ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 43 / 30 · COMMODITY MALWARE

Commodity RAT Fleets
Nine Cloud Candidates Amid Broadband Collisions

How 4,042 examined RAT endpoints across 2,771 hosts split into nine feed-labelled cloud and VPS relay candidates, 421 broadband ISP addresses, and 63 service collisions

Two-week observation window ending 20 November 2026 · Prepared November 2026
Full Report · Public edition · Version 1.1

19,519
Global feed records labelled as AsyncRAT, QuasarRAT, Remcos, DcRAT, VenomRAT, or RaptorRAT

9
Feed-labelled dedicated cloud and VPS relay candidates retained for analyst review (1 AsyncRAT, 4 QuasarRAT, 2 DcRAT, 2 VenomRAT)

421 / 250
Residential and mobile telecom ISP endpoints withheld across 250 dynamic subscriber hosts

414
Examined endpoints presenting default Quasar, DcRat, AsyncRAT, or VenomRAT certificate names

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

How 4,042 examined RAT endpoints across 2,771 hosts split into nine feed-labelled cloud and VPS relay candidates, 421 broadband ISP addresses, and 63 service collisions

01	Executive Summary	04
02	Six RAT Families, One Analytical Model	07
03	Core Open-Source .NET Fleets: AsyncRAT and QuasarRAT	10
04	AsyncRAT Forks, Native Remcos Listeners, and RaptorRAT	13
05	Default Builder Certificates, Broadband ISP Churn, and Collisions	16
06	Feed Labels Against Corroborated C2 Evidence	20
07	What Defenders Should Do	23
08	Detection Rules	25
09	Indicators	28
10	Scope and Limits	31
11	Sources	33
A	Indicators of compromise (10)	35
B	Detection rules (4 files)	36

IN SHORT

Odin by Brandefense examined 4,042 endpoints across 2,771 hosts during a two-week window ending 20 November 2026. The sample came from 19,519 feed records labelled as AsyncRAT, QuasarRAT, Remcos, DcRAT, VenomRAT, and RaptorRAT. Nine dedicated cloud and VPS endpoints carried upstream confirmed labels without independent C2 behaviour, while 421 endpoints sat on residential and mobile telecom networks. Retain the nine cloud and VPS candidates for analyst review, hunt default RAT certificates on the wire, and keep subscriber ISP leases out of static firewall feeds.

01

CHAPTER 01

Executive Summary

IN SHORT

Odin by Brandefense examined 4,042 endpoints across 2,771 hosts during a two-week window ending 20 November 2026. The sample came from 19,519 feed records labelled as AsyncRAT, QuasarRAT, Remcos, DcRAT, VenomRAT, and RaptorRAT. Nine dedicated cloud and VPS endpoints carried upstream confirmed labels without independent C2 behaviour, while 421 endpoints sat on residential and mobile telecom networks. Retain the nine cloud and VPS candidates for analyst review, hunt default RAT certificates on the wire, and keep subscriber ISP leases out of static firewall feeds.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Public C# `.NET` builders and commercial Windows remote administration kits ship with built-in self-signed TLS certificates and fixed listener ports. Threat feeds flag thousands of addresses whenever external scans match those default certificates, Windows TLS fingerprints, or common ports.

Blocking every feed match cuts off home broadband customers after DHCP lease rotation and breaks servers behind shared cloud certificates. Because none of the nine cloud and VPS relay candidates shares an IPv4 address or custom certificate across two RAT families, we present the six families as an analytical model.

1. **Nine dedicated cloud and VPS candidates require analyst review across 4,042 examined endpoints.**
   Out of 19,519 global feed records across six Windows RAT families, nine dedicated endpoints carried upstream confirmed labels: one AsyncRAT, four QuasarRAT, two DcRAT, and two VenomRAT candidates. Because the evidence records certificates, JARMs, ports, and hosting without independent C2 behaviour, all 4,042 endpoint addresses are withheld.

2. **Subscriber telecom networks host 421 RAT-labelled endpoints on 250 dynamic hosts.**
   Operators running home or dynamic-DNS listeners behind consumer routers account for 148 VenomRAT, 99 DcRAT, 89 QuasarRAT, 55 AsyncRAT, and 30 Remcos endpoints. Eighty-nine of those telecom endpoints present default builder certificates such as `CN=Quasar Server CA` or `CN=AsyncRAT Server`, including one feed-labelled residential cable lease whose connection details are withheld.

3. **Shared TLS certificates, service ports, and open directories create 63 collision cases.**
   Forty-nine endpoints sit behind shared Cloudflare origin, proxy, or appliance certificates, ten use standard SSH, DNS, mail, or database ports, and four expose open HTTP folders. Meanwhile, commodity `.NET` RAT configurations in the sample also use TCP port 3389 (including 33 DcRAT endpoints) to blend into Remote Desktop traffic.

Three Decisions for Leaders

1. **Do not block commodity RAT feed addresses without independent C2 corroboration.** Retain the nine dedicated cloud and VPS candidates for analyst review with a 7-day cloud and 30-day VPS window. Keep all 4,042 uncorroborated endpoints out of static firewall block lists.
2. **Replace static ISP blocks with TLS certificate and port-3389 hunting.** Alert on default builder certificate names (`CN=AsyncRAT Server`, `CN=Quasar Server CA`, `CN=DcRat`, and `CN=VenomRAT`) and non-RDP Windows processes opening external TCP 3389.
3. **Never block on a Windows TLS fingerprint or default RAT port alone.** Use cyber threat intelligence to assess JARM fingerprints and default builder ports (2404, 4449, 4782, 6606, 8848) as hunting leads paired with endpoint process logs.

The Study in Numbers

Number	What it is
19,519	Global feed records labelled as AsyncRAT, QuasarRAT, Remcos, DcRAT, VenomRAT, or RaptorRAT
4,042 / 2,771	Endpoints examined across 2,771 distinct hosts (all 4,042 endpoint addresses withheld)
9	Feed-labelled dedicated cloud and VPS relay candidates retained for analyst review (1 AsyncRAT, 4 QuasarRAT, 2 DcRAT, 2 VenomRAT)
421 / 250	Residential and mobile telecom ISP endpoints withheld across 250 dynamic subscriber hosts
414	Examined endpoints presenting default Quasar, DcRat, AsyncRAT, or VenomRAT certificate names
63	Collision cases withheld across 49 shared certificates, 10 standard service ports, and 4 open directories
10	Indicators published in Appendix A (10 TLS JARM fingerprints for hunting; 0 IPv4 addresses)

How to use this report.

Security leaders can start with the Short Read on commodity RAT infrastructure, then Chapters 1, 6, and 7. Network and Windows platform engineers should read Chapters 3, 4, 5, and 7. SOC analysts and detection engineers should go directly to Chapters 8 and 9. Threat intelligence analysts will find the hosting breakdown and scope limits in Chapters 2, 5, 6, and 10.

02

CHAPTER 02

Six RAT Families, One Analytical Model

IN SHORT

None of the nine feed-labelled cloud and VPS relay candidates shares an IPv4 address or custom certificate across two RAT families. Retain those candidates for analyst review, and use our six-family analytical model to focus controls on shared Windows RAT builder habits.

INSIDE THIS CHAPTER

Six families from 19,519 feed records to 4,042 examined endpoints 07

Where the infrastructure sits 08

02 Six RAT Families, One Analytical Model

Open-source C# `.NET` remote administration trojans (AsyncRAT, QuasarRAT, and the AsyncRAT forks DcRAT and VenomRAT) sit alongside native Windows remote control tools (Remcos) and smaller clusters (RaptorRAT). Their builder programs circulate widely on public repositories and criminal forums. Hundreds of unrelated operators start listeners without changing the built-in TLS certificates or default TCP ports.

We compared addresses and X.509 certificates across all nine dedicated cloud and VPS relay candidates. Zero candidate IPv4 addresses and zero custom TLS certificates span two different RAT families.

Within DcRAT alone, the two VPS review candidates (connection details withheld) share one server certificate and one JARM fingerprint. Across the wider sample, 77 hosts carry labels from two (73 hosts) or three (4 hosts) families because operators reuse `.NET` certificates or forward several ports on one router. Figure 1 shows the four functional groups in our analytical model and what defenders can do with each.

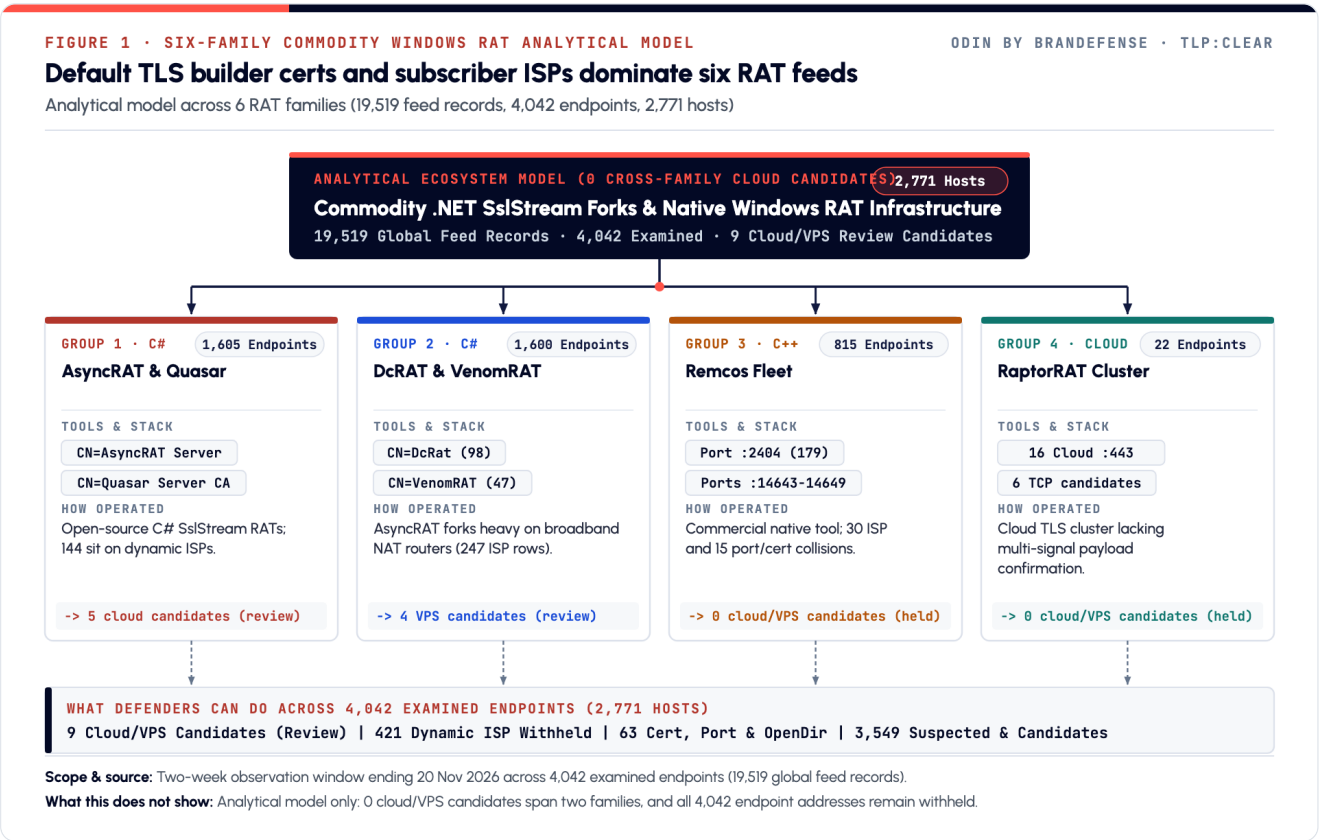


Figure 1: Default TLS builder certs and subscriber ISPs dominate six RAT feeds

Six families from 19,519 feed records to 4,042 examined endpoints

Feed records across the six labels total 19,519 entries. They span 81 countries for AsyncRAT, 77 for QuasarRAT, 54 for DcRAT, 53 for Remcos, 50 for VenomRAT, and 6 for RaptorRAT. Older dates attached to some feed records are the upstream listing dates recorded on those entries, not continuous observation by Odin by Brandefense.

Our examined set covers 4,042 endpoints across 2,771 distinct hosts. Nine dedicated cloud and VPS endpoints (0.22%) carried upstream confirmed labels without independent C2 behaviour. Their addresses and connection details are withheld for analyst review alongside all 4,033 other endpoints (99.78%) on 2,762 hosts.

RAT family	Feed records	Examined	Hosts	Cloud/VPS review candidates	Broadband ISP	Collisions	Suspected and candidates
AsyncRAT	11,559	804	356	1 (cloud, withheld)	55	3 (2 cert, 1 port)	745 (33 suspected, 712 candidates)
QuasarRAT	2,958	801	670	4 (cloud, withheld)	89	20 (12 cert, 6 port, 2 open-dir)	688 (5 suspected, 683 candidates)
Remcos	2,066	815	539	0 (all withheld)	30	15 (12 cert, 3 port)	770 (22 suspected, 748 candidates)
DcRAT	1,620	800	635	2 (VPS, withheld)	99	16 (14 cert, 2 open-dir)	683 (9 suspected, 674 candidates)
VenomRAT	1,294	800	625	2 (VPS, withheld)	148	9 (9 shared cert)	641 (0 suspected, 641 candidates)
RaptorRAT	22	22	22	0 (all withheld)	0	0	22 (16 suspected, 6 candidates)

Where the infrastructure sits

Ten countries account for 71.9% (2,906 endpoints) of the 4,042 examined ports. The United States leads with 878 endpoints (21.7%), followed by Germany (401, 9.9%), the Netherlands (394, 9.7%), Vietnam (231, 5.7%), and Hong Kong (221, 5.5%). China (203, 5.0%), France (199, 4.9%), Colombia (130, 3.2%), the United Kingdom (126, 3.1%), and Türkiye (123, 3.0%) complete the top ten.

Commercial cloud or VPS provider	Examined endpoints	Primary RAT labels seen
Contabo GmbH and Contabo Inc.	144	AsyncRAT (97 combined), DcRAT
Alibaba Cloud (US and Hangzhou)	132	AsyncRAT, QuasarRAT, DcRAT
Fiba Cloud Operation Company, LLC	113	Remcos, DcRAT
HostPapa	112	Remcos, QuasarRAT
Amazon.com, Inc.	92	QuasarRAT, VenomRAT, RaptorRAT
M247 Europe SRL	90	Remcos, DcRAT
Shenzhen Tencent Computer Systems	77	DcRAT, QuasarRAT
OVH SAS	72	DcRAT, QuasarRAT, VenomRAT
VPS Dedicated LLC	65	AsyncRAT
Microsoft Corporation	64	QuasarRAT, AsyncRAT, RaptorRAT

Hosting splits into two distinct groups. Feed-labelled cloud and VPS candidates and suspected cloud clusters sit on the commercial cloud and VPS providers listed above. By contrast, the 421 dynamic ISP endpoints cluster on broadband and mobile networks in Vietnam (155), Colombia (94), China (38), France (18), Morocco (18), Canada (11), and Russia (11).

CHAPTER 03

Core Open-Source .NET Fleets: AsyncRAT and QuasarRAT

IN SHORT

AsyncRAT and QuasarRAT account for 14,517 of the 19,519 global feed records (74.4%). Five dedicated cloud endpoints (one AsyncRAT and four QuasarRAT) carried upstream confirmed labels without independent C2 corroboration. Retain those cloud candidates for analyst review and hunt default RAT TLS certificates on the wire.

INSIDE THIS CHAPTER

AsyncRAT: port 3389 TLS camouflage and cloud port 443 clusters 10

QuasarRAT: four cloud relay candidates and a broadband feed false positive 11

03 Core Open-Source .NET Fleets: AsyncRAT and QuasarRAT

Both AsyncRAT (MITRE ATT&CK S1087) and QuasarRAT (S0262) encrypt traffic with Microsoft's `.NET` `System.Net.Security.SslStream` library over Windows SChannel sockets (T1573.002). If the builder's bundled X.509 keystore remains unchanged, the server presents a fixed self-signed certificate subject during TLS 1.2 handshakes (T1587.003). The bundled keystore is `ServerCertificate.p12` or `quasar.p12`. Figure 2 maps all six RAT cohorts and co-exposed staging directories to their network surfaces and detections, where external attack surface management helps defenders spot exposed services.

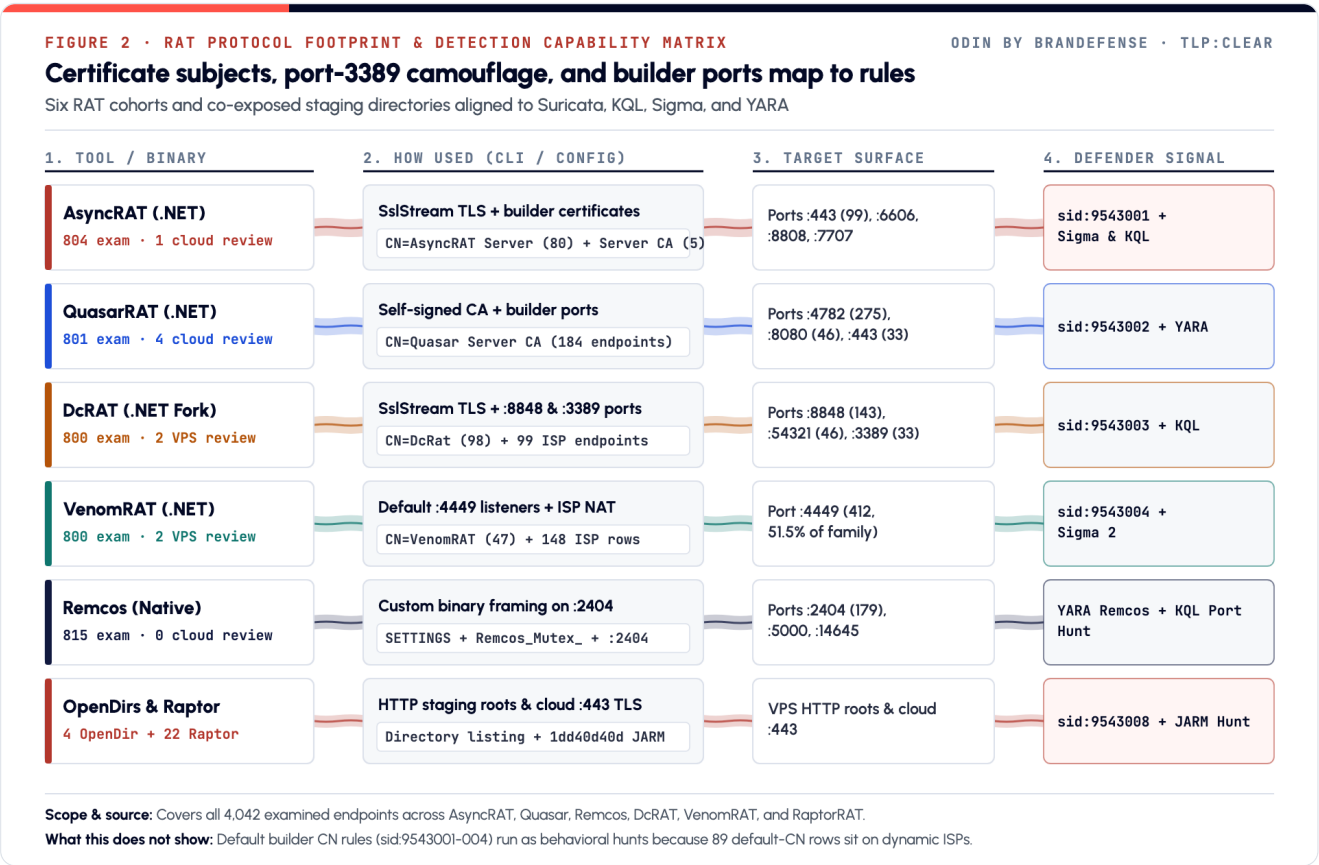


Figure 2: Certificate subjects, port-3389 camouflage, and builder ports map to rules

AsyncRAT: port 3389 TLS camouflage and cloud port 443 clusters

AsyncRAT accounts for 11,559 feed records across 81 countries, from which we examined 804 endpoints on 356 hosts (2.26 ports per host). Default AsyncRAT ports remain common across the sample: TCP 6606 (76 endpoints), 8808 (56), 7707 (42), and 8282 (42), alongside TCP 443 (99) and 8080 (15).

Only one AsyncRAT endpoint carried an upstream confirmed label on dedicated cloud hosting. Its address, provider, port, and certificate details are withheld for a seven-day analyst review window because the inventory records no live beacon or workbench link. Across the wider `.NET` RAT sample, operators also bind TLS listeners to non-standard or camouflaged ports such as TCP 3389 so outbound traffic blends into Remote Desktop sessions (T1571). Alerting on non-RDP Windows processes that open external TCP 3389 catches that camouflage without relying on static IP blocks.

Thirty-three AsyncRAT endpoints sit in our suspected group (all withheld). Twenty-six of them form a TCP 443 cluster on cloud servers in China, Singapore, Vietnam, Thailand, Germany, South Korea, and Indonesia sharing JARM prefix `29d29d00029d`. The other 770 AsyncRAT endpoints split into 712 single-signal candidates, 55 dynamic telecom ISP addresses, 2 shared-certificate collisions, and 1 MySQL port 3306 collision.

QuasarRAT: four cloud relay candidates and a broadband feed false positive

QuasarRAT accounts for 2,958 feed records across 77 countries, with 801 endpoints examined on 670 hosts. Its default builder port TCP 4782 dominates the sample with 275 endpoints (34.3%), followed by 8080 (46), 443 (33), 4783 (13), and 80 (12). Default `CN=Quasar Server CA` certificates appear on 184 endpoints across the full study (176 labelled QuasarRAT, 5 Remcos, 2 AsyncRAT, and 1 DcRAT).

Five QuasarRAT endpoints carried a "confirmed" label in the upstream feed, yet none had independent C2 behaviour in our evidence. One of the five sat on a residential cable broadband network. Its connection details and address are withheld because blocking a home cable lease punishes the next subscriber who receives it.

The other four feed-labelled QuasarRAT review candidates sit on commercial cloud providers. Their addresses, ports, providers, and host-specific fingerprints are withheld pending independent C2 corroboration. Two of the four present the default `CN=Quasar Server CA` certificate, and two present custom or instance-specific certificate subjects (withheld).

Three of the four QuasarRAT cloud records carry upstream listing dates of 1 August 2026. Because cloud virtual machine addresses return to shared pools when deleted, do not block these four endpoints from feed records alone. Retain them for analyst review with a seven-day cloud re-verification window.

04

CHAPTER 04

AsyncRAT Forks, Native Remcos Listeners, and RaptorRAT

IN SHORT

AsyncRAT forks DcRAT and VenomRAT included four feed-labelled VPS review candidates alongside 247 broadband ISP endpoints. Native Windows tool Remcos and cloud cluster RaptorRAT included zero upstream-confirmed endpoints. Withhold all 2,437 endpoints across these four families from static IP block lists, and hunt by certificate and process behavior instead.

INSIDE THIS CHAPTER

DcRAT and VenomRAT: four VPS review candidates vs broadband router sprawl 13

Why all 815 Remcos and 22 RaptorRAT endpoints are withheld 13

04 AsyncRAT Forks, Native Remcos Listeners, and RaptorRAT

Criminal developers regularly fork AsyncRAT's C# project to package credential stealers, hidden VNC desktop modules, or reverse SOCKS proxies. DcRAT and VenomRAT keep AsyncRAT's `.NET SslStream` handshake, whereas Remcos is a native C++ and Delphi remote control program with its own binary wire framing.

DcRAT and VenomRAT: four VPS review candidates vs broadband router sprawl

Within DcRAT (1,620 feed records; 800 examined endpoints on 635 hosts), default port 8848 leads with 143 endpoints, followed by 54321 (46), 3389 (33), 8080 (33), and 443 (30). Two dedicated VPS endpoints carried upstream confirmed labels for DcRAT. Their addresses, provider, ports, shared certificate, and JARM details are withheld for a 30-day analyst review window pending independent C2 behaviour.

Across the remaining 798 DcRAT rows, 99 sit on dynamic telecom ISPs and 14 hit shared-certificate collisions. Two more sit on VPS hosts that also serve an open HTTP directory (connection details withheld). Nine DcRAT endpoints are suspected and 674 remain single-signal candidates.

Within VenomRAT (1,294 feed records; 800 examined endpoints on 625 hosts), default port 4449 accounts for 412 endpoints (51.5%). VenomRAT also shows the heaviest residential telecom footprint in the study, with 148 broadband and mobile ISP endpoints (18.5%), led by 110 endpoints on a national carrier in Vietnam.

Only two dedicated VPS endpoints carried upstream confirmed labels for VenomRAT. Their addresses, providers, ports, certificates, and JARM fingerprints are withheld for a 30-day analyst review window pending independent C2 corroboration.

Why all 815 Remcos and 22 RaptorRAT endpoints are withheld

Remcos (MITRE ATT&CK S0332; 2,066 feed records; 815 examined endpoints on 539 hosts) uses custom binary framing over raw TCP rather than `.NET SslStream` (T1095). Its default port 2404 accounts for 179 examined endpoints, alongside port 5000 (21) and sequential ports 14643 through 14649 (48). Port 2404 rarely completes a standard TLS handshake and is also the IANA port for IEC 60870-5-104 power-grid telecontrol, producing 179 zero-cipher JARM prefixes (000000000000).

Zero Remcos endpoints met multi-signal confirmation. Instead, the 815 rows split into 22 suspected endpoints, 30 dynamic ISP rows, 12 shared certificates, 3 SSH/DNS/SMTPS port collisions (ports 22, 53, and 465), and 748 candidates.

RaptorRAT (22 feed records; 22 examined endpoints on 22 hosts) likewise produced zero confirmed addresses. Sixteen suspected cloud endpoints listen on TCP 443 and share JARM prefix `1dd40d40d000`.

Six more single-signal candidates listen on TCP 2222 (4), 8080 (1), and 80 (1) without multi-signal confirmation. Because that port 443 JARM also appears on eight unconfirmed Remcos and VenomRAT candidates without payload confirmation, all 22 RaptorRAT addresses stay withheld.

Family	Feed-labelled review candidates	Hosting category	Withheld details	Review window
AsyncRAT	1	Commercial cloud	Address, provider, port, certificate, and JARM withheld	7-day analyst review
QuasarRAT	4	Commercial cloud	Addresses, providers, ports, certificates, and JARMs withheld (3 listed 1 Aug 2026)	7-day analyst review
DcRAT	2	Dedicated VPS	Addresses, provider, ports, shared certificate, and JARM withheld	30-day analyst review
VenomRAT	2	Dedicated VPS	Addresses, providers, ports, certificates, and JARMs withheld	30-day analyst review

05

CHAPTER 05

Default Builder Certificates, Broadband ISP Churn, and Collisions

IN SHORT

Default `.NET` RAT certificates appear on 414 examined endpoints, including 89 on residential and mobile broadband networks. Hunt those certificate names in network TLS logs, and never turn broadband router addresses into permanent firewall blocks.

INSIDE THIS CHAPTER

Five default `.NET` builder certificates on 414 endpoints 16

Windows SChannel JARMs, CDN certificates, and standard service ports 17

05 Default Builder Certificates, Broadband ISP Churn, and Collisions

Treating a malware builder's default X.509 certificate name as proof of a static adversary server breaks consumer networks. Small-scale RAT operators often run listeners on home computers behind fiber or cable routers, forwarding ports 4449, 4782, 5000, 6606, or 8848 with dynamic DNS.

When an ISP reassigns a broadband lease, a former listener's IPv4 address may move to another subscriber. Figure 3 shows how broadband DHCP churn turns naive IP block lists into outages, and how wire-level TLS hunting avoids that trap.

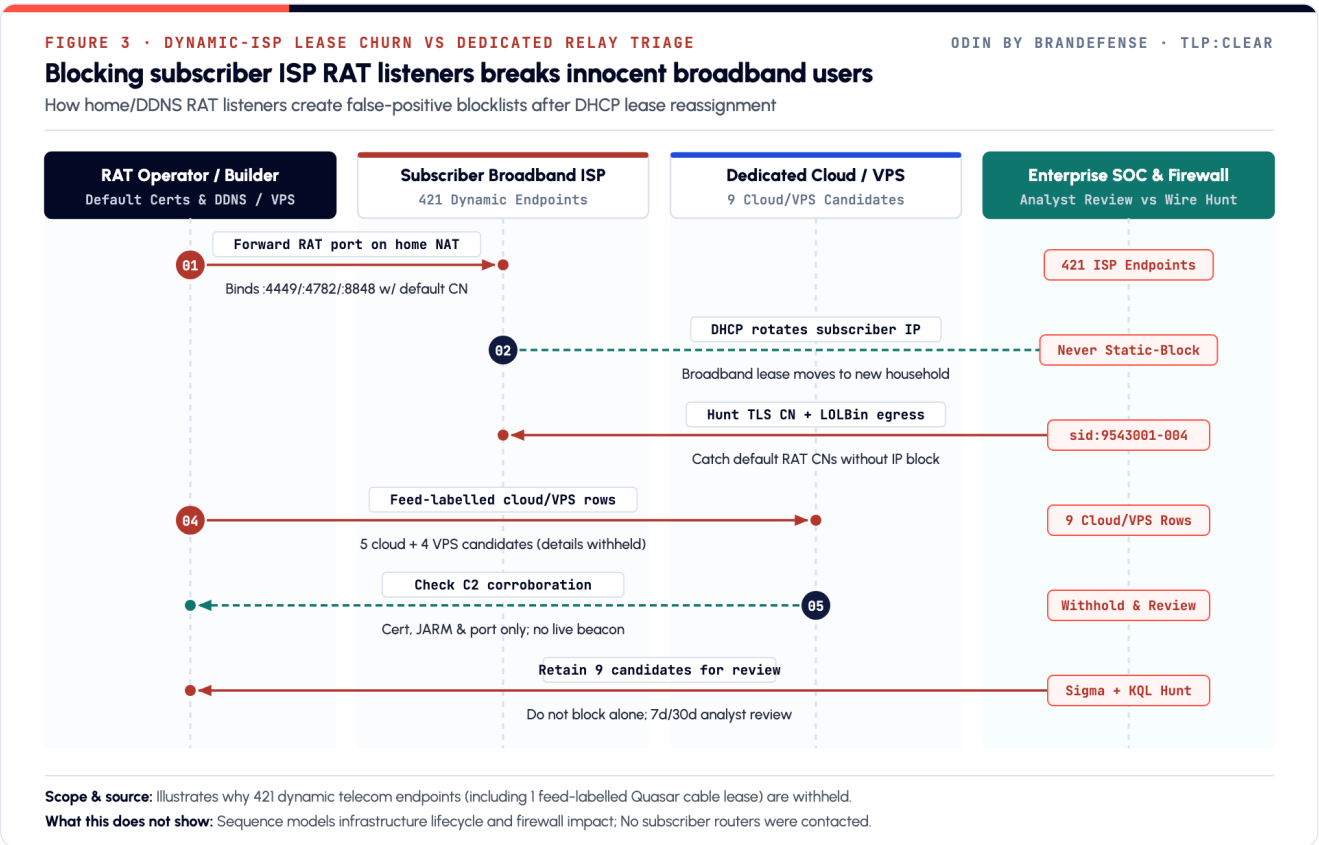


Figure 3: Blocking subscriber ISP RAT listeners breaks innocent broadband users

Five default `.NET` builder certificates on 414 endpoints

Across all 4,042 examined endpoints, 2,129 (52.7%) returned no TLS certificate on raw TCP or reset sockets, while 1,913 presented an X.509 certificate. Five built-in `.NET` RAT builder subject strings account for 414 endpoints (21.6% of all TLS-bearing endpoints):

- **`CN=Quasar Server CA`**: 184 endpoints across 164 hosts (2 feed-labelled cloud relay candidates, 38 broadband ISP endpoints, 5 port collisions, and 139 single-signal candidates).
- **`CN=DcRat`**: 98 endpoints across 69 hosts (0 cloud review candidates, 18 broadband ISP endpoints, and 80 single-signal candidates).
- **`CN=AsyncRAT Server`**: 80 endpoints across 52 hosts (0 cloud review candidates, 20 broadband ISP endpoints, and 60 single-signal candidates).

- `CN=VenomRAT`: 47 endpoints across 35 hosts (0 cloud review candidates, 13 broadband ISP endpoints, and 34 single-signal candidates).
- `CN=AsyncRAT Server CA`: 5 candidate endpoints across 5 hosts.

More than one in five default-certificate endpoints (89 of 414, or 21.5%) sits on a residential or mobile telecom network. Feeds also cross-label families sharing `.NET` builder code: `CN=DcRat` appears on 73 DcRAT, 9 VenomRAT, 8 Remcos, 6 AsyncRAT, and 2 QuasarRAT records.

On individual broadband routers in the sample, multiple forwarded AsyncRAT and DcRAT listener ports (ports withheld) appear alongside consumer router certificates. Figure 4 compares the nine feed-labelled cloud and VPS relay candidate stacks with the 421 dynamic ISP and 63 collision endpoints.

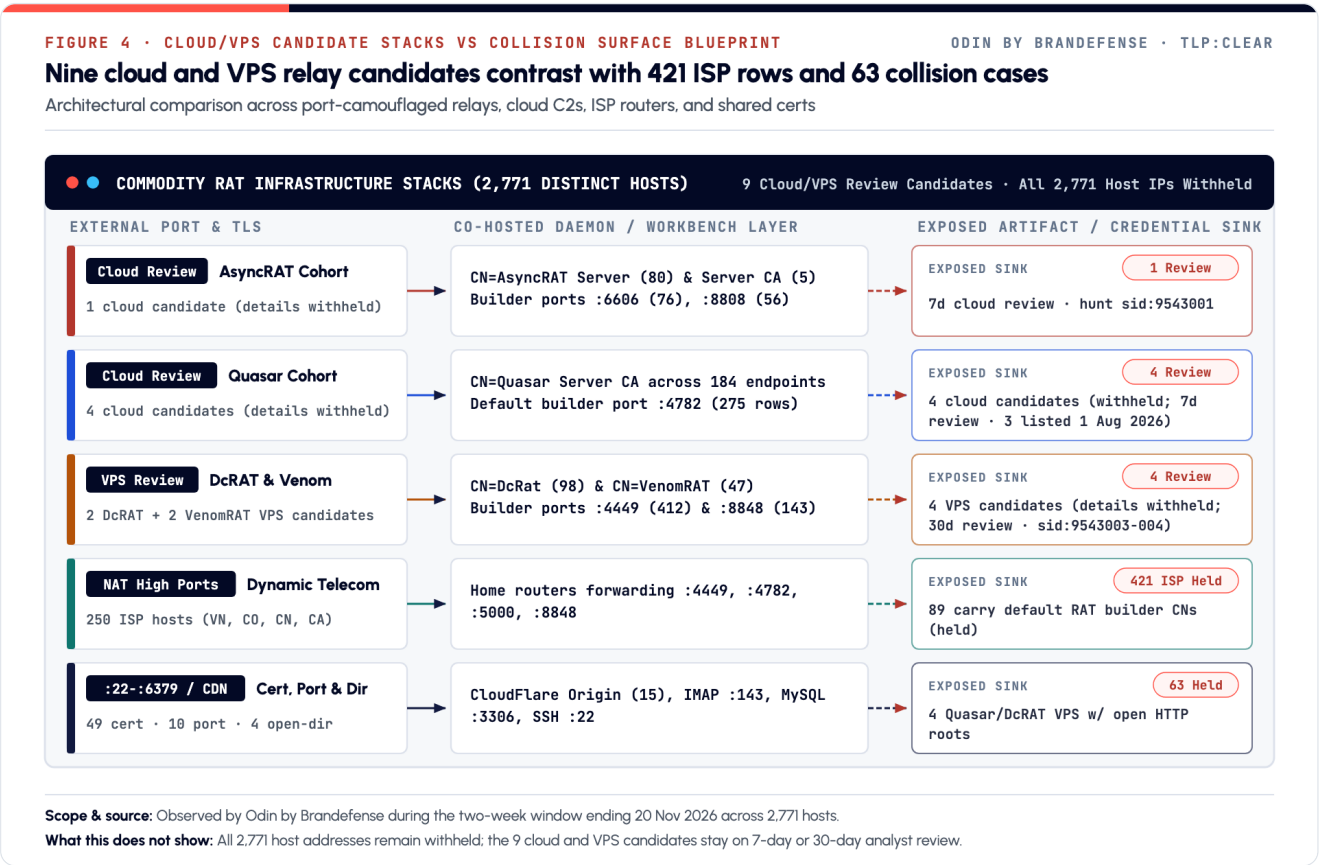


Figure 4: Nine cloud and VPS relay candidates contrast with 421 ISP rows and 63 collision cases

Windows SChannel JARMs, CDN certificates, and standard service ports

Three collision patterns explain why single-signal TLS and port feeds generate false positives:

- **Windows SChannel JARM overlap (1,973 JARM-bearing endpoints across 315 unique JARMs).** C# `SslStream` inherits the underlying Windows Server TLS cipher list. As a result, the most common JARM in our study (`22b22b00022b` prefix, 231 endpoints) spans five RAT feed labels (80 AsyncRAT, 73 DcRAT, 53 VenomRAT, 14 Remcos, and 11 QuasarRAT) plus ordinary Windows servers.
- **CDN, proxy, WinRM, and router certificates (49 shared-certificate collisions plus generic defaults).** Fifteen shared-certificate rows sit behind `CN=CloudFlare Origin Certificate`. Across the wider candidate set, feeds also swept up `CN=localhost` (85), `CN=Server` (41), `CN=TRAEFIK DEFAULT CERT` (25), `CN=Cloudbase-Init WinRM` (24), `CN=AnyDesk Client` (20), `CN=ASUS` (11), and `CN=Vigor Router` (9).

· **Standard service ports (10) and co-exposed open directories (4).** Ten endpoints collided with everyday service ports: three IMAP port 143 listeners at a US cloud provider, two MySQL port 3306 servers, Redis (6379), and MS-SQL (1433). Three Remcos rows collided with SSH (22), DNS (53), and SMTPS (465). Four more VPS hosts (two QuasarRAT and two DcRAT; connection details withheld) also served open HTTP directories.

Certificate or collision group	Endpoints	Hosts	Cloud/VPS review	Withheld IPs	Defender guidance
`CN=Quasar Server CA` (default)	184	164	2	184	Hunt with SID 9543002; require independent C2 evidence before blocking
`CN=DcRat` (default)	98	69	0	98	Hunt with SID 9543003; seen across five feed labels
`CN=AsyncRAT Server` and `Server CA`	85	57	0	85	Hunt with SID 9543001; 80 `Server` plus 5 `Server CA`
`CN=VenomRAT` (default)	47	35	0	47	Hunt with SID 9543004; 27.7% sit on broadband ISPs
Custom or cloud candidate certs	7	7	7	7	All 7 candidate-specific certificate subjects withheld pending C2 corroboration
Generic proxy, WinRM, and router certs	238	190	0	238	`localhost` (85), `Traefik` (25), `WinRM` (24), `CloudFlare` (17)
Dynamic subscriber telecom pool	421	250	0	421	Never block by IP; DHCP rotation hits innocent users
Standard ports (10) and open dirs (4)	14	14	0	14	Ports 22, 53, 143, 465, 1433, 3306, 6379 and 4 HTTP dirs

CHAPTER 06

Feed Labels Against Corroborated C2 Evidence

IN SHORT

Feeding raw commodity RAT lists into a firewall could block 250 subscriber IPs, 42 shared-certificate hosts, and 10 mail, DNS, SSH, or database endpoints. Only nine dedicated cloud and VPS endpoints (0.22% of examined endpoints, 0.046% of global feed records) carried upstream confirmed labels. Even those nine require analyst review rather than automatic blocking until live C2 behaviour corroborates them.

INSIDE THIS CHAPTER

Four other ways to read the 4,042 examined endpoints 21

06 Feed Labels Against Corroborated C2 Evidence

Good infrastructure defense is defined by what it refuses to block. Figure 5 compares the global feed volume for each RAT family and the 421-endpoint broadband ISP pool against the nine feed-labelled cloud and VPS relay candidates.

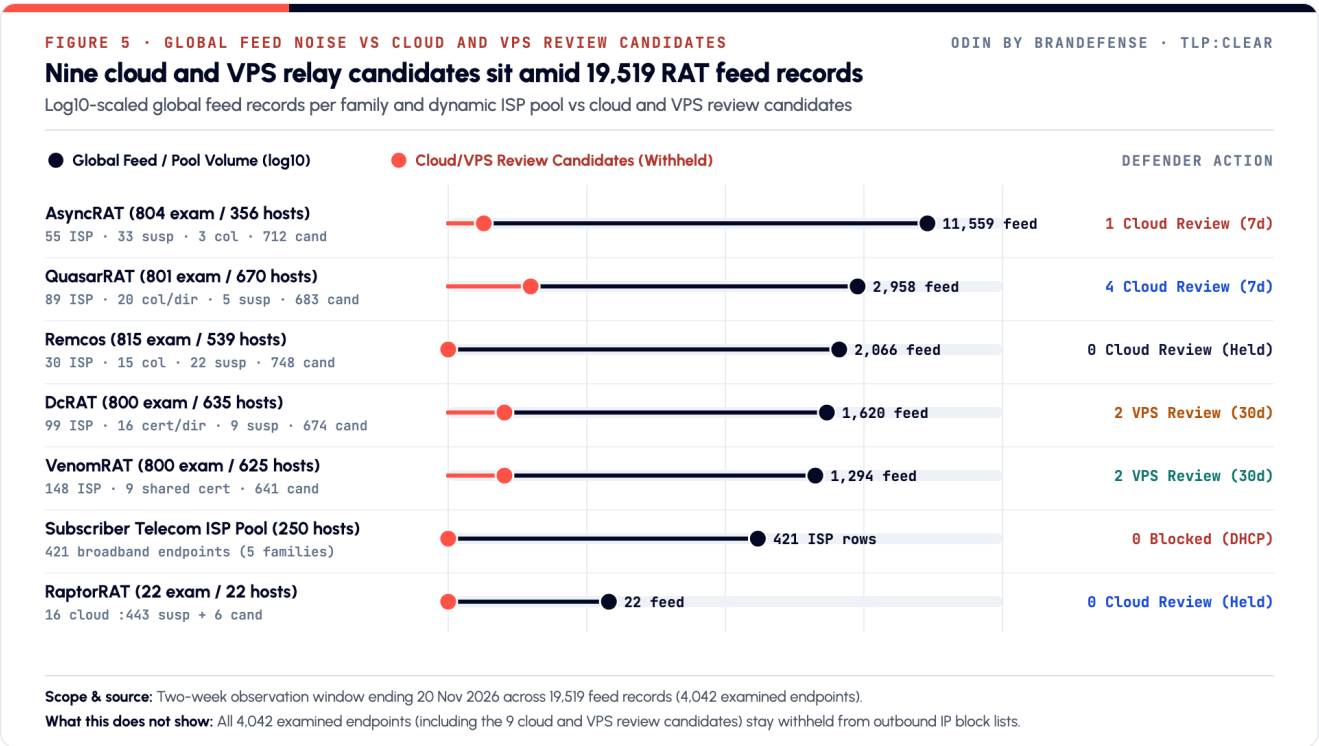


Figure 5: Nine cloud and VPS relay candidates sit amid 19,519 RAT feed records

Group in our analysis	Feed "confirmed"	Feed "suspected"	Feed "candidate"	Total	Share	Action
Dedicated cloud and VPS relay candidates	9	0	0	9	0.22%	Withhold; analyst review (7-day cloud / 30-day VPS)
Suspected C2 clusters	0	85	0	85	2.10%	Withhold; hunt with TLS and endpoint logs
Dynamic broadband and mobile ISPs	1	2	418	421	10.42%	Withhold; hunt by TLS cert instead
Shared-certificate collisions	0	0	49	49	1.21%	Withhold; do not block shared origins
Standard service-port collisions	0	3	7	10	0.25%	Withhold; protect SSH, DNS, mail, DBs
Co-exposed open HTTP directories	0	0	4	4	0.10%	Withhold; hunt staging pages with SID 9543008
Single-signal candidates	0	0	3,464	3,464	85.70%	Withhold; insufficient evidence to block

Four other ways to read the 4,042 examined endpoints

1. **Home routers and expired dynamic DNS domains (421 endpoints on 250 hosts).** Operators may forward ports on consumer routers (`CN=ASUS`, `CN=Vigor Router`, `CN=OpenWrt`) via dynamic DNS. Even when a feed records `CN=Quasar Server CA` on cable broadband, DHCP rotation passes that address to ordinary subscribers.
2. **Normal Windows Server SChannel, WinRM, and RDP services (3,464 candidates).** Legitimate `.NET` services, Remote Desktop, and `Cloudbase-Init WinRM` (24 endpoints) use the same Windows SChannel cipher suites (`22b22b` and `14d14d` JARM prefixes) as `.NET` RATs.
3. **CDN origins, reverse proxies, and core ports (59 collision endpoints).** Reverse proxies (`CloudFlare Origin Certificate` on 15, `TRAEFIK DEFAULT CERT` on 25) and services on ports 22, 53, 143, 465, 1433, 3306, and 6379 trip loose heuristics.
4. **Lab servers and reassigned cloud addresses (4 open-directory hosts and 5 cloud candidates).** Open-source AsyncRAT and QuasarRAT builds are used in test labs, and the five cloud candidates (three last listed on 1 August 2026) require seven-day lease review.

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

Perimeter firewall and network security

- [ ] Do not block these addresses from this evidence alone; retain them for analyst review until each has current, independent C2 corroboration. Apply a seven-day re-verification window for the five cloud candidates and a 30-day window for the four VPS candidates.
- [ ] Allow outbound public TCP 3389 only from approved Remote Desktop jump hosts (`CORP-RDP-GW-01`, `CORP-PAM-BASTION-01`) and native Windows Remote Desktop binaries (`C:\Windows\System32\mstsc.exe`).

SOC and endpoint detection

- [ ] Deploy the Suricata 7 signatures in Chapter 8 on perimeter and TLS-inspection links so `CN=AsyncRAT Server`, `CN=Quasar Server CA`, `CN=DcRat`, and `CN=VenomRAT` handshakes trigger hunts even when dynamic DNS or cloud addresses rotate.
- [ ] Run the Sentinel KQL query and Sigma rules across Windows endpoint logs. Alert whenever PowerShell, MSBuild, RegAsm, InstallUtil, WScript, or MSHTA connects to external port 3389 or default RAT ports (2404, 4449, 4782, 4783, 6606, 7707, 8808, 8848).
- [ ] Scan unpacked `.NET` assemblies, quarantine files, and memory dumps with the three YARA rules for `SslStream` certificate-pinning strings, `Quasar Server CA` client stubs, and Remcos `SETTINGS` resources.

Threat intelligence management

- [ ] Keep residential broadband and mobile telecom ASNs, shared Cloudflare or Traefik certificates, standalone JARM matches, and standard ports (22, 53, 143, 465, 1433, 3306, 6379) out of automatic firewall block lists.

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 25

How the KQL, Sigma, and YARA rules work 25

MITRE ATT&CK mapping 26

08 Detection Rules

The full rules are in Appendix B. The Suricata file passes a static syntax lint with zero problems. The rules were not loaded on live Suricata, Sentinel Kusto, Sigma, or YARA engines for this edition, so test and tune them in your environment before turning on automated containment.

Rule file	What it catches
`odin_commodity_dotnet_rats_asyncrat_quasar_dcrat_venomrat.rules`	Suricata 7 (5 rules): default AsyncRAT, Quasar, DcRat, and VenomRAT TLS certificates and co-exposed RAT open directories
`sentinel_commodity_dotnet_rat_c2_and_tls_hunt.kql`	Microsoft Sentinel and Defender XDR: traffic to analyst-listed review addresses, non-RDP processes using external TCP 3389, and `.NET` or script binaries hitting default RAT ports
`sigma_commodity_dotnet_rat_tls_and_port_anomalies.yml`	Sigma (2 Windows network rules): non-RDP processes opening external port 3389, and scripting or `.NET` binaries reaching default RAT ports
`odin_commodity_dotnet_rat_config_and_cert_artifacts.yar`	YARA (3 rules): AsyncRAT/DcRAT/VenomRAT `SslStream` configs, Quasar `Quasar Server CA` client stubs, and Remcos `SETTINGS` binaries

How the Suricata signatures work

SID	Looks for	Why
9543001	`CN=AsyncRAT Server`	Default AsyncRAT certificate (hunt)
9543002	`CN=Quasar Server CA`	Default QuasarRAT CA subject (hunt)
9543003	`CN=DcRat`	Default DcRAT certificate (hunt)
9543004	`CN=VenomRAT`	Default VenomRAT certificate (hunt)
9543008	HTTP 200 directory listing with RAT or `.p12` names	Co-exposed RAT staging directory (hunt)

TLS 1.3 encrypts server certificates on the wire, whereas `.NET SslStream` on older Windows Server builds often negotiates TLS 1.2 where Suricata reads `tls.cert_subject` directly. Keep SIDs 9543001 through 9543004 and 9543008 in behavioral-hunt mode where sandboxes run.

How the KQL, Sigma, and YARA rules work

The Sentinel KQL query looks back seven days across `DeviceNetworkEvents` and `DeviceProcessEvents`. Connections to the listed IPv4 addresses bypass host exclusions and return an `AlertAction` label for analyst review; this query does not isolate devices. Unconfirmed traffic alerts when non-RDP binaries open public TCP

3389 outside exact `mstsc.exe` or `msrdcw.exe` paths and jump hosts. A companion join flags PowerShell, MSBuild, RegAsm, InstallUtil, WScript, or MSHTA hitting default RAT ports (2404, 4449, 4782, 4783, 6606, 7707, 8808, 8848, 54321).

Rule	Level / Action	Fires when
Sigma Rule 1 (`...0002`)	High	Non-RDP Windows binary connects to public TCP 3389
Sigma Rule 2 (`...0003`)	High	Scripting or `.NET` binary connects to ports 2404, 4449, 4782, 4783, 6606, 7707, 8808, or 8848
`ODIN_AsyncRAT_DcRAT_VenomRAT_DotNet_Config_Strings`	Hunt / Contain	File (<2 MB) holds AsyncRAT/DcRat/VenomRAT, `SslStream` / `ServerCertificate`, and `Pastebin` / `Aes256`
`ODIN_QuasarRAT_Client_Certificate_And_Config_Artifact`	Hunt / Contain	File (<2 MB) holds `Quasar Server CA` with two Quasar client or `SslStream` markers
`ODIN_Remcos_Native_Settings_And_Mutex_Artifact`	Hunt / Contain	File (<2 MB) holds `BreakingSecurity` / `Remcos_Mutex_` with `SETTINGS`, `Watchdog`, or :2404

MITRE ATT&CK mapping

Tactic	Technique	Study observation	Detection
Resource Development	T1587.003 Digital Certificates	Default self-signed certs on 414 endpoints	SIDs 9543001 to 9543004; YARA
Command and Control	T1573.002 Asymmetric Cryptography	`.NET` `SslStream` TLS handshakes	SIDs 9543001 to 9543004; KQL
Command and Control	T1571 Non-Standard Port	Builder ports (4449, 4782, 6606, 8848) and TCP 3389 camouflage	Sigma (`...0002`, `...0003`); KQL
Command and Control	T1095 Non-Application Layer Protocol	Native TCP framing on Remcos 2404	Sigma (`...0003`); KQL; YARA

09

CHAPTER 09

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

JARM fingerprints to hunt 28

Certificate subject names (text-only hunt indicators) 29

09 Indicators

Appendix A holds 10 published indicators: 10 TLS JARM fingerprints for hunting. There are no published IPv4 addresses or file hashes in this report; the nine feed-labelled cloud and VPS relay addresses from the v1.0 register are withheld pending independent, address-specific C2 corroboration. Five more text-only indicators appear below: the five default `.NET` builder certificate names seen across 414 examined endpoints.

Type	Count	Where to use it
TLS JARM fingerprints	10	TLS hunting paired with endpoint process logs; never block on a JARM alone
Default builder certificate subjects (text-only)	5	Suricata (`tls.cert_subject`) and Zeek TLS hunting across 414 endpoints
Feed-labelled cloud and VPS relay addresses	9 (withheld)	Withheld from public block lists; retain for analyst review (7-day cloud / 30-day VPS)

JARM fingerprints to hunt

JARM fingerprint	Family	Context
`22b22b00022b22b22b22b22b22b22bd3b67dd3674d9af9dd91c1955a35d0e9`	AsyncRAT	Windows SChannel TLS fingerprint shared across 231 examined endpoints across five RAT families
`2ad2ad16d2ad2ad0002ad2ad2ad2add3b67dd3674d9af9dd91c1955a35d0e9`	QuasarRAT	Windows SChannel TLS fingerprint shared across 121 examined endpoints
`40d40d40d00040d00042d42d000000e1ea2a807a629b496b664cf07ad7c08d`	QuasarRAT	QuasarRAT TLS fingerprint for wire-level hunting paired with endpoint process logs
`14d14d16d14d14d08c14d14d14d14dfd9c9d14e4f4f67f94f0359f8b28f532`	QuasarRAT	Windows SChannel TLS fingerprint shared across 160 examined endpoints across the sample
`29d29d00029d29d00042d42d000000301510f56407964db9434a9bb0d4ee4a`	QuasarRAT	QuasarRAT TLS fingerprint shared across two cloud endpoints (hunt with process logs)
`2ad2ad16d2ad2ad00042d42d00000061256d32ed7779c14686ad100544dc8d`	DcRAT	TLS fingerprint shared across 75 examined endpoints across the sample
`27d40d40d00040d00042d43d000000d2e61cae37a985f75ecafb81b33ca523`	VenomRAT	TLS fingerprint shared across 29 examined endpoints across the sample
`3fd3fd0003fd3fd21c42d42d000000bdfc58c9a46434368cf60aa440385763`	VenomRAT	TLS fingerprint shared across 50 examined endpoints across the sample
`29d29d00029d29d21c41d41d000000bb69d90480e646db0082ba62fb24e3c8`	AsyncRAT	Shared across 26 suspected port 443 cloud endpoints
`1dd40d40d00040d00042d43d000000831b6af40378e2dd35eeac4e9311926e`	RaptorRAT	Shared across all 16 suspected port 443 cloud endpoints

Certificate subject names (text-only hunt indicators)

Certificate subject	Family	Observed scope and use
`CN=Quasar Server CA`	QuasarRAT	Default builder CA across 184 endpoints (2 cloud review candidates, 38 ISP); SID 9543002
`CN=DcRat`	DcRAT	Default builder CN across 98 endpoints (18 ISP); SID 9543003
`CN=AsyncRAT Server`	AsyncRAT	Default builder CN across 80 endpoints (20 ISP); SID 9543001
`CN=AsyncRAT Server CA`	AsyncRAT	Builder root CA variant across 5 candidate endpoints; TLS hunting
`CN=VenomRAT`	VenomRAT	Default builder CN across 47 endpoints (13 ISP); SID 9543004

We withhold all 4,042 examined endpoint addresses across 2,771 hosts. That total covers 9 cloud and VPS review candidates, 85 suspected C2 endpoints, 421 broadband ISP rows, 63 collision rows, and 3,464 single-signal candidates. Every unverified Windows machine hostname and all seven custom or host-specific certificate subjects on the cloud and VPS review candidates are also withheld.

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

- **Observation window.** Findings cover a two-week observation window ending 20 November 2026. Earlier dates on three QuasarRAT cloud entries (1 August 2026) are upstream feed listing dates, which is one reason those addresses remain on analyst review rather than automatic block lists.
- **Feed labels are not attribution.** Family labels and upstream "confirmed" tags come from threat feeds. Because our inventory records hosting, port, certificate, and JARM without a live protocol response, beacon, or workbench link, all nine upstream-confirmed cloud and VPS addresses are withheld pending independent C2 corroboration.
- **Analytical model.** Zero dedicated cloud or VPS relay candidates and zero custom certificates link two different RAT families. Figures 1 and 3 present an analytical model of shared builder patterns and broadband ISP churn rather than a single actor.
- **No subscriber routers contacted.** Odin by Brandefense did not interact with victim endpoints or consumer routers, and all 250 dynamic telecom hosts remain unpublished.
- **Rule validation.** Detection rules were checked statically and were not loaded on live engines for this edition.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. MITRE ATT&CK. S1087 AsyncRAT. https://attack.mitre.org/software/S1087/ (verified 20 November 2026)
2. MITRE ATT&CK. S0262 QuasarRAT. https://attack.mitre.org/software/S0262/ (verified 20 November 2026)
3. MITRE ATT&CK. S0332 Remcos. https://attack.mitre.org/software/S0332/ (verified 20 November 2026)
4. MITRE ATT&CK. T1573.002 Encrypted Channel: Asymmetric Cryptography. https://attack.mitre.org/techniques/T1573/002/ (verified 20 November 2026)
5. MITRE ATT&CK. T1587.003 Develop Capabilities: Digital Certificates. https://attack.mitre.org/techniques/T1587/003/ (verified 20 November 2026)
6. MITRE ATT&CK. T1571 Non-Standard Port. https://attack.mitre.org/techniques/T1571/ (verified 20 November 2026)
7. Microsoft Learn. System.Net.Security.SslStream Class. https://learn.microsoft.com/en-us/dotnet/api/system.net.security.sslstream (verified 20 November 2026)
8. Salesforce Engineering. JARM: active TLS server fingerprinting. https://github.com/salesforce/jarm (verified 20 November 2026)
9. MITRE ATT&CK. T1095 Non-Application Layer Protocol. https://attack.mitre.org/techniques/T1095/ (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 (10) 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
R43-IOC-01	JARM	22b22b00022b22b22b22b22b22bd3b67dd3674d9af9dd91c1955a35d0e9		asyncrat · Windows SChannel TLS fingerprint shared across 231 examined endpoints across five RAT families (hunt with endpoint process logs)	HUNT
R43-IOC-02	JARM	2ad2ad16d2ad2ad0002ad2ad2ad2add3b67dd3674d9af9dd91c1955a35d0e9		quasar · Windows SChannel TLS fingerprint shared across 121 examined endpoints (hunt with endpoint process logs)	HUNT
R43-IOC-03	JARM	40d40d40d00040d00042d42d000000e1ea2a807a629b496b664cf07ad7c08d		quasar · QuasarRAT TLS fingerprint for wire-level hunting paired with endpoint process logs	HUNT
R43-IOC-04	JARM	14d14d16d14d14d08c14d14d14d14dfd9c9d14e4f4f67f94f0359f8b28f532		quasar · Windows SChannel TLS fingerprint shared across 160 examined endpoints across the sample (hunt with endpoint process logs)	HUNT
R43-IOC-05	JARM	29d29d00029d29d00042d42d000000301510f56407964db9434a9bb0d4ee4a		quasar · QuasarRAT TLS fingerprint shared across two cloud endpoints (hunt with endpoint process logs)	HUNT
R43-IOC-06	JARM	2ad2ad16d2ad2ad00042d42d000000061256d32ed7779c14686ad100544dc8d		dcrat · TLS fingerprint shared across 75 examined endpoints across the sample (hunt with endpoint process logs)	HUNT
R43-IOC-07	JARM	27d40d40d00040d00042d43d000000d2e61cae37a985f75ecafb81b33ca523		venomrat · TLS fingerprint shared across 29 examined endpoints across the sample (hunt with endpoint process logs)	HUNT
R43-IOC-08	JARM	3fd3fd0003fd3fd21c42d42d000000bdfc58c9a46434368cf60aa440385763		venomrat · TLS fingerprint shared across 50 examined endpoints across the sample (hunt with endpoint process logs)	HUNT
R43-IOC-09	JARM	29d29d00029d29d21c41d41d000000bb69d90480e646db0082ba62fb24e3c8	443	asyncrat · TLS fingerprint shared across 26 suspected port 443 cloud endpoints	HUNT
R43-IOC-10	JARM	1dd40d40d00040d00042d43d000000831b6af40378e2dd35eeac4e9311926e	443	raptorrat · TLS fingerprint shared across 16 suspected port 443 cloud endpoints	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_commodity_dotnet_rat_config_and_cert_artifacts.yar

YARA

```
/*
   Odin by Brandefense - Monograph 43 / 46 Bounded YARA Detection Suite
   Covers: AsyncRAT/DcRAT/VenomRAT .NET SslStream & ServerCertificate stubs,
           QuasarRAT Server CA configuration artifacts, and Remcos native
           SETTINGS markers.
   Classification: TLP:CLEAR
*/

rule ODIN_AsyncRAT_DcRAT_VenomRAT_DotNet_Config_Strings {
    meta:
        author = "Odin by Brandefense"
        monograph = "43 / 46"
        date = "2026-11-20"
        action = "hunt_and_contain"
        description = "Matches .NET commodity RAT configuration and SslStream certificate-pinning markers shared
by AsyncRAT, DcRAT, and VenomRAT"
    strings:
        $fam1 = "AsyncRAT Server" ascii wide
        $fam2 = "DcRat" ascii wide
        $fam3 = "VenomRAT" ascii wide
        $tls1 = "ServerCertificate" ascii wide
        $tls2 = "SslStream" ascii wide
        $tls3 = "X509Certificate2" ascii wide
        $cfg1 = "Pastebin" ascii wide
        $cfg2 = "Aes256" ascii wide
        $cfg3 = "Anti_Analysis" ascii wide
    condition:
        filesize < 2MB and 1 of ($fam*) and 1 of ($tls*) and 1 of ($cfg*)
}

rule ODIN_QuasarRAT_Client_Certificate_And_Config_Artifact {
    meta:
        author = "Odin by Brandefense"
        monograph = "43 / 46"
        date = "2026-11-20"
        action = "hunt_and_contain"
        description = "Matches QuasarRAT embedded Quasar Server CA certificate subject alongside client
serialization or SslStream markers"
    strings:
        $ca = "Quasar Server CA" ascii wide
        $q1 = "Quasar.Common" ascii wide
        $q2 = "Quasar.Client" ascii wide
        $q3 = "VerifyHash" ascii wide
        $q4 = "SslStream" ascii wide
    condition:
        filesize < 2MB and $ca and 2 of ($q*)
}

rule ODIN_Remcos_Native_Settings_And_Mutex_Artifact {
    meta:
        author = "Odin by Brandefense"
        monograph = "43 / 46"
        date = "2026-11-20"
        action = "hunt_and_contain"
        description = "Matches Remcos native client configuration resource, mutex, and BreakingSecurity builder
strings"
    strings:
        $r1 = "BreakingSecurity" ascii wide
        $r2 = "Remcos_Mutex_" ascii wide
        $r3 = "remcos.exe" ascii wide nocase
```

```
        $s1 = "SETTINGS" ascii wide
        $s2 = "Watchdog" ascii wide
        $s3 = ":2404" ascii wide
    condition:
        filesize < 2MB and 1 of ($r*) and 2 of ($s*)
}
```

odin_commodity_dotnet_rats_asyncrat_quasar_dcrat_venomrat.rules

SURICATA

```
# Odin by Brandefense - Monograph 43 / 46 Suricata 7.x Detection Suite
# Coverage: Default self-signed builder certificates (AsyncRAT, QuasarRAT, DcRAT, VenomRAT)
#           and co-exposed commodity RAT staging HTTP directories
# Classification: TLP:CLEAR

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R43] HUNT Default AsyncRAT Server Self-Signed TLS
Certificate"; flow:established,to_client; tls.cert_subject; bsize:>17; content:"CN=AsyncRAT Server";
fast_pattern; classtype:trojan-activity; sid:9543001; rev:1; metadata:created_at 2026_11_20, deployment
perimeter_and_proxy, policy behavioral_hunt, tactic command_and_control;)

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R43] HUNT Default Quasar Server CA Self-Signed TLS
Certificate"; flow:established,to_client; tls.cert_subject; bsize:>18; content:"CN=Quasar Server CA";
fast_pattern; classtype:trojan-activity; sid:9543002; rev:1; metadata:created_at 2026_11_20, deployment
perimeter_and_proxy, policy behavioral_hunt, tactic command_and_control;)

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R43] HUNT Default DcRat Self-Signed TLS Server
Certificate"; flow:established,to_client; tls.cert_subject; bsize:>7; content:"CN=DcRat"; fast_pattern; pcre:"/
CN=DcRat(?:,|$)/"; classtype:trojan-activity; sid:9543003; rev:1; metadata:created_at 2026_11_20, deployment
perimeter_and_proxy, policy behavioral_hunt, tactic command_and_control;)

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R43] HUNT Default VenomRAT Self-Signed TLS Server
Certificate"; flow:established,to_client; tls.cert_subject; bsize:>10; content:"CN=VenomRAT"; fast_pattern;
pcre:"/CN=VenomRAT(?:,|$)/"; classtype:trojan-activity; sid:9543004; rev:1; metadata:created_at 2026_11_20,
deployment perimeter_and_proxy, policy behavioral_hunt, tactic command_and_control;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R43] HUNT Co-Exposed Commodity RAT Staging Open-
Directory Listing"; flow:established,to_client; http.stat_code; bsize:3; content:"200"; http.response_body;
bsize:>48; content:"Directory listing for "; fast_pattern; pcre:"/(asyncrat|quasar|dcrat|venomrat|remcos|
raptorrat|ServerCertificate\.p12)/i"; classtype:attempted-recon; sid:9543008; rev:1; metadata:created_at
2026_11_20, deployment perimeter_and_proxy, policy behavioral_hunt, tactic resource_development;)
```

sentinel_commodity_dotnet_rat_c2_and_tls_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 43 / 46 Microsoft Sentinel & Defender XDR Hunt
// Correlates outbound endpoint connections to analyst-listed AsyncRAT, QuasarRAT, DcRAT,
// and VenomRAT relay candidates (populate ListedCommodityRatIps after independent C2 corroboration),
// non-RDP processes opening external TCP/3389 sessions, and .NET/LOLBin execution paired with
// default commodity RAT builder ports (2404, 4449, 4782, 4783, 6606, 7707, 8808, 8848, 54321)
// Classification: TLP:CLEAR

let Lookback = 7d;
let ApprovedRedTeamHosts = dynamic(["REDTEAM-WIN-SIM-01", "PURPLE-RAT-LAB-01"]);
let ApprovedRdpJumpHosts = dynamic(["CORP-RDP-GW-01", "CORP-PAM-BASTION-01"]);
let ApprovedRdpBinaries = dynamic([
    @"c:\windows\system32\mstsc.exe",
    @"c:\windows\syswow64\mstsc.exe",
    @"c:\program files\remote desktop\msrdcw.exe"
]);
let ListedCommodityRatIps = dynamic([
    "192.0.2.10", "198.51.100.10"
]);
let DefaultRatPorts = dynamic([2404, 4449, 4782, 4783, 6606, 7707, 8808, 8848, 54321]);
let SuspiciousHostProcs = DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedRedTeamHosts))
```

```
    | where FileName in~ ("powershell.exe", "pwsh.exe", "msbuild.exe", "regasm.exe", "regsvcs.exe",
"installutil.exe", "vbc.exe", "csc.exe", "wscript.exe", "cscript.exe", "mshta.exe")
    | project ProcTime = Timestamp, DeviceId, DeviceName, ProcessId, ProcessCreationTime, ProcFile = FileName,
FolderPath, ProcessCommandLine;
DeviceNetworkEvents
| where Timestamp >= ago(Lookback)
| where RemoteIPType == "Public"
| extend InitPathLower = tolower(InitiatingProcessFolderPath)
| where RemoteIP in (ListedCommodityRatIps)
    or (
        not(DeviceName in~ (ApprovedRedTeamHosts))
        and (
            RemotePort in (DefaultRatPorts)
            or (RemotePort == 3389 and not(DeviceName in~ (ApprovedRdpJumpHosts)) and not(InitPathLower in
(ApprovedRdpBinaries)))
        )
    )
| summarize
    NetFirstSeen = min(Timestamp),
    NetLastSeen = max(Timestamp),
    DistinctRemoteIps = dcount(RemoteIP),
    ListedIpHits = countif(RemoteIP in (ListedCommodityRatIps)),
    Port3389NonRdpHits = countif(RemotePort == 3389 and not(InitPathLower in (ApprovedRdpBinaries))),
    SampleEndpoints = make_set(strcat(RemoteIP, ":", tostring(RemotePort)), 12),
    ConnectionCount = count()
    by DeviceId, DeviceName, InitiatingProcessId, InitiatingProcessCreationTime, InitiatingProcessFileName,
InitiatingProcessFolderPath
| join kind=leftouter (SuspiciousHostProcs) on DeviceId, DeviceName, $left.InitiatingProcessId ==
$right.ProcessId, $left.InitiatingProcessCreationTime == $right.ProcessCreationTime
| extend AlertAction = case(
    ListedIpHits > 0, "REVIEW_LISTED_COMMODITY_RAT_C2_CANDIDATE",
    isnotempty(ProcessCommandLine) and Port3389NonRdpHits > 0, "CONTAIN_LOLBIN_EXTERNAL_PORT_3389_CAMOUFLAGE",
    isnotempty(ProcessCommandLine), "CONTAIN_LOLBIN_DEFAULT_COMMODITY_RAT_PORT",
    Port3389NonRdpHits > 0, "HUNT_NON_RDP_PROCESS_EXTERNAL_PORT_3389",
    "HUNT_DEFAULT_COMMODITY_RAT_BUILDER_PORT"
)
| project NetFirstSeen, NetLastSeen, ProcTime, DeviceName, AlertAction, ListedIpHits, InitiatingProcessFileName,
InitiatingProcessFolderPath, ProcFile, ProcessCommandLine, SampleEndpoints, DistinctRemoteIps, ConnectionCount
| sort by ListedIpHits desc, NetLastSeen desc
```

sigma_commodity_dotnet_rat_tls_and_port_anomalies.yml

SIGMA

```
title: Non-RDP Windows Process Initiating External TCP 3389 Connection
id: 7a4319d0-8e14-4c62-9b11-004300000002
status: experimental
description: Detects Windows processes other than native Remote Desktop clients initiating outbound external TCP
3389 connections, matching AsyncRAT and DcRAT port-3389 TLS camouflage.
author: Odin by Brandefense
date: 2026-11-20
modified: 2026-11-20
references:
  - https://attack.mitre.org/techniques/T1571/
  - https://attack.mitre.org/software/S1087/
logsource:
  category: network_connection
  product: windows
detection:
  selection:
    Initiated: 'true'
    DestinationPort: 3389
  filter_private_dest:
    DestinationIsIpv6: 'false'
    DestinationIp|cidr:
      - '10.0.0.0/8'
      - '172.16.0.0/12'
      - '192.168.0.0/16'
      - '127.0.0.0/8'
  filter_approved_rdp_clients:
```

```
    Image:
      - 'C:\Windows\System32\mstsc.exe'
      - 'C:\Windows\SysWOW64\mstsc.exe'
      - 'C:\Program Files\Remote Desktop\msrdcw.exe'
  filter_approved_jump_hosts:
    ComputerName:
      - 'CORP-RDP-GW-01'
      - 'CORP-PAM-BASTION-01'
  condition: selection and not 1 of filter_*
falsepositives:
  - Third-party remote administration consoles approved by exact executable path
level: high
tags:
  - attack.command-and-control
  - attack.t1571
---
title: Windows Scripting or .NET Host Connecting to Default Commodity RAT Ports
id: 7a4319d0-8e14-4c62-9b11-004300000003
status: experimental
description: Detects Windows .NET compilation, injection, or scripting utilities initiating outbound connections
to default Remcos (2404), VenomRAT (4449), Quasar (4782), AsyncRAT (6606, 7707, 8808), or DcRAT (8848) builder
ports.
author: Odin by Brandefense
date: 2026-11-20
modified: 2026-11-20
references:
  - https://attack.mitre.org/software/S1087/
  - https://attack.mitre.org/software/S0332/
logsource:
  category: network_connection
  product: windows
detection:
  selection_proc:
    Image:
      - 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe'
      - 'C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe'
      - 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe'
      - 'C:\Windows\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe'
      - 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe'
      - 'C:\Windows\Microsoft.NET\Framework64\v4.0.30319\RegAsm.exe'
      - 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe'
      - 'C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe'
      - 'C:\Windows\System32\wscript.exe'
      - 'C:\Windows\System32\mshta.exe'
  selection_net:
    Initiated: 'true'
    DestinationPort:
      - 2404
      - 4449
      - 4782
      - 4783
      - 6606
      - 7707
      - 8808
      - 8848
  filter_approved_lab:
    ComputerName:
      - 'REDTEAM-WIN-SIM-01'
      - 'PURPLE-RAT-LAB-01'
  condition: selection_proc and selection_net and not filter_approved_lab
falsepositives:
  - Authorized purple-team simulation hosts listed in filter_approved_lab
level: high
tags:
  - attack.command-and-control
  - attack.t1095
  - attack.t1571
```

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/commodity-rat-fleets/
Short Read · brandefense.io/odin-research/blog/default-rat-certificates/
Defense Kit · brandefense.io/odin-research/defense-kit/