INTRUVENT

COMPREHENSIVE THREAT HUNTING GUIDE

Volt Typhoon

Chinese State-Sponsored APT Targeting U.S. Critical Infrastructure

Version 1.0
Published December 7, 2025
Classification TLP:CLEAR
Hunt Duration 4-6 Hours

Intruvent Technologies
INT-THG-2025-VOLT-TYPHOON-v1.0

Table of Contents

Overview

Executive Summary 3
Threat Actor Intelligence 4
MITRE ATT&CK Mapping 5

Hunt Modules

Module 1: Network Edge Device Compromise 6
Module 2: Living-off-the-Land Detection 8
Module 3: Credential Harvesting 10
Module 4: Lateral Movement 12
Module 5: Port Proxy and Tunneling 14
Module 6: Defense Evasion 16
Module 7: Data Staging and Exfiltration 18
Module 8: OT/SCADA Pre-Positioning 20

Appendices

Appendix A: Indicators of Compromise 22
Appendix B: Detection Rules 25
Appendix C: References 28

1 Executive Summary

Critical Infrastructure Threat

- Volt Typhoon is a PRC state-sponsored threat actor targeting U.S. critical infrastructure
- Maintained persistent access in victim environments for up to 5 years
- Pre-positioning for potential destructive/disruptive attacks during conflict
- CISA, NSA, and FBI have issued multiple joint advisories (AA23-144A, AA24-038A)

Volt Typhoon (also known as VOLTZITE, Bronze Silhouette, Vanguard Panda, and Insidious Taurus) is a People's Republic of China (PRC) state-sponsored advanced persistent threat (APT) group that has been active since at least mid-2021. Unlike traditional cyber espionage operations focused on data theft, Volt Typhoon's primary objective is **pre-positioning within U.S. critical infrastructure** to enable potential disruption or destruction of services during a major geopolitical conflict.

Key Characteristics

Attribution	PRC State-Sponsored (High Confidence)
MITRE ATT&CK ID	G1017
Active Since	Mid-2021
Primary Targets	Communications, Energy, Water, Transportation, Maritime, Government
Geographic Focus	United States, Guam, U.S. Pacific Territories
Operational Style	Living-off-the-Land (LOTL), Hands-on-Keyboard
Dwell Time	Up to 5 years observed

Why This Hunt Matters

Volt Typhoon represents a strategic threat unlike typical financially-motivated or espionage-focused adversaries. Their demonstrated ability to maintain long-term access to OT/ICS environments—including exfiltrating SCADA diagrams and relay documentation—positions them to cause physical damage to infrastructure during a potential conflict scenario. Organizations in critical infrastructure sectors must proactively hunt for Volt Typhoon indicators.

2 Threat Actor Intelligence

Volt Typhoon HIGH CONFIDENCE

MOTIVATION
State-Sponsored / Pre-Positioning

FIRST OBSERVED
Mid-2021

SKILL LEVEL
Advanced

ALIASES
VOLTZITE, Bronze Silhouette, Vanguard Panda, DEV-0391, Insidious Taurus

CAMPAIGN
KV Botnet, Versa Director Zero-Day

TARGETS
U.S. Critical Infrastructure

Confirmed Victim Sectors

According to CISA and FBI advisories, Volt Typhoon has successfully compromised organizations in:

- **Communications:** Telecommunications providers, ISPs
- **Energy:** Electric utilities, power generation
- **Water:** Water treatment and distribution facilities
- **Transportation:** Aviation, rail, maritime
- **Government:** Federal, state, and local agencies

Recent Activity (2024-2025)

Date	Event	Details
Jan 2024	FBI Disruption	FBI disrupted KV Botnet operations, removed malware from SOHO routers
Feb 2024	CISA Advisory AA24-038A	Joint advisory confirming 5-year persistent access to victims
Jun-Aug 2024	Versa Director Zero-Day	CVE-2024-39717 exploitation targeting ISPs/MSPs
Nov 2024	Singtel Breach	Singapore telecom confirmed Volt Typhoon intrusion (eradicated)
Mar 2025	U.S. Utility Access	Maintained access to utility OT systems for nearly a year

Strategic Assessment

U.S. intelligence agencies assess that Volt Typhoon's targeting and behavior patterns indicate preparation for potential disruptive or destructive cyber operations against U.S. critical infrastructure in the event of a major crisis or conflict with the United States. This assessment is based on:

- Focus on OT/ICS asset documentation rather than traditional data theft
- Long dwell times without apparent exfiltration objectives
- Geographic concentration on Guam and Pacific-region infrastructure
- Targeting of sectors essential to military logistics and civilian resilience

3 MITRE ATT&CK Mapping

Volt Typhoon employs an extensive range of tactics and techniques, with particular emphasis on defense evasion through living-off-the-land binaries. The following mapping is based on MITRE ATT&CK Group G1017.

Primary Tactics

Tactic	Key Techniques	Hunt Priority
Initial Access	T1190 (Exploit Public-Facing App), T1133 (External Remote Services)	Critical
Execution	T1059.001 (PowerShell), T1059.003 (cmd), T1047 (WMI)	High
Persistence	T1078 (Valid Accounts), T1505.003 (Web Shell)	Critical
Defense Evasion	T1070 (Indicator Removal), T1036 (Masquerading), T1562 (Impair Defenses)	Critical
Credential Access	T1003.001 (LSASS), T1003.003 (NTDS), T1552 (Unsecured Creds)	Critical
Discovery	T1018 (Remote System), T1082 (System Info), T1083 (File Discovery)	High
Lateral Movement	T1021.001 (RDP), T1570 (Lateral Tool Transfer)	High
Collection	T1560 (Archive Data), T1005 (Local Data), T1056.001 (Keylogging)	Medium
C2	T1090 (Proxy), T1071 (App Layer Protocol), T1573 (Encrypted Channel)	High

Living-off-the-Land Tools

Volt Typhoon heavily relies on native Windows utilities to evade detection:

Tool	MITRE ID	Volt Typhoon Usage
`cmd.exe`	S0106	Primary command execution, discovery commands
`wmic.exe`	-	System enumeration, remote execution
`netsh.exe`	S0108	Port proxy configuration, firewall manipulation
`ntdsutil.exe`	-	NTDS.dit extraction for offline cracking
`certutil.exe`	S0160	File download, Base64 encoding/decoding
`wevtutil.exe`	S0645	Event log clearing and enumeration
`reg.exe`	S0075	Registry queries and modifications

4 Module 1: Network Edge Device Compromise

Duration: 45 minutes **Priority:** Critical **Data Sources:** Firewall logs, VPN logs, vulnerability scans

Hunt Objective

Identify evidence of exploitation or compromise of internet-facing network devices, which represent Volt Typhoon's primary initial access vector.

Background

Volt Typhoon has exploited multiple vulnerabilities in network edge devices from Fortinet, Ivanti, Cisco, NETGEAR, and Versa Networks. They also compromise SOHO routers to build botnets (KV Botnet) for proxying C2 traffic.

Exploited Vulnerabilities

CVE	Vendor/Product	CVSS	Type
CVE-2024-21762	Fortinet FortiOS SSL VPN	9.8	Out-of-Bounds Write
CVE-2024-21887	Ivanti Connect Secure	9.1	Command Injection
CVE-2023-46805	Ivanti Connect Secure	8.2	Auth Bypass
CVE-2024-39717	Versa Director	7.2	Arbitrary File Upload
CVE-2022-40684	Fortinet FortiOS	9.8	Auth Bypass

Hunt Steps

1.1 Inventory Vulnerable Devices

Identify all internet-facing devices running vulnerable firmware:

```
Vulnerability Management Query
# Check for Fortinet devices with vulnerable firmware
# FortiOS versions prior to 7.4.3, 7.2.7, 7.0.14, 6.4.15

# Check for Ivanti Connect Secure pre-patch versions
# Vulnerable: 9.x, 22.x before 22.4R2.3

# Check for Versa Director pre-22.1.4
# Vulnerable: Versa Director < 22.1.4
```

1.2 Check for Fortinet Compromise Indicators

Search for suspicious admin accounts created during CVE-2022-40684 exploitation:

```
Fortinet CLI
# Check for known malicious admin accounts
config system admin
    get | grep -E "fortinet-tech-support|fortigate-tech-support"
end

# Review all admin accounts created recently
diagnose sys top-level-admin
```

Indicator

Admin accounts named `fortinet-tech-support` or `fortigate-tech-support` are strong indicators of CVE-2022-40684 exploitation.

1.3 Review VPN Authentication Logs

Hunt for anomalous VPN connections that may indicate credential theft or exploitation:

```
KQL - Microsoft Sentinel
// Hunt for VPN logins from unusual locations
SigninLogs
| where TimeGenerated > ago(90d)
| where AppDisplayName contains "VPN"
| summarize
    LoginCount = count(),
    DistinctIPs = dcount(IPAddress),
    DistinctLocations = dcount(Location)
    by UserPrincipalName
| where DistinctLocations > 5
| order by DistinctLocations desc
```

1.4 Check for Web Shell Artifacts

Search for known Volt Typhoon web shell filenames:

```
File System Search (Windows/Linux)
# Windows - Search for known web shells
Get-ChildItem -Path C:\inetpub\wwwroot -Recurse -Include "AuditReport.jspx","iisstart.aspx" -ErrorAction SilentlyContinue

# Linux/Versa - Search for fy.sh web shell
find /var/versa -name "fy.sh" -o -name "*.jspx" 2>/dev/null
```

Expected Findings

- ☐ Unpatched edge devices with known Volt Typhoon-exploited CVEs
- ☐ Suspicious admin accounts on Fortinet devices
- ☐ VPN authentications from unexpected geographic locations
- ☐ Web shell files in web server directories
- ☐ Evidence of configuration changes outside maintenance windows

5 Module 2: Living-off-the-Land Detection

Duration: 45 minutes **Priority:** Critical **Data Sources:** EDR, Windows Event Logs, Command Line Auditing

Hunt Objective

Detect malicious use of legitimate Windows administrative tools (LOTL) that Volt Typhoon uses to evade detection.

Background

Volt Typhoon's signature technique is leveraging built-in Windows utilities for all phases of their operations. This makes detection challenging as these tools have legitimate administrative uses. Focus on context, command-line arguments, and execution chains.

Hunt Steps

2.1 Hunt for Suspicious WMIC Usage

Volt Typhoon uses WMIC for discovery and remote execution:

```
KQL - Microsoft Defender
DeviceProcessEvents
| where TimeGenerated > ago(30d)
| where FileName =~ "wmic.exe"
| where ProcessCommandLine has_any (
    "path win32_logicaldisk get",
    "process call create",
    "/node:",
    "shadowcopy",
    "os get"
)
| project TimeGenerated, DeviceName, AccountName, ProcessCommandLine, InitiatingProcessFileName
| order by TimeGenerated desc
```

2.2 Hunt for netsh Port Proxy Configuration

A key Volt Typhoon technique is establishing port proxies for traffic tunneling:

```
KQL - Windows Security Events
DeviceProcessEvents
| where TimeGenerated > ago(30d)
| where FileName =~ "netsh.exe"
| where ProcessCommandLine has_all ("interface", "portproxy")
| project TimeGenerated, DeviceName, AccountName, ProcessCommandLine
| order by TimeGenerated desc

// Also check registry for existing PortProxy configurations
DeviceRegistryEvents
| where RegistryKey contains @"SYSTEM\CurrentControlSet\Services\PortProxy"
| project TimeGenerated, DeviceName, RegistryKey, RegistryValueName, RegistryValueData
```

Detection Note

Any `netsh interface portproxy` commands should be investigated. Legitimate uses are rare outside of specific development scenarios.

2.3 Hunt for certutil Abuse

Volt Typhoon uses certutil for file downloads and Base64 encoding:

```
KQL
DeviceProcessEvents
| where TimeGenerated > ago(30d)
| where FileName =~ "certutil.exe"
| where ProcessCommandLine has_any (
    "-urlcache",
    "-decode",
    "-encode",
    "-f http",
    "-split"
)
| project TimeGenerated, DeviceName, AccountName, ProcessCommandLine
| order by TimeGenerated desc
```

2.4 Hunt for Discovery Command Sequences

Look for rapid sequences of discovery commands characteristic of hands-on-keyboard activity:

```
KQL
// Find systems with multiple discovery commands in short timeframes
DeviceProcessEvents
| where TimeGenerated > ago(7d)
| where FileName in~ ("ipconfig.exe", "systeminfo.exe", "net.exe",
    "whoami.exe", "tasklist.exe", "netstat.exe", "ping.exe", "nltest.exe")
| summarize
    CommandCount = count(),
    Commands = make_set(FileName),
    CommandLines = make_set(ProcessCommandLine)
    by DeviceName, AccountName, bin(TimeGenerated, 5m)
| where CommandCount >= 5
| order by TimeGenerated desc
```

Expected Findings

- ☐ WMIC commands targeting remote systems or sensitive data
- ☐ netsh portproxy configurations (tunneling indicators)
- ☐ certutil download or encoding operations
- ☐ Clusters of discovery commands from single accounts
- ☐ PowerShell with encoded commands or download cradles

6 Module 3: Credential Harvesting

Duration: 30 minutes **Priority:** Critical **Data Sources:** EDR, Security Events, File System

Hunt Objective

Identify attempts to extract credentials from LSASS, NTDS.dit, or stored credential caches.

Hunt Steps

3.1 Hunt for NTDS.dit Extraction

Volt Typhoon extracts the Active Directory database for offline password cracking:

```
KQL
// Hunt for ntdsutil usage
DeviceProcessEvents
| where TimeGenerated > ago(30d)
| where FileName =~ "ntdsutil.exe"
| where ProcessCommandLine has_any (
    "activate instance ntds",
    "create full",
    "ifm"
)
| project TimeGenerated, DeviceName, AccountName, ProcessCommandLine

// Hunt for volume shadow copy access to ntds.dit
DeviceProcessEvents
| where ProcessCommandLine has_all ("vssadmin", "create shadow") or
        ProcessCommandLine has_all ("wmic", "shadowcopy", "create")
```

3.2 Hunt for File Masquerading (.gif extension)

Volt Typhoon renames ntds.dit with .gif extension to evade detection:

```
KQL
// Look for large .gif files in temp directories
DeviceFileEvents
| where TimeGenerated > ago(30d)
| where FolderPath contains @"\Windows\Temp" or FolderPath contains @"\Users\Public"
| where FileName endswith ".gif"
| where FileSize > 10000000  // NTDS.dit typically > 10MB
| project TimeGenerated, DeviceName, FolderPath, FileName, FileSize
```

3.3 Hunt for Credential Store Access

Check for access to stored credentials (OpenSSH, PuTTY, RealVNC, Chrome):

```
KQL
DeviceFileEvents
| where TimeGenerated > ago(30d)
| where FolderPath has_any (
    @"\OpenSSH\",
    @"\.ssh\",
    @"\PuTTY\Sessions",
    @"\RealVNC\",
    @"\Chrome\User Data\Local State"
)
| project TimeGenerated, DeviceName, ActionType, FolderPath, FileName, InitiatingProcessFileName
```

3.4 Hunt for Keylogger Artifacts

Volt Typhoon has deployed keyloggers on domain controllers:

```
File Search
# Search for known keylogger artifact
Get-ChildItem -Path C:\ -Filter "rult3uil.log" -Recurse -ErrorAction SilentlyContinue

# Check for suspicious log files on DCs
Get-ChildItem -Path C:\Windows\Temp -Filter "*.log" |
    Where-Object { $_.Length -gt 1KB -and $_.LastWriteTime -gt (Get-Date).AddDays(-30) }
```

Expected Findings

- ☐ ntdsutil.exe execution with IFM parameters
- ☐ Volume shadow copy creation on domain controllers
- ☐ Large files with mismatched extensions (.gif containing database)
- ☐ Access to SSH keys, VNC credentials, or browser credential stores
- ☐ rult3uil.log keylogger artifacts

7 Module 4: Lateral Movement

Duration: 30 minutes **Priority:** High **Data Sources:** Windows Security Events, EDR, Network Logs

Hunt Objective

Detect lateral movement using RDP, WMI, PsExec, and Impacket tools.

Hunt Steps

4.1 Hunt for Impacket Wmiexec Patterns

Volt Typhoon uses Impacket's wmiexec for remote command execution:

```
KQL
// Detect Impacket wmiexec pattern
DeviceProcessEvents
| where TimeGenerated > ago(30d)
| where InitiatingProcessFileName =~ "wmiprvse.exe"
| where FileName =~ "cmd.exe"
| where ProcessCommandLine matches regex @"/Q\s+/c\s+.*\\\\127\.0\.0\.1\\ADMIN\$\\__\d+\.\d+"
| project TimeGenerated, DeviceName, AccountName, ProcessCommandLine
```

High Fidelity Indicator

The pattern `/Q /c * \\127.0.0.1\ADMIN$\__* 2>&1` is highly indicative of Impacket wmiexec and should trigger immediate investigation.

4.2 Hunt for Anomalous RDP Activity

Volt Typhoon moves to Domain Controllers via RDP:

```
KQL
// Hunt for RDP to sensitive systems (DCs, OT jump hosts)
DeviceNetworkEvents
| where TimeGenerated > ago(30d)
| where RemotePort == 3389
| where RemoteIP has_any ("DC", "domain", "scada", "hmi", "plc")  // Adjust for your naming
| summarize
    ConnectionCount = count(),
    SourceDevices = make_set(DeviceName)
    by RemoteIP, bin(TimeGenerated, 1d)
| order by ConnectionCount desc

// Also check Event ID 4624 Type 10 (Remote Interactive)
SecurityEvent
| where EventID == 4624
| where LogonType == 10
| where TargetUserName != "SYSTEM"
| project TimeGenerated, Computer, TargetUserName, IpAddress
```

4.3 Hunt for Service Account Abuse

Look for domain admin or service accounts used from unexpected systems:

```
KQL
// Identify accounts logging into many systems (potential lateral movement)
SecurityEvent
| where TimeGenerated > ago(7d)
| where EventID == 4624
| where LogonType in (3, 10)  // Network and RDP logons
| summarize
    SystemCount = dcount(Computer),
    Systems = make_set(Computer, 20)
    by TargetUserName
| where SystemCount > 10
| order by SystemCount desc
```

Expected Findings

- ☐ Impacket wmiexec command patterns in process logs
- ☐ RDP connections to domain controllers from workstations
- ☐ Single accounts authenticating to many systems
- ☐ WMI-based remote process creation

8 Module 5: Port Proxy and Tunneling

Duration: 30 minutes **Priority:** High **Data Sources:** Registry, EDR, Network Flow

Hunt Objective

Detect Fast Reverse Proxy (FRP) usage and netsh port proxy configurations used for C2 tunneling.

Hunt Steps

5.1 Check for Active Port Proxy Configurations

Query current port proxy settings across systems:

```
PowerShell (run across systems)
# Check current port proxy configuration
netsh interface portproxy show all

# Check registry for port proxy entries
Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\PortProxy\v4tov4\tcp" -ErrorAction SilentlyContinue
```

5.2 Hunt for FRP Binaries by Hash

Search for known Volt Typhoon FRP binary hashes:

```
KQL
// Hunt for known FRP hashes (sample from Unit 42)
let FRPHashes = dynamic([
    "baeffeb5fdef2f42a752c65c2d2a52e84fb57efc906d981f89dd518c314e231c",
    "b4f7c5e3f14fb57be8b5f020377b993618b6e3532a4e1eb1eae9976d4130cc74",
    "4b0c4170601d6e922cf23b1caf096bba2fade3dfcf92f0ab895a5f0b9a310349"
]);
DeviceFileEvents
| where SHA256 in (FRPHashes)
| project TimeGenerated, DeviceName, FileName, FolderPath, SHA256
```

5.3 Hunt for Suspicious Tunneling Processes

Look for processes with tunnel-like behavior:

```
KQL
// Hunt for tunnel-related process names and command lines
DeviceProcessEvents
| where TimeGenerated > ago(30d)
| where ProcessCommandLine has_any (
    "frpc",
    "frps",
    "proxy",
    "tunnel",
    "portfwd",
    "ngrok"
) or FileName in~ ("frpc.exe", "frps.exe", "plink.exe")
| project TimeGenerated, DeviceName, FileName, ProcessCommandLine, AccountName
```

Expected Findings

- ☐ Active netsh portproxy configurations
- ☐ Registry keys under Services\PortProxy
- ☐ FRP binaries matching known hashes
- ☐ Processes establishing persistent outbound connections

9 Module 6: Defense Evasion

Duration: 30 minutes **Priority:** High **Data Sources:** Windows Events, EDR, File System

Hunt Objective

Detect log clearing, process masquerading, and security tool tampering.

Hunt Steps

6.1 Hunt for Event Log Clearing

Volt Typhoon selectively clears Windows Event Logs:

```
KQL
// Detect wevtutil clear-log commands
DeviceProcessEvents
| where TimeGenerated > ago(30d)
| where FileName =~ "wevtutil.exe"
| where ProcessCommandLine has_any ("cl", "clear-log", "clear")
| project TimeGenerated, DeviceName, AccountName, ProcessCommandLine

// Also check for Event ID 1102 (Security log cleared)
SecurityEvent
| where EventID == 1102
| project TimeGenerated, Computer, Account
```

6.2 Hunt for Process Masquerading

Look for known masqueraded process names used by Volt Typhoon:

```
KQL
// Known Volt Typhoon masqueraded process names
let SuspiciousNames = dynamic([
    "cisco_up.exe", "cl64.exe", "vm3dservice.exe",
    "watchdogd.exe", "Win.exe", "WmiPreSV.exe"
]);
DeviceProcessEvents
| where TimeGenerated > ago(30d)
| where FileName in~ (SuspiciousNames)
| project TimeGenerated, DeviceName, FileName, FolderPath, ProcessCommandLine

// Detect WmiPrvSE.exe typosquatting
DeviceProcessEvents
| where FileName =~ "WmiPreSV.exe"  // Note: Missing 'r' - typosquat
| project TimeGenerated, DeviceName, FolderPath, ProcessCommandLine
```

6.3 Hunt for Security Tool Tampering

Volt Typhoon disables security tools and firewalls:

```
KQL
// Detect netsh firewall disabling
DeviceProcessEvents
| where FileName =~ "netsh.exe"
| where ProcessCommandLine has_any (
    "firewall set opmode disable",
    "advfirewall set allprofiles state off"
)
| project TimeGenerated, DeviceName, AccountName, ProcessCommandLine

// Detect service stopping (Windows Defender, AV services)
DeviceProcessEvents
| where FileName in~ ("sc.exe", "net.exe")
| where ProcessCommandLine has_any ("stop", "delete")
    and ProcessCommandLine has_any ("windefend", "defender", "antivirus", "security")
| project TimeGenerated, DeviceName, AccountName, ProcessCommandLine
```

Expected Findings

- ☐ wevtutil log clearing commands
- ☐ Event ID 1102 (Security log cleared)
- ☐ Processes with masqueraded names from unexpected paths
- ☐ Firewall or security service tampering

10 Module 7: Data Staging and Exfiltration

Duration: 30 minutes **Priority:** Medium **Data Sources:** File System, EDR, Network Logs

Hunt Objective

Identify data staging in temporary directories and archive creation for exfiltration.

Hunt Steps

7.1 Hunt for Password-Protected Archives

Volt Typhoon creates password-protected 7-Zip archives for exfiltration:

```
KQL
// Hunt for 7-Zip with password parameter
DeviceProcessEvents
| where TimeGenerated > ago(30d)
| where FileName in~ ("7z.exe", "7za.exe")
| where ProcessCommandLine has "-p"
| project TimeGenerated, DeviceName, AccountName, ProcessCommandLine

// Hunt for archive creation in temp directories
DeviceFileEvents
| where TimeGenerated > ago(30d)
| where FolderPath has_any (@"\Windows\Temp", @"\Users\Public", @"\ProgramData")
| where FileName endswith_any (".7z", ".zip", ".rar")
| where FileSize > 1000000  // > 1MB
| project TimeGenerated, DeviceName, FolderPath, FileName, FileSize
```

7.2 Hunt for Staging Locations

Check known staging locations for sensitive files:

```
PowerShell
# Check for ntds.dit staging
Get-ChildItem -Path "C:\Windows\Temp" -Filter "ntds*" -Recurse -ErrorAction SilentlyContinue

# Check for systeminfo.dat (known staging file)
Get-ChildItem -Path "C:\Users\Public\Documents" -Filter "systeminfo.dat" -ErrorAction SilentlyContinue

# Check for large files in temp directories
Get-ChildItem -Path "C:\Windows\Temp", "C:\Users\Public" -Recurse |
    Where-Object { $_.Length -gt 10MB } |
    Select-Object FullName, Length, LastWriteTime
```

Expected Findings

- ☐ Password-protected archive creation commands
- ☐ Large archive files in temp directories
- ☐ NTDS.dit copies outside normal backup locations
- ☐ systeminfo.dat or similar staging files

11 Module 8: OT/SCADA Pre-Positioning

Duration: 30 minutes **Priority:** Critical (for CI organizations) **Data Sources:** File System, Network Segmentation Logs, Jump Host Logs

Hunt Objective

Detect reconnaissance and access attempts targeting OT/ICS systems and documentation.

Critical Infrastructure Alert

Volt Typhoon's primary objective is pre-positioning in OT environments. CISA has confirmed exfiltration of SCADA diagrams and relay documentation from energy and water utilities. This module is essential for organizations with OT/ICS systems.

Hunt Steps

8.1 Hunt for OT Documentation Access

Look for access to OT-related documentation and diagrams:

```
KQL
// Hunt for file access to OT-related content
DeviceFileEvents
| where TimeGenerated > ago(90d)
| where FileName has_any (
    "scada", "hmi", "plc", "ics", "dcs",
    "relay", "substation", "wiring", "diagram",
    "p&id", "pid", "network diagram"
)
| where ActionType in ("FileRead", "FileAccessed", "FileCopied")
| project TimeGenerated, DeviceName, ActionType, FileName, FolderPath, InitiatingProcessAccountName

// Hunt for access to engineering file types
DeviceFileEvents
| where FileName endswith_any (
    ".dwg", ".dxf",  // CAD files
    ".vsd", ".vsdx", // Visio diagrams
    ".pdf"           // Documentation
)
| where FolderPath has_any ("engineering", "scada", "ics", "ot", "operations")
| project TimeGenerated, DeviceName, ActionType, FileName, FolderPath
```

8.2 Hunt for IT/OT Boundary Violations

Check for unauthorized access attempts from IT to OT networks:

```
Firewall/Network Log Query
// Example: Palo Alto Networks firewall
// Look for allowed or denied connections from IT to OT zones

// Check jump host authentication logs
SecurityEvent
| where Computer has_any ("jumphost", "bastion", "ot-access")
| where EventID == 4624
| summarize
    LogonCount = count(),
    DistinctUsers = dcount(TargetUserName)
    by SourceIP = IpAddress, bin(TimeGenerated, 1d)
| where LogonCount > 10
| order by LogonCount desc
```

8.3 Check for PuTTY Session Harvesting

Volt Typhoon targets PuTTY sessions for OT device access:

```
KQL / PowerShell
// Hunt for PuTTY registry access
DeviceRegistryEvents
| where RegistryKey contains @"\SimonTatham\PuTTY\Sessions"
| project TimeGenerated, DeviceName, RegistryKey, ActionType, InitiatingProcessFileName

// PowerShell: Check for saved PuTTY sessions
Get-ChildItem -Path "HKCU:\Software\SimonTatham\PuTTY\Sessions" -ErrorAction SilentlyContinue |
    ForEach-Object { Get-ItemProperty $_.PSPath }
```

Expected Findings

- ☐ Access to SCADA diagrams, relay documentation, P&IDs
- ☐ Unauthorized connections from IT systems to OT networks
- ☐ Abnormal jump host authentication patterns
- ☐ PuTTY session registry access or harvesting
- ☐ Enumeration of OT-related network shares

A Appendix: Indicators of Compromise

IOC Sourcing

Indicators sourced from CISA AA24-038A, Unit 42 Palo Alto Networks, Microsoft Threat Intelligence, and MITRE ATT&CK. IOC currency verified: December 7, 2025.

File Hashes - FRP (Fast Reverse Proxy) Binaries

SHA256 hashes of malicious FRP binaries used by Volt Typhoon (Source: Unit 42):

```
SHA256 Hashes
baeffeb5fdef2f42a752c65c2d2a52e84fb57efc906d981f89dd518c314e231c
b4f7c5e3f14fb57be8b5f020377b993618b6e3532a4e1eb1eae9976d4130cc74
4b0c4170601d6e922cf23b1caf096bba2fade3dfcf92f0ab895a5f0b9a310349
c0fc29a52ec3202f71f6378d9f7f9a8a3a10eb19acb8765152d758aded98c76d
d6ab36cb58c6c8c3527e788fc9239d8dcc97468b6999cf9ccd8a815c8b4a80af
9dd101caee49c692e5df193b236f8d52a07a2030eed9bd858ed3aaccb406401a
450437d49a7e5530c6fb04df2e56c3ab1553ada3712fab02bd1eeb1f1adbc267
93ce3b6d2a18829c0212542751b309dacbdc8c1d950611efe2319aa715f3a066
```

File Hashes - Malicious Binaries

```
SHA256 Hashes
f4dd44bc19c19056794d29151a5b1bb76afd502388622e24c863a8494af147dd
ef09b8ff86c276e9b475a6ae6b54f08ed77e09e169f7fc0872eb1d427ee27d31
d6ebde42457fe4b2a927ce53fc36f465f0000da931cfab9b79a36083e914ceca
472ccfb865c81704562ea95870f60c08ef00bcd2ca1d7f09352398c05be5d05d
66a19f7d2547a8a85cee7a62d0b6114fd31afdee090bd43f36b89470238393d7
```

File Artifacts

Indicator	Type	Context
`AuditReport.jspx`	Web Shell	Known Volt Typhoon web shell filename
`iisstart.aspx`	Web Shell	Web shell masquerading as IIS default page
`fy.sh`	Web Shell	Discovered on compromised Cisco routers
`rult3uil.log`	Keylogger	Keylogger output file on domain controllers
`*.gif (large files)`	Masquerading	NTDS.dit renamed with .gif extension

Masqueraded Process Names

Process Name	Context
`cisco_up.exe`	Masquerading as Cisco update
`cl64.exe`	Generic masquerading
`vm3dservice.exe`	Masquerading as VMware component
`watchdogd.exe`	Masquerading as watchdog service
`Win.exe`	Generic Windows masquerading
`WmiPreSV.exe`	Typosquat of WmiPrvSE.exe

Fortinet Compromise Indicators

Indicator	Context
`fortinet-tech-support`	Admin account created via CVE-2022-40684
`fortigate-tech-support`	Admin account created via CVE-2022-40684

Registry Keys

Registry Path	Context
`HKLM\SYSTEM\CurrentControlSet\Services\PortProxy\v4tov4\tcp`	netsh port proxy persistence

B Appendix: Detection Rules

Rule Development Note

The following detection rules are custom-developed by Intruvent Technologies based on documented behavioral indicators from CISA, Microsoft, and Unit 42 threat intelligence. Rule IDs are unique to this publication and not registered with SigmaHQ. All Sigma rules are marked as `experimental` status and should be tested thoroughly in your environment before production deployment.

Sigma Rule - netsh Port Proxy

```
Sigma
title: Volt Typhoon - netsh Port Proxy Configuration
id: volt-typhoon-portproxy-001
status: experimental
description: Detects netsh portproxy commands used by Volt Typhoon for tunneling
author: Intruvent Technologies
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        Image|endswith: '\netsh.exe'
        CommandLine|contains|all:
            - 'interface'
            - 'portproxy'
    condition: selection
falsepositives:
    - Legitimate network configuration by administrators
level: high
tags:
    - attack.command_and_control
    - attack.t1090
    - attack.g1017
```

Sigma Rule - Impacket Wmiexec

```
Sigma
title: Volt Typhoon - Impacket Wmiexec Pattern
id: volt-typhoon-impacket-001
status: experimental
description: Detects Impacket wmiexec execution pattern used by Volt Typhoon
author: Intruvent Technologies
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\wmiprvse.exe'
        Image|endswith: '\cmd.exe'
        CommandLine|contains|all:
            - '/Q'
            - '/c'
            - '\\127.0.0.1\ADMIN$\__'
    condition: selection
falsepositives:
    - Very low - this pattern is highly specific to Impacket
level: critical
tags:
    - attack.execution
    - attack.lateral_movement
    - attack.t1047
    - attack.g1017
```

Sigma Rule - NTDS.dit Extraction

```
Sigma
title: Volt Typhoon - NTDS.dit Extraction via ntdsutil
id: volt-typhoon-ntds-001
status: experimental
description: Detects NTDS.dit extraction attempts characteristic of Volt Typhoon
author: Intruvent Technologies
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        Image|endswith: '\ntdsutil.exe'
        CommandLine|contains:
            - 'activate instance ntds'
            - 'create full'
            - 'ifm'
    condition: selection
falsepositives:
    - Legitimate Active Directory backup operations
level: high
tags:
    - attack.credential_access
    - attack.t1003.003
    - attack.g1017
```

Palo Alto Networks Threat Prevention Signatures

The following Threat Prevention signature IDs detect Volt Typhoon activity:

Signature ID	Description
91676	Volt Typhoon related detection
92734	Volt Typhoon related detection
91362	Volt Typhoon related detection
90829	Volt Typhoon related detection
91363	Volt Typhoon related detection
86360	Volt Typhoon related detection

C Appendix: References

[1] MITRE ATT&CK - Volt Typhoon (G1017)
https://attack.mitre.org/groups/G1017/

[2] CISA Advisory AA24-038A - PRC State-Sponsored Actors Compromise and Maintain Persistent Access to U.S. Critical Infrastructure (February 2024)
https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-038a

[3] CISA Advisory AA23-144A - PRC State-Sponsored Cyber Actor Living off the Land to Evade Detection (May 2023)
https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-144a

[4] Microsoft Threat Intelligence - Volt Typhoon targets US critical infrastructure with living-off-the-land techniques
https://www.microsoft.com/en-us/security/blog/2023/05/24/volt-typhoon-targets-us-critical-infrastructure-with-living-off-the-land-techniques/

[5] Unit 42 Palo Alto Networks - Threat Brief: Attacks on Critical Infrastructure Attributed to Insidious Taurus (Volt Typhoon)
https://unit42.paloaltonetworks.com/volt-typhoon-threat-brief/

[6] Intel 471 - Threat Hunting Case Study: Looking for Volt Typhoon
https://www.intel471.com/blog/threat-hunting-case-study-looking-for-volt-typhoon

[7] Versa Networks Security Bulletin - Volt Typhoon Exploitation of N-Day and Zero-Day Vulnerabilities
https://versa-networks.com/blog/versa-security-bulletin-volt-typhoon-exploitation-of-n-day-and-zero-day-vulnerabilities/

[8] Fortinet FortiGuard Labs - Volt Typhoon Threat Actor
https://fortiguard.fortinet.com/threat-actor/5564/volt-typhoon

[9] Lumen Technologies - CVE-2024-39717 Versa Director Zero-Day Discovery
https://www.lumen.com/en-us/about/newsroom.html

[10] NCERT Philippines - Understanding Volt Typhoon: Key IOCs and TTPs
https://www.ncert.gov.ph/2024/11/13/understanding-volt-typhoon-key-iocs-and-ttps/