ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 33 / 30 · C2 INFRASTRUCTURE

GPU Marketplace Snipers and Evasive Tunnels

AmneziaWG, X-UI and Relay C2

What one exposed compute workbench and 624 tunnel and C2 endpoints show about obfuscated relays, and which look-alikes to keep off block lists

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

45
Entries in one exposed workbench directory (108.88 MB, 37 unique SHA-256 digests)

4
Differently sized copies of the GPU marketplace bidding script, from 9,417 to 11,438 bytes

4
Tunnel tools staged together: AmneziaWG, WireGuard, X-UI and FRP

624
Tunnel, redirector and C2 endpoints on 263 hosts, in eight feed-labeled families

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

What one exposed compute workbench and 624 tunnel and C2 endpoints show about obfuscated relays, and which look-alikes to keep off block lists

01	Executive Summary	04
02	What Was Found: One Workbench and 624 Relay Endpoints	07
03	The GPU Marketplace Sniper and Its Workload Scripts	10
04	Four Tunnel Tools on One Host: AmneziaWG, WireGuard, X-UI and FRP	13
05	Relay and C2 Families: GOST, Daybreak, GhostSocks and RedGuard	16
06	Look-Alikes That Should Not Be Blocked, and Other Explanations	19
07	What Defenders Should Do	22
08	Detection Rules	24
09	Indicators	27
10	Scope and Limits	29
11	Sources	31
A	Indicators of compromise (2)	33
B	Detection rules (4 files)	34

IN SHORT

An exposed Netherlands workbench held four differently sized copies of a GPU marketplace bidding script. It was catalogued separately from the two-week relay observation window ending 18 November 2026. The same directory stages AmneziaWG, WireGuard, X-UI and FRP tunnel tooling. Separately, we reviewed 624 tunnel, redirector and C2 endpoints on 263 hosts. Only 65 are confirmed C2 or SOCKS5 relays. Hunt obfuscated tunnel tools outside approved bastions, and block only corroborated relays.

01

CHAPTER 01

Executive Summary

IN SHORT

An exposed Netherlands workbench held four differently sized copies of a GPU marketplace bidding script. It was catalogued separately from the two-week relay observation window ending 18 November 2026. The same directory stages AmneziaWG, WireGuard, X-UI and FRP tunnel tooling. Separately, we reviewed 624 tunnel, redirector and C2 endpoints on 263 hosts. Only 65 are confirmed C2 or SOCKS5 relays. Hunt obfuscated tunnel tools outside approved bastions, and block only corroborated relays.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 04

01 Executive Summary

Tools built to slip past deep packet inspection (DPI) or network address translation now serve three kinds of user. Privacy users, grey-market compute traders and intrusion operators all run the same AmneziaWG, X-UI, FRP and GOST stacks. This report looks at both ends.

One exposed workbench holds these tools together; its owner's activity is unconfirmed. Defenders can use cyber threat intelligence to review the wider set of 624 relay endpoints and decide which ones belong on a block list.

1. **A GPU bidding bot shares one directory with four tunnel tools.**
   An open web directory in the Netherlands holds 45 entries. Four similarly named Python files appear designed to bid on spot GPU rentals on a legitimate marketplace. Next to them sit AmneziaWG and WireGuard installers, client profiles, an X-UI panel archive and Fast Reverse Proxy (FRP) binaries.

2. **Default certificates and shared ports explain most relay noise.**
   Of 624 endpoints, 510 carry the HypeAgent feed label. Of those, 158 serve only the default Kubernetes ingress certificate and 297 sit on 19 multi-port reverse-proxy hosts, 278 of them on three hubs. Only nine present the default certificate of the GOST tunnel tool.

3. **Only 65 of 624 endpoints are confirmed C2 or SOCKS5 relays.**
   They sit on 49 commercial hosting addresses. Paired Daybreak listeners on ports 1500 and 1501 lead the group, with one GhostSocks relay. A feed marks 29 more endpoints as confirmed, but they sit on mail ports, carrier networks or remote-desktop servers.

Three Decisions for Leaders

1. **Govern egress to GPU marketplaces and obfuscated VPNs.** Unapproved Python polling of GPU rental APIs and UDP or TLS tunnels that hide WireGuard headers should only run from named bastions.
2. **Keep look-alikes off automated block lists.** Default Kubernetes certificates, mail and database ports, dynamic carrier networks and dual-use VNC servers cause outages when blocked on a feed label alone.
3. **Hunt fingerprints, not port numbers.** Pair tunnel binaries on hosts with the certificate subjects and JARM values in this report.

The Study in Numbers

Number	What it is
45	Entries in one exposed workbench directory (108.88 MB, 37 unique SHA-256 digests)
4	Differently sized copies of the GPU marketplace bidding script, from 9,417 to 11,438 bytes
4	Tunnel tools staged together: AmneziaWG, WireGuard, X-UI and FRP
624	Tunnel, redirector and C2 endpoints on 263 hosts, in eight feed-labeled families
158	Endpoints that serve only the default Kubernetes ingress certificate
65	Confirmed C2 and SOCKS5 endpoints on 49 addresses (10.4% of 624)
29	Endpoints a feed marks confirmed that sit on mail ports, carrier networks or VNC servers

How to use this report.

Security leaders can read Chapters 1, 6 and 7. Cloud, network and ML platform teams should read Chapters 3 and 4. SOC and detection engineers should start with Chapters 8 and 9. CTI analysts will find the relay families in Chapters 5 and 6.

02

CHAPTER 02

What Was Found: One Workbench and 624 Relay Endpoints

IN SHORT

Two separate sets of evidence sit side by side in this report. No address, certificate or file hash links the workbench to the 624 relay endpoints, so the combined picture is an analytical model.

INSIDE THIS CHAPTER

The workbench in five layers 07

The relay set in eight families 08

02 What Was Found: One Workbench and 624 Relay Endpoints

Censorship-evasion protocols and reverse proxies blur the line between privacy tools, compute automation and C2 relays. AmneziaWG and X-UI defeat state DPI. FRP and GOST cross NAT boundaries. All four turn up on personal VPS nodes, GPU rental bots and multi-tier C2 redirectors alike (T1572, T1090.002).

We looked at two sets of evidence. The first is one exposed workbench directory. The second is 624 endpoints on 263 hosts that public feeds label as tunnel, redirector or C2 infrastructure.

No shared address, certificate or file digest connects the two. Inside the relay set, only one Hong Kong host spans two feed families.

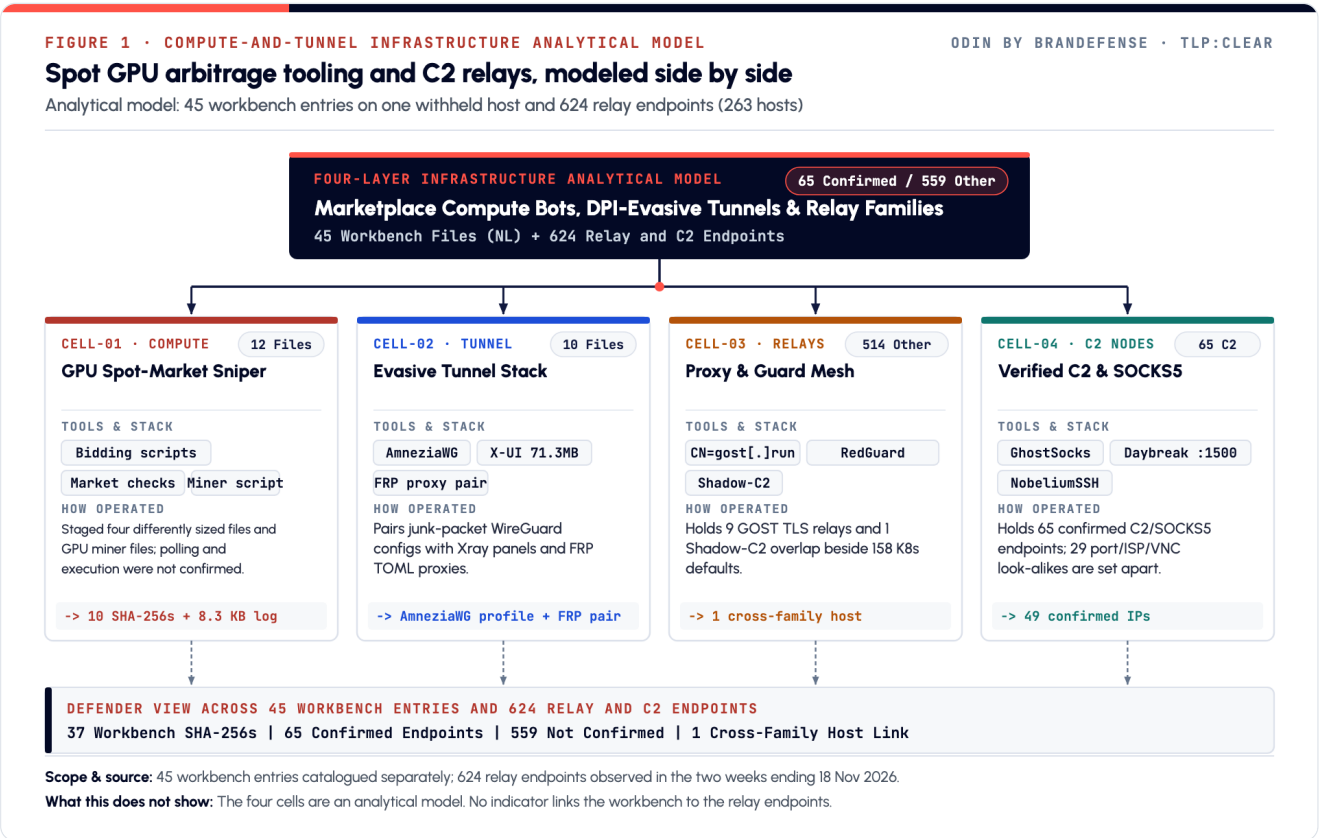


Figure 1: Spot GPU arbitrage tooling and C2 relays, modeled side by side

The workbench in five layers

Defenders can use external attack surface management to find open directories like the one on a Netherlands host that exposed 45 entries over plain HTTP. Forty entries list a size, adding up to 108,883,402 bytes (108.88 MB). Five lock, marker and helper entries list no size.

Thirty-nine entries carry SHA-256 digests, 37 of them unique. Files on the server carry timestamps from as early as 2 August 2026. The host address is withheld.

Layer	Entries	Hashed / unique	Bytes	Main contents
GPU marketplace sniper	8	8 / 7	64,889	Bidding files, a market-check helper and an output log (names and paths withheld)
GPU workload provisioning	4	3 / 3	99,817	Runtime install script, miner launcher, node-layout file and screen helper (names and paths withheld)
DPI-evasive tunnel stack	10	9 / 9	108,653,689	X-UI panel archive, AmneziaWG and WireGuard installers, FRP binaries and TOML configs (names and paths withheld)
Shell and SSH state	9	9 / 9	54,147	Shell history, SSH key material, SSH host-key list (names and paths withheld)
Container and shell metadata	14	10 / 9	10,860	Docker Buildx files, shell profiles, caches, lock files (names and paths withheld)

The relay set in eight families

The 624 endpoints come from eight families. The family names are the feeds' own labels. They describe matching network patterns, not attributed threat groups.

Feed family label	Endpoints	Hosts	Confirmed	Share of 624
HypeAgent	510	171	0	81.7%
Daybreak	103	83	62	16.5%
UltraVNC	3	3	0	0.5%
NobeliumSSH	2	2	1	0.3%
RedGuard	2	2	0	0.3%
Shadow-C2	2	1	0	0.3%
GhostSocks	1	1	1	0.2%
TsundereBotnet	1	1	1	0.2%

CHAPTER 03

The GPU Marketplace Sniper and Its Workload Scripts

IN SHORT

Four similarly named Python files and two helper scripts sit in the exposed directory alongside a log file. Shell scripts beside them are named to prepare GPU machines for compute or mining jobs.

INSIDE THIS CHAPTER

Four differently sized sniper files 10

Workload scripts and SSH footprint 11

03 The GPU Marketplace Sniper and Its Workload Scripts

Decentralized compute marketplaces let hardware owners rent out idle GPUs for machine learning, rendering and other jobs. Clore[.]ai is one of them, and it is a legitimate service. Spot prices move as machines come online. So "sniper" scripts poll the order book fast and grab cheap GPUs before other bidders do.

Compute arbitrage can serve a business or a mining operation. Inside an enterprise, a bidding bot on stolen credentials or a rogue cloud workload brings shadow billing and policy risk (T1583.003).

We worked only from file names, sizes, extensions and hashes. We did not run any script and did not touch any marketplace account. Nothing here shows whether a GPU lease was won.

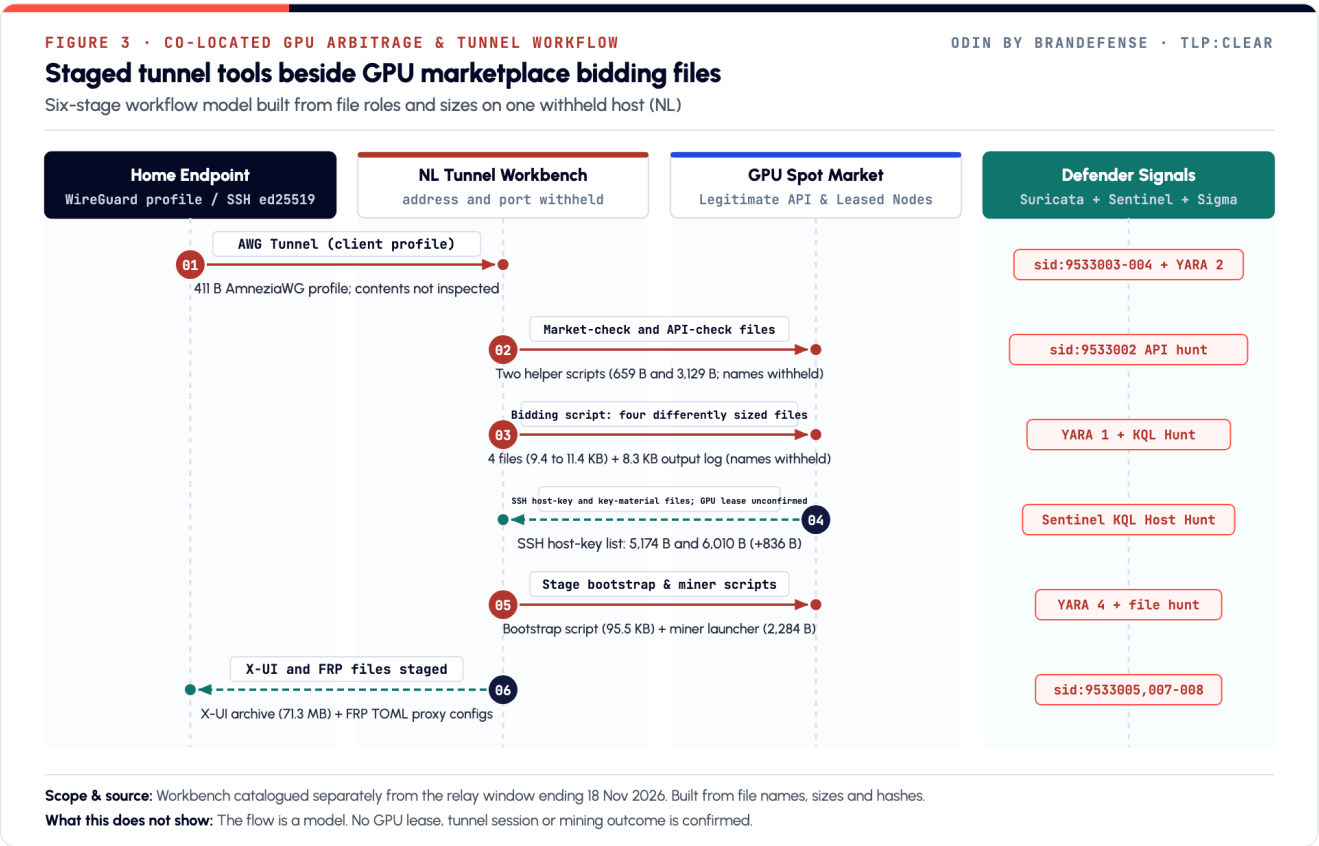


Figure 3: Staged tunnel tools beside GPU marketplace bidding files

Four differently sized sniper files

The directory contains an editor search history and an editor choice file. Four similarly named Python files differ in size; their editing order and changes were not established.

File label	File	Size (bytes)	Notes
File A (order unknown)	Name withheld	9,417	Contents and changes uninspected
File B (order unknown)	Name withheld	10,626	Contents and changes uninspected
Files C and D (order unknown)	Names withheld (two identical copies)	10,659 each	Contents and changes uninspected
File E (order unknown)	Name withheld	11,438	Contents and changes uninspected

Two smaller Python helper scripts (659 bytes and 3,129 bytes, names withheld) are named for market and API checks. An output log (8,302 bytes, name withheld) sits beside them; their contents and use were not established.

Workload scripts and SSH footprint

A cheap GPU lease only pays off if the job starts before a higher bidder takes the machine. Four provisioning files sit next to the sniper.

File	Size (bytes)	Role
Bootstrap script (name withheld)	95,521	Runtime and driver bootstrap script
Miner launcher (name withheld)	2,284	GPU compute or mining launcher
Layout file (name withheld)	2,012	Node and service layout JSON file
Screen helper (name withheld)	no listed size	Screen helper for the miner

The SSH host-key list file (path withheld) grew from 5,174 bytes in its older copy to 6,010 bytes, an increase of 836 bytes. The entries were not inspected, so the added bytes do not establish which hosts were contacted or whether a GPU lease occurred. The directory also exposes an Ed25519 private key and an authorized-keys file (paths withheld). We publish no key material and no hashes of these files.

04

CHAPTER 04

Four Tunnel Tools on One Host: AmneziaWG, WireGuard, X-UI and FRP

IN SHORT

Ten tunnel files make up 99.8% of the workbench bytes. Together they cover UDP header obfuscation, TLS proxy panels and NAT reverse proxies, so no single-protocol control catches them all.

INSIDE THIS CHAPTER

WireGuard and AmneziaWG staged together 13

X-UI, FRP and the shell trail 14

04 Four Tunnel Tools on One Host: AmneziaWG, WireGuard, X-UI and FRP

Plain VPN handshakes and cleartext SSH flows are throttled or blocked by many ISPs, enterprise perimeters and national DPI firewalls. The workbench owner staged four tunnel technologies in one root directory, next to a 42,084-byte shell history.

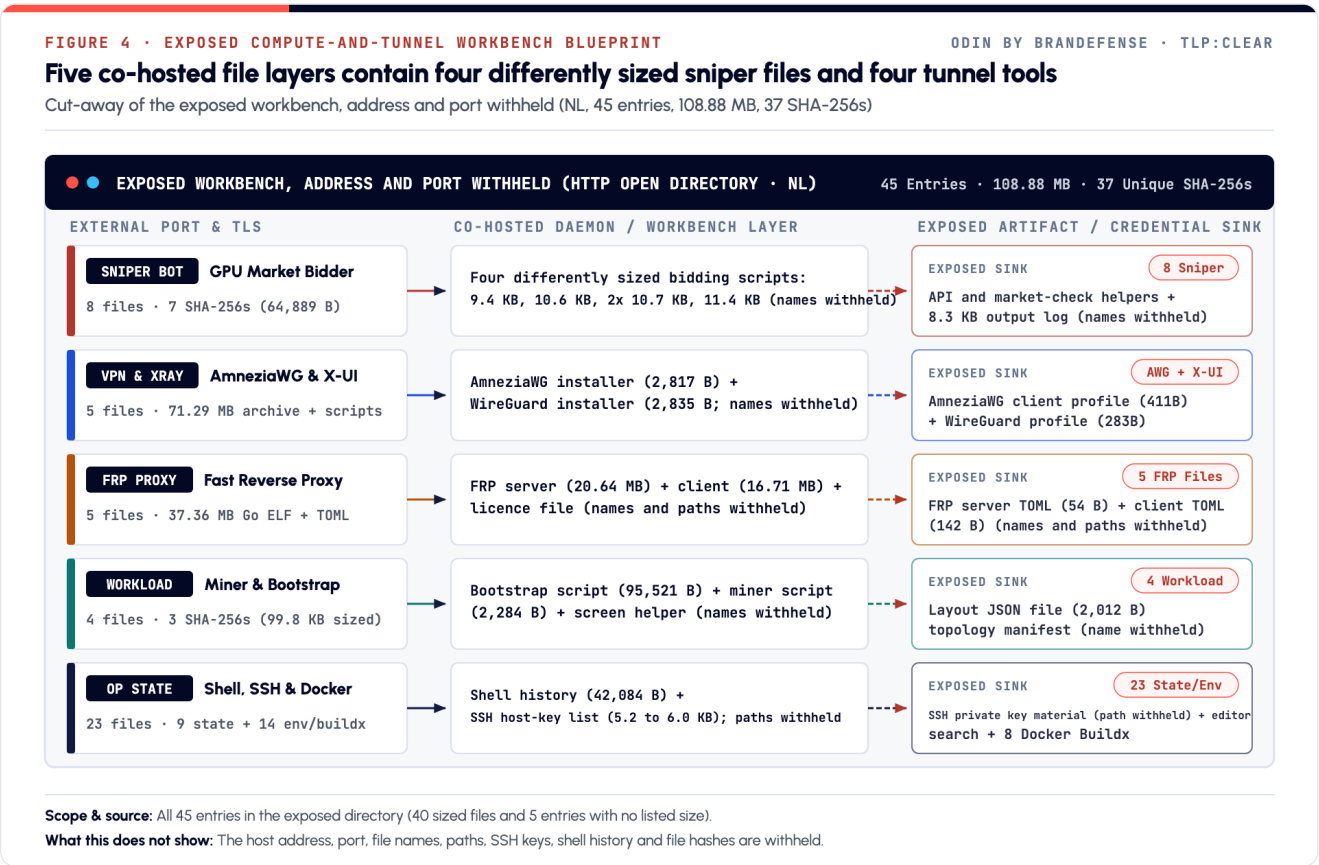


Figure 4: Five co-hosted file layers contain four differently sized sniper files and four tunnel tools

WireGuard and AmneziaWG staged together

Standard WireGuard uses fixed message type values (1 to 4) and fixed-length handshake messages. Per the AmneziaWG documentation (Source 2), these are 148 bytes to start, 92 bytes to reply and 64 bytes for a cookie. Source 1 gives the field layouts. DPI engines fingerprint and drop those predictable packets easily.

The workbench holds both a WireGuard installer script (2,835 bytes, name withheld) and an AmneziaWG installer script (2,817 bytes, name withheld). That pairing shows both installers were staged; their order of use is unknown. AmneziaWG keeps WireGuard's cryptography and adds settings that change what the packets look like on the wire.

AmneziaWG setting	What it does
`Jc`, `Jmin`, `Jmax`	Sends a number of random junk packets of random length before the handshake
`S1`, `S2`	Adds random padding in front of handshake messages
`H1` to `H4`	Replaces the fixed message type values 1 to 4

Two client profiles fit this picture. A plain WireGuard profile is 283 bytes. An AmneziaWG profile in a client-profile subfolder (path withheld) is 411 bytes.

The 128-byte difference is consistent with additional settings, but the catalogue does not establish the profiles' contents. Both profile names point to the owner's own home machine, so we do not publish them.

X-UI, FRP and the shell trail

When UDP is blocked outright, users fall back to TLS tunnels and reverse port forwarding. Six more tunnel files cover that path.

File	Size (bytes)	What it is
X-UI panel archive (name withheld)	71,288,093	X-UI web panel archive for multi-user Xray-core tunnels (VLESS, VMess, Trojan, Shadowsocks)
FRP server binary (name and path withheld)	20,639,928	FRP server, Go ELF binary
FRP client binary (name and path withheld)	16,707,768	FRP client, Go ELF binary
Licence file (name and path withheld)	11,358	Apache-2.0 licence shipped with FRP
FRP client config (name and path withheld)	142	142-byte client TOML config
FRP server config (name and path withheld)	54	54-byte minimal server TOML config

FRP introduced TOML configuration in version 0.52.0; the directory metadata does not establish the staged binaries' version. A reverse proxy of this kind exposes a GPU or management port that sits behind NAT.

Eight Docker Buildx and token-seed files and the shell history show container builds and interactive administration on the host. All of it was readable over plain HTTP.

05

CHAPTER 05

Relay and C2 Families: GOST, Daybreak, GhostSocks and RedGuard

IN SHORT

Intrusion operators use the same kinds of Go tunnels, filtering redirectors and paired listeners to shield their controllers. Certificate subjects and JARM pairs separate real relays from cloud noise better than port numbers.

INSIDE THIS CHAPTER

HypeAgent: three populations, nine GOST relays 16

Daybreak: paired listeners on ports 1500 and 1501 17

GhostSocks, NobeliumSSH, TsundereBotnet and RedGuard 17

05 Relay and C2 Families: GOST, Daybreak, GhostSocks and RedGuard

Operators hide their back-end controllers behind Go tunnels (GOST and GhostSocks), traffic-filtering HTTP redirectors (RedGuard) and dual-port C2 listeners (Daybreak). Figure 2 maps each capability in this report to the detection rules in Chapter 8.

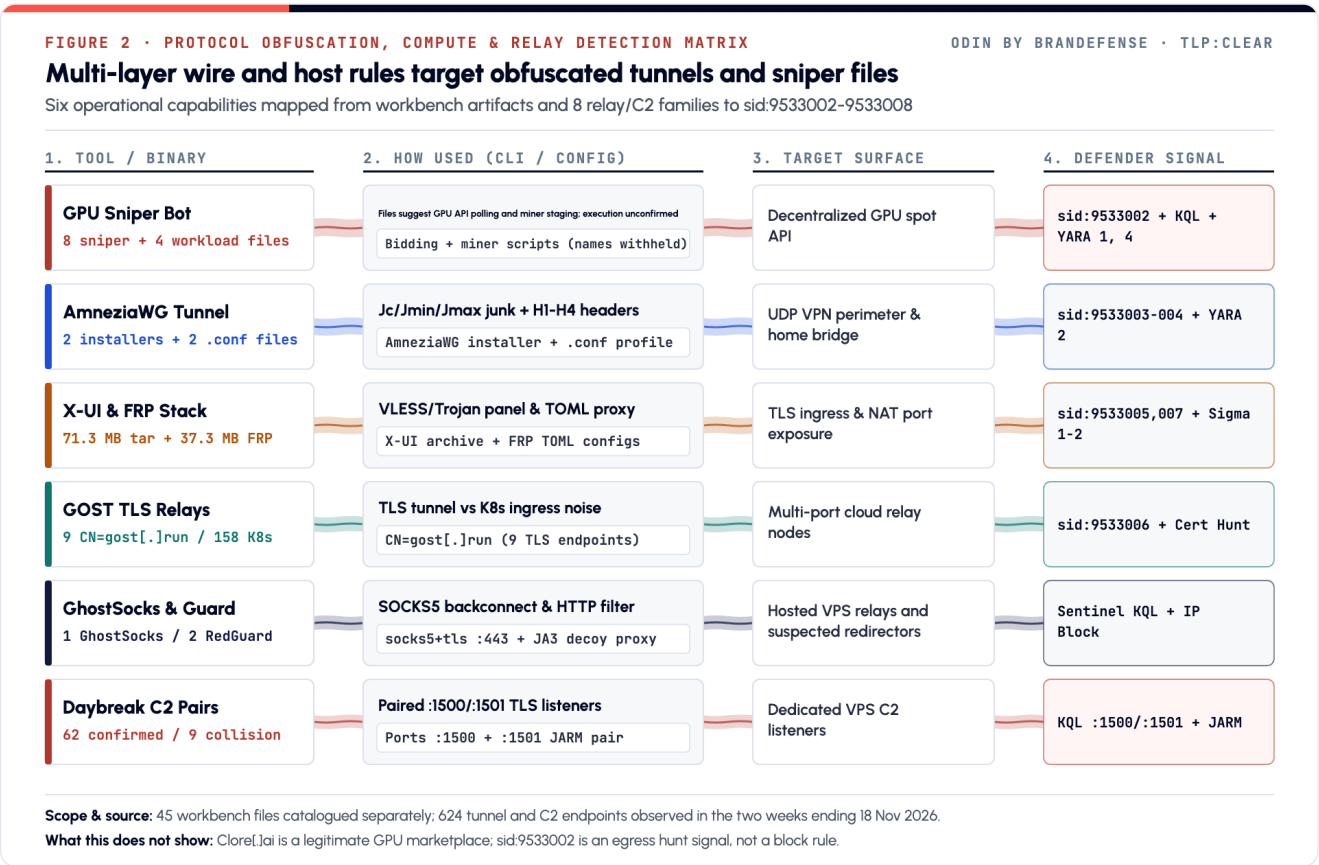


Figure 2: Multi-layer wire and host rules target obfuscated tunnels and sniper files

HypeAgent: three populations, nine GOST relays

HypeAgent makes up 81.7% of the relay set: 510 endpoints on 171 hosts. The feed marks every one of them as suspected, and none is confirmed. Three distinct groups explain 464 of the 510 endpoints (91.0%).

HypeAgent group	Endpoints	Hosts	What it is
Multi-port reverse-proxy hubs	297	19	Per-port certificates on cloud proxy hubs. Three Tencent Cloud hubs with 161, 91 and 26 ports produce 278 rows (54.5% of HypeAgent).
Default Kubernetes ingress	158	109	`CN=Kubernetes_Ingress_Controller_Cert` on major cloud providers. 71 share the default `ingress-nginx` JARM.
GOST TLS relays	9	6	`CN=gost[.]run`, the default self-signed certificate of the GOST tunnel tool (Source 10)

The 109 Kubernetes hosts are 63.7% of the 171 HypeAgent hosts. Their 158 endpoints run ordinary ingress controllers that happen to match a broad Go and TLS pattern.

The nine GOST endpoints sit on six suspected hosts across nine distinct service ports (ports withheld). One withheld Hong Kong host spans the GOST and Shadow-C2 feed families. Its address and service ports are withheld.

Daybreak: paired listeners on ports 1500 and 1501

Daybreak accounts for 62 of the 65 confirmed endpoints, on 46 commercial VPS addresses. Of these, 43 (69.4%) listen on port 1500 (27 rows) or port 1501 (16 rows). Each port has its own stable JARM value, listed in Chapter 9.

The other 19 confirmed Daybreak endpoints sit on 17 hosts. They use HTTPS and proxy ports such as 443, 80 and 10003. Their certificates carry names such as `rsproxy`, `allproxy`, `ldlr-redirect` and `litescrape-executor`.

GhostSocks, NobeliumSSH, TsundereBotnet and RedGuard

Family	Confirmed	What we saw
GhostSocks	1	A Go reverse SOCKS5 relay over TLS on a Hetzner server, port 443
NobeliumSSH	1 of 2	One OVH server on port 3389 with certificate `CN=server33916`; one suspected cloud host (address and port withheld)
TsundereBotnet	1	One listener on port 3389
RedGuard	0 of 2	Two suspected redirectors in China and the United States (addresses and ports withheld)

GhostSocks gives an operator a reverse SOCKS5 tunnel (RFC 1928) wrapped in TLS. RedGuard is a C2 front-end guard.

It checks the JA3 fingerprint and HTTP Host header of each visitor, and sends unwanted visitors to a harmless decoy site (Source 11). The NobeliumSSH name is a feed label only. Nothing here ties these hosts to any named state group.

CHAPTER 06

Look-Alikes That Should Not Be Blocked, and Other Explanations

IN SHORT

Blocking every endpoint a feed calls confirmed would hit mail servers, carrier customers and IT admin tools. Keep 559 of the 624 endpoints on review lists, and block only the 65 confirmed relays.

INSIDE THIS CHAPTER

Why 29 feed-confirmed endpoints are not on the block list 19

Four ways to read the evidence 20

06 Look-Alikes That Should Not Be Blocked, and Other Explanations

Raw feeds mix up shared reverse-proxy certificates, mail and database services on reused VPS addresses, and normal admin tools with real C2 servers. Of the 624 endpoints, 65 (10.4%) are confirmed. The other 559 (89.6%) stay unpublished.

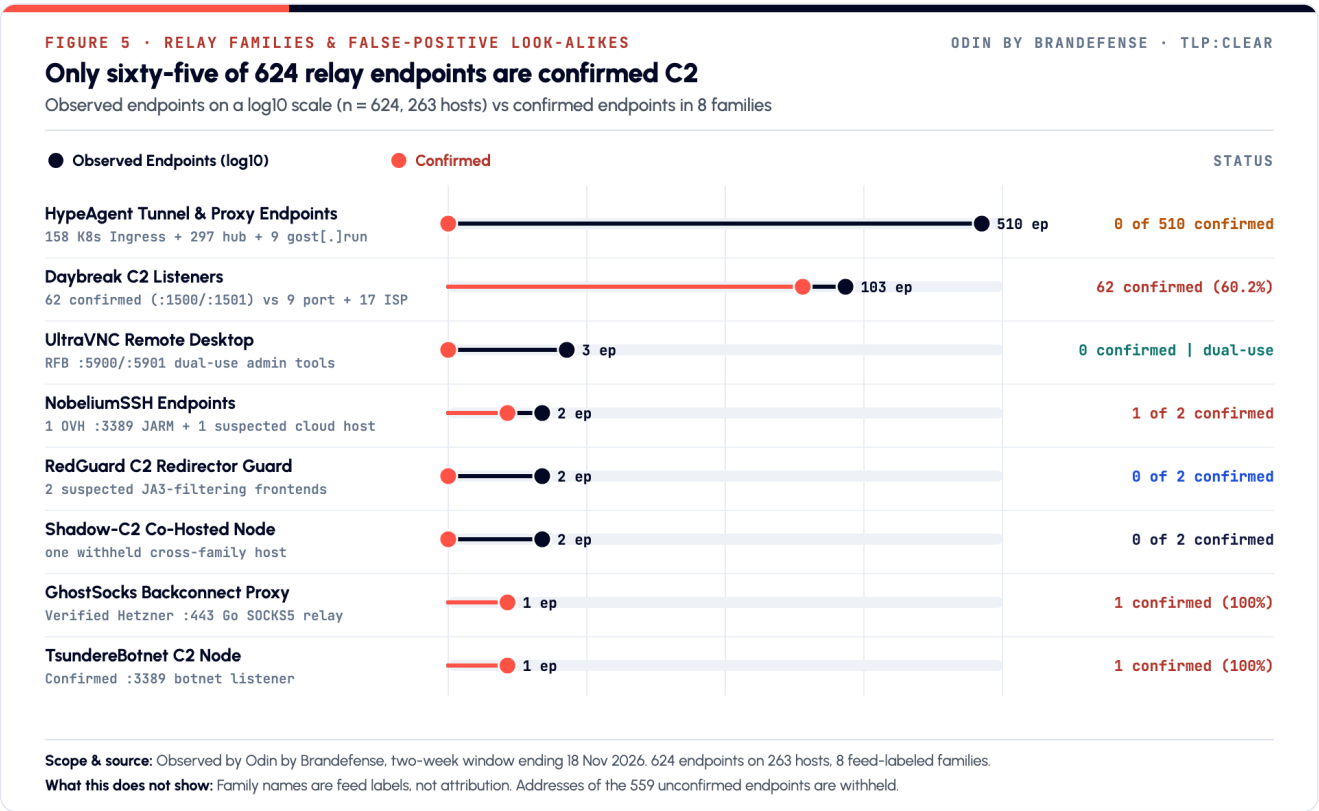


Figure 5: Only sixty-five of 624 relay endpoints are confirmed C2

Why 29 feed-confirmed endpoints are not on the block list

The feeds mark 94 endpoints as confirmed. Of these, 29 (30.9%) belong to one of three look-alike groups. Blocking them would cause outages without stopping an operator.

Look-alike group	Endpoints	Why it is not blocked
Daybreak label on mail, database and FTP ports	9	Ports 587 (2), 465 (2), 5432 (2), 21, 143 and 993 host normal services
Daybreak label on dynamic carrier networks	17	Addresses on subscriber networks are reassigned to new customers
UltraVNC servers on ports 5900 and 5901	3	UltraVNC is a common remote-desktop tool for IT staff

Across the whole set, 27 endpoints sit on dynamic carrier networks: 17 Daybreak and 10 HypeAgent. Six more Daybreak endpoints are candidates only. Five of them carry breach-and-attack-simulation listener certificates, and one is a home NAS.

Group	Endpoints	Addresses	Status
Confirmed C2 and SOCKS5 relays	65	49	Block
Suspected	514	183	Review, address withheld
Dynamic carrier networks	27	23	Review, address withheld
Mail, database and FTP ports	9	8	Review, address withheld
Candidates	6	6	Review, address withheld
Dual-use VNC	3	3	Review, address withheld

Four ways to read the evidence

1. **A compute trader, not an intruder.** The sniper, miner script and tunnel stack may belong to an independent GPU trader working around regional DPI. Nothing links this host to any intrusion. We treat its owner as a user, not an operator, and withhold its address, port, file names and file hashes.
2. **Normal services on shared ports.** Nine Daybreak-labeled endpoints run on SMTP, IMAP, PostgreSQL and FTP ports. Blocking them risks mail and database outages.
3. **Kubernetes and simulation tools.** Default ingress certificates and breach-and-attack-simulation listeners match broad Go and TLS patterns. They are not C2 servers.
4. **Carrier churn and dual-use VNC.** Carrier addresses move between customers, and UltraVNC supports normal IT work. Both stay on time-limited review lists instead of block lists.

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

Network and SOC

- [ ] Alert on AmneziaWG (`amneziawg-go`, `awg-quick`), X-UI, FRP (`frpc`, `frps`), GOST and GhostSocks processes outside named bastions with Sigma rules 1 and 2. Use Suricata SIDs 9533003 to 9533007 for network retrievals and the GOST certificate.
- [ ] Hunt `CN=gost[.]run` certificates and the paired Daybreak JARM values on ports 1500 and 1501.

Cloud and ML platform

- [ ] Audit outbound traffic from corporate and cloud networks to GPU rental APIs such as `api.clore[.]ai` (SID 9533002). Require job approval on GPU clusters.
- [ ] Hunt GPU bidding and miner launch scripts on compute nodes with the YARA rules and Linux process logs.

Threat intelligence

- [ ] Before any feed reaches an automated block list, strip default Kubernetes ingress certificates and mail, database and FTP ports (21, 143, 465, 587, 993, 3306, 5432). Strip dynamic carrier networks and UltraVNC on ports 5900 and 5901 as well.
- [ ] Require an independent second signal before blocking any feed-labelled relay address at the perimeter for 30 days. Recheck the hosting lease before extending a block.

Attack surface and endpoints

- [ ] Find open HTTP directory listings on your own address space. SID 9533008 fires when a server on your network returns a Python `http.server` ("Directory listing for /") or Nginx `autoindex on` ("Index of /") listing that names these tunnel tools. Look for exposed SSH keys, shell history and `.toml` or `.conf` tunnel files.
- [ ] Scan Linux developer workstations, jump hosts and container build caches with the YARA rules for AmneziaWG profiles and FRP TOML files.

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 24
How the other rules work 24
MITRE ATT&CK mapping 25

08 Detection Rules

The full rules are in Appendix B. Suricata, YARA and Sigma engines were not available to us, so the rules were not loaded on live engines. The Suricata file passes a syntax lint with 0 problems. Test every rule in your own environment before production use.

Rule file	What it catches
`odin_gpu_sniper_amneziawg_xui_and_ghostsocks.rules`	Suricata 7: seven signatures for Python polling of the GPU marketplace API, AmneziaWG, X-UI and FRP downloads, the GOST certificate and open tunnel-tool directory listings
`sentinel_gpu_sniper_amneziawg_xui_and_socks_hunt.kql`	Microsoft Sentinel and Defender XDR: tunnel, marketplace-API and miner processes joined with connections to corroborated relays, ports 1500 and 1501, or the marketplace API
`sigma_amneziawg_xui_frp_and_gpu_miner_staging.yml`	Sigma (2 rules): AmneziaWG and X-UI processes, and FRP, GOST and GhostSocks processes on Linux
`odin_gpu_sniper_amneziawg_xui_and_frp_stack.yar`	YARA (4 rules): GPU marketplace bidding scripts, AmneziaWG profiles, FRP TOML configs and GPU miner launch scripts

How the Suricata signatures work

SID	Looks for	Why
9533002	Host `api.clore[.]ai` with a `python-requests` user agent	Hunt only: flags scripted polling of a legitimate marketplace (needs TLS inspection)
9533003	Outbound `GET` for an AmneziaWG `.sh`, `.tar.gz`, `.deb` or `.rpm` file	Catches download of an AmneziaWG installer or package
9533004	Outbound `GET` for an AmneziaWG `.conf` client profile	Catches download of an AmneziaWG client profile
9533005	Outbound `GET` for an `x-ui` archive whose URI ends `.tar.gz`	Catches download of the X-UI panel
9533006	Server certificate subject `CN=gost[.]run`	Flags a GOST tunnel using its default certificate
9533007	Outbound `GET` for `frpc` or `frps`, with or without `.toml`	Catches download of FRP binaries or configs
9533008	A 200 Python `http.server` or Nginx autoindex listing served from your network that names AmneziaWG, X-UI, FRP or GhostSocks	Flags an open directory exposing tunnel tooling on your own hosts

Keep SID 9533002 in alert-only mode, because the marketplace is legitimate. Add `pass` rules for approved ML sandboxes and for any authorized GOST test relays (SID 9533006).

How the other rules work

The KQL query looks back 7 days. Its process side finds `amneziawg-go`, `awg-quick`, `x-ui`, `frpc`, `frps`, `gost` or `ghostsocks`, or command lines naming the marketplace API, mining pool URLs or tunnel config files. Its network side finds connections to a defender-supplied list of corroborated relay addresses, to the marketplace API or `gost[.]run`, or non-browser traffic to ports 1500 and 1501. All of these network matches are kept after the join, including those with no matching process.

A full outer join on the device ties the two sides together. Each match gets one of four labels. Defenders can isolate corroborated relay contact, hunt command lines combining marketplace or miner and tunnel markers, or review port-pair and marketplace traffic.

Two named lists hold exact hostnames of approved GPU schedulers and network bastions. There are no wildcards. Exclude internal ERP traffic on ports 1500 and 1501 when tuning.

The two Sigma rules read Linux process creation logs. Rule 1 flags AmneziaWG and X-UI binaries and install paths. Rule 2 flags FRP, GOST and GhostSocks binaries, FRP TOML config names and `socks5+tls://` or `socks5+wss://` arguments.

Each rule excludes two exact hostnames in a `Computer` field; map it to your backend's hostname field. For GPU bidding or miner launch scripts on Linux hosts, defenders should also hunt command lines that call marketplace APIs or `stratum+` pools outside approved schedulers (T1583.003).

YARA rule	Fires when
`ODIN_Clore_GPU_Marketplace_Sniper_Script`	The marketplace API host appears with two order, spot, marketplace or auth markers, in files under 128 KB
`ODIN_AmneziaWG_Obfuscated_Tunnel_Config_or_Installer`	An `[Interface]` block or AmneziaWG name appears with `Jc`, `Jmin`, `Jmax` and `H1` or `S1`, in files under 64 KB
`ODIN_FRP_TOML_Reverse_Proxy_Config_Artifact`	`serverAddr` and `serverPort` appear with one of `[[proxies]]`, `remotePort` or `auth.token`, or `bindPort` appears with `auth.token`, in files under 64 KB. A server config with `bindPort` only does not match
`ODIN_Compute_Tunnel_Workbench_Manifest_or_Miner_Script`	A `stratum+` pool URL appears with `nvidia-smi` or `--algo`, in files under 256 KB

MITRE ATT&CK mapping

Tactic	Technique	How it shows up	Detection
Command and Control	T1572 Protocol Tunneling	AmneziaWG, X-UI, FRP and GOST tunnels	SIDs 9533003 to 9533007; Sigma rules 1 and 2; KQL
Command and Control	T1090.002 Proxy: External Proxy	GhostSocks SOCKS5 relays, RedGuard redirectors and Daybreak listeners	SID 9533006; KQL
Resource Development	T1583.003 Acquire Infrastructure: Virtual Private Server	Rented VPS relays and leased GPU nodes	SID 9533002; YARA rules 1 and 4; KQL

09

CHAPTER 09

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Confirmed relay addresses 27

Fingerprints and certificate subjects 27

09 Indicators

Appendix A holds 2 published indicators: 2 JARM fingerprints. No network addresses, workbench file names or file hashes are published. The certificate names below stay in this report only.

Type	Count	Where to use it
Network addresses (withheld)	0	Withheld pending an independent second signal; do not block
JARM fingerprints	2	TLS hunting on ports 1500 and 1501
Certificate subjects	6	TLS inspection and certificate search (text only)
Workbench file names and hashes (withheld)	0	Withheld because the workbench owner is not confirmed as an adversary

Confirmed relay addresses

Eight feed-labelled relay addresses are withheld pending an independent second signal for each exact address. Do not block them.

Fingerprints and certificate subjects

Indicator	Type	Use
`21d14d00021d21d21c42d43d00041d0eea9d6a7e6a07bccd28e28ed272ad91`	JARM, Daybreak port 1500	Hunt with the port and certificate, do not block alone
`3fd3fd0003fd3fd00042d42d00000000f78d2dc0ce6e5bbc5b8149a4872356`	JARM, Daybreak port 1501	Hunt with the port and certificate, do not block alone
`CN=gost[.]run`	Certificate subject, GOST default	Hunt; seen on 9 endpoints on 6 hosts
`rsproxy`, `allproxy`, `ldlr-redirect`, `litescrape-executor`	Certificate names, Daybreak HTTPS servers	Hunt in TLS logs
`server33916`	Certificate name, NobeliumSSH server	Hunt in TLS logs
`29d29d00029d29d00042d42d00000023f2ae7180b8a0816654f2296c007d93`	JARM, default `ingress-nginx`	Benign default; suppress, never block

Withheld: the workbench address, port, file names, paths and all 37 unique file hashes from the workbench. Also withheld: the eight feed-labelled relay addresses, the other relay addresses, the Hong Kong cross-family host and its ports, and the reverse-proxy hubs. The two client profile names are withheld because they describe the owner's home machine.

10

CHAPTER 10

Scope and Limits

IN SHORT

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

10 Scope and Limits

- **Window.** The relay findings cover a two-week observation window ending 18 November 2026. The workbench was catalogued separately; older dates are timestamps on its files.
- **Files, not outcomes.** We worked from file names, sizes and hashes. We did not run any script, use any key or contact the marketplace. Nothing confirms a GPU lease, a tunnel session or a mining job.
- **Owner, not operator.** The workbench may belong to an independent compute trader. Nothing links it to an intrusion.
- **One model, two data sets.** No address, certificate or hash links the workbench to the relay endpoints. The combined view is an analytical model.
- **Feed labels are not attribution.** HypeAgent, Daybreak, NobeliumSSH, RedGuard and the other family names come from feeds. They do not name a threat group.
- **Rule testing.** Suricata, YARA and Sigma engines were not available, so the rules were not loaded on live engines. The Suricata file passes a syntax lint. Test and tune all rules in your own environment.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. WireGuard. Protocol and Cryptography. https://www.wireguard.com/protocol/ (verified 20 November 2026)
2. Amnezia. AmneziaWG documentation. https://docs.amnezia.org/documentation/amnezia-wg/ (verified 20 November 2026)
3. IETF. RFC 1928: SOCKS Protocol Version 5. https://datatracker.ietf.org/doc/html/rfc1928 (verified 20 November 2026)
4. Kubernetes ingress-nginx. TLS/HTTPS user guide (default self-signed certificate). https://kubernetes.github.io/ingress-nginx/user-guide/tls/ (verified 20 November 2026)
5. MITRE ATT&CK. T1583.003 Acquire Infrastructure: Virtual Private Server. https://attack.mitre.org/techniques/T1583/003/ (verified 20 November 2026)
6. MITRE ATT&CK. T1572 Protocol Tunneling. https://attack.mitre.org/techniques/T1572/ (verified 20 November 2026)
7. MITRE ATT&CK. T1090.002 Proxy: External Proxy. https://attack.mitre.org/techniques/T1090/002/ (verified 20 November 2026)
8. Salesforce. JARM active TLS server fingerprinting. https://github.com/salesforce/jarm (verified 20 November 2026)
9. fatedier/frp. FRP README, TOML configuration since v0.52.0. https://github.com/fatedier/frp (verified 20 November 2026)
10. GOST. TLS tutorial. https://gost.run/en/tutorials/tls/ (verified 20 November 2026)
11. wikiZ/RedGuard. README. https://github.com/wikiZ/RedGuard (verified 20 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 (2) 33

B. Detection rules (4 files) 34

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
R33-IOC-01	JARM	21d14d00021d21d21c42d43d00041d0eea9d6a7e6a07bccd28e28ed272ad91	1500	Daybreak paired C2 listener, port 1500 · hunt with port and certificate	HUNT
R33-IOC-02	JARM	3fd3fd0003fd3fd00042d42d00000000f78d2dc0ce6e5bbc5b8149a4872356	1501	Daybreak paired C2 listener, port 1501 · hunt with port and certificate	HUNT

B Detection Rules

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

odin_gpu_sniper_amneziawg_xui_and_frp_stack.yar

YARA

```
/*
   Odin by Brandefense - Monograph 33 / 35 YARA Artifact Suite
   Scope: Bounded structural detection rules for Python GPU spot-marketplace sniper bots,
          AmneziaWG junk-packet/header-obfuscated tunnel profiles, FRP v0.52+ TOML reverse-proxy
          configs, and co-located compute-and-tunnel workbench manifests.
   Classification: TLP:CLEAR
*/

rule ODIN_Clore_GPU_Marketplace_Sniper_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "33 / 35"
        classification = "TLP:CLEAR"
        purpose = "hunt"
        description = "Matches Python automated bidding and orderbook polling scripts targeting decentralized GPU
spot marketplaces"
    strings:
        $api1 = "api.clore.ai" ascii
        $ord1 = "create_order" ascii
        $ord2 = "spot" ascii
        $ord3 = "marketplace" ascii
        $hdr1 = "auth" ascii
    condition:
        filesize < 128KB and $api1 and 2 of ($ord1, $ord2, $ord3, $hdr1)
}

rule ODIN_AmneziaWG_Obfuscated_Tunnel_Config_or_Installer
{
    meta:
        author = "Odin by Brandefense"
        monograph = "33 / 35"
        classification = "TLP:CLEAR"
        purpose = "hunt"
        description = "Matches AmneziaWG client/server configurations or installers defining junk-packet (Jc/
Jmin/Jmax) and header (H1-H4) obfuscation"
    strings:
        $if1 = "[Interface]" ascii
        $awg1 = "amneziawg" ascii nocase
        $jc = "Jc" ascii
        $jmin = "Jmin" ascii
        $jmax = "Jmax" ascii
        $h1 = "H1" ascii
        $s1 = "S1" ascii
    condition:
        filesize < 64KB and ($if1 or $awg1) and $jc and $jmin and $jmax and ($h1 or $s1)
}

rule ODIN_FRP_TOML_Reverse_Proxy_Config_Artifact
{
    meta:
        author = "Odin by Brandefense"
        monograph = "33 / 35"
        classification = "TLP:CLEAR"
        purpose = "hunt"
        description = "Matches modern FRP v0.52+ TOML client and server reverse-proxy configurations"
    strings:
        $s1 = "serverAddr" ascii
        $s2 = "serverPort" ascii
        $b1 = "bindPort" ascii
```

```
        $p1 = "[[proxies]]" ascii
        $p2 = "remotePort" ascii
        $a1 = "auth.token" ascii
    condition:
        filesize < 64KB and (($s1 and $s2 and ($p1 or $p2 or $a1)) or ($b1 and $a1))
}

rule ODIN_Compute_Tunnel_Workbench_Manifest_or_Miner_Script
{
    meta:
        author = "Odin by Brandefense"
        monograph = "33 / 35"
        classification = "TLP:CLEAR"
        purpose = "hunt"
        description = "Matches GPU miner launch scripts referencing stratum pools and GPU runtime flags"
    strings:
        $m1 = "stratum+" ascii
        $m2 = "nvidia-smi" ascii
        $m3 = "--algo" ascii
    condition:
        filesize < 256KB and $m1 and ($m2 or $m3)
}
```

odin_gpu_sniper_amneziawg_xui_and_ghostsocks.rules

SURICATA

```
# Odin by Brandefense - Monograph 33 / 35 Suricata 7.x Detection Suite
# Coverage: Automated GPU spot-market API polling, AmneziaWG packages, scripts and client configs,
#           X-UI panel archives, GOST (CN=gost.run) TLS certs, FRP reverse-proxy binaries/configs,
#           and open directory listings exposing tunnel tooling
# Classification: TLP:CLEAR

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R33] HUNT Non-Browser Python Spot-Order Polling to
Clore GPU API"; flow:established,to_server; http.host; bsize:12; content:"api.clore.ai"; fast_pattern;
http.user_agent; bsize:>14; content:"python-requests"; classtype:policy-violation; sid:9533002; rev:2;
metadata:created_at 2026_11_20, updated_at 2026_11_20, deployment tls_egress_proxy, policy behavioral_hunt;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R33] AmneziaWG Obfuscated WireGuard Package or Script
Retrieval"; flow:established,to_server; http.method; bsize:3; content:"GET"; http.uri; bsize:>11;
content:"amneziawg"; fast_pattern; pcre:"/amneziawg[^\/]*\.(sh|tar\.gz|deb|rpm)$/i"; classtype:policy-violation;
sid:9533003; rev:3; metadata:created_at 2026_11_20, updated_at 2026_11_22, deployment perimeter, tactic
command_and_control;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R33] Open-Directory AmneziaWG Client Tunnel Profile
Retrieval"; flow:established,to_server; http.method; bsize:3; content:"GET"; http.uri; bsize:>8; content:"awg";
fast_pattern; content:".conf"; distance:1; pcre:"/awg[^\/]*\/[^\/]+\.conf$/i"; classtype:trojan-activity; sid:
9533004; rev:2; metadata:created_at 2026_11_20, updated_at 2026_11_22, deployment perimeter, tactic
command_and_control;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R33] X-UI Multi-Protocol Tunnel Panel Archive
Retrieval"; flow:established,to_server; http.method; bsize:3; content:"GET"; http.uri; bsize:>11; content:"x-ui";
fast_pattern; content:".tar.gz"; distance:0; pcre:"/x-ui[^\/]*\.tar\.gz$/i"; classtype:trojan-activity; sid:
9533005; rev:3; metadata:created_at 2026_11_20, updated_at 2026_11_22, deployment perimeter, tactic
command_and_control;)

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R33] GOST Tunnel Default Server Certificate Observed
(CN=gost.run)"; flow:established,to_client; tls.cert_subject; bsize:>10; content:"CN=gost.run"; fast_pattern;
classtype:trojan-activity; sid:9533006; rev:1; metadata:created_at 2026_11_20, deployment perimeter, tactic
command_and_control;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R33] Open-Directory FRP Reverse Proxy Binary or TOML
Config Retrieval"; flow:established,to_server; http.method; bsize:3; content:"GET"; http.uri; bsize:>4;
content:"frp"; fast_pattern; pcre:"/\/frp[cs](\.toml)?$/"; classtype:trojan-activity; sid:9533007; rev:2;
metadata:created_at 2026_11_20, updated_at 2026_11_22, deployment perimeter, tactic command_and_control;)
```

```
alert http $HOME_NET any -> any any (msg:"ODIN [R33] Open-Directory Tunnel Tooling Listing Served From Home Net";
flow:established,to_client; http.stat_code; bsize:3; content:"200"; http.response_body; bsize:>32;
content:"<title>"; fast_pattern; pcre:"/(Directory listing for|Index of) \//"; pcre:"/(amneziawg|awg-quick|x-ui|
frpc|frps|ghostsocks)/i"; classtype:attempted-recon; sid:9533008; rev:3; metadata:created_at 2026_11_20,
updated_at 2026_11_22, deployment perimeter, tactic reconnaissance;)
```

sentinel_gpu_sniper_amneziawg_xui_and_socks_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 33 / 35 Microsoft Sentinel & Defender XDR Hunt
// Correlates AmneziaWG/X-UI/FRP/GOST/GhostSocks tunnel processes and GPU sniper/miner scripts
// with outbound connections to confirmed C2 relays and Daybreak :1500/:1501 pairs
// Classification: TLP:CLEAR

let Lookback = 7d;
let ApprovedGpuClusterHosts = dynamic(["ML-BATCH-SCHEDULER-01", "GPU-SLURM-HEAD-01"]);
let ApprovedNetopsBastions = dynamic(["NETOPS-VPN-GW-01", "SEC-VALIDATION-LNX-01"]);
let ConfirmedRelayIps = dynamic([]);
let TunnelAndSniperMarkers = dynamic([
    "api.clore.ai", "awg-quick", "amneziawg-go", "amneziawg", "/usr/local/x-ui/",
    "frpc.toml", "frps.toml", "socks5+tls://", "socks5+wss://", "ghostsocks",
    "gost.run", "stratum+"
]);
let SuspiciousTunnelOrSniperProcs = DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedGpuClusterHosts)) and not(DeviceName in~ (ApprovedNetopsBastions))
    | where FileName in~ ("amneziawg-go", "awg-quick", "x-ui", "frpc", "frps", "gost", "ghostsocks")
        or ProcessCommandLine has_any (TunnelAndSniperMarkers)
    | project ProcTime = Timestamp, DeviceId, DeviceName, InitiatingProcessFileName, FileName,
ProcessCommandLine;
DeviceNetworkEvents
| where Timestamp >= ago(Lookback)
| where not(DeviceName in~ (ApprovedGpuClusterHosts)) and not(DeviceName in~ (ApprovedNetopsBastions))
| where RemoteIP in (ConfirmedRelayIps)
    or RemoteUrl in~ ("api.clore.ai", "gost.run")
    or RemoteUrl startswith "api.clore.ai/" or RemoteUrl startswith "gost.run/"
    or tostring(parse_url(RemoteUrl).Host) in~ ("api.clore.ai", "gost.run")
    or (RemotePort in (1500, 1501) and InitiatingProcessFileName !in~ ("chrome.exe", "msedge.exe", "firefox"))
| summarize
    NetFirstSeen = min(Timestamp),
    NetLastSeen = max(Timestamp),
    DistinctRemoteIps = dcount(RemoteIP),
    ConfirmedRelayHits = countif(RemoteIP in (ConfirmedRelayIps)),
    MarketplaceHits = countif(RemoteUrl =~ "api.clore.ai" or RemoteUrl startswith "api.clore.ai/" or
tostring(parse_url(RemoteUrl).Host) =~ "api.clore.ai"),
    GostHits = countif(RemoteUrl =~ "gost.run" or RemoteUrl startswith "gost.run/" or
tostring(parse_url(RemoteUrl).Host) =~ "gost.run"),
    PortPairHits = countif(RemotePort in (1500, 1501) and InitiatingProcessFileName !in~ ("chrome.exe",
"msedge.exe", "firefox")),
    SampleEndpoints = make_set(strcat(RemoteIP, ":", tostring(RemotePort)), 12),
    SampleUrls = make_set(RemoteUrl, 8),
    ConnectionCount = count()
    by DeviceId, DeviceName, InitiatingProcessFileName
| join kind=fullouter (SuspiciousTunnelOrSniperProcs) on DeviceId, DeviceName
| extend ResolvedDevice = coalesce(DeviceName, DeviceName1)
| where isnotempty(ProcessCommandLine) or ConfirmedRelayHits > 0 or MarketplaceHits > 0 or PortPairHits > 0 or
GostHits > 0
| extend AlertDisposition = case(
    ConfirmedRelayHits > 0, "ISOLATE_CONFIRMED_RELAY_OR_DAYBREAK_C2",
    ProcessCommandLine has_any ("api.clore.ai", "stratum+") and ProcessCommandLine has_any ("amneziawg", "x-ui",
"frpc", "frps", "gost", "ghostsocks"), "HUNT_COMBINED_SNIPER_TUNNEL_COMMAND_LINE",
    isnotempty(ProcessCommandLine), "HUNT_EVASIVE_TUNNEL_OR_GPU_SNIPER_STAGING",
    "REVIEW_DAYBREAK_PORT_PAIR_OR_MARKETPLACE_EGRESS"
)
| project NetFirstSeen, NetLastSeen, ProcTime, ResolvedDevice, AlertDisposition, InitiatingProcessFileName,
FileName, ProcessCommandLine, SampleEndpoints, SampleUrls, ConnectionCount
| sort by coalesce(ProcTime, NetLastSeen) desc
```

sigma_amneziawg_xui_frp_and_gpu_miner_staging.yml

SIGMA

```
title: AmneziaWG Obfuscated WireGuard or X-UI Multi-Protocol Tunnel Provisioning on Linux
id: 4b92e1a7-6d18-4f53-9c42-202611203301
status: experimental
description: Detects execution of AmneziaWG junk-packet obfuscated WireGuard installers, userspace daemons, or X-
UI multi-protocol Xray tunnel management panels outside approved network bastion hosts.
author: Odin by Brandefense
date: 2026/11/20
tags:
  - attack.command-and-control
  - attack.t1572
  - attack.t1090.002
logsource:
  category: process_creation
  product: linux
detection:
  selection_img:
    Image|endswith:
      - '/amneziawg-go'
      - '/awg-quick'
      - '/x-ui'
  selection_cli:
    CommandLine|contains:
      - 'amneziawg'
      - 'awg-quick'
      - '/usr/local/x-ui/'
  filter_netops_bastion:
    Computer:
      - 'NETOPS-VPN-GW-01'
      - 'NETOPS-TUNNEL-LAB-01'
  condition: (selection_img or selection_cli) and not 1 of filter_*
falsepositives:
  - Authorized network engineering bastions evaluating DPI-resilient VPN or Xray proxy configurations
level: high
---
title: FRP Reverse Proxy TOML Stack or GOST and GhostSocks SOCKS5 Backconnect Execution
id: 8d14c3f9-2a75-4e91-b038-202611203302
status: experimental
description: Identifies execution of Fast Reverse Proxy (frpc/frps) with TOML configuration profiles, GOST TLS
tunnels, or GhostSocks backconnect SOCKS5 relays on non-proxy enterprise hosts.
author: Odin by Brandefense
date: 2026/11/20
tags:
  - attack.command-and-control
  - attack.t1572
  - attack.t1090.002
logsource:
  category: process_creation
  product: linux
detection:
  selection_bin:
    Image|endswith:
      - '/frpc'
      - '/frps'
      - '/gost'
      - '/ghostsocks'
  selection_args:
    CommandLine|contains:
      - 'frpc.toml'
      - 'frps.toml'
      - 'socks5+tls://'
      - 'socks5+wss://'
      - 'gost.run'
  filter_approved_proxy:
    Computer:
      - 'DEV-INGRESS-PROXY-01'
```

```
      - 'SEC-RELAY-TEST-01'
  condition: (selection_bin or selection_args) and not 1 of filter_*
falsepositives:
  - Registered developer ingress tunnels or authorized red-team relay validation servers
level: high
```

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/gpu-marketplace-snipers-and-evasive-tunnels/
Short Read · brandefense.io/odin-research/blog/gpu-snipers-amneziawg-and-x-ui/
Defense Kit · brandefense.io/odin-research/defense-kit/