ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 06 / 30 · CREDENTIAL THEFT

Keystrokes Before Submit

AiTM Workbenches, noVNC and Ligolo-ng Tunnels

What 2,750 recently active phishing, proxy and tunnel listeners and three exposed workbenches show about post-MFA intrusion

Two-week observation window ending 18 November 2026 · Data cutoff 16 November 2026 · Prepared November 2026
Full Report · Public edition · Version 1.1

2,750
Recently active phishing, proxy and tunnel listeners on 2,212 IPv4 hosts in 69 countries

26,329
Global banner records for GoPhish (23,207), Mitmproxy (1,877), Ligolo-ng (938) and EvilGoPhish (307)

48
Listeners with commercial proxy (44) or awareness-training (4) certificate names, held out of blocklists

1,196
Files on three exposed workbenches: 247, 117 (Host B) and 832 (Host C)

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

What 2,750 recently active phishing, proxy and tunnel listeners and three exposed workbenches show about post-MFA intrusion

01	Executive Summary	04
02	The Landscape: Phishing, Proxy and Tunnel Listeners	07
03	A Keystroke Streamer Beside Evilginx2 and Cobalt Strike	11
04	M365 Enumeration Scripts and a Co-Hosted noVNC Client	15
05	Ligolo-ng Tunnel Relays and the YoroTrooper Cluster	19
06	Keeping Vendors and Dual-Use Tools off Blocklists	23
07	What Defenders Should Do	26
08	Detection Rules	28
09	Indicators	32
10	Scope and Limits	35
11	Sources	37
A	Indicators of compromise (3)	39
B	Detection rules (4 files)	40

IN SHORT

In a two-week window, with data up to 16 November 2026, Odin by Brandefense reviewed 2,750 recently active phishing, proxy and tunnel listeners and three exposed workbenches. One server holds a kit built to capture keystrokes before a login form is submitted, beside Evilginx2 and Cobalt Strike. Another pairs a Mitmproxy listener with noVNC client files. Ligolo-ng relays favour port 11601 and HTTPS. These are separate findings, not one campaign. Require origin-bound passkeys, block outbound ports 11601 to 11610, and keep vendor proxies off blocklists.

01

CHAPTER 01

Executive Summary

IN SHORT

In a two-week window, with data up to 16 November 2026, Odin by Brandefense reviewed 2,750 recently active phishing, proxy and tunnel listeners and three exposed workbenches. One server holds a kit built to capture keystrokes before a login form is submitted, beside Evilginx2 and Cobalt Strike. Another pairs a Mitmproxy listener with noVNC client files. Ligolo-ng relays favour port 11601 and HTTPS. These are separate findings, not one campaign. Require origin-bound passkeys, block outbound ports 11601 to 11610, and keep vendor proxies off blocklists.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Attackers who want to get past multifactor authentication (MFA) use three kinds of tools. Phishing managers such as GoPhish send the lures. Adversary-in-the-middle (AiTM) proxies such as Evilginx2 and Mitmproxy sit between the user and the real login page and capture session cookies. Tunnel relays such as Ligolo-ng then carry the attacker into internal networks. This report looks at the infrastructure behind each stage, and at three exposed operator workbenches.

1. **A cloned login kit is built to send keystrokes before the user submits.**
   On Host A in the Philippines (address withheld, because a red-team exercise cannot be excluded), a 247-file open directory holds a static login cloner and a seven-line keystroke script. The script is designed to send typed text to the server when the user changes field or closes the tab. Evilginx2 and Cobalt Strike directories sit on the same server. Who operates it, and whether it is an authorized red-team host, is unknown.

2. **Two unlinked workbenches hold M365 enumeration scripts and noVNC client files.**
   Host C in Indonesia (832 files) holds scripts written to check whether Microsoft 365 accounts exist and to test IMAP logins. Host B in India (117 files) pairs a confirmed Mitmproxy listener with a noVNC browser-viewer client. Replaying cookies from Host B would also need a browser, a VNC server and a WebSocket bridge. None was observed.

3. **Recently active Ligolo-ng relays favour default port 11601 and HTTPS 443.**
   Of 347 recently active Ligolo-ng relays on 310 IPs in 42 countries, 37.2% listen on TCP 11601 and 27.1% on TCP 443. Global records hold 938 Ligolo-ng servers in 54 countries. Eleven relays carry third-party feed labels that link them to YoroTrooper-tagged infrastructure.

Three Decisions for Leaders

1. **Require origin-bound passkeys at sign-in.** W3C WebAuthn passkeys with device-compliance Conditional Access stop reverse-proxy and remote-viewer lures at the login step. Add Entra ID Token Protection and DBSC pilots for protection after sign-in.
2. **Block default Ligolo-ng ports and hunt non-browser tunnels.** Deny outbound TCP 11601 to 11610. Correlate risky Entra ID sign-ins with non-browser TLS or WebSocket connections from the same user within 24 hours.
3. **Keep blocklists for corroborated hosts only.** Commercial proxy networks, awareness-training platforms, appliance certificates and stock open-source file hashes do not belong on blocklists. Keep single-banner listeners and dual-use workspaces on watchlists until a lure or egress signal appears.

The Study in Numbers

Number	What it is
2,750	Recently active phishing, proxy and tunnel listeners on 2,212 IPv4 hosts in 69 countries
26,329	Global banner records for GoPhish (23,207), Mitmproxy (1,877), Ligolo-ng (938) and EvilGoPhish (307)
48	Listeners with commercial proxy (44) or awareness-training (4) certificate names, held out of blocklists
136,742 bytes	Size of the log file beside the keystroke receiver on Host A (1,310 lines, contents not established)
1,196	Files on three exposed workbenches: 247, 117 (Host B) and 832 (Host C)
347	Recently active Ligolo-ng relays on 310 IPs in 42 countries
3	Published indicators: custom script hashes from Host A (selected set of 28; addresses withheld)

How to use this report.

Security and identity leaders can read Chapters 1, 4 and 7. SOC engineers and threat hunters should start with Chapters 3, 5, 8 and 9. CTI analysts will find the infrastructure detail in Chapters 2 to 6.

02

CHAPTER 02

The Landscape: Phishing, Proxy and Tunnel Listeners

IN SHORT

Most phishing and proxy listeners on the internet are unconfirmed. Among those that are confirmed, 48 carry the certificate names of a commercial proxy network or an awareness-training vendor, so raw listener feeds are not safe to block.

INSIDE THIS CHAPTER

Three stages, three tool families 07

Global records and recent activity 07

How the 2,750 listeners break down 08

GoPhish: port 3333 and shared certificates 08

Mitmproxy: a commercial proxy collision 09

02 The Landscape: Phishing, Proxy and Tunnel Listeners

Three stages, three tool families

As an analytical model, the tools in this report map to three stages of a post-MFA intrusion. The model describes tool roles; it does not claim one actor uses all three.

Stage	Tools	What they do	MITRE ATT&CK
Lure delivery	GoPhish, EvilGoPhish	Send phishing campaigns and track clicks	(delivery)
Credential and session interception	Evilginx2, Mitmproxy	Proxy the real login page and capture cookies	T1557, T1539
Network pivoting	Ligolo-ng	Tunnel Layer-3 traffic into internal subnets	T1572

Global records and recent activity

Global records count every banner on file for each framework. The recently active set is the smaller group that answered within the observation window. It includes 1,200 GoPhish and 1,200 Mitmproxy listeners (a capped selection), all 347 recently active Ligolo-ng relays and 3 FRP control panels as a baseline.

Framework	Global records (countries)	Recently active (IPs / countries)	Main ports in the active set	Top hosting providers in the active set
GoPhish	23,207 (119)	1,200 (1,080 / 62)	3333 (764, 63.7%), 443 (142), 80 (32), 8443 (31), 3000 (16), 21 (11)	DigitalOcean (137), Amazon.com (127), OVH SAS (112), Microsoft (73), Google (54)
Mitmproxy	1,877 (46)	1,200 (820 / 38)	32300 (304, 25.3%), 8081 (102), 8080 (80), 8443 (57), 8888 (19), 443 (19)	ALFA TELECOM (190), Hangzhou Alibaba (145), Serverel (114), Hetzner (55), DigitalOcean (49)
Ligolo-ng	938 (54)	347 (310 / 42)	11601 (129, 37.2%), 443 (94, 27.1%), 80 (20), 8443 (14), 8080 (14), 2083 (12)	OVH SAS (33), Microsoft (28), DigitalOcean (26), UpCloud (22), Amazon.com (21), Contabo (21)
EvilGoPhish	307 (32)	Not in the active set	443, 8443, 3333 (local API)	Cloud VPS; source found on Host A
FRP (baseline)	n/a	3 (3 / 3)	7500 (3)	Snowd Security, Hangzhou Alibaba, Unesty (1 each)

The 2,750 listeners sit on 2,212 IPv4 hosts in 69 countries and 462 autonomous systems.

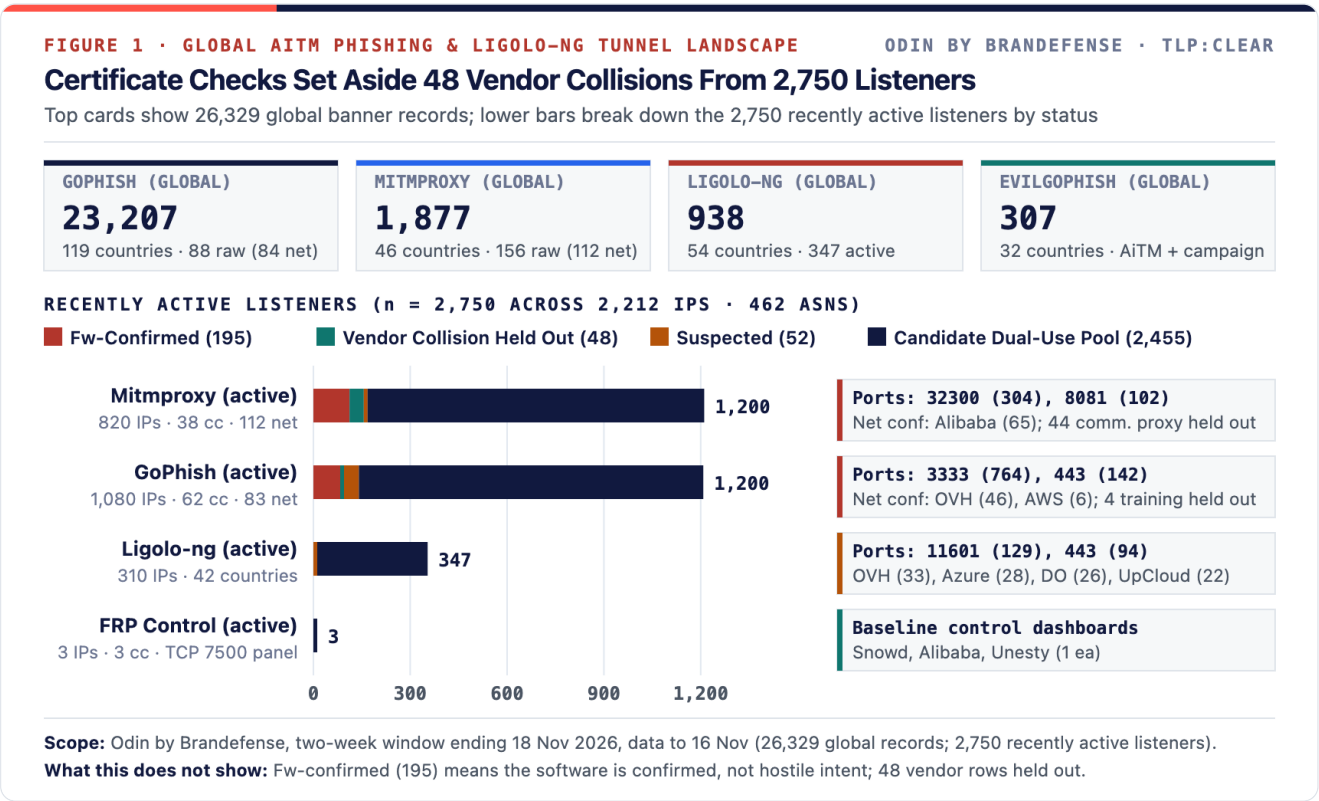


Figure 1: Certificate checks set aside 48 vendor collisions from 2,750 listeners

How the 2,750 listeners break down

"Framework confirmed" names the software. It does not mean hostile intent.

Status	Listeners	What it contains	Handling
Watchlist, framework confirmed	193	110 Mitmproxy, 72 GoPhish web or admin listeners, 11 FTPS rows reusing a GoPhish certificate	Watch; block only after a lure or egress signal
Watchlist, vendor certificate name	48	44 commercial proxy and 4 awareness-training certificate names, chain trust not verified	Watch; verify the certificate chain before excluding or blocking
Suspected	50	46 GoPhish, 1 Mitmproxy, 3 Ligolo-ng	Send to threat hunting
Candidate	2,452	1,067 GoPhish, 1,043 Mitmproxy, 339 Ligolo-ng, 3 FRP	Hold; do not block
Held out	7	2 FortiGate appliance certificates, 5 hits in a U.S. public-sector network	Keep off blocklists

Figure 1 counts the 7 held-out rows inside their software status, so its legend shows 195 confirmed, 52 suspected and 2,455 candidates.

GoPhish: port 3333 and shared certificates

Of the 1,200 GoPhish listeners, 764 (63.7%) expose the default admin port 3333. Of those, 756 are candidates and 8 are confirmed.

Four confirmed rows carry the certificate name of a commercial security-awareness training vendor. Ownership and certificate-chain trust were not verified, so they are held out as possible collisions. That leaves 83 confirmed GoPhish rows: 72 web or admin listeners plus 11 FTPS rows.

The 72 web or admin listeners run on 443 (30), 8443 (12), 3000 (11), 3333 (8) and other web ports (11). OVH SAS hosts 46 of the 83 rows (55.4%), followed by Amazon.com (6), DigitalOcean (5) and netcup GmbH (1).

The 11 rows on TCP port 21 are FTPS daemons on OVH SAS. They reuse the same self-signed provider-default certificate that GoPhish uses on 443 and 8443 on the same cloned VPS hosts (33 rows share it). This is certificate reuse across services, not GoPhish speaking FTP.

Mitmproxy: a commercial proxy collision

Mitmproxy candidates cluster on high proxy ports: 32300 (304, on ALFA TELECOM s.r.o. and Serverel Inc.), 8081 (102) and 8080 (78 candidates plus 2 confirmed).

Of 156 confirmed Mitmproxy rows, 44 (28.2%) carry certificate names that match a commercial proxy network. Chain trust was not verified, so they stay on the watchlist as possible collisions. They make up all 37 of DigitalOcean's confirmed Mitmproxy rows and 7 of 8 on Constant/Vultr.

Removing the 44 leaves 112 confirmed rows, or 110 after setting aside 2 AWS listeners with `CN=FortiGate` appliance certificates. Hangzhou Alibaba Advertising hosts 65 of the 112 (58.0%). Of these, 64 share two private-range certificate names, `172.29.97.219` and `172.18.9.199`, on 6xxx ports. Smaller groups sit on CV Griya Cloud Server (7), Contabo (7), Hetzner (6), AROSSCLOUD (1), HostPapa (1), Psychz Networks (1) and the small provider hosting Host B.

03

CHAPTER 03

A Keystroke Streamer Beside Evilginx2 and Cobalt Strike

IN SHORT

An open directory on Host A holds three toolsets on one server: a static login cloner with a keystroke script, an Evilginx2 installation, and Cobalt Strike files. A 136,742-byte log file sits beside the keystroke receiver; its contents are not established.

INSIDE THIS CHAPTER

Why capture keystrokes before submit 11

The static cloner kit 11

Evilginx2 and Cobalt Strike on the same server 12

File hashes for hunting 13

03 A Keystroke Streamer Beside Evilginx2 and Cobalt Strike

Why capture keystrokes before submit

A static phishing clone only receives a password when the user clicks Submit. If the user types part of a password, notices the wrong domain and closes the tab, a normal clone gets nothing. The kit described here is designed to close that gap.

On 16 November 2026, Odin by Brandefense observed an open directory on Host A (Philippines). It exposes 247 files across 591 directories (4,258,037 bytes). Figure 2 maps how the parts fit together.

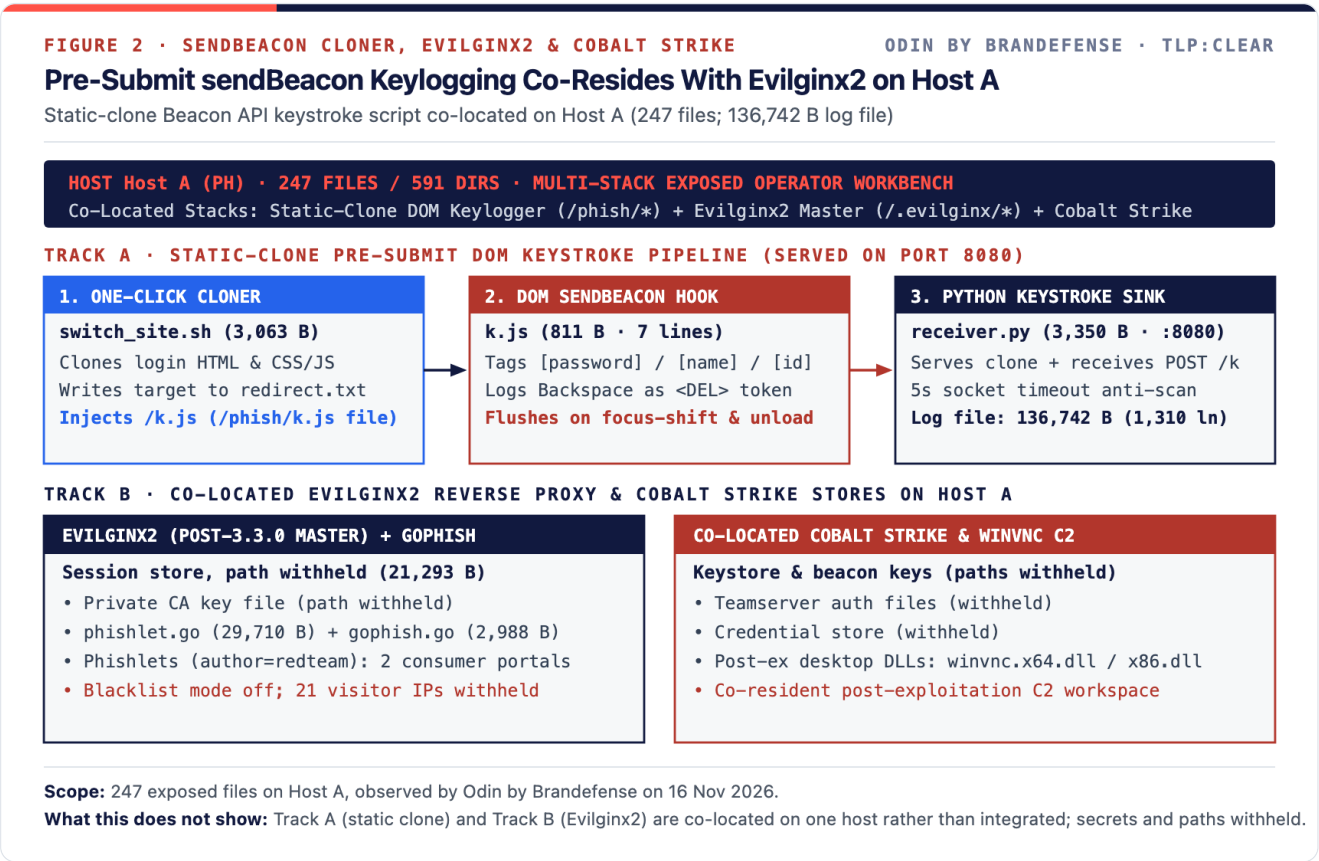


Figure 2: Pre-submit sendBeacon keylogging co-resides with Evilginx2 on Host A

The static cloner kit

The kit has three files under `/phish/`. The table describes each one by structure only. No script content is reproduced.

File	Size and lines	Role
`switch_site.sh`	3,063 bytes, 87 lines	Bash cloner. Downloads a target login page and its assets with a `Chrome/133.0.0.0` User-Agent, inserts a script tag that loads `k.js`, and writes the real login URL to `redirect.txt` (51 bytes)
`k.js`	811 bytes, 7 lines	Keystroke script. Labels each input field by type, name or id, marks deletions with a text token, and sends the buffer to the server path `/k`
`receiver.py`	3,350 bytes, 89 lines	Python web server on port 8080. Serves the clone and the script, accepts posts to `/k`, sets a 5-second socket timeout and a wildcard CORS header

The keystroke script uses the browser Beacon API (`sendBeacon`). A browser queues a Beacon request in the background and sends it even while the tab is closing. The request is a small `text/plain` POST. The script falls back to a keep-alive `fetch` only when the Beacon API is missing.

The script sends its buffer at two moments: when the user moves to a different field, and when the page unloads. That is why the kit can collect partial input from a user who never submits.

The cloner fetches every asset path without logging in. On single-page apps, this leaves a trace. Three files on the server with image names hold the same 27,793-byte HTML landing page instead of an image.

`receiver.py` carries comments in Simplified Chinese about protection against slow sockets. That is an artifact of whoever wrote the toolkit. It is not attribution of the operator.

Evilginx2 and Cobalt Strike on the same server

The same open directory holds a full Evilginx2 tree and a Cobalt Strike workspace. Paths and secrets are withheld.

Component	What is exposed
Evilginx2 source	`/evilginx2/` checkout built with Go 1.22. Its `CHANGELOG` opens with `# Unreleased` above `# 3.3.0`, so it is a post-3.3.0 master branch. Stock `core/phishlet.go` (29,710 bytes) and `core/gophish.go` (2,988 bytes)
Evilginx2 runtime	A BuntDB session store (21,293 bytes, 196 lines), a private root CA key file (1,679 bytes), a visitor blacklist with 21 addresses (withheld), and a config file (4,897 bytes)
Evilginx2 configuration	Binds its own public address (withheld) as `external_ipv4`, sets `blacklist.mode: off`, and lists two active lure paths
Phishlets	Two custom YAML phishlets (3,275 and 5,811 bytes) declare `author: 'redteam'` and target two Chinese consumer portals. Neither injects the keystroke script
Cobalt Strike	A keystore, teamserver authorization files, a beacon key pair, a credential store, the launcher `source-common.sh`, and TightVNC post-exploitation DLLs `winvnc.x64.dll` and `winvnc.x86.dll`

The static cloner and Evilginx2 are co-located, not integrated. The phishlets do not load `k.js`. The `author: 'redteam'` tag and the unknown authorization mean a red-team exercise cannot be excluded.

File hashes for hunting

Hashes were recorded at acquisition.

File	Size	SHA-256	Use
`/phish/k.js`	811 bytes	`dc89160b8501065099a69bc6a8f34ce085774878387d071f4e249eebb57a757b`	Block: custom keystroke script
`/phish/receiver.py`	3,350 bytes	`629fd5033240e77684003b632c3b247671ac2043a4e87ae7f420feb836551a45`	Block: custom clone server and `/k` receiver
`/phish/switch_site.sh`	3,063 bytes	`7da269d7b5b55d409b71702dc53d4329bbeb0afb0540ab3d98896c3625836c54`	Block: custom portal cloner
`/evilginx2/core/phishlet.go`	29,710 bytes	`00d134e5be68c83a13c35101cd26aca6efe6af608144c35add5edf3a4139f336`	Context only: stock upstream source
`/evilginx2/core/gophish.go`	2,988 bytes	`6097864f26657ac5add013af5e17f508678dec3eb57548dbf37ea09e2a02a78c`	Context only: stock upstream source

The two Evilginx2 source files are unmodified upstream code. Do not block them on their own.

CHAPTER 04

M365 Enumeration Scripts and a Co-Hosted noVNC Client

IN SHORT

Two separate, unlinked servers sit on the watchlist. Host C holds scripts written to enumerate Microsoft 365 accounts and test logins. Host B pairs a confirmed Mitmproxy listener with noVNC client files, which are one part of a possible replay setup.

INSIDE THIS CHAPTER

Why replay location matters 15
Host C: an AI pentest workspace with M365 recon scripts 16
Host B: Mitmproxy beside a noVNC client 16
What stops this kind of attack 17

04 M365 Enumeration Scripts and a Co-Hosted noVNC Client

Why replay location matters

When an attacker steals a session cookie through a cloud proxy and replays it from a laptop or VPN, identity systems often notice. Continuous Access Evaluation and risk-based Conditional Access react to the change of IP, network and location.

One way around this is to replay the cookie from the proxy server itself, through a remote browser. Host B shows one part of that setup. Figure 3 shows what was seen and what would still be needed.

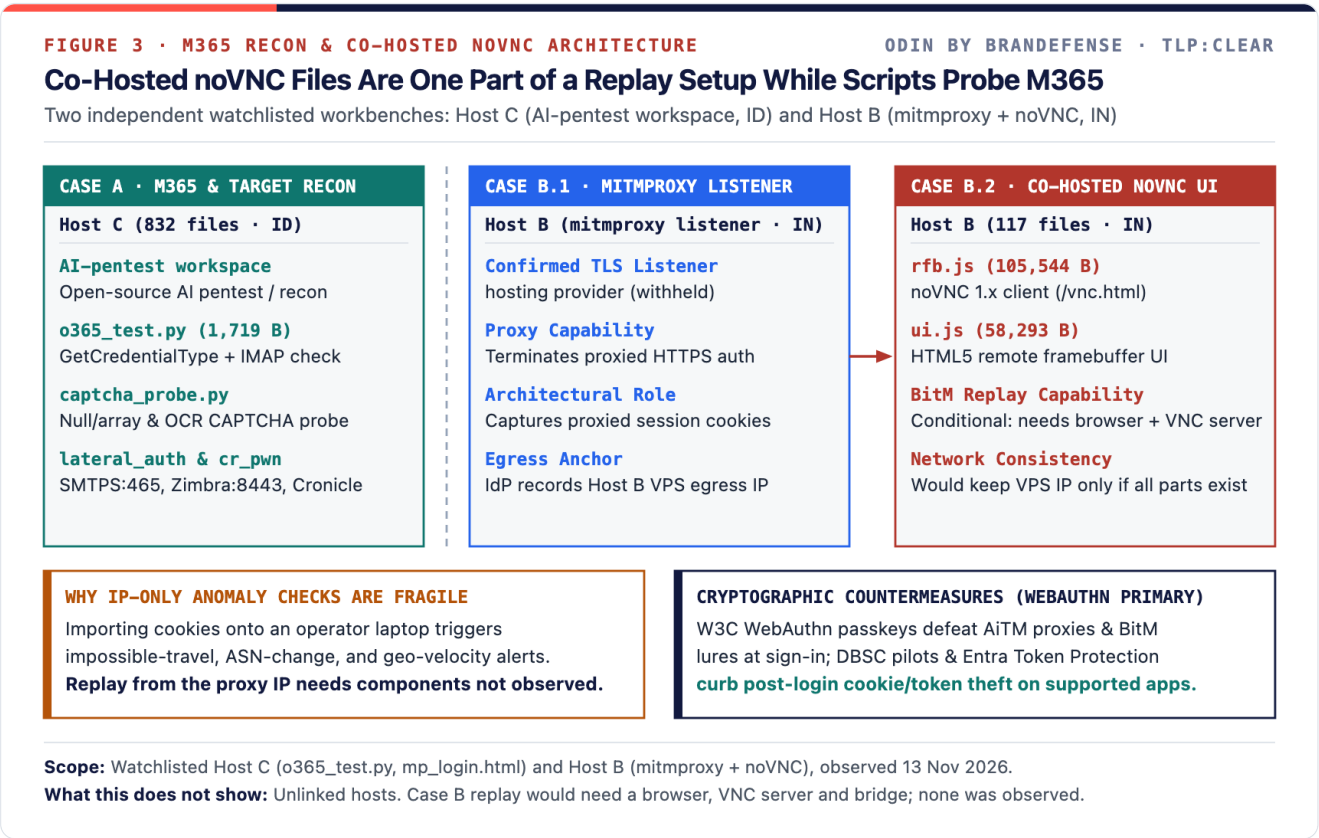


Figure 3: Co-hosted noVNC files are one part of a replay setup while scripts probe M365

Host C: an AI pentest workspace with M365 recon scripts

On 13 November 2026, Host C in Indonesia exposed 832 files (1,280,209 bytes). The directory was created by an open-source AI pentest workspace. That software is dual-use, and the scripts carry diagnostic strings such as `zz-diag` and `auth-test`. No result on any target is verified, so Host C stays on the watchlist and its address is withheld.

File	Size and lines	SHA-256	What it was written to do
`/o365_test.py`	1,719 bytes, 26	`24146b8cbbafc57995e77c7a65e799fc471374b27c06a7ac931a7282c2c9cfaa`	Query M365 `GetCredentialType` for 4 accounts and test IMAP login on port 993
`/mp_login.html`	12,573 bytes, 109	Withheld (copied third-party page)	Cloned merchant billing login page (also at `/mplogin.html`)
`/captcha_probe.py`	2,212 bytes, 51	`e0f5c20fa788887402f33aefbd98b5e6a5d33c2dc5af529adf72dcea7bbca871`	Test null and array CAPTCHA parameters and Tesseract OCR thresholds
`/lateral_auth.py`	2,153 bytes, 35	`e5c7d66e25771dded8a94810bd4ffe25ca05af325c30c666e29543a83d606f02`	Test credential reuse across SMTPS, Zimbra and Jenkins

Account enumeration. `o365_test.py` is titled as an account-existence check. It was written to call the public `login.microsoftonline[.]com/common/GetCredentialType` endpoint and print whether each account exists, its throttle status, domain type and preferred credential. The `isFidoSupported` flag in its request is a standard client field, not a tenant check. A second loop was written to test basic-auth IMAP against `outlook.office365[.]com:993`. Outcomes are not verified.

Lures and CAPTCHA tests.

No Microsoft 365 phishing page sits on Host C. The lures are merchant billing login clones and a dark-theme Jenkins login (`/login.html`, 2,337 bytes). `captcha_probe.py` was written to test edge-case CAPTCHA inputs against a merchant endpoint.

Credential reuse and scheduler scripts.

`lateral_auth.py` was written to test reuse on SMTPS port 465, the Zimbra preauth service on 8443 and the Jenkins login check. Two more scripts, `cr_pwn.py` and `cr_rce_proof.py`, were written to modify jobs on a merchant's Cronicle scheduler with a callback to Host C. Target-side outcomes are not verified.

Host B: Mitmproxy beside a noVNC client

Host B is a VPS at a small hosting provider in India. It runs a confirmed Mitmproxy TLS listener. On 13 November 2026, the same IPv4 address also served an open HTTP directory of 117 files (917,101 bytes). It is a complete noVNC 1.x HTML5 remote-framebuffer client, with a 2020 copyright.

File	Size	Note
`/core/rfb.js`	105,544 bytes, 2,988 lines	Stock noVNC core
`/app/ui.js`	58,293 bytes, 1,715 lines	Stock noVNC interface
`/core/websock.js`	10,593 bytes	Stock noVNC WebSocket layer
`/vnc.html`, `/vnc_lite.html`, `/utils/novnc_proxy`	n/a	Client pages and proxy helper

The listing shows static client files next to the proxy. It does not show a live VNC session or any cookie import. Replay from the same IP would also need a running browser, a VNC server, a WebSocket bridge and a cookie transfer. None of these was observed, and a noVNC client alone does not replay cookies. No login lure or victim session was found, so Host B stays on the watchlist and its address is withheld.

What stops this kind of attack

If an operator did run a browser on the proxy VPS, IP-consistency and impossible-travel rules would not fire. The main defence is at sign-in: origin-bound passkeys. A passkey is tied to the real site's origin, so it will not sign in through a reverse proxy or a remote viewer. A proxy can also strip DBSC headers, so DBSC alone is not enough at sign-in.

Control	When it acts	Scope and limits
W3C WebAuthn / FIDO2 passkeys	At sign-in (primary AiTM and browser-in-the-middle defence)	Origin-bound challenge defeats reverse-proxy and remote-viewer phishing
W3C DBSC	After sign-in, at cookie refresh	Work-in-progress Editor's Draft; browser pilots only
Entra ID Token Protection	After sign-in, token binding	Native apps on Windows, iOS/iPadOS and macOS, plus limited browser preview support

05

CHAPTER 05

Ligolo-ng Tunnel Relays and the YoroTrooper Cluster

IN SHORT

Ligolo-ng gives an attacker a routed tunnel into an internal network through one encrypted connection. Recently active relays sit on mainstream cloud and VPS networks and favour the default port 11601 or HTTPS 443.

INSIDE THIS CHAPTER

How Ligolo-ng works 19

Ports 20

Countries and hosting 20

YoroTrooper overlap 20

05 Ligolo-ng Tunnel Relays and the YoroTrooper Cluster

How Ligolo-ng works

Older pivoting through SOCKS proxies forces operators to wrap each tool in `proxychains`. That approach hooks sockets through `LD_PRELOAD`, handles ICMP and UDP poorly, and creates many short connections.

Ligolo-ng works differently. The relay on the operator side creates a virtual Layer-3 TUN interface (through `ip tuntap`, its own `interface_create` command, or `wintun.dll` on Windows). It handles the packets in Google's userland gVisor netstack.

The relay then tells an unprivileged Go agent on the target host to open ordinary TCP and UDP sockets. All traffic travels over a single TLS or WebSocket stream (MITRE ATT&CK T1572). Figure 4 summarises ports, hosting and the YoroTrooper overlap.

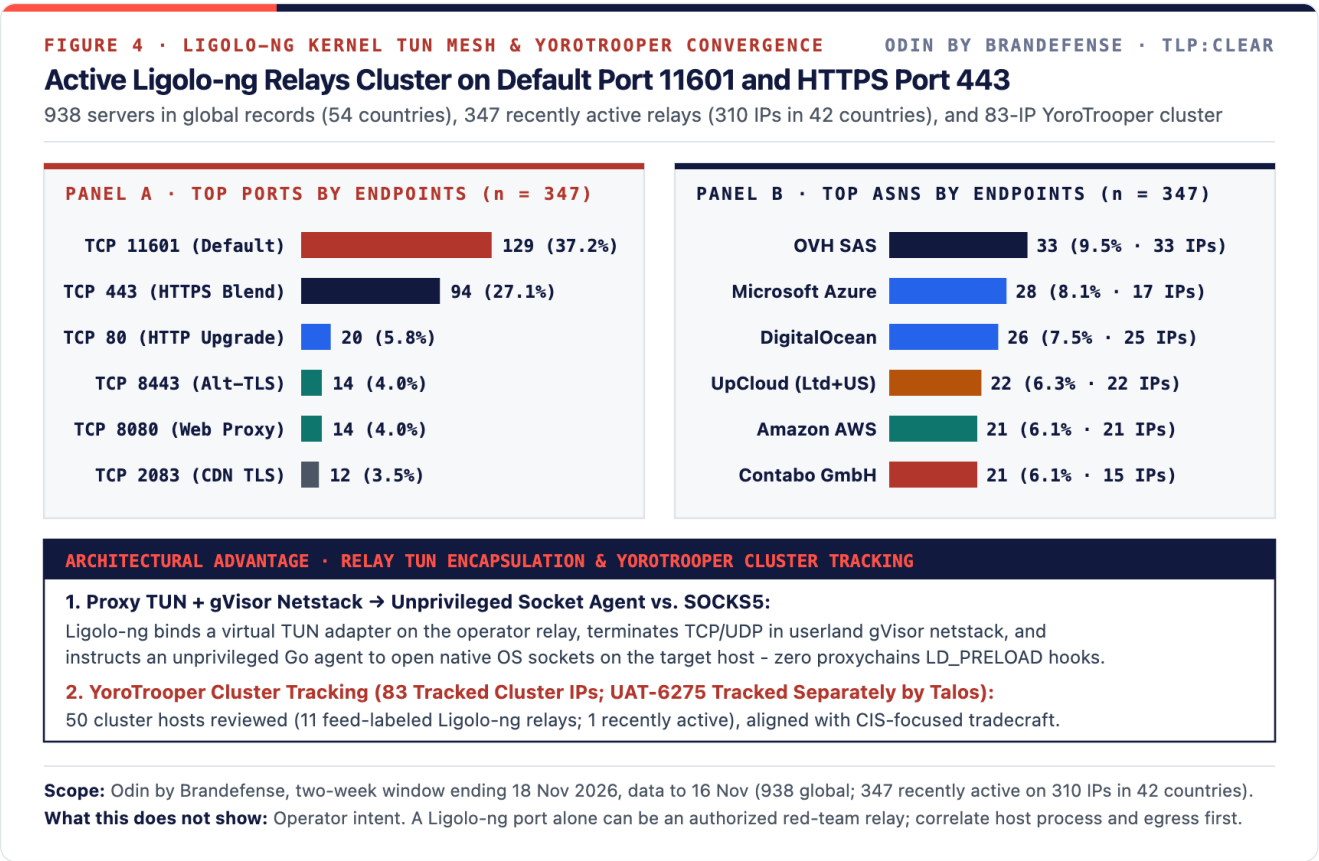


Figure 4: Active Ligolo-ng relays cluster on default port 11601 and HTTPS port 443

Ports

Port	Relays (share)	Profile	Typical hosting
TCP 11601	129 (37.2%)	Unchanged default relay port	OVH SAS (IN, FR), DigitalOcean (SG, DE), Contabo (DE), Hangzhou Alibaba (CN)
TCP 443	94 (27.1%)	HTTPS port to pass strict firewalls, with Let's Encrypt or self-signed TLS	Microsoft (US), Amazon.com (US, FR, KR), DigitalOcean (SG), Google (BE)
TCP 80	20 (5.8%)	Plain HTTP or WebSocket upgrade, or ACME HTTP-01 challenge	Amazon.com (CA), DigitalOcean (US), Contabo (FR), WorldStream (NL)
TCP 8443 and 8080	14 (4.0%) each, 28 (8.1%) combined	Alternate HTTPS and web-proxy ports, often paired on one relay	Microsoft (US), Oracle (CL), Yandex.Cloud (RU), Google (US)
TCP 2083, 11602 to 11607, 4445	19 (12 on 2083, 5 sequential, 2 on 4445)	CDN-compatible TLS port, multi-agent ports, a custom port	Microsoft (2083), HOSTING INDUSTRY (11601 to 11607), Contabo (4445)

Countries and hosting

The United States (94), India (40), France (29), Germany (27), Russia (21) and Singapore (19) hold 230 of the 347 relays (66.3%). The relays sit inside mainstream cloud and European VPS networks, so their outbound TLS blends with ordinary SaaS traffic.

Hosting provider (ASN)	Relays (share)	Distinct IPs	Pattern
OVH SAS (AS16276)	33 (9.5%)	33	Single-listener VPS relays in India, France, Singapore and Canada
Microsoft (AS8075)	28 (8.1%)	17	Azure relays pairing 443, 8443 and 2083 on shared IPs
DigitalOcean (AS14061)	26 (7.5%)	25	Single-port droplets on 11601 and 443 in US, SG, DE and GB
UpCloud (AS202053 / AS25697)	22 (6.3%)	22	14 on UpCloud Ltd and 8 on UpCloud USA, mostly on 11601
Amazon.com (AS16509 / AS14618)	21 (6.1%)	21	Single-listener EC2 relays, mostly 443 and 80
Contabo (AS51167)	21 (6.1%)	15	Multi-port European VPS nodes on 11601, 443, 80 and 4445

None of the 347 relays is confirmed: 5 are suspected and 342 are candidates. At the port and banner level, a Ligolo-ng relay looks the same as an authorized red-team relay. Five of the hits sit inside a U.S. public-sector network and are held out, since they may be banner collisions, authorized testing or exposure on the victim side.

YoroTrooper overlap

Cisco Talos documents YoroTrooper espionage campaigns against organisations in the Commonwealth of Independent States (CIS), Turkey and Europe. The group uses spear-phishing, credential harvesting, Python information stealers and reverse shells.

In a 2026 assessment of the Kazakhstan-nexus group UAT-6275, Talos noted the two may sit in a broader CIS-targeting cluster. Talos still tracks UAT-6275 separately, because it saw no direct indicator or tool overlap.

Separately from Talos, third-party feeds tag 83 IPv4 addresses as YoroTrooper-related infrastructure (tags NIK C2, Chvost, Laplas, PickleShell and SSDPShell). Of 50 of these hosts that we reviewed, 11 carry feed labels for Ligolo-ng, dated 14 to 15 October 2026. They sit on DigitalOcean (6), MTS PJSC (3), Contabo (1) and Enterprise Cloud Ltd. (1).

Only 1 of the 11 appears among the recently active relays. The other 10 showed no recent Ligolo-ng banner. Feed tags are not attribution, so these addresses are not published.

CHAPTER 06

Keeping Vendors and Dual-Use Tools off Blocklists

IN SHORT

Raw feeds label commercial proxies, training platforms and appliances as hostile. Separating them, and separating watchlists from blocklists, keeps enterprise controls accurate.

06 Keeping Vendors and Dual-Use Tools off Blocklists

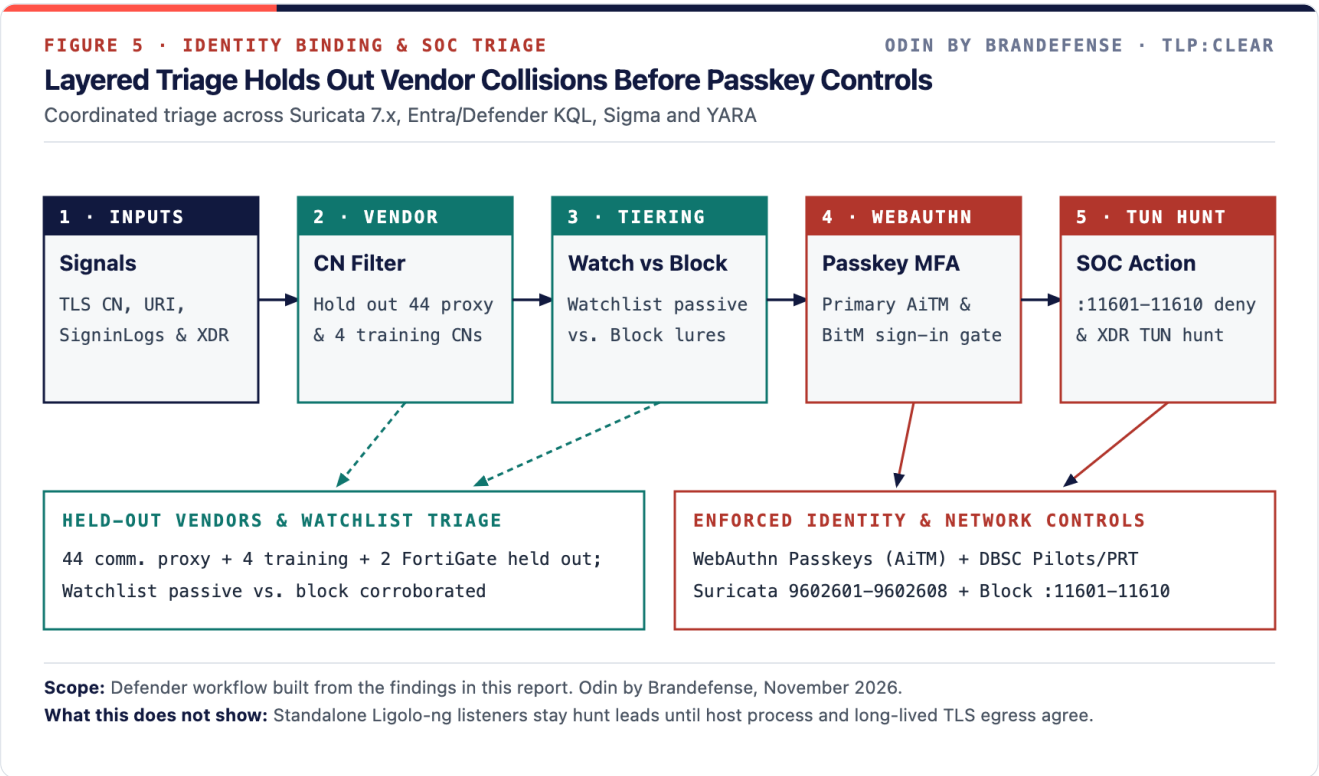


Figure 5: Layered triage holds out vendor collisions before passkey controls

Seven rules shape every number and indicator in this report. Each one is also a rule a SOC can apply to its own feeds.

1. **Awareness-training certificates are held out.** Four confirmed GoPhish rows on 2 AWS IPs carry a training vendor's certificate name. The 83 other confirmed GoPhish rows (84 in global records) stay on the watchlist. Unverified GoPhish banners (23,119 globally, 1,113 in the active set) are held.
2. **Commercial proxy and appliance certificates are held out.** Of 156 confirmed Mitmproxy rows, 44 (37 DigitalOcean and 7 Constant/Vultr) carry commercial proxy certificate names and 2 carry `CN=FortiGate`. The other 112 (110 without FortiGate) stay on the watchlist. Unverified Mitmproxy rows (1,721 globally, 1,044 in the active set) are held.
3. **Authorized testing servers and public-sector hits are excluded.** One authorized out-of-band testing server is left out of all hostile counts. The 5 Ligolo-ng hits in a U.S. public-sector network are excluded from adversary VPS claims and anonymised.
4. **Ligolo-ng listeners stay hunt leads.** The 347 relays (342 candidates, 5 suspected) are never promoted to a blocklist without endpoint egress evidence.
5. **Stock open-source files are never blocked alone.** The noVNC (`rfb.js`, `ui.js`) and Evilginx2 (`phishlet.go`, `gophish.go`) hashes only show co-hosting context. Blocking them alone would break legitimate Proxmox VE, OpenStack and VDI consoles.
6. **Red-team provenance is considered.** Host A holds a keystroke receiver, an Evilginx2 store and Cobalt Strike files together. Its authorization is unknown and a red-team exercise cannot be excluded, so its address is

withheld. Its custom script hashes stay published as tool indicators. Host B and Host C stay on watchlists with addresses withheld.

7. **Victim and third-party data are withheld.** The 21 visitor IPs in the Evilginx2 blacklist, the consumer portal domains, the Cronicle server address, merchant paths, API keys, tested credentials and all non-confirmed listener addresses are withheld.

In total, 24,840 unverified dual-use records are kept off blocklists. Vendor certificate names should only lead to exclusion when the certificate chain is verified as trusted. A self-signed certificate that copies a vendor name is a reason to look closer, not to exclude.

07

CHAPTER 07

What Defenders Should Do

IN SHORT

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

07 What Defenders Should Do

Identity

- [ ] Require origin-bound W3C WebAuthn passkeys with device-compliance Conditional Access as the primary control against AiTM proxies and noVNC lures.
- [ ] Turn off legacy IMAP and SMTP basic authentication.
- [ ] Add Entra ID Token Protection on supported native apps (Windows, iOS/iPadOS, macOS, plus limited browser preview support) and pilot browser DBSC on supported sites.

Network

- [ ] Block outbound traffic to TCP 11601 to 11610.
- [ ] Run the Sentinel query that links risky sign-ins to tunnel connections within 24 hours.
- [ ] Test the eight Suricata signatures in a lab, then deploy them at the perimeter, in EASM tooling and on decrypted proxy taps. Exclude sanctioned VDI consoles from SIDs 9602604 and 9602608.
- [ ] Alert on `text/plain` POSTs to a `/k` path whose body starts with a bracketed field label.

Endpoint

- [ ] Test, then deploy, the five Sigma rules for Ligolo-ng agents on Linux and Windows, unsanctioned `wintun.dll` loads, and noVNC or Evilginx2 processes on cloud VMs.

Triage

- [ ] Keep commercial proxy, awareness-training and FortiGate certificate rows, port-3333 pools and stock open-source hashes off blocklists.
- [ ] Keep passive listeners and dual-use workspaces on watchlists until a lure or egress signal corroborates them.

08

CHAPTER 08

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 Sentinel query works 28

How the Sigma rules work 29

How the YARA rules work 29

How to sort a host 30

08 Detection Rules

The full rules are in Appendix B. No rule was run on a live engine; test each one in your own environment before deployment.

Rule file	What it catches
`suricata_aitm_sendbeacon_and_ligolo.rules`	Suricata 7: eight signatures for keystroke beacons, exposed Evilginx2 and noVNC, M365 enumeration bursts, Ligolo-ng ports, non-browser WebSocket upgrades and VNC greetings
`sentinel_aitm_token_replay_and_ligolo_hunt.kql`	Microsoft Sentinel and Defender XDR: tunnel connections linked to risky Entra ID sign-ins by the same user
`sigma_evilginx_novnc_and_ligolo_agent.yml`	Sigma (5 rules): Ligolo-ng agents, unsanctioned `wintun.dll` loads, noVNC and Evilginx2 or cloner processes
`yara_aitm_keylogger_and_phish_toolkits.yar`	YARA (4 rules): the keystroke script, the receiver, the cloner, and the M365 and credential-reuse scripts

How the Suricata signatures work

SID	Looks for	Where to deploy
9602601	Outbound `text/plain` POST to a path ending in `/k`, whose body starts with a short bracketed field label	Perimeter or decrypted web proxy
9602602	Request URIs for exposed Evilginx2 runtime, phishlet, core or database folders	EASM and hunting
9602603	Inbound script that contains the Beacon API name, a deletion token and a page-unload event name close together	EASM and decrypted proxy
9602604	Inbound HTTP 200 response with noVNC client pages or `core/rfb.js`	EASM and hunting; exclude sanctioned VDI
9602605	Four or more `GetCredentialType` POSTs from one source in 120 seconds, from a client without the browser `sec-fetch-site` header	SSL-inspecting proxy or cloud tap
9602606	Outbound TLS client hello to ports 11601 to 11610	Perimeter; a port-policy alert
9602607	Outbound WebSocket upgrade without `Origin` or `Cookie` headers, rate-limited per source	Perimeter; a hunting rule
9602608	Inbound raw VNC `RFB 003.00x` server greeting	EASM and hunting; exclude sanctioned VDI

SID 9602605 tolerates both compact JSON and Python `json.dumps()` spacing in the request body. It excludes browsers by the missing header, not by user agent.

How the Sentinel query works

The query looks back 14 days. It first builds a list of successful Entra ID sign-ins that are risky: medium or high risk, risk events such as `anomalousToken`, or a sign-in from a high-concentration relay network on a non-compliant device. The relay network list holds six ASNs, including OVH (16276), Contabo (51167) and UpCloud (202053 and 25697).

It then takes device connections to ports 11601 to 11610, or to 443, 8443, 2083, 8080 and 4445 from non-browser processes whose command line names Ligolo-ng flags or `wintun`. It joins the two on the user account and keeps a sign-in only if the tunnel started within 24 hours of it.

The output labels each row. "Identity-correlated tunnel" means a risky sign-in matched. "Port policy only" means a connection to a default Ligolo-ng port with no matching sign-in. Connection counts are deduplicated so the join does not inflate them.

How the Sigma rules work

Rule	Fires when
1. Ligolo-ng Linux agent	A Linux process starts with a connect argument (`host:port`, `https://` or `wss://`) under a default agent name, or a renamed binary carries Ligolo flags such as `-ignore-cert`, `-accept-fingerprint`, `-retry` or a :1160x port
2. Ligolo-ng Windows agent	The same logic on Windows; `curl --connect-timeout` and `openssl s_client` are excluded
3. Wintun DLL load	`wintun.dll` loads into a process that is not an approved VPN client executable (exact paths for WireGuard, Tailscale and Cloudflare WARP; add your own)
4. noVNC on a cloud VM	noVNC or `websockify` starts on cloud or VPS hosts
5. Evilginx2 or cloner on a cloud VM	Evilginx2 or a `switch_site.sh` cloner starts on cloud or VPS hosts

A renamed agent that dials plain `host:443` without Ligolo flags will not trip rules 1 and 2. The Sentinel query covers that case through network correlation.

How the YARA rules work

Each YARA rule is limited to files under 64 KB and needs all of its strings to match.

YARA rule	Targets
`ODIN_AiTM_SendBeacon_PreSubmit_Keylogger_JS`	The 811-byte keystroke script
`ODIN_AiTM_CORS_Keystroke_Receiver_Python`	The Python clone server and `/k` receiver
`ODIN_AiTM_OneClick_Login_Site_Cloner_Shell`	The Bash portal cloner
`ODIN_M365_GetCredentialType_And_Lateral_Auth_Toolkit`	`o365_test.py` or `lateral_auth.py`

How to sort a host

The triage logic never blocks on a file path alone. It needs reviewed file content, or live interception or egress evidence. It checks for exposed hostile files before it applies any vendor exclusion, so a self-signed certificate that copies a vendor name does not get a free pass.

What the host shows	Action
Vendor certificate name with a verified trusted chain, FortiGate appliance, authorized testing server	Exclude
Vendor certificate name with unknown or failed chain trust	Watchlist
Unverified GoPhish on 3333, unverified Mitmproxy, Ligolo-ng on 11601 alone	Hold or hunt
A hit inside a public-sector network	Review only
Confirmed passive listener, co-hosted proxy and noVNC, dual-use recon workspace, workbench paths without reviewed content	Watchlist
Ligolo-ng relay with endpoint egress, keystroke receiver with reviewed content, active browser-in-the-middle	Block

Vendor names in this logic are placeholders. Supply your own allowlists, and mark a chain as trusted only when you have checked it.

09

CHAPTER 09

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Hunt 32

Context only (stock open-source files) 32

MITRE ATT&CK mapping 33

09 Indicators

Appendix A holds the 3 custom script hashes from Host A, whose address is withheld. The other 25 records in the selected set are watch, hunt or context entries described below.

Type	Count	Action
Host A (address withheld)	1	Hunt by its file hashes
Custom script hashes (SHA-256) from that host	3	Hunt
Watchlisted hosts (Host B, Host C)	2	Watch; addresses withheld
Host C script and lure hashes (SHA-256)	4	Watch
Framework-confirmed Mitmproxy and GoPhish listener profiles	9	Watch; addresses withheld
Stock open-source hashes (Evilginx2, noVNC)	4	Context only; do not block alone
Candidate Ligolo-ng relay profiles	5	Hunt egress only; addresses withheld

Hunt

Indicator	Port	What it is
`dc89160b8501065099a69bc6a8f34ce085774878387d071f4e249eebb57a757b`	n/a	`k.js` keystroke script
`629fd5033240e77684003b632c3b247671ac2043a4e87ae7f420feb836551a45`	n/a	`receiver.py` clone server and receiver
`7da269d7b5b55d409b71702dc53d4329bbeb0afb0540ab3d98896c3625836c54`	n/a	`switch_site.sh` portal cloner

Context only (stock open-source files)

SHA-256	File	Where found
`00d134e5be68c83a13c35101cd26aca6efe6af608144c35add5edf3a4139f336`	Evilginx2 `core/phishlet.go` (29,710 bytes)	Host A
`6097864f26657ac5add013af5e17f508678dec3eb57548dbf37ea09e2a02a78c`	Evilginx2 `core/gophish.go` (2,988 bytes)	Host A
`331109966c80bef620bcd09242d43dbc05fa83cb44eba439ad20148f86aa479e`	noVNC `core/rfb.js` (105,544 bytes)	Host B
`70ba76ca86ceaea27dc338bbfee4cebcfef1160812c8d410241f3667dff4864c`	noVNC `app/ui.js` (58,293 bytes)	Host B

Three of the four Host C hashes are in the Host C table in Chapter 4. The fourth, for the cloned login page `/mp_login.html`, is withheld because the page copies a third-party site. The nine watchlisted listener profiles sit on hosting providers in the United States, Vietnam, Germany, Russia and Indonesia. The five candidate relays sit on providers in the United States, China and India.

Withheld: the addresses of Host B, Host C, the nine watchlisted listeners and the five candidate relays; the 83 YoroTrooper-tagged addresses; the 21 visitor IPs; and all third-party portal, merchant and certificate names.

MITRE ATT&CK mapping

Tactic	Technique	Where it appears	Detection
Credential Access, Collection	T1557 Adversary-in-the-Middle	Evilginx2 tree on Host A; confirmed Mitmproxy listeners	SID 9602602; Sigma rule 5
Credential Access	T1539 Steal Web Session Cookie	Evilginx2 session store; Host B proxy and noVNC pairing	SIDs 9602604, 9602608; Sigma rule 4
Command and Control	T1572 Protocol Tunneling	347 recently active Ligolo-ng relays	SIDs 9602606, 9602607; Sentinel query; Sigma rules 1 to 3

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.** Findings come from a two-week observation window ending 18 November 2026, with a data cutoff of 16 November; most listeners were last seen from 12 November onward. The workbenches were observed on 13 and 16 November 2026.
- **Global records are totals, not a timeline.** The 26,329 global banner records count every record on file for each framework. They do not mean Odin watched these hosts for a long period, and they support no trend claims.
- **Confirmed is not hostile.** A framework-confirmed listener runs the software. It may be a red team, a researcher or a training platform. That is why confirmed listeners sit on watchlists.
- **Files show intent, not outcomes.** The scripts on all three workbenches show what they were written to do. No target-side result was verified, and the contents of the 136,742-byte log file are not established.
- **No campaign link.** Nothing links Host A, Host B and Host C to each other. They are three separate findings.
- **Feed tags are not attribution.** The YoroTrooper overlap rests on third-party feed labels. It is not tied to the Talos reports by a matching indicator.
- **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.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. MITRE ATT&CK T1557: Adversary-in-the-Middle. https://attack.mitre.org/techniques/T1557/ (verified 18 November 2026)
2. MITRE ATT&CK T1539: Steal Web Session Cookie. https://attack.mitre.org/techniques/T1539/ (verified 18 November 2026)
3. W3C. Beacon specification. https://www.w3.org/TR/beacon/ (verified 18 November 2026)
4. W3C. Web Authentication Level 3. https://www.w3.org/TR/webauthn-3/ (verified 18 November 2026)
5. W3C. Device Bound Session Credentials, Editor's Draft. https://w3c.github.io/webappsec-dbsc/ (verified 18 November 2026)
6. Microsoft. Token Protection in Entra ID Conditional Access. https://learn.microsoft.com/en-us/entra/identity/conditional-access/concept-token-protection (verified 18 November 2026)
7. MITRE ATT&CK T1572: Protocol Tunneling. https://attack.mitre.org/techniques/T1572/ (verified 18 November 2026)
8. nicocha30. Ligolo-ng repository. https://github.com/nicocha30/ligolo-ng (verified 18 November 2026)
9. Cisco Talos. YoroTrooper espionage campaign targets CIS, Turkey and Europe. https://blog.talosintelligence.com/yorotrooper-espionage-campaign-cis-turkey-europe/ (verified 18 November 2026)
10. Cisco Talos. Kazakhstan-nexus UAT-6275 targets CIS government organizations. https://blog.talosintelligence.com/break-the-silence-kazakhstan-nexus-uat-6275-targets-cis-government-organizations/ (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 (3) 39

B. Detection rules (4 files) 40

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
R06-IOC-01	SHA-256	dc89160b8501065099a69bc6a8f34ce085774878387d071f4e249eebb57a757b	8000	Pre-Submit DOM Keystroke sendBeacon Streamer SHA-256	HUNT
R06-IOC-02	SHA-256	629fd5033240e77684003b632c3b247671ac2043a4e87ae7f420feb836551a45	8000	CORS-Wildcard Keystroke & Redirect Receiver SHA-256	HUNT
R06-IOC-03	SHA-256	7da269d7b5b55d409b71702dc53d4329bbeb0afb0540ab3d98896c3625836c54	8000	One-Click Login Clone & Asset Rewriter Script SHA-256	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_aitm_token_replay_and_ligolo_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Report #06 Detection Engineering Suite
// Target: Microsoft Sentinel (Entra ID SigninLogs + Microsoft Defender XDR DeviceNetworkEvents)
// Context: Separates default-port Ligolo-ng egress sightings (PORT_POLICY_11601_11610_ONLY) from
//          24h identity-correlated tunnel sessions (IDENTITY_CORRELATED_TUNNEL) using dcount(NetworkEventKey)
//          (DeviceName + Timestamp + ReportId) and minif/take_anyif(..., InWindow) to prevent leftouter join
fan-out.
// Note: Excludes AS8075 (Microsoft AVD/W365) and commercial proxy ASNs from ASN triggers.

let Lookback = 14d;
let LigoloDefaultPorts = range(11601, 11610, 1);
let HighConcentrationRelayASNs = dynamic(["16276", "51167", "37963", "202053", "25697", "401701"]);
let SuspiciousIdentitySessions =
    SigninLogs
    | where TimeGenerated >= ago(Lookback) and ResultType == "0" and TokenIssuerType == "AzureAD"
    | where RiskLevelDuringSignIn in ("medium", "high")
         or RiskEventTypes_V2 has_any ("anomalousToken", "unfamiliarFeatures", "maliciousIPAddress")
         or (tostring(AutonomousSystemNumber) in (HighConcentrationRelayASNs) and
tostring(DeviceDetail.isCompliant) != "true")
    | where isnotempty(UserPrincipalName)
    | extend AccountUpnKey = tolower(UserPrincipalName)
    | project SignInTime = TimeGenerated, UserPrincipalName, AccountUpnKey, SignInIP = IPAddress,
              SignInASN = tostring(AutonomousSystemNumber), SessionId, AppDisplayName;
DeviceNetworkEvents
| where Timestamp >= ago(Lookback) and ActionType == "ConnectionSuccess" and RemoteIPType == "Public"
| where RemotePort in (LigoloDefaultPorts)
     or (RemotePort in (443, 8443, 2083, 8080, 4445)
         and InitiatingProcessCommandLine has_any ("-ignore-cert", "ligolo", "wintun", "-accept-fingerprint"))
| extend AccountUpnKey = tolower(InitiatingProcessAccountUpn),
         NetworkEventKey = strcat(DeviceName, "|", tostring(Timestamp), "|", tostring(ReportId))
| join kind=leftouter (SuspiciousIdentitySessions) on AccountUpnKey
| extend InWindow = isnotempty(SessionId) and Timestamp between (SignInTime .. (SignInTime + 24h))
| where RemotePort in (LigoloDefaultPorts) or InWindow
| summarize FirstTunnel = min(Timestamp), LastTunnel = max(Timestamp),
            TunnelConnections = dcount(NetworkEventKey),
            MatchedSignIns = dcountif(SessionId, InWindow),
            SampleCmd = take_any(InitiatingProcessCommandLine),
            CorrelatedSignInTime = minif(SignInTime, InWindow),
            CorrelatedSignInIP = take_anyif(SignInIP, InWindow),
            CorrelatedSignInASN = take_anyif(SignInASN, InWindow),
            CorrelatedSessionId = take_anyif(SessionId, InWindow),
            CorrelatedApp = take_anyif(AppDisplayName, InWindow)
        by DeviceName, InitiatingProcessAccountUpn, InitiatingProcessAccountName,
           RemoteIP, RemotePort, InitiatingProcessFileName
| extend DetectionClass = iff(MatchedSignIns > 0, "IDENTITY_CORRELATED_TUNNEL", "PORT_POLICY_11601_11610_ONLY")
| order by MatchedSignIns desc, TunnelConnections desc, FirstTunnel desc
```

sigma_evilginx_novnc_and_ligolo_agent.yml

SIGMA

```
title: Ligolo-ng Linux Agent Execution or Legacy Relay TUN Provisioning
id: 96a1c410-1f23-4b88-9d01-202611160601
status: experimental
description: >
  Detects Linux execution of the unprivileged Ligolo-ng Go socket agent dialing an external
  relay (-connect, --connect, or -connect= with host:port or https:// / wss:// URL, anchored
  so curl --connect-timeout and openssl s_client never match) across default agent binary
  names or flag/URL-bearing renamed binaries, plus manual ip tuntap ligolo virtual TUN setup.
```

```
  author: Odin by Brandefense
  date: 2026-11-16
  logsource:
    category: process_creation
    product: linux
  detection:
    selection_agent_image:
      Image|endswith:
        - '/agent'
        - '/ligolo-agent'
        - '/ligolo-ng'
      CommandLine|re: '(?:^|\s)--?connect(?:\s+|=)(?:(?:https?|wss?)://[A-Za-z0-9_.\[\]:/-]+|[A-Za-z0-9_.\[\]-]+:
  \d{2,5})(?:\s|$)'
    selection_agent_cli_name:
      CommandLine|re: '(?:^|[\s/])(?:ligolo-)?agent\s+.*--?connect(?:\s+|=)(?:(?:https?|wss?)://[A-Za-z0-9_.\
  [\]:/-]+|[A-Za-z0-9_.\[\]-]+:\d{2,5})(?:\s|$)'
    selection_agent_connect:
      CommandLine|re: '(?:^|\s)--?connect(?:\s+|=)(?:(?:https?|wss?)://[A-Za-z0-9_.\[\]:/-]+|[A-Za-z0-9_.\[\]-]+:
  \d{2,5})(?:\s|$)'
      CommandLine|contains:
        - '-ignore-cert'
        - '-accept-fingerprint'
        - '-retry'
        - ':1160'
        - 'ligolo'
        - 'https://'
        - 'wss://'
        - 'http://'
        - 'ws://'
    selection_relay_tun_setup:
      Image|endswith: '/ip'
      CommandLine|contains|all:
        - 'tuntap'
        - 'mode tun'
        - 'ligolo'
    condition: selection_agent_image or selection_agent_cli_name or selection_agent_connect or
  selection_relay_tun_setup
  falsepositives:
    - Authorized red-team tunneling exercises on designated Linux jump hosts or relay VMs
  level: high
  tags:
    - attack.command_and_control
    - attack.t1572
  ---
  title: Ligolo-ng Windows Unprivileged Socket Agent Execution
  id: 96d4f743-4c56-4e11-8a34-202611160604
  status: experimental
  description: >
    Detects Windows execution of the unprivileged Ligolo-ng Go agent (agent.exe or renamed binary)
    connecting to an external operator relay via host:port or https:// / wss:// URL (which
    terminates TUN traffic in proxy-side gVisor netstack).
  author: Odin by Brandefense
  date: 2026-11-16
  logsource:
    category: process_creation
    product: windows
  detection:
    selection_cli_connect:
      CommandLine|re: '(?:^|\s)--?connect(?:\s+|=)(?:(?:https?|wss?)://[A-Za-z0-9_.\[\]:/-]+|[A-Za-z0-9_.\[\]-]+:
  \d{2,5})(?:\s|$)'
      CommandLine|contains:
        - '-ignore-cert'
        - '-accept-fingerprint'
        - '-retry'
        - ':1160'
        - 'ligolo'
        - 'https://'
        - 'wss://'
        - 'http://'
        - 'ws://'
    selection_agent_image:
      Image|endswith:
```

```
      - '\agent.exe'
      - '\ligolo-agent.exe'
      - '\ligolo-ng.exe'
    CommandLine|re: '(?:^|\s)--?connect(?:\s+|=)(?:(?:https?|wss?)://[A-Za-z0-9_.\[\]:/-]+|[A-Za-z0-9_.\[\]-]+:
\d{2,5})(?:\s|$)'
  selection_agent_cli_name:
    CommandLine|re: '(?:^|[\s\\])(?:ligolo-)?agent(?:\.exe)?\s+.*--?connect(?:\s+|=)(?:(?:https?|wss?)://[A-Za-
z0-9_.\[\]:/-]+|[A-Za-z0-9_.\[\]-]+:\d{2,5})(?:\s|$)'
  condition: selection_cli_connect or selection_agent_image or selection_agent_cli_name
falsepositives:
  - Authorized adversary-simulation engagements with documented Rules of Engagement
level: critical
tags:
  - attack.command_and_control
  - attack.t1572
---
title: Unsanctioned Wintun Kernel Tunnel DLL Load Outside Approved VPN Paths
id: 96e5a854-5d67-4f22-9b45-202611160605
status: experimental
description: >
  Detects loading of wintun.dll by processes outside five approved enterprise VPN directories,
  covering Windows-hosted Ligolo-ng relay listeners, double-pivot relays, and custom TUN tools.
author: Odin by Brandefense
date: 2026-11-16
logsource:
  category: image_load
  product: windows
detection:
  selection_wintun:
    ImageLoaded|endswith: '\wintun.dll'
  filter_approved_vpns:
    # Exact executables only; add the approved VPN clients used in your fleet.
    Image:
      - 'C:\Program Files\WireGuard\wireguard.exe'
      - 'C:\Program Files\Tailscale\tailscaled.exe'
      - 'C:\Program Files\Cloudflare\Cloudflare WARP\warp-svc.exe'
  condition: selection_wintun and not filter_approved_vpns
falsepositives:
  - Portable developer VPN clients or authorized red-team relay workstations
level: high
tags:
  - attack.command_and_control
  - attack.t1572
---
title: Self-Hosted noVNC and Websockify Browser Workbench Execution (Cloud / VPS Telemetry)
id: 96b2d521-2a34-4c99-8e12-202611160602
status: experimental
description: >
  Cloud-workload / VPS process detection for startup of noVNC HTML5 RFB proxies or websockify
  bridges attached to local VNC displays on compromised Linux VMs or rogue staging servers.
author: Odin by Brandefense
date: 2026-11-16
logsource:
  category: process_creation
  product: linux
detection:
  selection_novnc:
    CommandLine|contains:
      - 'novnc_proxy'
      - 'websockify'
      - '/utils/launch.sh'
  selection_vnc_target:
    CommandLine|contains:
      - '--vnc'
      - 'localhost:590'
      - '127.0.0.1:590'
      - ':3000'
  condition: selection_novnc and selection_vnc_target
falsepositives:
  - Legitimate cloud VDI, Proxmox/OpenStack consoles, or containerized GUI laboratory environments
level: medium
tags:
```

```
    - attack.credential_access
    - attack.defense_evasion
    - attack.t1539
---
title: Evilginx2 Reverse-Proxy or Static Login Cloner Execution (Cloud / VPS Telemetry)
id: 96c3e632-3b45-4d00-9f23-202611160603
status: experimental
description: >
  Cloud-workload / VPS process detection for Evilginx2 phishlet servers (constrained by binary
  image so mkdir -p .../phishlets never fires) or static login-cloner / keystroke-sink scripts.
author: Odin by Brandefense
date: 2026-11-16
logsource:
  category: process_creation
  product: linux
detection:
  selection_evilginx:
    Image|endswith:
      - '/evilginx'
      - '/evilginx2'
    CommandLine|contains|all:
      - '-p '
      - 'phishlets'
  selection_cloner_specific:
    CommandLine|contains:
      - 'switch_site.sh'
      - '/phish/receiver.py'
  condition: selection_evilginx or selection_cloner_specific
falsepositives:
  - Authorized adversary-simulation engagements on designated cloud redirector VMs
level: high
tags:
  - attack.credential_access
  - attack.t1557
  - attack.t1539
```

suricata_aitm_sendbeacon_and_ligolo.rules

SURICATA

```
# Odin by Brandefense - Report #06 Suricata 7.x AiTM, sendBeacon Keylogger, noVNC/RFB & Tunnel Suite
# Classification: TLP:CLEAR
# Sensor Placement & False-Positive Calibration Notes:
#   - sid:9602601, 9602606, 9602607: Enterprise perimeter / decrypted web-proxy egress ($HOME_NET -> !$HOME_NET).
#     Note: sid:9602606 and sid:9602607 are hunt/policy-tier telemetry rules (correlate with host process
egress).
#   - sid:9602602, 9602603, 9602604, 9602608: EASM crawlers or decrypted proxy taps.
#     Note: sid:9602604 and sid:9602608 also match legitimate external Proxmox/OpenStack/VDI consoles unless
correlated with AiTM proxies.
#   - sid:9602605: Requires SSL-intercepting forward proxy or cloud workload tap; excludes HTTP/1.1 & HTTP/2
browsers via nocase sec-fetch-site negation and tolerates compact or json.dumps whitespace.

# 1. Perimeter / TLS Proxy: Pre-submit DOM keylogger asynchronous POST /k flush (text/plain sendBeacon string
body, all k.js field labels)
alert http $HOME_NET any -> !$HOME_NET any (msg:"ODIN-AITM Pre-Submit sendBeacon Keystroke Stream POST /k";
flow:established,to_server; http.method; content:"POST"; http.content_type; content:"text/plain"; nocase;
fast_pattern; http.uri; content:"/k"; endswith; pcre:"/^(?:\/[A-Za-z0-9_-]+)?\/k$/"; http.request_body;
content:"["; startswith; content:"] "; distance:1; within:42; pcre:"/^\[[A-Za-z0-9_.:-]{1,40}\] (?:<DEL>|\S)/";
classtype:credential-theft; sid:9602601; rev:3;)

# 2. EASM / Proxy Sensor: Exposed Evilginx2 runtime dotfile or phishlet repository directory URI
alert http any any -> any any (msg:"ODIN-AITM Exposed Evilginx2 Runtime or Phishlet Directory URI";
flow:established,to_server; http.uri; content:"evilginx"; nocase; fast_pattern; pcre:"/\/(?:\.evilginx\/|
evilginx2\/(?:phishlets|core|database)\/)/i"; classtype:trojan-activity; sid:9602602; rev:4;)

# 3. Inbound Phishing Lure / EASM (Server-to-Client): Pre-submit sendBeacon keylogger script delivery (<DEL> +
beforeunload)
alert http !$HOME_NET any -> $HOME_NET any (msg:"ODIN-AITM Inbound sendBeacon Pre-Submit Keylogger Script
Delivery"; flow:established,to_client; file.data; content:"sendBeacon"; fast_pattern; content:"<DEL>"; within:
400; content:"beforeunload"; classtype:credential-theft; sid:9602603; rev:3;)
```

```
# 4. EASM / Hunting Sensor (Server-to-Client): Exposed external noVNC HTML5 client response (exclude sanctioned
VDI/hypervisor consoles)
alert http !$HOME_NET any -> $HOME_NET any (msg:"ODIN-HUNT Exposed External noVNC HTML5 Client Response
(Correlate Co-Hosted Proxy)"; flow:established,to_client; http.stat_code; content:"200"; file.data;
content:"noVNC"; fast_pattern; pcre:"/(?:\/?(?:vnc\.html|vnc_lite\.html|core\/rfb\.js)|HTML5 VNC client)/i";
classtype:misc-activity; sid:9602604; rev:3;)

# 5. Decrypted TLS Proxy / Cloud Workload Egress: Scripted M365 GetCredentialType enumeration burst (matches
compact & json.dumps whitespace; nocase sec-fetch-site excludes HTTP/1.1 & HTTP/2 browsers)
alert http any any -> any any (msg:"ODIN-RECON Scripted M365 GetCredentialType Account Enumeration Burst";
flow:established,to_server; http.method; content:"POST"; http.uri; content:"/common/GetCredentialType"; nocase;
fast_pattern; http.header_names; content:!"sec-fetch-site"; nocase; http.request_body;
content:"\"isFidoSupported\""; nocase; content:"\"originalRequest\""; nocase; pcre:"/\"isFidoSupported\"\s*:
\s*true\b.*\"originalRequest\"\s*:\s*\"\"/is"; threshold:type both, track by_src, count 4, seconds 120;
classtype:attempted-recon; sid:9602605; rev:4;)

# 6. Perimeter Egress Port-Policy: Uncommon port range 11601-11610 outbound TLS ClientHello (default Ligolo-ng
relay range; correlate host process)
alert tls $HOME_NET any -> !$HOME_NET 11601:11610 (msg:"ODIN-POLICY Uncommon Port 11601-11610 Outbound TLS
Handshake (Candidate Ligolo-ng Port)"; flow:established,to_server; ssl_state:client_hello; classtype:misc-
activity; sid:9602606; rev:3;)

# 7. Perimeter Egress Hunting: Non-browser external WebSocket upgrade (nocase Origin/Cookie header negation +
rate limit; correlate host process)
alert http $HOME_NET any -> !$HOME_NET any (msg:"ODIN-HUNT Non-Browser External WebSocket Upgrade (Correlate Host
Process)"; flow:established,to_server; http.header; content:"Upgrade: websocket"; nocase; fast_pattern;
content:"Sec-WebSocket-Key:"; nocase; http.header_names; content:!"origin"; nocase; content:!"cookie"; nocase;
threshold:type limit, track by_src, count 1, seconds 300; classtype:policy-violation; sid:9602607; rev:5;)

# 8. EASM / Hunting Sensor (Server-to-Client): Exposed external raw VNC RFB 003 framebuffer server greeting
alert tcp !$HOME_NET any -> $HOME_NET any (msg:"ODIN-HUNT Exposed External VNC RFB 003 Framebuffer Server
Greeting"; flow:established,to_client; content:"RFB 003."; startswith; fast_pattern; pcre:"/^RFB 003\.
00[378]\x0a/"; classtype:misc-activity; sid:9602608; rev:1;)
```

yara_aitm_keylogger_and_phish_toolkits.yar

YARA

```
/*
   Odin by Brandefense - Report #06 YARA AiTM Keylogger, Site Cloner & M365 Enumeration Rules
   Classification: TLP:CLEAR
*/

rule ODIN_AiTM_SendBeacon_PreSubmit_Keylogger_JS {
    meta:
        author = "Odin by Brandefense"
        description = "Detects pre-submit DOM keydown streamer flushing partial/abandoned inputs via
navigator.sendBeacon"
        sha256_specimen = "dc89160b8501065099a69bc6a8f34ce085774878387d071f4e249eebb57a757b"
        classification = "TLP:CLEAR"
    strings:
        $s1 = "navigator.sendBeacon(" ascii
        $s2 = "keepalive:true" ascii
        $s3 = "<DEL>" ascii
        $s4 = "beforeunload" ascii
        $s5 = "document.activeElement" ascii
        $s6 = "type==='password'" ascii
    condition:
        filesize < 64KB and 4 of ($s*)
}

rule ODIN_AiTM_CORS_Keystroke_Receiver_Python {
    meta:
        author = "Odin by Brandefense"
        description = "Detects Python BaseHTTPRequestHandler CORS-wildcard keystroke sink and redirector"
        sha256_specimen = "629fd5033240e77684003b632c3b247671ac2043a4e87ae7f420feb836551a45"
        classification = "TLP:CLEAR"
    strings:
        $c1 = "BaseHTTPRequestHandler" ascii
```

```
        $c2 = "Access-Control-Allow-Origin" ascii
        $c3 = "SITE = '/root/phish/site'" ascii
        $c4 = "redirect.txt" ascii
        $c5 = "timeout = 5" ascii
        $c6 = "k.js" ascii
    condition:
        filesize < 64KB and $c1 and $c2 and 2 of ($c3, $c4, $c5, $c6)
}

rule ODIN_AiTM_OneClick_Login_Site_Cloner_Shell {
    meta:
        author = "Odin by Brandefense"
        description = "Detects bash login-portal cloner and relative/protocol-relative asset rewriter
(switch_site.sh)"
        sha256_specimen = "7da269d7b5b55d409b71702dc53d4329bbeb0afb0540ab3d98896c3625836c54"
        classification = "TLP:CLEAR"
    strings:
        $b1 = "#!/bin/bash" ascii
        $b2 = "(src|href)=\"[^\"]+\"" ascii
        $b3 = "^(data:|javascript:|#|mailto:|tel:)" ascii
        $b4 = "woff2?|ttf|eot|json|map" ascii
        $b5 = "abs=\"$PROTO://$HOST$u\"" ascii
    condition:
        filesize < 64KB and 4 of ($b*)
}

rule ODIN_M365_GetCredentialType_And_Lateral_Auth_Toolkit {
    meta:
        author = "Odin by Brandefense"
        description = "Detects Python M365 GetCredentialType account/federation enumerators and multi-service
credential stuffers"
        sha256_specimen = "24146b8cbbafc57995e77c7a65e799fc471374b27c06a7ac931a7282c2c9cfaa"
        classification = "TLP:CLEAR"
    strings:
        $o1 = "login.microsoftonline.com/common/GetCredentialType" ascii
        $o2 = "isFidoSupported" ascii
        $o3 = "IfExistsResult" ascii
        $o4 = "PrefCredential" ascii
        $l1 = "/service/preauth" ascii
        $l2 = "/j_spring_security_check" ascii
        $l3 = "smtplib.SMTP_SSL" ascii
    condition:
        filesize < 64KB and ((3 of ($o*)) or (all of ($l*)))
}
```

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/keystrokes-before-submit/
Short Read · brandefense.io/odin-research/blog/keystrokes-before-submit/
Defense Kit · brandefense.io/odin-research/defense-kit/