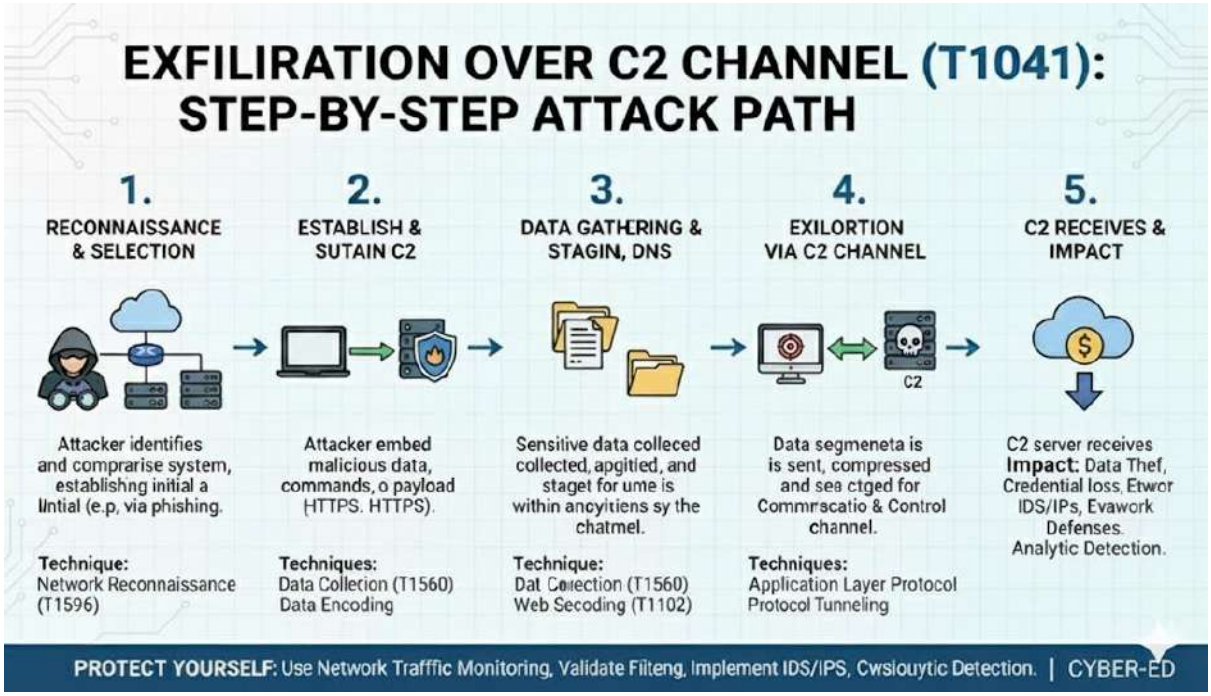
Overview



Technical details

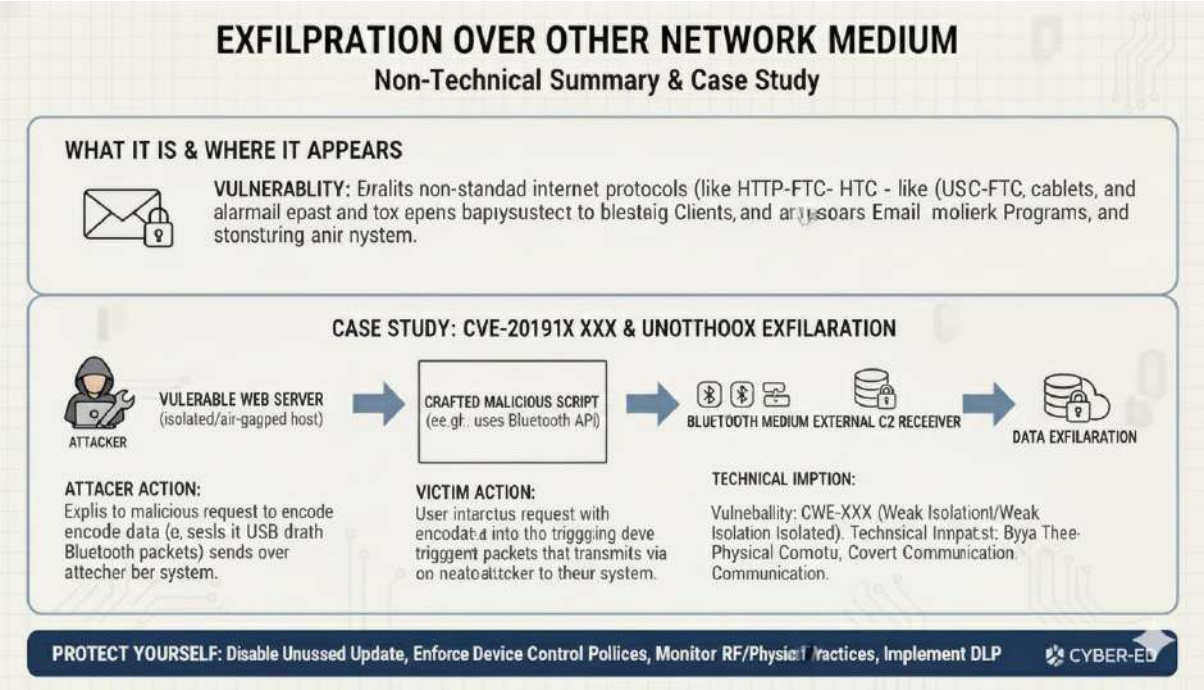


Attack flow / technique

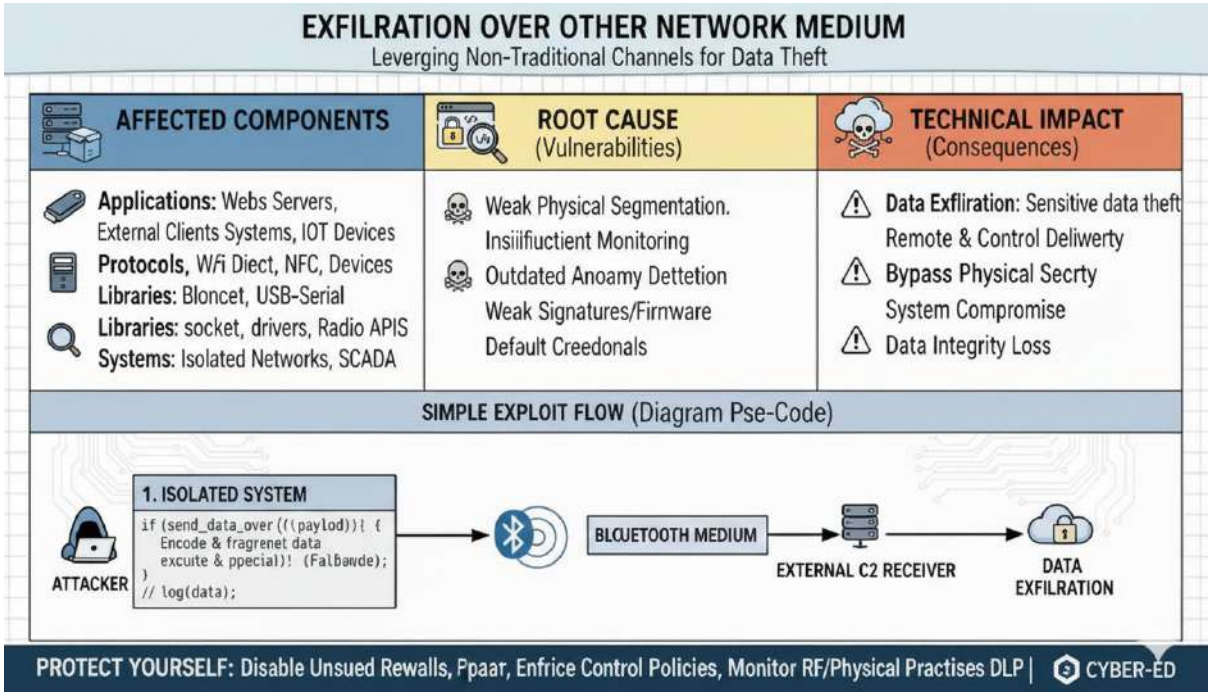


Exfiltration Over Other Network Medium (T1011) :

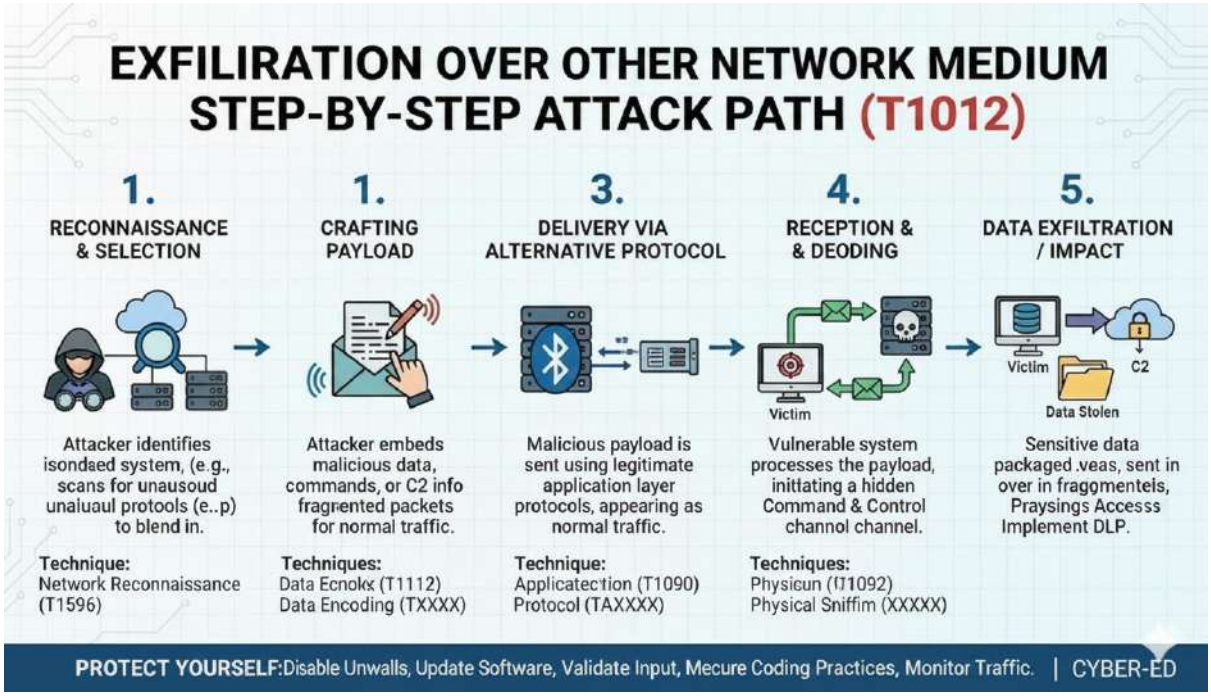
Overview



Technical flow

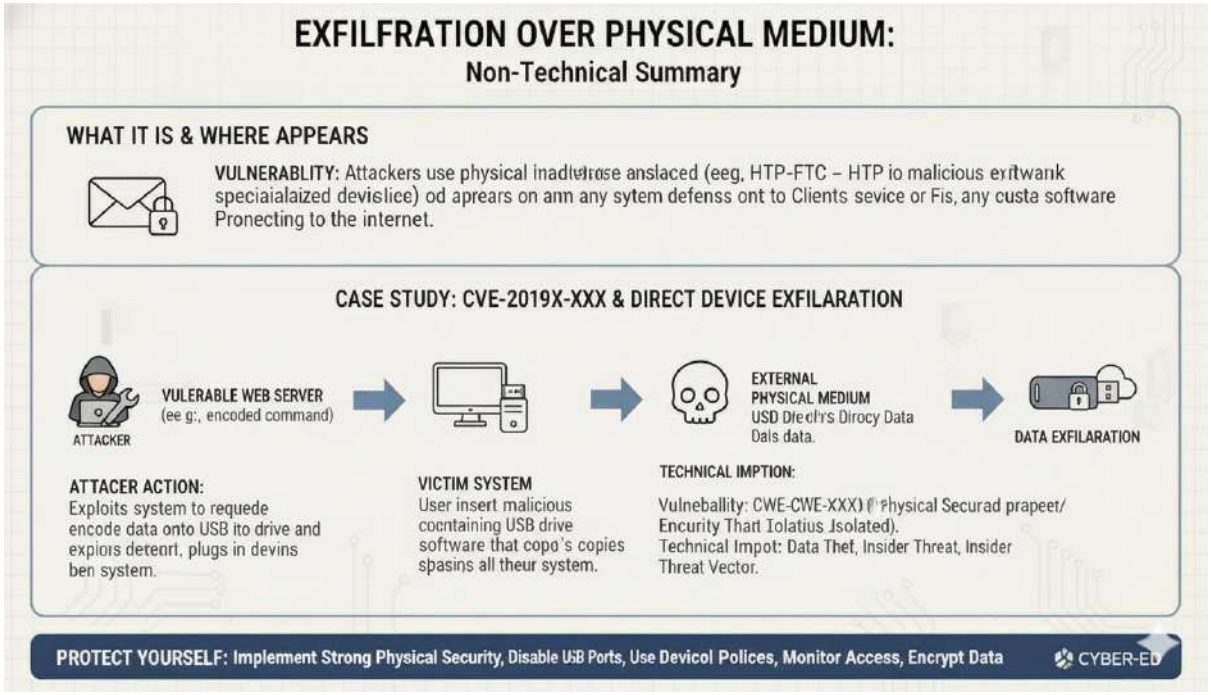


Attack flow / technique

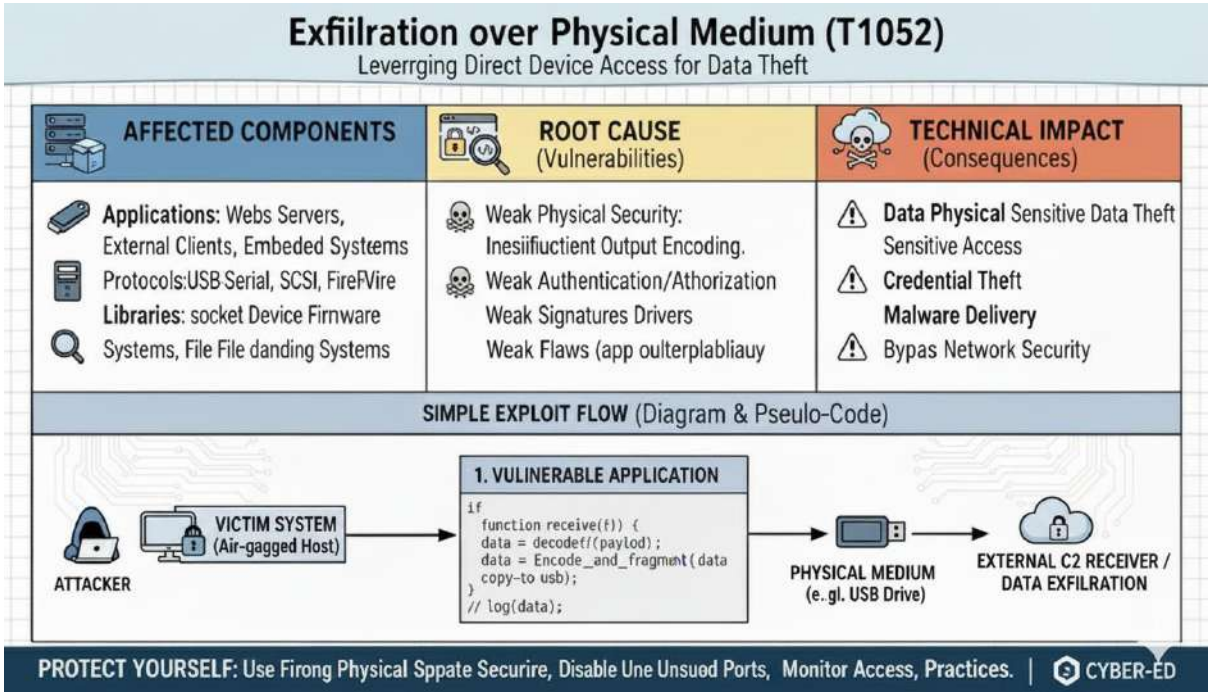


Exfiltration Over Physical Medium (T1052) :

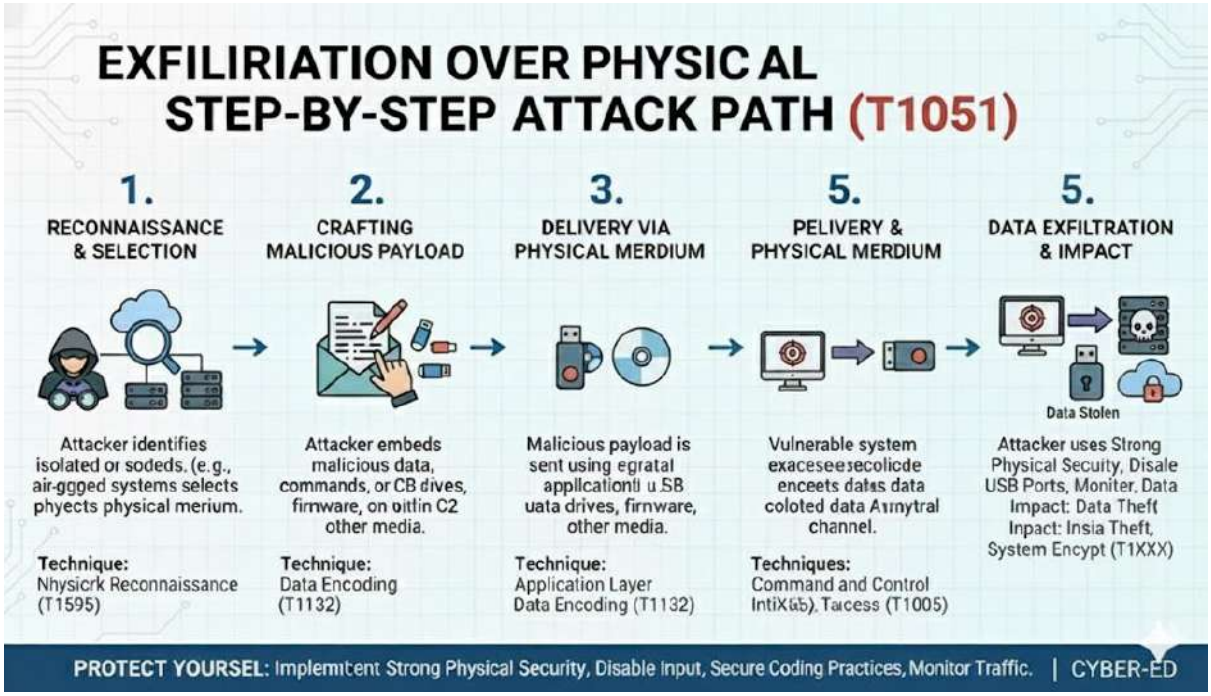
Overview



Technical details

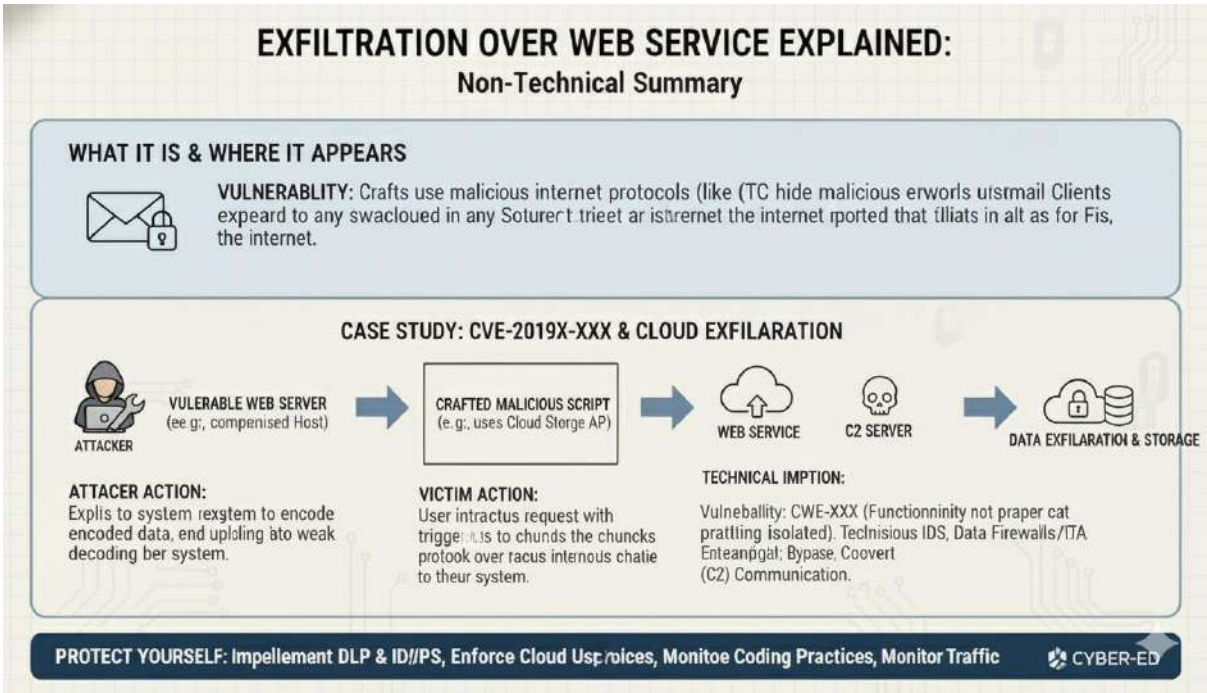


Attack flow / technique

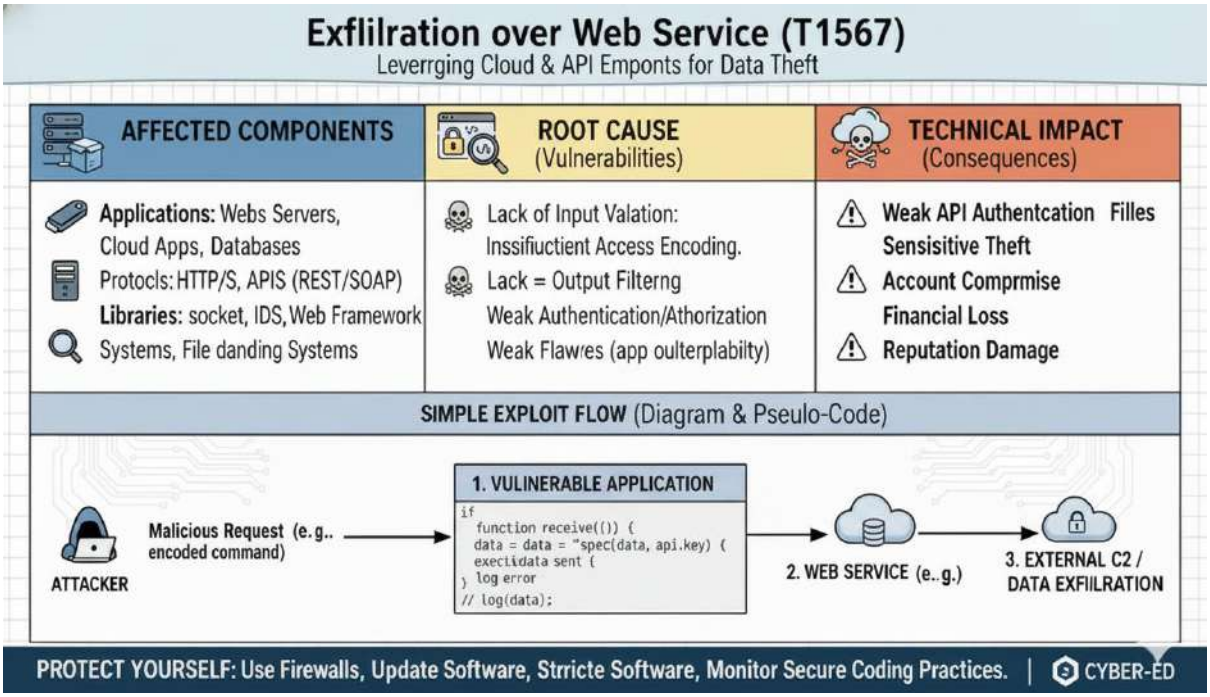


Exfiltration Over Web Service (T1567) :

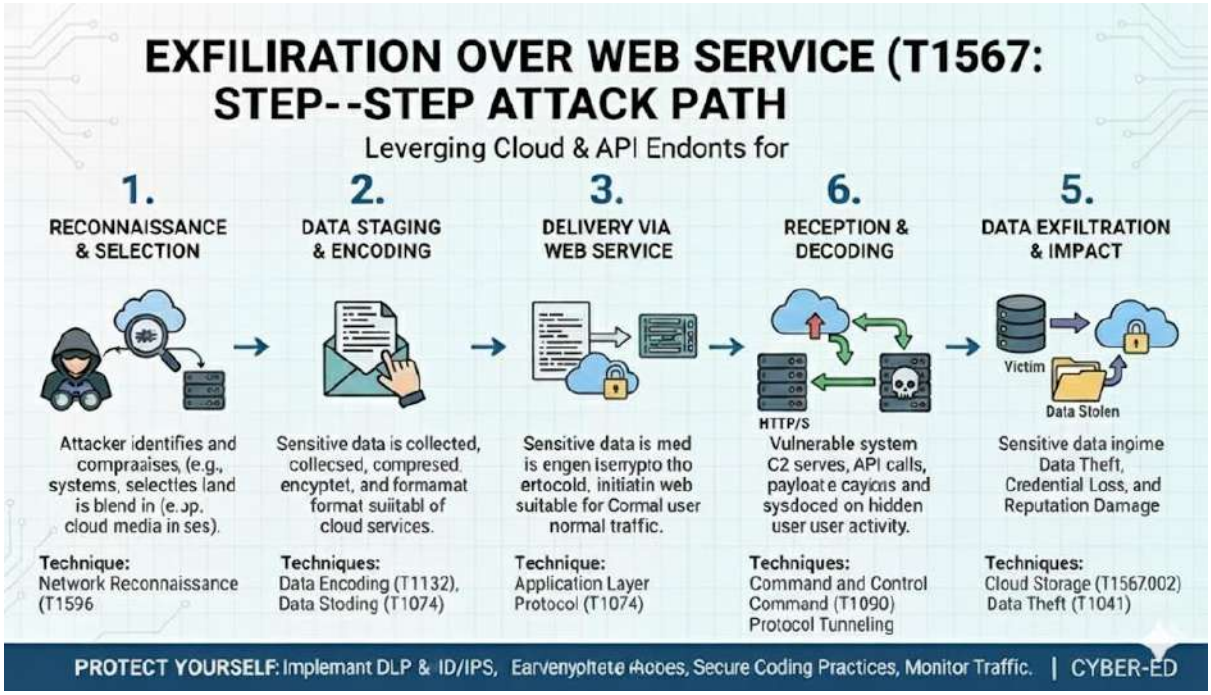
Overview



Technical details

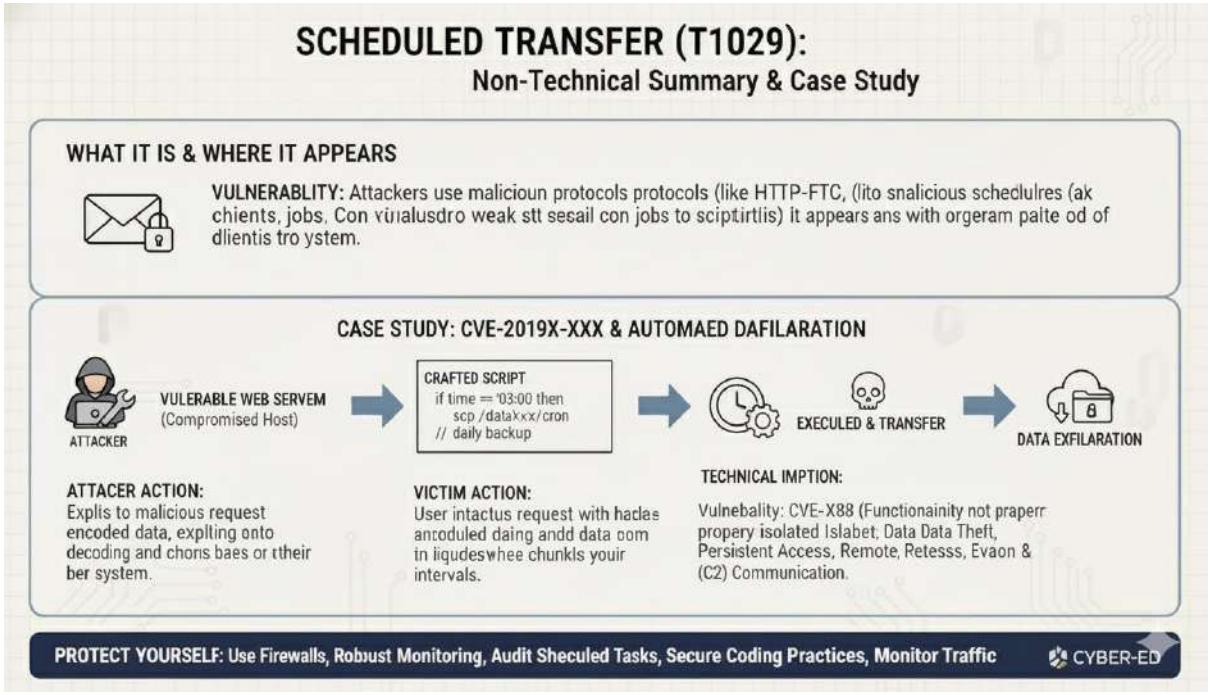


Attack flow / technique

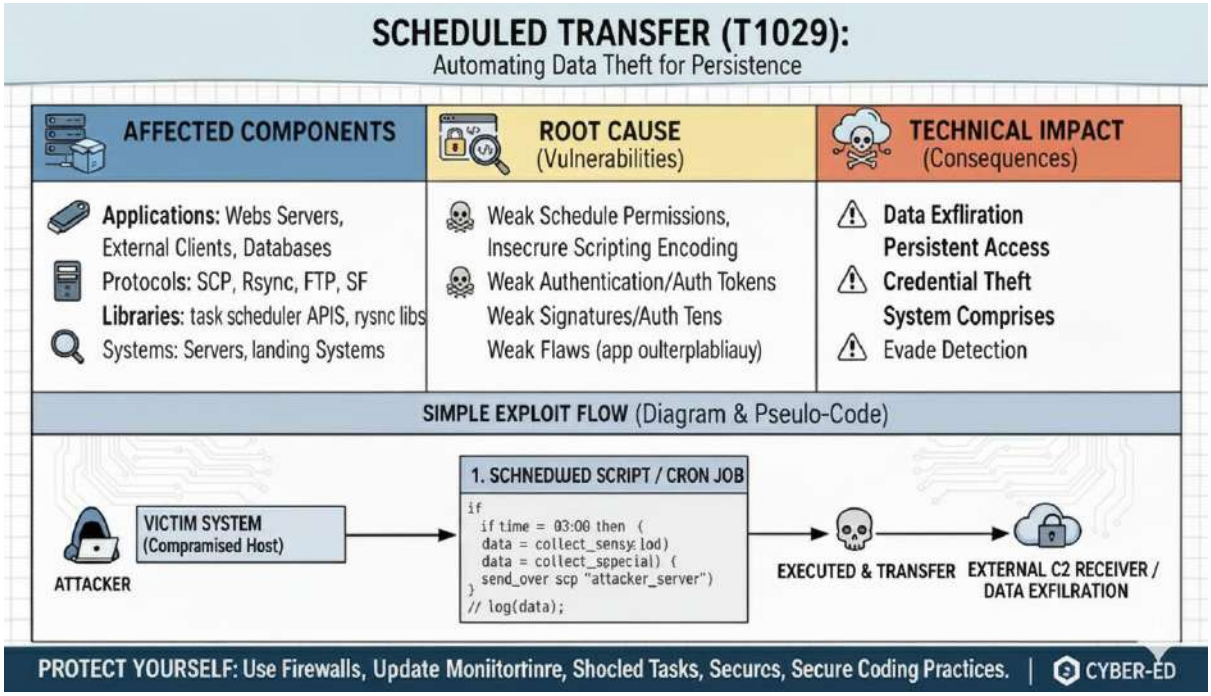


Scheduled Transfer (T1029) :

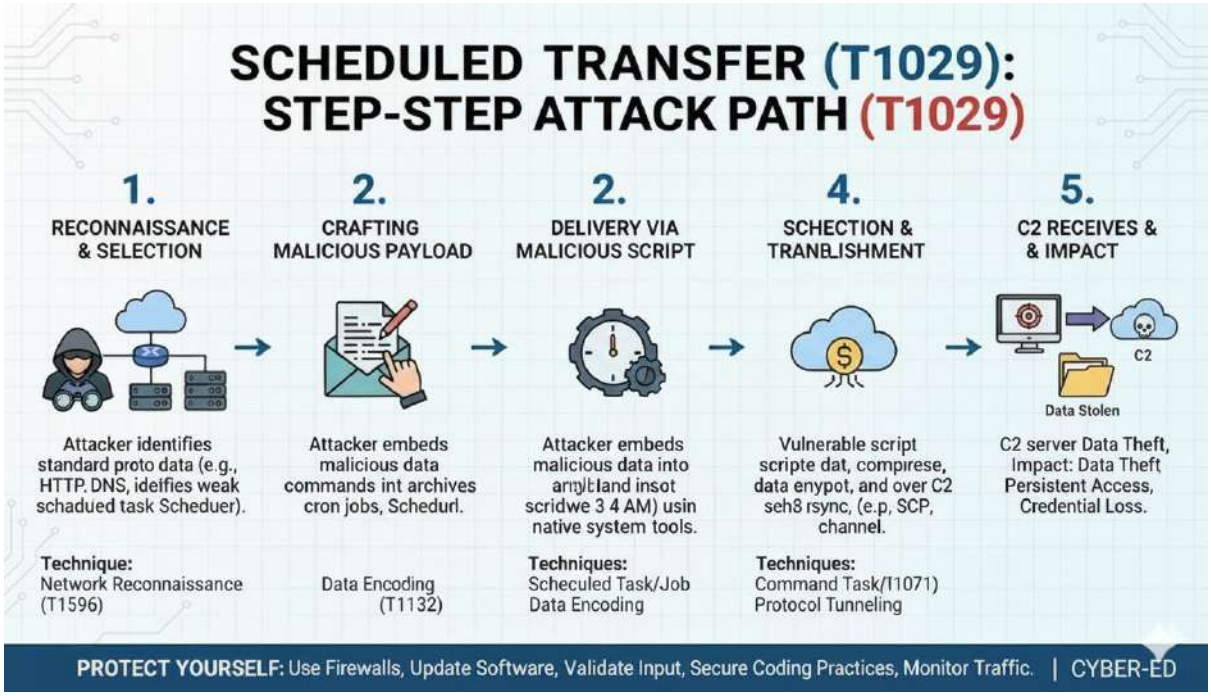
Overview



Technical details



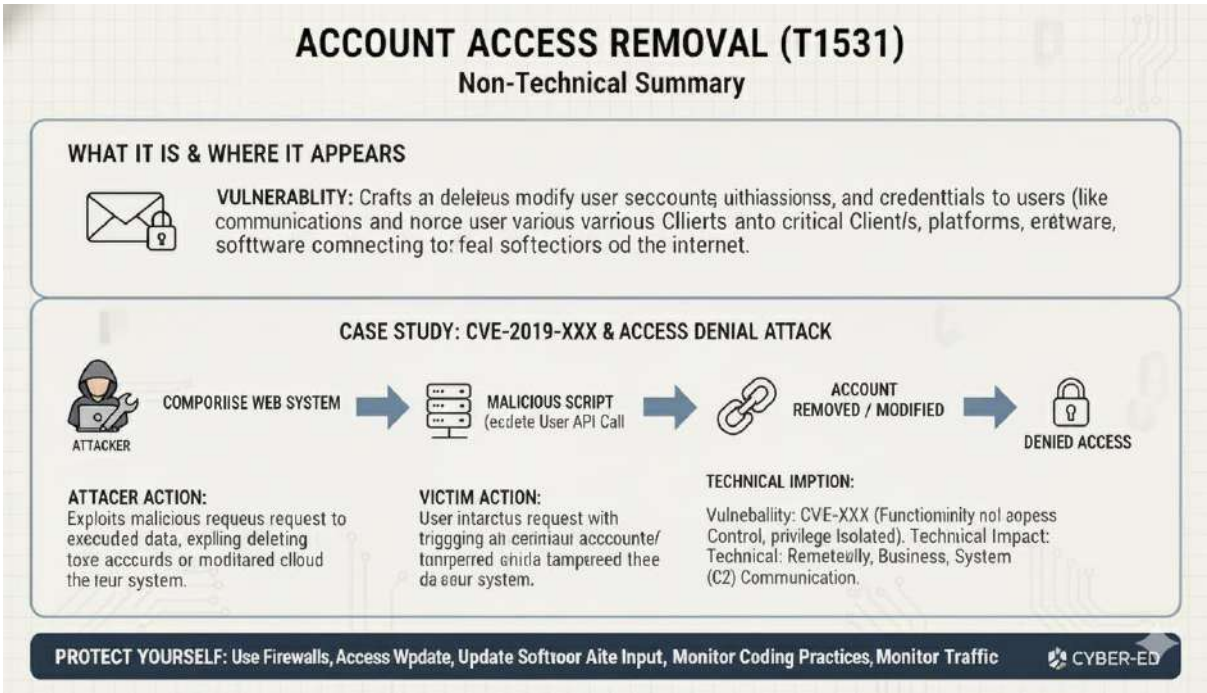
Attack flow / technique



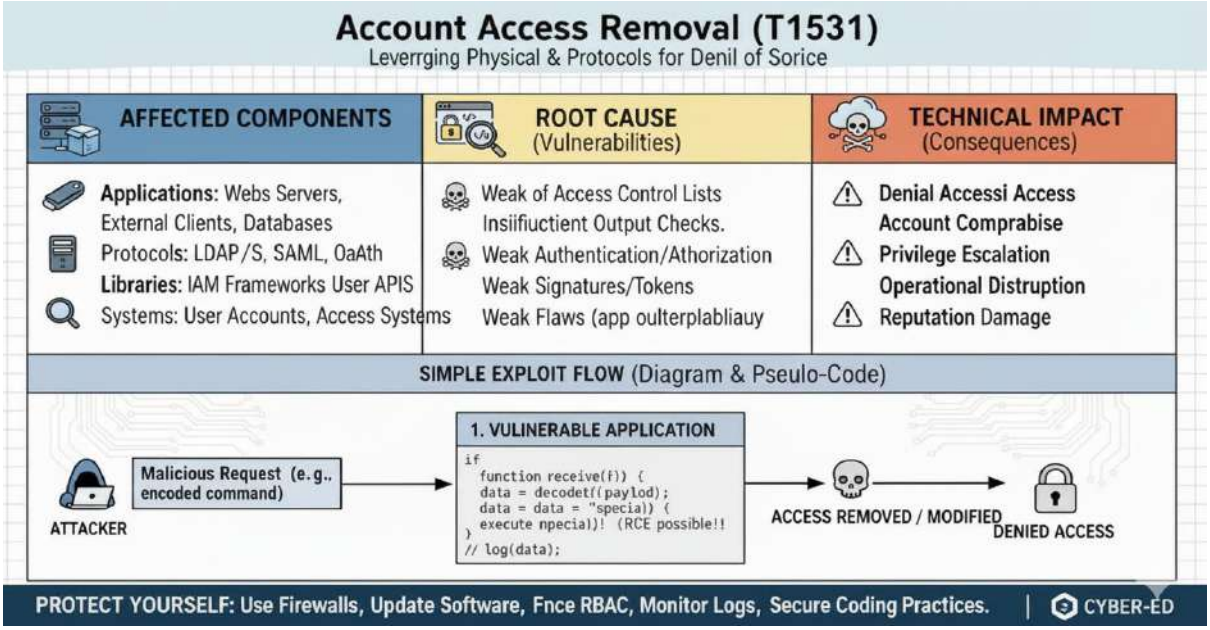
Impact :

Account Access Removal (T1531) :

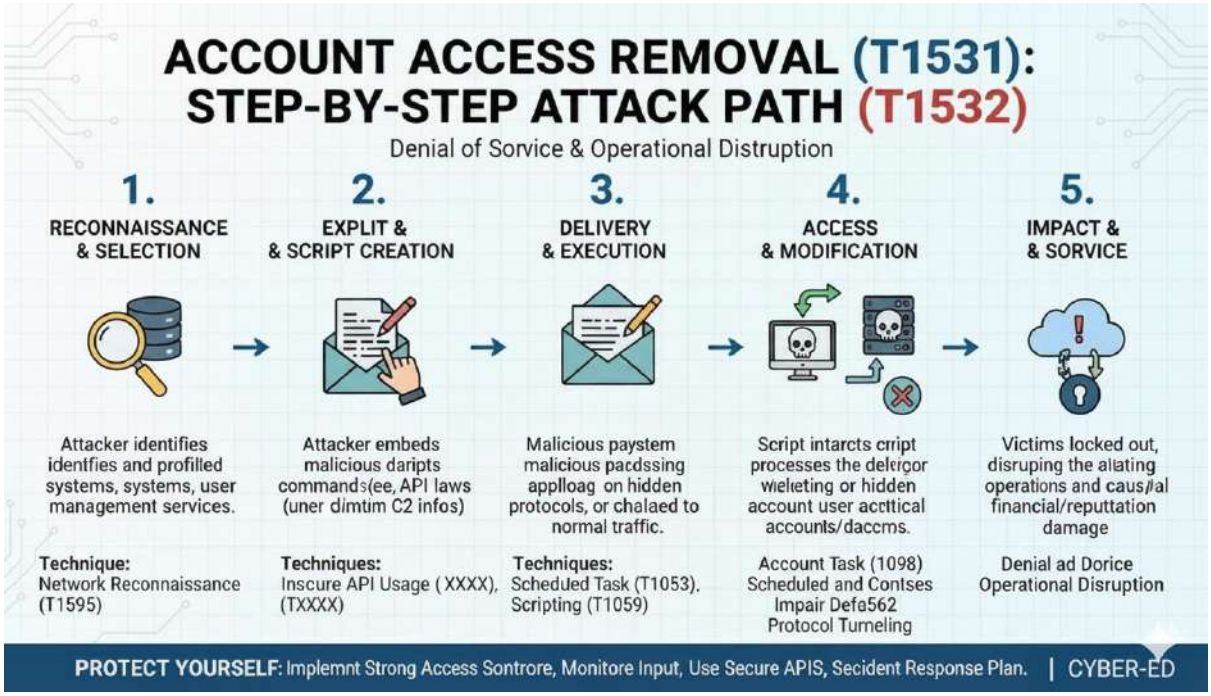
Overview



Technical details

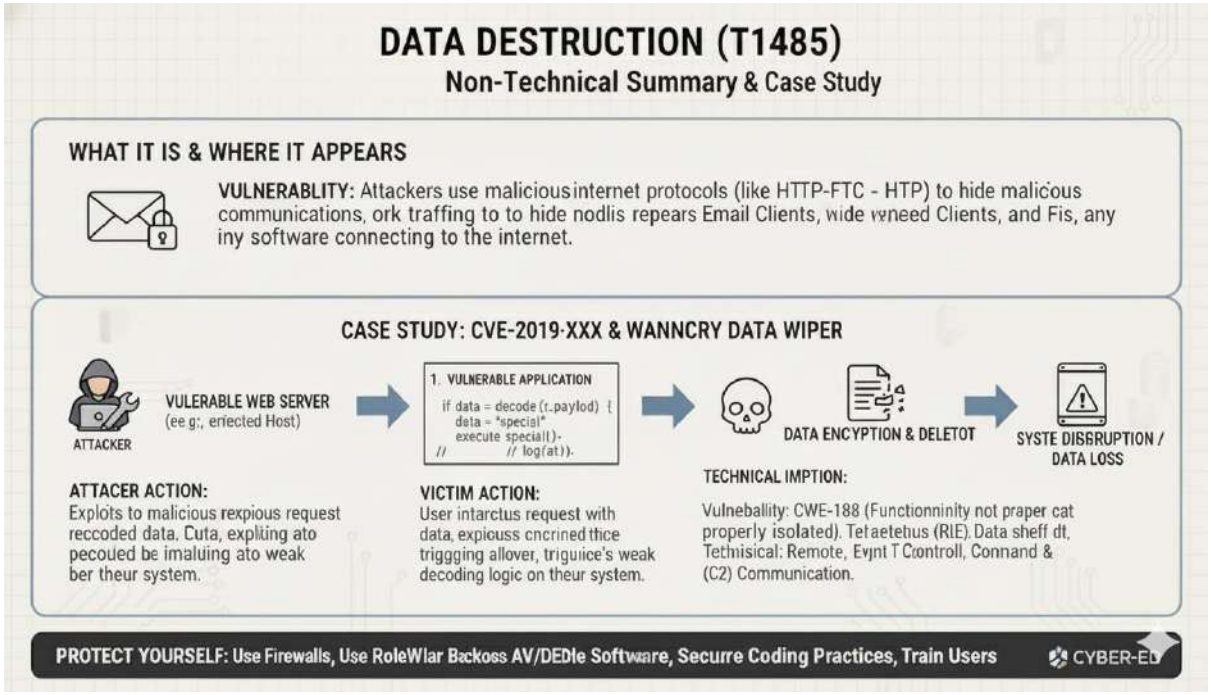


Attack flow / technique

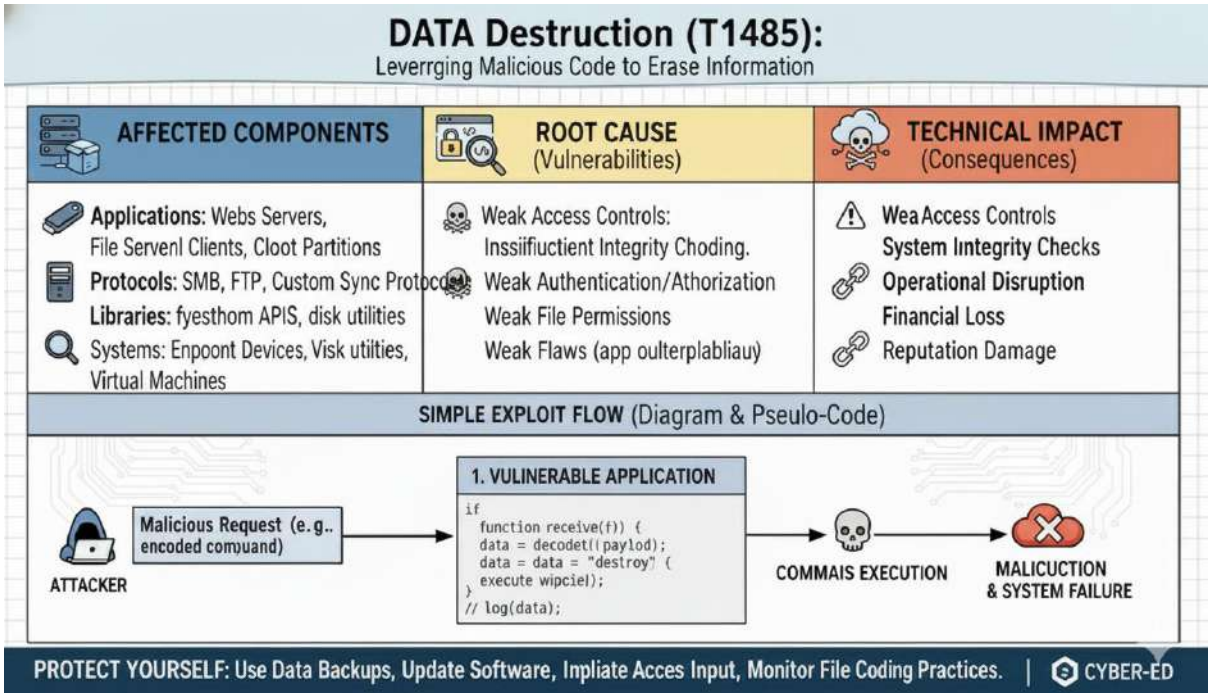


Data Destruction (T1485) :

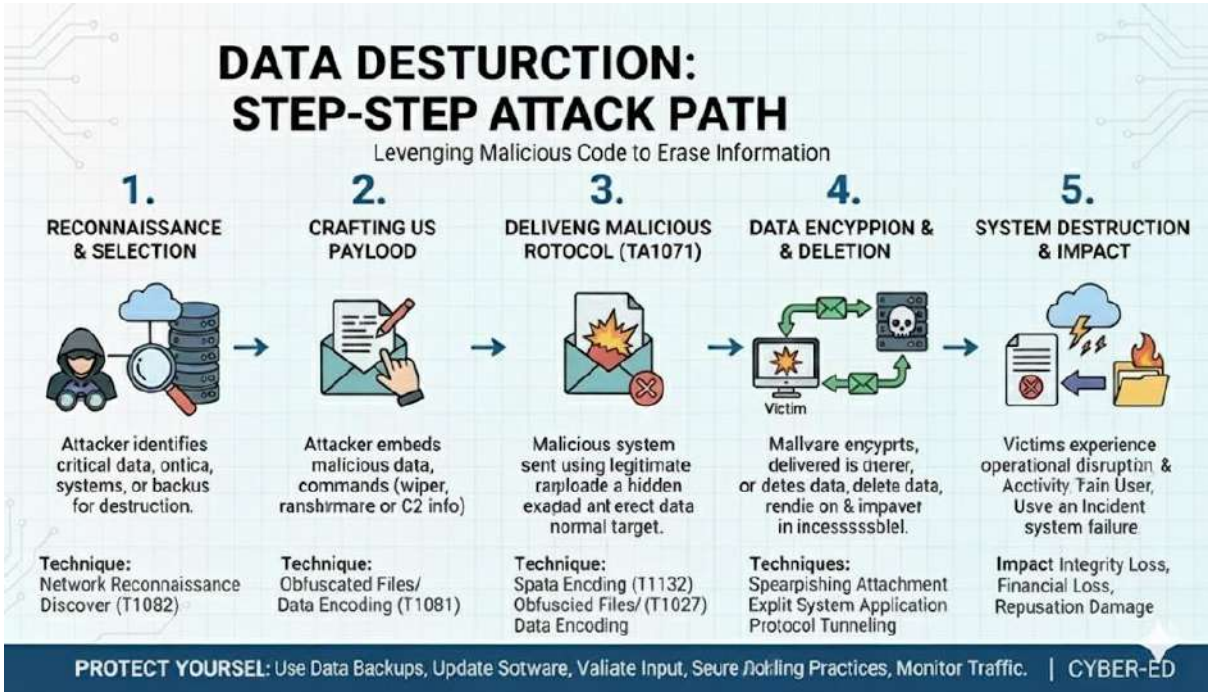
Overview



Technical details

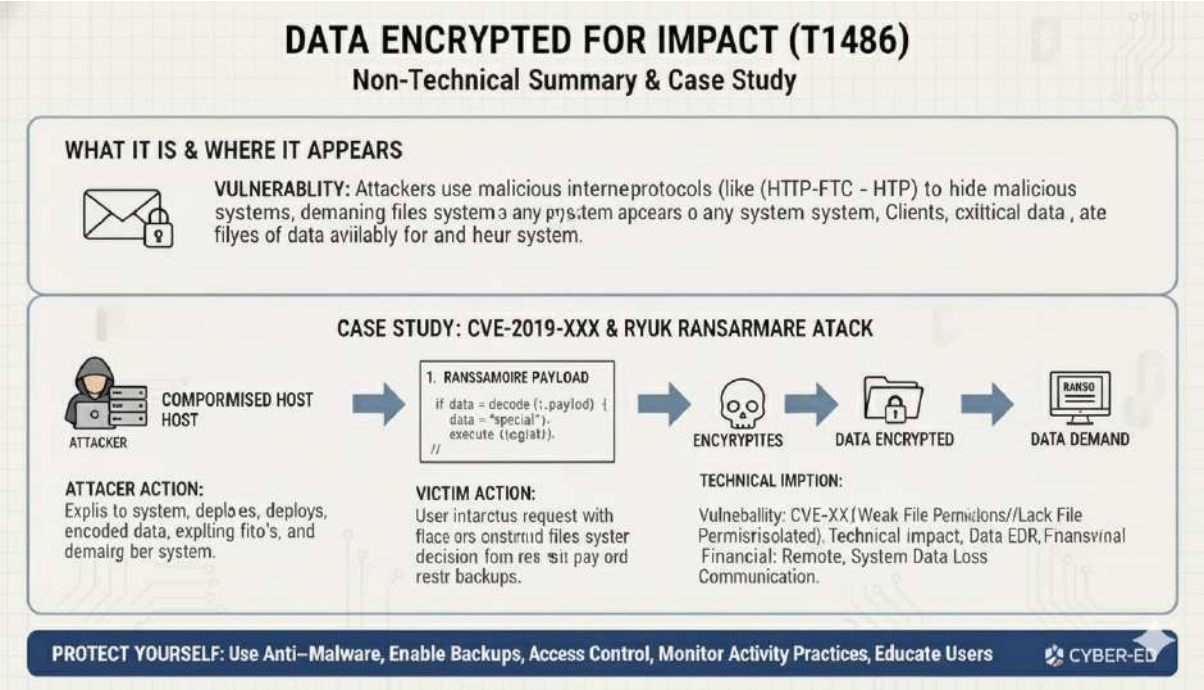


Attack flow / technique

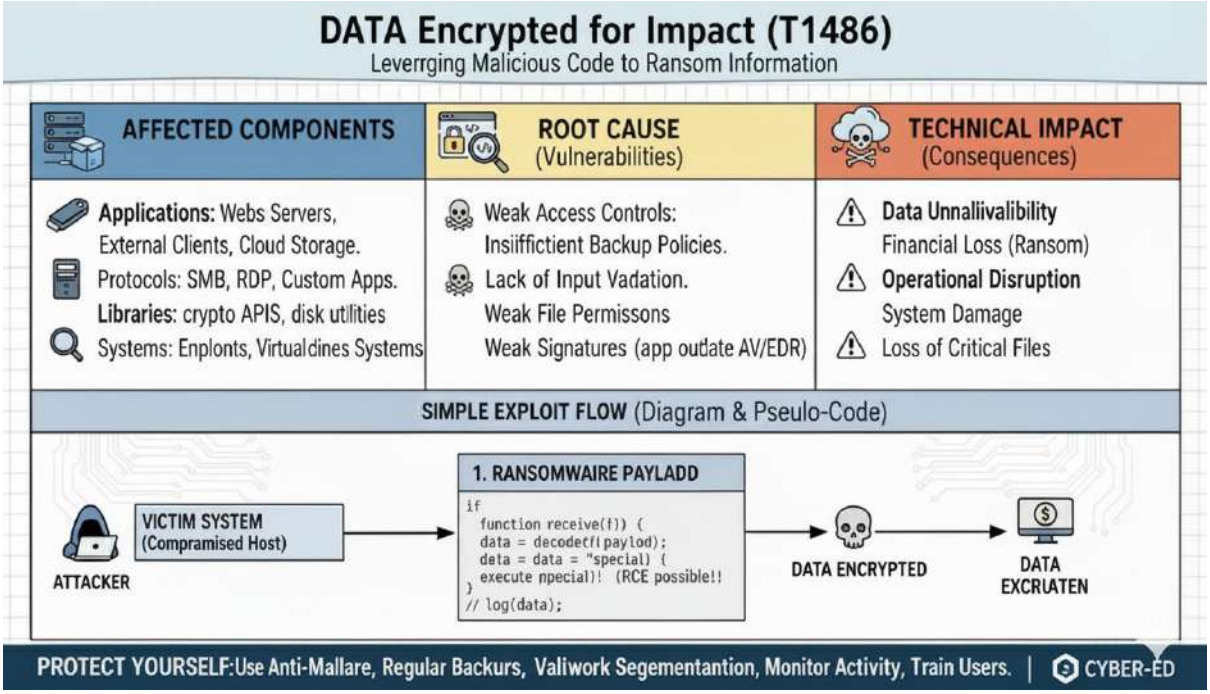


Data Encrypted for Impact (T1486) :

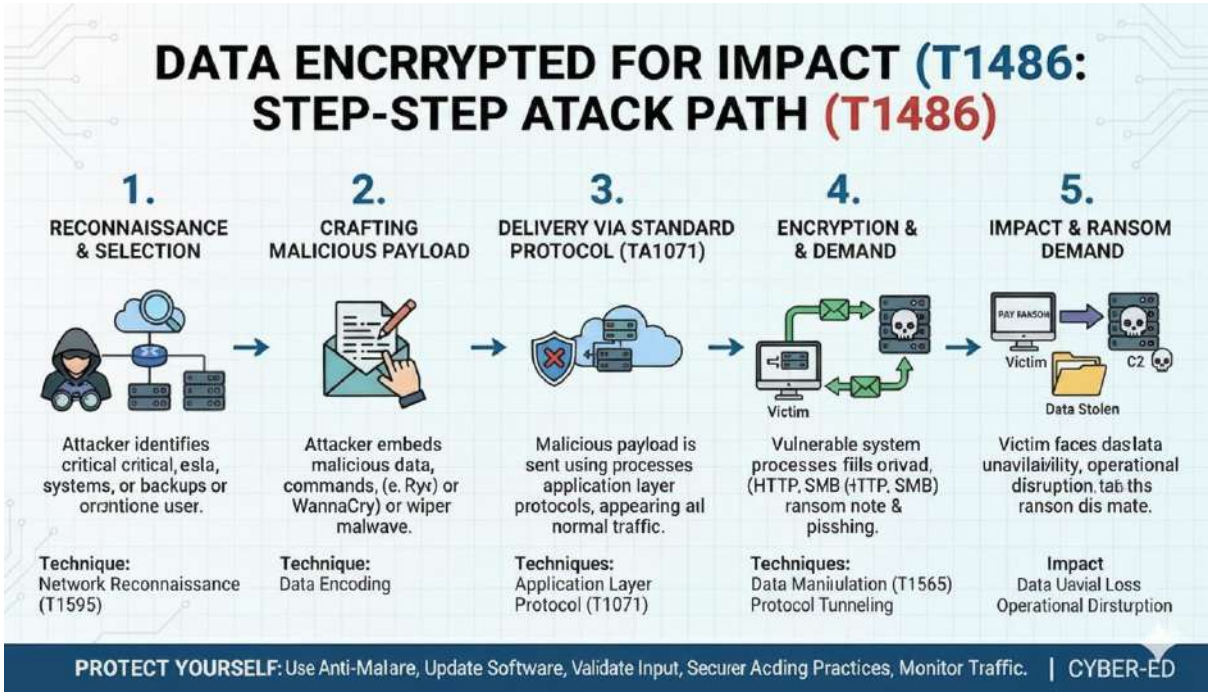
Overview



Technical details

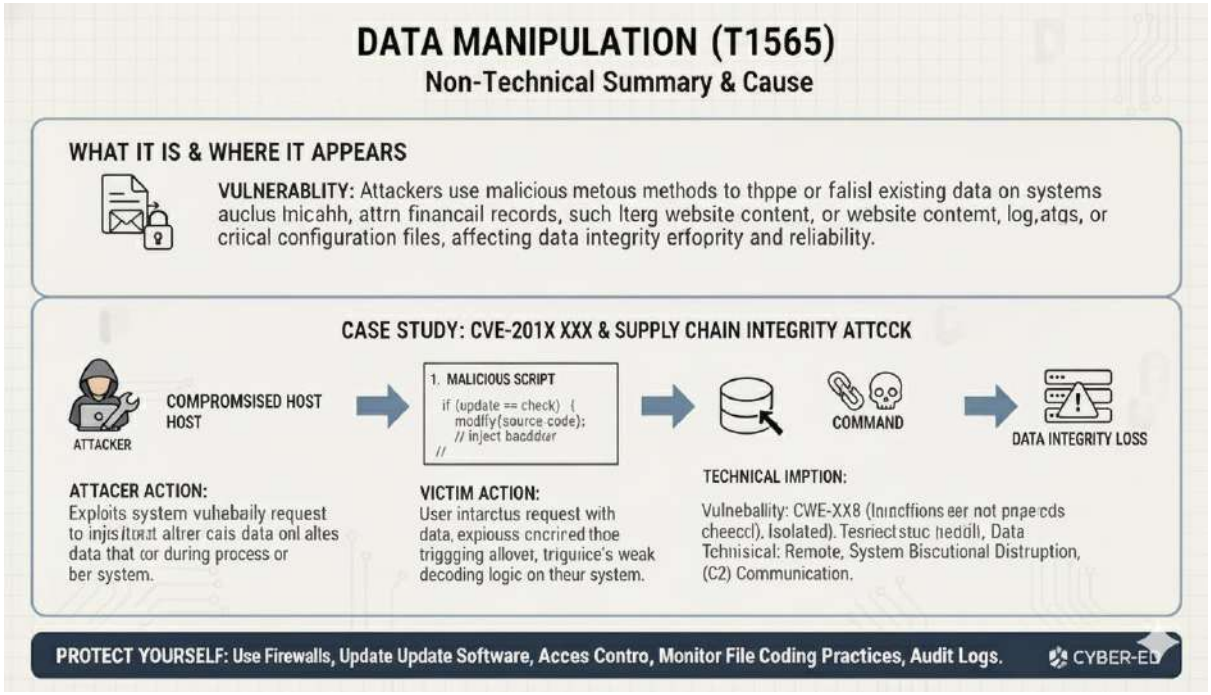


Attack flow / technique

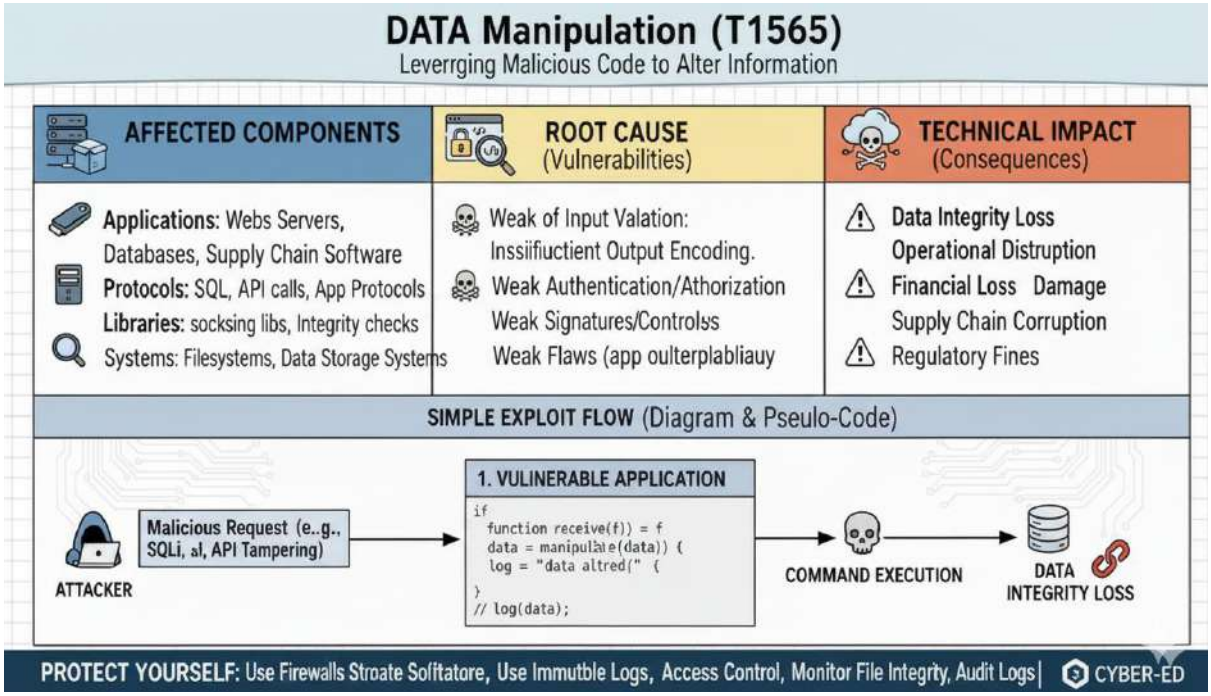


Data Manipulation (T1565) :

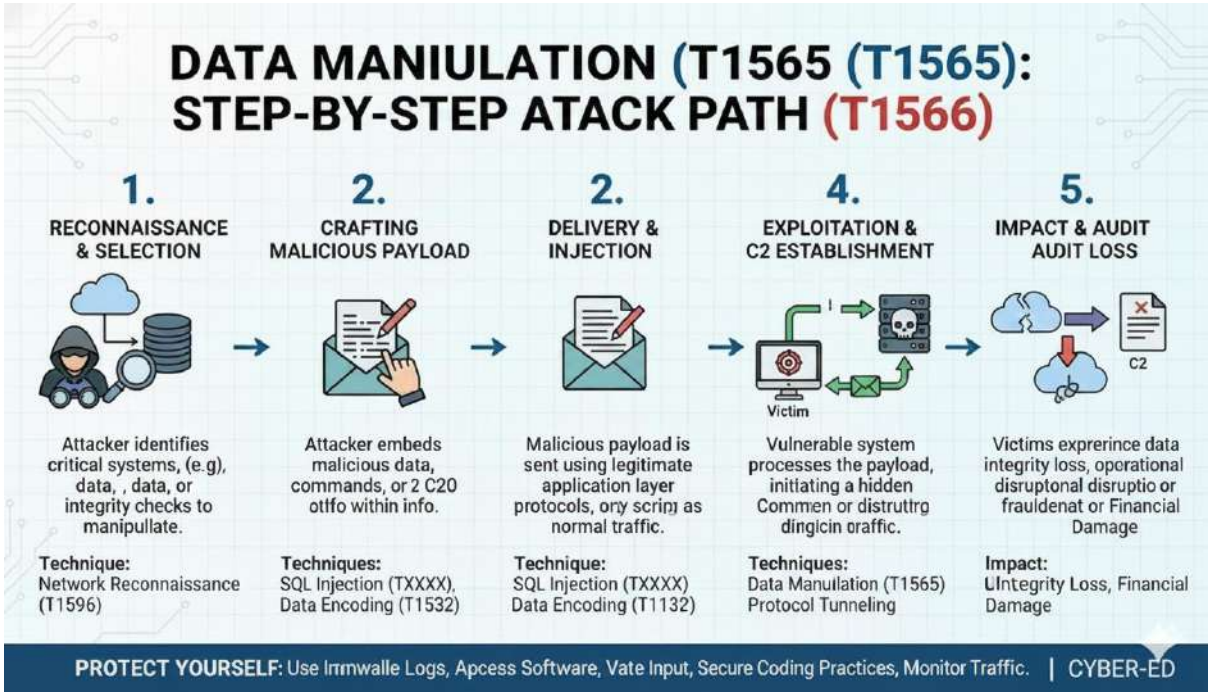
Overview



Technical details

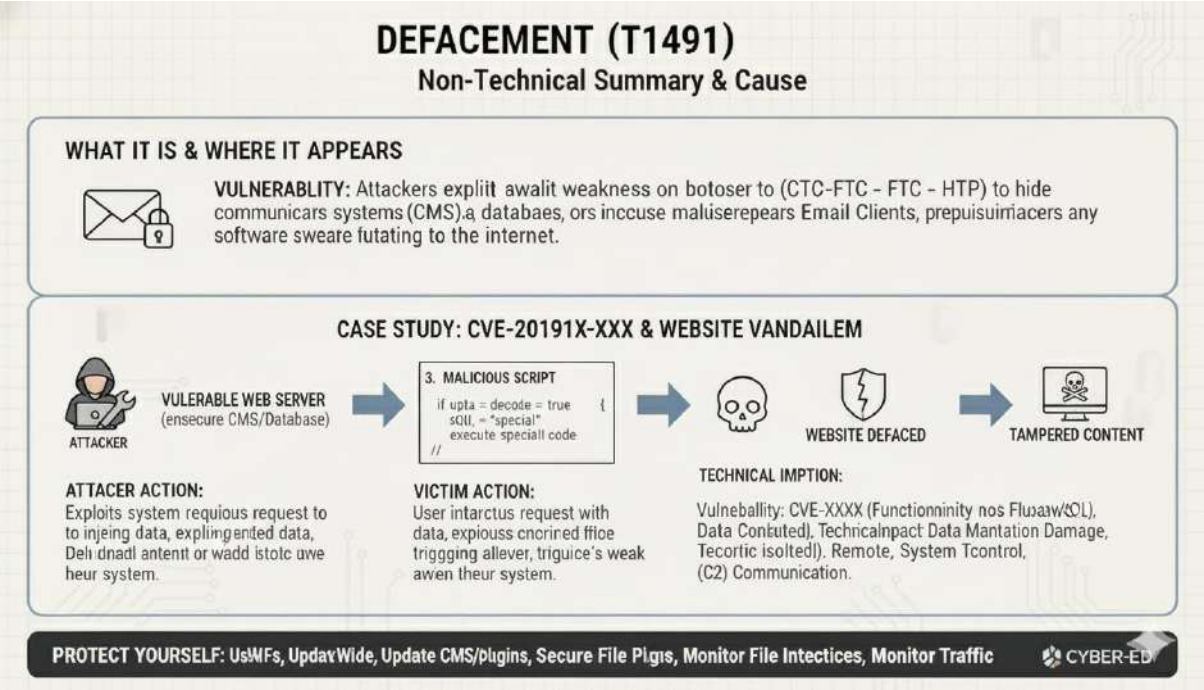


Attack flow / technique

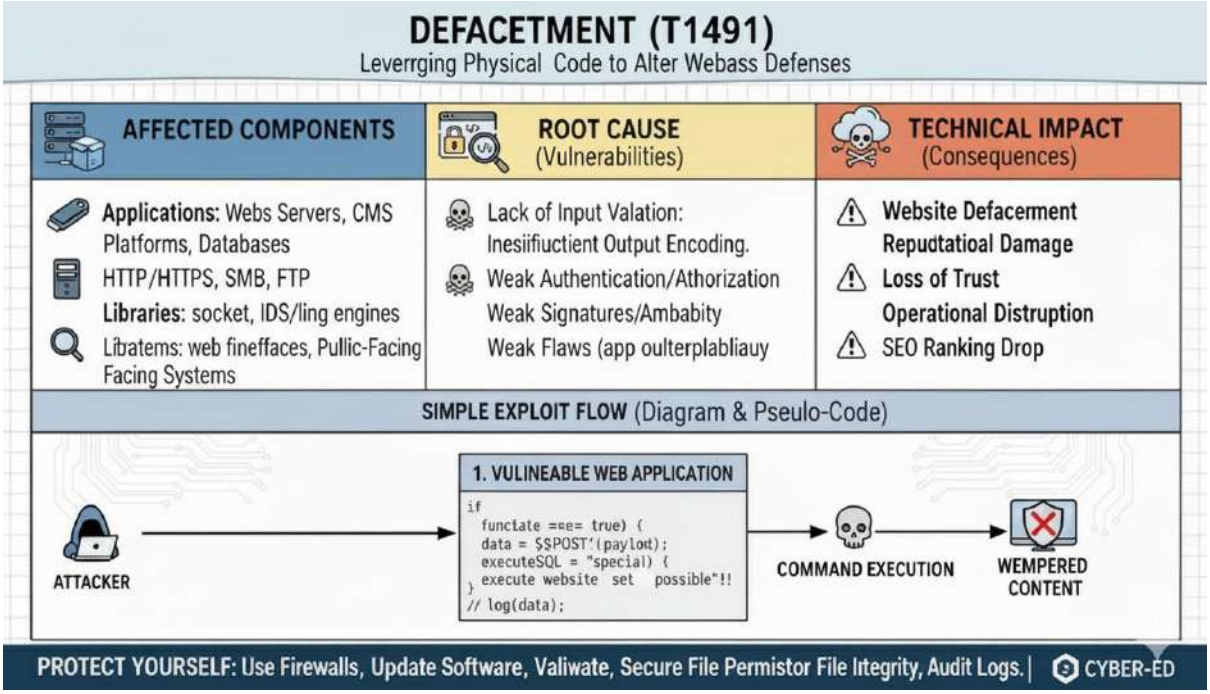


Defacement (T1491) :

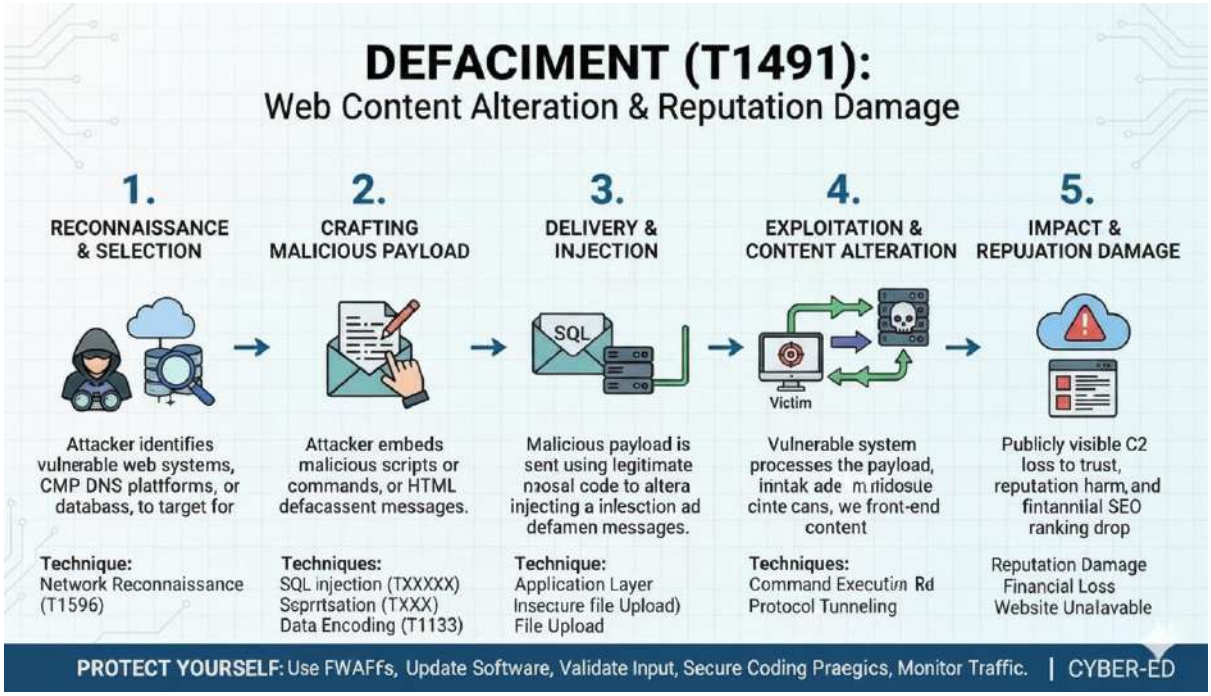
Overview



Technical details

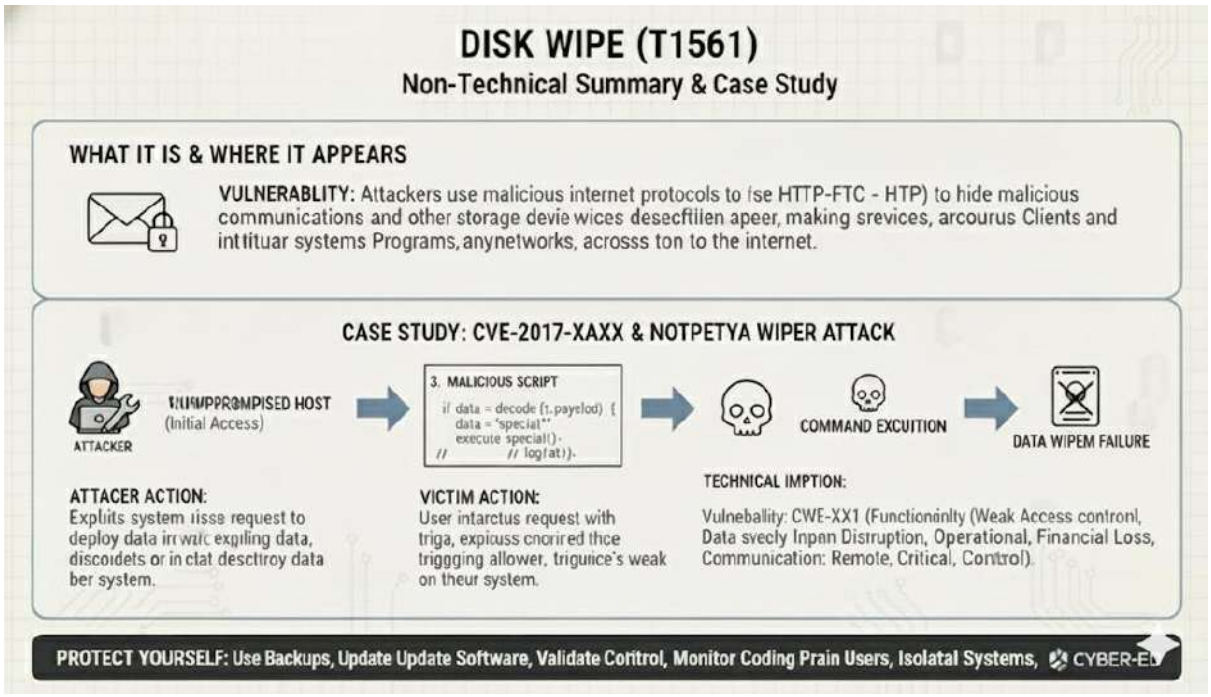


Attack flow / technique

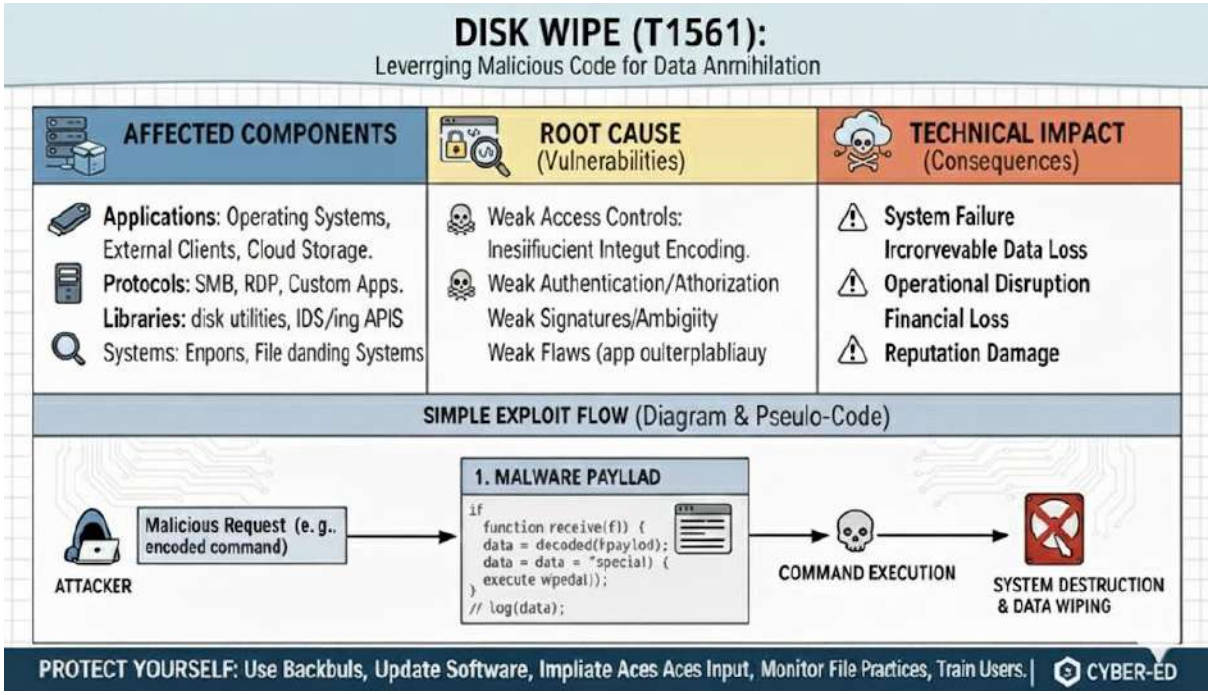


Disk Wipe (T1561) :

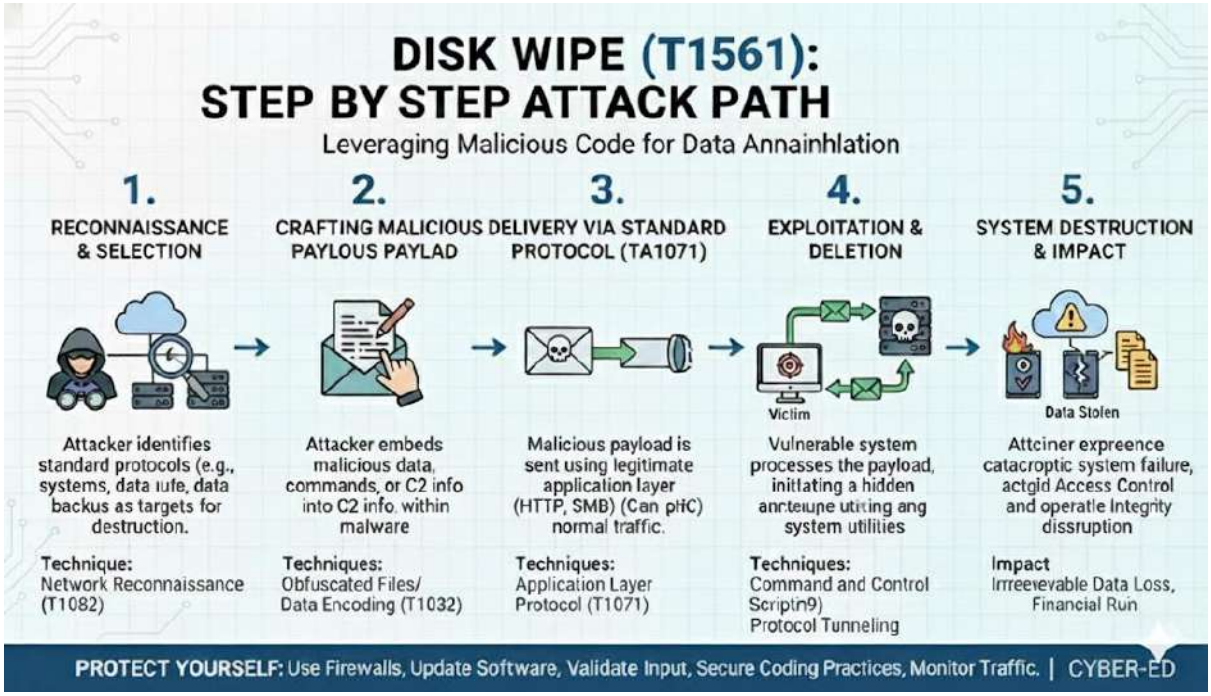
Overview



Technical details

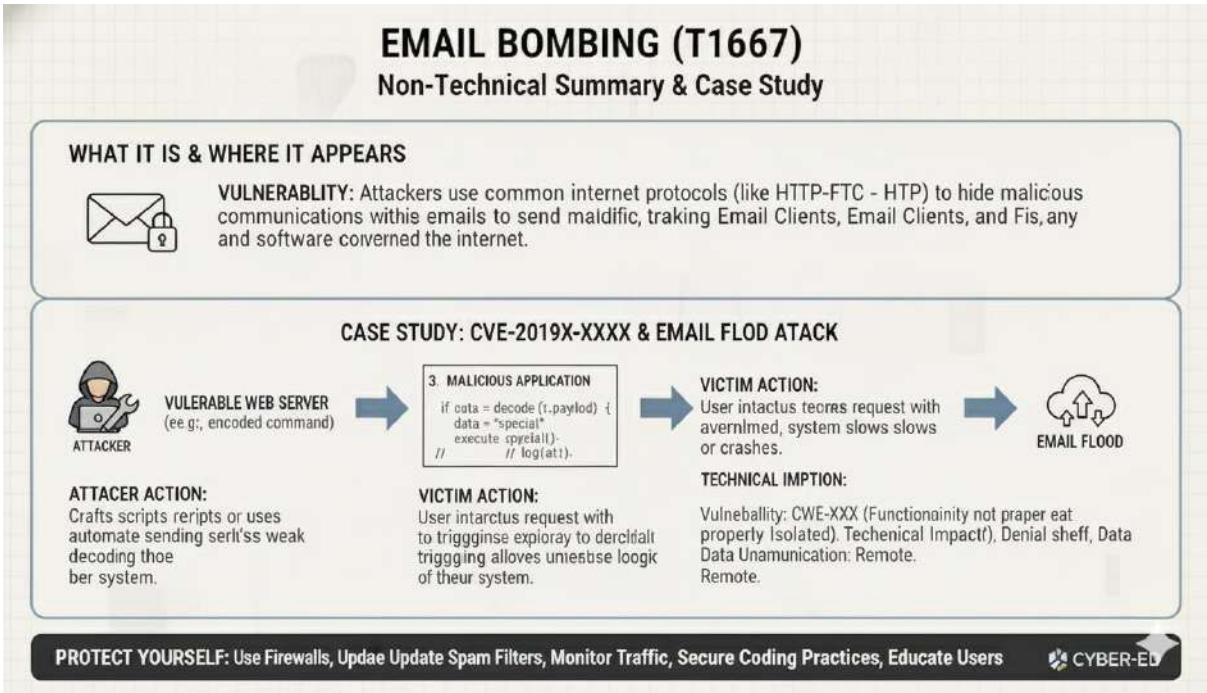


Attack flow / technique

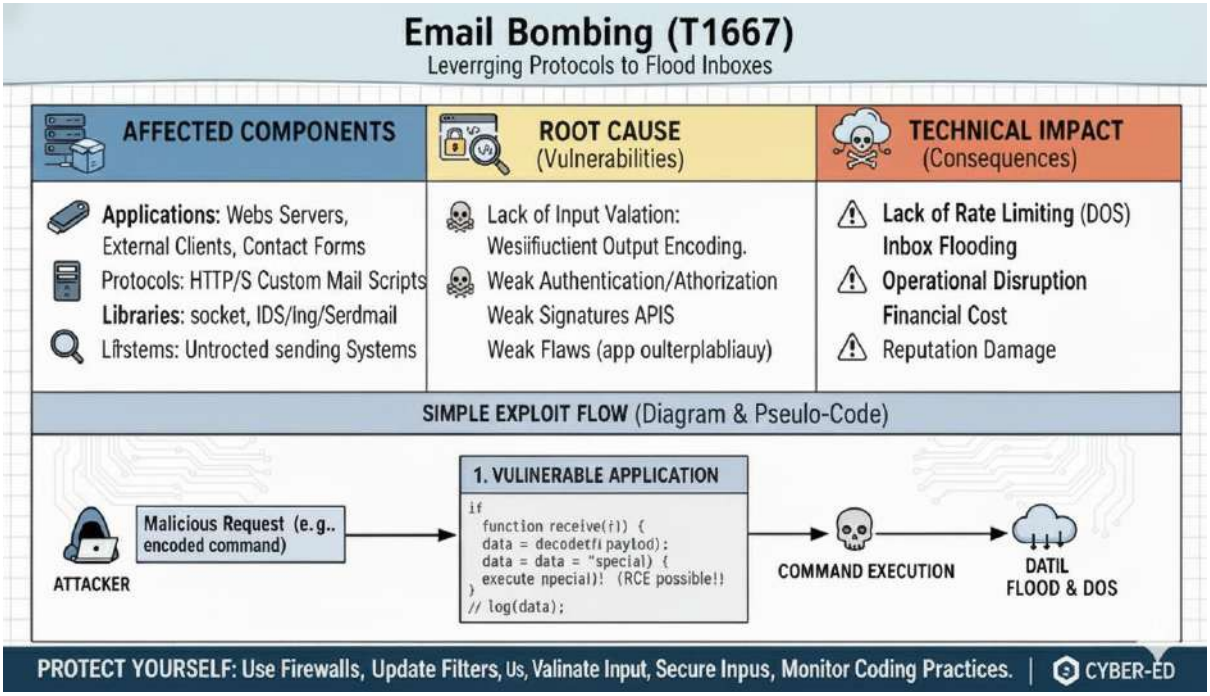


Email Bombing (T1667) :

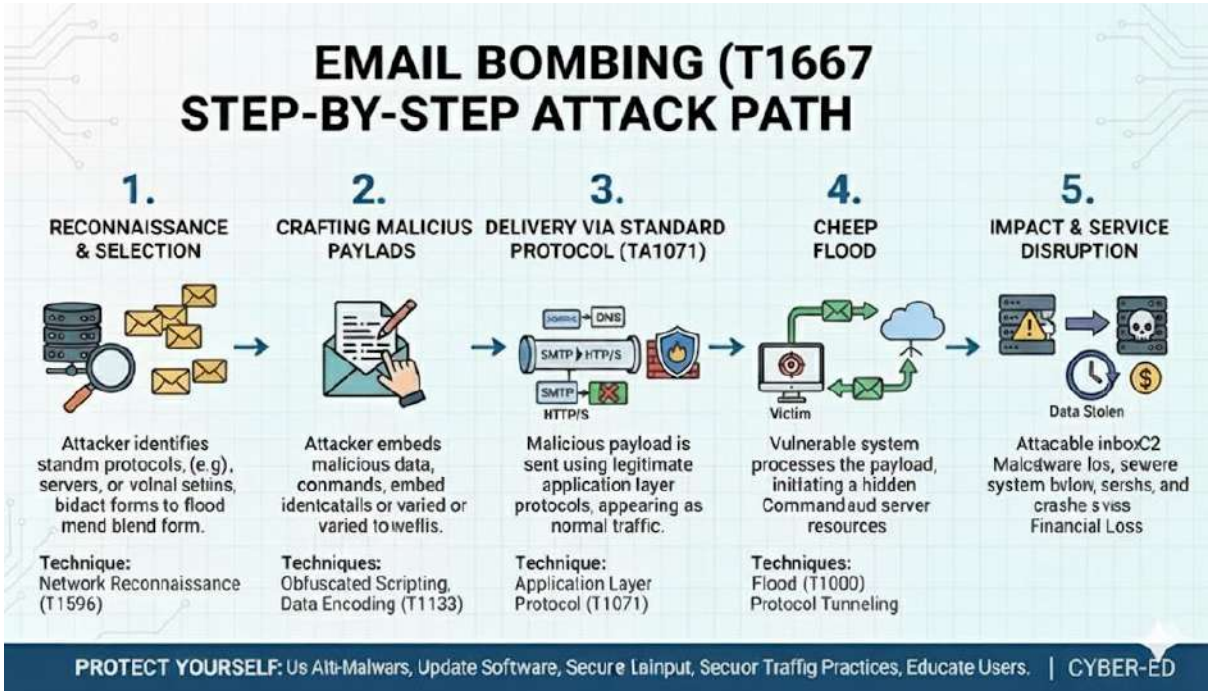
Overview



Technical details

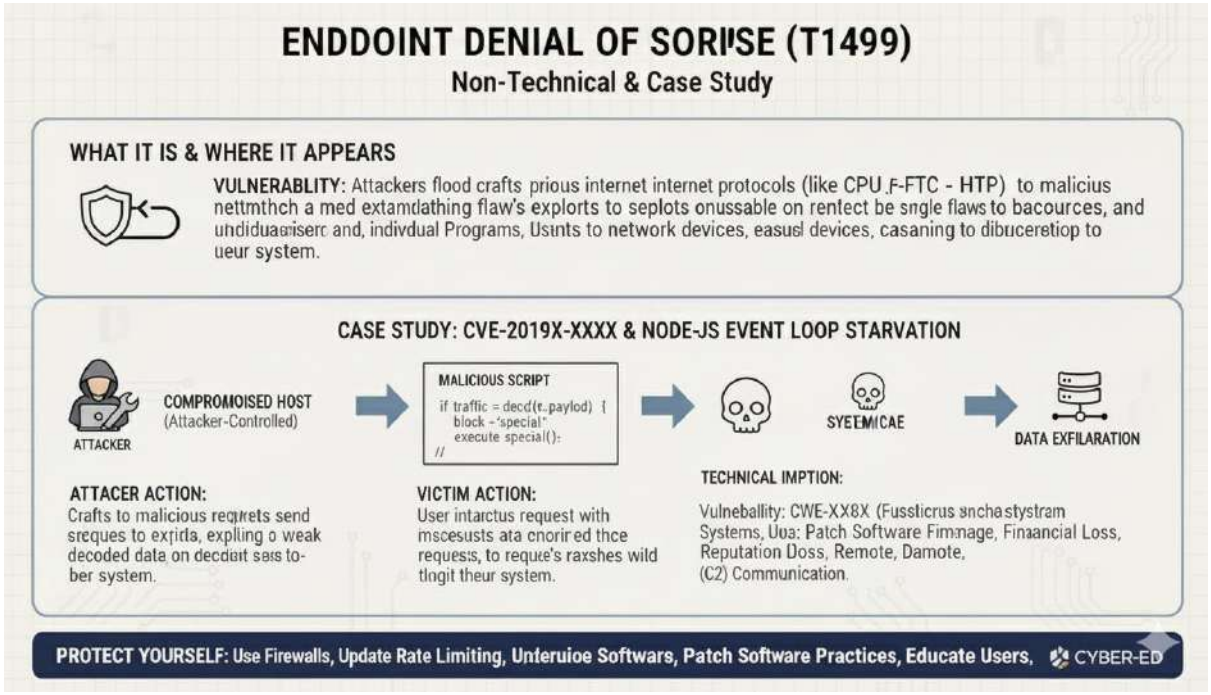


Attack flow / technique

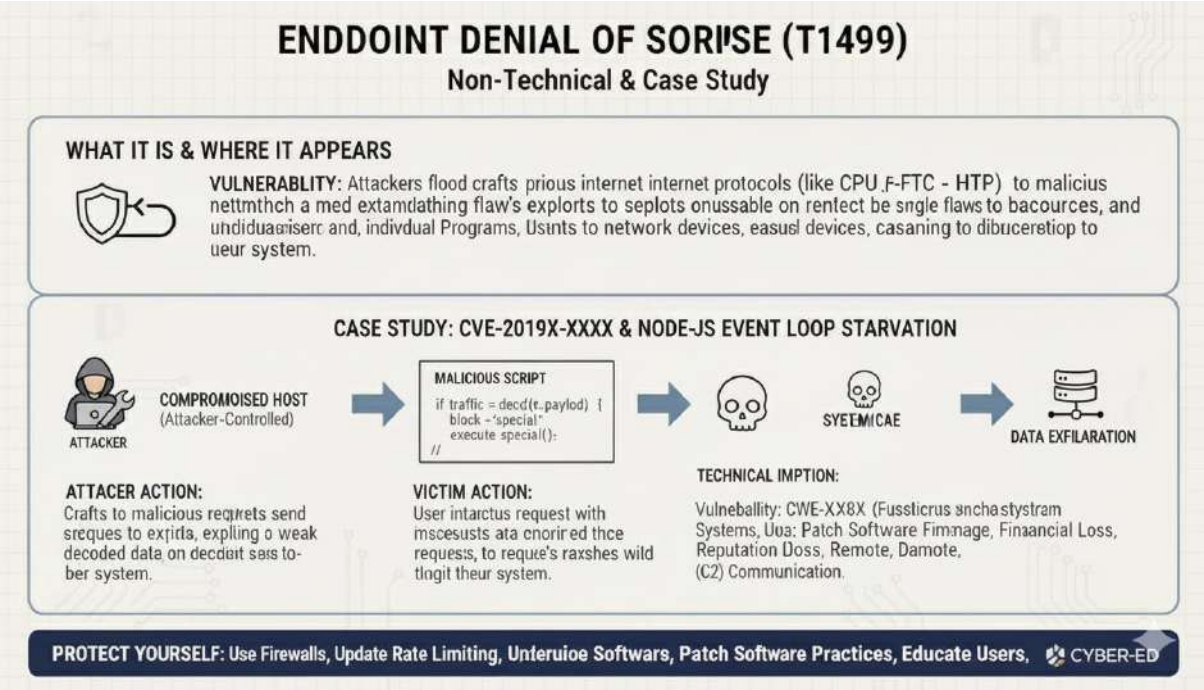


Endpoint Denial of Service (T1499) :

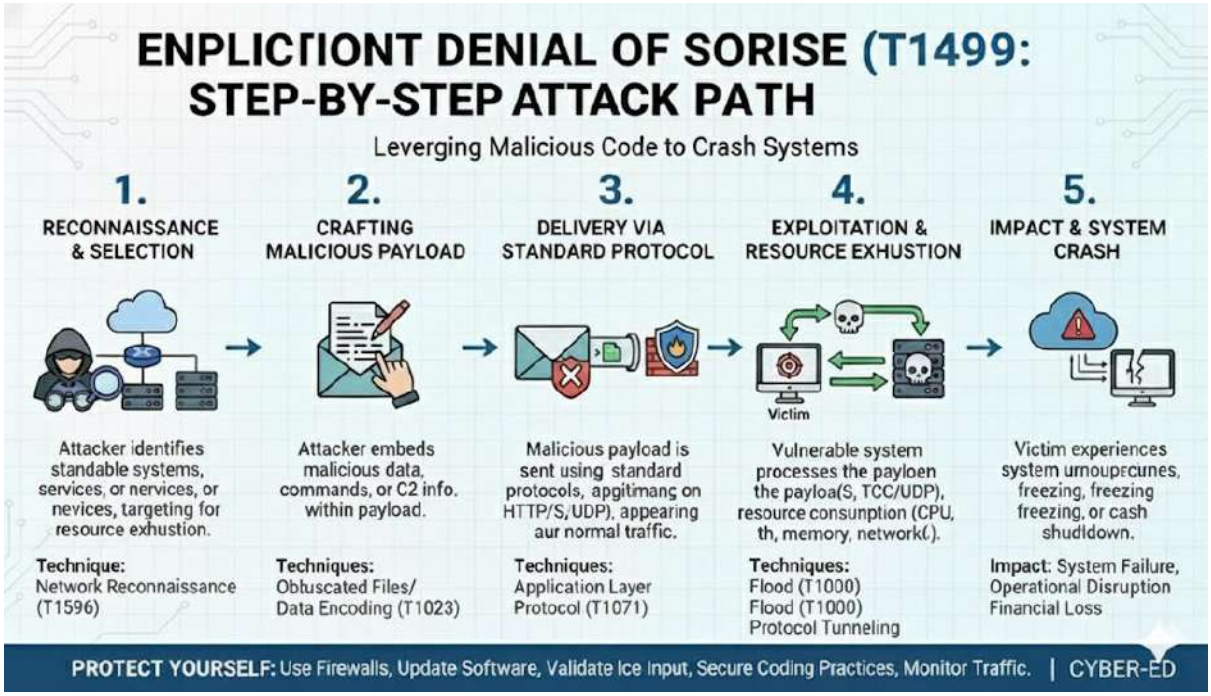
Overview



Technical details

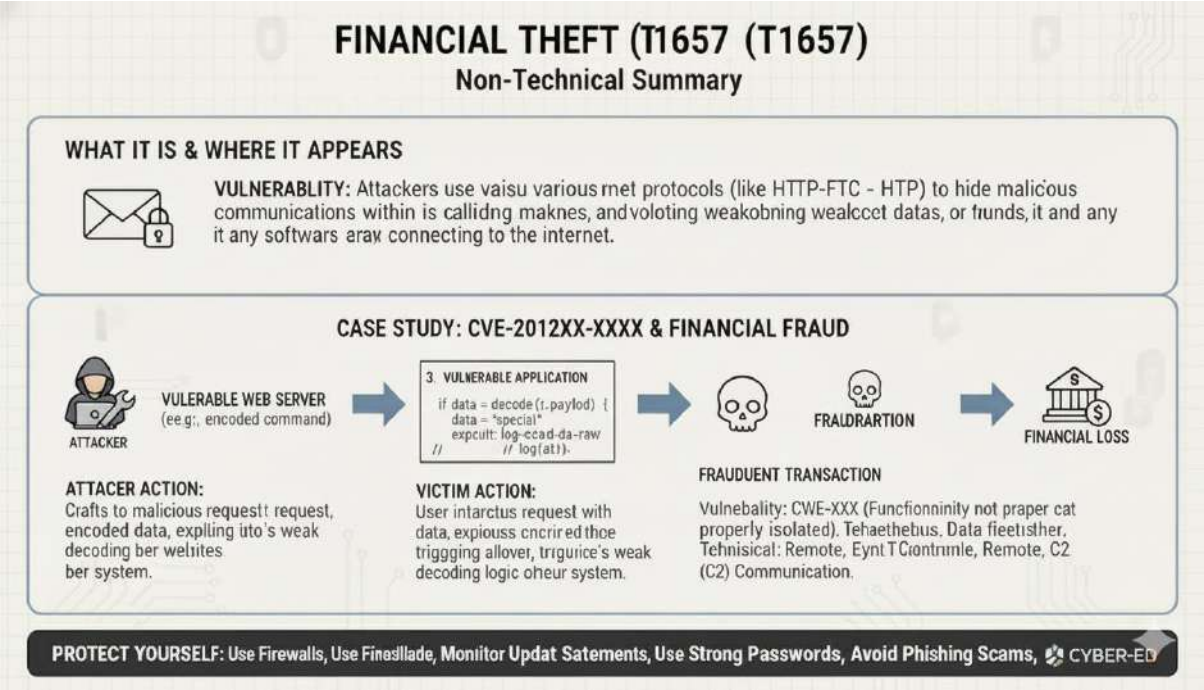


Attack flow / technique

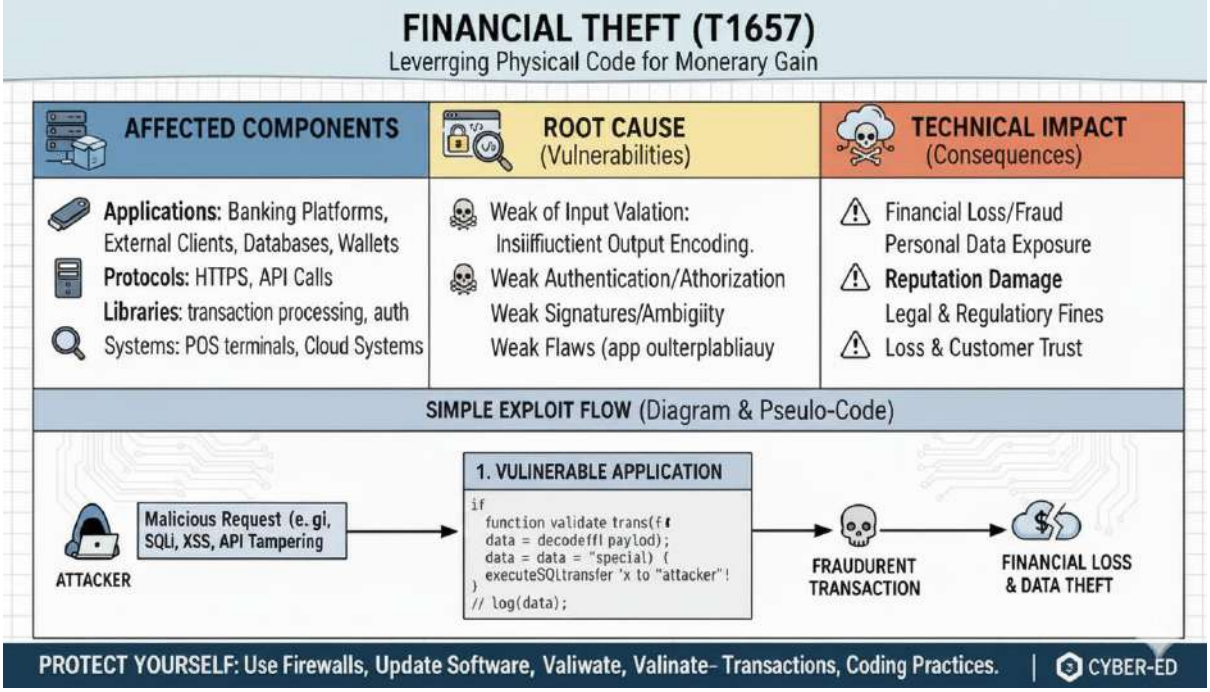


Financial Theft (T1657) :

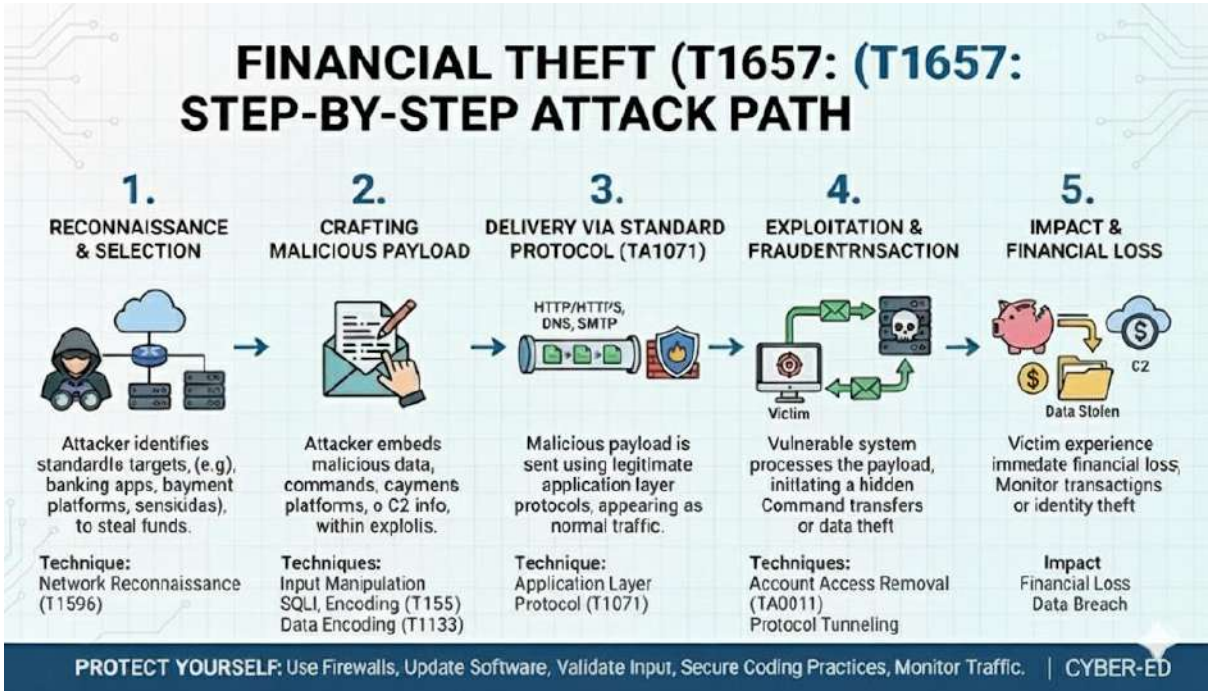
Overview



Technical details

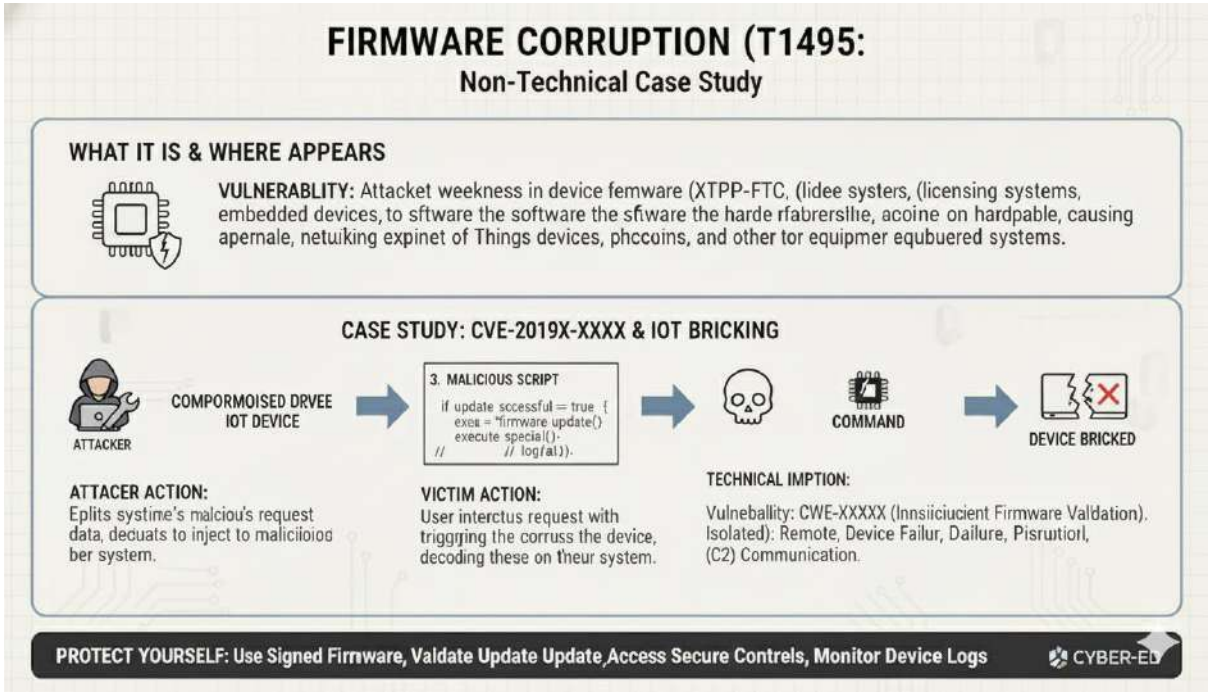


Attack flow / technique

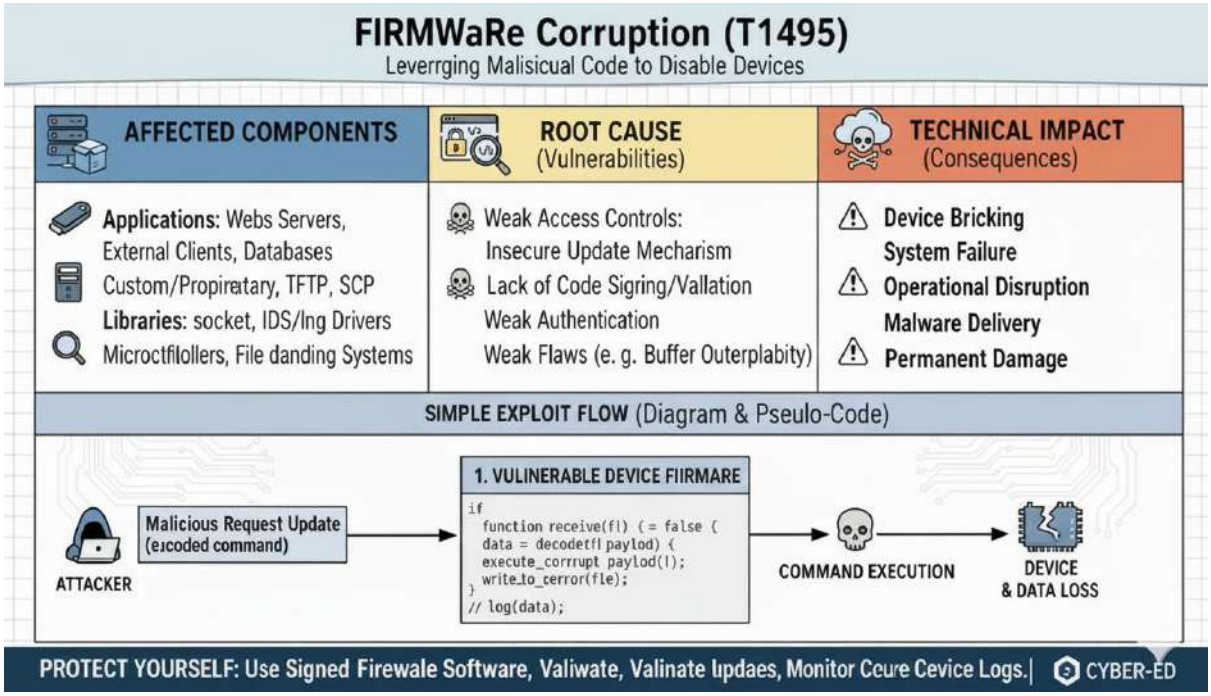


Firmware Corruption (T1495) :

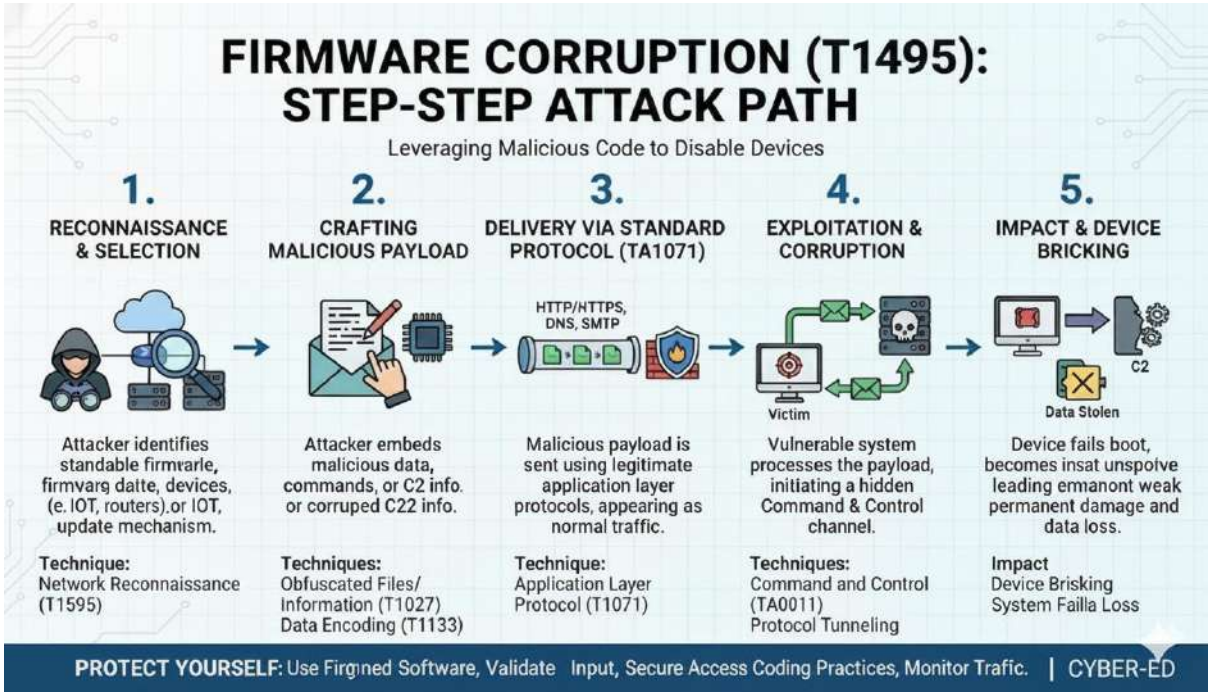
Overview



Technical details

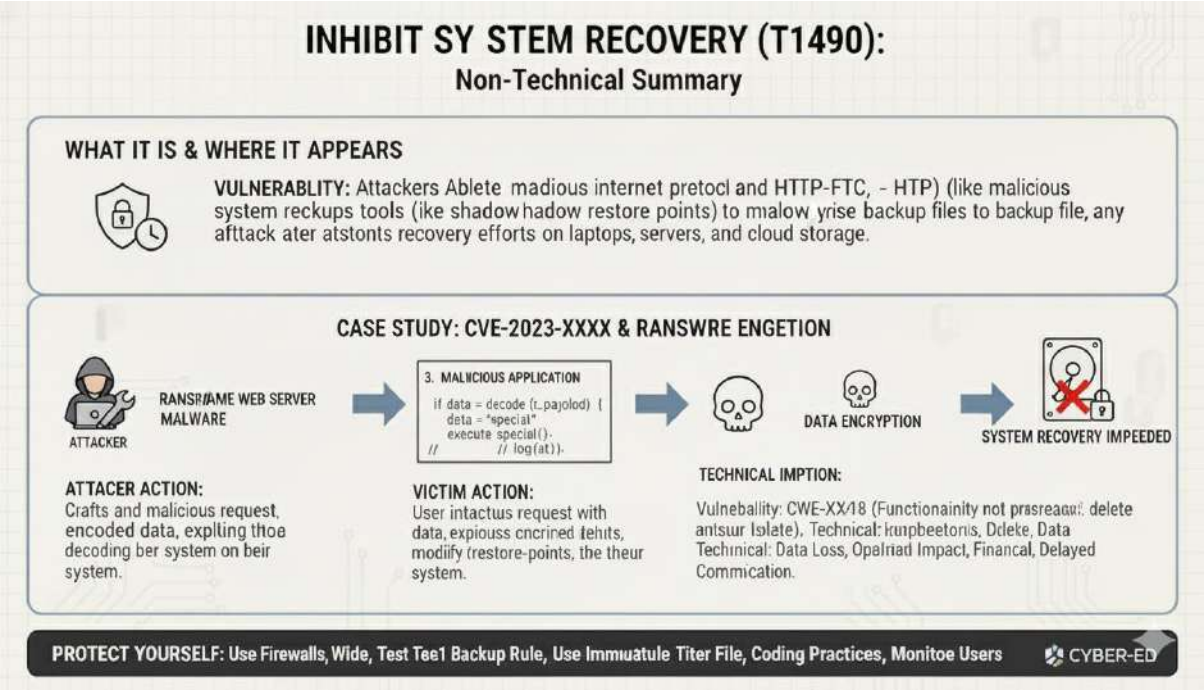


Attack flow / technique

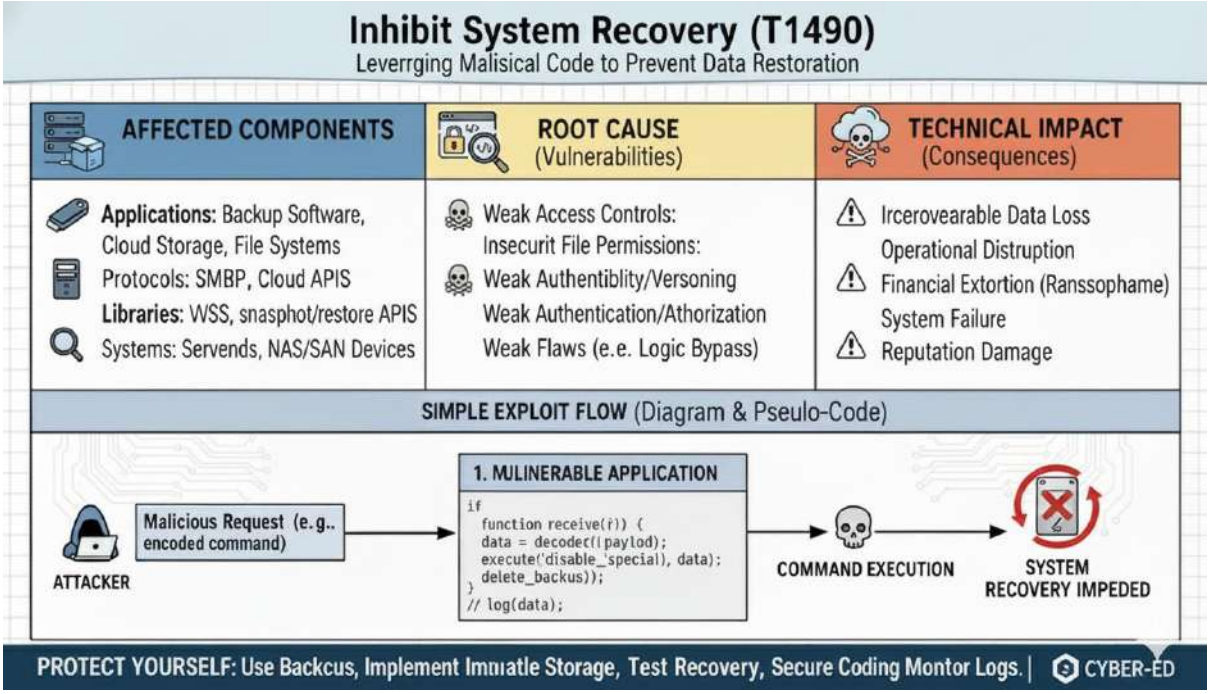


Inhibit System Recovery (T1490) :

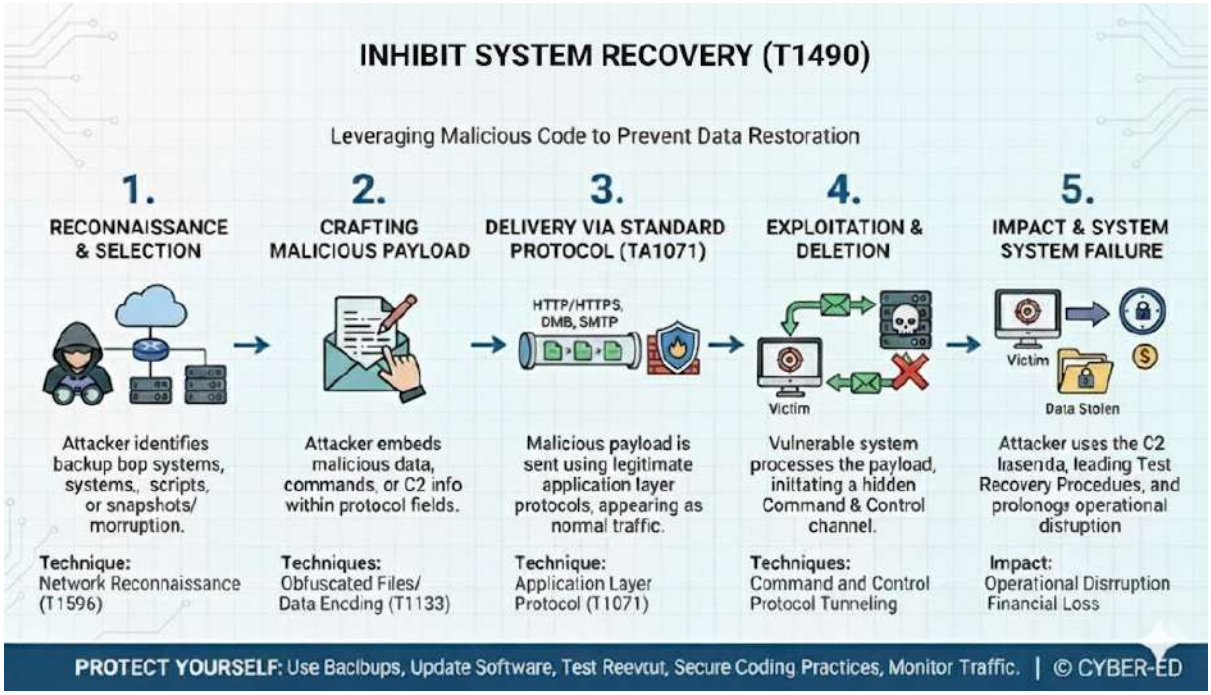
Overview



Technique details

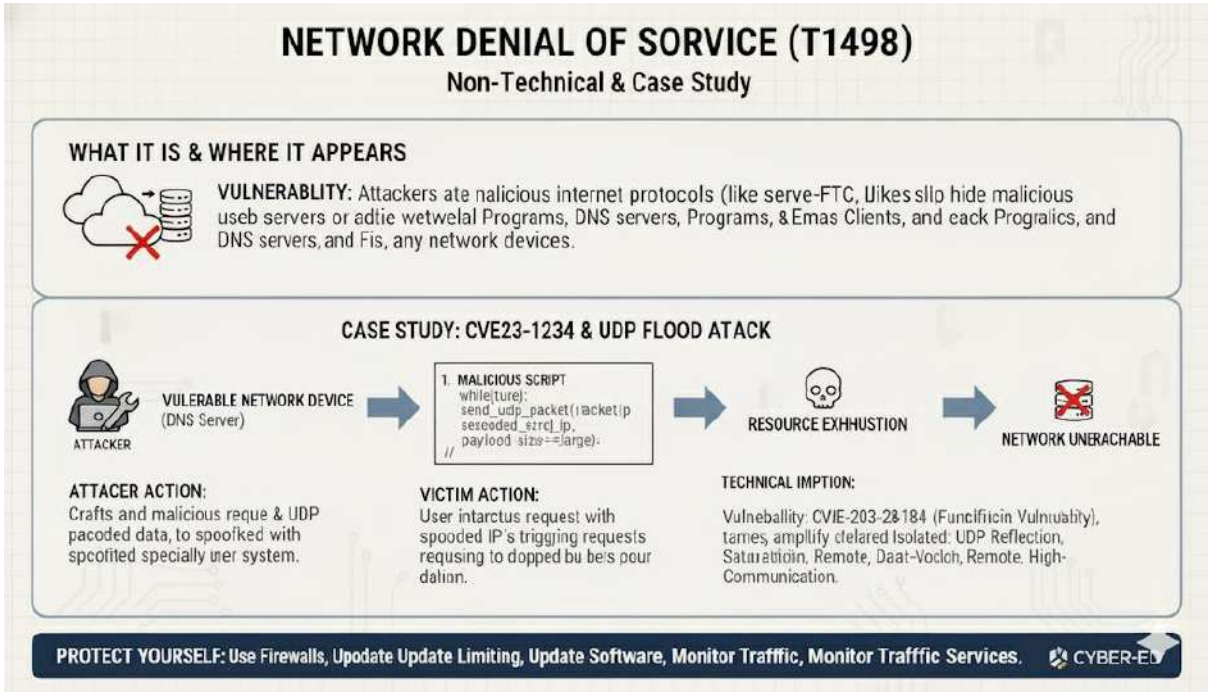


Attack flow / technique

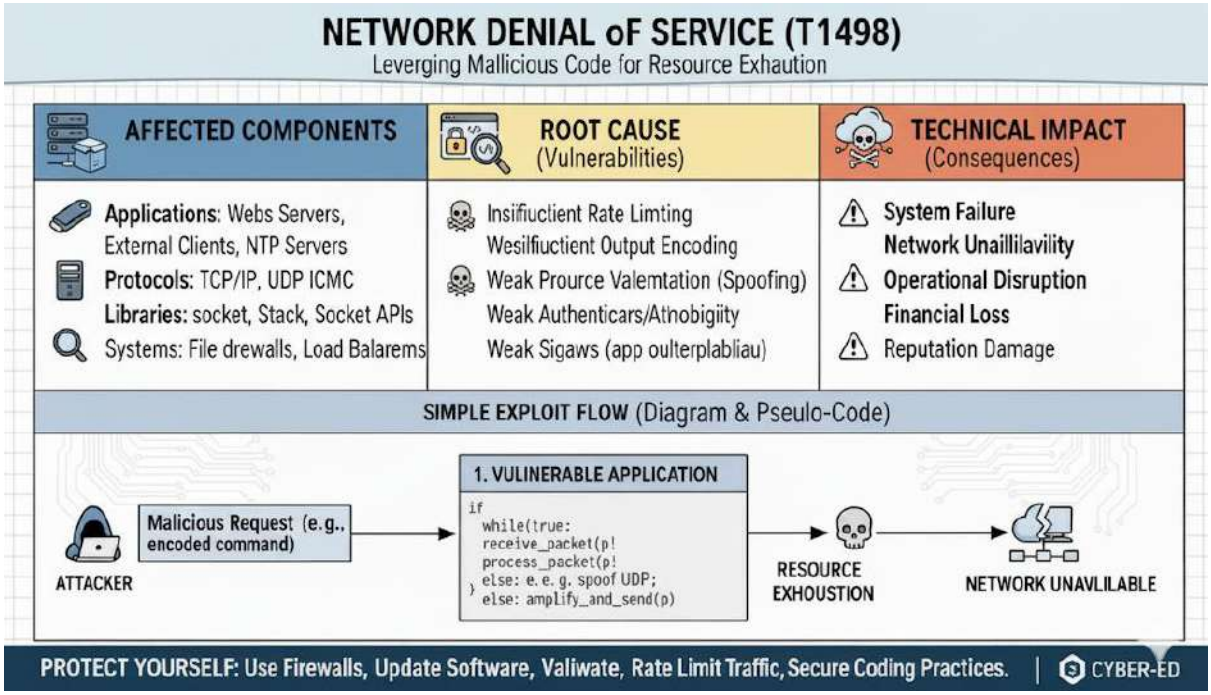


Network Denial Of Service (T1498) :

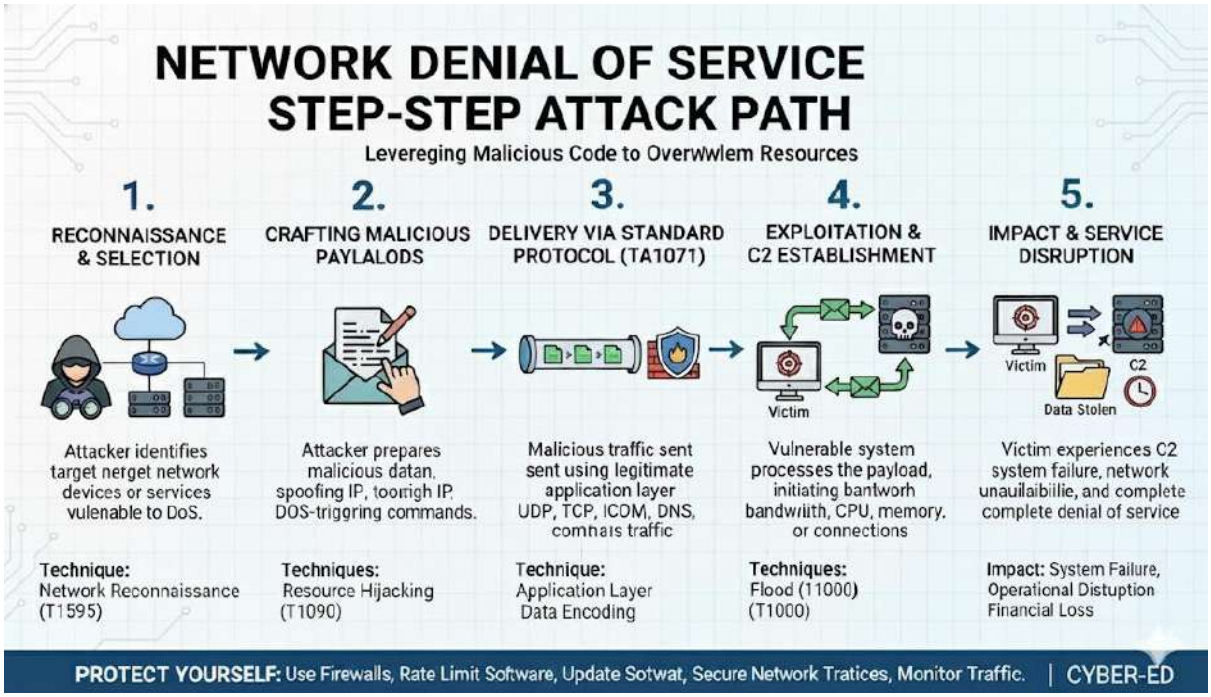
Overview



Technical details

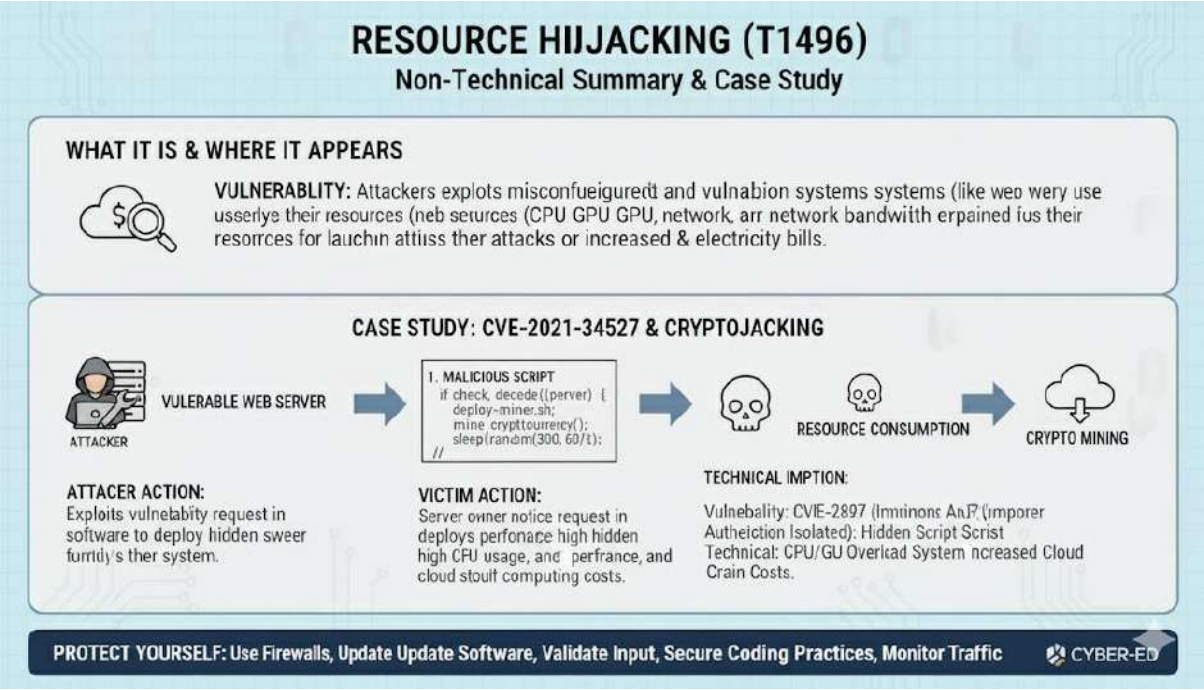


Attack flow / technique

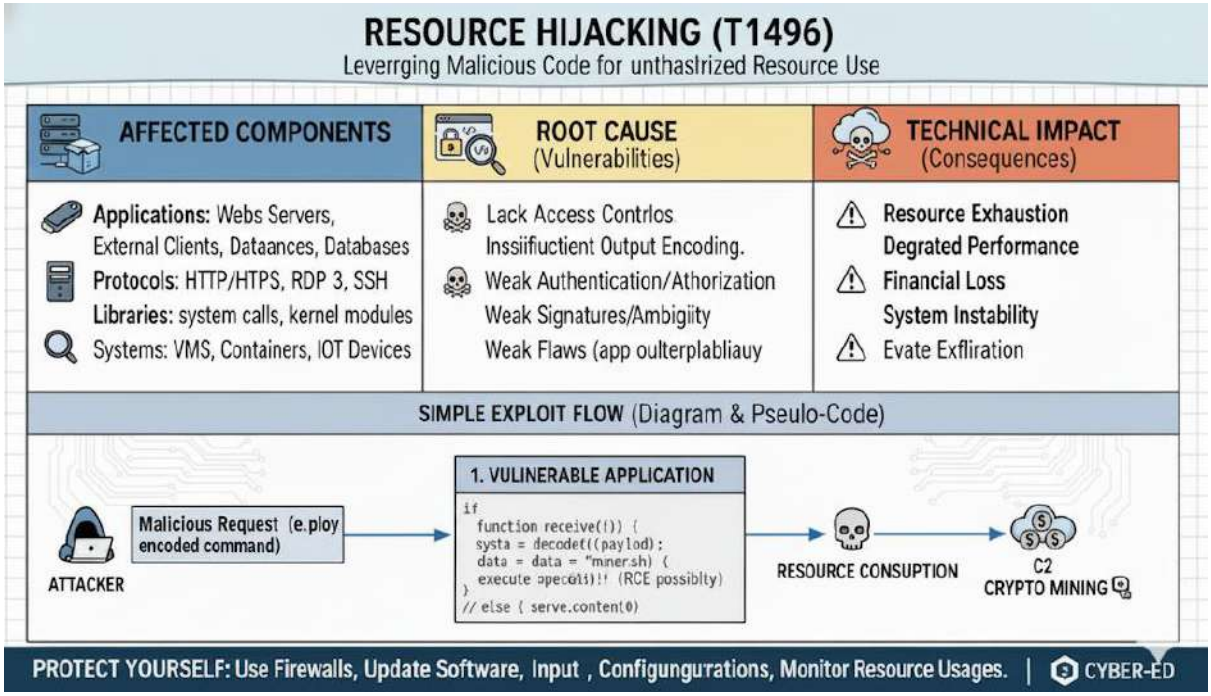


Resource Hijacking (T1496) :

Overview



Technical details



Attack flow/technique

RESOURCE HIJACKING (T1496:

Leverging Malicious Code for Unhahorized Resource Use

1. RECONNAISSANCE & SELECTION

Attacker identifies standard protems, (e.g., HTTP, DNS, lances, weak misinogrurationes controls sus inaget.

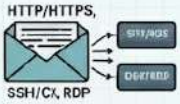
Technique:
Network Reconnaissance (T1595)

2. CRAFTING MALICIOUS PAYLOAD

Attacker embeds malicious data, commands, or C2 info scripet / botnet info.

Techniques:
Obfuscated Files/Data
Data Encoding

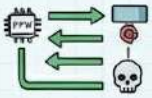
3. DELIVERY VIA STANDARD PROTOCOL (TA1071)



Malicious payload is sent using legitimate application layer protocols, appearing as normal traffic.

Technique:
Application Layer Protocol (T1071)

4. EXPLOITATION & CONSUMPTION



Vulnerable system processes the payload, initiating a hidden (CPU, GPU, & network bandwich.

Techniques:
Command and Control Protocol (T1071)

5. IMPACT &NANCIA LOSS

Victtim expreiences degrated performance, cloud sergy bils, eceasee and natnail loss resource detections

Impact
Financial Loss, System System Degraation

PROTECT YOURSELF: Use Firewalls, Update Software, Validate Input, Secure Coding Practices, Monitor Traffic. | CYBER-ED