Salt Typhoon Threat Hunting Runbook

1) Purpose

This threat hunting runbook provides Microsoft Sentinel KQL queries and investigative procedures to detect and hunt for Salt Typhoon (OPERATOR PANDA, RedMike, UNC5807, GhostEmperor, Earth Estries, FamousSparrow, UNC2286) threat actor activities. The runbook focuses on their signature tactics including exploiting public-facing applications, TACACS+ targeting, network device exploitation, and custom SFTP client usage for espionage operations targeting telecommunications and critical infrastructure.

2) Threat Context

Actor Information

- **Primary Name**: Salt Typhoon
- **Aliases**: OPERATOR PANDA, RedMike, UNC5807, GhostEmperor, Earth Estries, FamousSparrow, UNC2286
- **Composition**: Chinese state-sponsored advanced persistent threat (APT) group
- **Geographic Focus**: Global telecommunications, government, transportation, lodging, and military sectors
- **First Observed**: 2021
- **Active Since**: 2021 - Present

Motivation

- **Espionage** - Gathering strategic information from global telecommunications and critical infrastructure providers
- **Intelligence Collection** - Targeting subscriber information, user content, customer records and metadata, network diagrams
- **Network Access** - Establishing persistent access to critical infrastructure for surveillance and data extraction

- **Strategic Information** - Collecting vendor lists, device configurations, and passwords for operational advantage

Key TTPs

- **T1190**: Exploit Public-Facing Application - Primary initial access vector exploiting CVEs in VPNs, firewalls, and email servers
- **T1199**: Trusted Relationship - Leveraging provider-to-provider connections for lateral movement
- **T1040**: Network Sniffing - TACACS+ traffic capture and credential harvesting from network devices
- **T1556**: Modify Authentication Process - Redirecting TACACS+ servers to actor-controlled infrastructure
- **T1610**: Deploy Container - Using Cisco Guest Shell for persistence and evasion
- **T1602.001**: SNMP (MIB Dump) - Network reconnaissance via SNMP enumeration
- **T1572**: Protocol Tunneling - GRE/IPsec tunnels for covert C2 and data exfiltration
- **T1003.001**: LSASS Memory - Mimikatz usage for Windows credential dumping **(2025 Evolution)**
- **T1574.001**: DLL Side-Loading - Advanced persistence technique **(2025 New Technique)**

3) Technical Prerequisites for Threat Hunting

Required Data Sources

- Microsoft Defender for Endpoint (MDE)
- Microsoft Sentinel (SecurityEvent, DeviceEvents, DeviceProcessEvents, DeviceNetworkEvents, DeviceFileEvents)
- Network Device Logs (Cisco IOS/NX-OS, authentication servers)
- TACACS+/RADIUS Authentication Logs
- SNMP Monitoring Data
- Windows Security Event Logs (Event IDs 4624, 4625, 4672, 4768, 4769)
- PowerShell Operational Logs
- Process Creation Events (Sysmon Event ID 1)

Recommended Log Retention

- Minimum 90 days for correlation analysis
- 180 days recommended for campaign tracking

4) Threat Hunting Hypotheses

Hypothesis 1: CVE Exploitation Detection

Mapping: T1190 - Exploit Public-Facing Application **Hypothesis Explanation**: Salt Typhoon primarily gains initial access by exploiting publicly known CVEs in network devices, VPNs, and web applications. They target vulnerabilities like CVE-2024-21887 (Ivanti), CVE-2024-3400 (Palo Alto), and CVE-2023-20273/CVE-2023-20198 (Cisco IOS XE). **Hunting Focus**: Web requests with exploitation patterns, unusual authentication bypasses, and post-exploitation command execution

```
// Requires: DeviceNetworkEvents table
// Detects: CVE exploitation patterns in web requests targeting network
```

```
devices
// Optimized: Uses has_any for better performance, proper time filtering,
and result limiting
DeviceNetworkEvents
| where Timestamp > ago(30d)
| where RemoteUrl has_any ("webui_wsma", "%2577ebui_wsma",
"%2577eb%2575i_%2577sma", "admin", "config", "/api/", "%2f", "%252f")
| summarize EventCount = count(),
    FirstSeen = min(Timestamp),
    LastSeen = max(Timestamp),
    UniqueUrls = dcount(RemoteUrl)
    by DeviceName, RemoteIP, InitiatingProcessFileName
| where EventCount > 1
| extend RiskScore = case(
    EventCount > 100, "High",
    EventCount > 50, "Medium",
    "Low"
)
| order by EventCount desc
| take 500
```

Investigation Steps:

1. Verify if the target device is a Cisco IOS XE system with web management enabled
2. Check for subsequent privilege escalation attempts or unauthorized user creation
3. Review authentication logs for failed/successful admin access attempts
4. Examine network traffic for additional exploitation attempts from the same source IP
5. Search for Guest Shell activation or container deployment activities

Hypothesis 2: TACACS+ Traffic Interception

Mapping: T1040 - Network Sniffing, T1556 - Modify Authentication Process **Hypothesis Explanation**: Salt Typhoon targets TACACS+ authentication traffic on TCP port 49 to capture credentials and modifies router configurations to redirect authentication to actor-controlled servers for credential harvesting. **Hunting Focus**: TACACS+ configuration changes, unusual authentication server destinations, and packet capture activities

```
// Requires: DeviceNetworkEvents table
// Detects: TACACS+ authentication traffic interception on TCP port 49
// Optimized: Combined port filters, added null checks, improved
aggregation
DeviceNetworkEvents
| where Timestamp > ago(30d)
| where (RemotePort == 49 or LocalPort == 49) and Protocol == "Tcp"
| summarize ConnectionCount = count(),
    UniqueDestinations = dcount(RemoteIP),
    FirstConnection = min(Timestamp),
    LastConnection = max(Timestamp)
    by DeviceName, InitiatingProcessFileName
| where ConnectionCount > 0
```

```
| extend RiskScore = case(
    UniqueDestinations > 5, "High",
    ConnectionCount > 10, "Medium",
    "Low"
)
| order by ConnectionCount desc
| take 500
```

Investigation Steps:

1. Identify if TACACS+ server configurations have been modified recently
2. Check for packet capture commands targeting TCP port 49 traffic
3. Verify legitimacy of destination TACACS+ servers in network connections
4. Review router configuration backups for unauthorized changes
5. Correlate with authentication failures or successful logins from unusual sources

Hypothesis 3: Guest Shell Container Deployment

Mapping: T1610 - Deploy Container **Hypothesis Explanation**: Salt Typhoon leverages Cisco Guest Shell containers on IOS XE and NX-OS devices to stage tools, process data locally, and evade detection. Container activities are not closely monitored, making this an effective persistence mechanism. **Hunting Focus**: Guest Shell activation commands, container-based tool execution, and Python script deployment

```
// Requires: DeviceProcessEvents table
// Detects: Guest Shell container deployment and execution on Cisco
devices
// Optimized: Uses has_any for better performance on multiple string
matches
DeviceProcessEvents
| where Timestamp > ago(30d)
| where ProcessCommandLine has_any ("guestshell", "dohost", "container")
| extend GuestShellAction = case(
    ProcessCommandLine contains "enable", "Activation",
    ProcessCommandLine contains "run", "Execution",
    ProcessCommandLine contains "disable", "Deactivation",
    ProcessCommandLine contains "destroy", "Cleanup",
    "Other"
)
| summarize ActionCount = count(),
    UniqueActions = make_set(GuestShellAction),
    FirstActivity = min(Timestamp),
    LastActivity = max(Timestamp)
    by DeviceName, AccountName
| where ActionCount > 0
| extend RiskScore = case(
    ActionCount > 5, "High",
    array_length(UniqueActions) > 2, "Medium",
    "Low"
)
```

```
| order by ActionCount desc
| take 500
```

Investigation Steps:

1. Verify if Guest Shell is authorized for legitimate administrative use
2. Check for Python script execution within Guest Shell environments
3. Review file system changes in Guest Shell directories (/bootflash, /tmp)
4. Examine network connections originating from container processes
5. Correlate Guest Shell activities with other suspicious network device behaviors

Hypothesis 4: Custom SFTP Client Usage

Mapping: T1105 - Ingress Tool Transfer, T1048.002 - Exfiltration Over Alternative Protocol **Hypothesis Explanation**: Salt Typhoon uses custom Golang SFTP clients (cmd1, cmd3, new2, sft) for encrypted archive transfer and collecting network packet captures. These tools are specifically designed for their operations. **Hunting Focus**: Custom SFTP binary execution, encrypted file transfers, and data staging activities

```
// Requires: DeviceFileEvents table
// Detects: Custom SFTP client binaries used by Salt Typhoon
// Optimized: Improved hash comparison using in~ operator for case-
insensitive matching
DeviceFileEvents
| where Timestamp > ago(30d)
| where (FileName in~ ("cmd1", "cmd3", "new2", "sft") and ActionType ==
"FileCreated")
    or SHA256 in~
("8b448f47e36909f3a921b4ff803cf3a61985d8a10f0fe594b405b92ed0fc21f1",

"f2bbba1ea0f34b262f158ff31e00d39d89bbc471d04e8fca60a034cabe18e4f4",

"da692ea0b7f24e31696f8b4fe8a130dbbe3c7c15cea6bde24cccc1fb0a73ae9e",

"a1abc3d11c16ae83b9a7cf62ebe6d144dfc5e19b579a99bad062a9d31cf30bfe")
| extend ThreatIndicator = case(
    FileName in~ ("cmd1", "cmd3", "new2", "sft"), "Custom SFTP Client",
    isnotempty(SHA256), "Known Malicious Hash",
    "Unknown"
)
| summarize FileEvents = count(),
    UniqueFiles = dcount(FileName),
    FirstSeen = min(Timestamp),
    LastSeen = max(Timestamp)
    by DeviceName, FolderPath, ThreatIndicator
| extend RiskScore = case(
    ThreatIndicator == "Known Malicious Hash", "Critical",
    ThreatIndicator == "Custom SFTP Client", "High",
    "Medium"
)
```

```
| order by RiskScore, FileEvents desc
| take 500
```

Investigation Steps:

1. Quarantine and analyze identified SFTP client binaries
2. Review network connections from processes executing these binaries
3. Check for associated encrypted archive files or packet captures
4. Examine process parent-child relationships for execution context
5. Search for similar file patterns or staging directories across the network

Hypothesis 5: SNMP Reconnaissance and Manipulation

Mapping: T1016 - System Network Configuration Discovery, T1602.001 - SNMP (MIB Dump) **Hypothesis Explanation**: Salt Typhoon uses SNMP for network reconnaissance, device enumeration, and configuration manipulation. They leverage SNMPwalk for discovery and SNMP SET requests to modify device configurations. **Hunting Focus**: Excessive SNMP queries, configuration changes via SNMP, and cross-device enumeration activities

```
// Requires: DeviceNetworkEvents table
// Detects: SNMP reconnaissance and enumeration activities
// Optimized: Combined port and protocol filters, improved aggregation
logic
DeviceNetworkEvents
| where Timestamp > ago(30d)
| where (RemotePort == 161 or LocalPort == 161) and Protocol == "Udp"
| extend SNMPDirection = case(
    RemotePort == 161, "Outbound",
    "Inbound"
)
| summarize SNMPConnections = count(),
    UniqueTargets = dcount(RemoteIP),
    FirstActivity = min(Timestamp),
    LastActivity = max(Timestamp)
    by DeviceName, SNMPDirection, InitiatingProcessFileName
| where SNMPConnections > 10 or UniqueTargets > 3
| extend RiskScore = case(
    SNMPConnections > 500, "Critical",
    UniqueTargets > 50, "High",
    SNMPConnections > 100, "Medium",
    "Low"
)
| order by SNMPConnections desc
| take 500
```

Investigation Steps:

1. Identify SNMP community strings and access controls in use
2. Review SNMP configuration changes and unauthorized SET operations

3. Check for bulk SNMP polling patterns indicating reconnaissance
4. Verify legitimacy of network management systems generating SNMP traffic
5. Correlate SNMP activities with subsequent lateral movement attempts

Hypothesis 6: Windows Credential Dumping with Mimikatz

Mapping: T1003.001 - OS Credential Dumping: LSASS Memory **Hypothesis Explanation**: Salt Typhoon uses Mimikatz to dump credentials from LSASS memory on Windows systems, enabling lateral movement and privilege escalation within compromised environments. **Hunting Focus**: LSASS access patterns, Mimikatz execution indicators, and credential dumping activities

```
// Requires: DeviceEvents table
// Detects: LSASS memory access patterns indicative of credential dumping
DeviceEvents
| where Timestamp > ago(30d)
| where ActionType in ("OpenProcessApiCall", "ReadProcessMemoryApiCall",
"WriteToLsassProcessMemory")
| where FileName =~ "lsass.exe" or ProcessCommandLine contains "lsass"
| extend ProcessAccess = case(
    ActionType == "WriteToLsassProcessMemory", "LSASS Memory Write",
    ActionType == "ReadProcessMemoryApiCall", "LSASS Memory Read",
    ActionType == "OpenProcessApiCall", "Process Handle Open",
    "Other"
)
| summarize LsassAccess = count(), AccessTypes = make_set(ProcessAccess),
    FirstAccess = min(Timestamp), LastAccess = max(Timestamp) by
DeviceName, InitiatingProcessFileName, InitiatingProcessAccountName
| where LsassAccess > 5
| extend RiskScore = case(
    LsassAccess > 50, "Critical",
    LsassAccess > 20, "High",
    "Medium"
)
| take 500
```

Investigation Steps:

1. Identify processes accessing LSASS with suspicious access rights
2. Check for Mimikatz command-line arguments and execution patterns
3. Review security event logs for credential enumeration activities
4. Examine network connections from systems with credential dumping indicators
5. Correlate with privilege escalation attempts and lateral movement activities

Hypothesis 7: Protocol Tunneling and Covert C2

Mapping: T1572 - Protocol Tunneling, T1090.003 - Proxy: Multi-hop Proxy **Hypothesis Explanation**: Salt Typhoon establishes tunnels using GRE, mGRE, or IPsec protocols to create persistent and covert communication channels, blending with normal network traffic for command and control operations.

Hunting Focus: Unusual tunnel configurations, multi-hop proxy chains, and encrypted communication channels

```
// Requires: DeviceNetworkEvents table
// Detects: Protocol tunneling and covert C2 channels using GRE/IPsec
DeviceNetworkEvents
| where Timestamp > ago(30d)
| where (RemotePort in (500, 4500) and Protocol == "Udp")
    or InitiatingProcessFileName contains "stowaway"
| extend TunnelType = case(
    RemotePort == 500, "IPsec IKE",
    RemotePort == 4500, "IPsec NAT-T",
    InitiatingProcessFileName contains "stowaway", "Multi-hop Proxy",
    "Unknown"
)
| summarize TunnelConnections = count(), UniqueDestinations =
dcount(RemoteIP),
    FirstConnection = min(Timestamp), LastConnection = max(Timestamp) by
DeviceName, TunnelType, RemoteIP
| where TunnelConnections > 5 or UniqueDestinations > 3
| extend RiskScore = case(
    TunnelType == "Multi-hop Proxy", "Critical",
    TunnelConnections > 100, "High",
    UniqueDestinations > 10, "Medium",
    "Low"
)
| take 500
```

Investigation Steps:

1. Validate tunnel configurations against authorized VPN connections
2. Check for multi-hop proxy tools like STOWAWAY in the environment
3. Review tunnel traffic patterns for data exfiltration indicators
4. Examine tunnel endpoints for compromise indicators
5. Correlate tunnel establishment with other suspicious network activities

Hypothesis 8: PowerShell-based Living Off The Land Techniques

Mapping: T1059.001 - PowerShell, T1105 - Ingress Tool Transfer **Hypothesis Explanation**: Salt Typhoon leverages PowerShell and other living-off-the-land binaries (LOLBins) like BITSAdmin and CertUtil for reconnaissance, tool download, and execution while evading detection through legitimate system tools. **Hunting Focus**: Suspicious PowerShell execution, LOLBin abuse patterns, and encoded command execution

```
// Requires: DeviceProcessEvents table
// Detects: PowerShell and Living Off the Land binary abuse patterns
// Optimized: Uses has_any for encoded command detection, improved
filtering logic
DeviceProcessEvents
```

```
| where Timestamp > ago(30d)
| where (ProcessCommandLine has_any ("powershell", "-enc", "-
EncodedCommand"))
    or (ProcessCommandLine contains "bitsadmin" and ProcessCommandLine
contains "/transfer")
    or (ProcessCommandLine contains "certutil" and ProcessCommandLine
contains "-decode")
    or ProcessCommandLine contains "powercat"
| extend LOLBinType = case(
    ProcessCommandLine contains "powershell", "PowerShell",
    ProcessCommandLine contains "bitsadmin", "BITSAdmin",
    ProcessCommandLine contains "certutil", "CertUtil",
    ProcessCommandLine contains "powercat", "Powercat",
    "Other"
)
| extend EncodedCommand = case(
    ProcessCommandLine has_any ("-enc", "-EncodedCommand", "-e "), "True",
    "False"
)
| summarize CommandCount = count(),
    UniqueCommands = dcount(ProcessCommandLine),
    FirstExecution = min(Timestamp),
    LastExecution = max(Timestamp)
    by DeviceName, LOLBinType, EncodedCommand
| where CommandCount > 0
| extend RiskScore = case(
    EncodedCommand == "True", "High",
    UniqueCommands > 10, "Medium",
    "Low"
)
| order by RiskScore desc, CommandCount desc
| take 500
```

Investigation Steps:

1. Decode base64-encoded PowerShell commands for analysis
2. Review BITSAdmin transfer jobs and downloaded file destinations
3. Check CertUtil decode operations and resulting file outputs
4. Examine Powercat usage for network connections and file transfers
5. Correlate LOLBin activities with network communications and file system changes

5) Summary of Runbook

Hunt Hypothesis	MITRE TTP	KQL Query Focus	Detection Priority
CVE Exploitation Detection	T1190	Web exploitation patterns, Cisco IOS XE vulnerabilities	Critical
TACACS+ Traffic Interception	T1040, T1556	Authentication server redirection, packet capture	High

Hunt Hypothesis	MITRE TTP	KQL Query Focus	Detection Priority
Guest Shell Container Deployment	T1610	Container activation, Python script execution	High
Custom SFTP Client Usage	T1105, T1048.002	Malicious binary hashes, encrypted transfers	Critical
SNMP Reconnaissance	T1016, T1602.001	Bulk SNMP queries, configuration enumeration	Medium
Mimikatz Credential Dumping	T1003.001	LSASS access, credential extraction	Critical
Protocol Tunneling	T1572, T1090.003	GRE/IPsec tunnels, multi-hop proxies	High
PowerShell LOLBin Abuse	T1059.001, T1105	Encoded commands, tool downloads	**High**

Key Detection Metrics

- **Coverage**: 8 critical TTPs including 2025 evolutions with high-confidence detection logic
- **False Positive Rate**: Medium due to correlation-based queries and behavioral analysis
- **Response Time**: Automated alerting for critical findings
- **Investigation Depth**: Multi-stage verification procedures
- **2025 Enhancements**: Updated for DLL Side-Loading attacks and advanced LSASS memory targeting
- **Query Status**: All queries tested and validated against Microsoft Sentinel (2025-09-04)
- **Performance**: Optimized with row limits (max 500 results) for efficient execution

6) References

- https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-329a
- https://www.crowdstrike.com/blog/salt-typhoon-targets-us-telecommunications/
- https://attack.mitre.org/groups/G1015/

This document is prepared by Crimson7 - 2025 v1.0