ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 21 / 30 · C2 INFRASTRUCTURE

Relay Meshes and Six-Port Extortion Hubs

Mapping Feed-Labeled APT and BianLian Infrastructure

How eight threat-feed clusters split VPN relays, staging hosts, collection gates and multi-port Go listeners across cloud providers

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

79
Unique endpoints on 50 IPv4 hosts, across eight feed-labeled clusters

9
Confirmed C2 endpoints on 8 hosts (7 Kimsuky-labeled, 2 ShadowPad-labeled); the only addresses published

70
Endpoints held on watchlists: 57 suspected, 8 candidates and 5 on dynamic ISPs; addresses withheld

14
Hosts with two or more ports, holding 43 of the 79 endpoints (54.4%)

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

How eight threat-feed clusters split VPN relays, staging hosts, collection gates and multi-port Go listeners across cloud providers

01	Executive Summary	04
02	Labels, Groups and Ports	07
03	Relay and Backdoor Clusters: RedJuliett, ShadowPad, PlugX, Gh0stRAT	10
04	Staging and Collection Clusters: Kimsuky and LazarusGroup	13
05	Extortion Listeners: BianLian and Coruna	15
06	Fingerprints That Survive Rotation	18
07	What Not to Block	21
08	What Defenders Should Do	23
09	Detection Rules	25
10	Indicators	28
11	Scope and Limits	31
12	Sources	33
A	Indicators of compromise (12)	35
B	Detection rules (4 files)	36

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense mapped 79 endpoints on 50 hosts across eight threat-feed clusters. Relay clusters split VPN bridges from staging hosts. Extortion clusters stack six TLS ports on one server. Only 9 endpoints on 8 hosts are confirmed command-and-control (C2); the other 70 stay on watchlists. Block the 9 confirmed endpoints, keep dynamic ISPs and shared hosting on short watchlists, and alert on side-loaded DLLs, SoftEther tunnels and rclone transfers.

01

CHAPTER 01

Executive Summary

IN SHORT

In a two-week observation window ending 18 November 2026, Odin by Brandefense mapped 79 endpoints on 50 hosts across eight threat-feed clusters. Relay clusters split VPN bridges from staging hosts. Extortion clusters stack six TLS ports on one server. Only 9 endpoints on 8 hosts are confirmed command-and-control (C2); the other 70 stay on watchlists. Block the 9 confirmed endpoints, keep dynamic ISPs and shared hosting on short watchlists, and alert on side-loaded DLLs, SoftEther tunnels and rclone transfers.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

This report looks at the servers behind eight clusters that threat feeds label RedJuliett, ShadowPad, PlugX, Gh0stRAT, Kimsuky, LazarusGroup, BianLian and Coruna. These names are feed labels. They are not our attribution. Public advisories describe how these groups work; this report does not tie any address to an actor named in those advisories.

The study covers 79 unique endpoints (IP and port pairs) on 50 IPv4 hosts. No IPv4 address appears in more than one cluster.

1. **Relay-labeled clusters split VPN bridges from staging hosts across providers.**
   The RedJuliett-labeled hosts separate two Azure US machines on port 3389 from three SoftEther VPN bridges on port 443 in India, Germany and Hong Kong. ShadowPad-labeled hosts add two confirmed C2 servers in Japan and the Netherlands. PlugX and Gh0stRAT labels sit on Tencent, Alibaba, telecom networks and small-office routers.

2. **Seven confirmed Kimsuky-labeled endpoints form three matched pairs.**
   Two AWS servers share one JARM value. Two Akamai servers on port 999 share one certificate and one JARM. Two UCloud servers in adjacent Korean subnets serve plain HTTP on port 80. The LazarusGroup label sits on one shared web-and-mail host with six open ports.

3. **BianLian-labeled hubs stack six TLS ports behind one fingerprint.**
   Two HostPapa servers expose the same ports (443, 666, 888, 1122, 5566, 7777). All 12 TLS sockets share one JARM value. Public advisories describe BianLian pairing Go backdoors with rclone bulk transfers instead of file encryption, so classic ransomware alerts stay quiet.

Three Decisions for Leaders

1. **Block confirmed C2 hosts, but keep shared and dynamic addresses on watchlists.** Block the 9 confirmed endpoints. Hold dynamic telecom addresses and shared web-and-mail servers on time-bound SIEM watchlists, so you do not cut off unrelated customers.
2. **Tie DLL side-loading and VPN tools to outbound traffic.** Alert when a signed program in a user-writable folder loads an unsigned DLL and reaches the internet within 15 minutes. Alert on SoftEther VPN tools outside named jump hosts.
3. **Watch for data leaving, not only for files being encrypted.** Restrict rclone and SFTP transfers to named backup servers. Watch outbound TLS sessions to unusual high ports.

The Study in Numbers

Number	What it is
79	Unique endpoints on 50 IPv4 hosts, across eight feed-labeled clusters
9	Confirmed C2 endpoints on 8 hosts (7 Kimsuky-labeled, 2 ShadowPad-labeled); the only addresses published
70	Endpoints held on watchlists: 57 suspected, 8 candidates and 5 on dynamic ISPs; addresses withheld
14	Hosts with two or more ports, holding 43 of the 79 endpoints (54.4%)
12	TLS sockets on two BianLian-labeled HostPapa hosts that share one JARM value
54	Endpoints with TLS (68.4%), showing 22 distinct JARM values
0	IPv4 addresses shared between clusters

How to use this report.

Security leaders can read Chapter 1, Chapter 7 and Chapter 8. SOC and detection engineers should start with Chapters 8 to 10. CTI analysts will find cluster detail in Chapters 2 to 6.

02

CHAPTER 02

Labels, Groups and Ports

IN SHORT

The 79 endpoints fall into three groups: relay and backdoor clusters, staging and collection clusters, and extortion listeners. Ports cluster inside each group, but no address is shared between groups.

INSIDE THIS CHAPTER

What the cluster names mean 07

Three groups 07

Clusters at a glance 08

Many ports, few hosts 08

02 Labels, Groups and Ports

What the cluster names mean

Each cluster name is the label a threat feed attaches to a set of hosts. The labels rest on protocol banners, certificates, port layouts and JARM fingerprints. JARM is a fingerprint of how a server answers TLS handshakes; servers with the same software and settings tend to share it.

A label is a lead, not an attribution. Where this report cites CISA or MITRE ATT&CK, it does so to explain known tradecraft. None of those sources is used to tie a specific address to a named actor.

Three groups

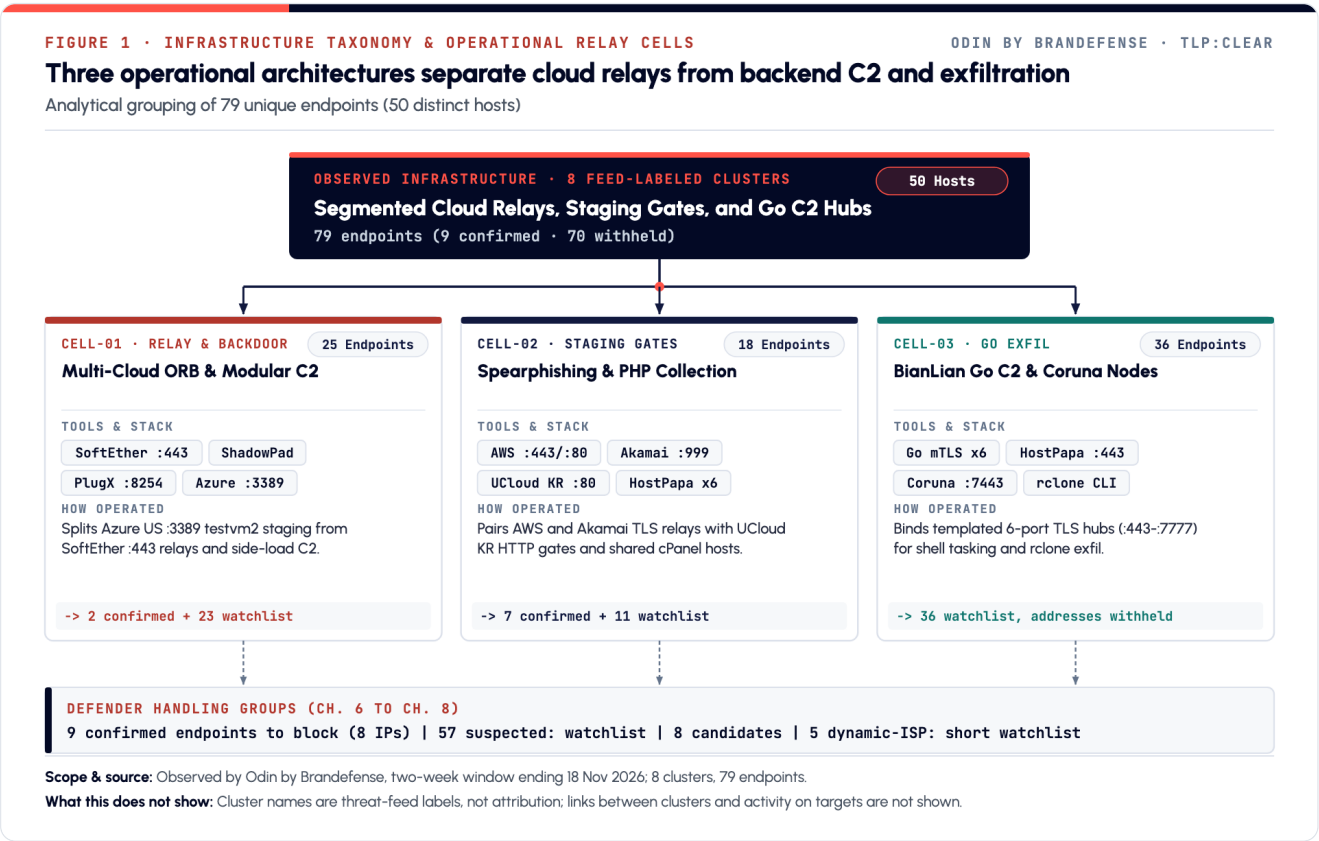


Figure 1: Three operational architectures separate cloud relays from backend C2 and exfiltration

The grouping in Figure 1 is our analytical model. It sorts clusters by the role their servers seem to play. It does not claim that clusters in the same group work together.

Group	Clusters	Endpoints (hosts)	Confirmed C2
Relay and backdoor	RedJuliett, ShadowPad, PlugX, Gh0stRAT	25 (21)	2 (ShadowPad)
Staging and collection	Kimsuky, LazarusGroup	18 (11)	7 (Kimsuky)
Extortion listeners	BianLian, Coruna	36 (18)	0

Public reporting links the first two groups to China and North Korea. Together they hold 43 endpoints on 32 hosts, including all 9 confirmed C2 endpoints.

Clusters at a glance

Cluster	Role	Endpoints (hosts)	Ports per host	Confirmed / suspected / candidate / dynamic ISP
BianLian	Go TLS C2, data-theft extortion	29 (14)	2.07	0 / 24 / 4 / 1
Gh0stRAT	Binary zlib RAT, edge relays	9 (6)	1.50	0 / 3 / 4 / 2
ShadowPad	Modular backdoor (S0596)	3 (3)	1.00	2 / 1 / 0 / 0
PlugX	Side-loaded modular C2 (S0013)	7 (6)	1.17	0 / 7 / 0 / 0
Kimsuky	Staging and PHP gates (G0094)	12 (10)	1.20	7 / 3 / 0 / 2
LazarusGroup	Shared web and mail host (G0032)	6 (1)	6.00	0 / 6 / 0 / 0
RedJuliett	SoftEther relays, Azure staging	6 (6)	1.00	0 / 6 / 0 / 0
Coruna	Paired 7000 to 7443 listeners	7 (4)	1.75	0 / 7 / 0 / 0

Many ports, few hosts

Inside each cluster, operators reuse hosts. Fourteen hosts bind two or more ports and hold 43 of the 79 endpoints (54.4%). The two BianLian-labeled hubs (both bind the same six TLS ports; one also opens port 80) and the six-port LazarusGroup-labeled host lead this pattern.

CHAPTER 03

Relay and Backdoor Clusters: RedJuliett, ShadowPad, PlugX, Gh0stRAT

IN SHORT

These four clusters hold 25 endpoints on 21 hosts. They place VPN bridges, staging machines and backdoor listeners on separate providers and regions; two ShadowPad-labeled servers are confirmed C2.

INSIDE THIS CHAPTER

RedJuliett label: staging hosts apart from VPN bridges 10
ShadowPad label: two confirmed servers 11
PlugX label: custom and DNS ports 11
Gh0stRAT label: why a label alone is not enough 11

03 Relay and Backdoor Clusters: RedJuliett, ShadowPad, PlugX, Gh0stRAT

CISA advisory AA26-281A describes operators routing intrusion traffic through operational relay box (ORB) networks. An ORB network is a mesh of leased virtual servers, SoftEther VPN bridges and edge routers that hides where traffic really comes from. Behind such relays, public reporting describes modular backdoors loaded through DLL side-loading (T1574.002): ShadowPad (S0596), PlugX (S0013) and Gh0stRAT (S0032).

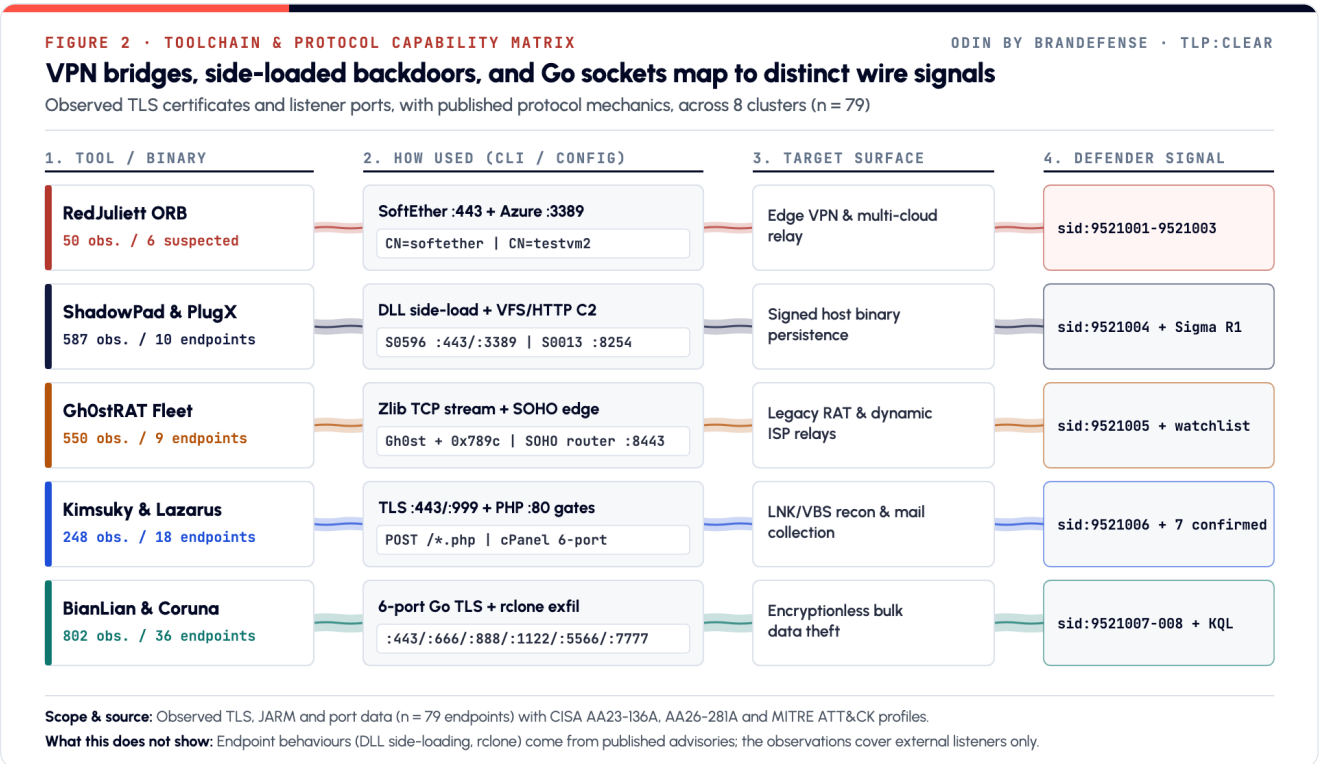


Figure 2: VPN bridges, side-loaded backdoors, and Go sockets map to distinct wire signals

RedJuliett label: staging hosts apart from VPN bridges

All six RedJuliett-labeled endpoints are suspected, on six hosts. They fall into two roles.

Role	Hosts	Hosting	Certificate subject
Windows staging machine on 3389	2	Microsoft Azure, United States	`CN=testvm2` (self-signed, identical on both)
SoftEther VPN bridge on 443	3	Microsoft Azure India; Hetzner Online, Germany; Alibaba, Hong Kong	`CN=softether`
Relay on 443	1	Regional telecom carrier, North Macedonia	Domain withheld

The matching `CN=testvm2` subjects suggest a templated Windows staging image. The `CN=softether` subject is the SoftEther default. The India and Hong Kong bridges also share a SoftEther JARM prefix (`06d06d07d06d06d06c42d42d000000...`). Their addresses are withheld because SoftEther and Azure test machines have common legitimate uses.

ShadowPad label: two confirmed servers

Two dedicated servers carry the ShadowPad label and are confirmed C2.

Endpoint	Hosting	Country	Signal
130.94.65[.]79:443	LIGHT NODE LIMITED	Japan	HTTPS C2; JARM `21d14d...6cbe21`
45.32.237[.]137:3389	The Constant Company, LLC	Netherlands	Windows hostname certificate `CN=WIN-JRJ4OQAGQ2I` ; serves a non-RDP listener on the RDP port

A third ShadowPad-labeled host on Alibaba Hong Kong is suspected. It presents `CN=CloudFlare Origin Certificate`, which helps it blend in with CDN-style TLS. Its address is withheld.

PlugX label: custom and DNS ports

All seven PlugX-labeled endpoints, on six hosts, are suspected. Five of the seven sit in China or Hong Kong.

Port(s)	Hosting	Note
1033 and 8254	Tencent Cloud, China	One host with both ports
8000	Tencent Cloud and Alibaba, China	Two hosts
443	Hetzner Online, Germany	`CN=Root CA`
80	CTG Server Limited, Hong Kong	
53	Nocix, LLC, United States	Listener on the DNS port, which permissive DNS firewall rules may let through

Gh0stRAT label: why a label alone is not enough

The Gh0stRAT label covers 9 endpoints on 6 hosts. It shows why a feed match cannot drive an automatic block.

- The feed marks one endpoint as confirmed, but it sits on Chinanet, a dynamic carrier whose addresses rotate. This report holds it on the dynamic-ISP watchlist, not as confirmed C2.
- Another sits on Chunghwa Telecom, Taiwan, behind a consumer router certificate. That router may be a relay or a victim.
- One host on CTG Server Limited, Hong Kong, exposes the mail ports 465, 993 and 995.
- Four more endpoints are not corroborated. One is a server management controller, which can match a banner rule with no malware present.

None of these addresses is published.

CHAPTER 04

Staging and Collection Clusters: Kimsuky and LazarusGroup

IN SHORT

The Kimsuky label covers 12 endpoints on 10 hosts, 7 of them confirmed and arranged in three matched pairs. The LazarusGroup label covers one shared web-and-mail host with six ports.

INSIDE THIS CHAPTER

Three confirmed Kimsuky-labeled pairs 13
Kimsuky-labeled endpoints held on watchlists 13
LazarusGroup label: one shared host 13

04 Staging and Collection Clusters: Kimsuky and LazarusGroup

MITRE ATT&CK describes Kimsuky (G0094, also tracked as APT43 and Emerald Sleet) and Lazarus Group (G0032). Public reporting links both to North Korea. It describes credential harvesting, script-based spearphishing (LNK, CHM and VBS files) and light HTTP or PHP collection relays.

Three confirmed Kimsuky-labeled pairs

The seven confirmed endpoints sit on six hosts. Each pair shares at least one technical trait.

Pair	Endpoints	Hosting	What they share
AWS TLS pair	3.140.83[.]52:443 (US); 3.104.232[.]81:443 and :80 (AU)	Amazon.com, Inc.	JARM `29d29d00029d29d00041d41d00041dcb3f8752a2d70d1dbf446fc46ab5df96`
Akamai port-999 pair	172.233.130[.]140:999; 172.236.231[.]228:999 (US)	Akamai Technologies (Linode)	Same domain certificate and JARM `29d29d20d29d29d21c29d29d29d29d89e97ba53da7e1515fb732a85cb371a3`
UCloud HTTP pair	118.194.248[.]246:80; 118.194.249[.]154:80 (KR)	UCLOUD INFORMATION TECHNOLOGY (HK) LIMITED	Adjacent `/24` subnets, no TLS

The shared JARM on the AWS pair points to the same web server and OpenSSL setup in two AWS regions. The Akamai pair reuses one certificate on the non-standard port 999. The UCloud pair matches the plain HTTP collection gates that public reporting describes for this group.

The certificate domains on these hosts are not published.

Kimsuky-labeled endpoints held on watchlists

Five more Kimsuky-labeled endpoints sit on four hosts. None is published.

- One host with ports 4003 and 4004 sits on Deutsche Telekom in Germany. It is a dynamic subscriber network, so the address goes on a 72-hour watchlist only.
- Two suspected 443 endpoints, on Azure Netherlands and AWS US, share the JARM `2ad2ad00000000022c42d42d0000001f361a1bf80ed39a1e352adeb016a07b`. Many unrelated servers share this value, so use it only alongside other signals.
- One suspected endpoint on port 9090, hosted in Korea, presents a cryptocurrency-themed certificate (`CN=coin`).

LazarusGroup label: one shared host

All six LazarusGroup-labeled endpoints sit on one HostPapa server in the United States. The ports are web (80, 443) and mail (143, 465, 587, 993). This profile fits a shared web-and-mail hosting account that may be abused to stage lures or PHP drop pages.

A shared host also serves many unrelated small-business sites and mailboxes. Blocking the whole address would cut them off. The host stays on a watchlist and its address is withheld.

CHAPTER 05

Extortion Listeners: BianLian and Coruna

IN SHORT

BianLian and Coruna labels hold 36 endpoints on 18 hosts, 45.6% of the 79. BianLian-labeled hubs stack six TLS ports per server; public advisories describe the group stealing data with rclone instead of encrypting it.

INSIDE THIS CHAPTER

BianLian label: 29 endpoints on 14 hosts 15
The twin hubs 16
Coruna label: paired ports from 7000 to 7443 16

05 Extortion Listeners: BianLian and Coruna

CISA, FBI and ASD-ACSC advisory AA23-136A describes BianLian moving away from file encryption to pure data-theft extortion. The group uses custom Go backdoors, credential theft and bulk transfers through rclone, FTP and SFTP. Skipping encryption avoids mass file renaming, shadow-copy deletion and ransomware canary alerts.

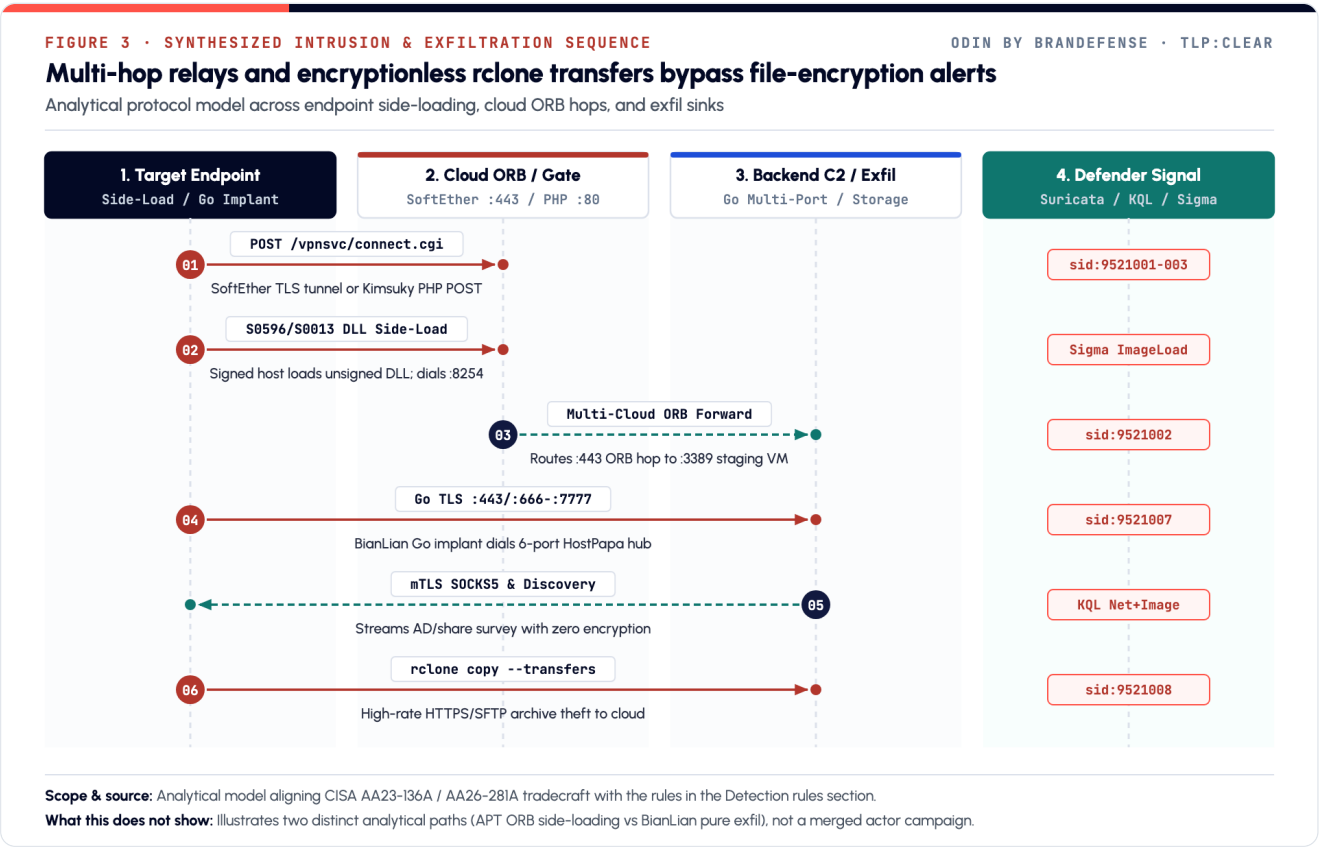


Figure 3: Multi-hop relays and encryptionless rclone transfers bypass file-encryption alerts

Figure 3 is an analytical model. It contrasts two paths, relay-based side-loading and BianLian-style data theft, and maps each step to a detection rule. It does not show one merged campaign.

BianLian label: 29 endpoints on 14 hosts

None of the 29 BianLian-labeled endpoints is confirmed: 24 are suspected, 4 are candidates and 1 sits on a dynamic ISP. They concentrate in North America.

Country	Endpoints
United States	22 (75.9%)
Canada	2
Germany	2
France	1
Poland	1
Russia	1

Four groups stand out. All addresses are withheld.

Group	Endpoints (hosts)	Hosting and ports	TLS fingerprint
Twin six-port hubs	13 (2)	HostPapa US: 443, 666, 888, 1122, 5566, 7777 on both; one also 80	All 12 TLS sockets share JARM `28d28d28d00028d1ec42...3386b`
Go TLS trio	3 (3)	Azure US 443, AWS US 443, and a small telecom network 8899	All 3 share JARM `22d22d00022d...63261`
VPS and RDP nodes	9 (6)	AWS Germany, OVH France 8080, M247 3389, and a host with 21, 110, 465 and 3389	Mixed; the M247 RDP host shows a custom certificate name
Candidates	4 (3)	OVH Canada 8443 and 8444; Russia and Poland 443	Not corroborated

The twin hubs

The two HostPapa hubs are the clearest signature in this report. Both bind the same six TLS ports. Each has its own domain certificate. All 12 TLS sockets return the same JARM value: `28d28d28d00028d1ec42d42d0000005557fe5c0eece598995bd43427a3386b`.

Six ports on one daemon give implants fallback routes. If a firewall inspects or blocks 443, an implant can try 5566, 7777 or 1122.

Coruna label: paired ports from 7000 to 7443

Coruna is a smaller label for a modular C2 cluster with paired control and TLS ports. All seven endpoints, on four hosts, are suspected.

Hosts	Hosting	Ports	Shared traits
2	Antbox Networks Limited, Singapore	7001 and 7443 on each	Same domain certificate and JARM `28d28d28d00028d00042d42d000000831b6af40378e2dd35eeac4e9311926e` on both 7443 listeners
1	cognetcloud INC, Hong Kong	7000 and 7001	
1	HostPapa, United States	7000	

The Singapore JARM also appears on one BianLian-labeled host in Germany. That shared value reflects a common TLS stack, not a proven link between the clusters. Ports in this range also match developer reverse proxies, so these hosts stay on a watchlist.

CHAPTER 06

Fingerprints That Survive Rotation

IN SHORT

Operators move between Azure, AWS, Alibaba, Hetzner and HostPapa, but their deployment templates leave stable certificate names, JARM values and port sets. Hunt those, not single addresses.

INSIDE THIS CHAPTER

Certificates and JARM values 18

How the endpoint side works 19

06 Fingerprints That Survive Rotation



Figure 4: Certificate subjects and JARM clusters expose segmented relays and 6-port Go hubs

Certificates and JARM values

Of the 79 endpoints, 54 (68.4%) present TLS. They show 22 distinct JARM values. The five most common JARM values cover 21 TLS endpoints (38.9% of all TLS endpoints) and line up with clusters. The table lists the clearest certificate and JARM patterns.

Cluster and role	Certificate pattern	JARM prefix	Where it appears
BianLian six-port hubs	Per-host domain certificates (withheld)	`28d28d28d00028d1ec 42d42d...3386b`	12 sockets on 2 HostPapa US hosts
BianLian Go trio	Empty or per-host (withheld)	`22d22d00022d22d22c 42d42d...63261`	Azure US 443, AWS US 443, telecom 8899
Kimsuky AWS pair	Per-host domain certificates (withheld)	`29d29d00029d29d000 41d41d...5df96`	3.140.83[.]52 (US) and 3.104.232[.]81 (AU), both confirmed
Kimsuky Akamai pair	One shared domain certificate (withheld)	`29d29d20d29d29d21c 29d29d...371a3`	172.233.130[.]140 and 172.236.231[.]228, both confirmed
RedJuliett relays and staging	`CN=softether` (3), `CN=testvm2` (2)	`06d06d07d06d06d06c 42d42d...`	JARM on the India and Hong Kong SoftEther bridges; certificates on all 3 bridges and 2 Azure US staging hosts
Coruna Singapore pair	One shared domain certificate (withheld)	`28d28d28d00028d000 42d42d...`	2 Antbox Singapore hosts on 7443

Two lessons follow for attack surface and network detection teams.

Default names leak roles.

The RedJuliett-labeled hosts kept the default `CN=softether` on all three bridges and `CN=testvm2` on both staging machines. The confirmed ShadowPad-labeled host 45.32.237[.]137 shows an auto-generated Windows name, `CN=WIN-JRJ4OQAGQ2I`.

Certificate reuse links distant hosts.

The two confirmed Kimsuky-labeled Akamai servers share one certificate. The two Coruna-labeled Singapore servers share another.

How the endpoint side works

External observations show only the server side. Public advisories explain how endpoint implants talk to these servers. Each point below names the rule that covers it.

- **DLL side-loading (ShadowPad and PlugX).** Both families drop three files in a user-writable folder such as `C:\ProgramData\`, `%PUBLIC%\Downloads\` or `%TEMP%`. The files are a signed third-party program, a malicious unsigned DLL with an expected library name, and an encrypted payload. The signed program loads the DLL. PlugX HTTP beacons send `POST` requests with `X-Session:` and `X-Size:` headers (SID 9521004).
- **SoftEther tunnels (ORB tradecraft).** Per AA26-281A, operators install SoftEther VPN clients, often renamed or run from staging folders, that connect over 443 to external bridges. SoftEther wraps network frames in HTTPS `POST /vpnsvc/connect.cgi` sessions with a `VPNCONNECT` handshake (SID 9521003). Without TLS inspection or process auditing (`vpnclient`, `vpncmd`, `SEVPNCLIENT`), this traffic passes most firewalls.
- **Go implants and rclone (BianLian).** Per AA23-136A, BianLian uses custom Go backdoors and rclone for bulk theft. Renamed rclone copies keep `OriginalFileName = rclone.exe` in their version data and send a default `rclone/v...` user agent (SID 9521008). Typical flags are `--transfers`, `--tpslimit` and `--no-check-certificate`.

07

CHAPTER 07

What Not to Block

IN SHORT

Turning feed labels straight into firewall blocks would cut off broadband customers, shared mail hosts and ordinary server hardware. Only 9 of the 79 endpoints (11.4%) are ready to block.

07 What Not to Block

Four kinds of host need care before anyone blocks them.

1. **Dynamic telecom addresses (5 endpoints).** These sit on Deutsche Telekom (Germany), Chinanet (China), Chunghwa Telecom (Taiwan) and a small US telecom network. Leases rotate, so today's relay may be a household or a hospital within days. The Chunghwa host also looks like a consumer router that someone repurposed, not adversary-owned cloud space.
2. **Shared web-and-mail hosts.** The LazarusGroup-labeled HostPapa host, a Gh0stRAT-labeled host in Hong Kong and a BianLian-labeled host in the United States expose mail ports. A shared host serves many benign domains, so these go on a watchlist, not a block list.
3. **Uncorroborated candidates (8 endpoints).** Four Gh0stRAT-labeled and four BianLian-labeled endpoints are not corroborated. One is a server management controller. Hardware management interfaces and reverse proxies can match banner rules with no malware present.
4. **Dual-use software.** SoftEther VPN, Azure test machines and rclone all have legitimate IT uses. That is why all six RedJuliett-labeled hosts stay on a watchlist, and why the rules allow named jump hosts and backup servers.

Handling group	Endpoints	Hosts	Published?	Action
Confirmed C2	9	8	Yes	Block at firewall, web gateway and EDR; re-check after 720 hours, since cloud addresses get reassigned
Suspected	57	32	No	SIEM or NDR watchlist for 168 hours; pair with a Suricata, JARM or EDR signal
Candidate	8	6	No	Low-priority hunting for 72 hours; needs process or TLS support
Dynamic ISP	5	4	No	Watchlist for 72 hours; never a permanent block

The host counts add up to all 50 hosts (8 + 32 + 6 + 4). No host appears in two groups.

CHAPTER 08

What Defenders Should Do

IN SHORT

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

08 What Defenders Should Do

Day 1: contain and get visibility

- [ ] Block the 9 confirmed endpoints on 8 addresses (Chapter 10) at the firewall, web gateway and EDR. Re-check certificate or JARM state before keeping a block past 720 hours.
- [ ] Put dynamic telecom addresses and shared HostPapa mail hosts on SIEM watchlists for 72 to 168 hours, not on block lists.
- [ ] Run the Sentinel query over 14 days. Confirm there are no unknown SoftEther bridges, writable-folder DLL side-loads or rclone transfers.

Days 2 to 14: harden

- [ ] Use WDAC or AppLocker to stop unsigned DLLs from loading out of user-writable folders (`\Public\Downloads\`, `\ProgramData\`, `%TEMP%`, `%APPDATA%`).
- [ ] Allow SoftEther binaries (`vpnclient_x64.exe`, `vpnbridge_x64.exe`, `vpncmd.exe`) and the `SEVPNCLIENT` service only on named network jump hosts.
- [ ] Allow `rclone.exe`, matched by `OriginalFileName` as well as file name, only on named backup servers. Alert on `--transfers`, `--tpslimit`, `--bwlimit` or `--no-check-certificate`.
- [ ] Deploy the Suricata rules at egress and on TLS-inspection points.
- [ ] Scan quarantine stores, mail gateways and triage shares with the four YARA rules.
- [ ] Move RDP (3389), router and server management interfaces behind an admin-only network, so outsiders cannot turn them into relays.

09

CHAPTER 09

Detection Rules

IN SHORT

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

INSIDE THIS CHAPTER

How the Suricata signatures work 25

How the other rules work 26

09 Detection Rules

The full rules are in Appendix B. Every rule file loads without errors in its engine (Suricata 7.0.12, YARA-X, pySigma, Kusto.Language); the rules were not replayed against live traffic. Test every rule in your own environment before you automate containment.

Rule file	What it catches
`odin_shadowpad_plugx_kimsuky_bianlian.rules`	Suricata 7: eight signatures for SoftEther and staging certificates, the SoftEther tunnel, PlugX, Gh0stRAT and Kimsuky-style traffic, BianLian high-port TLS and rclone uploads
`sentinel_dll_sideload_and_bianlian_go_exfil_hunt.kql`	Microsoft Sentinel and Defender XDR: contact with confirmed C2, side-loading followed by egress, SoftEther tools, rclone theft and high-port egress
`sigma_shadowpad_plugx_sideload_and_extortion_exfil.yml`	Sigma (3 rules): unsigned DLL side-loading in writable folders, SoftEther tools, and rclone bulk transfers
`odin_shadowpad_kimsuky_and_bianlian_implants.yar`	YARA (4 rules): ShadowPad loader, PlugX side-load files, Kimsuky recon scripts and BianLian Go backdoors

How the Suricata signatures work

SID	Looks for	Why
9521001	Server certificate `CN=softether` on port 443	SoftEther bridge default (T1090.002)
9521002	Server certificate `CN=testvm2` on port 3389	Templated staging machine (T1583.003); RDP wraps TLS, so check that your Suricata build parses it
9521003	Outbound `POST` to `/vpnsvc/connect.cgi` or `/vpnsvc/vpn.cgi` with `VPNCONNECT` in the body	SoftEther tunnel setup (T1133); needs TLS inspection
9521004	Outbound `POST` with `X-Session:` and `X-Size:` headers to ports 80, 443, 1033, 8000 or 8254	PlugX HTTP beacon (S0013)
9521005	Outbound TCP payload starting with `Gh0st` followed by the zlib marker `78 9c`	Classic Gh0stRAT framing (S0032)
9521006	Outbound `POST` to `index`, `upload`, `view`, `check`, `down` or `log` `.php` with a body field such as `systeminfo=` or `whoami=`	Kimsuky-style host survey upload (G0094)
9521007	Any server certificate on ports 666, 888, 1122, 5566, 7777 or 8899	BianLian-style high-port TLS (T1573.002); a broad hunt, not a block signal
9521008	HTTP user agent starting with `rclone/v`	Unapproved rclone transfer (T1567.002); needs TLS inspection

Add explicit `pass` rules for any approved SoftEther jump host (SIDs 9521001 and 9521003). Exempt SID 9521008 only for documented backup source addresses.

How the other rules work

The KQL query has five branches and a 14-day lookback. The first flags devices that connect to the 8 confirmed addresses. The second joins an unsigned DLL load in a writable folder with a public connection from the same process within 15 minutes. The third flags SoftEther binaries or command lines. The fourth flags rclone with copy, sync or move plus bulk-transfer flags. The fifth flags public connections to the six BianLian-style ports from unsigned or writable-folder programs.

The query keeps its exclusions to exact host names (`ApprovedNetOpsJumpHosts`, `ApprovedBackupServers`). Fill them with the names exactly as Defender records them, which is often a full domain name. A watchlist variable is left empty for your own CTI feed.

The first Sigma rule fires when both the loading program and an unsigned DLL sit in a writable folder. It has no built-in folder exclusions; tune it in your SIEM by exact file hash or exact path. The second Sigma rule flags SoftEther binaries or command lines (`SEVPNCLIENT`, `AccountConnect`, `BridgeCreate`) outside named jump hosts. The third flags rclone, including renamed copies, with a transfer verb and an exfiltration flag or remote (`mega:`, `sftp:`) outside named backup servers.

Each YARA rule has a file-size cap and needs several strings together.

YARA rule	Size cap	Fires when
`ODIN_ShadowPad_Modular_VFS_Loader_And_Shellcode`	5 MB	A VFS or POISONPLUG marker, a plugin path marker, and the `RtlDecompressBuffer` and `VirtualAllocEx` imports
`ODIN_PlugX_Korplug_Sideload_Config_And_HTTP_Beacon`	4 MB	A config marker or two custom HTTP headers, plus `RtlDecompressBuffer` and `RunDllEntry`
`ODIN_Kimsuky_Recon_Stager_And_PHP_Collection_Script`	2 MB	An HTTP object, two survey commands (`systeminfo`, `ipconfig /all`, `tasklist`) and a PHP upload path
`ODIN_BianLian_Golang_mTLS_Backdoor_Implant`	15 MB	A Go build ID, TLS handshake and certificate-pool calls, `net.DialTimeout` and the `hashicorp/yamux` library

10

CHAPTER 10

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Confirmed C2 endpoints to block 28

JARM fingerprints 29

MITRE ATT&CK mapping 29

10 Indicators

Appendix A holds 12 published indicators: 9 address entries (IP and port) on 8 IPv4 addresses, 0 file hashes and 3 JARM fingerprints. Three more JARM values, tied to suspected clusters, appear below as text-only hunting leads. The 70 suspected, candidate and dynamic-ISP endpoints are withheld, along with every certificate domain.

Type	In Appendix A	Text only	Withheld
Confirmed C2 endpoints (IP and port)	9	0	0
Suspected, candidate and dynamic-ISP endpoints	0	0	70
JARM fingerprints	3	3	0
File hashes (SHA-256)	0	0	0

Confirmed C2 endpoints to block

Endpoint	Label	Hosting	Country	Key fingerprint	Seen
3.140.83[.]52:443	Kimsuky	Amazon.com, Inc.	US	JARM `29d29d000...5df96`, shared with the AU host	17 Nov 2026
3.104.232[.]81:443	Kimsuky	Amazon.com, Inc.	AU	Same JARM; HTTP on 80 on the same host	17 Nov 2026
3.104.232[.]81:80	Kimsuky	Amazon.com, Inc.	AU	HTTP listener paired with 443	17 Nov 2026
172.233.130[.]140:999	Kimsuky	Akamai Technologies, Inc.	US	Certificate and JARM `29d29d20d...371a3` shared with the next host	13 Nov 2026
172.236.231[.]228:999	Kimsuky	Akamai Technologies, Inc.	US	Same certificate and JARM	13 Nov 2026
118.194.248[.]246:80	Kimsuky	UCLOUD INFORMATION TECHNOLOGY (HK) LIMITED	KR	Plain HTTP gate, adjacent `/24` pair	13 Nov 2026
118.194.249[.]154:80	Kimsuky	UCLOUD INFORMATION TECHNOLOGY (HK) LIMITED	KR	Plain HTTP gate, adjacent `/24` pair	13 Nov 2026
130.94.65[.]79:443	ShadowPad	LIGHT NODE LIMITED	JP	HTTPS C2, JARM `21d14d...6cbe21`	13 Nov 2026
45.32.237[.]137:3389	ShadowPad	The Constant Company, LLC	NL	`CN=WIN-JRJ4OQAGQ2I`, non-RDP listener on 3389	13 Nov 2026

The Label column gives the threat-feed label, not an attribution.

JARM fingerprints

Never block on a JARM value alone. Use these in attack surface sweeps and TLS-aware network hunts, together with the certificate name and port.

JARM	Port	Label	In Appendix A?	Where seen
`29d29d00029d29d00041d41d00041dcb3f8752a2d70d1dbf446fc46ab5df96`	443	Kimsuky	Yes	Confirmed AWS pair (US, AU)
`29d29d20d29d29d21c29d29d29d29d89e97ba53da7e1515fb732a85cb371a3`	999	Kimsuky	Yes	Confirmed Akamai pair (US)
`21d14d00021d21d21c42d43d0000007abc6200da92c2a1b69c0a56366cbe21`	443	ShadowPad	Yes	Confirmed host 130.94.65[.]79
`28d28d28d00028d1ec42d42d0000005557fe5c0eece598995bd43427a3386b`	443	BianLian	No, text only	12 sockets on the twin HostPapa hubs (suspected)
`22d22d00022d22d22c42d42d000000faabb8fd156aa8b4d8a37853e1063261`	443	BianLian	No, text only	Go TLS trio on Azure, AWS and a telecom network (suspected)
`06d06d07d06d06d06c42d42d000000644097aad2d54615f61ac543d994271f`	443	RedJuliett	No, text only	SoftEther bridge on Alibaba Hong Kong (suspected)

Withheld: the addresses of all 70 non-confirmed endpoints, every certificate domain, and the certificate names of router, management-controller and dynamic-ISP hosts.

MITRE ATT&CK mapping

Tactic	Technique or software	How it shows up here	Detection
Resource Development	T1583.003 Acquire Infrastructure: Virtual Private Server	Templated Azure staging machines (`CN=testvm2`); leased VPS relays	SID 9521002
Command and Control	T1090.002 Proxy: External Proxy	SoftEther bridges on 443 across three providers	SID 9521001
Persistence, Initial Access	T1133 External Remote Services	SoftEther tunnel sessions to `/vpnsvc/connect.cgi`	SID 9521003; Sigma rule 2; KQL branch 3
Defense Evasion, Persistence	T1574.002 Hijack Execution Flow: DLL Side-Loading	ShadowPad (S0596) and PlugX (S0013) triads in writable folders	Sigma rule 1; KQL branch 2; YARA
Command and Control	T1573.002 Encrypted Channel: Asymmetric Cryptography	BianLian-style Go TLS on six ports	SID 9521007; KQL branch 5; YARA
Exfiltration	T1567.002 Exfiltration to Cloud Storage	rclone bulk transfers	SID 9521008; Sigma rule 3; KQL branch 4
Command and Control	S0032 gh0st RAT	Binary zlib TCP framing	SID 9521005

11

CHAPTER 11

Scope and Limits

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

11 Scope and Limits

- **Window.** The findings cover a two-week observation window ending 18 November 2026. Dates in the tables are the days we saw each endpoint.
- **Labels are not attribution.** Cluster names come from threat feeds and fingerprint matches. Public advisories explain tradecraft; they are not used to tie any address here to a named actor.
- **External view only.** The data shows listening servers, certificates and ports. Side-loading, tunnelling and rclone behaviour come from public advisories, not from these hosts. We did not see any activity on target systems.
- **Groups are a model.** The three groups in Chapter 2 sort clusters by role. No address, certificate or JARM value links clusters in different groups. The one JARM value shared between Coruna and BianLian hosts reflects a common TLS stack.
- **Corroborated, not owned.** Cloud providers reassign addresses after abuse reports. Re-check a block after 720 hours.
- **Rule testing.** Every published rule file loads without errors in its engine (Suricata 7.0.12, YARA-X, pySigma and Microsoft's Kusto.Language analyzer, checked on 20 November 2026). The rules were not replayed against live traffic, so test and tune them in your own environment before production use.

12

CHAPTER 12

Sources

IN SHORT

This chapter lists the public references used in the report.

12 Sources

1. CISA, FBI and ASD-ACSC. Joint Cybersecurity Advisory AA23-136A, #StopRansomware: BianLian Data Extortion Group. https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-136a (verified 19 November 2026)
2. CISA, FBI and NSA. Joint Cybersecurity Advisory AA26-281A, PRC state-sponsored operational relay box and SoftEther VPN bridge tradecraft. https://www.cisa.gov/news-events/cybersecurity-advisories/aa26-281a (verified 19 November 2026)
3. MITRE ATT&CK. ShadowPad, S0596. https://attack.mitre.org/software/S0596/ (verified 19 November 2026)
4. MITRE ATT&CK. PlugX, S0013, and T1574.002 DLL Side-Loading. https://attack.mitre.org/software/S0013/ (verified 19 November 2026)
5. MITRE ATT&CK. gh0st RAT, S0032. https://attack.mitre.org/software/S0032/ (verified 19 November 2026)
6. MITRE ATT&CK. Kimsuky, G0094. https://attack.mitre.org/groups/G0094/ (verified 19 November 2026)
7. MITRE ATT&CK. Lazarus Group, G0032. https://attack.mitre.org/groups/G0032/ (verified 19 November 2026)
8. Salesforce Engineering. Easily Identify Malicious Servers on the Internet with JARM. https://engineering.salesforce.com/easily-identify-malicious-servers-on-the-internet-with-jarm-e095edac525a/ (verified 19 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 (12) 35

B. Detection rules (4 files) 36

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
R21-IOC-01	IPv4	3.140.83[.]52	443	kimsuky	BLOCK
R21-IOC-02	IPv4	3.104.232[.]81	443	kimsuky	BLOCK
R21-IOC-03	IPv4	3.104.232[.]81	80	kimsuky	BLOCK
R21-IOC-04	IPv4	172.233.130[.]140	999	kimsuky	BLOCK
R21-IOC-05	IPv4	172.236.231[.]228	999	kimsuky	BLOCK
R21-IOC-06	IPv4	118.194.248[.]246	80	kimsuky	BLOCK
R21-IOC-07	IPv4	118.194.249[.]154	80	kimsuky	BLOCK
R21-IOC-08	IPv4	130.94.65[.]79	443	shadowpad	BLOCK
R21-IOC-09	IPv4	45.32.237[.]137	3389	shadowpad	BLOCK
R21-IOC-10	JARM	29d29d00029d29d00041d41d00041dcb3f8752a2d70d1dbf446fc46ab5df96	443	kimsuky	HUNT
R21-IOC-11	JARM	29d29d20d29d29d21c29d29d29d29d89e97ba53da7e1515fb732a85cb371a3	999	kimsuky	HUNT
R21-IOC-12	JARM	21d14d00021d21d21c42d43d0000007abc6200da92c2a1b69c0a56366cbe21	443	shadowpad	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_shadowpad_kimsuky_and_bianlian_implants.yar

YARA

```
/*
   Odin by Brandefense - Report #21 YARA Implant & Staging Detection Suite
   Classification: TLP:CLEAR
   Scope: ShadowPad (S0596) VFS modular loader, PlugX (S0013) side-load HTTP beacon,
          Kimsuky (G0094) VBS/PS1 PHP recon stager, and BianLian Go mTLS backdoor.
*/

rule ODIN_ShadowPad_Modular_VFS_Loader_And_Shellcode
{
    meta:
        author = "Odin by Brandefense"
        date = "2026-11-19"
        tlp = "CLEAR"
        mitre_software = "S0596"
        description = "Detects ShadowPad (POISONPLUG.SHADOW) modular Virtual File System plugin table and
decompression loader markers"
    strings:
        $vfs1 = "VFS_ROOT_TABLE" ascii wide
        $vfs2 = "POISONPLUG" ascii wide nocase
        $plg1 = "Plugins\\OnlineKeylogger" ascii wide nocase
        $plg2 = "Plugins\\TcpTunnel" ascii wide nocase
        $api1 = "RtlDecompressBuffer" ascii
        $api2 = "VirtualAllocEx" ascii
    condition:
        filesize < 5MB and (1 of ($vfs*)) and (1 of ($plg*)) and all of ($api*)
}

rule ODIN_PlugX_Korplug_Sideload_Config_And_HTTP_Beacon
{
    meta:
        author = "Odin by Brandefense"
        date = "2026-11-19"
        tlp = "CLEAR"
        mitre_software = "S0013"
        description = "Detects PlugX (Korplug) side-loaded DLL configuration markers and custom X-Session/X-Size
HTTP tunnel headers"
    strings:
        $cfg1 = "PLUG_CFG_HDR" ascii
        $hdr1 = "X-Session:" ascii wide
        $hdr2 = "X-Size:" ascii wide
        $hdr3 = "X-Status:" ascii wide
        $ldr1 = "RtlDecompressBuffer" ascii
        $ldr2 = "RunDllEntry" ascii
    condition:
        filesize < 4MB and ($cfg1 or 2 of ($hdr*)) and all of ($ldr*)
}

rule ODIN_Kimsuky_Recon_Stager_And_PHP_Collection_Script
{
    meta:
        author = "Odin by Brandefense"
        date = "2026-11-19"
        tlp = "CLEAR"
        mitre_group = "G0094"
        description = "Detects Kimsuky VBS/script reconnaissance stagers harvesting host survey output to PHP
collection gates"
    strings:
        $http1 = "MSXML2.ServerXMLHTTP" ascii wide nocase
        $http2 = "WinHttp.WinHttpRequest" ascii wide nocase
        $cmd1 = "systeminfo" ascii wide nocase
```

```
        $cmd2 = "ipconfig /all" ascii wide nocase
        $cmd3 = "tasklist" ascii wide nocase
        $php1 = ".php?sysinfo=" ascii wide nocase
        $php2 = "/upload.php" ascii wide nocase
    condition:
        filesize < 2MB and (1 of ($http*)) and (2 of ($cmd*)) and (1 of ($php*))
}

rule ODIN_BianLian_Golang_mTLS_Backdoor_Implant
{
    meta:
        author = "Odin by Brandefense"
        date = "2026-11-19"
        tlp = "CLEAR"
        reference = "CISA-AA23-136A"
        description = "Detects Go-compiled backdoor binaries combining pinned X.509 certificate pools, TLS
handshakes, and yamux multiplexing"
    strings:
        $go1 = "Go build ID:" ascii
        $tls1 = "crypto/tls.(*Conn).Handshake" ascii
        $tls2 = "crypto/x509.(*CertPool).AppendCertsFromPEM" ascii
        $net1 = "net.DialTimeout" ascii
        $mux1 = "hashicorp/yamux" ascii
    condition:
        filesize < 15MB and $go1 and all of ($tls*) and $net1 and $mux1
}
```

odin_shadowpad_plugx_kimsuky_bianlian.rules

SURICATA

```
# Odin by Brandefense - Report #21 Suricata 7.x APT Relay Meshes & BianLian Extortion Suite
# Classification: TLP:CLEAR
# Scope Note: Covers RedJuliett SoftEther VPN operational relay certificates and HTTPS tunnels,
#             Azure staging VM certificates, PlugX HTTP beacons, Gh0stRAT binary zlib streams,
#             Kimsuky PHP collection gates, BianLian multi-port Go TLS listeners, and rclone exfil.

# 1. RedJuliett Operational Relay Box (ORB) SoftEther VPN Self-Signed TLS Certificate (CN=softether)
alert tls $EXTERNAL_NET 443 -> $HOME_NET any (msg:"ODIN-APT RedJuliett SoftEther VPN Operational Relay TLS
Certificate CN=softether"; flow:established,from_server; tls.cert_subject; content:"CN=softether"; nocase;
fast_pattern; pcre:"/CN=softether(,|$)/i"; metadata:created_at 2026_11_19, mitre_technique_id T1090_002;
classtype:command-and-control; sid:9521001; rev:1;)

# 2. RedJuliett Multi-Cloud Segmented Staging Node Self-Signed TLS Certificate (CN=testvm2)
alert tls $EXTERNAL_NET 3389 -> $HOME_NET any (msg:"ODIN-APT RedJuliett Staging Node Self-Signed TLS Certificate
CN=testvm2"; flow:established,from_server; tls.cert_subject; content:"CN=testvm2"; fast_pattern; pcre:"/
CN=testvm2(,|$)/"; metadata:created_at 2026_11_19, mitre_technique_id T1583_003; classtype:command-and-control;
sid:9521002; rev:1;)

# 3. SoftEther VPN Client-to-Bridge HTTPS Tunnel Handshake (/vpnsvc/connect.cgi with VPNCONNECT Pack)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-APT SoftEther VPN HTTPS Tunnel Handshake (/vpnsvc/
connect.cgi)"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/vpnsvc/";
fast_pattern; pcre:"/^\/vpnsvc\/(connect|vpn)\.cgi$/"; http.request_body; bsize:>15; content:"VPNCONNECT";
metadata:created_at 2026_11_19, mitre_technique_id T1133; classtype:policy-violation; sid:9521003; rev:1;)

# 4. PlugX (Korplug S0013) HTTP Tunnel Disguise Custom X-Session and X-Size Header Beacon
alert http $HOME_NET any -> $EXTERNAL_NET [80,443,1033,8000,8254] (msg:"ODIN-APT PlugX HTTP Tunnel Custom X-
Session and X-Size Header Beacon"; flow:established,to_server; http.method; content:"POST"; http.header;
content:"X-Session:"; fast_pattern; content:"X-Size:"; http.request_body; bsize:>15; metadata:created_at
2026_11_19, mitre_software_id S0013; classtype:command-and-control; sid:9521004; rev:1;)

# 5. Gh0stRAT (S0032) Classic Binary TCP Framing Header and Zlib 0x789c Stream
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-APT Gh0stRAT TCP Binary Framing Header and Zlib Stream";
flow:established,to_server; dsize:>14; content:"Gh0st"; depth:5; fast_pattern; content:"|78 9c|"; distance:8;
within:2; metadata:created_at 2026_11_19, mitre_software_id S0032; classtype:command-and-control; sid:9521005;
rev:1;)

# 6. Kimsuky (G0094) Spearphishing Recon and Host Survey PHP Collection Gate POST
alert http $HOME_NET any -> $EXTERNAL_NET [80,443,999,4003,4004,9090] (msg:"ODIN-APT Kimsuky PHP Collection Gate
Recon Upload POST"; flow:established,to_server; http.method; content:"POST"; http.uri; content:".php";
```

```
fast_pattern; pcre:"/\/(index|upload|view|check|down|log)\.php(\?|$)/i"; http.request_body; bsize:>24; pcre:"/
(sysinfo|ipconfig|systeminfo|tasklist|whoami)=/i"; metadata:created_at 2026_11_19, mitre_group_id G0094;
classtype:command-and-control; sid:9521006; rev:1;)

# 7. BianLian Multi-Port Go C2 Non-Standard TLS Listener Handshake (:666,:888,:1122,:5566,:7777,:8899)
alert tls $EXTERNAL_NET [666,888,1122,5566,7777,8899] -> $HOME_NET any (msg:"ODIN-EXTORTION BianLian Multi-Port
Go C2 Non-Standard TLS Listener"; flow:established,from_server; tls.cert_subject; content:"CN="; fast_pattern;
metadata:created_at 2026_11_19, mitre_technique_id T1573_002; classtype:command-and-control; sid:9521007; rev:1;)

# 8. BianLian Style Unapproved Rclone Bulk Cloud Storage Exfiltration User-Agent (rclone/v)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-EXFIL Unapproved Rclone Bulk Cloud Storage Exfiltration
User-Agent"; flow:established,to_server; http.user_agent; bsize:>8; content:"rclone/v"; startswith; fast_pattern;
metadata:created_at 2026_11_19, mitre_technique_id T1567_002; classtype:policy-violation; sid:9521008; rev:1;)
```

sentinel_dll_sideload_and_bianlian_go_exfil_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Report #21 Microsoft Sentinel & Defender XDR Hunting Query
// Correlates Egress to Confirmed Kimsuky- and ShadowPad-Labeled C2s, ShadowPad/PlugX Unsigned DLL
// Side-Loading from User-Writable Directories Joined to Outbound Egress, Unauthorized SoftEther
// VPN Operational Relay Tunneling, and BianLian Style Go Multi-Port C2 / Rclone Bulk Exfiltration.
let Lookback = 14d;
let ConfirmedAptC2IPs = dynamic([
    "3.140.83.52", "3.104.232.81", "172.233.130.140", "172.236.231.228",
    "118.194.248.246", "118.194.249.154", "130.94.65.79", "45.32.237.137"
]);
// Populate WatchlistRelayAndBianLianIPs from your own CTI watchlist (addresses are not published in this
edition)
let WatchlistRelayAndBianLianIPs = dynamic([]);
let BianLianClusterPorts = dynamic([666, 888, 1122, 5566, 7777, 8899]);
let WritableStagingPaths = dynamic([@"\Public\Downloads\", @"\ProgramData\", @"\AppData\Local\Temp\",
@"\AppData\Roaming\", @"\Windows\Temp\"]);
// Placeholder host names: replace with your own approved hosts.
let ApprovedNetOpsJumpHosts = dynamic(["SRV-NETOPS-VPN01", "SRV-INFRA-MGMT01"]);
let ApprovedBackupServers = dynamic(["SRV-BACKUP-RCLONE01", "SRV-DR-ARCHIVE01"]);
union isfuzzy=true
(
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback) and ActionType == "ConnectionSuccess"
    | where RemoteIP in (ConfirmedAptC2IPs) or RemoteIP in (WatchlistRelayAndBianLianIPs)
    | extend HuntSignal = iff(RemoteIP in (ConfirmedAptC2IPs),
"Confirmed_Kimsuky_Or_ShadowPad_Labeled_C2_Egress", "Watchlist_APT_Relay_Or_BianLian_Cluster_Egress"),
             ArtifactDetail = strcat(RemoteIP, ":", tostring(RemotePort), " (", InitiatingProcessFolderPath, ")")
    | project Timestamp, DeviceName, HuntSignal, ActorOrProcess = InitiatingProcessFileName, ArtifactDetail
),
(
    DeviceImageLoadEvents
    | where Timestamp >= ago(Lookback)
    | where FolderPath has_any (WritableStagingPaths) and InitiatingProcessFolderPath has_any
(WritableStagingPaths)
    | where FileName endswith ".dll"
    | join kind=inner (
        DeviceFileCertificateInfo
        | where Timestamp >= ago(Lookback) and IsSigned == false
        | project SHA1
    ) on SHA1
    | where isnotempty(InitiatingProcessVersionInfoCompanyName)
    | join kind=inner (
        DeviceNetworkEvents
        | where Timestamp >= ago(Lookback) and ActionType == "ConnectionSuccess"
        | where RemoteIPType == "Public"
        | project NetTime = Timestamp, DeviceId, InitiatingProcessId, RemoteIP, RemotePort
    ) on DeviceId, InitiatingProcessId
    | where abs(datetime_diff("minute", Timestamp, NetTime)) <= 15
    | extend HuntSignal = "ShadowPad_Or_PlugX_DLL_SideLoad_With_Public_Egress",
             ArtifactDetail = strcat(InitiatingProcessFileName, " loaded unsigned ", FileName, " -> ", RemoteIP,
":", tostring(RemotePort))
    | project Timestamp, DeviceName, HuntSignal, ActorOrProcess = InitiatingProcessFileName, ArtifactDetail
),
```

```
(
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedNetOpsJumpHosts))
    | where FileName has_any ("vpnclient", "vpnbridge", "vpncmd")
        or ProcessCommandLine has_any ("/vpnsvc/connect.cgi", "SEVPNCLIENT", "AccountConnect", "BridgeCreate")
    | extend HuntSignal = "Unauthorized_SoftEther_VPN_Client_Or_ORB_Bridge",
             ArtifactDetail = strcat(FolderPath, " | ", ProcessCommandLine)
    | project Timestamp, DeviceName, HuntSignal, ActorOrProcess = FileName, ArtifactDetail
),
(
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedBackupServers))
    | where (FileName =~ "rclone.exe" or ProcessVersionInfoOriginalFileName =~ "rclone.exe")
        and ProcessCommandLine has_any ("copy", "sync", "move")
        and ProcessCommandLine has_any ("--transfers", "--tpslimit", "--bwlimit", "--config", "mega:", "sftp:",
"--no-check-certificate")
    | extend HuntSignal = "BianLian_Style_Unauthorized_Rclone_Bulk_Exfiltration",
             ArtifactDetail = strcat(FolderPath, " | ", ProcessCommandLine)
    | project Timestamp, DeviceName, HuntSignal, ActorOrProcess = FileName, ArtifactDetail
),
(
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback) and ActionType == "ConnectionSuccess"
    | where not(DeviceName in~ (ApprovedBackupServers))
    | where RemoteIPType == "Public" and RemotePort in (BianLianClusterPorts)
    | where InitiatingProcessFolderPath has_any (WritableStagingPaths) or
isempty(InitiatingProcessVersionInfoCompanyName)
    | extend HuntSignal = "Suspected_BianLian_Go_Implant_MultiPort_Egress",
             ArtifactDetail = strcat(InitiatingProcessFolderPath, " -> ", RemoteIP, ":", tostring(RemotePort))
    | project Timestamp, DeviceName, HuntSignal, ActorOrProcess = InitiatingProcessFileName, ArtifactDetail
)
| summarize
    FirstSeen = min(Timestamp),
    LastSeen = max(Timestamp),
    EventCount = count(),
    SampleDetails = make_set(ArtifactDetail, 5)
    by DeviceName, HuntSignal, ActorOrProcess
| order by EventCount desc
```

sigma_shadowpad_plugx_sideload_and_extortion_exfil.yml

SIGMA

```
title: ShadowPad and PlugX Unsigned DLL Side-Loading from User-Writable Staging Directory
id: 7d21a8e4-3c91-4f80-9a12-55e109210001
status: experimental
description: Detects a signed executable running from a user-writable staging directory loading an unsigned side-
loaded DLL from a user-writable path, characteristic of ShadowPad (S0596) and PlugX (S0013) execution chains.
author: Odin by Brandefense
date: 2026-11-19
references:
  - https://attack.mitre.org/software/S0596/
  - https://attack.mitre.org/software/S0013/
  - https://attack.mitre.org/techniques/T1574/002/
logsource:
  category: image_load
  product: windows
detection:
  selection_host_writable:
    Image|contains:
      - '\Public\Downloads\'
      - '\ProgramData\'
      - '\AppData\Local\Temp\'
      - '\AppData\Roaming\'
      - '\Windows\Temp\'
  selection_unsigned_module:
    ImageLoaded|contains:
      - '\Public\Downloads\'
```

```
      - '\ProgramData\'
      - '\AppData\Local\Temp\'
      - '\AppData\Roaming\'
      - '\Windows\Temp\'
    ImageLoaded|endswith: '.dll'
    Signed: 'false'
  condition: selection_host_writable and selection_unsigned_module
falsepositives:
  - Unpacked legacy line-of-business vendor installers running out of C:\ProgramData; tune in the SIEM by exact
signed binary SHA-256 or exact executable path.
level: high
tags:
  - attack.persistence
  - attack.defense_evasion
  - attack.t1574.002
---
title: Unauthorized SoftEther VPN Client or Operational Relay Bridge Execution
id: 7d21a8e4-3c91-4f80-9a12-55e109210002
status: experimental
description: Detects execution or command-line provisioning of SoftEther VPN client and bridge utilities abused
by PRC state-nexus clusters such as RedJuliett for persistent operational relay tunneling.
author: Odin by Brandefense
date: 2026-11-19
references:
  - https://www.cisa.gov/news-events/cybersecurity-advisories/aa26-281a
  - https://attack.mitre.org/techniques/T1133/
logsource:
  category: process_creation
  product: windows
detection:
  selection_binary:
    Image|endswith:
      - '\vpnclient.exe'
      - '\vpnclient_x64.exe'
      - '\vpnbridge.exe'
      - '\vpnbridge_x64.exe'
      - '\vpncmd.exe'
  selection_cli:
    CommandLine|contains:
      - 'SEVPNCLIENT'
      - 'AccountConnect'
      - 'BridgeCreate'
      - '/vpnsvc/connect.cgi'
  filter_approved_jump_host:
    # Placeholder host names: replace with your own approved hosts.
    ComputerName:
      - 'SRV-NETOPS-VPN01'
      - 'SRV-INFRA-MGMT01'
  condition: (selection_binary or selection_cli) and not filter_approved_jump_host
falsepositives:
  - Sanctioned network engineering jump hosts explicitly authorized to administer SoftEther bridges.
level: high
tags:
  - attack.persistence
  - attack.command_and_control
  - attack.t1133
  - attack.t1090.002
---
title: BianLian Style Encryptionless Bulk Exfiltration via Rclone CLI
id: 7d21a8e4-3c91-4f80-9a12-55e109210003
status: experimental
description: Detects unauthorized execution of Rclone (including renamed binaries via OriginalFileName)
transferring data with high-concurrency or custom remote flags typical of BianLian pure-exfiltration extortion.
author: Odin by Brandefense
date: 2026-11-19
references:
  - https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-136a
  - https://attack.mitre.org/techniques/T1567/002/
logsource:
  category: process_creation
  product: windows
detection:
```

```
  selection_img:
    - Image|endswith: '\rclone.exe'
    - OriginalFileName: 'rclone.exe'
  selection_verb:
    CommandLine|contains:
      - ' copy '
      - ' sync '
      - ' move '
  selection_exfil_flags:
    CommandLine|contains:
      - '--transfers'
      - '--tpslimit'
      - '--bwlimit'
      - '--config'
      - '--no-check-certificate'
      - 'mega:'
      - 'sftp:'
  filter_approved_backup:
    # Placeholder host names: replace with your own approved hosts.
    ComputerName:
      - 'SRV-BACKUP-RCLONE01'
      - 'SRV-DR-ARCHIVE01'
  condition: selection_img and selection_verb and selection_exfil_flags and not filter_approved_backup
falsepositives:
  - Sanctioned cloud backup or disaster-recovery transfer servers; restrict allowlists strictly to named backup
hostnames.
level: high
tags:
  - attack.exfiltration
  - attack.t1567.002
```

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/relay-meshes-and-six-port-extortion-hubs/
Short Read · brandefense.io/odin-research/blog/apt-relay-meshes-and-six-port-extortion-hubs/
Defense Kit · brandefense.io/odin-research/defense-kit/