ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 07 / 30 · POST-EXPLOITATION

Exposed Sliver Workbenches

PHP Exploit Clones, PAM Backdoors and GET Exfiltration

What 600 Sliver endpoints and four open operator servers show about modern post-exploitation kits

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

600
Sliver-associated endpoints last seen 13 to 15 November 2026, on 452 IPv4 hosts in 41 countries

36
Confirmed Sliver listeners, on 34 IP addresses; 64 more are suspected and 500 are candidates

73.8%
Share of the 600 endpoints (443) on four ports: 443, 8888, 31337 and 8443

3,760
Files listed across four open operator servers (3,171 unique paths)

What Is in This Report. 12 chapters and two appendices.

What 600 Sliver endpoints and four open operator servers show about modern post-exploitation kits

01	Executive Summary	04
02	The Sliver Footprint	07
03	Four Open Operator Servers	11
04	PHP2Shell on a Live C2 Server	13
05	Linux PAM Backdoor, Systemd Loop and Task Queue	16
06	Exposed Teamserver Logs and GET Exfiltration	20
07	Telling Real Sliver Servers from Look-Alikes	23
08	What Defenders Should Do	26
09	Detection Rules	28
10	Indicators	32
11	Scope and Limits	35
12	Sources	37
A	Indicators of compromise (22)	39
B	Detection rules (4 files)	41

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense studied 600 Sliver-associated endpoints and four open operator servers. Only 36 endpoints are confirmed Sliver listeners. On those servers, operators keep cloned PHP exploit labs, a Linux PAM backdoor, a disguised systemd loop and a script that leaks data through HTTP GET requests. Patch PHP to a fixed release, watch PAM and systemd changes, block direct workstation HTTP egress, and block only listeners with a Sliver-specific signal.

01

CHAPTER 01

Executive Summary

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense studied 600 Sliver-associated endpoints and four open operator servers. Only 36 endpoints are confirmed Sliver listeners. On those servers, operators keep cloned PHP exploit labs, a Linux PAM backdoor, a disguised systemd loop and a script that leaks data through HTTP GET requests. Patch PHP to a fixed release, watch PAM and systemd changes, block direct workstation HTTP egress, and block only listeners with a Sliver-specific signal.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 04

01 Executive Summary

Sliver is an open-source command-and-control (C2) framework from BishopFox. Red teams use it for authorized tests, and intruders use it too. This report looks at where Sliver listeners run and at what four careless operators left in open web folders next to them.

1. **One confirmed Sliver server also holds PHP exploit clones and Linux persistence.**
   On 31.56.228[.]136 in France, three public PHP2Shell repositories sit next to a Sliver listener on port 31337. They target CVE-2026-109618 and CVE-2026-109619. The same server stages a PAM password backdoor, a disguised systemd service and a file-based task queue.

2. **Open folders leak Sliver audit logs and a GET exfiltration script.**
   Three more servers expose Sliver databases, TLS keys and operator logs over plain HTTP. One of them, 154.8.211[.]145, stages PrintSpoofer and a PowerShell script called `x.ps1`. The script cuts Base64 data into HTTP GET query strings of up to 2,000 characters to slip past POST-only inspection.

3. **Most Sliver look-alikes are not Sliver.**
   Of 600 endpoints, 36 are confirmed, 64 suspected and 500 candidates. Generic `CN=localhost` certificates and an auto-generated MySQL certificate resemble Sliver but are not proof. Git logs also show the PHP exploits were cloned from public repositories, not written by the operator.

Three Decisions for Leaders

1. **Patch PHP everywhere it faces the internet.** Upgrade to PHP 8.2.35, 8.3.36, 8.4.27 or 8.5.12. Turning off session upload progress or moving to PHP-FPM only reduces exposure; it does not fix the flaws.
2. **Watch Linux login plumbing and close HTTP side doors.** Monitor PAM modules and systemd units, and alert when `sshd` or `sshd-session` starts a shell script. Block direct HTTP egress from workstations and log long Base64 values in GET query strings.
3. **Block only on Sliver-specific evidence.** Do not auto-block on a localhost certificate, a MySQL, Redis or FTP port, or a training-tool host. Check Git history before calling cloned code a zero-day.

The Study in Numbers

Number	What it is
600	Sliver-associated endpoints last seen 13 to 15 November 2026, on 452 IPv4 hosts in 41 countries
36	Confirmed Sliver listeners, on 34 IP addresses; 64 more are suspected and 500 are candidates
73.8%	Share of the 600 endpoints (443) on four ports: 443, 8888, 31337 and 8443
3,760	Files listed across four open operator servers (3,171 unique paths)
2,000	Maximum Base64 characters per HTTP GET request in the staged `x.ps1` script
22	Published indicators: 15 confirmed C2 IPs and 7 file hashes (selected set of 28)

How to use this report.

Security leaders can read Chapter 1 and Chapter 8. Platform and Linux engineers should read Chapters 4, 5 and 8. SOC and detection teams should start with Chapters 7, 9 and 10. CTI analysts will find the infrastructure detail in Chapters 2 and 3.

02

CHAPTER 02

The Sliver Footprint

IN SHORT

The 600 endpoints spread across ordinary commercial clouds, not one bulletproof host. Four ports carry 73.8% of them, and only 36 show a Sliver-specific signal.

INSIDE THIS CHAPTER

What Sliver is 07

The set of endpoints in this report 07

Ports 08

Countries and hosting 08

Confirmed, suspected and candidate endpoints 08

02 The Sliver Footprint

What Sliver is

Sliver gives each implant its own keys. It supports mutual TLS (mTLS, where both sides present certificates), HTTP(S), WireGuard and DNS transports. Operators connect to the server through a "multiplayer" console on port 31337 by default.

The set of endpoints in this report

This report covers 600 Sliver-associated endpoints last seen between 13 and 15 November 2026. They sit on 452 IPv4 addresses, in 41 countries and 156 Autonomous Systems (144 hosting organizations). All 36 confirmed and all 64 suspected endpoints are included, together with 500 candidates.

These 600 are drawn from a wider set of 5,471 Sliver-linked host records across 82 countries. That wider set carries upstream history from earlier public records. It is not the span of our own observation. All percentages in this report describe the 600 endpoints only.



Figure 1: Four TCP Ports Carry 73.8% of Observed Sliver Endpoints Across Commercial Clouds

Ports

Four ports carry 443 of the 600 endpoints (73.8%). Another 42 (7.0%) use four secondary ports.

Port	Endpoints	Share	Role in Sliver deployments
443	167	27.8%	HTTPS or mTLS implant listener that blends into web traffic
8888	128	21.3%	Default Sliver mTLS implant listener when left unchanged
31337	95	15.8%	Default multiplayer operator console and mTLS port
8443	53	8.8%	Alternate HTTPS or mTLS listener
80, 9443, 8080, 24245	42	7.0%	HTTP staging (19 on 80, 8 on 8080) and custom TLS (9 on 9443, 6 on 24245); all are candidates

Countries and hosting

The endpoints concentrate in the United States (155), China (81), Hong Kong (74), Germany (46), Singapore (35), the Netherlands (26) and Japan (26). They spread across mainstream cloud providers.

Hosting organization	Endpoints	Share	ASN
DigitalOcean, LLC	45	7.5%	AS14061
The Constant Company, LLC	39	6.5%	AS20473
Shenzhen Tencent Computer Systems	34	5.7%	17 on AS45090, 17 on AS132203
Amazon.com, Inc.	33	5.5%	AS16509, AS14618
Hangzhou Alibaba Advertising	33	5.5%	AS37963
OVH SAS	19	3.2%	AS16276
Alibaba (US) Technology	17	2.8%	AS45102
Hetzner Online GmbH	16	2.7%	AS24940, AS212317

Confirmed, suspected and candidate endpoints

Suspected and candidate endpoints stay on watchlists, because many ordinary web servers share the TLS patterns Sliver often uses.

Group	Endpoints
Confirmed	36
Suspected	64
Candidate	500

Sliver ships public defaults that make good hunting pivots: the `CN=multiplayer` operator certificate on port 31337 and a default mTLS JARM (`000000...4c1b01`) on port 8888.

Two IP addresses listen on both 31337 and 8888, which is why 36 confirmed endpoints sit on 34 IPs. The confirmed group splits into 23 mTLS listeners and 13 HTTP(S) C2 profiles.

Across the 100 confirmed and suspected endpoints, 89 present a certificate common name. These are `localhost` (52), `multiplayer` (20), `server.example[.]com` (2) and 15 custom or spoofed names. Ninety carry a JARM value. Generic Go and HTTPS stacks share many JARM values with developer servers, so JARM alone does not confirm Sliver.

03

CHAPTER 03

Four Open Operator Servers

IN SHORT

Operators often start a quick Python web server in their home folder to move tools. When they do, the whole Sliver state folder becomes browsable. Four such servers list 3,760 files.

03 Four Open Operator Servers

The four servers below exposed 3,760 file entries across seven HTTP listeners. Some files appear on two ports, so the unique count is 3,171. On 31.56.228[.]136, ports 8899 and 9999 mirror each other. On 154.8.211[.]145, ports 8088 and 9002 serve subfolders of 8899.

Server	Open HTTP and C2 ports	Files (unique)	Country and network	Main exposed items
US workbench (address withheld)	8000	1,417 (1,417)	US, NetLab Global (AS979)	Sliver SQLite database, `/wp/_sliver_db.py`, TLS key bundles, `sliver-client.log`, Sliver `Go 1.27.3` source tree
31.56.228[.]136	8899, 9999; Sliver on 31337	1,124 (562)	FR, AS56971 Cloud	Three cloned PHP2Shell repositories, `toomanysecrets.sh`, `systemd-resolver-helper.service`, `/c2/2b17bb46.queue`
Candidate host (withheld)	Withheld	850 (850)	FI, European cloud network	A staged implant binary next to an OSEP training builder
154.8.211[.]145	8088, 8899, 9002; Sliver on 31337	369 (342)	CN, Shenzhen Tencent (AS45090)	`/.sliver/logs/audit.json`, `console.log`, `sliver.json`, PrintSpoofer, `x.ps1`

The Finnish host is withheld. Its builder carries `Author: OSEP training tool` metadata, and its Sliver port is unconfirmed. It may belong to a student or an authorized tester.

The GitHub account names in this report (`slabbdev`, `murrez`, `Zer061` and `9bie`) belong to public repository authors. Their code was cloned. Nothing links them to the operators of these servers.

04

CHAPTER 04

PHP2Shell on a Live C2 Server

IN SHORT

The operator of 31.56.228[.]136 cloned three public exploit repositories for two PHP flaws within four minutes. Only an upgrade to a fixed PHP release closes both flaws.

INSIDE THIS CHAPTER

What was on the server 13

How the two flaws work, in brief 13

Cloned, not written 14

What fixes it 14

04 PHP2Shell on a Live C2 Server

What was on the server

Ports 8899 and 9999 on 31.56.228[.]136 listed 1,124 file entries (562 unique paths, 86 directories). The same host runs a confirmed Sliver mTLS listener on `31337/tcp`. Three directories hold cloned PHP2Shell proof-of-concept workspaces.

Their root-cause note cites two flaws, both rated CVSS v4.0 9.2:

CVE	Advisory	Weakness	Fixed in
CVE-2026-109618	GHSA-w8hw-w9fx-j79w	CWE-755	PHP 8.2.35, 8.3.36, 8.4.27, 8.5.12
CVE-2026-109619	GHSA-6wmf-4rv5-p3rp	CWE-416, CWE-824	PHP 8.2.35, 8.3.36, 8.4.27, 8.5.12

Both were fixed together in `php-src` commit `506e87fa7dcdd306975e5b8f8265ee03527aacbf`. The fixed releases came out between 20 and 22 October 2026. The upstream note on the server lists only 8.3.36, 8.4.27 and 8.5.12. The official PHP advisories and ChangeLog also include 8.2.35.

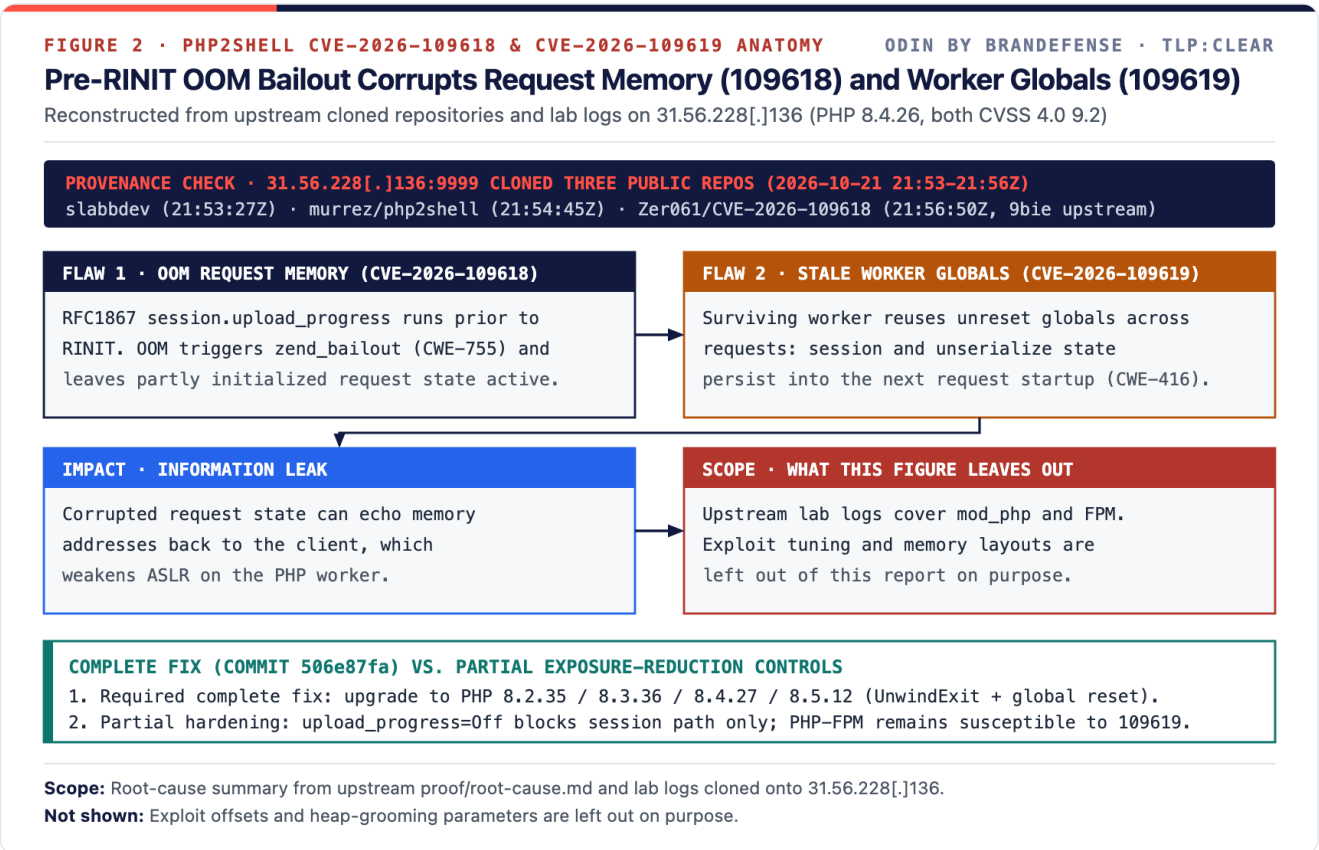


Figure 2: Pre-RINIT OOM Bailout Corrupts Request Memory (109618) and Worker Globals (109619)

How the two flaws work, in brief

This section stays at the design level. Exploit offsets, buffer sizes and heap-grooming steps are left out on purpose.

CVE-2026-109618. PHP parses multipart uploads before a request is fully set up when `session.upload_progress.enabled` is on. That is the default in standard `php.ini` templates. If the upload exhausts `memory_limit` at that stage, PHP aborts without cleanup. Partly set-up request memory is left behind.

CVE-2026-109619. A persistent worker, such as Apache `prefork` with `mod_php` or a reused PHP-FPM child, then serves the next request. Some globals were never reset. Session and unserialize state from the failed request carry into the next one.

> **What defenders can see on the wire.**
>
> In the upstream lab logs, corrupted session data returned memory addresses inside serialized HTTP response bodies. Suricata SID 9702602 looks for that response pattern.

Cloned, not written

The Git logs (`.git/logs/HEAD`) in all three directories show a single clone entry each, with no local commits. The clone timestamps on the server are all on 21 October 2026, the same days the fixed releases appeared.

Directory on the server	Reflog size	Git clone timestamp (UTC)	Upstream repository	Local commits
`/php2shell-cve-2026-109618-109619`	210 B	21 Oct 2026, 21:53:27	`slabbdev/php2shell-cve-2026-109618-109619`	0
`/php2shell`	185 B	21 Oct 2026, 21:54:45	`murrez/php2shell`	0
`/CVE-2026-109618`	191 B	21 Oct 2026, 21:56:50	`Zer061/CVE-2026-109618` , tracking `9bie`	0

So the research notebooks on the server came from the public repositories. These include `ASLR_PATHS.md` (21,811 bytes) and lab logs for PHP 8.4.26 and PHP 8.5.4. Their hashes are listed as context only in the file table in Chapter 5. That keeps researchers who clone public advisories out of file-hash block lists.

`ASLR_PATHS.md` also explains why remote use against a default `php.ini` with ASLR on is constrained. Partial overwrites break adjacent address bytes. The default `session.serialize_handler=php` also rejects pipe characters in session keys, unless the application uses specific session objects or the `php_serialize` handler.

What fixes it

Upgrading to PHP 8.2.35, 8.3.36, 8.4.27 or 8.5.12 or later is the only complete fix. The patch replaces the abort with a clean unwind and resets globals.

Setting `session.upload_progress.enabled = Off` blocks only the session upload progress path. PHP-FPM is still exposed to known CVE-2026-109619 paths.

CHAPTER 05

Linux PAM Backdoor, Systemd Loop and Task Queue

IN SHORT

Next to the PHP labs and the Sliver listener, 31.56.228[.]136 stages a three-part Linux persistence kit. It survives password changes and restarts without needing a Sliver implant.

INSIDE THIS CHAPTER

The PAM backdoor 16

The disguised systemd service 17

The file-based task queue 17

Files from 31.56.228[.]136:9999 18

05 Linux PAM Backdoor, Systemd Loop and Task Queue

A Sliver implant can be quarantined by an endpoint agent. The operator here also staged light shell and service files that do not depend on it. Test strings such as `C2_ALIVE` and output from `uid=997` fit either an intrusion staging host or an adversary-emulation lab.

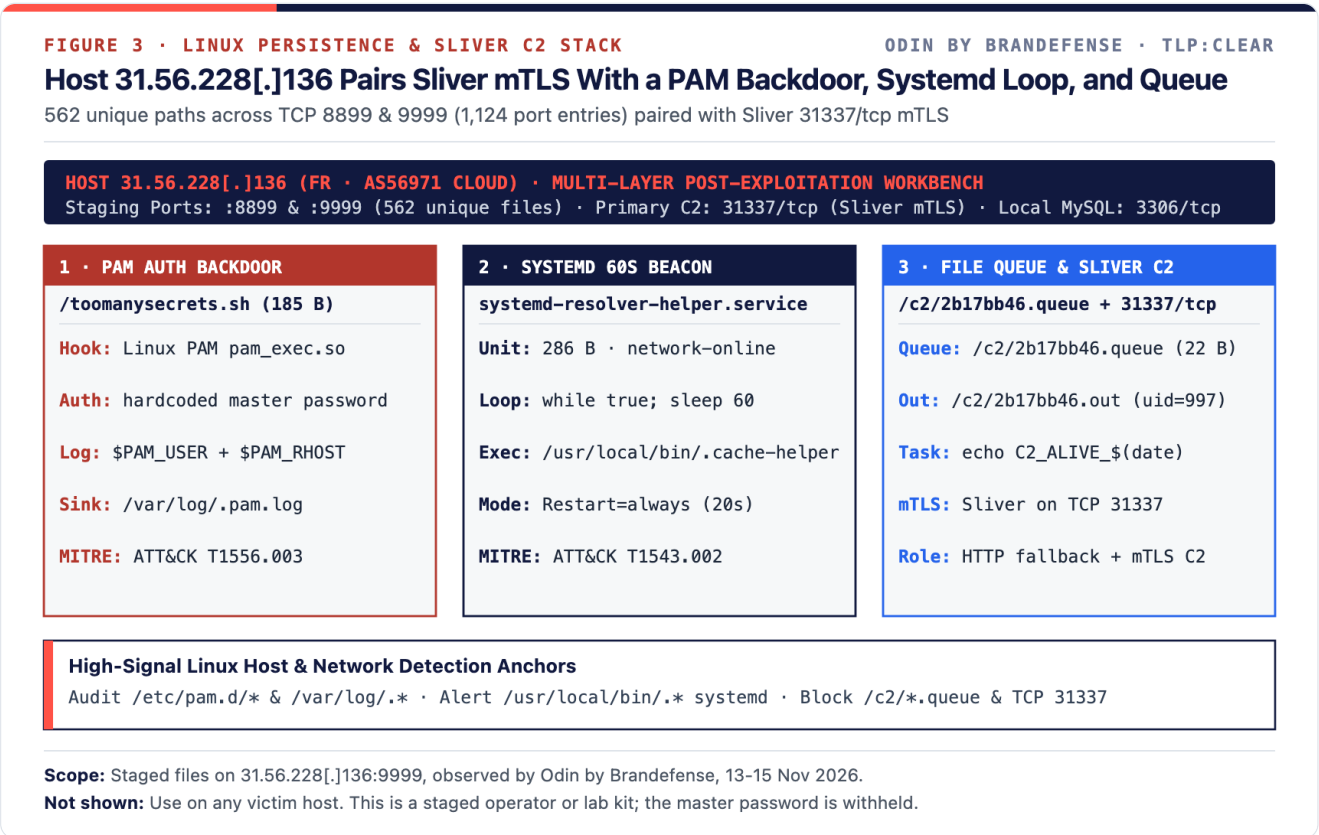


Figure 3: Host 31.56.228[.]136 Pairs Sliver mTLS With a PAM Backdoor, Systemd Loop, and Queue

The PAM backdoor

PAM (Pluggable Authentication Modules) decides whether a Linux login succeeds. `/toomanysecrets.sh` is a 185-byte shell script meant to run through `pam_exec.so` with `expose_authtok` (MITRE ATT&CK T1556.003). It would be added to `/etc/pam.d/common-auth` or `/etc/pam.d/sshd` as a `sufficient` module.

Function	What it does
Master password	Compares the submitted password with a hardcoded operator secret. On a match it grants `sshd`, `sudo` or `su` access for any local account, without touching `/etc/shadow`. The secret is withheld.
Credential harvest	For every other login, it writes the user (`$PAM_USER`), source host (`$PAM_RHOST`) and plaintext password to `/var/log/.pam.log`, then lets normal `pam_unix.so` checks continue.

The script body is not published. The SHA-256 is in the file table at the end of this chapter.

Why `sshd-session` matters. OpenSSH 9.8 moved session handling out of `sshd`. On Ubuntu 24.04 and later (including 26.04), Debian 13 and RHEL 10, `pam_exec.so` hooks start under `sshd-session`. Process rules must watch both `sshd` and `sshd-session` starting `.sh` files or scripts in `/tmp/`.

Neither `pam_exec.so` nor `/var/log/.pam.log` shows up in the process command line. So pair process rules with file-integrity alerts on `/etc/pam.d/*`, `/lib*/security/*` and `/var/log/.*`.

The disguised systemd service

`/systemd-resolver-helper.service` (286 bytes, 13 lines) keeps a callback alive without any SSH login. Its name and `Description=Systemd Resolver Helper` mimic the real `systemd-resolved` (T1543.002, T1036.004).

Setting	Value
Start order	`After=network-online.target`, under `multi-user.target`
Restart	`Restart=always`, `RestartSec=20`
Action	A `/bin/bash -c` loop that runs every 60 seconds
Payload	A hidden dot-prefixed binary, `/usr/local/bin/.cache-helper`

The file-based task queue

The `/c2/` folder holds a minimal HTTP polling queue. The task file `/c2/2b17bb46.queue` (22 bytes) holds `echo C2_ALIVE_$(date)`. The output file `/c2/2b17bb46.out` (97 bytes) holds Base64 and plaintext output from a Linux service account (`uid=997 gid=997 groups=997`). That account may belong to a target or a lab.

Files from 31.56.228[.]136:9999

Path on the server	Size	SHA-256	Role
`/toomanysecrets.sh`	185 B	`112967dcd581e2cc9b9afd6270321ddbdcb1b984cf0e1198c081e19c6f3968d1`	PAM master password and credential logger
`/systemd-resolver-helper.service`	286 B	`0df9cb861c20639c86d8edb7336b5af7a190093d842b3b0587704d897ebf72d3`	60-second loop running `/usr/local/bin/.cache-helper`
`/c2/2b17bb46.queue`	22 B	`29c7b6d0b65c9c1d623e0fc7290b044b371d2110e68c0e2e813f8e3d8e668122`	HTTP task queue
`/c2/2b17bb46.out`	97 B	`319364ec66d66cb27f96526e28ea124a28a0e3a0e0bc9448c30c81809efff770`	Task output (`uid=997`)
`/php2shell-cve-2026-109618-109619/.git/logs/HEAD`	210 B	`e765c76646af9338a1b2a450ad10c61cdd51377e8af28d9f0a1c74557817166c`	Clone record, `slabbdev`
`/php2shell/.git/logs/HEAD`	185 B	`60d90245690fd23df57890899efd91f967f021da015d6a96d2aee826c60492fc`	Clone record, `murrez`
`/CVE-2026-109618/.git/logs/HEAD`	191 B	`011980676b827e6a69403c09375bd6d570e9ce024584a67a5ebd1fb57b2c0756`	Clone record, `Zer061`
`/CVE-2026-109618/ASLR_PATHS.md`	21,811 B	`e2936511010e01a6c86b61b93f9645950c2f7e70d23a8291b0d65894ac6bc729`	Upstream ASLR notebook (context only)
`/CVE-2026-109618/logs/aslr_evidence.md`	2,357 B	`f9bc698f961e402618a8a40ff137339e66f0f548a76a53a47f354c5b61bbdd2c`	Upstream lab log (context only)
`/CVE-2026-109618/logs/g18_ASLR_ZEND_103.md`	5,289 B	`232ddfbfc02ff178537e45d3682766a7c8563af9e0837424ad8035e3a3716607`	Upstream Apache 2.4.68 `prefork` lab log (context only)
`/php2shell-cve-2026-109618-109619/proof/root-cause.md`	1,288 B	`dc3afe545d17a12aefde0bffc3b1fd6aea5aac11c4e0da84ccd70f7e616fe743`	Upstream summary of commit `506e87fa` (context only)

CHAPTER 06

Exposed Teamserver Logs and GET Exfiltration

IN SHORT

Three more open servers show Sliver audit logs, databases, PrintSpoofer and a training builder. One stages a PowerShell script that hides data in HTTP GET query strings to avoid POST-only inspection.

INSIDE THIS CHAPTER

What each server exposed 20

Inside `x.ps1` 21

06 Exposed Teamserver Logs and GET Exfiltration

When operators serve their home folders, `~/.sliver/` and `~/.sliver-client/` become readable. These hold the server's internal state. Credentials, session cookies and script bodies are left out of this report.

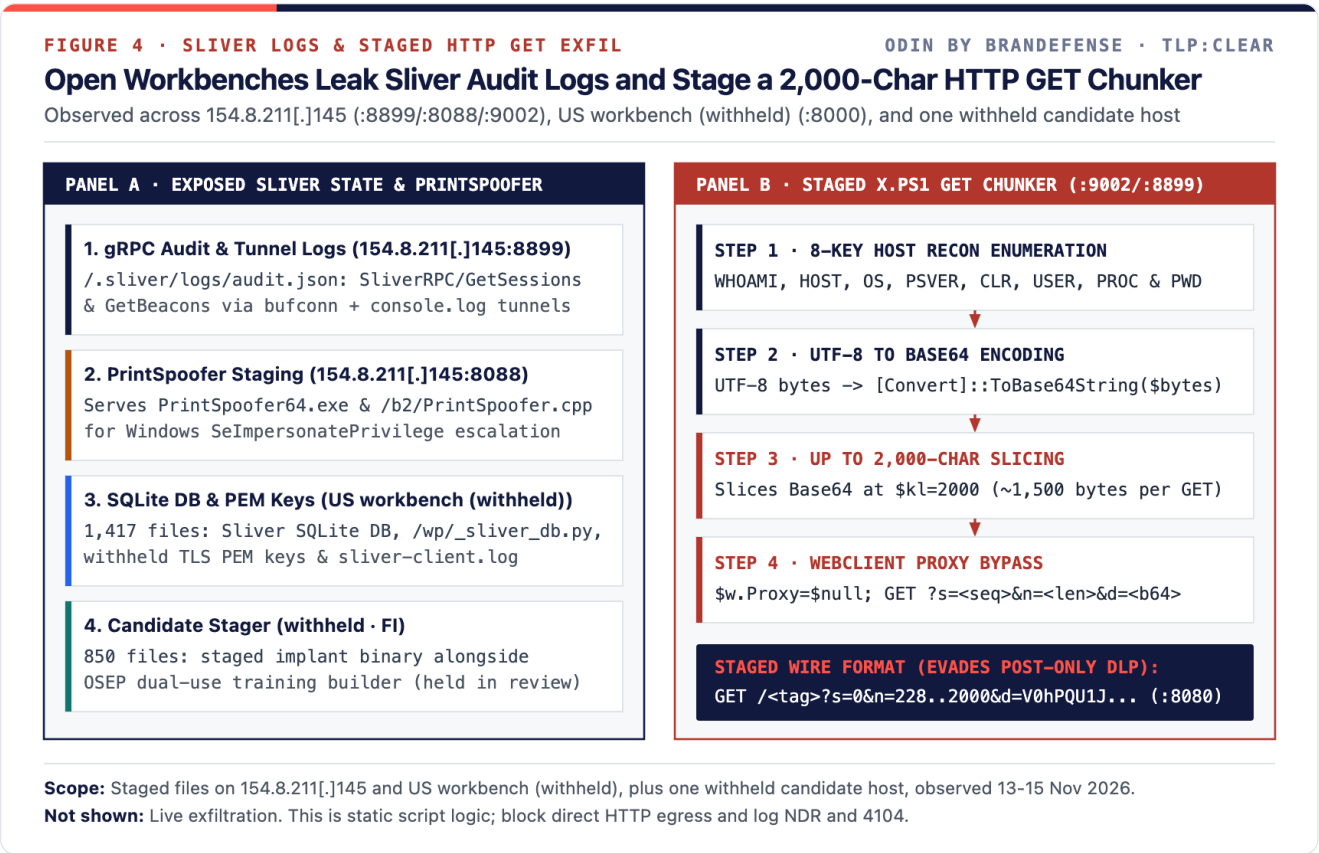


Figure 4: Open Workbenches Leak Sliver Audit Logs and Stage a 2,000-Char HTTP GET Chunker

What each server exposed

154.8.211[.]145 (China). This host runs a confirmed Sliver mTLS listener on `31337/tcp`. Port 8899 lists 342 files, port 8088 lists 26 files under `/ps`, and port 9002 serves one file under `/mlet_9002`. Together they hold teamserver logs, server configuration and traffic-encoder modules.

Port 8088 also stages `PrintSpoofer64.exe` and `/b2/PrintSpoofer.cpp`. PrintSpoofer abuses `SeImpersonatePrivilege` through the `\\pipe\spoolss` named pipe to escalate privileges on Windows (T1134.001). The server's `/httpd.log` records local `GET /PrintSpoofer64.exe` requests.

The US workbench (address withheld).

This NetLab Global server lists 1,417 files. They include a Sliver SQLite database with its `WAL` and `SHM` journals, five `/wp/_sliver_*.py` wrappers and a Sliver `Go 1.27.3` source tree. TLS certificate and private-key bundles are also exposed; their paths are withheld. `/.sliver-client/sliver-client.log` records operator client setup and tunnel activity.

Candidate host (Finland, withheld).

This server lists 850 files. It stages an implant binary next to a Python process-hollowing builder with OSEP training metadata. Its Sliver port is unconfirmed, so it stays in review and is not published.

Server and port	File	Size	What it shows
154.8.211[.]145:8899	`/.sliver/logs/audit.json`	821 B	Operator gRPC calls over `bufconn`, such as `/rpcpb.SliverRPC/GetVersion`, `GetSessions` and `GetBeacons`
154.8.211[.]145:8899	`/.sliver/logs/console.log`	1,170 B	Tunnel loop from `tunnels.go`, next to `sliver.log` (32,063 B) and `sliver.json` (59,738 B)
154.8.211[.]145:8088	`/b2/PrintSpoofer.cpp`	13,670 B	Named-pipe impersonation source with `-i`, `-d` and `-c` options
the US workbench	`/.sliver-client/sliver-client.log`	262,144 B	A 256 KiB, 3,633-line log of client config creation and tunnel loops
Candidate host (withheld)	Training builder (name withheld)	28,921 B	OSEP training builder cloned from a public repository

Inside `x.ps1`

The same 3,067-byte, 75-line script sits at `154.8.211[.]145:9002/x.ps1` and `:8899/mlet_9002/x.ps1`. This analysis covers the script's static logic. No live exfiltration traffic was captured.

The script targets a common gap. Many web proxies and data loss prevention (DLP) rules inspect `POST` bodies but pass `GET` URLs without a look. Sending data in the URL of a plain HTTP request maps to MITRE ATT&CK T1048.003, not T1041 (exfiltration over the C2 channel).

Step	What the script is written to do
1. Collect	Fill a buffer with eight keys: `WHOAMI=`, `HOST=`, `OS=`, `PSVER=`, `CLR=`, `USER=`, `PROC=` and `PWD=`
2. Encode	Convert the text to Base64 with `[Convert]::ToBase64String`
3. Slice	Cut the Base64 into pieces of up to 2,000 characters (`$kl = 2000`), about 1,500 decoded bytes each
4. Send	Set `$w.Proxy = $null` on a `Net.WebClient`, then call `DownloadString()` on `http://154.8.211[.]145:8080/<tag>?s=<seq>&n=<length>&d=<chunk>`

The eight-key summary is small, about 170 to 280 bytes. In Base64 that is roughly 228 to 376 characters, so it fits in one request. URI rules should therefore fire at 64 or more Base64 characters in the `d=` value, not only at large sizes.

Blind spots. With the proxy set to null, the script connects straight to the server. Explicit web proxy logs never see it unless the firewall blocks direct workstation HTTP. Socket-level endpoint logs also drop the query string. Defenders need network URI rules, PowerShell script block logging (Event ID 4104) or inspected HTTP connection events to see the `d=` values.

CHAPTER 07

Telling Real Sliver Servers from Look-Alikes

IN SHORT

Three traps produce false Sliver alerts or false attribution: cloned code mistaken for original work, database certificates that resemble C2, and generic localhost certificates. Each has a simple check.

INSIDE THIS CHAPTER

Cloned public code 23
The MySQL collision 24
Localhost certificates 24

07 Telling Real Sliver Servers from Look-Alikes

A single indicator often looks like Sliver when it is not. Blocking on it either inflates what an actor can do or fills SOC queues with harmless servers.

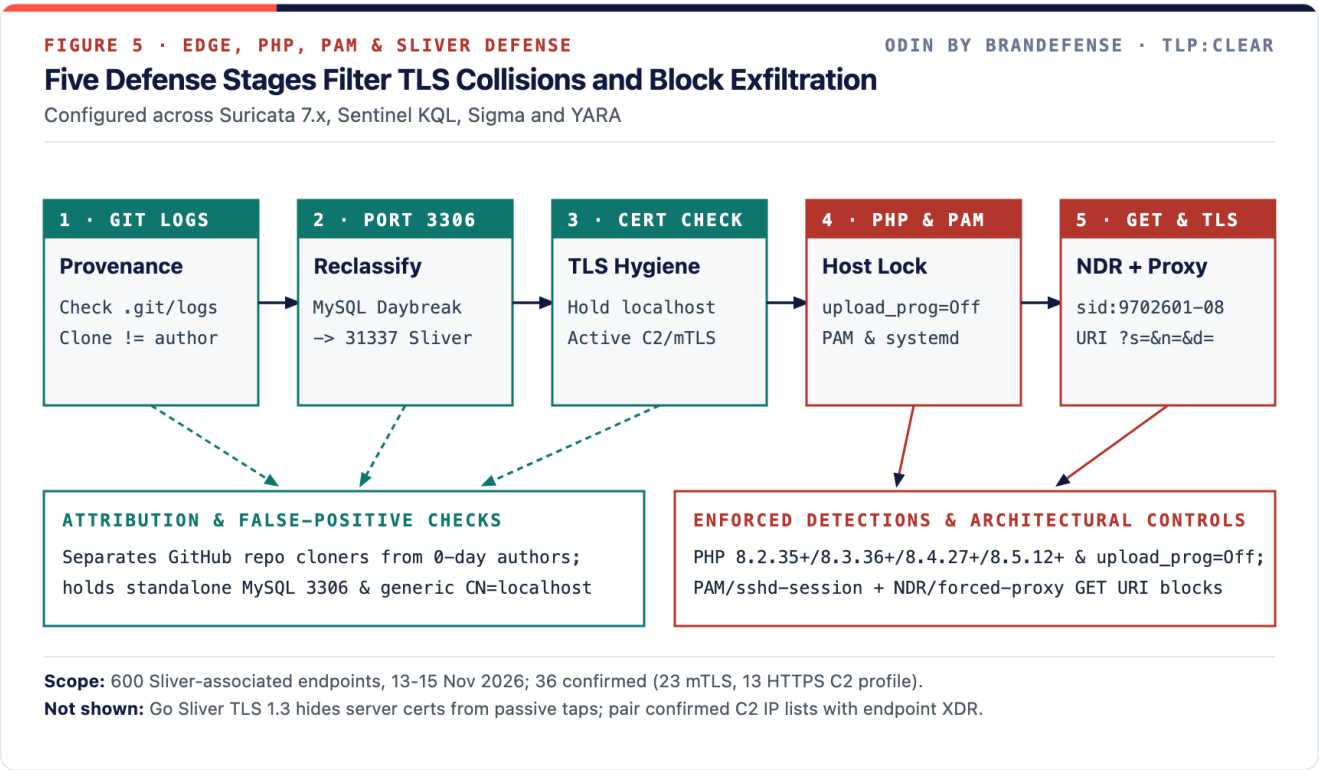


Figure 5: Five Defense Stages Filter TLS Collisions and Block Exfiltration

Trap	What it looks like	How to tell
Cloned code read as authorship	Exploit code and research notes on an operator server	`.git/logs/HEAD` shows only a `clone:` entry and no local commits. Treat the operator as a user of public code, not a zero-day author.
MySQL certificate read as C2	A public JA4X C2 heuristic called Daybreak matches the MySQL 8.x auto-generated certificate	Check the Sliver ports. On 31.56.228[.]136, port 3306 carries `MySQL_Server_8.0.46_Auto_Generated_Server_Certificate`, while port 31337 carries the real Sliver certificate.
Generic localhost certificate read as C2	`CN=localhost` with a common Go or nginx JARM	Self-signed web and container proxies present the same thing. Require an mTLS or HTTP(S) C2 profile match first.

Cloned public code

The three PHP2Shell directories on 31.56.228[.]136 are unmodified clones dated 21 October 2026 on the server. The operator adopted public proof-of-concept code. The four upstream notebook hashes are therefore context only, not block indicators.

The Finnish candidate host shows the same pattern with a training builder. Its OSEP metadata and missing C2 confirmation keep it out of block lists.

The MySQL collision

The Daybreak heuristic uses the JA4X value `7022c563de38_7022c563de38_1c314e97029a`. Auto-generated MySQL 8.x certificates share that structure. On both 31.56.228[.]136 and 154.8.211[.]145, port 31337 presents `CN=multiplayer` with issuer `CN=operators` and the default Sliver JARM `000000...4c1b01`. That is the real signal.

The six standalone MySQL 3306, Redis 6379 and FTP 21 candidates do not belong on a C2 block list.

Localhost certificates

Of the 41 suspected `CN=localhost` endpoints, 37 carry generic HTTPS JARM values and 4 carry none. All 41 stay off block lists, along with the other non-confirmed endpoints (564 in total). Only the 23 mTLS listeners and 13 confirmed HTTP(S) C2 profiles are block-ready.

CHAPTER 08

What Defenders Should Do

IN SHORT

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

08 What Defenders Should Do

Patch and harden PHP

- [ ] Upgrade every internet-facing PHP runtime to 8.2.35, 8.3.36, 8.4.27 or 8.5.12 or later.
- [ ] Where you cannot upgrade at once, set `session.upload_progress.enabled = Off` and isolate PHP-FPM pools. Treat this as exposure reduction only.

Protect Linux logins and services

- [ ] Put `/etc/pam.d/*` and `/lib*/security/*` under file-integrity monitoring (`rpm -Va`, `debsums` or `auditd`).
- [ ] Alert when `sshd`, `sshd-session`, `sudo` or `su` starts a `.sh` file or a script in `/tmp/`.
- [ ] Alert on new hidden files in `/var/log/.*` and `/usr/local/bin/.*`.
- [ ] Hunt `.service` units in `/etc/systemd/system/` that run hidden binaries or `sleep 60` shell loops.

Close the GET exfiltration gap

- [ ] Block direct outbound HTTP from workstations so proxy-bypass scripts cannot connect.
- [ ] Log and alert on GET query strings that carry 64 or more Base64 characters in `?s=...&n=...&d=` form.
- [ ] Turn on PowerShell script block logging (Event ID 4104).

Block Sliver infrastructure carefully

- [ ] Block the 15 confirmed C2 IPs in Chapter 10 on their listed ports.
- [ ] Because Go TLS 1.3 hides the server certificate from passive network taps, pair the IP list with endpoint alerts on outbound 31337 and 8888.
- [ ] Do not auto-block on `CN=localhost`, on database or FTP ports, or on training-tool hosts. Check Git history before calling cloned code a zero-day.
- [ ] Confirm authorized red-team scopes and current cloud IP ownership before a permanent block.

09

CHAPTER 09

Detection Rules

IN SHORT

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

INSIDE THIS CHAPTER

How the Suricata signatures work 28

How the KQL hunt works 28

How the Sigma and YARA rules work 29

How to sort a host 30

09 Detection Rules

The full rules are in Appendix B. Test them in your own Suricata 7.x, Sentinel, Sigma and YARA deployments before you block on them.

Rule file	What it catches
`suricata_php2shell_sliver_and_get_exfil.rules`	Suricata 7: eight signatures for PHP2Shell upload anomalies and leaks, GET exfiltration, task queue polling, Sliver operator certificates and staged tool downloads
`sentinel_pam_systemd_and_get_exfil_hunt.kql`	Microsoft Sentinel and Defender for Endpoint: PAM and systemd changes, login daemons starting scripts, the 60-second loop, chunked GET exfiltration and Sliver port egress
`sigma_linux_pam_backdoor_and_sliver_staging.yml`	Sigma (5 rules): PAM shell hooks, PAM file changes, the masquerading systemd loop, the `x.ps1` script block, and PrintSpoofer use
`yara_php2shell_pam_and_sliver_workbenches.yar`	YARA (4 rules): the PAM harvester, the systemd loop unit, the GET exfiltrator, and exposed Sliver audit logs and databases

How the Suricata signatures work

SID	Looks for	Notes
9702601	A multipart `POST` of 4 MB or more, or chunked, that carries `PHP_SESSION_UPLOAD_PROGRESS`	Hunt priority, rate-limited per source. Best where upload progress is off, since large legitimate uploads can match.
9702602	`s:775:"` in an HTTP response body followed by pointer-like bytes	Serialized response-body memory leak
9702603	An outbound `GET` with `?s=<n>&n=<n>&d=` and 64 or more Base64 characters	The `x.ps1` pattern (T1048.003)
9702604	Requests for `/c2/<hex>.queue` or `/c2/<hex>.out`	File-based task queue polling
9702605	A TLS server certificate with `CN=multiplayer`	Hunt only. Works on TLS 1.2 and below; corroborate port and JARM.
9702606	Requests for `/.sliver/logs/audit.json`, `console.log`, `sliver.log` or `sliver-client.log`	Someone reading an exposed operator server
9702607	Outbound requests for `PrintSpoofer64.exe`, `.exe`, `.cpp` or `.h`	Staged privilege-escalation download
9702608	Outbound requests for `/x.ps1` or `/mlet_9002/x.ps1`	Staged exfiltration script download

Native Go Sliver over TLS 1.3 encrypts the server certificate. Pinned-CA mTLS also resists TLS interception. So SID 9702605 only sees older TLS sessions. Cover TLS 1.3 with the confirmed IP list and endpoint alerts on ports 31337 and 8888.

How the KQL hunt works

The query looks back 14 days and has three parts. The first reads file events in `/etc/pam.d`, the `security` module folders, `/var/log`, the systemd unit folders and `/usr/local/bin`. It flags hidden log files, PAM changes, new or changed `.service` units and hidden binaries in `/usr/local/bin`.

The second part reads process events. It flags `sshd`, `sshd-session`, `sudo`, `su` or `login` starting a shell that runs `toomanysecrets.sh`, a `.sh` file or a script in `/tmp/`, `/var/tmp/` or `/dev/shm/`. Normal login shells and MOTD scripts are excluded. It also flags a hidden `/usr/local/bin/.` binary run with `sleep 60`.

The third part reads network events. It flags inspected URIs with a 64-character or longer Base64 `d=` value. It also flags public connections to ports 31337 or 8888 from anything other than a browser. Results are grouped by category, file and remote address.

How the Sigma and YARA rules work

Rule	Fires when
Sigma: login daemon spawning PAM shell script	`sshd`, `sshd-session`, `sudo`, `su` or `login` starts a `.sh` or `/tmp/` script (T1556.003); excludes `sshd -c` and login shells
Sigma: PAM change or hidden credential log	Writes to `/etc/pam.d/*`, `/lib*/security/*` or `/var/log/.*`
Sigma: masquerading systemd service	A `while true` and `sleep 60` loop runs `/usr/local/bin/.*` (T1543.002, T1036.004)
Sigma: PowerShell chunked GET exfiltration	Event ID 4104 script text with `ToBase64String`, `DownloadString`, `.Proxy = $null` and `?s=&n=&d=` together (T1048.003)
Sigma: PrintSpoofer use	PrintSpoofer binaries, or `-i` or `-d` with `-c cmd` or `-c powershell` (T1134.001); excludes `curl.exe`, `ssh.exe` and `git.exe`
YARA: PAM harvester	At least three PAM harvester strings, such as `$PAM_USER`, `$PAM_RHOST` and a `/var/log/.` sink, in a file under 10 KB
YARA: systemd loop unit	At least three strings of the `systemd-resolver-helper.service` loop pattern, in a file under 10 KB
YARA: GET exfiltrator	At least three exfiltrator strings, such as `$kl = 2000` and `$w.Proxy = $null`, in a file under 50 KB
YARA: exposed Sliver workbench	Two gRPC audit strings (such as `"remote_ip":"bufconn"` and `/rpcpb.SliverRPC/`), or two tunnel or console log strings, or database helper strings, in a file under 5 MB

Each YARA rule needs several strings and has a file size bound, so a single word in a large unrelated file does not fire it.

How to sort a host

What the host shows	Result
Confirmed Sliver C2 plus a PAM or PHP workbench on the same host	Critical: block and hunt
An exposed Sliver teamserver with the GET exfiltration script	Critical: block and hunt
An exposed Sliver database workbench	Critical: block and hunt
A Sliver mTLS handshake or a matched HTTP(S) C2 profile	Confirmed C2: block
PAM backdoor or PHP2Shell files	Confirmed workbench: hunt
A training-tool builder with no C2 confirmation	Candidate staging: analyst review
`CN=localhost`, a custom certificate or an unverified Sliver JARM alone	Suspected: hold or review, do not block
A standalone MySQL, Redis or FTP port	False positive

It also labels a host whose exploit code is only cloned from public repositories as a public proof-of-concept adopter, not an author.

10

CHAPTER 10

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Confirmed Sliver C2 endpoints 32

File hashes for hunting 33

MITRE ATT&CK mapping 33

10 Indicators

Appendix A holds 22 published indicators: the 15 confirmed Sliver IPs and 7 file hashes. The selected set below has 28 records; the rest stay in the text or are withheld.

Type	Count	Use
Confirmed Sliver C2 IPs (with port)	15	Block
Hunt artifacts: 7 file hashes and the exposed workbench address (withheld)	8	Hunt
Candidate dual-use host	1	Withheld; do not auto-block
Cloned upstream repository hashes	4	Context only; do not block

Withheld: the Finnish candidate host and its port, paths and project names; the PAM master password; the exposed database and key paths; and all suspected and candidate addresses.

Confirmed Sliver C2 endpoints

IP	Port	Network (country)	Key fingerprint
31.56.228[.]136	31337	AS56971 Cloud (FR)	`CN=multiplayer`; PHP2Shell and PAM workbench
154.8.211[.]145	31337	Shenzhen Tencent (CN)	`CN=multiplayer`; exposed audit logs, PrintSpoofer, `x.ps1`
13.49.66[.]51	31337	Amazon.com, Inc. (SE)	`CN=multiplayer`, JARM 000000…
34.122.21[.]68	31337	Google LLC (US)	`CN=multiplayer`
54.248.136[.]140	31337	Amazon.com, Inc. (JP)	`CN=multiplayer`, JARM 000000…
161.97.143[.]156	31337	Contabo GmbH (GB)	`CN=multiplayer`, JARM 000000…
146.103.42[.]94	31337	Regxa Company for Information Technology (DE)	`CN=multiplayer`, JARM 000000…
45.154.246[.]236	31337	New Hosting Technologies (RU)	`CN=multiplayer`
129.225.178[.]74	31337	Oracle Corporation (JP)	`CN=multiplayer`, JARM 000000…
176.65.149[.]95	8443	Pfcloud UG (NL)	HTTP(S) C2 profile, `CN=localhost`, JARM `3fd3fd...`
194.48.248[.]211	8443	ALEXHOST S.R.L. (MD)	HTTP(S) C2 profile, `CN=sonder.chrysalisfail[.]com`, JARM `3fd3fd...`
164.90.149[.]44	443	DigitalOcean, LLC (US)	HTTP(S) C2 profile, `CN=localhost`, JARM `21d21d...`
162.254.35[.]170	443	Namecheap, Inc. (US)	HTTP(S) C2 profile, `CN=localhost`, JARM `29d29d...`
65.109.139[.]13	443	Hetzner Online GmbH (FI)	HTTP(S) C2 profile, `CN=localhost`, JARM `3fd21c...`
165.227.231[.]125	443	DigitalOcean, LLC (GB)	HTTP(S) C2 profile, `CN=localhost`, JARM `3fd21b...`

The exposed US workbench (port 8000, NetLab Global) is withheld. Its Sliver C2 port is suspected, not confirmed, and the host also runs a dual-use Interactsh server.

File hashes for hunting

Three hunt hashes come from 31.56.228[.]136: `/toomanysecrets.sh`, the systemd unit and the queue file. They are in the file table in Chapter 5, with the four context-only hashes. The other rows there are reference only. The four hunt hashes below come from 154.8.211[.]145.

File	Port	Size	SHA-256
`/.sliver/logs/audit.json`	8899	821 B	`e1aea4aaf1ff93fe955481eb23ad95e9a758da1dc082246c7d66f43d5440cc9a`
`/.sliver/logs/console.log`	8899	1,170 B	`25c31cbe5ea34b563288f3977535bc312a5e3c997b6042c9e6a04ca48ed73633`
`/x.ps1` and `/mlet_9002/x.ps1`	9002, 8899	3,067 B	`b3f9bdc86a33abc690935f4445c0fba9b23d9180c64d6085f959e7aee7dcd130`
`/b2/PrintSpoofer.cpp`	8088	13,670 B	`2817faafad5dc1b3f4176202285a6790e953fc7f70ae2f8fae1c9b5b42cd5030`

MITRE ATT&CK mapping

Tactic	Technique	How it appears here	Detection
Credential Access, Persistence	T1556.003 Modify Authentication Process: Pluggable Authentication Modules	`toomanysecrets.sh` master password and `/var/log/.pam.log` harvest	Sigma PAM rules; YARA PAM harvester; KQL file and process parts
Persistence	T1543.002 Create or Modify System Process: Systemd Service	`systemd-resolver-helper.service` 60-second loop	Sigma systemd rule; YARA loop unit
Defense Evasion	T1036.004 Masquerading: Masquerade Task or Service	Service name mimics `systemd-resolved`; hidden `.cache-helper` binary	Sigma systemd rule; KQL hidden binary check
Exfiltration	T1048.003 Exfiltration Over Unencrypted Non-C2 Protocol	`x.ps1` Base64 chunks in HTTP GET query strings	SID 9702603; Sigma 4104 rule; YARA exfiltrator
Privilege Escalation	T1134.001 Access Token Manipulation: Token Impersonation/Theft	PrintSpoofer staged on 154.8.211[.]145	SID 9702607; Sigma PrintSpoofer rule

11

CHAPTER 11

Scope and Limits

IN SHORT

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

11 Scope and Limits

- **Window.** The findings come from a two-week observation window ending 18 November 2026. The 600 endpoints were last seen between 13 and 15 November 2026. The 5,471-record figure reflects upstream history, not our observation span.
- **Dates on the servers.** The 21 October 2026 dates are Git clone timestamps written on the operator's server. They show when the repositories were cloned, not when we looked.
- **Percentages describe the 600.** Port, country and hosting shares describe the 600 endpoints in this report. They do not project onto the wider record set.
- **Staged files only.** The PAM backdoor, systemd unit, task queue and `x.ps1` are staged files. We saw no evidence that they ran on any victim, and no live exfiltration traffic was captured.
- **Lab or intrusion.** Test strings such as `C2_ALIVE` and `uid=997` output fit both an intrusion staging host and an adversary-emulation lab. Some operators may be authorized testers.
- **Cloned research is not the operator's work.** The PHP2Shell notebooks and lab logs came from public repositories. They do not show that this operator used the exploit against anyone.
- **Rule testing.** Every published rule file loads without errors in its engine (Suricata 7.0.12, YARA-X, pySigma and Microsoft's Kusto.Language analyzer, checked on 20 November 2026). The rules were not replayed against live traffic, so test and tune them in your own environment before production use.

12

CHAPTER 12

Sources

IN SHORT

This chapter lists the public references used in the report.

12 Sources

1. PHP Manual. Session Upload Progress. https://www.php.net/manual/en/session.upload-progress.php (verified 18 November 2026)
2. MITRE ATT&CK. T1556.003 Pluggable Authentication Modules. https://attack.mitre.org/techniques/T1556/003/ (verified 18 November 2026)
3. BishopFox. Sliver documentation. https://sliver.sh/docs (verified 18 November 2026)
4. MITRE ATT&CK. T1048.003 Exfiltration Over Unencrypted Non-C2 Protocol. https://attack.mitre.org/techniques/T1048/003/ and T1041 Exfiltration Over C2 Channel. https://attack.mitre.org/techniques/T1041/ (verified 18 November 2026)
5. MITRE ATT&CK. T1543.002 Systemd Service. https://attack.mitre.org/techniques/T1543/002/ (verified 18 November 2026)
6. MITRE ATT&CK. T1134.001 Token Impersonation/Theft. https://attack.mitre.org/techniques/T1134/001/ (verified 18 November 2026)
7. PHP. php-src security advisories GHSA-w8hw-w9fx-j79w and GHSA-6wmf-4rv5-p3rp. https://github.com/php/php-src/security/advisories (verified 18 November 2026)
8. itm4n. PrintSpoofer repository. https://github.com/itm4n/PrintSpoofer (verified 18 November 2026)
9. PHP. PHP 8 ChangeLog. https://www.php.net/ChangeLog-8.php (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 (22) 39

B. Detection rules (4 files) 41

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
R07-IOC-01	IPv4	31.56.228[.]136	31337	Confirmed Sliver mTLS C2 + PHP2Shell + PAM Backdoor Host	BLOCK
R07-IOC-02	IPv4	154.8.211[.]145	31337	Confirmed Sliver mTLS C2 (31337/tcp) + Exposed Workbench (:8899/:8088/:9002)	BLOCK
R07-IOC-03	SHA-256	112967dcd581e2cc9b9afd6270321ddbdcb1b984cf0e1198c081e19c6f3968d1	9999	Linux PAM Master-Password & Credential Logger Backdoor SHA-256	HUNT
R07-IOC-04	SHA-256	0df9cb861c20639c86d8edb7336b5af7a190093d842b3b0587704d897ebf72d3	9999	Masquerading Systemd 60s Loop Persistence Unit SHA-256	HUNT
R07-IOC-05	SHA-256	29c7b6d0b65c9c1d623e0fc7290b044b371d2110e68c0e2e813f8e3d8e668122	9999	Lightweight File-Backed HTTP C2 Task Queue SHA-256	HUNT
R07-IOC-06	SHA-256	e1aea4aaf1ff93fe955481eb23ad95e9a758da1dc082246c7d66f43d5440cc9a	8899	Exposed Sliver gRPC Operator Audit Log SHA-256	HUNT
R07-IOC-07	SHA-256	25c31cbe5ea34b563288f3977535bc312a5e3c997b6042c9e6a04ca48ed73633	8899	Exposed Sliver Tunnel & Console Stream Log SHA-256	HUNT
R07-IOC-08	SHA-256	b3f9bdc86a33abc690935f4445c0fba9b23d9180c64d6085f959e7aee7dcd130	9002	PowerShell 2,000-Char (~1,500-Byte) Chunked HTTP GET Exfiltrator SHA-256	HUNT
R07-IOC-09	SHA-256	2817faafad5dc1b3f4176202285a6790e953fc7f70ae2f8fae1c9b5b42cd5030	8088	Staged SeImpersonatePrivilege PrintSpoofer Source SHA-256	HUNT
R07-IOC-10	IPv4	13.49.66[.]51	31337	Confirmed Sliver C2 Teamserver / mTLS Listener	BLOCK
R07-IOC-11	IPv4	176.65.149[.]95	8443	Confirmed Sliver HTTPS C2 Listener (Active Profile Verified)	BLOCK
R07-IOC-12	IPv4	34.122.21[.]68	31337	Confirmed Sliver C2 Teamserver / mTLS Listener	BLOCK
R07-IOC-13	IPv4	164.90.149[.]44	443	Confirmed Sliver HTTPS C2 Listener (Active Profile Verified)	BLOCK
R07-IOC-14	IPv4	162.254.35[.]170	443	Confirmed Sliver HTTPS C2 Listener (Active Profile Verified)	BLOCK
R07-IOC-15	IPv4	65.109.139[.]13	443	Confirmed Sliver HTTPS C2 Listener (Active Profile Verified)	BLOCK
R07-IOC-16	IPv4	54.248.136[.]140	31337	Confirmed Sliver C2 Teamserver / mTLS Listener	BLOCK
R07-IOC-17	IPv4	194.48.248[.]211	8443	Confirmed Sliver HTTPS C2 Listener (Active Profile Verified)	BLOCK
R07-IOC-18	IPv4	161.97.143[.]56	31337	Confirmed Sliver C2 Teamserver / mTLS Listener	BLOCK
R07-IOC-19	IPv4	146.103.42[.]94	31337	Confirmed Sliver C2 Teamserver / mTLS Listener	BLOCK
R07-IOC-20	IPv4	165.227.231[.]125	443	Confirmed Sliver HTTPS C2 Listener (Active Profile Verified)	BLOCK

ID	Type	Indicator	Port	Context	Action
R07-IOC-21	IPv4	45.154.246[.]236	31337	Confirmed Sliver C2 Teamserver / mTLS Listener	BLOCK
R07-IOC-22	IPv4	129.225.178[.]74	31337	Confirmed Sliver C2 Teamserver / mTLS Listener	BLOCK

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_pam_systemd_and_get_exfil_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Report #07 Detection Engineering Suite
// Target: Microsoft Sentinel / Microsoft Defender for Endpoint (Linux & Windows XDR)
// Telemetry Scope Note: Socket-level DeviceNetworkEvents (ConnectionSuccess) on bare-IP HTTP
// connections omit query strings in RemoteUrl, and x.ps1 sets $w.Proxy = $null to bypass explicit
// web proxies. Block direct workstation HTTP egress (forcing traffic through a transparent proxy)
// and pair HttpConnectionInspected -> AdditionalFields.uri with NDR (sid:9702603) and EventID 4104.

let Lookback = 14d;
let SliverDefaultPorts = dynamic([31337, 8888]);
let PamAndSystemdEvents = union
    (
        DeviceFileEvents
        | where Timestamp >= ago(Lookback)
        | where FolderPath has_any ("/etc/pam.d", "/lib/security", "/lib64/security", "/lib/x86_64-linux-gnu/
security", "/var/log", "/etc/systemd/system", "/usr/lib/systemd/system", "/lib/systemd/system", "/usr/local/bin")
        | extend FullPath = iff(FolderPath endswith strcat("/", FileName) or isempty(FileName), FolderPath,
strcat(trim_end("/", FolderPath), "/", FileName))
        | extend HuntCategory = case(
            FullPath matches regex @"^/var/log/\.[^/]+$" or FileName == ".pam.log",
                "Linux_PAM_Hidden_Credential_Log",
            (FullPath startswith "/etc/pam.d/" or FullPath startswith "/lib/security/" or FullPath startswith "/
lib64/security/" or FullPath startswith "/lib/x86_64-linux-gnu/security/") and ActionType in ("FileCreated",
"FileModified", "FileRenamed"),
                "Linux_PAM_Config_Or_Module_Modification",
            FullPath matches regex @"^/(etc|usr/lib|lib)/systemd/system/[^/]+\.service$" and ActionType in
("FileCreated", "FileModified", "FileRenamed"),
                "Linux_Systemd_Unit_Creation_Or_Modification",
            FullPath matches regex @"^/usr/local/bin/\.[^/]+$",
                "Hidden_DotBinary_Staging_UsrLocalBin",
            "None"
        )
        | where HuntCategory != "None"
        | project Timestamp, DeviceName, HuntCategory, Artifact = FullPath,
                  Detail = InitiatingProcessCommandLine, RemoteIP = "", RemotePort = toint(0)
    ),
    (
        DeviceProcessEvents
        | where Timestamp >= ago(Lookback)
        | extend HuntCategory = case(
            InitiatingProcessFileName in ("sshd", "sshd-session", "sudo", "su", "login")
                and FileName in ("bash", "sh", "dash", "zsh")
                and (ProcessCommandLine has "toomanysecrets.sh" or ProcessCommandLine matches regex @"(/tmp/|/
var/tmp/|/dev/shm/|/usr/local/bin/\.|[A-Za-z0-9_.-]+\.sh\b)"),
                "Linux_AuthDaemon_PAM_Exec_Shell_Hook",
            ProcessCommandLine matches regex @"/usr/local/bin/\.[A-Za-z0-9_-]+" and ProcessCommandLine has "sleep
60",
                "Masquerading_Systemd_60s_Beacon_Loop",
            "None"
        )
        | where HuntCategory != "None"
        | project Timestamp, DeviceName, HuntCategory, Artifact = FileName,
                  Detail = ProcessCommandLine, RemoteIP = "", RemotePort = toint(0)
    );
let NetworkExfilAndSliver =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback) and ActionType in ("ConnectionSuccess", "HttpConnectionInspected") and
RemoteIPType == "Public"
    | extend InspectedUri = coalesce(tostring(parse_json(AdditionalFields).uri), RemoteUrl)
    | extend HuntCategory = case(
        extract(@"[?&]s=\d+&n=[1-9]\d{1,3}&d=([A-Za-z0-9+/=%]{64,})", 1, InspectedUri) != "",
```

```
            "Chunked_Base64_HTTP_GET_Query_Exfil",
        ActionType == "ConnectionSuccess" and RemotePort in (SliverDefaultPorts),
            "Suspected_Sliver_mTLS_Or_Operator_Egress",
        "None"
    )
    | where HuntCategory != "None"
    | project Timestamp, DeviceName, HuntCategory, Artifact = InitiatingProcessFileName,
              Detail = InspectedUri, RemoteIP = tostring(RemoteIP), RemotePort = toint(RemotePort);
union PamAndSystemdEvents, NetworkExfilAndSliver
| summarize
    FirstSeen = min(Timestamp),
    LastSeen = max(Timestamp),
    EventCount = count(),
    DistinctHosts = dcount(DeviceName),
    SampleDetails = make_set(Detail, 5)
    by HuntCategory, Artifact, RemoteIP, RemotePort
| order by DistinctHosts desc, EventCount desc
```

sigma_linux_pam_backdoor_and_sliver_staging.yml

SIGMA

```
title: Linux Authentication Daemon Spawning Unexpected PAM Shell Script
id: 97a1c410-3e11-4a92-8b01-202611160701
status: experimental
description: >
  Detects Linux authentication daemons (sshd, OpenSSH 9.8+ sshd-session, sudo, su, login)
  spawning shell interpreters that execute standalone .sh scripts or hidden/writable-directory
  payloads via pam_exec.so hooks (such as toomanysecrets.sh), excluding ordinary
  interactive login shells and sshd -c command sessions.
author: Odin by Brandefense
date: 2026-11-16
logsource:
  category: process_creation
  product: linux
detection:
  selection_auth_parent:
    ParentImage|endswith:
      - '/sshd'
      - '/sshd-session'
      - '/login'
      - '/su'
      - '/sudo'
  selection_shell:
    Image|endswith:
      - '/bash'
      - '/sh'
      - '/dash'
      - '/zsh'
  selection_script_target:
    CommandLine|contains:
      - 'toomanysecrets.sh'
      - '.sh'
      - '/tmp/'
      - '/var/tmp/'
      - '/dev/shm/'
      - '/usr/local/bin/.'
  condition: selection_auth_parent and selection_shell and selection_script_target
falsepositives:
  - Approved site-specific pam_exec.so helper .sh scripts invoked during SSH login; scope exclusions by approved
script path and correlate with /etc/pam.d/ or /var/log/.* file events
level: high
tags:
  - attack.credential_access
  - attack.persistence
  - attack.t1556.003
---
title: Linux PAM Configuration Modification or Hidden Credential Log Sink
id: 97a2c410-3e11-4a92-8b01-202611160705
status: experimental
```

```
  description: >
    Detects file creation or modification inside /etc/pam.d/, Linux PAM security
    module directories, or hidden dot-prefixed credential harvesting log files under
    /var/log/ (such as /var/log/.pam.log written via shell redirection inside pam_exec.so scripts).
  author: Odin by Brandefense
  date: 2026-11-16
  logsource:
    category: file_event
    product: linux
  detection:
    selection_pamd_or_mod:
      TargetFilename|startswith:
        - '/etc/pam.d/'
        - '/lib/security/'
        - '/lib64/security/'
        - '/lib/x86_64-linux-gnu/security/'
    selection_hidden_varlog:
      TargetFilename|startswith: '/var/log/.'
    filter_package_managers:
      Image:
        - '/usr/bin/dpkg'
        - '/usr/bin/apt-get'
        - '/usr/bin/rpm'
        - '/usr/bin/dnf'
        - '/usr/sbin/pam-auth-update'
    condition: (selection_pamd_or_mod or selection_hidden_varlog) and not filter_package_managers
  falsepositives:
    - Unattended configuration management updates modifying /etc/pam.d/ outside standard package managers
  level: critical
  tags:
    - attack.credential_access
    - attack.persistence
    - attack.t1556.003
  ---
  title: Masquerading Systemd Service Executing Hidden Binary in UsrLocalBin
  id: 97b2d521-4f22-4b03-9c12-202611160702
  status: experimental
  description: >
    Detects systemd-launched bash loops executing hidden dot-prefixed binaries inside
    /usr/local/bin/, matching the systemd-resolver-helper.service persistence pattern.
  author: Odin by Brandefense
  date: 2026-11-16
  logsource:
    category: process_creation
    product: linux
  detection:
    selection_hidden_exec:
      CommandLine|contains:
        - '/usr/local/bin/.'
        - '.cache-helper'
    selection_loop:
      CommandLine|contains|all:
        - 'while true'
        - 'sleep 60'
    condition: selection_hidden_exec and selection_loop
  falsepositives:
    - None expected in standard enterprise Linux distributions
  level: critical
  tags:
    - attack.persistence
    - attack.privilege_escalation
    - attack.t1543.002
    - attack.t1036.004
  ---
  title: PowerShell ScriptBlock Proxy-Bypassing Chunked HTTP GET Exfiltration
  id: 97c3e632-5a33-4c14-8d23-202611160703
  status: experimental
  description: >
    Detects PowerShell ScriptBlock execution (EventID 4104) bypassing explicit WebClient
    proxies and transmitting Base64-encoded reconnaissance output in 2,000-character
    query-string chunks (?s=...&n=...&d=), matching staged utility x.ps1.
  author: Odin by Brandefense
```

```
  date: 2026-11-16
  logsource:
    category: ps_script
    product: windows
  detection:
    selection_ps_get_chunk:
      ScriptBlockText|contains|all:
        - 'ToBase64String'
        - 'DownloadString'
        - '.Proxy = $null'
        - '?s='
        - '&n='
        - '&d='
    condition: selection_ps_get_chunk
  falsepositives:
    - Authorized red-team adversary emulation exercises testing query-string DLP inspection
  level: high
  tags:
    - attack.exfiltration
    - attack.t1048.003
  ---
  title: Windows PrintSpoofer SeImpersonatePrivilege Execution
  id: 97d4f743-6b44-4d25-9e34-202611160704
  status: experimental
  description: >
    Detects Windows PrintSpoofer privilege-escalation execution abusing SeImpersonatePrivilege
    via named-pipe impersonation and interactive/desktop/command CLI flags (-i, -d <id>, -c <shell>).
  author: Odin by Brandefense
  date: 2026-11-16
  logsource:
    category: process_creation
    product: windows
  detection:
    selection_binary:
      - Image|endswith:
          - '\PrintSpoofer64.exe'
          - '\PrintSpoofer.exe'
      - OriginalFileName: 'PrintSpoofer.exe'
    selection_cli_mode:
      CommandLine|re: '\s-(?:i|d\s+\d+)\b'
    selection_cli_cmd:
      CommandLine|re: '(?i)\s-c\s+"?(?:cmd|powershell)(?:\.exe)?\b'
    filter_common_cli_tools:
      Image:
        - 'C:\Windows\System32\curl.exe'
        - 'C:\Windows\System32\OpenSSH\ssh.exe'
        - 'C:\Program Files\Git\cmd\git.exe'
        - 'C:\Program Files\Git\bin\git.exe'
        - 'C:\Program Files\Git\mingw64\bin\git.exe'
    condition: selection_binary or (selection_cli_mode and selection_cli_cmd and not filter_common_cli_tools)
  falsepositives:
    - Rare administrative wrappers passing PrintSpoofer-style -i/-d <session> alongside -c cmd/powershell
  level: high
  tags:
    - attack.privilege_escalation
    - attack.t1134.001
```

suricata_php2shell_sliver_and_get_exfil.rules

SURICATA

```
# Odin by Brandefense - Report #07 Suricata 7.x PHP2Shell, PAM/Queue C2 & Sliver Suite
# Classification: TLP:CLEAR
# Purpose: Perimeter and internal NDR rules covering CVE-2026-109618 multipart OOM triggers
#          (hunt-priority on upload-progress endpoints), serialized HTTP response-body pointer
#          leaks, x.ps1 chunked Base64 HTTP GET exfiltration, file-backed /c2/*.queue polling,
#          and Sliver multiplayer / workbench exposures.

# 1. Hunt-priority multipart PHP_SESSION_UPLOAD_PROGRESS OOM anomaly (best deployed where
session.upload_progress.enabled=Off; legitimate >=4MB progress-bar uploads may match)
```

```
alert http any any -> $HOME_NET any (msg:"ODIN-SLV PHP2Shell CVE-2026-109618 Multipart Upload Progress OOM
Anomaly"; flow:established,to_server; http.method; content:"POST"; http.content_type; content:"multipart/form-
data"; nocase; http.header; pcre:"/^(?:Content-Length:\s*(?:[4-9]\d{6}|[1-9]\d{7,})|Transfer-Encoding:
\s*chunked)/mi"; http.request_body; bsize:>4096; content:"PHP_SESSION_UPLOAD_PROGRESS"; nocase; fast_pattern;
threshold:type limit, track by_src, count 1, seconds 60; classtype:web-application-attack; priority:3; sid:
9702601; rev:4;)

# 2. PHP2Shell serialized HTTP response-body 2MB Zend MM heap pointer leak (s:775:"..." with raw little-endian
x86_64 pointer bytes)
alert http $HOME_NET any -> any any (msg:"ODIN-SLV PHP2Shell Serialized HTTP Response Body Heap Pointer Leak";
flow:established,from_server; http.response_body; content:"s:775:\""; fast_pattern; content:"|7f 00 00|";
distance:0; within:775; classtype:attempted-admin; sid:9702602; rev:3;)

# 3. PowerShell x.ps1 chunked Base64 HTTP GET query-string exfiltration (matches ~228-376 char single-frame recon
up to 2,000-char / ~1,500B slices)
alert http $HOME_NET any -> !$HOME_NET any (msg:"ODIN-SLV PowerShell x.ps1 Chunked Base64 HTTP GET Exfiltration";
flow:established,to_server; http.method; content:"GET"; http.uri; content:"?s="; fast_pattern; content:"&n=";
distance:0; content:"&d="; distance:0; pcre:"/[?&]s=\d+&n=[1-9]\d{1,3}&d=[A-Za-z0-9+\/=%]{64,}/";
classtype:policy-violation; sid:9702603; rev:4;)

# 4. Lightweight file-backed HTTP C2 task queue polling (/c2/*.queue or /c2/*.out)
alert http any any -> any any (msg:"ODIN-SLV Lightweight File-Backed HTTP C2 Task Queue Polling";
flow:established,to_server; http.uri; content:"/c2/"; fast_pattern; pcre:"/\/c2\/[a-f0-9]{6,16}\.(?:queue|out)(?:
\?|$)/i"; classtype:command-and-control; sid:9702604; rev:3;)

# 5. Possible Sliver multiplayer operator certificate hunt signal (legacy TLS <= 1.2 only; Go TLS 1.3 encrypts
certs and pinned mTLS resists SSL inspection - corroborate port/JARM before blocking)
alert tls !$HOME_NET any -> $HOME_NET any (msg:"ODIN-SLV Possible Sliver Multiplayer Cert Hunt Signal (TLS 1.2
Operator Session - Corroborate Port/JARM)"; flow:established,from_server; tls.cert_subject;
content:"CN=multiplayer"; nocase; fast_pattern; classtype:command-and-control; priority:3; sid:9702605; rev:4;)

# 6. Exposed Sliver operator workbench gRPC audit stream or tunnel log retrieval
alert http any any -> any any (msg:"ODIN-SLV Exposed Sliver Workbench Audit Stream or Console Log Access";
flow:established,to_server; http.uri; content:"/.sliver"; nocase; fast_pattern; pcre:"/\/\.sliver(?:-client)?\/
(?:logs\/(?:audit|sliver)\.json|logs\/(?:console|sliver)\.log|sliver-client\.log)(?:\?|$)/i"; classtype:trojan-
activity; sid:9702606; rev:4;)

# 7. Staged PrintSpoofer privilege-escalation binary or source retrieval over HTTP
alert http $HOME_NET any -> !$HOME_NET any (msg:"ODIN-SLV Staged PrintSpoofer Privilege Escalation HTTP
Retrieval"; flow:established,to_server; http.uri; content:"PrintSpoofer"; nocase; fast_pattern; pcre:"/\/
PrintSpoofer(?:64\.exe|\.exe|\.cpp|\.h)(?:\?|$)/i"; classtype:trojan-activity; sid:9702607; rev:3;)

# 8. Staged PowerShell x.ps1 HTTP GET exfiltrator retrieval (/x.ps1 or /mlet_9002/x.ps1)
alert http $HOME_NET any -> !$HOME_NET any (msg:"ODIN-SLV Staged PowerShell x.ps1 Exfiltrator HTTP Retrieval";
flow:established,to_server; http.uri; content:"/x.ps1"; nocase; fast_pattern; pcre:"/\/(?:mlet_9002\/)?x\.ps1(?:
\?|$)/i"; classtype:trojan-activity; sid:9702608; rev:2;)
```

yara_php2shell_pam_and_sliver_workbenches.yar

YARA

```
/*
   Odin by Brandefense - Report #07 YARA PAM Backdoor, Systemd Loop & Sliver Workbench Rules
   Classification: TLP:CLEAR
*/

rule ODIN_Linux_PAM_Credential_Harvester_Backdoor {
    meta:
        author = "Odin by Brandefense"
        description = "Detects Linux PAM pam_exec credential-harvesting shell scripts reading stdin tokens and
logging PAM environment variables to hidden /var/log/.* sinks"
        classification = "TLP:CLEAR"
        mitre_attack = "T1556.003"
    strings:
        $s1 = "$PAM_USER" ascii
        $s2 = "$PAM_RHOST" ascii
        $s3 = "/var/log/." ascii
        $s4 = "read " ascii
    condition:
        filesize < 10KB and 3 of them
```

```
}

rule ODIN_Masquerading_Systemd_Cache_Helper_Loop {
    meta:
        author = "Odin by Brandefense"
        description = "Detects masquerading systemd unit files executing hidden /usr/local/bin/.* binaries in 60s
loops"
        classification = "TLP:CLEAR"
        mitre_attack = "T1543.002"
    strings:
        $u1 = "Description=Systemd Resolver Helper" ascii
        $u2 = "/usr/local/bin/.cache-helper" ascii
        $u3 = "while true; do" ascii
        $u4 = "sleep 60; done" ascii
        $u5 = "Restart=always" ascii
    condition:
        filesize < 10KB and 3 of them
}

rule ODIN_PowerShell_Chunked_HTTP_GET_Exfiltrator {
    meta:
        author = "Odin by Brandefense"
        description = "Detects PowerShell x.ps1 utility slicing Base64 host recon into 2,000-character (~1,500-
byte) HTTP GET query chunks"
        classification = "TLP:CLEAR"
        mitre_attack = "T1048.003"
    strings:
        $p1 = "[Convert]::ToBase64String($bytes)" ascii nocase
        $p2 = "$kl = 2000" ascii
        $p3 = "$w.Proxy = $null" ascii nocase
        $p4 = "'?s=' + $k + '&n=' + $n + '&d=' + $chunk" ascii
        $p5 = "DownloadString($u)" ascii nocase
    condition:
        filesize < 50KB and 3 of them
}

rule ODIN_Exposed_Sliver_Audit_Stream_And_DB_Workbench {
    meta:
        author = "Odin by Brandefense"
        description = "Detects exposed Sliver teamserver gRPC audit.json streams, server/tunnel console logs, or
SQLite DB helper scripts"
        classification = "TLP:CLEAR"
        mitre_attack = "T1071.001"
    strings:
        $g1 = "/rpcpb.SliverRPC/GetSessions" ascii
        $g2 = "/rpcpb.SliverRPC/GetBeacons" ascii
        $g3 = "\\\"remote_ip\\\":\\\"bufconn\\\"" ascii
        $g4 = "/rpcpb.SliverRPC/GetVersion" ascii
        $t1 = "tunnels.go:45: Creating single instance of tunnels." ascii
        $t2 = "Starting tunnel data loop ..." ascii
        $t3 = "Waiting for TunnelData ..." ascii
        $s1 = "sliver/server/configs/database.go" ascii
        $s2 = "sliver/server/db/sql-go.go" ascii
        $s3 = "Connecting to SQLite database" ascii
        $s4 = "initializing traffic encoder map..." ascii
    condition:
        filesize < 5MB and (
            2 of ($g*) or
            2 of ($t*) or
            2 of ($s*)
        )
}
```

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/exposed-sliver-workbenches/
Short Read · brandefense.io/odin-research/blog/open-sliver-servers-reveal-php-exploits-pam-backdoors-and-quiet/
Defense Kit · brandefense.io/odin-research/defense-kit/