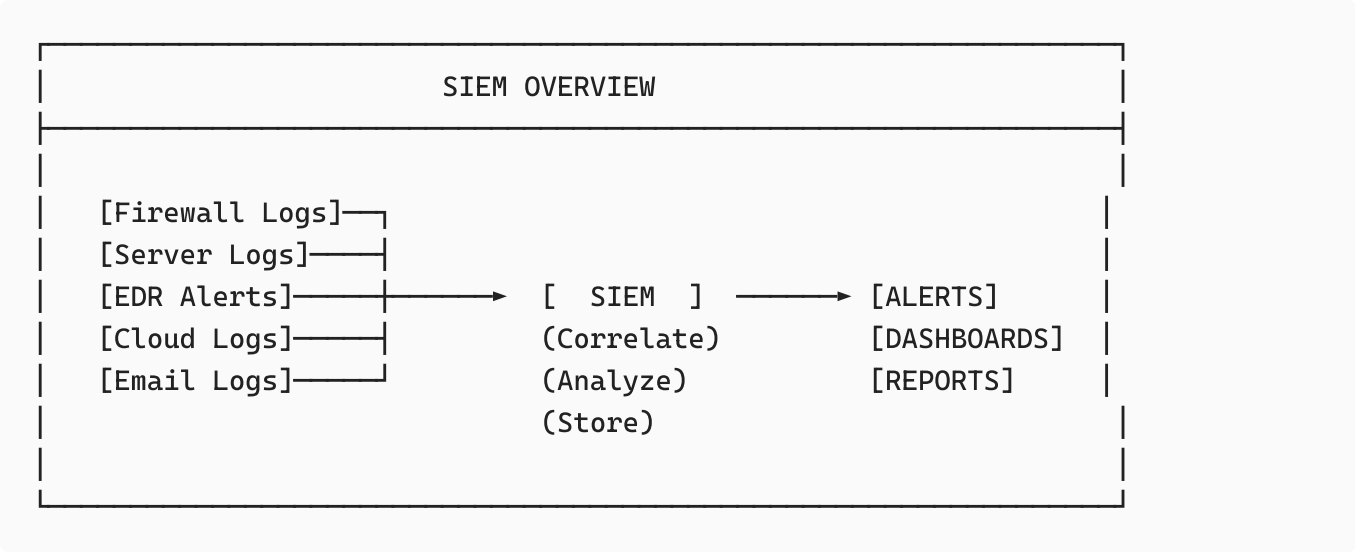
01,SIEM,Architecture, Log Sources, Alert Triage & Investigation Process, False Positives & How to Reduce Them, SOAR & Playbooks

SIEM = Security Information and Event Management

It collects logs from everywhere in your organization, correlates them, and alerts you when something suspicious happens.



The Two Parts of SIEM:

Component	What it does	Example
SIM (Security Information Management)	Long-term storage, compliance, and historical reporting.	"Generate a report of all admin logins from the last 6 months."
SEM (Security Event Management)	Real-time monitoring, event correlation, and instant alerting.	"Trigger an alert **NOW** if 10 failed logins happen within 1 minute."
SIEM (Combined)	The complete solution for both real-time detection and historical analysis.	"Alert me to a brute force attack (SEM) and store the logs for 1 year (SIM)."

Need of SIEM

Without SIEM:

```
Analyst: "I need to check if this IP attacked us"
         → Log into firewall... search...
         → Log into AD... search...
         → Log into web server... search...
         → 3 hours later... still searching
```

With SIENM:

```
Analyst: Search "src_ip=192.168.1.100"
         → All logs from all sources in ONE place
         → 30 seconds... done!
```

SIEM Capabilities

```
┌──────────────────────────────────────────────────────────┐
│                  SIEM CAPABILITIES                       │
├──────────────────┬───────────────────────────────────────┤
│ Log Collection   │ Gather logs from 100s of sources      │
├──────────────────┼───────────────────────────────────────┤
│ Normalization    │ Convert all logs to common format     │
├──────────────────┼───────────────────────────────────────┤
│ Correlation      │ Connect related events together       │
├──────────────────┼───────────────────────────────────────┤
│ Alerting         │ Trigger alerts on suspicious          │
│                  │ patterns                              │
├──────────────────┼───────────────────────────────────────┤
│ Dashboards       │ Visualize security posture            │
├──────────────────┼───────────────────────────────────────┤
│ Retention        │ Store logs for compliance             │
│                  │ (90 days, 1 year, etc.)               │
└──────────────────┴───────────────────────────────────────┘
```

Popular SIEM Tools

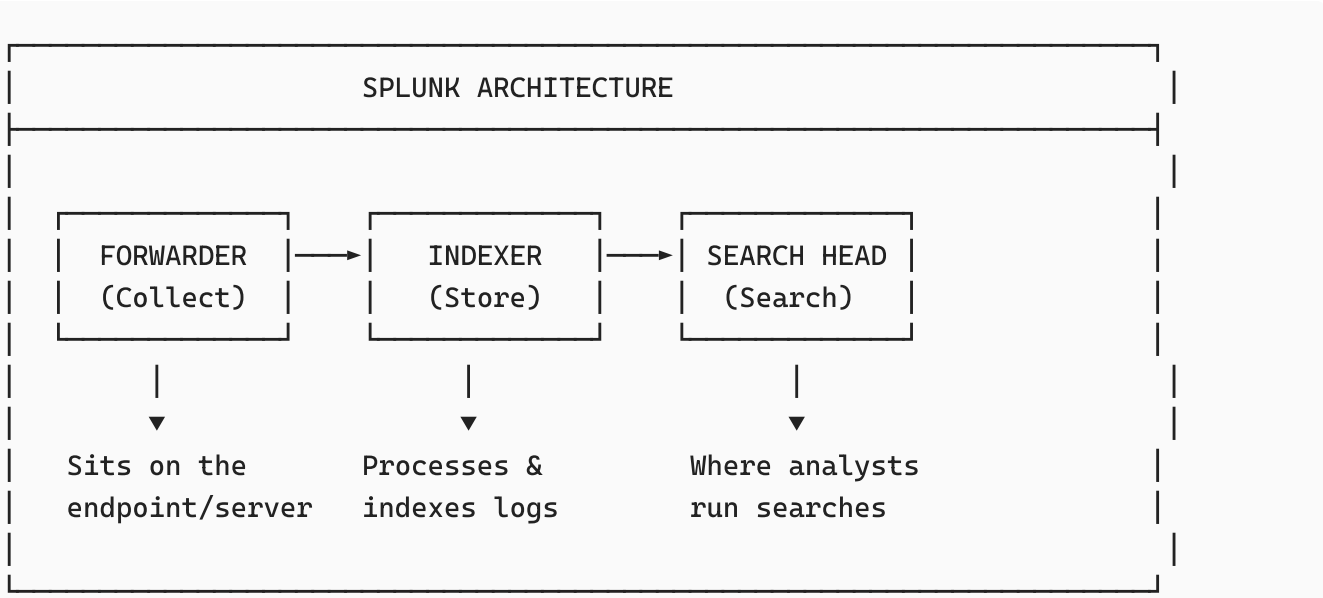
SIEM Tool	Type	Common In	2026 Market Note
Splunk	Hybrid (On-Prem/Cloud)	Enterprise, Fortune 500	Still the #1 "Power User" tool. Great for complex queries.
Microsoft Sentinel	Cloud-Native (SaaS)	Azure / Office 365 Shops	Rapidly growing due to easy integration with Windows/Azure.
IBM QRadar	Hybrid (On-Prem/Cloud)	Large Banks & Govt	Known for strong correlation and "Offense" management.

SIEM Tool	Type	Common In	2026 Market Note
Elastic SIEM (ELK)	Open Source / Managed	Startups, Tech-focused	Fast and highly customizable; popular for "Big Data" logging.
Google Chronicle	Cloud-Native	Google Cloud (GCP)	Uses "Planet-scale" search; very fast at searching years of data.
CrowdStrike LogScale	Cloud-Native	Modern, High-Speed SOCs	Formerly Humio; built for massive ingest speeds and "Live" data.

SIEM Architecture

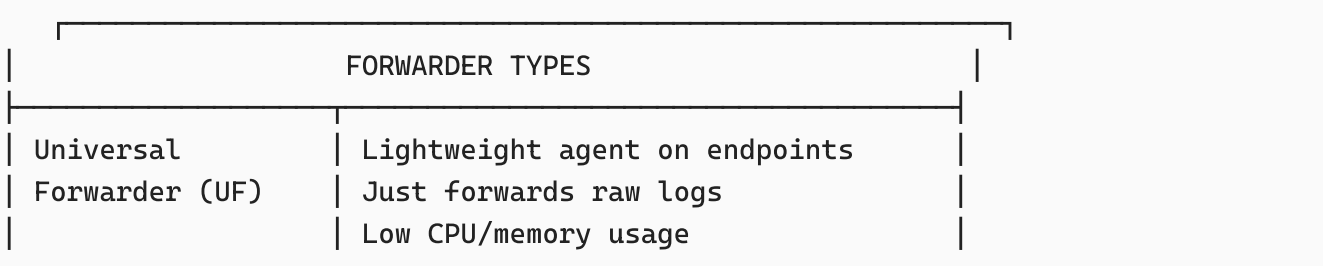
Splunk

Three-Tier Architecture:



Component

1. Forwarders (Data Collection)



```
├──────────────────┼──────────────────────────────────────┤
| Heavy Forwarder  | Can parse/filter before sending      |
| (HF)             | Used at network edge                 |
|                  | Higher resource usage                |
├──────────────────┼──────────────────────────────────────┤
| Syslog Input     | Receives syslog from firewalls,      |
|                  | network devices                      |
└──────────────────┴──────────────────────────────────────┘
```

2. Indexers (Storage & Processing)
   Receives, parses, indexes, and stores logs. This is where your data lives.

```
   Raw Log:
"Mar 15 10:23:45 webserver sshd[12345]: Failed password for admin from
192.168.1.50"

         |
         ▼  [INDEXING PROCESS]

┌──────────────────────────────────────────────────────┐
|  _time = 2024-03-15 10:23:45                         |
|  host = webserver                                     |
|  source = /var/log/auth.log                          |
|  sourcetype = syslog                                 |
|  _raw = "Failed password for admin from 192.168.1.50"  |
└──────────────────────────────────────────────────────┘
         |
         ▼
    Stored in index (like a database)
```

3. Search Head (Query Interface)
   This is where the analyst work!

```
 -- Example Splunk Search (SPL)
index=security sourcetype=WinEventLog:Security EventCode=4625
| stats count by src_ip, user
| where count > 10
| sort -count
```

This search finds: Brute force attempts (more than 10 failed logins per IP/user)

Log Sources (On-Prem vs Cloud)

On-Premises Log Sources:

```
┌──────────────────────────────────────────────────────────────────┐
│                    ON-PREM LOG SOURCES                           │
├──────────────────────┬───────────────────────────────────────────┤
│ Windows Servers      │ Security, System, Application Event Logs  │
├──────────────────────┼───────────────────────────────────────────┤
│ Linux Servers        │ /var/log/auth.log, /var/log/syslog        │
├──────────────────────┼───────────────────────────────────────────┤
│ Firewalls            │ Palo Alto, Fortinet, Cisco ASA logs       │
├──────────────────────┼───────────────────────────────────────────┤
│ Active Directory     │ DC Security logs (4624, 4625, 4768...)    │
├──────────────────────┼───────────────────────────────────────────┤
│ Proxy/Web Filter     │ Zscaler, BlueCoat, Squid logs             │
├──────────────────────┼───────────────────────────────────────────┤
│ Email Gateway        │ Proofpoint, Mimecast logs                 │
├──────────────────────┼───────────────────────────────────────────┤
│ EDR                  │ CrowdStrike, Defender, SentinelOne        │
└──────────────────────┴───────────────────────────────────────────┘
```

Cloud Log Sources

Critical cloud logs you MUST know:

AWS

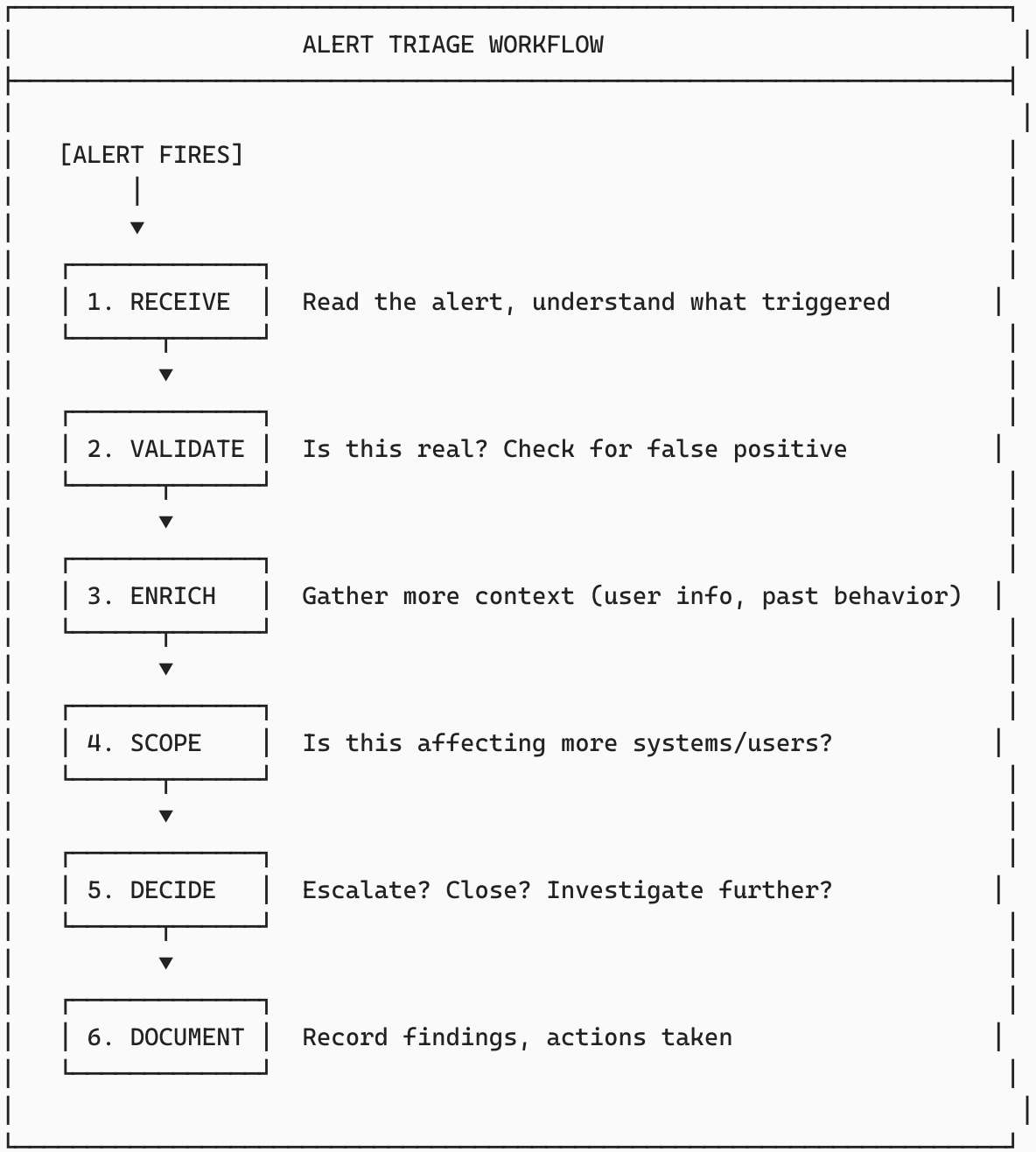
CloudTrail (MUST) → Every API call (who logged into console, who launched EC2, etc.)
VPC Flow Logs → Network traffic
GuardDuty → Built-in threat detection
S3 Access Logs

Azure

Azure Activity Log → Control plane (who created a VM, changed RBAC)
Sign-in Logs (Entra ID) → Critical for identity attacks
NSG Flow Logs → Network
Microsoft Defender for Cloud alerts

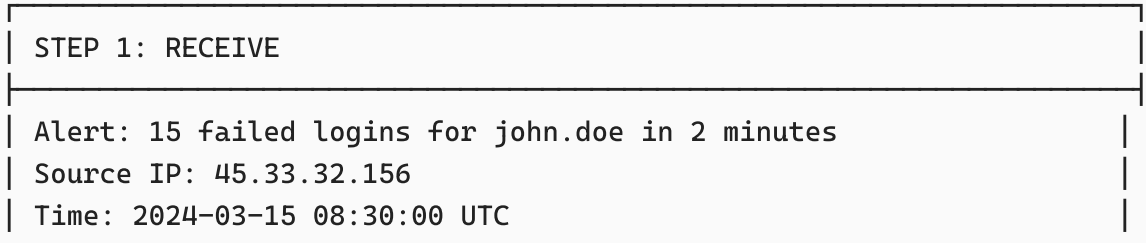
Alert Triage & Investigation Process

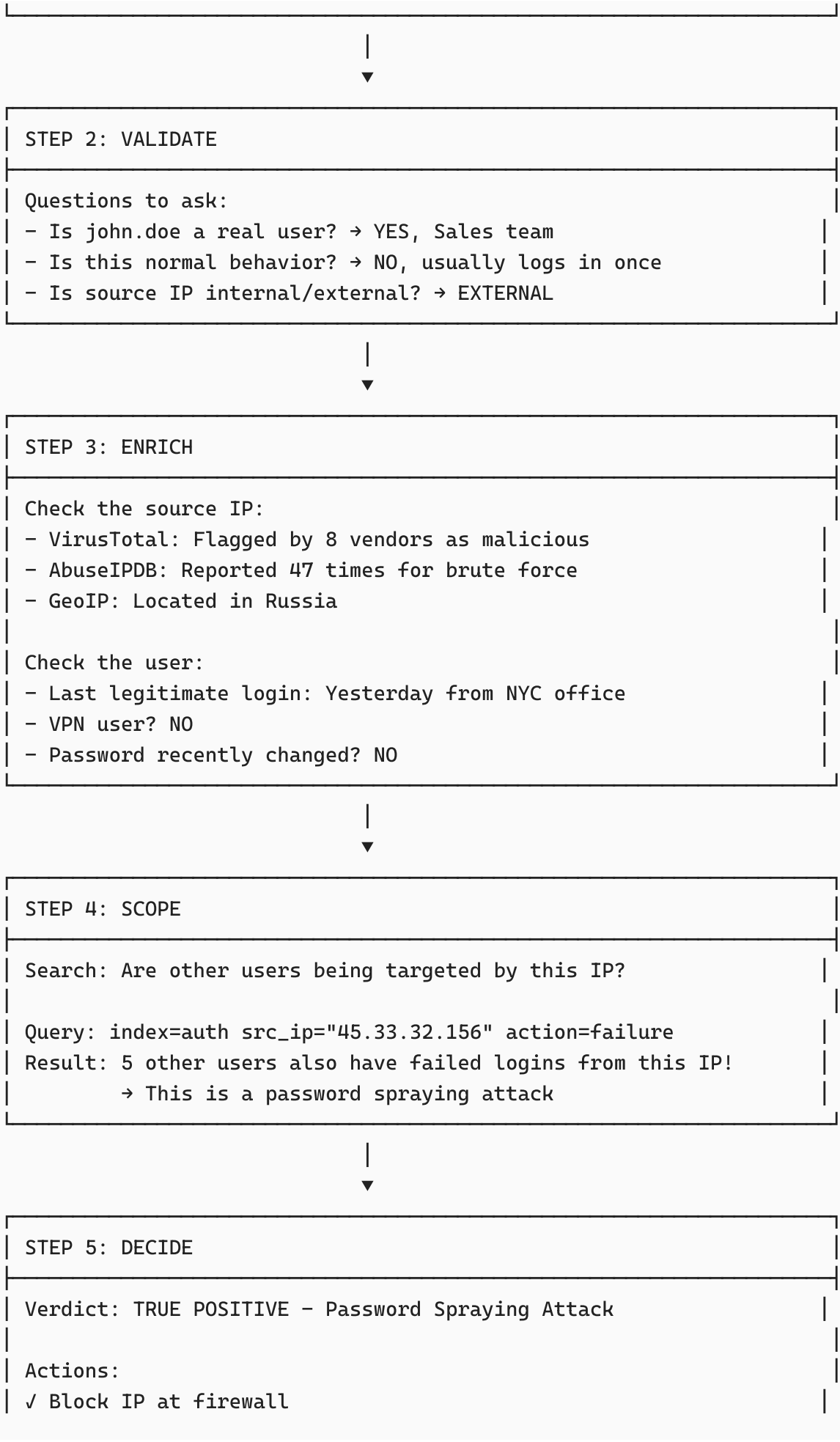
The Triage Workflow:



Real-World Example:

Alert: Multiple failed logins detected for user john.doe




STEP 2: VALIDATE
Questions to ask:
- Is john.doe a real user? → YES, Sales team
- Is this normal behavior? → NO, usually logs in once
- Is source IP internal/external? → EXTERNAL
STEP 3: ENRICH
Check the source IP:
- VirusTotal: Flagged by 8 vendors as malicious
- AbuseIPDB: Reported 47 times for brute force
- GeoIP: Located in Russia
Check the user:
- Last legitimate login: Yesterday from NYC office
- VPN user? NO
- Password recently changed? NO
STEP 4: SCOPE
Search: Are other users being targeted by this IP?
Query: index=auth src_ip="45.33.32.156" action=failure
Result: 5 other users also have failed logins from this IP!
→ This is a password spraying attack
STEP 5: DECIDE
Verdict: TRUE POSITIVE - Password Spraying Attack
Actions:
✓ Block IP at firewall

```
| ✓ Check if any login succeeded (DATA BREACH?)                          |
| ✓ Force password reset for targeted users                              |
| ✓ Escalate to L2/Incident Response                                     |
└────────────────────────────────────────────────────────────────────────┘
                                  |
                                  ▼
┌────────────────────────────────────────────────────────────────────────┐
| STEP 6: DOCUMENT                                                       |
├────────────────────────────────────────────────────────────────────────┤
| Ticket #12345                                                          |
| Summary: Password spraying attack from 45.33.32.156                    |
| Affected: 6 users (john.doe, jane.smith, etc.)                         |
| Actions: IP blocked, passwords reset, escalated to IR                  |
| MITRE: T1110.003 (Password Spraying)                                   |
| Status: Escalated                                                      |
└────────────────────────────────────────────────────────────────────────┘
```

False Positives & How to Reduce Them

What is a False Positive

```
┌──────────────────────────────────────────────────────────────┐
|                    ALERT OUTCOMES                            |
├──────────────────────┬───────────────────────────────────────┤
| TRUE POSITIVE (TP)   | Alert fired + Actually malicious      |
|                      | "Correct alert - real attack"         |
├──────────────────────┼───────────────────────────────────────┤
| FALSE POSITIVE (FP)  | Alert fired + NOT malicious           |
|                      | "Wrong alert - wasted time"           |
├──────────────────────┼───────────────────────────────────────┤
| TRUE NEGATIVE (TN)   | No alert + Nothing malicious          |
|                      | "Correctly quiet"                     |
├──────────────────────┼───────────────────────────────────────┤
| FALSE NEGATIVE (FN)  | No alert + Actually malicious         |
|                      | "WORST CASE - missed attack!"         |
└──────────────────────┴───────────────────────────────────────┘
```

Common Causes of False Positives:

```
┌──────────────────────────┬─────────────────────────────────────┐
| CAUSE                    | EXAMPLE                             |
├──────────────────────────┼─────────────────────────────────────┤
```

Overly broad rules	Alert on ANY PowerShell execution
Legitimate admin activity	IT admin running vulnerability scan
Known business processes	Backup server connecting to many IPs
Outdated threat intel	Blocking IP that's now legitimate
Time zone issues	"After hours login" for remote worker

How to Reduce False Positives:

1. Tuning Alert Rules

```
-- BAD RULE (Too Broad):
index=windows EventCode=4688 process_name="powershell.exe"
| alert

-- GOOD RULE (More Specific):
index=windows EventCode=4688 process_name="powershell.exe"
| where NOT match(user, "^(svc_|admin_)")     -- Exclude service accounts
| where NOT match(parent_process, "SCCM")      -- Exclude known tools
| where match(command_line, "-enc|-nop|-exec bypass")  -- Suspicious flags
| alert
```

2. Whitelisting/Allowlisting

```
┌──────────────────────────────────────────────────────────┐
│                  WHITELIST EXAMPLES                      │
├──────────────────────────────────────────────────────────┤
│ IP Whitelist:                                            │
│   - Vulnerability scanner: 10.1.1.50                     │
│   - Backup server: 10.1.1.100                            │
│                                                          │
│ User Whitelist:                                          │
│   - Service accounts: svc_backup, svc_monitoring         │
│                                                          │
│ Process Whitelist:                                       │
│   - Known admin tools: psexec.exe (when run by IT)       │
└──────────────────────────────────────────────────────────┘
```

3. Adding Context to Alerts

```
Instead of: "Failed login detected"

Better:     "Failed login detected"
            + User's normal login pattern
            + Source IP reputation
            + Time of day analysis
            + Number of failures in timeframe
```

4. Risk-Based Alerting

```
┌─────────────────────────────────────────────────────────────────┐
│                    RISK SCORING                                 │
├─────────────────────────────────────────────────────────────────┤
│                                                                 │
│   Event: PowerShell execution                    +10 points    │
│   + Encoded command                              +20 points    │
│   + Running from temp folder                     +15 points    │
│   + User is not IT admin                         +25 points    │
│   + First time this user ran PowerShell         +30 points     │
│   ──────────────────────────────────────────────────────       │
│   TOTAL RISK SCORE:                              100 points    │
│                                                                 │
│   Threshold for alert: 50 points                               │
│   Verdict: ALERT! (Score exceeds threshold)                    │
│                                                                 │
└─────────────────────────────────────────────────────────────────┘
```

SOAR & Playbooks

SOAR = Security Orchestration, Automation, and Response

```
┌─────────────────────────────────────────────────────────────────┐
│                        SOAR EXPLAINED                           │
├─────────────────────────────────────────────────────────────────┤
│                                                                 │
│  ORCHESTRATION:  Connect all your security tools together      │
│                  (SIEM ↔ EDR ↔ Firewall ↔ Ticketing)           │
│                                                                 │
│  AUTOMATION:     Run tasks automatically without human input   │
│                  (Enrich IOCs, block IPs, disable accounts)    │
│                                                                 │
│  RESPONSE:       Take action to contain/remediate threats      │
```

```
│                 (Isolate endpoint, revoke session)              │
│                                                                 │
└─────────────────────────────────────────────────────────────────┘
```

Popular SOAR Platforms

Platform	Notes	Why it's used in 2026
Splunk SOAR (Phantom)	Deeply integrates with Splunk SIEM.	Known for high-level "Visual Playbooks" and massive scale.
Microsoft Sentinel	Uses "Logic Apps" for automation.	The default for Azure-heavy environments; easy to set up.
Cortex XSOAR (Palo Alto)	The current market leader.	Has the largest library of pre-built integrations (over 1,000+).
Swimlane	Cloud-native, low-code platform.	Popular for its "Turbine" engine which handles high-speed automation.
Tines	API-first, "No-code" automation.	Extremely flexible; loved by analysts because it works with any tool.

Playbook

A playbook is a documented set of steps to respond to a specific type of alert.

```
┌─────────────────────────────────────────────────────────────────┐
│              PLAYBOOK: Phishing Email Response                  │
├─────────────────────────────────────────────────────────────────┤
│                                                                 │
│  TRIGGER: User reports suspicious email                         │
│                                                                 │
│  STEP 1: Extract IOCs from email                                │
│          ├── Sender address                                     │
│          ├── URLs in body                                       │
│          ├── Attachment hashes                                  │
│                                                                 │
│  STEP 2: Check IOCs against threat intel                        │
│          ├── VirusTotal lookup                                  │
│          ├── URLScan.io check                                   │
│          ├── Internal blocklist check                           │
│                                                                 │
│  STEP 3: If malicious:                                          │
│          ├── Block sender domain in email gateway               │
```

```
│          ├── Block URL in proxy                                    │
│          ├── Search for other recipients                           │
│          ├── Delete email from all mailboxes                       │
│                                                                    │
│  STEP 4: If clicked:                                               │
│          ├── Isolate user's endpoint                               │
│          ├── Force password reset                                  │
│          ├── Revoke active sessions                                │
│                                                                    │
│  STEP 5: Document and close ticket                                 │
│                                                                    │
└────────────────────────────────────────────────────────────────────┘
```

example of automation

```python
import requests

def check_abuseipdb(ip, api_key):
    url = f"https://api.abuseipdb.com/api/v2/check"
    headers = {'Key': api_key, 'Accept': 'application/json'}
    params = {'ipAddress': ip, 'maxAgeInDays': 90}
    r = requests.get(url, headers=headers, params=params)
    return r.json()['data']['abuseConfidenceScore']

# In playbook: if score > 85 → auto-block on firewall
```

IOCs (Indicators of Compromise)

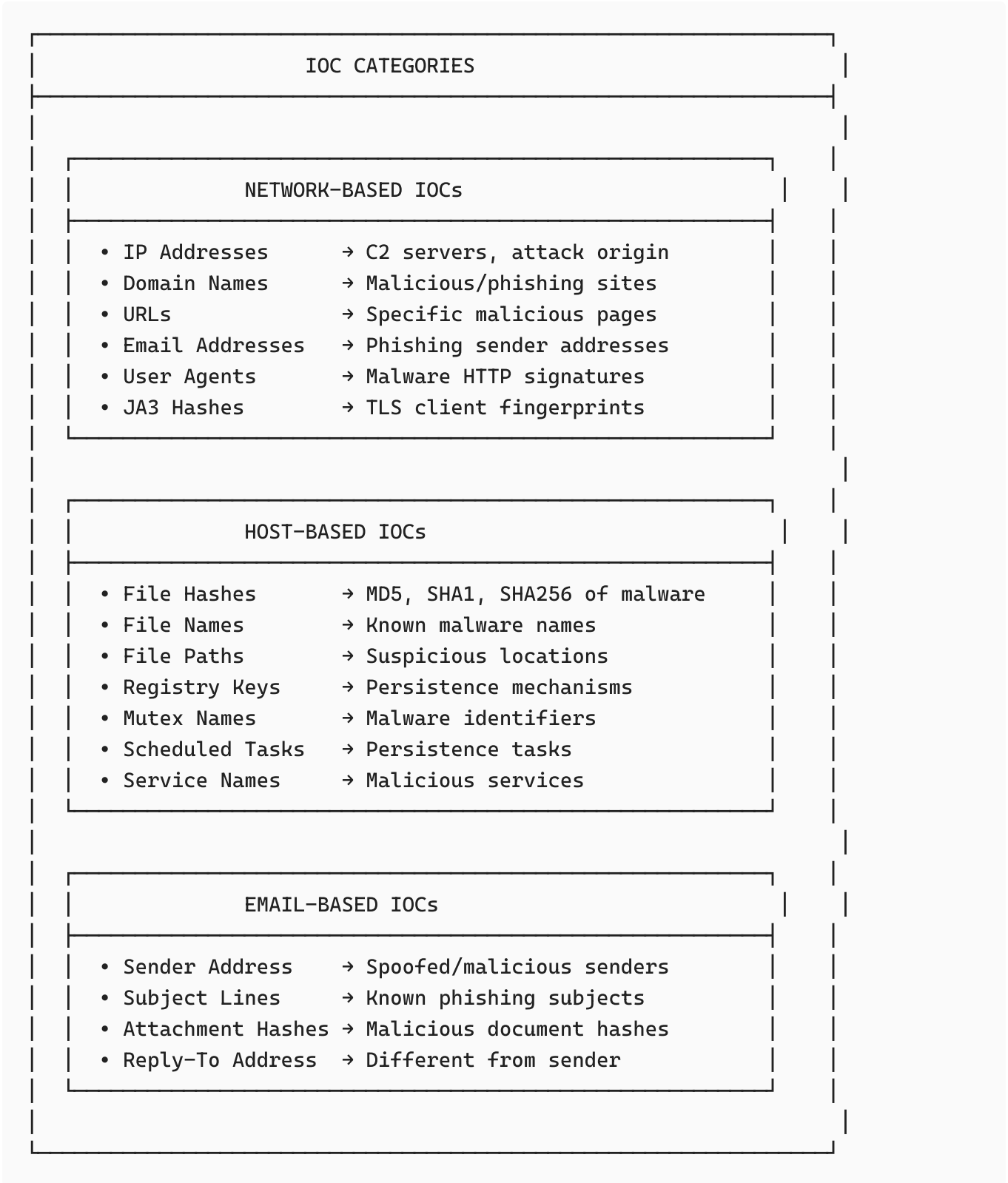
IOC = Evidence that a system has been compromised

IOCs are pieces of forensic data (evidence) that indicate a computer or network has already been breached. They are usually static—meaning they don't change based on context.

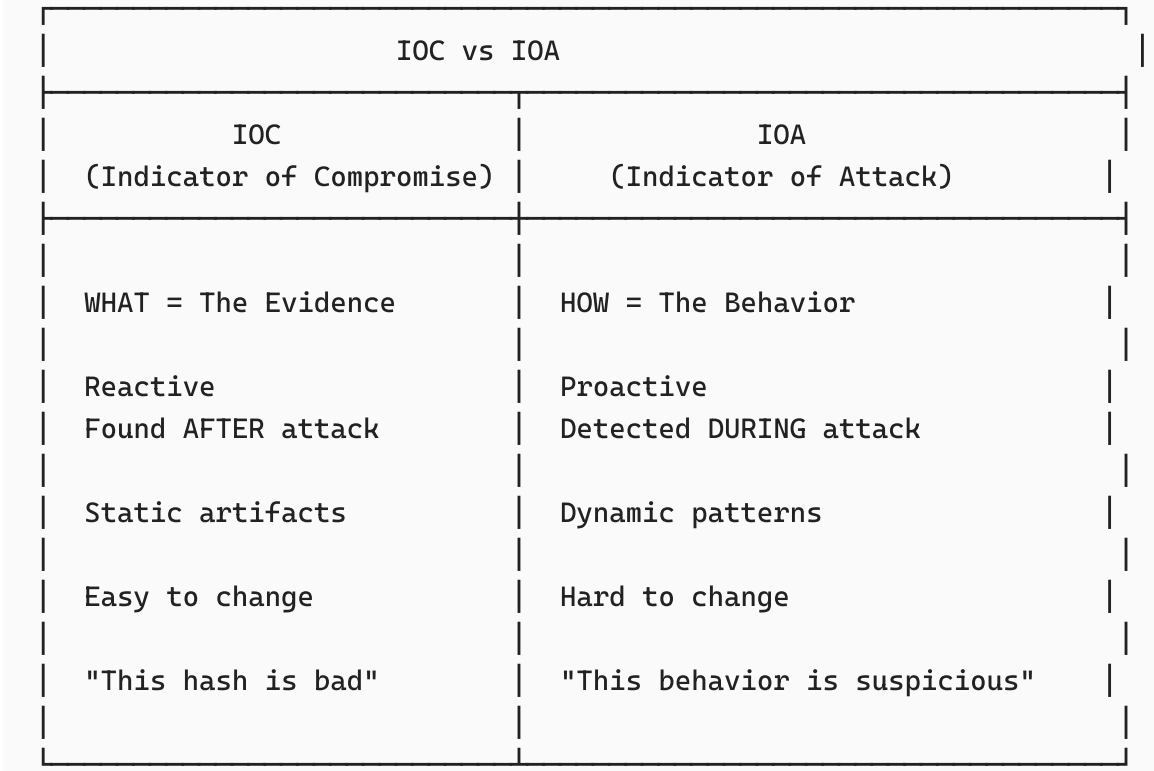
```
┌────────────────────────────────────────────────────────────────────┐
│                                                                    │
│   PHYSICAL CRIME SCENE          CYBER ATTACK                       │
│   ═══════════════════          ═══════════                         │
│                                                                    │
│   Fingerprints          ──►     File Hashes                        │
│   License Plate         ──►     IP Addresses                       │
│   Phone Number          ──►     Domain Names                       │
│   DNA Evidence          ──►     Malware Signatures                 │
```

```
┌──────────────────────────────────────────────────────────────┐
│   Bullet Casings        ──────►     Registry Keys            │
│   Ransom Note           ──────►     Ransom Note Files        │
│                                                              │
└──────────────────────────────────────────────────────────────┘
```

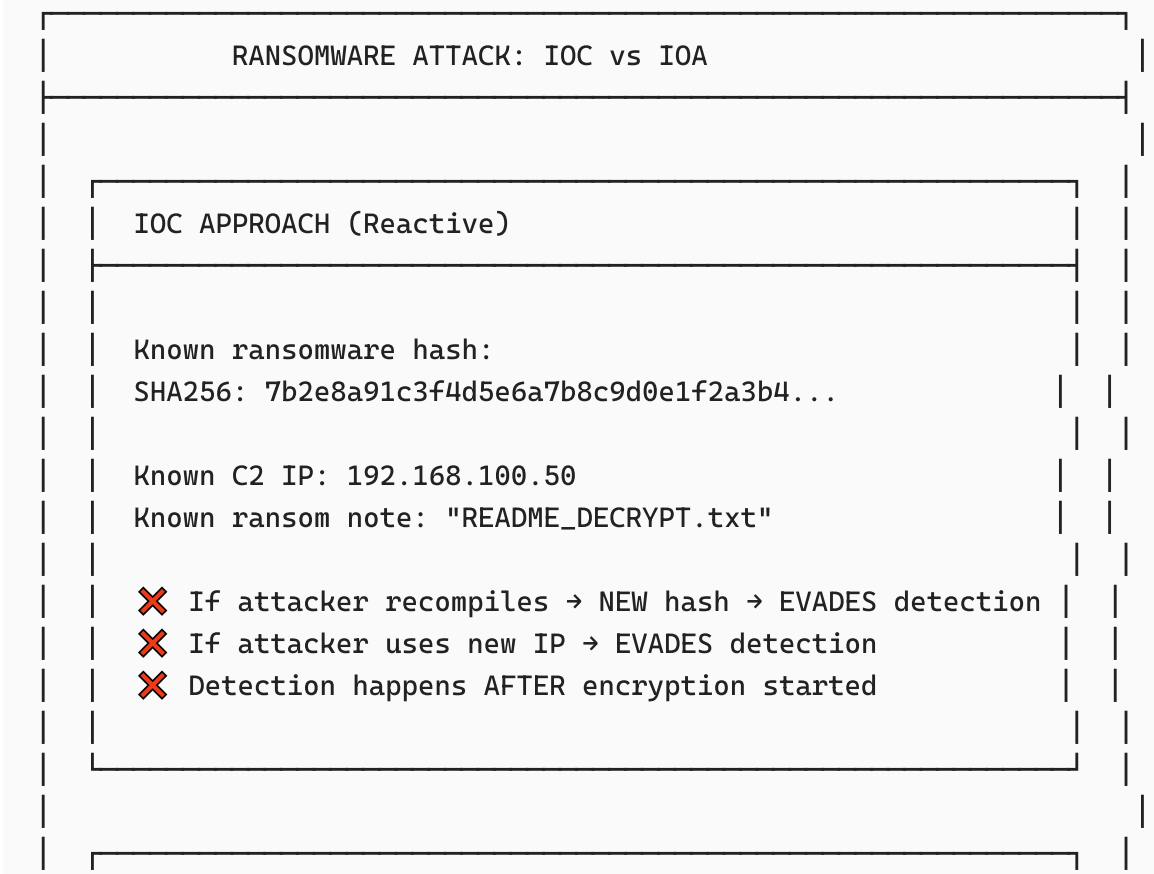
Types of IOCs

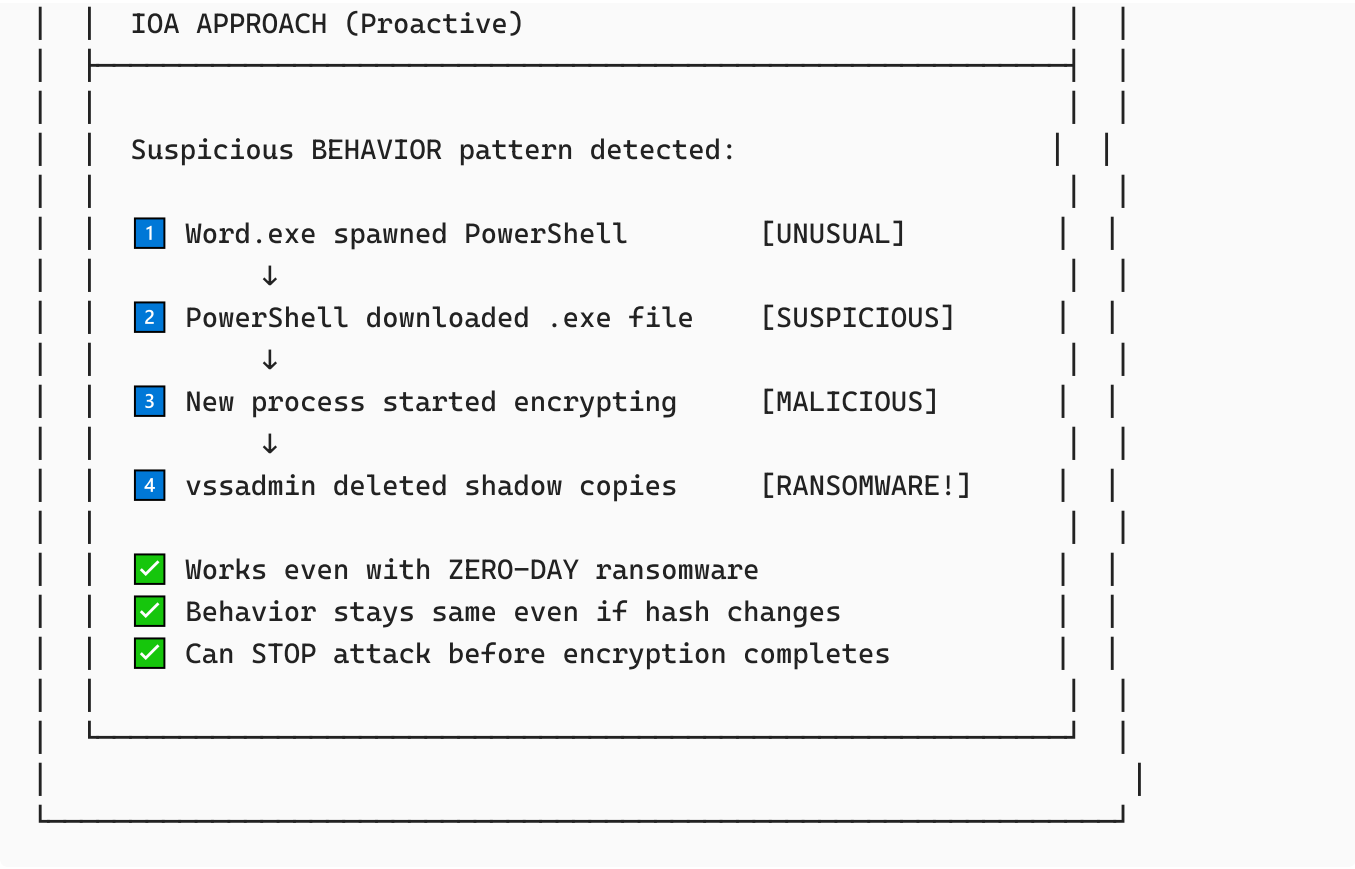


IOC vs IOA (Indicators of Attack)



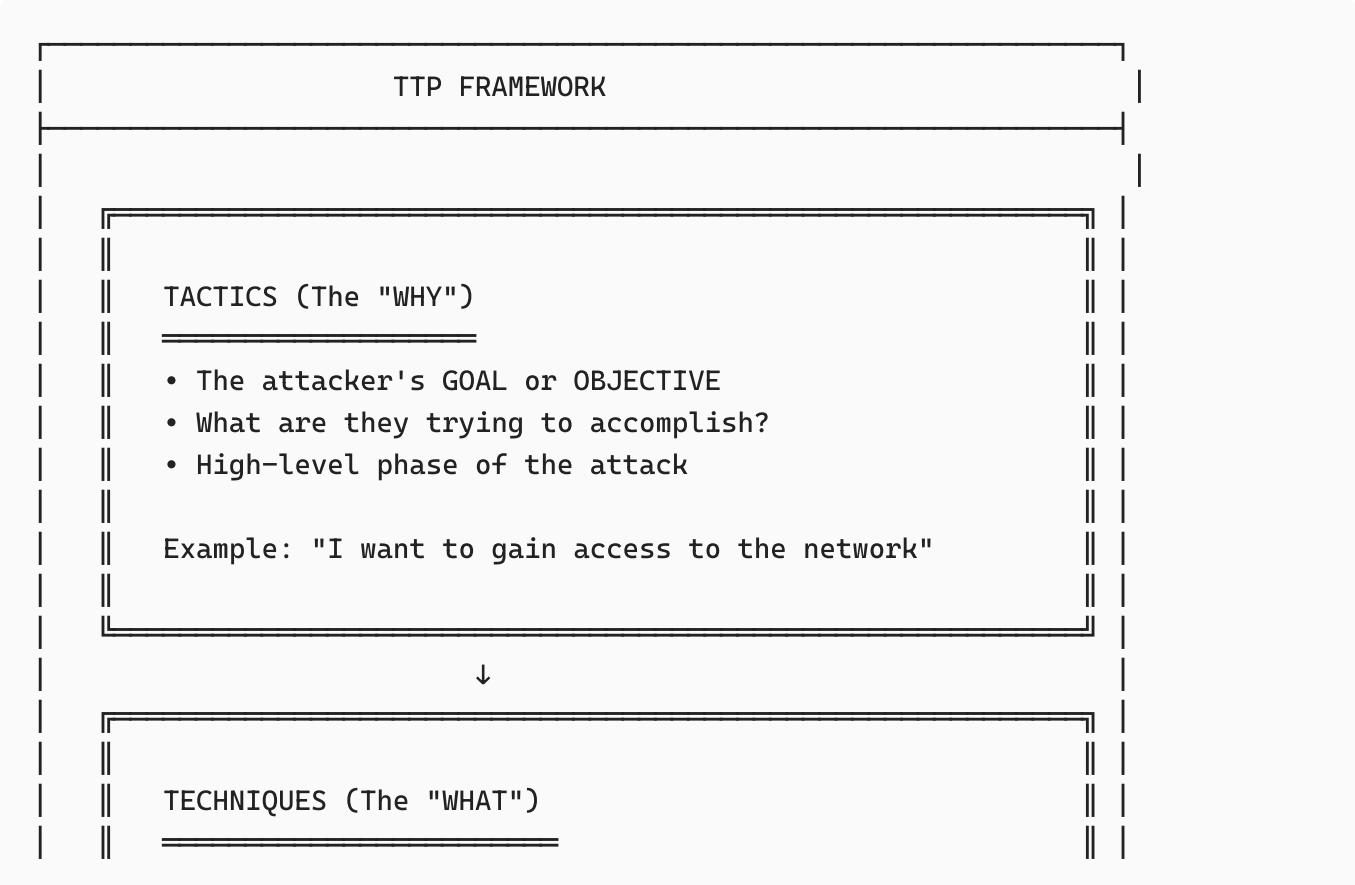
example:

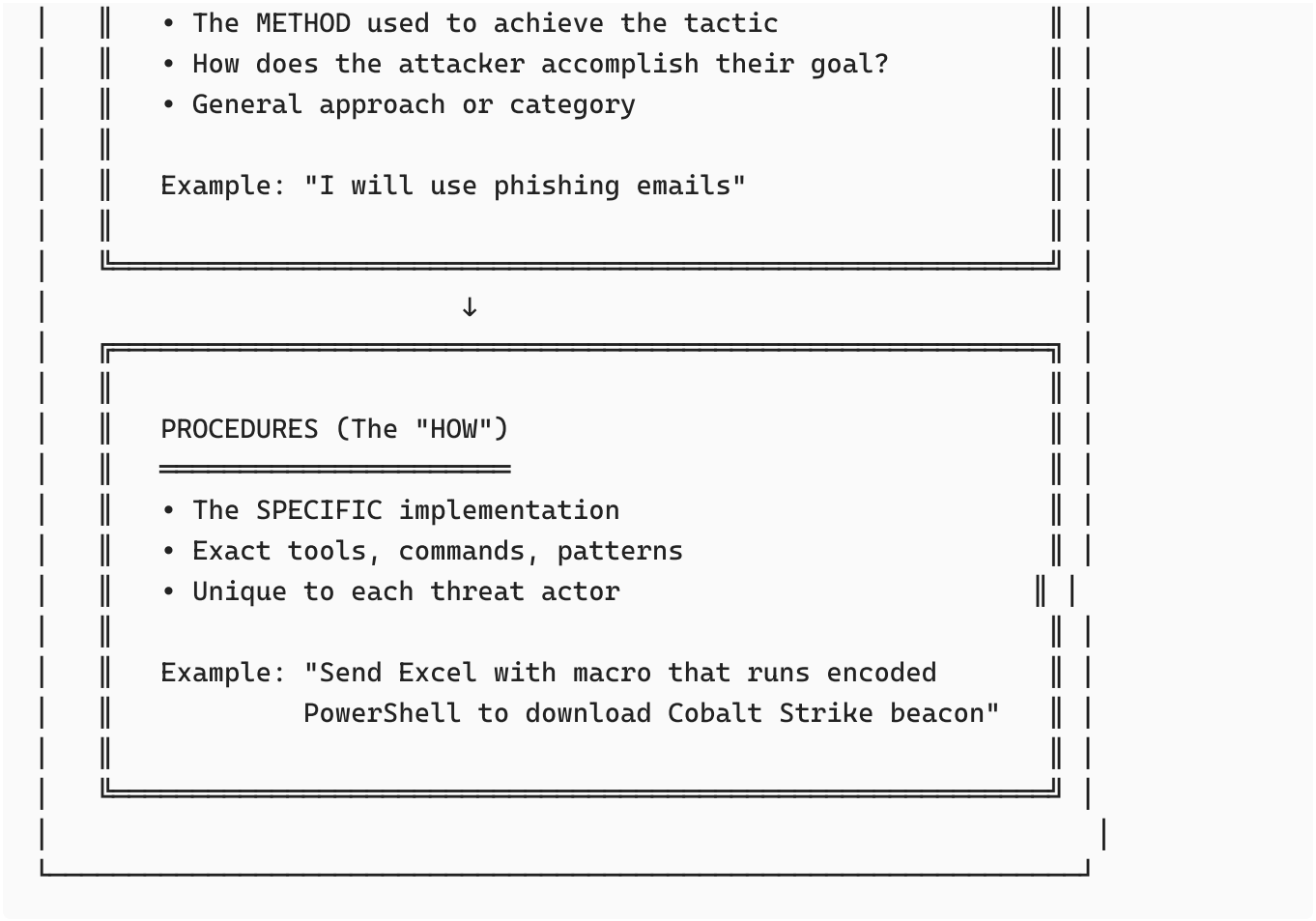




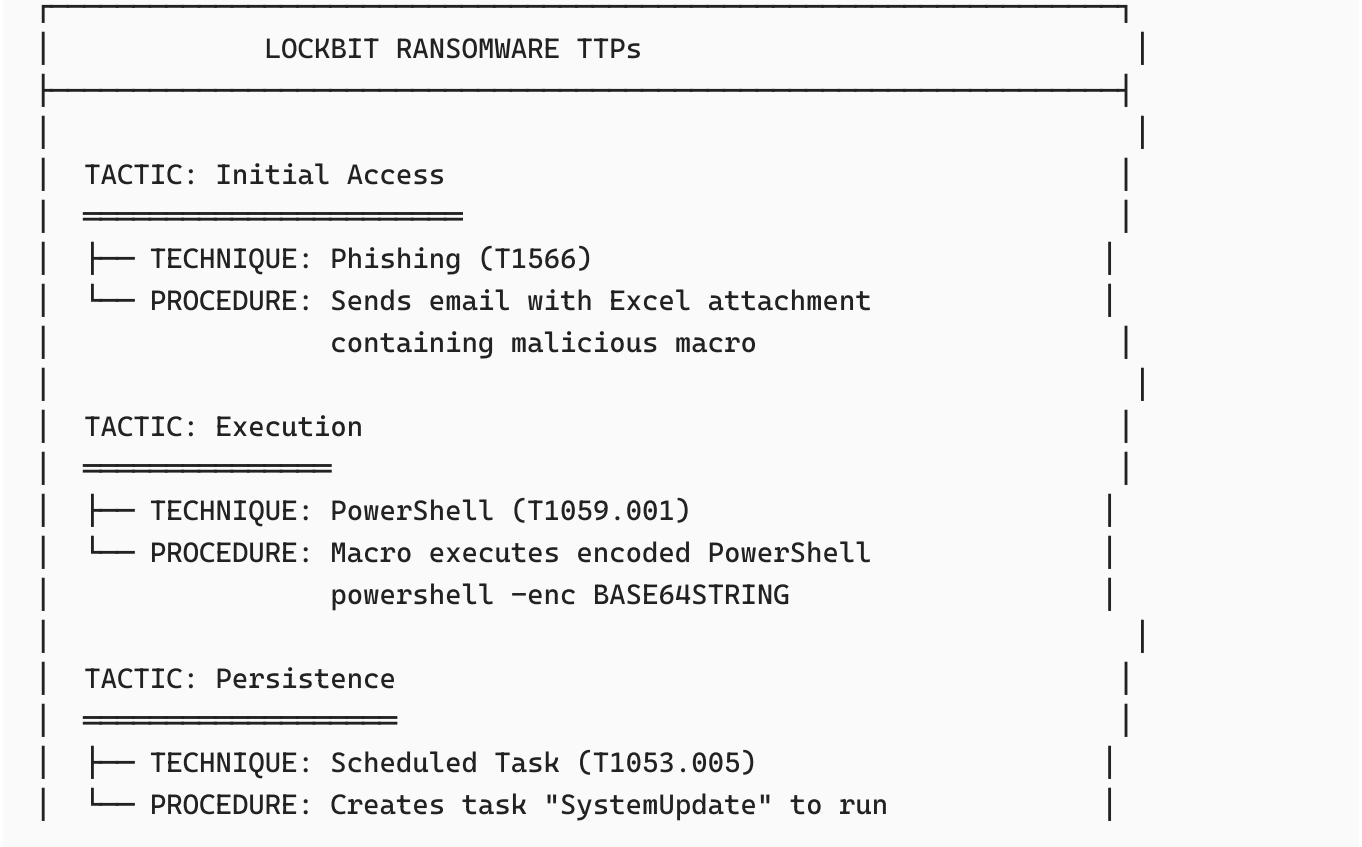
TTPs = Tactics, Techniques, and Procedures

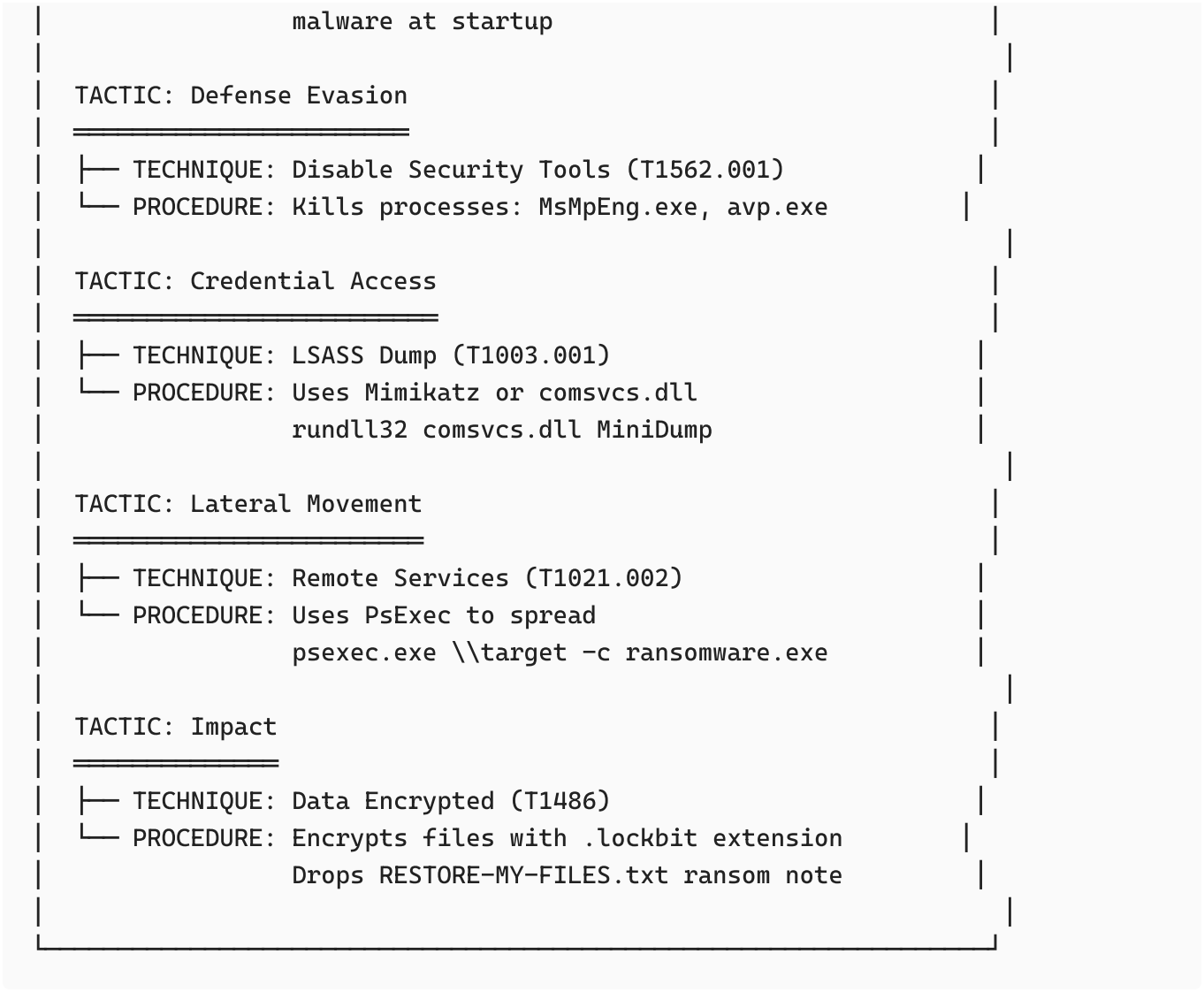
This is the behavioral fingerprint of how attackers operate.





Ransomware Attack TTPs




malware at startup
TACTIC: Defense Evasion
TECHNIQUE: Disable Security Tools (T1562.001)
PROCEDURE: Kills processes: MsMpEng.exe, avp.exe
TACTIC: Credential Access
TECHNIQUE: LSASS Dump (T1003.001)
PROCEDURE: Uses Mimikatz or comsvcs.dll
rundll32 comsvcs.dll MiniDump
TACTIC: Lateral Movement
TECHNIQUE: Remote Services (T1021.002)
PROCEDURE: Uses PsExec to spread
psexec.exe \\target -c ransomware.exe
TACTIC: Impact
TECHNIQUE: Data Encrypted (T1486)
PROCEDURE: Encrypts files with .lockbit extension
Drops RESTORE-MY-FILES.txt ransom note