ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 01 / 30 · INTRUSION WORKFLOWS

# Staged Domain and Hypervisor Intrusion

## Database Pivots and Selective Rollback

What 364 files on one exposed staging server show about a web-to-domain-to-ESXi workflow, and what its rollback script would hide

Two-week observation window ending 18 November 2026 · Prepared November 2026
Full Report · Public edition · Version 1.1

**364**
Staged scripts and outputs on the server (311 in the root folder, 52 in a project folder, 1 in /loot)

**62.1%**
Share of Python files (226 of 364)

**96 / 17**
At least 96 scripts (26.4%) reference 17 internal RFC1918 hosts across 3 subnets

**1,458**
Files across four comparable staging toolkits (364 here plus 1,094 in three peers)

# **What Is in This Report.** 11 chapters and two appendices.

What 364 files on one exposed staging server show about a web-to-domain-to-ESXi workflow, and what its rollback script would hide


| | | |
|---|---|---|
| 01 | Executive Summary | 04 |
| 02 | The Staging Server and Its Toolkit | 07 |
| 03 | Perimeter Entry and HTTP Tunneling (Stages 0 to 2) | 10 |
| 04 | Database Pivots, Parallel Tracks and Domain Replication (Stages 3 to 5) | 14 |
| 05 | Hypervisor Probing, Dead Ends and Selective Rollback (Stages 6 to 8) | 18 |
| 06 | Peer Toolkits and Alternative Explanations | 21 |
| 07 | What Defenders Should Do | 24 |
| 08 | Detection Rules | 27 |
| 09 | Indicators | 31 |
| 10 | Scope and Limits | 33 |
| 11 | Sources | 35 |
| A | Indicators of compromise (29) | 37 |
| B | Detection rules (5 files) | 39 |


IN SHORT

Odin by Brandefense found an exposed staging server holding 364 attack files. The files were first seen before the window. The files stage a path from a web shell through SQL Server and Oracle databases to Active Directory replication and VMware ESXi probing. A rollback script would disable SQL command execution and delete two temporary admins, but leaves a hidden local admin. Alert on configuration changes and short-lived admin accounts, not only on current state.

01

CHAPTER 01

# Executive Summary

IN SHORT

Odin by Brandefense found an exposed staging server holding 364 attack files. The files were first seen before the window. The files stage a path from a web shell through SQL Server and Oracle databases to Active Directory replication and VMware ESXi probing. A rollback script would disable SQL command execution and delete two temporary admins, but leaves a hidden local admin. Alert on configuration changes and short-lived admin accounts, not only on current state.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

# 01 Executive Summary

The server sat open on the internet with no authentication. It held the operator's scripts, batch files and one captured database output. The files show how the operator planned to move from a perimeter web application to the domain controller and the hypervisor layer. We hold no logs from the targeted network, so every step below is staged capability unless a captured output says otherwise.

1. **Python on the staging server did the heavy work.**
   Python made up 226 of the 364 files (62.1%). These scripts were written to run from the staging server and handle padding-oracle math, SOCKS5 relays, SSH automation, offline hash cracking and raw LDAP checks. Target-side work used built-in tools instead: 87 PowerShell files (23.9%), 25 shell scripts (6.9%) and 19 Oracle PL/SQL scripts (5.2%).

2. **Staged database and Linux paths aim to bridge the DMZ and Active Directory.**
   The core workflow targets internal SQL Server instances and an RPC-enabled linked server. Parallel scripts probe an Oracle GIS database, Linux hosts over SSH and an internal web app. At least 96 scripts (26.4%) reference 17 internal RFC1918 addresses across three private subnets, plus one swept /24 range.

3. **The rollback script cleans what posture scans check, and nothing else.**
   As written, a PowerShell rollback script disables `xp_cmdshell` and deletes two temporary administrator accounts. It leaves out a local administrator hidden from the Windows sign-in screen, and it leaves web-root staging fragments in place.

## Three Decisions for Leaders

1. **Alert on configuration changes, not only on compliance state.** Send SQL Server configuration-change events and short-lived admin account lifecycles to your SIEM. A rolled-back escalation then still triggers an investigation.
2. **Separate hypervisor and out-of-band management from database networks.** Put ESXi, vCenter, HP iLO and storage switches on their own management segment. Use credentials outside Active Directory and restrict port 427.
3. **Treat database linked-server logins as high-privilege identity trusts.** Review SQL Server linked-login mappings with RPC-out enabled, and Oracle Java permissions, on the same schedule as Active Directory domain trusts.

## The Study in Numbers

| Number | What it is |
|---|---|
| **364** | Staged scripts and outputs on the server (311 in the root folder, 52 in a project folder, 1 in `/loot`) |
| **About 2 days** | Span over which the files were first seen |
| **62.1%** | Share of Python files (226 of 364) |
| **96 / 17** | At least 96 scripts (26.4%) reference 17 internal RFC1918 hosts across 3 subnets |
| **1,458** | Files across four comparable staging toolkits (364 here plus 1,094 in three peers) |
| **29** | Representative files with SHA-256 hashes for hunting, across Stages 0 to 8 |

**How to use this report.**

Security leaders can read Chapter 1 and the attribution limits in Chapter 6. SOC and detection engineers should start with Chapters 3 to 5 and the rules in Chapter 8. Database and VMware administrators can use the checklist in Chapter 7.

02

CHAPTER 02

# The Staging Server and Its Toolkit

IN SHORT

The 364 files sit on one external staging server in three locations. Most quick experiments live in a flat root folder, while a curated project folder holds the staged domain workflow.

INSIDE THIS CHAPTER

Where the files sit 07

How to read staged files 07

Language split 07

# 02 The Staging Server and Its Toolkit

## Where the files sit

The staging server's address is withheld, because an authorized red team cannot be ruled out. The operator named the project folder after the target organisation and the date 11 September 2026. Its numbered subfolders carry Chinese names.

| Location | Files | Share | What it holds |
|---|---|---|---|
| Root folder | 311 | 85.4% | Quick-iteration Telerik lab, Oracle PL/SQL, Linux SSH and web file-read scripts |
| `/loot` | 1 | 0.3% | `q12.out`, a captured Oracle Java compilation and error output |
| Project folder | 52 | 14.3% | 48 files in seven numbered subfolders, 3 batch orchestrators, and 1 extra `dcsmb.py` revision |

The project subfolders are numbered 01, 02 and 04 to 08. Folder 03 is absent. The names describe each stage.

| Folder | Plain meaning |
|---|---|
| 01 | Entry: web shell on the target's GIS web application |
| `02_隧道_rtun2` | Tunnel: the `rtun2` HTTP-to-TCP relay |
| 04 | Domain identity: domain administrator escalation |
| 05 | DCSync and the domain controller |
| `06_横向扩展` | Lateral expansion: AD, ESXi and management networks |
| `07_前期跳板_101_8_55` | Early jump host 10.101.8.55 |
| `08_未能奏效的尝试` | Failed attempts |

## How to read staged files

Three limits apply to everything that follows.

- **Files, not activity.** The files show intent and tooling. They do not show when, or whether, anything ran on a target.
- **Some counts are lower bounds.** Working exploit code and plaintext secrets are left out of this report. Counts such as the 96 scripts with internal addresses, or the 21 files with syntax or compilation errors, rest on file descriptions rather than full code.
- **Only one file records target contact.** `/loot/q12.out` holds an Oracle `ORA-29535` error and a pre-Java 7 `ProcessBuilder.redirectOutput(File)` compilation error. Everything else is staged capability.

## Language split

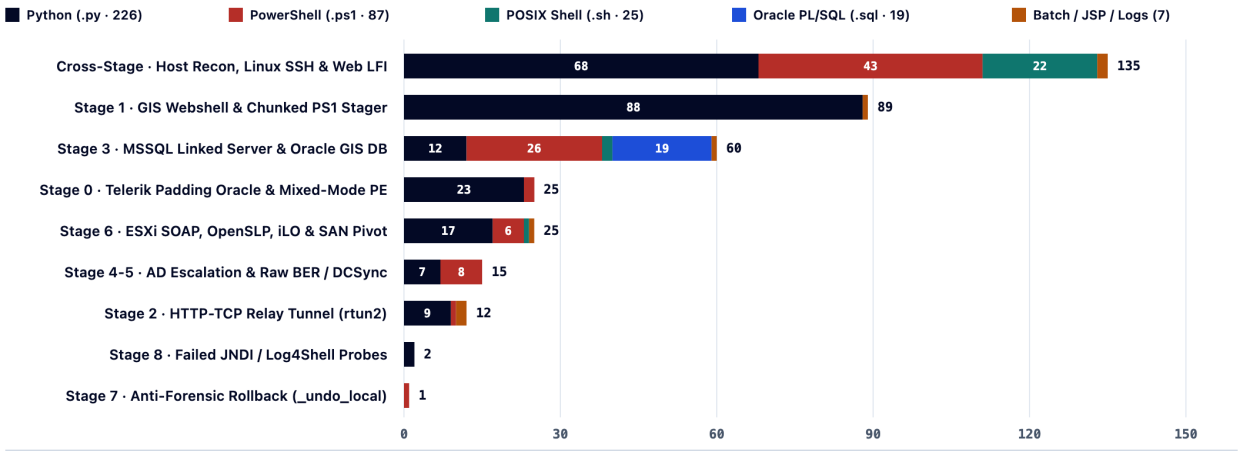
The operator kept the logic on the staging server rather than dropping compiled implants or Python runtimes on Windows targets. External Python handled cryptography, SOCKS5 multiplexing, PuTTY `plink` SSH relays and `MS-DRSR` directory replication. Target-side PowerShell called built-in components: `System.Data.SqlClient`, `certutil.exe`, `Advapi32.dll` and `cmd.exe`.

FIGURE 1 · TOOLKIT ARCHITECTURE & LANGUAGE SPLIT

ODIN BY BRANDEFENSE · TLP:CLEAR

**Python Drives Cryptography & Relays; PowerShell & Oracle PL/SQL Drive DB Pivots (n = 364)**

Artifact count by functional kill-chain group and runtime language across the dated project folder and staging root




**Scope & source:** 364 staged operational files grouped by folder and file type (226 .py, 87 .ps1, 25 .sh, 19 Oracle .sql, 7 other).
**What this does not show:** File counts measure iterative operator script revisions staged over about two days, not lines of code or runtime duration.

Figure 1: Toolkit architecture and language split

| Language | Files | Share | Mean / median size (bytes) | Size range (bytes) | Main role |
|---|---|---|---|---|---|
| Python ( `.py` ) | 226 | 62.1% | 2,733 / 1,090 | 334 to 37,822 | Crypto oracles, SOCKS and SSH relays, raw LDAP, DCSync |
| PowerShell ( `.ps1` ) | 87 | 23.9% | 1,841 / 1,394 | 169 to 7,334 | MSSQL jump-host wrappers, AD user staging, rollback |
| Shell ( `.sh` ) | 25 | 6.9% | 2,578 / 1,519 | 99 to 10,014 | Linux SSH privilege escalation, NFSv3/v4 droppers, file-read collection |
| Oracle PL/SQL ( `.sql` ) | 19 | 5.2% | 1,743 / 1,372 | 124 to 4,775 | `SQL*Plus` Java procedures, `DBMS_SCHEDULER` , `UTL_FILE` probes |
| Other ( `.bat` , `.txt` , `.jsp` , `.out` , `.log` ) | 7 | 1.9% | 1,698 / 1,219 | 29 to 3,977 | Batch orchestrators, JSP command web shells, captured Oracle output |

CHAPTER 03

# Perimeter Entry and HTTP Tunneling (Stages 0 to 2)

IN SHORT

The workflow targets a Windows GIS web server on a non-standard port. Scripts stage a web shell, chunked file delivery, and an HTTP-to-TCP relay that gives the operator SOCKS5 and port-forward access inside.

INSIDE THIS CHAPTER

A local Telerik lab 10
Chunked delivery and web shell hygiene 10
The HTTP-to-TCP relay 10
Key files, Stages 0 to 2 11

# 03 Perimeter Entry and HTTP Tunneling (Stages 0 to 2)

The perimeter address, port and application path of the targeted web server are withheld.

## A local Telerik lab

The root folder holds 38 files (`#001` to `#038`) outside the project folders. Of these, 25 (23 Python, 2 PowerShell) are Stage 0 padding-oracle and assembly-patching scripts. The other 13 are local lab files.

The script headers cite Progress Telerik UI for ASP.NET AJAX advisories CVE-2026-13181 to CVE-2026-13184 [1]. The vendor lists versions 2010.1.309 to 2026.2.519 as affected and 2026.2.708 as fixed. The scripts point at a local IIS lab (`http://127.0.0.1:8090`, `C:\TelerikLab`) and build on the older CVE-2019-18935 [2]. In the suite, `build_mixedmode.py` builds a mixed-mode .NET assembly that can load native code.

None of these 38 scripts names the perimeter target. The project's entry folder and Tomcat helpers (`axis/u.jsp`, `tjsp.jsp`) do target a live ASP.NET and Tomcat stack. The lab may have been used to test a payload before delivery, or it may be separate research on a shared server.

## Chunked delivery and web shell hygiene

Large HTTP POST bodies and inline PowerShell often trip web application firewalls. The operator staged three workarounds.

1. **Base64 chunks.** Python uploaders slice a `.ps1` script into small Base64 fragments prefixed `_pushp`. They write them into the web application folder or `C:\Windows\Temp`.
2. **GET fallback and markers.** `up.py` delivers fragments over HTTP GET when POST connections reset, and reassembles them with PowerShell `Set-Content`. JSP shells return `GBK` output prefixed `JSPOK|`. Background PowerShell launchers poll for `PSBG_DONE`.
3. **Competing web shell check.** `_find_local.ps1` scans `C:\app` and `C:\inetpub` for other web shell names such as `u.jsp`, `tun`, `shell`, `behind`, `godzilla` and `.war`.

## The HTTP-to-TCP relay

The operator staged the `rtun2` relay with `axis/u.jsp` and `tung2.jsp`, plus `GBK+CRLF` batch generators. This design fits a perimeter firewall that blocks direct inbound connections to internal subnets. It binds three loopback listeners on the staging server.

| Listener | Purpose |
|---|---|
| `127.0.0.1:1080` (also 11080 in `reuse_mssql.py`) | SOCKS5 proxy for `smbsweep.py`, `shares.py`, `depothunt.py`, `esxiauth.py` and `dcsync.py` |
| `127.0.0.1:10389` | Forward to domain controller 10.101.8.202 port 389 (LDAP) |
| `127.0.0.1:10445` | Forward to domain controller 10.101.8.202 port 445 (SMB) |

## Key files, Stages 0 to 2

| File | ID | Size (bytes) | Role |
|---|---|---|---|
| `telerik_oracle_attack.py` | `#017` | 10,284 | Tests the CVE-2026-13182 `RadAsyncUpload` padding and JSON-parse error difference |
| `build_mixedmode.py` | `#026` | 37,822 | Builds a mixed-mode .NET assembly that loads native code |
| `dl.py` | `#039` | 1,927 | Serves Base64 payloads over HTTP (:8081) for `certutil` retrieval |
| `psrun.py` | `#094` | 2,619 | Slices PowerShell into `_pushp` chunks for the ASP.NET web shell |
| `B_all.bat` | `#096` | 497 | Top-level wrapper that calls `RUN_ALL.bat` |
| `gx.py` | `#098` | 1,252 | Web shell command client with retries (later 2,015-byte revision `#196`) |
| `sy.py` | `#108` | 613 | Relays commands to `axis/u.jsp` and converts `GBK` output to UTF-8 |
| `RUN_ALL.bat` | `#122` | 2,445 | `GBK+CRLF` pipeline chaining tunnel setup, SQL escalation and DCSync |
| `_find_local.ps1` | `#146` | 460 | Looks for competing web shells |
| `tjsp.jsp` | `#169` | 563 | Minimal JSP `cmd.exe /c` shell returning output prefixed `JSPOK\ \|` |
| `hx.py` | `#172` | 2,020 | Main HTTP client for the perimeter web shell |
| `up.py` | `#198` | 2,568 | HTTP GET fragment uploader |
| `psbg.py` | `#269` | 4,833 | Background PowerShell launcher polling for `PSBG_DONE` |
| `_gen_runall.py` | `#271` | 3,934 | Generates an updated `RUN_ALL.bat` |
| `tunkick2.py` | `#314` | 1,462 | Redeploys the Tomcat relay JSPs to restore SOCKS workers |

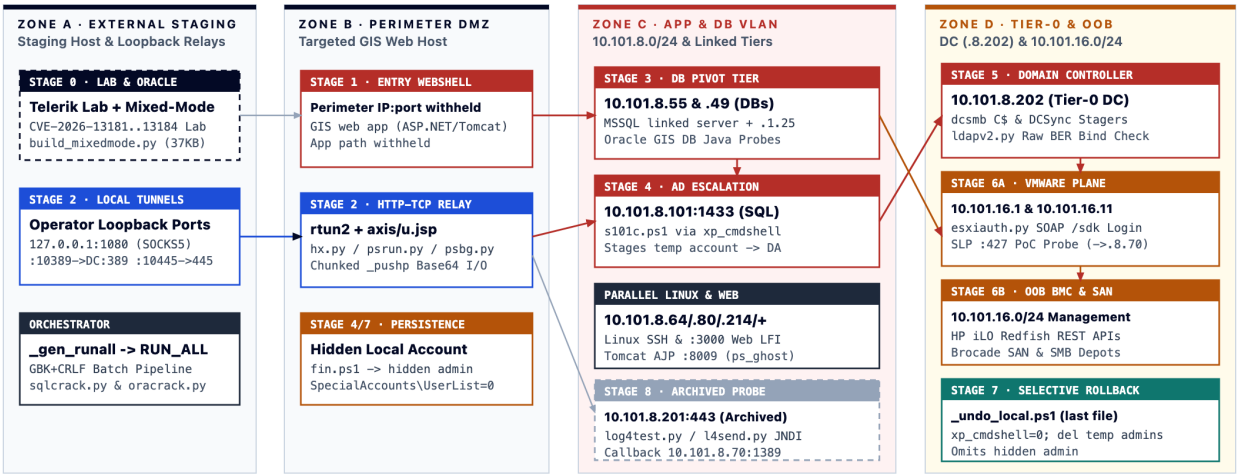
Figure 2 shows the staged path hop by hop, from the staging server through the DMZ and application network to the management plane.

FIGURE 2 · RECONSTRUCTED KILL-CHAIN & INTERNAL NETWORK TOPOLOGY

ODIN BY BRANDEFENSE · TLP:CLEAR

**Staged Scripts Target a Path From the Perimeter Web Tier Through Databases to AD and ESXi**

Reconstructed from hardcoded RFC1918 targets, SOCKS/port-forward bindings, and database pivots across 17 internal host IPs




**Scope & source:** 17 internal RFC1918 host IPs and ports extracted across 96 primary scripts (plus 10.103.4.0/24 sweep in m55k.ps1); perimeter IP and port withheld.

**What this does not show:** Arrows reflect staged script targets and tunnel bindings; Stage 0 targeted 127.0.0.1:8090 lab, and .8.201 JNDI probes were archived in folder 08 (failed attempts).

Figure 2: Reconstructed internal network path targeted by the staged scripts

04

CHAPTER 04

# Database Pivots, Parallel Tracks and Domain Replication (Stages 3 to 5)

IN SHORT

SQL Server jump hosts carry most of the internal movement, with Oracle, Linux and web tracks running in parallel. The workflow then stages temporary admins, a hidden admin, and DCSync scripts against the domain controller.

INSIDE THIS CHAPTER

SQL Server jump hosts and the linked server 14

Oracle, Linux and internal web tracks 14

Two kinds of admin accounts 15

DCSync stagers and raw LDAP bind checks 15

# 04 Database Pivots, Parallel Tracks and Domain Replication (Stages 3 to 5)

At least 96 scripts (26.4%) reference 17 internal hosts in `10.101.8.0/24`, `10.101.16.0/24` and `192.168.1.25`, plus the swept range `10.103.4.0/24`. Outcomes on these internal servers are not verified.

## SQL Server jump hosts and the linked server

The MSSQL tooling targets SQL Server on 10.101.8.48, 10.101.8.55 and 10.101.8.101 (port 1433). There are no `.sql` files for MSSQL. The operator used small PowerShell `System.Data.SqlClient` wrappers instead: 12 jump-host scripts in folders 04 and 07, each 578 to 1,132 bytes. Python `pymssql` relays (`mssql_try.py`, `reuse_mssql.py`, `mssql_via_socks.py`) complete the set.

- **Linked-server traversal.** Early scripts list linked-server logins on 10.101.8.55 with `sp_helplinkedsrvlogin`. They then use an RPC-enabled linked server. Later wrappers send `OPENQUERY` calls toward 192.168.1.25 (`m55dom.ps1`) and run `xp_cmdshell` ping and ARP sweeps across 10.103.4.0/24 (`m55k.ps1`).
- **Credential queries and offline cracking.** Other scripts query `sys.credentials`, SQL Agent job steps in `msdb`, and `sys.sql_logins` password hashes (0x0100 SHA-1 format). The set also holds an offline cracker, `sqlcrack.py`, which has an indentation bug.

| File | ID | Size (bytes) | Role against 10.101.8.55 |
|---|---|---|---|
| `m55g.ps1` | `#095` | 769 | Enables linked-server RPC options and runs remote `xp_cmdshell` |
| `m55.ps1` | `#100` | 734 | Tests SQL sysadmin credential candidates and checks the `sysadmin` role |
| `m55h.ps1` | `#121` | 638 | Enables `xp_cmdshell` on the linked server and runs host discovery |
| `m55adm.ps1` | `#156` | 776 | Adds a temporary local admin to `Administrators` and RDP groups |
| `m55e.ps1` | `#175` | 578 | Runs `arp -a` and queries the linked server with `OPENQUERY` |
| `m55dom.ps1` | `#248` | 687 | Queries cross-segment host 192.168.1.25 via `OPENQUERY` |
| `m55cred.ps1` | `#270` | 641 | Queries `sys.credentials` and SQL Agent jobs |
| `m55c.ps1` | `#289` | 773 | Reads `sys.sql_logins` hashes and the `sys.servers` catalog |
| `m55b.ps1` | `#327` | 802 | Lists server properties, `sys.configurations` and databases |
| `sqlcrack.py` | `#360` | 1,871 | Offline 0x0100 SHA-1 cracker (indentation error) |

## Oracle, Linux and internal web tracks

Scripts in the root folder run three parallel tracks across `10.101.8.0/24`.

- **Oracle Java procedures.** All 19 `.sql` files are Oracle `SQL*Plus` scripts aimed at the target's GIS database on 10.101.8.49, run through `sq.py` and `sqrun.py`. They grant `DBMS_JAVA` permissions, compile Java procedures (`HrExec`, `HrRun3`, `OSExec`) and stage Windows registry hive dumps for HTTP PUT upload. The captured `q12.out` records an `ORA-29535` error from that database.

- **Registry and SSH harvesting.** `hputty2.py` and `hssh.py` parse `HKEY_USERS` hives for saved PuTTY, WinSCP, mRemoteNG and SSH sessions. In all, 61 files reference PuTTY `plink`, `pscp`, SSH or NFSv3/v4 workflows (`rootsudo.py`, `nfs3.py`, `nfs4.py`). They target six Linux hosts: 10.101.8.49, .64, .72, .103, .107 and .196.
- **Internal web app and Ghostcat.** 103 files have names starting with `ps_gw`, `gw` or `lfi`. Eight of them test `/cs/log?filename=` directory traversal on 10.101.8.80:3000; the rest are delivery launchers and service probes. Separately, nine scripts target the Apache Tomcat AJP Ghostcat flaw (CVE-2020-1938 [8]) on port 8009, and seven name 10.101.8.214.

| File | ID | Size (bytes) | Role |
|---|---|---|---|
| `q14.sql` | `#051` | 3,483 | Java procedure `HrExec` runs commands and PUTs registry hives to :8080 |
| `q18.sql` | `#052` | 3,546 | Java procedure `HrRun3` decodes and runs `.bat` payloads |
| `q12.out` | `#161` | 3,161 | Captured output with `ORA-29535` and a pre-Java 7 `ProcessBuilder` error |
| `q17.sql` | `#213` | 1,643 | Saves `SAM`, `SYSTEM`, `SECURITY` and `SOFTWARE` hives through the database |
| `q10.sql` | `#238` | 1,203 | Reads command output from `C:\Windows\Temp` with `UTL_FILE` |
| `exfil1.ps1` | `#272` | 1,091 | PowerShell version that saves `SAM`, `SYSTEM` and `SECURITY` for upload |
| `mkq18.py` | `#306` | 4,010 | Rebuilds `q18.sql` with Base64 batch commands in Java source |
| `q5.sql` | `#309` | 1,372 | Lists Oracle password hashes, DB links, scheduler credentials and privileges |
| `hssh.py` | `#332` | 1,681 | Searches user folders and registry for SSH keys (syntax bug) |
| `hputty2.py` | `#351` | 1,796 | Lists saved PuTTY, WinSCP, mRemoteNG and RDP sessions per user SID |

## Two kinds of admin accounts

On SQL Server 10.101.8.101:1433, `xp_cmdshell` wrappers and root PowerShell scripts set up privileged accounts on two tracks. Account names are withheld.

1. **Short-lived accounts.** `s101c.ps1` creates a temporary domain account and adds it to `Domain Admins` for replication. `m55adm.ps1` adds a temporary local account to `Administrators` and `Remote Desktop Users`. The rollback script names both for deletion.
2. **A hidden local account.** Drafted in `mkuser.ps1` and `mkuser.py` and finished in `fin.ps1`, this track creates a local administrator. It hides the account from the sign-in screen with `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList` [7]. It then checks RDP registry settings and tests an interactive logon with `LogonUser` from `Advapi32.dll`.

## DCSync stagers and raw LDAP bind checks

DCSync abuses the directory replication protocol (`MS-DRSR`) to request password hashes from a domain controller [4].

1. **Share access and replication.** `dcsmb.py` tests `C$` share access on 10.101.8.202 over SOCKS5. `dcsync.py` calls Impacket `NTDSHashes` (`IDL_DRSGetNCChanges`) through `127.0.0.1:1080`. The set also holds an updated `dcsmb.py` and a root `dcsync.py` draft with an indentation bug. No `.ntds` output was captured, so hash replication is not verified.

2. **Raw LDAP binds.** `dombind.ps1` and the dependency-free `ldapv2.py` test binds against 10.101.8.202:389. The Python script builds RFC 4511 [10] BER `SimpleBindRequest` frames by hand (0x30, 0x60, 0x80) over `127.0.0.1:10389`. A root `dcsmb.py` compares the temporary domain admin with a deliberately wrong password over `127.0.0.1:10445`.

| File | ID | Size (bytes) | Role |
|---|---|---|---|
| `sql101.ps1` | `#064` | 935 | Tests SQL sysadmin credential candidates on 10.101.8.101, with lockout notes |
| `s101c.ps1` | `#075` | 694 | Creates the temporary domain admin via `xp_cmdshell` |
| `mkuser.ps1` | `#101` | 1,737 | First ADSI draft of the hidden local admin |
| `s101.ps1` | `#145` | 781 | Enables `xp_cmdshell` and lists domain users and services |
| `sql101b.ps1` | `#214` | 1,132 | Host and domain group enumeration via `xp_cmdshell` |
| `dcsmb.py` | `#256` | 1,432 | Extra revision testing `C$` on the DC via SOCKS5 |
| `dcsync.py` | `#288` | 3,452 | Project DCSync stager using Impacket `NTDSHashes` |
| `dcsmb.py` | `#295` | 1,641 | Updated `C$` readability check on the DC |
| `dcsync.py` | `#299` | 1,204 | Minimal root draft over `127.0.0.1:1080` (indentation bug) |
| `dombind.ps1` | `#326` | 715 | PowerShell LDAP bind tester |
| `ldapv2.py` | `#335` | 1,070 | Raw BER `SimpleBindRequest` tester |
| `dcsmb.py` | `#339` | 897 | Differential SMB check against a wrong-password control |
| `fin.ps1` | `#347` | 2,248 | Hides the local admin under `SpecialAccounts\UserList` and tests `LogonUser` |

05

CHAPTER 05

# Hypervisor Probing, Dead Ends and Selective Rollback (Stages 6 to 8)

IN SHORT

Alongside the AD work, scripts target the VMware and out-of-band management network. The last files are a rollback script that reverts what a posture scan would check and leaves a hidden admin behind.

INSIDE THIS CHAPTER

Toward virtualization and out-of-band management 18

Documented dead ends 18

The selective rollback 18

What a posture scan would miss 19

# 05 Hypervisor Probing, Dead Ends and Selective Rollback (Stages 6 to 8)

## Toward virtualization and out-of-band management

Scripts in folder `06_横向扩展` target `10.101.16.0/24`, which holds VMware and hardware management interfaces.

- **AD and SYSVOL harvesting.** `adrecon.py` queries LDAP over SOCKS5 for LAPS passwords (`ms-Mcs-AdmPwd`), Kerberoastable SPNs, AS-REP roastable accounts and `adminCount=1` members. `smbsweep.py` scans SYSVOL policies for Group Policy Preferences `cpassword` strings.
- **Passive ESXi build discovery.** `depothunt.py` and `shares.py` search SMB shares for VMware offline depot `.zip` files, `.vib` packages, `.iso` images and `id_rsa` keys. The `slpd` and `libc` binaries inside a depot reveal the ESXi patch level without touching the host.
- **Credential tests on management interfaces.** `esxiauth.py` tests ESXi `/sdk` SOAP logins (`RetrieveServiceContent`, then `sessionManager`). `creds16.ps1` and `creds16b.ps1` test credentials with lockout-aware delays across ESXi, HP iLO Redfish, Cisco and Brocade SAN switches.
- **vCenter and OpenSLP.** `vprobe.py` probes 10.101.16.11 on ports 443, 5480, 9443 and TCP 427 (`slpd`). `slpctl.ps1` sends Service Location Protocol packets to 10.101.16.1. In `06_横向扩展/poc/`, `exp_leak.py` and `exp15160138.py` adapt a public PoC for the OpenSLP heap overflow CVE-2021-21974 (VMSA-2021-0002 [3]). They point a reverse shell at 10.101.8.70:32222, served by `rshandler.py`. The files do not show whether 10.101.16.1 was unpatched or reached.

## Documented dead ends

Folder `08_未能奏效的尝试` ("failed attempts") holds two Log4Shell (CVE-2021-44228 [9]) probes.

1. **External callback.** `log4test.py` sends JNDI strings, including one in a POST XML username field, over SOCKS5 TLS to 10.101.8.201:443. It points the callback at a commercial out-of-band testing domain.
2. **Internal callback.** `l4send.py` (a later file) retries with a callback to an internal listener on 10.101.8.70:1389 (`l4listen.ps1`). Filing both under "failed attempts" suggests the operator gave up on this vector.

## The selective rollback

`_undo_local.ps1` is the rollback script. It connects to the SQL jump hosts, sets `xp_cmdshell` back to 0, and deletes the two temporary admin accounts.

| File | ID | Size (bytes) | Role |
|---|---|---|---|
| `creds16.ps1` | `#040` | 6,168 | Rate-limited credential tester for ESXi, HP iLO, Cisco and Brocade |
| `rshandler.py` | `#056` | 4,600 | File-driven reverse-shell listener (`cmd.txt`, `shell_out.txt`) |
| `smbsweep.py` | `#057` | 14,155 | SMB scanner and SYSVOL `cpassword` decryptor |
| `l4listen.ps1` | `#099` | 531 | Temporary TCP 1389 callback listener |
| `vprobe.py` | `#174` | 9,150 | Probes vCenter ports and the TCP 427 OpenSLP precondition |
| `adrecon.py` | `#195` | 12,301 | `ldap3` harvester for LAPS, SPNs and `adminCount=1` |

| File | ID | Size (bytes) | Role |
|---|---|---|---|
| `shares.py` | #197 | 6,201 | SMB share search for `.iso`, `.vib`, password files and `id_rsa` |
| `slpctl.ps1` | #221 | 6,082 | Positive and negative SLP control packets to 10.101.16.1 to .4 |
| `creds16b.ps1` | #241 | 7,264 | Refined SOAP, Redfish and HTTP Basic credential tester |
| `depothunt.py` | #242 | 5,772 | Finds ESXi offline depots to extract `slpd` and `libc` |
| `poc/exp15160138.py` | #257 | 19,461 | Modified public CVE-2021-21974 PoC for ESXi build 15160138 |
| `poc/exp_leak.py` | #258 | 20,265 | Modified public CVE-2021-21974 leak and RCE script |
| `esxiauth.py` | #296 | 7,545 | ESXi `/sdk` SOAP login tester |
| `remote_syscheck.txt` | #297 | 1,219 | Base64 JSP `cmd.exe /c` servlet with a red-team deployment and rollback comment |
| `log4test.py` | #311 | 1,696 | Archived JNDI probe |
| `l4send.py` | #338 | 1,010 | Archived JNDI retry with internal callback |
| `_undo_local.ps1` | #359 | 2,666 | Disables `xp_cmdshell` and deletes both temporary admins |
| `sptest.py` | #364 | 454 | SMB password-spraying and pass-the-hash wrapper |

## What a posture scan would miss

Figure 3 models what happens if `_undo_local.ps1` and `fin.ps1` run as written.

FIGURE 3 · POST-ROLLBACK DETECTION VISIBILITY MATRIX

ODIN BY BRANDEFENSE · TLP:CLEAR

**Point-in-Time Posture Scans Miss Staged SQL and AD Changes If _undo_local.ps1 Runs as Written**

Comparison of hypothetical post-rollback static configuration state against event-log and wire evidence across six targeted surfaces

| TARGETED SURFACE | STATIC STATE IF ROLLBACK RAN | SURVIVING EVENT & WIRE EVIDENCE | PRIMARY DETECTION SOURCE |
|---|---|---|---|
| **1. GIS Web Root**<br>App upload folder | **UNVERIFIED · No staged cleanup**<br>script removes _pushp staging chunks | **HTTP POST burst to web handler + w3wp.exe /**<br>tomcat9.exe spawning cmd.exe / powershell.exe | **EDR Process Tree +**<br>Suricata HTTP _pushp |
| **2. MSSQL Jump Hosts**<br>10.101.8.55 & .101 | **REVERTED · sys.configurations shows**<br>xp_cmdshell = 0 if _undo_local.ps1 ran | **SQL ErrorLog Event ID 15457 config-change sequence:**<br>0 -> 1 followed by 1 -> 0 (+ SQL Audit / EDR) | **MSSQL Event 15457 +**<br>Linked-Server Audit |
| **3. Escalation Accounts**<br>Temp Domain / Local Admins | **DELETED · Both accounts removed from**<br>AD & local Admins if _undo_local.ps1 ran | **SID-correlated lifecycle: Event ID 4720 ->**<br>4728/4732/4756 (MemberSid) -> 4726 (<= 72h) | **Sentinel SecurityEvent**<br>4720 / 4728..56 / 4726 |
| **4. Domain Controller**<br>10.101.8.202 (AD) | **NO AGENT · MS-DRSR & LDAP scripts**<br>stage zero files or keys on DC disk | **Cleartext raw ASN.1 BER Bind + Event ID 4662**<br>GUID 1131f6ad-9c07-11d1-f79f-00c04fc2dcd2 | **DC Security 4662 +**<br>Suricata Port 389 BER |
| **5. Stealth Persistence**<br>Hidden Local Admin (fin.ps1) | **OMITTED BY ROLLBACK · Hidden from UI**<br>if fin.ps1 ran; untouched by _undo_local | **HKLM\...\Winlogon\SpecialAccounts\UserList**<br>DWORD = 0 + Advapi32!LogonUser check | **Sysmon Event ID 13 +**<br>Registry Posture Check |
| **6. Hypervisor & BMC**<br>10.101.16.1 / .11 | **NO WINDOWS EDR · Out-of-band ESXi**<br>& BMC appliances lack AD EDR agents | **ESXi /var/run/log/hostd*.log SOAP Login events +**<br>TCP/UDP 427 OpenSLP PoC probe packets | **ESXi Syslog + VLAN**<br>ACL / NDR Port 427 |

**Scope & source:** Modeled from staged _undo_local.ps1 (#359, 2,666 B) and fin.ps1 (#347, 2,248 B) in the dated project folder.
**What this does not show:** Target-side script execution is unverified from staged files alone; log survival requires off-host SIEM/NDR forwarding.

Figure 3: Point-in-time posture scans miss staged SQL and AD changes if the rollback runs as written

- **What looks clean.** `sys.configurations` shows `xp_cmdshell = 0` on the jump hosts. `Domain Admins` and local `Administrators` no longer list the temporary accounts.
- **What survives.** No staged cleanup script removes the `_pushp` fragments in the web folder. The hidden local admin from `fin.ps1` is not in the rollback. DCSync and raw LDAP operations against 10.101.8.202 write no files to the domain controller, so only event logs and network records keep a trace.

CHAPTER 06

# Peer Toolkits and Alternative Explanations

IN SHORT

Three other staging toolkits from the same period share the same shape: a flat working folder, curated subfolders, and an application-layer relay into internal data stores. The files cannot tell a hostile intrusion from a red-team test, a lab or a decoy.

INSIDE THIS CHAPTER

Four staging toolkits compared 21

Who is behind it: four hypotheses 22

# 06 Peer Toolkits and Alternative Explanations

## Four staging toolkits compared

We compared this toolkit (Kit B) with three other multi-stage staging servers seen in the same period. Together they hold 1,458 files.

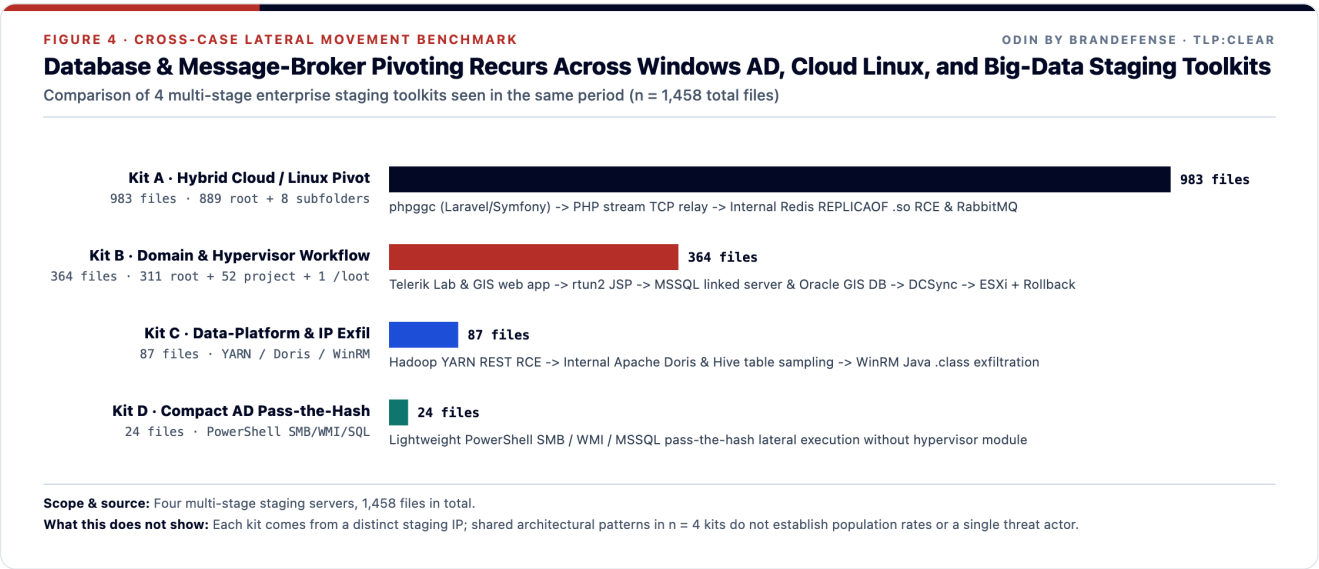



Figure 4: Database and message-broker pivoting recurs across four staging toolkits

| Kit | Files | Layout | Staged path |
|---|---|---|---|
| **Kit A** | 983 | 889 in root, 94 in 8 subfolders | `phpggc` Laravel and Symfony deserialization, PHP stream TCP relay, internal Redis `REPLICAOF` `.so` RCE and RabbitMQ |
| **Kit B** (this report) | 364 | 311 in root, 52 in project folder, 1 in `/loot` | Telerik lab and GIS web shell, `rtun2` JSP relay, MSSQL linked server and Oracle GIS database, DCSync and ESXi probes, rollback |
| **Kit C** | 87 | 87 in root | Unauthenticated Hadoop YARN REST RCE, internal Apache Doris and Hive sampling, WinRM Java `.class` exfiltration |
| **Kit D** | 24 | 24 in root | Local WCF and SQL credential extraction, PowerShell SMB, WMI and MSSQL pass-the-hash |

Two patterns repeat across Windows AD, Linux cloud and big-data targets.

1. **Flat root plus curated subfolders.** Operators iterate in the root folder (85.4% of files in Kit B, 90.4% in Kit A). They move repeatable chains into subfolders (the project folder in Kit B; `bundle` and `chains` in Kit A).
2. **Application-layer relays into data stores.** JSP and ASP.NET chunked relays in Kit B mirror PHP stream-socket relays in Kit A. Each kit pivots through high-trust data stores: MSSQL and Oracle (B), Redis and RabbitMQ (A), Hadoop YARN, Doris and Hive (C), and MSSQL (D).

Four kits are too few to estimate how common these patterns are, and nothing here links the kits to one actor.

## Who is behind it: four hypotheses

Several clues point in different directions. The file `remote_syscheck.txt` carries a red-team deployment and rollback comment. Folder 08 archives failed attempts. Kit A holds a probable tester account and cleanup scripts. The 38 root-folder files target a local lab, and the ESXi folder adapts public exploit code.

| Hypothesis | For | Against |
|---|---|---|
| **Hostile or gray-hat intrusion** | The rollback leaves the hidden admin in place. Scripts target an enterprise GIS schema, a linked server and 17 internal hosts. The modified OpenSLP PoCs could crash ESXi `slpd`. | No victim-side EDR, SIEM or flow data confirms entry, hash replication or impact on a live network. |
| **Authorized red team or contractor** | The red-team comment in `remote_syscheck.txt`, numbered stage folders, `RUN_ALL.bat` automation, commercial out-of-band testing, lockout-aware delays, negative SLP controls and tester markers in Kit A. | Leaving 364 scripts and internal topology on an open server breaks assessment OPSEC. The rollback also omits the hidden admin. |
| **Research lab or reused open-source fixture** | The 38 root-folder files (10.4%) target a local IIS lab. The ESXi PoCs are public. 21 drafts have syntax or compilation errors, typical of iterative testing. | Later revisions name 17 internal hosts, GIS schemas, and an `ORA-29535` output from a pre-Java 7 Oracle JVM. A stock fixture would not hold these unless run against a custom range. |
| **Honeypot or decoy server** | All four kits sat on open HTTP workspaces in the same period, a posture some research honeypots use. | Revised file versions, including broken drafts fixed later and `GBK`-encoded Chinese tooling, reflect active editing, not a static lure. |

**Verdict.** The staged files cannot settle which explanation is right. We hold no data from the referenced network, so this report does not claim that a real enterprise was breached. The conclusions cover the workflow mechanics and the controls needed to detect them.

07

CHAPTER 07

# What Defenders Should Do

IN SHORT

This chapter lists the actions defenders should take, in priority order.

INSIDE THIS CHAPTER

Post-rollback checklist 25

# 07 What Defenders Should Do

These steps follow the staged workflow from web entry to the domain and the hypervisor, ordered by urgency. The post-rollback checklist at the end of this chapter records who owns each check.

**Within 24 hours**

- [ ] Search endpoints and web folders for the six target-side PowerShell hashes in the Indicators chapter.
- [ ] Run a fleet-wide check of `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList` for hidden accounts.
- [ ] Check SQL Server error logs for `xp_cmdshell` changing from 0 to 1 and back to 0, using Event ID 15457 across all rotated archives.

**Within 72 hours**

- [ ] Turn on the 72-hour short-lived admin lifecycle query in Sentinel and Defender XDR, including member servers.
- [ ] Enable Directory Service Access auditing and a replication-rights SACL on domain controllers, so Event 4662 records DCSync attempts.
- [ ] Scan web upload folders and `App_Data` for `_pushp`, `PSBG_DONE` and `JSPOK|` markers.

**Within 7 to 30 days**

- [ ] Upgrade internet-facing Telerik UI for ASP.NET AJAX to 2026.2.708 or later (CVE-2026-13182).
- [ ] Audit linked-server logins with RPC-out or data access enabled, and Oracle `DBA_JAVA_POLICY` grants.
- [ ] Isolate ESXi `/sdk`, TCP and UDP 427 (`slpd`), HP iLO Redfish and SAN interfaces from application and database subnets. Disable `slpd` where VMSA-2021-0002 applies.
- [ ] Move LDAP simple binds to LDAPS or SASL, and alert on cleartext simple binds to domain controllers.

## Post-rollback checklist

Use this checklist in purple-team exercises or web-to-domain intrusion reviews. Fill in the owner and status columns locally.

| Check | Owner | Evidence | Status | Next review |
|---|---|---|---|---|
| Internet-facing Telerik UI assemblies inventoried and at 2026.2.708 or later | | | | |
| Web upload folders and `App_Data` scanned with the YARA rules | | | | |
| SQL Event 15457 history (`show advanced options`, `xp_cmdshell`) compared with `sys.configurations`, SQL Audit and EDR | | | | |
| Linked-server logins and Oracle `DBA_JAVA_POLICY` grants audited | | | | |
| `SpecialAccounts\UserList` checked on every Windows host | | | | |
| Short-lived admin lifecycle rules active in Sentinel and Defender XDR | | | | |
| DC replication SACL (4662) and port 389 BER rules active | | | | |
| ESXi `/sdk`, port 427, HP iLO Redfish and SAN networks isolated | | | | |

CHAPTER 08

# Detection Rules

> IN SHORT
>
> This chapter describes each published rule and where to deploy it.

INSIDE THIS CHAPTER


SQL Server audit 27

Sentinel and Defender XDR query 27

Sigma rules 28

ESXi hunting 28

Suricata signatures 28

YARA rules 29

MITRE ATT&CK mapping 29

Rule testing 29

# 08 Detection Rules

The full rules are in Appendix B. Each file targets a step of the staged workflow that leaves a trace even after rollback.

| Rule file | What it catches |
|---|---|
| `sql_linked_server_and_event15457_audit.sql` | T-SQL: lists linked servers that allow RPC-out or data access, and finds `xp_cmdshell` enable-then-disable sequences in error logs while the option is now off |
| `sentinel_short_lived_admin_lifecycle.kql` | Sentinel and Defender XDR: accounts created, added to a privileged group and deleted within 72 hours |
| `sigma_staged_workflow.yml` | Sigma (3 rules): hidden `UserList` accounts, replication-right access from non-allowlisted principals, and web workers spawning chunked stagers |
| `suricata_staging_and_ldap_bind.rules` | Suricata 7 (10 rules): cleartext LDAP simple binds, `_pushp` chunks, `PSBG_DONE` and `JSPOK\|` markers, and `${jndi:` probes |
| `yara_webroot_and_persistence.yar` | YARA (3 rules): chunked stager scripts, minimal JSP command shells, and hidden-admin PowerShell |

## SQL Server audit

Part A lists linked servers with `is_rpc_out_enabled` or `is_data_access_enabled` set, with their login mappings. It shows configuration only, not traversal use. Full results need `ALTER ANY LINKED SERVER` and `ALTER ANY LOGIN`, or `setupadmin` or `sysadmin`; lower rights silently omit rows.

Part B reads error logs 0 to 6 for Event ID 15457 [5] and compares them with `sys.configurations`. Its strongest result flags an `xp_cmdshell` 0-to-1 change followed by a 1-to-0 change while the option is now off. Event 15457 records a configured-value change, not proof that `RECONFIGURE` made it active. Pair a hit with SQL Server Audit or `sqlservr.exe` spawning `cmd.exe` before calling it a real enable-and-rollback cycle.

## Sentinel and Defender XDR query

The `SecurityEvent` query joins three events on the account SID within 72 hours: creation (4720), addition to a privileged group (4728, 4732 or 4756) and deletion (4726). Privileged groups are matched by SID: Domain Admins (-512), Enterprise Admins (-519), local Administrators ( `S-1-5-32-544` ) and Remote Desktop Users ( `S-1-5-32-555` ). A companion query does the same with `IdentityDirectoryEvents` in Defender XDR. Exclude PAM or just-in-time provisioning accounts only when tied to verified tickets.

## Sigma rules

| Rule | Looks for | Needs |
|---|---|---|
| Hidden local account | A DWORD value of 0 set under `Winlogon\SpecialAccounts\UserList` (T1564.002) | Sysmon Event ID 13 |
| Replication-right access | Event 4662 with access mask 0x100 and a replication extended-right GUID, from a principal outside an explicit list of DC and Entra Connect accounts (T1003.006) | Directory Service Access auditing and a replication SACL |
| Web worker stager | `w3wp.exe`, Tomcat or `java.exe` spawning `cmd.exe`, PowerShell or `certutil` with arguments such as `-enc`, `FromBase64String`, `Set-Content`, `_pushp` or `PSBG_DONE` (T1505.003, T1059.001, T1059.003) | Sysmon Event ID 1 or Event 4688 with command lines |

In the replication rule, list exact DC machine accounts and Entra Connect sync accounts. Never exclude wildcards such as `*$`, `MSOL_*` or `AAD_*`, or an attacker account matching those prefixes could slip through. The web worker rule does not see unmarked `cmd.exe /c` runs through `tjsp.jsp`; pair it with Suricata SID 2026091109 and the JSP YARA rule.

## ESXi hunting

On ESXi hosts, review active and rotated `hostd`, `auth` and `syslog` logs for bursts of SOAP and SSH login failures per source address, and for `slpd` crashes. Disable OpenSLP where it is not needed.

## Suricata signatures

| SID | Looks for | Why |
|---|---|---|
| 2026091101 | LDAPv3 simple bind with a non-empty password, short-form BER | Catches `ldapv2.py`-style binds to port 389 (T1078.002) |
| 2026091102 | Same, definite 0x81 length for long DNs | Covers longer AD distinguished names |
| 2026091103 | Same, definite 0x82 length | Covers very long DNs |
| 2026091104 | Same, Windows `wldap32` 0x84 encoding | Covers binds from Windows clients |
| 2026091105 | `_pushp` in an HTTP request body | Chunked web shell staging over POST (T1505.003) |
| 2026091106 | `_pushp` in an HTTP URI | The GET fallback |
| 2026091107 | `PSBG_DONE` in an HTTP response | Background PowerShell completion marker |
| 2026091108 | `${jndi:` with `ldap`, `ldaps`, `rmi` or `dns` in HTTP headers | Internal Log4Shell probes (T1190) |
| 2026091109 | `JSPOK\|` in an HTTP response | Output of the minimal JSP command shell |
| 2026091110 | `${jndi:` in an HTTP request body | The POST XML variant in `log4test.py` |

The LDAP rules need visibility of domain controller port 389 traffic. Allowlist legacy printers or bind accounts until they move to LDAPS or SASL. The JNDI rules on port 443 need TLS decryption or a reverse proxy in front of Suricata. They ship with `$HOME_NET` as source so they load on stock configs; exclude your vulnerability scanners once defined.

## YARA rules

| YARA rule | Fires when |
|---|---|
| `ODIN_R01_Webroot_Chunked_PS1_Stager` | `_pushp` appears with `PSBG_DONE`, `FromBase64String`, `WriteAllText` or `Set-Content`, or `PSBG_DONE` appears with `FromBase64String`, in a file under 64 KB |
| `ODIN_R01_JSP_Cmd_Webshell_Markers` | `JSPOK\|` appears, or `cmd.exe /c` and `request.getParameter` appear with `Runtime.getRuntime().exec` or `ProcessBuilder` |
| `ODIN_R01_PS1_SpecialAccounts_Hidden_Admin` | `Winlogon\SpecialAccounts\UserList` appears with `advapi32.dll` and `LogonUser`, or with `fDenyTSConnections` |

The Base64-wrapped JSP in `remote_syscheck.txt` must be decoded before the JSP rule can see it. For mixed-mode assemblies, look for .NET files in web upload folders that also import native `kernel32.dll` execution APIs. For LDAP, inspect `BindRequest` frames from web-tier hosts.

## MITRE ATT&CK mapping

| Tactic | Technique | Staged behaviour | Detection |
|---|---|---|---|
| Initial Access | T1190 Exploit Public-Facing Application | JNDI probes against 10.101.8.201 | Suricata 2026091108, 2026091110 |
| Persistence | T1505.003 Web Shell | ASP.NET and JSP shells, `_pushp` chunks | Suricata 2026091105 to 2026091107, 2026091109; Sigma rule 3; YARA |
| Execution | T1059.001 PowerShell, T1059.003 Windows Command Shell | Web workers launching PowerShell and `cmd.exe` stagers | Sigma rule 3 |
| Defense Evasion | T1564.002 Hidden Users | `fin.ps1` hides a local admin under `UserList` | Sigma rule 1; YARA hidden-admin rule |
| Credential Access | T1003.006 DCSync | `dcsync.py` with Impacket `NTDSHashes` | Sigma rule 2 |
| Defense Evasion, Persistence | T1078.002 Domain Accounts | Raw LDAP bind and SMB checks with domain credentials | Suricata 2026091101 to 2026091104 |

## Rule testing

Every rule file loads without errors in its engine (Suricata 7.0.12, YARA-X, pySigma, Kusto.Language); the rules were not replayed against live traffic. Test them in your own environment before production alerting.

09

CHAPTER 09

# Indicators

IN SHORT

This chapter lists the published indicators and explains what stays withheld.

INSIDE THIS CHAPTER

Target-side PowerShell hashes 31

Staging servers 31

# 09 Indicators

Appendix A holds the 29 file hashes, which cover Stages 0 to 8. The four staging server addresses are withheld.

| Type | Count |
|---|---|
| Target-side PowerShell file hashes (SHA-256), for endpoint hunting | 6 |
| Staging-side file hashes (22 Python, 1 batch), for context | 23 |
| External staging servers (addresses withheld) | 4 |

Each hash is the SHA-256 of the original file on the staging server. The 23 staging-side files were built to run from the operator's server over HTTP and SOCKS relays. Their hashes will not appear on victim disks, so hunt their network and log traces with the rules in Chapter 8.

## Target-side PowerShell hashes

| Stage | File | Size (bytes) | SHA-256 |
|---|---|---|---|
| 4 | `sql101.ps1` | 935 | `2ebf58a92c2e5066d7347eff9d58e2c2e1c9ca36f4c01a9fd8ecd299d65a4183` |
| 4 | `s101c.ps1` | 694 | `5bc431e751d0113b9c38140728627da9841da20030c6c505ad4e401fc0bf81fb` |
| 4 | `fin.ps1` | 2,248 | `e9568e4f3219a0c8f6300ada70d85027fda4efaaa335c9b845130ae229672662` |
| 6 | `slpctl.ps1` | 6,082 | `7978881fccdccf76ebc2b7753cc90825eb0775dd16ef6772d7bf95aee196ee4a` |
| 6 | `creds16b.ps1` | 7,264 | `ae7bc4c75b662208ccb61a53a0cb05321fcf87af68829a1280f817b7aa57ce3b` |
| 7 | `_undo_local.ps1` | 2,666 | `24737d6be02f88d7c0b01b34d1d316325397cd77f4c51ede26fa3dc93c2f0317` |

## Staging servers

| Server | Kit | Files |
|---|---|---|
| Address withheld | Kit B (this report) | 364 |
| Address withheld | Kit A | 983 |
| Address withheld | Kit C | 87 |
| Address withheld | Kit D | 24 |

All four addresses are withheld, because an authorized red team or test range cannot be ruled out for any of them. The file hashes stay published: they describe tools, not their owner.

Withheld: the four staging server addresses, the targeted perimeter address, port and application path; organisation, database, linked-server and schema names; account names; credentials; the commercial out-of-band domain; and the tester account in Kit A.

10

CHAPTER 10

# Scope and Limits

IN SHORT

This chapter sets out what the evidence can show and what it cannot.

# 10 Scope and Limits

- **Window.** The findings come from a two-week observation window ending 18 November 2026. The staging files themselves were first seen earlier; this report describes what they contain, not when they were written.
- **Staged is not done.** The files show intent and tooling. They do not show that any script ran on a target.
- **One captured output.** Only `/loot/q12.out` records contact with a target system, an Oracle error. No `.ntds` file, account log or hypervisor log was captured.
- **No victim data.** We hold no logs from the referenced network. We cannot say whether a real enterprise, a red-team client or a test range sat behind the internal addresses.
- **Counts are lower bounds.** Working code and secrets are left out, so counts such as the 96 scripts with internal addresses are minimums.
- **Small comparison.** The four kits are a small sample. Shared patterns do not show how common these methods are, or that one actor runs all four.
- **Rules need local testing.** See the rule testing note in Chapter 8.

11

CHAPTER 11

# Sources

IN SHORT

This chapter lists the public references used in the report.

# 11 Sources


1. Progress Telerik. Cryptographic oracle vulnerability in RadAsyncUpload (CVE-2026-13182). https://www.telerik.com/products/aspnet-ajax/documentation/knowledge-base/kb-security-rau-padding-oracle-cve-2026-13182 (verified 18 November 2026)
2. NIST NVD. CVE-2019-18935, Progress Telerik UI for ASP.NET AJAX insecure deserialization. https://nvd.nist.gov/vuln/detail/CVE-2019-18935 (verified 18 November 2026)
3. Broadcom / VMware. VMSA-2021-0002, ESXi OpenSLP heap overflow (CVE-2021-21974) and workaround. https://support.broadcom.com/web/ecx/support-content-notification/-/external/content/SecurityAdvisories/0/23599 (verified 18 November 2026)
4. Microsoft Open Specifications. [MS-DRSR] `IDL_DRSGetNCChanges`. https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-drsr/b63730ac-614c-431c-9501-28d6aca91894 (verified 18 November 2026)
5. Microsoft. Database Engine events and errors 15000 to 15999 (Event 15457). https://learn.microsoft.com/en-us/sql/relational-databases/errors-events/database-engine-events-and-errors-15000-to-15999 and `xp_cmdshell` server configuration option. https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/xp-cmdshell-server-configuration-option (verified 18 November 2026)
6. MITRE ATT&CK. T1003.006 OS Credential Dumping: DCSync. https://attack.mitre.org/techniques/T1003/006/ (verified 18 November 2026)
7. MITRE ATT&CK. T1564.002 Hide Artifacts: Hidden Users. https://attack.mitre.org/techniques/T1564/002/ (verified 18 November 2026)
8. NIST NVD. CVE-2020-1938, Apache Tomcat AJP (Ghostcat). https://nvd.nist.gov/vuln/detail/CVE-2020-1938 (verified 18 November 2026)
9. NIST NVD. CVE-2021-44228, Apache Log4j2 JNDI (Log4Shell). https://nvd.nist.gov/vuln/detail/CVE-2021-44228 (verified 18 November 2026)
10. IETF. RFC 4511, LDAP: The Protocol, Section 4.2 Bind Operation. https://datatracker.ietf.org/doc/html/rfc4511#section-4.2 (verified 18 November 2026)

APPENDICES

# Indicators and Detection Rules

FOR DEFENDERS

Appendix A lists the indicators published for this study. Appendix B carries every published detection rule in full.

INSIDE

A. Indicators of compromise (29) 37

B. Detection rules (5 files) 39

# A Indicators of Compromise

Values are defanged for display. The CSV, JSON and STIX files on the online edition carry the raw values. Values the report marks as withheld are not listed.

| ID | Type | Indicator | Port | Context | Action |
|---|---|---|---|---|---|
| R01-IOC-01 | SHA-256 | 9cb603561bfddf4bcda418eb91b12dc420f942e92fe37e5592e95beb35f35c82 | | telerik_oracle_attack.py · Stage 0: Telerik / ViewState Cryptographic Oracle | HUNT |
| R01-IOC-02 | SHA-256 | 6fe1405fb45d62ab9712222a1327543fcf8acc395d51b5aaf1e3bbe5e317b48d | | build_mixedmode.py · Stage 0: Telerik / ViewState Cryptographic Oracle | HUNT |
| R01-IOC-03 | SHA-256 | 7c30b9335e1b959e9ab0f0976b5ef57a4ce0432a3c88f50fee6f9a74036b74a4 | | psrun.py · Stage 1: GIS Webshell & Chunked PowerShell Stager | HUNT |
| R01-IOC-04 | SHA-256 | 336d5524bebd3eeed980a5cb08ab9df393295602e2f896f46bee9e1aa910e7a7 | | hx.py · Stage 1: GIS Webshell & Chunked PowerShell Stager | HUNT |
| R01-IOC-05 | SHA-256 | 3ed59e26b9565b932afc301e4a56a7a54b67ab537053b5110c8d3ccd82fd9dc6 | | psbg.py · Stage 1: GIS Webshell & Chunked PowerShell Stager | HUNT |
| R01-IOC-06 | SHA-256 | 8afedc9459796f7de0758e7eaea32b827e366333eb90d5d5b19b4033119fc1e9 | | dl.py · Stage 2: HTTP-TCP Tunneling (rtun2) | HUNT |
| R01-IOC-07 | SHA-256 | 3fa8bfb85fe32a6e59e41cf369c691021aff57a5076d31d5833e8ba613a4822b | | sy.py · Stage 2: HTTP-TCP Tunneling (rtun2) | HUNT |
| R01-IOC-08 | SHA-256 | 70ce638c51e7ac1a959dc0ae990d9df1894ed018d73f68ecee9eaa1c4a1ac816 | | RUN_ALL.bat · Stage 2: HTTP-TCP Tunneling (rtun2) | HUNT |
| R01-IOC-09 | SHA-256 | d763aeefc3133bfcbca8c0ca6ec01cdba318da059ed597f645e2f967d955c515 | | _gen_runall.py · Stage 2: HTTP-TCP Tunneling (rtun2) | HUNT |
| R01-IOC-10 | SHA-256 | 4283d8dd470fda7ea61fa846d143dc7cefd6138ad9fc5d77928e0b21dfae71b4 | | tunkick2.py · Stage 2: HTTP-TCP Tunneling (rtun2) | HUNT |
| R01-IOC-11 | SHA-256 | bae408daf616b9d10f991e06c342187e3aa0daceebf1355fc964e916b464a7d9 | | sqlcrack.py · Stage 3: Database Pivoting (MSSQL & Oracle GISDB) | HUNT |
| R01-IOC-12 | SHA-256 | 2ebf58a92c2e5066d7347eff9d58e2c2e1c9ca36f4c01a9fd8ecd299d65a4183 | | sql101.ps1 · Stage 4-5: AD Identity Escalation & Raw BER DCSync | HUNT |
| R01-IOC-13 | SHA-256 | 5bc431e751d0113b9c38140728627da9841da20030c6c505ad4e401fc0bf81fb | | s101c.ps1 · Stage 4-5: AD Identity Escalation & Raw BER DCSync | HUNT |
| R01-IOC-14 | SHA-256 | e9568e4f3219a0c8f6300ada70d85027fda4efaaa335c9b845130ae229672662 | | fin.ps1 · Stage 4-5: AD Identity Escalation & Raw BER DCSync | HUNT |
| R01-IOC-15 | SHA-256 | 3f306ceec2e9972a3f2a9539f1c429eea351f5086db228ca3b4c279181cd28cb | | dcsync.py · Stage 4-5: AD Identity Escalation & Raw BER DCSync | HUNT |
| R01-IOC-16 | SHA-256 | aaa4d993332f6155a72358bb5e11528a23c40212e709b672e973cfafe7ebc7e7 | | dcsmb.py · Stage 4-5: AD Identity Escalation & Raw BER DCSync | HUNT |
| R01-IOC-17 | SHA-256 | 943f2efcda3f33f7ef873c5be752fe6cad023d0ce170abc75423b55b056a6225 | | ldapv2.py · Stage 4-5: AD Identity Escalation & Raw BER DCSync | HUNT |
| R01-IOC-18 | SHA-256 | 15149e9d4494bb97f9779f01093172aa4102953777454af828b5b2c306a9aaae | | rshandler.py · Stage 6: Hypervisor (ESXi), BMC (iLO) & SAN Pivot | HUNT |
| R01-IOC-19 | SHA-256 | 61f9b9851d6c1e6d3fe570bf68389fa9b7c6ef85906a5899104f2bbd46f7e987 | | smbsweep.py · Stage 6: Hypervisor (ESXi), BMC (iLO) & SAN Pivot | HUNT |
| R01-IOC-20 | SHA-256 | 99adb79793a83ef893d2801df37550999d43adc8b766b27495f626b0772bf2ce | | vprobe.py · Stage 6: Hypervisor (ESXi), BMC (iLO) & SAN Pivot | HUNT |

| ID | Type | Indicator | Port | Context | Action |
|---|---|---|---|---|---|
| R01-IOC-21 | SHA-256 | 902c4114a5476f9b7accdfebc8dfa7ff3fe56ed42a5923795463800f0f8d9ce6 | | adrecon.py · Stage 6: Hypervisor (ESXi), BMC (iLO) & SAN Pivot | HUNT |
| R01-IOC-22 | SHA-256 | 7978881fccdccf76ebc2b7753cc90825eb0775dd16ef6772d7bf95aee196ee4a | | slpctl.ps1 · Stage 6: Hypervisor (ESXi), BMC (iLO) & SAN Pivot | HUNT |
| R01-IOC-23 | SHA-256 | ae7bc4c75b662208ccb61a53a0cb05321fcf87af68829a1280f817b7aa57ce3b | | creds16b.ps1 · Stage 6: Hypervisor (ESXi), BMC (iLO) & SAN Pivot | HUNT |
| R01-IOC-24 | SHA-256 | f54f8cc31b0b7cba1e0e7ff67e371e0adc9854110bd6c8e42826980092a7171a | | depothunt.py · Stage 6: Hypervisor (ESXi), BMC (iLO) & SAN Pivot | HUNT |
| R01-IOC-25 | SHA-256 | 5f7577cd946d64b56dbd0395ecfc82b5799da697121b5aa05bd4190dcaae89bb | | exp_leak.py · Stage 6: Hypervisor (ESXi), BMC (iLO) & SAN Pivot | HUNT |
| R01-IOC-26 | SHA-256 | 9a987a64174314648c17ac21fcb07166758b7018c3068ad6c3ebd50b04ed3193 | | esxiauth.py · Stage 6: Hypervisor (ESXi), BMC (iLO) & SAN Pivot | HUNT |
| R01-IOC-27 | SHA-256 | 24737d6be02f88d7c0b01b34d1d316325397cd77f4c51ede26fa3dc93c2f0317 | | _undo_local.ps1 · Stage 7: Anti-Forensic Rollback | HUNT |
| R01-IOC-28 | SHA-256 | 3c6f49b10dceab9ecb2c64c2581027da4ef6f0580c0c96a5b858512bb297b6fb | | log4test.py · Stage 8: Failed JNDI / Alternative Probes | HUNT |
| R01-IOC-29 | SHA-256 | 4db277e5258d34d88f53243e1f556c59864b3a4acc0846a172734f02486c319f | | l4send.py · Stage 8: Failed JNDI / Alternative Probes | HUNT |

# B Detection Rules

Each rule file loads in its own engine (Kusto, pySigma, Suricata 7, YARA-X). Tune thresholds and allow-lists to your environment before alerting or blocking. The same files can be downloaded from the online edition.

## sentinel_short_lived_admin_lifecycle.kql

KQL (SENTINEL / DEFENDER XDR)

```
let lookback = 72h;
let created_users = SecurityEvent
    | where TimeGenerated >= ago(lookback) and EventID == 4720
    | where isnotempty(TargetSid)
    | project CreatedTime = TimeGenerated, TargetUserName, TargetSid, Creator = SubjectUserName, CreatedHost =
Computer;
let priv_group_adds = SecurityEvent
    | where TimeGenerated >= ago(lookback) and EventID in (4728, 4732, 4756)
    | where TargetSid endswith "-512" or TargetSid endswith "-519"
        or TargetSid in ("S-1-5-32-544", "S-1-5-32-555")
        or (isempty(TargetSid) and TargetUserName has_any ("Domain Admins", "Administrators", "Enterprise
Admins", "Remote Desktop Users"))
    | project EscalatedTime = TimeGenerated, MemberSid, GroupName = TargetUserName, GroupSid = TargetSid,
GroupEventID = EventID;
let deleted_users = SecurityEvent
    | where TimeGenerated >= ago(lookback) and EventID == 4726
    | where isnotempty(TargetSid)
    | project DeletedTime = TimeGenerated, TargetSid, Deleter = SubjectUserName, DeletedHost = Computer;
created_users
| join kind=inner (deleted_users) on TargetSid
| where DeletedTime > CreatedTime and (DeletedTime - CreatedTime) <= 72h
| join kind=inner (priv_group_adds) on $left.TargetSid == $right.MemberSid
| where EscalatedTime >= CreatedTime and EscalatedTime <= DeletedTime
| project CreatedTime, EscalatedTime, DeletedTime, Lifespan = DeletedTime - CreatedTime,
          TransientAccount = TargetUserName, TargetSid, GroupName, GroupSid, GroupEventID, Creator, Deleter,
CreatedHost;

// Companion Query: Microsoft Defender XDR Advanced Hunting (IdentityDirectoryEvents)
let xdr_lookback = 72h;
let xdr_creates = IdentityDirectoryEvents
    | where Timestamp >= ago(xdr_lookback) and ActionType =~ "Account Created"
    | extend CorrelationId = coalesce(tostring(AdditionalFields["TargetAccountSid"]), TargetAccountUpn)
    | where isnotempty(CorrelationId)
    | project CreatedTime = Timestamp, CorrelationId, AccountUpn = TargetAccountUpn, AccountName =
TargetAccountDisplayName, DC = DestinationDeviceName;
let xdr_group_adds = IdentityDirectoryEvents
    | where Timestamp >= ago(xdr_lookback) and ActionType =~ "Group Membership changed"
    | extend TargetGroup = coalesce(tostring(AdditionalFields["TO.GROUP"]),
tostring(AdditionalFields["Group.Name"]))
    | where TargetGroup has_any ("Domain Admins", "Enterprise Admins", "Administrators", "Remote Desktop Users")
    | extend CorrelationId = coalesce(tostring(AdditionalFields["TargetAccountSid"]), TargetAccountUpn)
    | where isnotempty(CorrelationId)
    | project EscalatedTime = Timestamp, CorrelationId, TargetGroup;
let xdr_deletes = IdentityDirectoryEvents
    | where Timestamp >= ago(xdr_lookback) and ActionType =~ "Account Deleted"
    | extend CorrelationId = coalesce(tostring(AdditionalFields["TargetAccountSid"]), TargetAccountUpn)
    | where isnotempty(CorrelationId)
    | project DeletedTime = Timestamp, CorrelationId;
xdr_creates
| join kind=inner (xdr_deletes) on CorrelationId
| where DeletedTime > CreatedTime and (DeletedTime - CreatedTime) <= 72h
| join kind=inner (xdr_group_adds) on CorrelationId
| where EscalatedTime between (CreatedTime .. DeletedTime)
| project CreatedTime, EscalatedTime, DeletedTime, Lifespan = DeletedTime - CreatedTime,
          CorrelationId, AccountUpn, AccountName, TargetGroup, DC
```

## sigma_staged_workflow.yml

SIGMA

```
title: Hidden Local Account Persistence via Winlogon SpecialAccounts UserList
id: 8d2f4e91-5c3a-4b10-9e77-204194532501
status: experimental
description: >
  Detects registry DWORD values set to 0 under Winlogon\SpecialAccounts\UserList,
  matching staged fin.ps1 / mkuser.py persistence (<HIDDEN_LOCAL_ADMIN>) designed to hide local
  accounts from interactive Windows sign-in surfaces. Prerequisite: requires Sysmon Event ID 13
  (RegistryEvent Value Set) supplying TargetObject and Details; native Windows Security Event 4657
  uses ObjectName, ObjectValueName, and NewValue (with a registry SACL) and requires separate mapping.
author: Odin by Brandefense
date: 2026-11-16
logsource:
  category: registry_set
  product: windows
detection:
  selection:
    TargetObject|contains: '\CurrentVersion\Winlogon\SpecialAccounts\UserList\'
    Details: 'DWORD (0x00000000)'
  condition: selection
falsepositives:
  - Rare legacy OEM provisioning utilities
level: high
tags:
  - attack.persistence
  - attack.defense_evasion
  - attack.t1564.002
---
title: Active Directory Replication-Right Access (Event 4662) from Non-Allowlisted Principal
id: 9b4a71c2-8e11-44d2-a91c-204194532502
status: experimental
description: >
  Detects MS-DRSR directory replication extended-right checks (Event ID 4662) by principals
  outside an explicit Domain Controller and Entra Connect allowlist (such as temporary admin
  <TEMP_DOMAIN_ADMIN> in dcsync.py). Prerequisite: requires 'Audit Directory Service Access = Success' GPO
  and a SACL on the Domain Naming Context root auditing Replicating Directory Changes rights.
author: Odin by Brandefense
date: 2026-11-16
logsource:
  product: windows
  service: security
detection:
  selection:
    EventID: 4662
    AccessMask:
      - '0x100'
      - '0x00000100'
      - '256'
    Properties|contains:
      - '1131f6aa-9c07-11d1-f79f-00c04fc2dcd2'
      - '1131f6ad-9c07-11d1-f79f-00c04fc2dcd2'
      - '89e95b76-444d-4c62-991a-0facbeda640c'
  filter_authorized_dc_accounts:
    SubjectUserName:
      - 'DC01$'
      - 'DC02$'
  filter_entra_connect:
    SubjectUserName:
      - 'MSOL_0123456789ab'
      - 'AAD_9876543210fe'
  filter_local_system:
    SubjectLogonId: '0x3e7'
  condition: selection and not filter_authorized_dc_accounts and not filter_entra_connect and not
filter_local_system
falsepositives:
  - Populate filter_authorized_dc_accounts with actual Domain Controller computer accounts and authorized gMSA
provisioning principals (e.g., provAgentgMSA$), and replace the placeholder SubjectUserName values in
filter_entra_connect with the tenant's exact approved Entra Connect sync account sAMAccountName values (never
exclude wildcard '*$', 'MSOL_*', or 'AAD_*' prefixes, or compromised member-server and user accounts matching
```

```
  those prefixes will evade detection)
  level: critical
  tags:
    - attack.credential_access
    - attack.t1003.006
  ---
  title: Web Server Worker Spawning Chunked PowerShell or Command Stager
  id: 7c19b3a4-2f08-4e61-b8a2-204194532503
  status: experimental
  description: >
    Detects IIS (w3wp.exe) or Apache Tomcat (tomcat*.exe / java.exe) spawning cmd.exe,
    PowerShell, or certutil with encoded, bypass, or chunked file-reassembly arguments
    matching marked staging commands from psrun.py, psbg.py, push.py, and dl.py. For
    unmarked cmd.exe /c execution via tjsp.jsp, correlate with Suricata wire rule
    sid:2026091109 (JSPOK|7C| response prefix) and YARA rule ODIN_R01_JSP_Cmd_Webshell_Markers.
  author: Odin by Brandefense
  date: 2026-11-16
  logsource:
    category: process_creation
    product: windows
  detection:
    selection_parent:
      ParentImage|endswith:
        - '\w3wp.exe'
        - '\tomcat8.exe'
        - '\tomcat9.exe'
        - '\tomcat10.exe'
        - '\tomcat11.exe'
        - '\java.exe'
    selection_child:
      Image|endswith:
        - '\cmd.exe'
        - '\powershell.exe'
        - '\pwsh.exe'
        - '\certutil.exe'
    selection_cli:
      CommandLine|contains:
        - '-enc'
        - 'ExecutionPolicy'
        - 'Bypass'
        - 'FromBase64String'
        - 'Set-Content'
        - 'WriteAllText'
        - '_pushp'
        - 'PSBG_DONE'
        - '-decode'
        - '-urlcache'
    condition: selection_parent and selection_child and selection_cli
  falsepositives:
    - CI/CD build agents running under java.exe that invoke PowerShell with ExecutionPolicy Bypass
  level: high
  tags:
    - attack.persistence
    - attack.execution
    - attack.t1505.003
    - attack.t1059.001
    - attack.t1059.003
```

## sql_linked_server_and_event15457_audit.sql

SQL

```
-- Part A: Audit linked-server configurations that PERMIT RPC-Out (EXEC AT) or Data Access (OPENQUERY) traversal;
it lists configuration only and does not detect traversal use
-- Permissions: Default public mappings (local_principal_id = 0) are visible with VIEW ANY DEFINITION (SQL Server
2019 and earlier) or VIEW SERVER SECURITY STATE (SQL Server 2022+), but restricted linked servers and other
logins' sys.linked_logins rows are silently omitted unless the caller holds ALTER ANY LINKED SERVER and ALTER ANY
LOGIN ON SERVER (or setupadmin / sysadmin).
SELECT
    s.server_id,
```

```
    s.name AS linked_server,
    s.product,
    s.provider,
    s.data_source,
    s.is_rpc_out_enabled,
    s.is_data_access_enabled,
    SUSER_NAME(ll.local_principal_id) AS local_login,
    ll.uses_self_credential,
    ll.remote_name AS mapped_remote_login,
    s.modify_date
FROM sys.servers AS s
LEFT JOIN sys.linked_logins AS ll
    ON s.server_id = ll.server_id
WHERE s.is_linked = 1
  AND (s.is_rpc_out_enabled = 1 OR s.is_data_access_enabled = 1)
ORDER BY s.is_rpc_out_enabled DESC, s.modify_date DESC;

-- Part B: Event 15457 configured-value (0 -> 1 and 1 -> 0) change sequence across ErrorLogs 0..6 vs. current
sys.configurations (value and value_in_use)
-- Note: Event 15457 records a configured-value change ("Run the RECONFIGURE statement to install") and does not
by itself prove RECONFIGURE activated value_in_use = 1; correlate with SQL Server Audit or endpoint process
telemetry (sqlservr.exe -> cmd.exe) to confirm effective runtime activation.
-- Permissions: sysadmin, or securityadmin for sys.xp_readerrorlog plus (SQL Server 2022+ non-sysadmin) VIEW
SERVER PERFORMANCE STATE to read sys.configurations
DECLARE @max_logs INT = 6;
DECLARE @log_idx INT = 0;
DECLARE @errlog TABLE (
    LogArchive INT NOT NULL,
    LogDate DATETIME NULL,
    ProcessInfo NVARCHAR(128) NULL,
    LogText NVARCHAR(4000) NULL
);
DECLARE @stage TABLE (
    LogDate DATETIME NULL,
    ProcessInfo NVARCHAR(128) NULL,
    LogText NVARCHAR(4000) NULL
);

WHILE @log_idx <= @max_logs
BEGIN
    BEGIN TRY
        DELETE FROM @stage;
        INSERT INTO @stage (LogDate, ProcessInfo, LogText)
        EXEC sys.xp_readerrorlog @log_idx, 1, N'xp_cmdshell';
        INSERT INTO @errlog SELECT @log_idx, LogDate, ProcessInfo, LogText FROM @stage;

        DELETE FROM @stage;
        INSERT INTO @stage (LogDate, ProcessInfo, LogText)
        EXEC sys.xp_readerrorlog @log_idx, 1, N'show advanced options';
        INSERT INTO @errlog SELECT @log_idx, LogDate, ProcessInfo, LogText FROM @stage;
    END TRY
    BEGIN CATCH
        BREAK;
    END CATCH;
    SET @log_idx = @log_idx + 1;
END;

SELECT
    e.LogDate,
    e.LogArchive,
    e.ProcessInfo,
    c.configured_value AS current_configured_value,
    c.value_in_use AS current_value_in_use,
    CASE
        WHEN c.value_in_use = 0
         AND (e.LogText LIKE N'%Configuration option ''xp_cmdshell'' changed from 0 to 1%' OR e.LogText LIKE
N'%''xp_cmdshell''%0%1%RECONFIGURE%')
         AND EXISTS (
             SELECT 1 FROM @errlog AS d
             WHERE (d.LogText LIKE N'%Configuration option ''xp_cmdshell'' changed from 1 to 0%' OR d.LogText
LIKE N'%''xp_cmdshell''%1%0%RECONFIGURE%')
               AND d.LogDate >= e.LogDate
```

```
         )
        THEN N'HISTORICAL_CONFIG_CHANGE_SEQUENCE_WHILE_CURRENTLY_DISABLED'
        WHEN c.value_in_use = 0
         AND (e.LogText LIKE N'%Configuration option ''xp_cmdshell'' changed from 0 to 1%' OR e.LogText LIKE
N'%''xp_cmdshell''%0%1%RECONFIGURE%')
        THEN N'HISTORICAL_CONFIG_0_TO_1_WHILE_CURRENTLY_DISABLED'
        WHEN c.value_in_use = 0
         AND (e.LogText LIKE N'%Configuration option ''xp_cmdshell'' changed from 1 to 0%' OR e.LogText LIKE
N'%''xp_cmdshell''%1%0%RECONFIGURE%')
        THEN N'HISTORICAL_CONFIG_1_TO_0_CHANGE_EVENT'
        WHEN c.value_in_use = 1
         AND (e.LogText LIKE N'%Configuration option ''xp_cmdshell'' changed from 0 to 1%' OR e.LogText LIKE
N'%''xp_cmdshell''%0%1%RECONFIGURE%')
        THEN N'CONFIG_0_TO_1_EVENT_WHILE_CURRENTLY_IN_USE'
        ELSE N'SHOW_ADVANCED_OPTIONS_CONFIG_EVENT_15457'
    END AS audit_classification,
    e.LogText
FROM @errlog AS e
CROSS JOIN (
    SELECT
        CAST(value AS INT) AS configured_value,
        CAST(value_in_use AS INT) AS value_in_use
    FROM sys.configurations
    WHERE name = N'xp_cmdshell'
) AS c
WHERE e.LogText LIKE N'%''xp_cmdshell''%0%1%RECONFIGURE%'
   OR e.LogText LIKE N'%''xp_cmdshell''%1%0%RECONFIGURE%'
   OR e.LogText LIKE N'%''show advanced options''%[01]%[01]%RECONFIGURE%'
ORDER BY e.LogDate ASC;
```

## suricata_staging_and_ldap_bind.rules

SURICATA

```
alert tcp $HOME_NET any -> $HOME_NET 389 (msg:"ODIN-R01 LDAPv3 Cleartext SimpleBind Non-Empty Password - Short-
Form BER (ldapv2.py)"; flow:established,to_server; content:"|30|"; content:"|60|"; distance:3; within:12;
content:"|02 01 03 04|"; distance:1; within:9; fast_pattern; byte_test:1,<,128,0,relative; byte_jump:
1,0,relative; content:"|80|"; distance:0; within:1; byte_test:1,!=,0,0,relative; byte_test:1,!=,128,0,relative;
content:!"|84 00 00 00 00|"; distance:0; within:5; threshold:type limit, track by_src, count 1, seconds 3600;
classtype:policy-violation; sid:2026091101; rev:2; metadata:created_at 2026_11_16, confidence medium,
mitre_technique_id T1078.002;)
alert tcp $HOME_NET any -> $HOME_NET 389 (msg:"ODIN-R01 LDAPv3 Cleartext SimpleBind Non-Empty Password - Definite
BER 0x81 Long DN"; flow:established,to_server; content:"|30|"; content:"|60|"; distance:3; within:12; content:"|
02 01 03 04 81|"; distance:1; within:10; fast_pattern; byte_jump:1,0,relative; content:"|80|"; distance:0;
within:1; byte_test:1,!=,0,0,relative; byte_test:1,!=,128,0,relative; content:!"|84 00 00 00 00|"; distance:0;
within:5; threshold:type limit, track by_src, count 1, seconds 3600; classtype:policy-violation; sid:2026091102;
rev:2; metadata:created_at 2026_11_16, confidence medium, mitre_technique_id T1078.002;)
alert tcp $HOME_NET any -> $HOME_NET 389 (msg:"ODIN-R01 LDAPv3 Cleartext SimpleBind Non-Empty Password - Definite
BER 0x82 Long DN"; flow:established,to_server; content:"|30|"; content:"|60|"; distance:3; within:12; content:"|
02 01 03 04 82|"; distance:1; within:10; fast_pattern; byte_jump:2,0,relative,big; content:"|80|"; distance:0;
within:1; byte_test:1,!=,0,0,relative; byte_test:1,!=,128,0,relative; content:!"|84 00 00 00 00|"; distance:0;
within:5; threshold:type limit, track by_src, count 1, seconds 3600; classtype:policy-violation; sid:2026091103;
rev:1; metadata:created_at 2026_11_16, confidence medium, mitre_technique_id T1078.002;)
alert tcp $HOME_NET any -> $HOME_NET 389 (msg:"ODIN-R01 LDAPv3 Cleartext SimpleBind Non-Empty Password - Windows
wldap32 0x84 BER"; flow:established,to_server; content:"|30|"; content:"|60 84|"; distance:3; within:13;
content:"|02 01 03 04 84|"; distance:4; within:9; fast_pattern; byte_jump:4,0,relative,big; content:"|80|";
distance:0; within:1; byte_test:1,!=,0,0,relative; byte_test:1,!=,128,0,relative; content:!"|84 00 00 00 00|";
distance:0; within:5; threshold:type limit, track by_src, count 1, seconds 3600; classtype:policy-violation; sid:
2026091104; rev:1; metadata:created_at 2026_11_16, confidence medium, mitre_technique_id T1078.002;)
alert http any any -> $HOME_NET any (msg:"ODIN-R01 HTTP POST Chunked Webshell PowerShell Staging Marker
(_pushp)"; flow:established,to_server; http.request_body; content:"_pushp"; fast_pattern; classtype:web-
application-attack; sid:2026091105; rev:1; metadata:created_at 2026_11_16, confidence high, mitre_technique_id
T1505.003;)
alert http any any -> $HOME_NET any (msg:"ODIN-R01 HTTP GET Fallback Chunked Webshell Staging Marker (_pushp in
URI)"; flow:established,to_server; http.uri; content:"_pushp"; fast_pattern; classtype:web-application-attack;
sid:2026091106; rev:1; metadata:created_at 2026_11_16, confidence high, mitre_technique_id T1505.003;)
alert http $HOME_NET any -> any any (msg:"ODIN-R01 Webshell Execution Completion Marker in HTTP Response
(PSBG_DONE)"; flow:established,to_client; http.response_body; content:"PSBG_DONE"; fast_pattern; classtype:web-
application-attack; sid:2026091107; rev:2; metadata:created_at 2026_11_16, confidence high, mitre_technique_id
T1505.003;)
```

```
alert http $HOME_NET any -> $HOME_NET any (msg:"ODIN-R01 Internal HTTP/Decrypted-TLS JNDI Lookup Probe in
Header"; flow:established,to_server; http.header; content:"${jndi:"; fast_pattern; nocase; pcre:"/\$\{jndi:
(?:ldap|ldaps|rmi|dns):\/\//i"; threshold:type limit, track by_src, count 1, seconds 600; classtype:attempted-
admin; sid:2026091108; rev:2; metadata:created_at 2026_11_16, confidence medium, mitre_technique_id T1190;)
alert http $HOME_NET any -> any any (msg:"ODIN-R01 JSP Command Webshell Output Prefix in HTTP Response (JSPOK|)";
flow:established,to_client; http.response_body; content:"JSPOK|7C|"; fast_pattern; classtype:web-application-
attack; sid:2026091109; rev:2; metadata:created_at 2026_11_18, confidence high, mitre_technique_id T1505.003;)
alert http $HOME_NET any -> $HOME_NET any (msg:"ODIN-R01 Internal HTTP/Decrypted-TLS JNDI Lookup Probe in Request
Body (log4test.py)"; flow:established,to_server; http.request_body; content:"${jndi:"; fast_pattern; nocase;
pcre:"/\$\{jndi:(?:ldap|ldaps|rmi|dns):\/\//i"; threshold:type limit, track by_src, count 1, seconds 600;
classtype:attempted-admin; sid:2026091110; rev:1; metadata:created_at 2026_11_18, confidence medium,
mitre_technique_id T1190;)
```

## yara_webroot_and_persistence.yar

YARA

```
/*
   Odin by Brandefense - Report #01 Host & Webroot YARA Suite (v1.0)
   Targets residual on-disk artifacts modeled in Chapter 5:
     1) Chunked Base64 PowerShell staging markers (_pushp / PSBG_DONE)
     2) Minimal plaintext JSP command webshells matching tjsp.jsp (JSPOK|); Base64-staged files must be decoded
before scanning
     3) PowerShell SpecialAccounts\UserList + Advapi32!LogonUser hidden persistence (fin.ps1)
*/

rule ODIN_R01_Webroot_Chunked_PS1_Stager {
    meta:
        author = "Odin by Brandefense"
        date = "2026-11-16"
        version = "1.0"
        tlp = "CLEAR"
        description = "Detects _pushp chunked Base64 staging scripts or output markers in web upload folders"
    strings:
        $m1 = "_pushp" ascii wide
        $m2 = "PSBG_DONE" ascii wide
        $m3 = "FromBase64String" ascii wide nocase
        $m4 = "WriteAllText" ascii wide nocase
        $m5 = "Set-Content" ascii wide nocase
    condition:
        filesize < 64KB and (($m1 and ($m2 or $m3 or $m4 or $m5)) or ($m2 and $m3))
}

rule ODIN_R01_JSP_Cmd_Webshell_Markers {
    meta:
        author = "Odin by Brandefense"
        date = "2026-11-16"
        version = "1.0"
        tlp = "CLEAR"
        description = "Detects minimal plaintext JSP cmd.exe webshells matching tjsp.jsp (JSPOK|); Base64-staged
files require decoding before scanning"
    strings:
        $j1 = "JSPOK|" ascii wide
        $j2 = "Runtime.getRuntime().exec" ascii wide
        $j3 = "ProcessBuilder" ascii wide
        $j4 = "cmd.exe /c" ascii wide nocase
        $j5 = "request.getParameter" ascii wide
    condition:
        filesize < 64KB and ($j1 or ($j4 and $j5 and ($j2 or $j3)))
}

rule ODIN_R01_PS1_SpecialAccounts_Hidden_Admin {
    meta:
        author = "Odin by Brandefense"
        date = "2026-11-16"
        version = "1.0"
        tlp = "CLEAR"
        description = "Detects PowerShell stealth local admin scripts hiding accounts via Winlogon
SpecialAccounts\\UserList and testing tokens via LogonUser (fin.ps1)"
    strings:
```

```
        $s1 = "Winlogon\\SpecialAccounts\\UserList" ascii wide nocase
        $s2 = "advapi32.dll" ascii wide nocase
        $s3 = "LogonUser" ascii wide
        $s4 = "fDenyTSConnections" ascii wide nocase
    condition:
        filesize < 64KB and $s1 and (($s2 and $s3) or $s4)
}
```

ODIN

# About Odin Research

Odin Research is the public research series of Odin by Brandefense. It documents adversary infrastructure seen on the internet and turns each finding into indicators and detection rules that defenders can use. Victims and third parties stay anonymous, addresses appear only for confirmed adversary infrastructure, and no working attack code is published.

Odin is the agentic analyst inside the Brandefense threat intelligence platform, which unifies Cyber Threat Intelligence, Digital Risk Protection and External Attack Surface Management.

Online edition, indicators and rules · brandefense.io/odin-research/reports/staged-domain-and-hypervisor-intrusion/
Short Read · brandefense.io/odin-research/blog/an-exposed-staging-server-shows-a-web-to-domain/
Defense Kit · brandefense.io/odin-research/defense-kit/