ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 45 / 30 · POST-EXPLOITATION

Loader and .NET C2 Infrastructure
Zero Confirmed Relays

How 2,385 loader and C2 endpoints yield five unconfirmed relay candidates alongside an exposed Covenant and sRDI workbench

Prepared November 2026
Full Report · Public edition · Version 1.1

836 / 799
Post-exploitation C2 endpoints on 799 hosts (Covenant, ValleyRAT, PoshC2, BYOB, NimPlant)

5 / 0
Dedicated cloud and VPS relay candidates (3 Pikabot and 2 PoshC2; 0 independently confirmed or published)

0
Shared hosts between the loader group, the post-exploitation group, and the Japanese workbench

3
Published indicators of compromise

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

How 2,385 loader and C2 endpoints yield five unconfirmed relay candidates alongside an exposed Covenant and sRDI workbench

01	Executive Summary	04
02	Two Stages of Intrusion Infrastructure in One Analytical Model	07
03	Five Initial-Access Loader Feeds and Three Pikabot Relay Candidates	10
04	Post-Exploitation .NET, PowerShell, Nim, and Python C2 Fleets	13
05	Inside an Open Covenant Credential-Model and sRDI Directory in Japan	16
06	Feed Labels Against Unconfirmed Relay Candidates	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 (3)	35
B	Detection rules (4 files)	36

IN SHORT

In a historical feed review completed 20 November 2026, Odin by Brandefense examined 2,385 endpoints across 2,294 distinct hosts spanning five initial-access loaders and five post-exploitation command-and-control frameworks. We also reviewed an open web directory in Japan reporting 2,025 files that exposed Covenant credential models and sRDI shellcode converters. Five dedicated cloud and virtual private server relay candidates lacked independent confirmation, so all 2,385 addresses are withheld. Withhold unverified relay blocks, hunt default PoshC2 and Covenant TLS certificates, and monitor in-memory .NET assembly loads.

01

CHAPTER 01

Executive Summary

IN SHORT

In a historical feed review completed 20 November 2026, Odin by Brandefense examined 2,385 endpoints across 2,294 distinct hosts spanning five initial-access loaders and five post-exploitation command-and-control frameworks. We also reviewed an open web directory in Japan reporting 2,025 files that exposed Covenant credential models and sRDI shellcode converters. Five dedicated cloud and virtual private server relay candidates lacked independent confirmation, so all 2,385 addresses are withheld. Withhold unverified relay blocks, hunt default PoshC2 and Covenant TLS certificates, and monitor in-memory .NET assembly loads.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Enterprise intrusions rarely use one implant from initial delivery through Active Directory escalation. Commodity malware loaders take the early risk on user workstations before operators bring in post-exploitation frameworks built in C#, PowerShell, Python, or Nim.

None of the 1,495 loader hosts in this study is shared with the 799 post-exploitation hosts or the Japanese open directory. We therefore present this two-stage handoff strictly as an analytical model, and we withhold all 2,385 unconfirmed endpoints and every workbench file hash.

1. **Five initial-access loader feeds span 1,549 endpoints with zero confirmed relays.**
   Pikabot leads the loader set with 721 endpoints on 708 hosts, followed by Amadey (509), Latrodectus (225), and Matanbuchus (85). Nine more suspected endpoints on 5 hosts carry the upstream Coruna feed label.

Five Pikabot records carried a confirmed label in upstream feeds, yet two Italian hosts sat on standard FTP port 21. Three Pikabot relay candidates on Amazon and Microsoft cloud networks lacked an independent second signal, so their addresses stay withheld.

2. **Five post-exploitation C2 families add 836 endpoints with two PoshC2 relay candidates.**
   Covenant accounts for 340 endpoints on 334 hosts, with 92.1% on default port 7443, alongside ValleyRAT (265), PoshC2 (124), BYOB (90), and NimPlant (17). Thirteen PoshC2 servers still present the framework's built-in certificate name `CN=P18055077`, including two unconfirmed relay candidates in Germany and Australia. Fourteen Covenant listeners keep `CN=Covenant`, and 11 of 17 NimPlant listeners use HTTP port 80.

3. **An open directory in Japan exposes Covenant credential models and sRDI scripts.**
   Reporting 2,025 files on HTTP port 80 across 308 sampled entries, the server exposes Covenant C# classes for Kerberos tickets, NTLM hashes, passwords, and custom HTTP listener headers. Nearby folders hold SharpSploit credential tasks, embedded Mimikatz and Donut libraries, and 14 sRDI reflective DLL shellcode scripts next to Microsoft .NET SDK archives and synthetic data-loss test files.

Three Decisions for Leaders

1. **Withhold unverified relay blocks and keep noisy loader feeds out of firewalls.** Require an independent confirmation signal before blocking the five Pikabot and PoshC2 relay candidates, and send shared web-hosting, FTP, DNS, and residential telecom records to hunting.
2. **Hunt default C2 certificates and watch in-memory .NET assembly loads.** Alert on `CN=P18055077` and `CN=Covenant` server certificates, and turn on Windows AMSI and .NET Common Language Runtime logging to catch reflective DLL or C# loads from PowerShell and Nim processes.
3. **Harden Kerberos tickets and lock down internal testing directories.** Protect privileged service accounts against Kerberoasting and ticket extraction, and ban unauthenticated HTTP file servers on hosts that store red-team tools or data-loss test suites.

The Study in Numbers

Number	What it is
2,385 / 2,294	Examined endpoints on 2,294 distinct hosts across 10 loader and post-exploitation families
1,549 / 1,495	Initial-access loader endpoints on 1,495 hosts (Pikabot, Amadey, Latrodectus, Matanbuchus, Coruna)
836 / 799	Post-exploitation C2 endpoints on 799 hosts (Covenant, ValleyRAT, PoshC2, BYOB, NimPlant)
5 / 0	Dedicated cloud and VPS relay candidates (3 Pikabot and 2 PoshC2; 0 independently confirmed or published)
2,385 / 2,294	Unconfirmed endpoints on 2,294 hosts kept out of block lists (all addresses withheld)
2,025 / 308	Reported and sampled files on the Japanese Covenant and sRDI HTTP workbench (0 hashes published)
0	Shared hosts between the loader group, the post-exploitation group, and the Japanese workbench

How to use this report.

Security leaders can focus on Chapters 1, 6, and 7. Active Directory, Windows endpoint, and cloud engineers should read Chapters 3, 4, 5, and 7. SOC analysts and detection engineers can go straight to Chapters 8 and 9. Analysts using cyber threat intelligence will find the hosting breakdown and study limits in Chapters 2, 3, 4, 6, and 10.

02

CHAPTER 02

Two Stages of Intrusion Infrastructure in One Analytical Model

IN SHORT

None of the 1,495 loader hosts shares an IPv4 address with the 799 post-exploitation C2 hosts or the Japanese open directory. We present them together as an analytical model of how modern intrusions split delivery from post-exploitation, and we withhold block-list guidance until relay candidates are independently confirmed.

INSIDE THIS CHAPTER

Where the 2,385 endpoints are hosted 08

02 Two Stages of Intrusion Infrastructure in One Analytical Model

Modern Windows intrusions divide work across two separate layers of infrastructure. Phishing and malvertising campaigns rely on initial-access loaders such as Pikabot, Latrodectus, Amadey, and Matanbuchus. Those loaders survive fast takedowns on disposable cloud instances or hijacked web servers. When a target host matches an operator's interest, interactive control shifts to post-exploitation frameworks in .NET, PowerShell, Nim, or Python, including Covenant, PoshC2, NimPlant, BYOB, and ValleyRAT.

Treating every feed record across both layers as a firewall block breaks shared websites, DNS resolvers, FTP servers, and broadband users. None of the 1,549 loader endpoints shares a host with the 836 post-exploitation endpoints or the Japanese workbench. Inside the post-exploitation set alone, three unconfirmed hosts carry two framework labels. One German host pairs Covenant port 7443 with PoshC2 port 443, and two US and UK hosts pair Covenant port 7443 with BYOB port 5000.

Because no host links the loader group to the post-exploitation group, the connection between them is functional rather than a single attributed operator. Figure 1 maps the four groups in this analytical model and shows what defenders can do with each one.

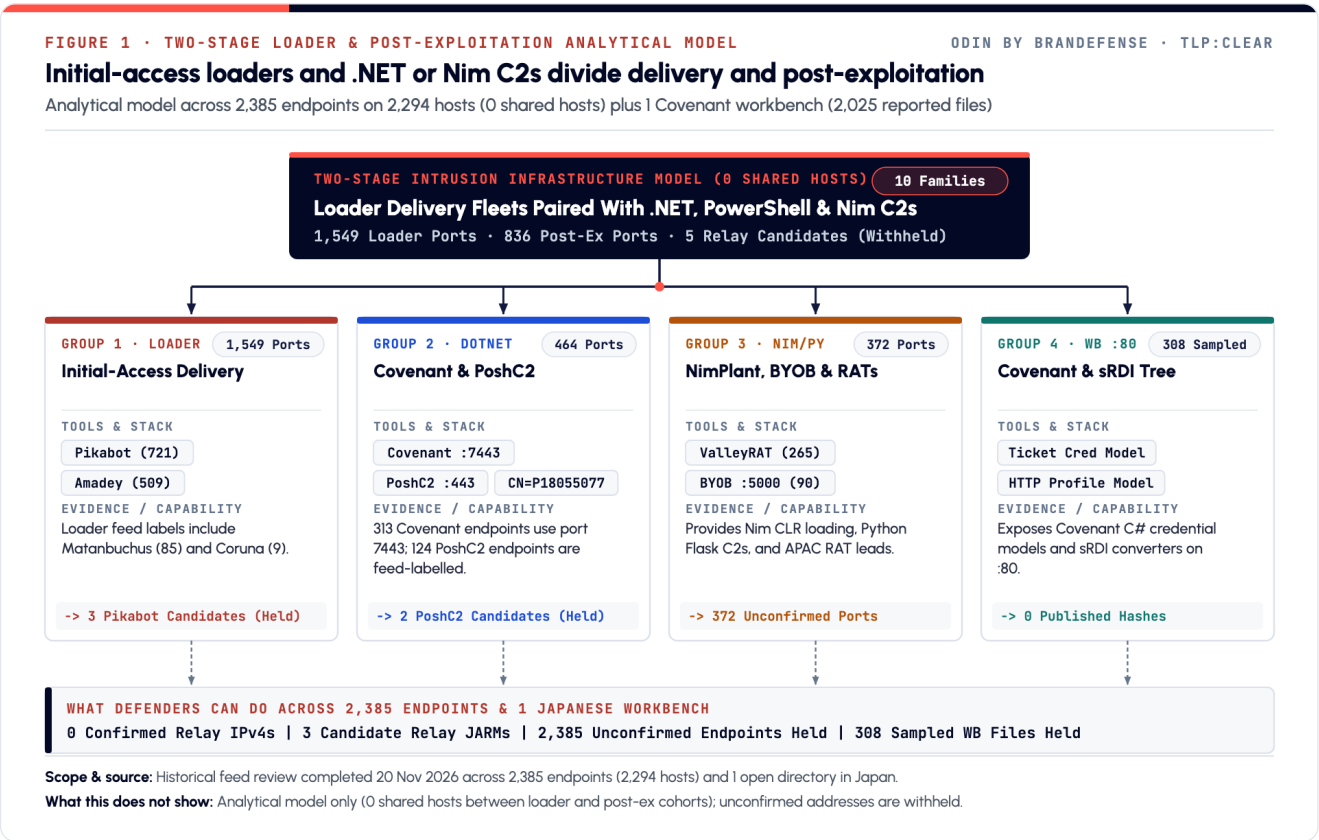


Figure 1: Initial-access loaders and .NET or Nim C2s divide delivery and post-exploitation

Study group	Feed or server records	Distinct hosts	Published addresses	Recommended defender action
Initial-access loaders (5 families)	1,549 endpoints	1,495	0	Withhold 3 Pikabot relay candidates; hunt TLS and HTTP
Post-exploitation C2s (5 families)	836 endpoints	799	0	Withhold 2 PoshC2 relay candidates; hunt default certs
Japanese Covenant and sRDI workbench	2,025 reported files (308 sampled)	1	0	Hunt class and function markers; harden Kerberos and CLR

Where the 2,385 endpoints are hosted

Hosting choices differ sharply between mass-delivery loader leads and operator-managed C2 servers. The United States (623, 26.1%), unattributed codes (435, 18.2%), Hong Kong (188, 7.9%), and Germany (185, 7.8%) lead the corpus. The Netherlands (182, 7.6%), Russia (136, 5.7%), and the United Kingdom (97, 4.1%) bring those seven groups to 77.4% of the 2,385 endpoints.

Unconfirmed Pikabot records sit heavily on domain registrars and shared web-hosting providers, where external attack surface management can help defenders check abused folders before blocking shared IPs. By contrast, Latrodectus candidates cluster on three European hosting providers, Covenant and PoshC2 favor major cloud platforms, and ValleyRAT clusters in Hong Kong.

Hosting provider	Endpoints	Primary families on this network
Railnet LLC	149	Latrodectus (141 on TCP 443), Amadey
Network Solutions, LLC	115	Pikabot shared-hosting candidates
Amazon.com, Inc.	113	Covenant (46), PoshC2 (30, 1 relay candidate), Pikabot (16, 2 relay candidates), BYOB (14)
Cloudflare, Inc.	79	Pikabot (73), Amadey (6) reverse-proxied origins
GoDaddy.com, LLC	76	Pikabot (75) shared-hosting candidates
DigitalOcean, LLC	73	Covenant (48), PoshC2 (18), BYOB (6)
Namecheap, Inc.	44	Pikabot (41), NimPlant (1)
Unified Layer	41	Pikabot (41) shared-hosting candidates
dus.net GmbH	37	Latrodectus (37 on TCP 443)
CTG Server Limited	36	ValleyRAT (36 in Hong Kong)
Hetzner Online GmbH	35	Pikabot (18), PoshC2 (3, 1 relay candidate), Covenant
Akamai Connected Cloud	34	Covenant (13), BYOB (11), PoshC2 (5)

Fifteen additional endpoints sit on institutional networks whose names are withheld, and 15 Amadey candidate rows sit on an individual registrant allocation whose name is also withheld.

CHAPTER 03

Five Initial-Access Loader Feeds and Three Pikabot Relay Candidates

IN SHORT

Five loader feeds account for 1,549 endpoints across 1,495 hosts, or 64.9% of the dataset. Three dedicated Pikabot relay candidates on Amazon and Microsoft cloud networks lacked an independent confirmation signal, while two other feed-confirmed entries hit FTP port 21. Withhold all 1,549 loader addresses from block lists and route them to TLS and proxy hunting.

INSIDE THIS CHAPTER

Pikabot: three unconfirmed cloud relay candidates and 718 other records 10

Amadey, Latrodectus, Matanbuchus, and the Coruna feed label 11

03 Five Initial-Access Loader Feeds and Three Pikabot Relay Candidates

Initial-access loaders act as delivery brokers. They start on a user workstation through an email attachment, script, or malvertising archive. Next, they check the local environment and pull down a second-stage payload over HTTP or HTTPS (T1071.001).

Because threat feeds mix dedicated loader C2 servers with hijacked websites that briefly hosted a stage-one payload, blocking an unverified loader feed causes immediate collateral damage. Figure 2 links each loader, post-exploitation, and workbench group to the detection rules in Chapter 8.

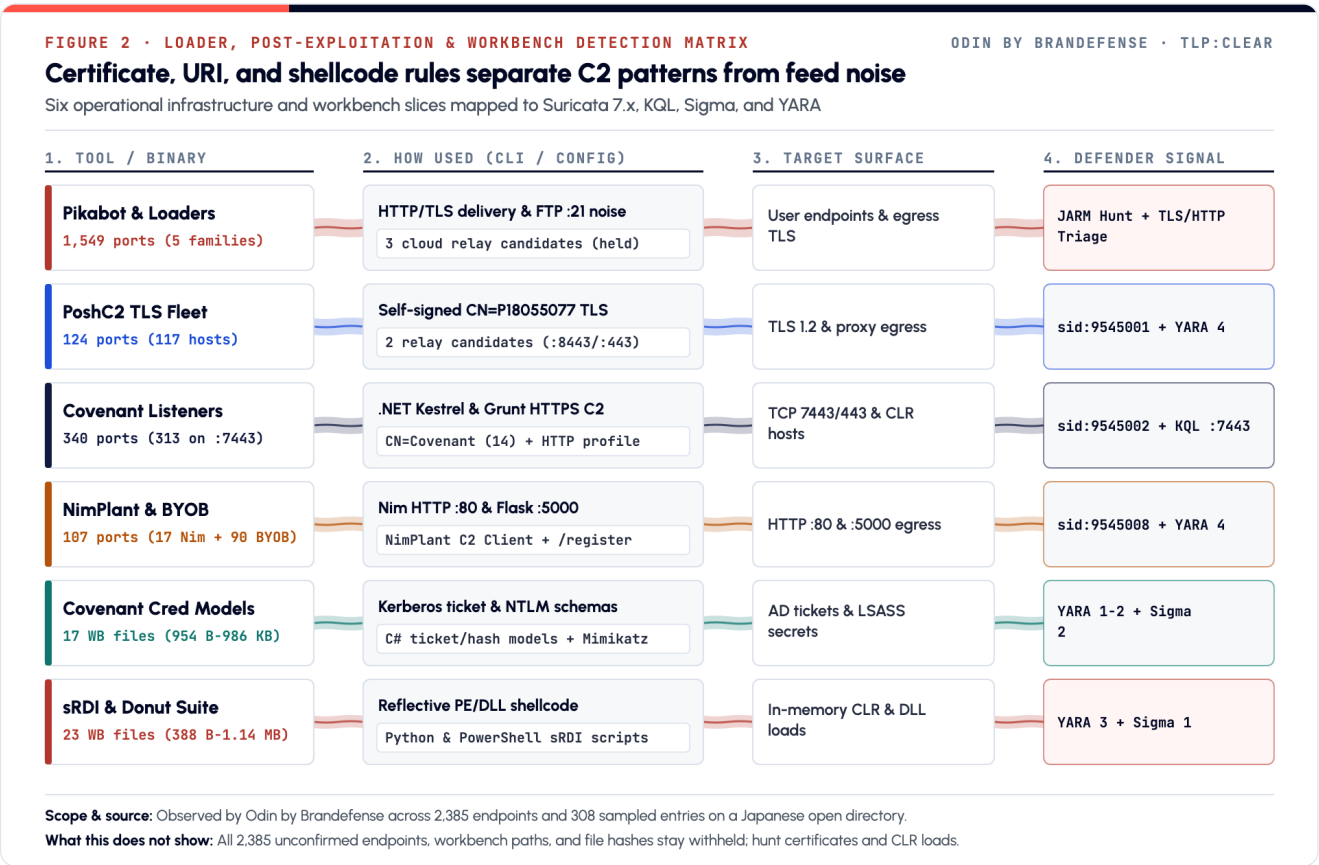


Figure 2: Certificate, URI, and shellcode rules separate C2 patterns from feed noise

Pikabot: three unconfirmed cloud relay candidates and 718 other records

Pikabot (MITRE ATT&CK S1145) is the largest single family in the study, with 721 endpoints across 708 hosts in 44 countries. TCP 443 accounts for 661 endpoints (91.7%). Other ports are 80 (13), 1194 (9), 2083 (8), and 2222 (7). Ports 2224 and 5000 each have four; port 13783 has three.

Five Pikabot endpoints carried a confirmed label in upstream feed records, yet two did not hold up under inspection. Both sat on standard FTP port 21 at an Italian domain and web-hosting registrar, which points to abused shared FTP accounts rather than dedicated Pikabot C2 listeners.

Three Pikabot endpoints observed on 19 November 2026 sit on commercial cloud instances without shared-hosting panels and share one withheld TLS certificate domain. The package does not document an independent protocol response, beacon, or workbench link for those three relay candidates, so their addresses stay withheld:

Family	Relay candidate and port	Country and provider	Transport and JARM	Defender action
Pikabot	Withheld IPv4 (port 80)	Germany, Amazon.com, Inc.	Plain HTTP relay candidate	Withhold block until verified
Pikabot	Withheld IPv4 (port 443)	Germany, Amazon.com, Inc.	HTTPS TLS relay candidate	Withhold block until verified
Pikabot	Withheld IPv4 (port 443)	Ireland, Microsoft Corporation	HTTPS, JARM in Chapter 9	Hunt JARM; withhold block

These cloud records lack an independent second confirmation signal, and elastic IPv4 leases on AWS and Microsoft Azure rotate quickly. Defenders should therefore hunt their TLS fingerprint rather than blocking the addresses.

The other 718 Pikabot endpoints stay unpublished. They comprise 11 suspected endpoints, 2 FTP port 21 collisions, 18 telecom ISP addresses, 11 shared-certificate collisions, 1 open-directory collision, and 675 single-signal candidates. Within those candidates, 149 endpoints share one default Apache and cPanel TLS JARM fingerprint on shared web-hosting servers, showing why JARM values on web ports must never drive automatic blocks.

Amadey, Latrodectus, Matanbuchus, and the Coruna feed label

Four other loader families make up the remaining 828 initial-access endpoints across 787 hosts, with zero confirmed block-list addresses:

- **Amadey (509 endpoints on 490 hosts, S1025).** Unlike TLS-heavy loaders, 485 of 509 Amadey candidates (95.3%) use HTTP port 80 (24 use 443) for PHP panel check-ins across 21 attributed countries and 343 unattributed records. Russia leads with 69 endpoints, followed by the Netherlands (20), Germany (20), Finland (19), and the United States (12), with 2 shared-certificate rows and 507 candidates.
- **Latrodectus (225 endpoints on 225 hosts, S1160).** Every Latrodectus candidate (100.0%) uses TCP 443 across the Netherlands (110), Germany (68), the United Kingdom (35), unattributed records (8), and the United States (4). Railnet LLC (141, 62.7%), dus.net GmbH (37, 16.4%), and metaspinner net GmbH (22, 9.8%) host 200 of 225 servers (88.9%), alongside 14 institutional rows and 6 shared-certificate rows.
- **Matanbuchus (85 endpoints on 67 hosts).** Matanbuchus combines ports 443 (46) and 80 (28) with high ports 59619 (4), 4433 (3), 4443 (2), and 63994 (2) across unattributed networks (51), Russia (28), I-SERVERS LTD (17), and M247 Europe SRL (9).
- **Coruna feed label (9 suspected endpoints on 5 hosts).** Coruna is an upstream feed label rather than a verified malware family or actor. Observed on 13 and 19 November 2026 across Hong Kong (4 endpoints), Singapore (4), and the United States (1), four of its five hosts use paired ports. Two Singapore hosts on Antbox Networks Limited listen on 7001 and 7443 with the same JARM. One Hong Kong host on cognetcloud INC listens on 7000 and 7001, a second on NetLab Global listens on 80 and 443, and one US host on HostPapa listens on 7000. All five addresses are withheld, and Chapter 6 reconciles all five loader families.

CHAPTER 04

Post-Exploitation .NET, PowerShell, Nim, and Python C2 Fleets

IN SHORT

Five post-exploitation families account for 836 endpoints across 799 hosts. Two dedicated PoshC2 relay candidates present the built-in `CN=P18055077` certificate but lack an independent confirmation signal, so all 836 addresses are withheld. Hunt `CN=P18055077` and `CN=Covenant` certificates alongside 313 Covenant port 7443, 78 BYOB port 5000, and 11 NimPlant port 80 listeners.

INSIDE THIS CHAPTER

Covenant on port 7443 and two PoshC2 relay candidates with `CN=P18055077` 13

NimPlant on HTTP port 80, BYOB on port 5000, and ValleyRAT in Hong Kong 14

04 Post-Exploitation .NET, PowerShell, Nim, and Python C2 Fleets

Frameworks built on the .NET Common Language Runtime (CLR), PowerShell's `System.Management.Automation` engine, or Nim with inline CLR hosting let operators load post-exploitation assemblies straight from memory (T1620, T1059.001). Because Covenant, PoshC2, NimPlant, and BYOB are public open-source repositories used by internal red teams every day, single-signal port matches must stay out of automatic block lists.

Figure 3 shows how initial loader delivery, sRDI shellcode conversion, NimPlant or PoshC2 profiles, and Covenant ticket models fit together in our analytical capability model.

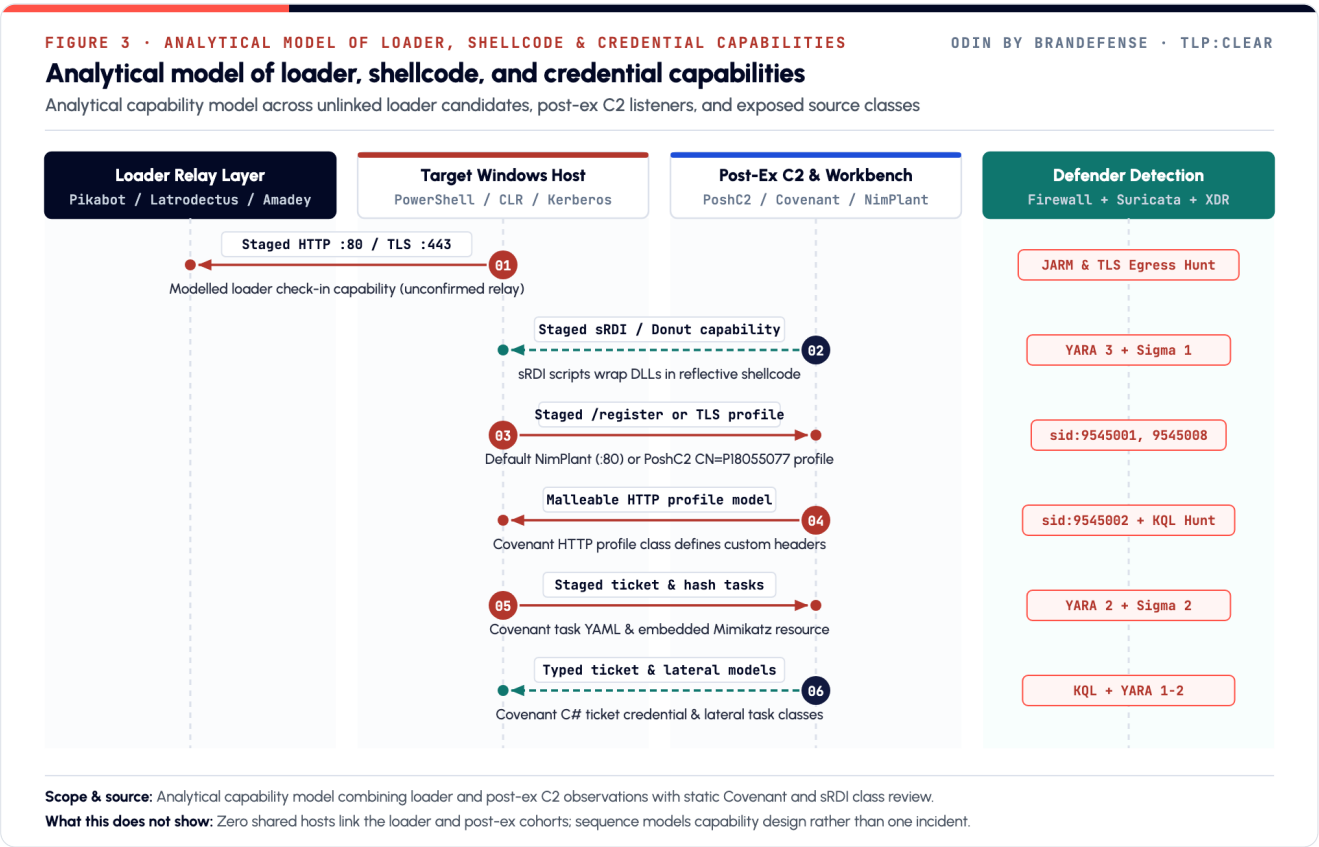


Figure 3: Analytical model of loader, shellcode, and credential capabilities

Covenant on port 7443 and two PoshC2 relay candidates with `CN=P18055077`

Covenant (S1155) and PoshC2 (S0378) make up 464 of the 836 post-exploitation endpoints (55.5%).

Covenant accounts for 340 endpoints on 334 hosts across 38 countries. Of those, 313 endpoints (92.1%) listen on TCP 7443, Covenant's default Kestrel HTTPS management port. The rest use ports 443 (12), 8443 (5), 27443 (2), 9443 (2), and 32768 (2).

Fourteen listeners still present Covenant's default `CN=Covenant` certificate. Eighteen listeners share one Kestrel TLS JARM fingerprint, including all four suspected port 443 servers observed on 17 and 19 November 2026 on Amazon and Microsoft cloud networks. Port 7443 portals are common in test labs, and 17 candidates hit telecom ISP (10), shared ingress certificate (6), or open-directory (1) collisions. All 340 Covenant addresses are therefore withheld.

PoshC2 contributes 124 endpoints on 117 hosts across 21 countries, with 98 (79.0%) on its default `BindPort: 443`, plus 5 on 8443 and 5 on 80. Thirteen PoshC2 servers (10.5%) present the certificate Common Name `CN=P18055077`. Nettitude's open-source PoshC2 codebase generates a 2,048-bit RSA self-signed certificate on first run. Its hardcoded subject fields are `C=US`, `ST=Minnesota`, `L=Minnetonka`, `O=Pajfds`, `OU=Jethpro`, and `CN=P18055077`.

Two dedicated PoshC2 relay candidates presenting `CN=P18055077` lack an independent protocol response or beacon in the package, so their addresses stay withheld:

Family	Relay candidate and port	Country and provider	Certificate and JARM	Defender action
PoshC2	Withheld IPv4 (port 8443)	Germany, Hetzner Online GmbH	`CN=P18055077`, JARM in Chapter 9	Hunt cert and JARM; do not block
PoshC2	Withheld IPv4 (port 443)	Australia, Amazon.com, Inc.	`CN=P18055077`, JARM in Chapter 9	Hunt cert and JARM; do not block

One candidate was observed on 19 November 2026 on Hetzner (port 8443), and the second from 13 to 19 November 2026 on AWS Sydney (port 443). Both belong in certificate and JARM hunts until each address is independently verified. The other 122 PoshC2 records (3 telecom ISP, 2 Kubernetes fake-certificate collisions, and 117 single-signal candidates) also stay withheld.

NimPlant on HTTP port 80, BYOB on port 5000, and ValleyRAT in Hong Kong

Compiled Nim implants and Python or C++ frameworks cover the remaining 372 post-exploitation endpoints across 349 hosts:

- **NimPlant (17 endpoints on 17 hosts).** Written in Nim by Cas van Cooten, NimPlant supports in-memory `execute-assembly`, `powershell`, and `shinject` commands under `riskyMode = true`. Its default configuration uses `type = "HTTP"`, `port = 80`, `userAgent = "NimPlant C2 Client"`, and `/register`, `/task`, and `/result`. Eleven of 17 endpoints (64.7%) listen on HTTP port 80, alongside 443 (3) and single ports on 8080, 8443, and 8880. They span the United States (5), Hong Kong (3), Australia (2), Pakistan (2), and five single-host countries. One port 80 server on Microsoft Azure (19 November 2026) is suspected, two hit shared certificates, one uses a telecom ISP, and 13 are candidates.
- **BYOB (90 endpoints on 87 hosts).** Build Your Own Botnet binds its Python Flask web app on TCP 5000 by default, accounting for 78 of 90 endpoints (86.7%), plus 443 (7) and 80 (2). Five use dynamic telecom ISPs and 85 are candidates.
- **ValleyRAT (265 endpoints on 247 hosts).** ValleyRAT clusters in Asia-Pacific, with 176 of 265 endpoints (66.4%) in Hong Kong, followed by the United States (24), South Korea (16), Singapore (16), China (9), and Japan (9). Common ports include 80 (44), 6666 (24), 8888 (17), 90 (10), and 288 (9). Six records hit standard DNS port 53, 5 hit Cloudflare Origin certificates, 3 sit on telecom ISPs, and 1 co-hosts an open directory, leaving all 265 withheld (see Chapter 6).

CHAPTER 05

Inside an Open Covenant Credential-Model and sRDI Directory in Japan

IN SHORT

An open HTTP directory on port 80 in Japan reported 2,025 exposed files, with 308 sampled entries and 220 retained file paths totaling 3.09 GB. Static review separates 68 Covenant C# credential-model, HTTP-profile, and sRDI shellcode files from 152 .NET SDK archives, runtime DLLs, and synthetic data-loss test blobs. Lock down testing directories and harden Kerberos ticket hygiene.

INSIDE THIS CHAPTER

How Covenant structures harvested Kerberos tickets, NTLM hashes, and HTTP profiles 16

Reflective DLL shellcode (`sRDI`), Donut, ConfuserEx, and why zero hashes are published 17

05 Inside an Open Covenant Credential-Model and sRDI Directory in Japan

External port scans show where listeners sit, whereas an exposed directory shows how a .NET C2 server models stolen credentials, edits HTTP headers, and prepares shellcode. In our historical feed review completed 20 November 2026, one server in Japan on TCP port 80 exposes a Covenant and sRDI workbench. It sits beside a 7,574-byte Python file (path withheld).

Seventy retained rows were first listed from 3 to 12 August 2026; 150 were first listed on 18 November 2026. Across 25 directory records reporting 2,025 files, 308 entries were sampled and 220 path records were retained (88 residual entries were omitted at source). Figure 4 breaks down all five layers on that server.



Figure 4: Open workbench pairs Covenant ticket models with sRDI and .NET SDKs

How Covenant structures harvested Kerberos tickets, NTLM hashes, and HTTP profiles

Inside the Covenant source tree (113 entries total; paths withheld), 17 files (7.7% of the 220 retained rows) define Covenant's credential models and Mimikatz integration. Another 28 files (12.7%) define HTTP profiles and Grunt management.

Rather than saving plain text logs, Covenant parses stolen Windows secrets into typed C# model classes served by an 11,850-byte credential API controller (paths withheld). Six Blazor UI components render those credential tables, led by an 8,869-byte credential form, a 2,880-byte ticket table, and a 2,485-byte hash table. Three smaller Blazor components (2,121, 1,295, and 954 bytes; paths withheld) handle password tables and credential creation:

Exposed Covenant file class (path withheld)	Size (bytes)	Technical role in .NET post-exploitation
Kerberos ticket credential model (C# source)	2,787	Stores stolen Kerberos TGT and TGS tickets, service principal names, and `.kirbi` byte strings (T1558)
NTLM hash credential model (C# source)	2,541	Indexes NTLM, LM, and NetNTLMv1/v2 password hashes by hash type for pass-the-hash reuse
Password credential model (C# source)	2,131	Stores plaintext domain or local account passwords tied to realm and username
Base credential model and credential-type enum (C# source)	1,882 and 1,787	Base credential class and enum (companion server credential model adds 12,014 bytes)
SharpSploit credential task definition (YAML)	47,957	Defines C# Grunt credential-extraction tasks, paired with a 56,741-byte DPAPI reference document
Embedded 64-bit Mimikatz resource library (DLL)	986,624	Embedded 64-bit Mimikatz binary (32-bit companion adds 936,960 bytes) compiled into Grunt tasks
HTTP profile header model (C# source)	1,279	Defines custom HTTP request and response headers (companion Blazor editor adds 2,492 bytes)
SharpSploit lateral-movement task definition (YAML)	24,316	Defines WMI, DCOM, PowerShell remoting, and SCM lateral-movement tasks for Grunt implants

Alongside the HTTP profile header model, the server exposes C# source files for Grunt implants (12,813 bytes), bridge listeners (8,364 bytes), a script launcher (8,183 bytes), and embedded resources (1,462 bytes; paths withheld). It also exposes core API and service source files (1,061,292 and 250,630 bytes) and an OpenAPI specification (176,559 bytes; paths withheld). Together they show why port-only or default-header signatures miss customized Covenant listeners once an operator edits HTTP profile headers.

Reflective DLL shellcode (`sRDI`), Donut, ConfuserEx, and why zero hashes are published

Twenty-three retained files (10.5%; paths withheld) supply reflective shellcode conversion and .NET assembly obfuscation (T1620). Fourteen files belong to Nick Landers' open-source sRDI suite. They include two Python shellcode converter scripts (29,707 and 2,445 bytes) and a PowerShell shellcode converter (40,563 bytes; paths withheld). The suite also holds PowerShell shellcode loader, output, and PE-header helper scripts (23,485, 940, and 47,376 bytes; paths withheld).

Nine companion reference libraries (paths withheld) add a Donut .NET assembly-to-shellcode DLL (50,688 bytes) alongside .NET metadata and PE parser libraries (1,144,320 and 152,576 bytes). That group also holds six ConfuserEx obfuscator libraries, led by renamer and core modules (313,344 and 178,688 bytes; paths withheld).

The remaining 152 retained rows (69.1%) are ordinary build chains and synthetic test files. Seventy entries (31.8%; paths withheld) include 11 Microsoft `.NET` SDK archives (.NET 5.0 through .NET 10.0 preview, up to 308,891,280 bytes). That group also holds 36 standard .NET 3.5 and 4.0 reference assemblies and 23 web UI bundle files.

Another 82 entries (37.3%; paths withheld) combine 8 standalone 7-Zip archiver files and 2 WinDbg debugger archives (192.5 MB each). They sit beside synthetic Base64 and ROT13 data-loss test blobs and protocol test images (paths and filenames withheld).

Workbench file class (paths withheld)	Retained rows	Share	Size range (bytes)	What it holds and how we handle it
Covenant credential and ticket models	17	7.7%	954 to 986,624	C# ticket, hash, and password model classes, SharpSploit credential task YAML, and embedded Mimikatz DLLs (hunt class markers)
sRDI and .NET shellcode tooling	23	10.5%	388 to 1,144,320	Python and PowerShell sRDI shellcode converters, Donut shellcode library, and ConfuserEx obfuscator DLLs (hunt CLI and YARA)
Covenant HTTP profile and core source	28	12.7%	133 to 1,061,292	HTTP profile header models, Grunt and bridge listener classes, and core Covenant API source (hunt class markers)
.NET SDKs, runtime DLLs, and web bundles	70	31.8%	6,448 to 308,891,280	11 Microsoft .NET SDK archives, 36 standard .NET reference assemblies, and 23 web UI bundle files (do not block)
Synthetic exfil test blobs and archivers	82	37.3%	1,079 to 192,459,823	Synthetic Base64 and ROT13 test payloads, 7-Zip archivers, WinDbg archives, and protocol test images (do not block)

Because the open directory mixes public Covenant and sRDI source trees with synthetic Base64 and ROT13 test payloads, the host could belong to an enterprise data-loss-prevention or purple-team engineer. We therefore withhold the server's address, its file paths and filenames, and all 308 sampled file hashes.

06

CHAPTER 06

Feed Labels Against Unconfirmed Relay Candidates

IN SHORT

Turning all 2,385 feed-labelled loader and C2 endpoints into firewall blocks would disrupt GoDaddy and Network Solutions shared web hosts, Cloudflare origins, FTP and DNS servers, and residential broadband users. Because even the five dedicated relay candidates (0.21%) lack an independent second signal, withhold all 2,385 addresses from block lists and hunt certificates and runtime behavior instead.

INSIDE THIS CHAPTER

Four other ways to read the evidence 21

06 Feed Labels Against Unconfirmed Relay Candidates

Good infrastructure defense depends as much on what stays out of a block list as on what goes in. Figure 5 compares the total number of examined endpoints per family against the five dedicated relay candidates.

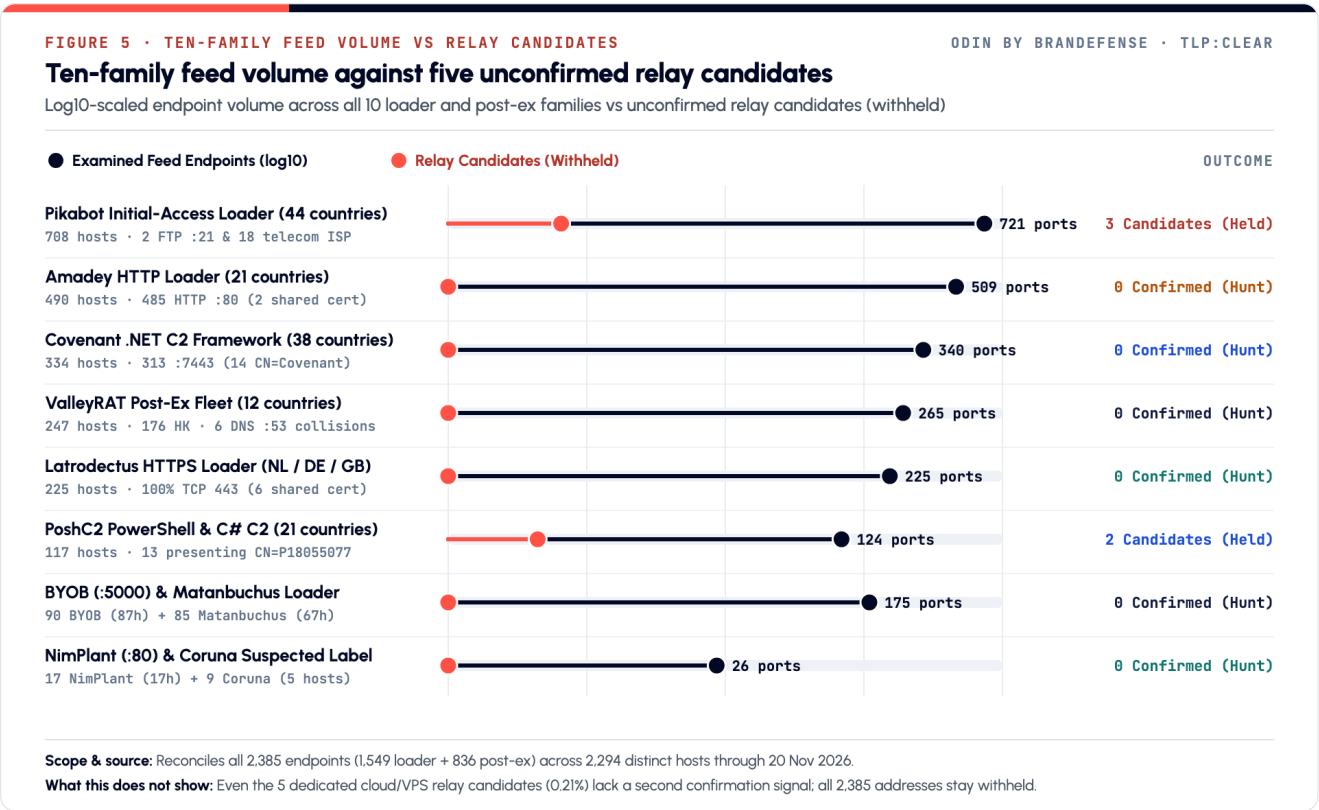


Figure 5: Ten-family feed volume against five unconfirmed relay candidates

Family	Group	Endpoints	Hosts	Relay candidates (withheld)	Other unconfirmed records (hunt only, addresses withheld)
Pikabot	Loader	721	708	3	718 (11 suspected, 2 FTP 21, 18 telecom ISP, 11 shared cert, 1 open dir, 675 candidates)
Amadey	Loader	509	490	0	509 (2 shared cert, 507 HTTP/HTTPS candidates)
Covenant	Post-ex C2	340	334	0	340 (4 suspected, 10 telecom ISP, 6 shared cert, 1 open dir, 319 port 7443/443 candidates)
ValleyRAT	Post-ex C2	265	247	0	265 (6 DNS port 53, 5 shared cert, 3 telecom ISP, 1 open dir, 250 candidates)
Latrodectus	Loader	225	225	0	225 (6 shared cert, 219 TCP 443 candidates)
PoshC2	Post-ex C2	124	117	2	122 (3 telecom ISP, 2 shared cert, 117 candidates)
BYOB	Post-ex C2	90	87	0	90 (5 telecom ISP, 85 port 5000/443 candidates)
Matanbuchus	Loader	85	67	0	85 (1 shared cert, 84 candidates)
NimPlant	Post-ex C2	17	17	0	17 (1 suspected, 2 shared cert, 1 telecom ISP, 13 HTTP/HTTPS candidates)
Coruna (feed label)	Loader	9	5	0	9 suspected across HK, SG, and US
Total	**10 families**	**2,385**	**2,294**	**5 (0 confirmed)**	**2,380 across 2,289 hosts (all 2,385 withheld)**

Four other ways to read the evidence

1. **Authorized testing and DLP validation.** Covenant on port 7443, PoshC2, NimPlant on port 80, and BYOB on port 5000 are common red-team tools. The Japanese open directory also holds synthetic Base64, ROT13, and DNS or protocol test images typical of data-loss tests (paths withheld). Withholding unconfirmed C2 hosts, workbench paths, and all 308 sampled hashes protects testing teams and unmodified GitHub clones.
2. **Hijacked shared web hosting.** Over 345 Pikabot candidates sit on Network Solutions, GoDaddy, Cloudflare, Unified Layer, and Namecheap, where one IPv4 address serves many benign sites beside one abused PHP folder. Even the three dedicated AWS and Azure Pikabot relay candidates stay withheld until an independent signal confirms each address.
3. **Port and shared-certificate collisions.** Two feed-confirmed Pikabot records sat on FTP port 21 in Italy, and six ValleyRAT records sat on DNS port 53. Another 35 endpoints presented `CloudFlare Origin Certificate` (12), `Kubernetes Ingress Controller Fake Certificate` (3), or shared multi-tenant JARMs (20). Never block on a port or JARM alone.
4. **Broadband IP churn and feed labels.** Forty candidates sit on residential or regional telecom ISPs (`Comcast Cable`, `Korea Telecom`, `Telecom Italia`, `Vodafone Italia`, `Orange`, `Vietnam Posts and Telecommunications`), where DHCP leases rotate. Meanwhile, the nine Coruna rows reflect an upstream feed label across five hosts without malware samples.

07

CHAPTER 07

What Defenders Should Do

IN SHORT

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

07 What Defenders Should Do

Network and firewall

- [ ] Withhold outbound firewall blocks on the five unconfirmed Pikabot and PoshC2 relay candidates until an independent protocol response, beacon, or workbench link verifies each address.
- [ ] Inspect outbound TLS 1.2 handshakes and decrypted proxy sessions for default `CN=P18055077` (PoshC2) and `CN=Covenant` server certificates, plus outbound HTTP requests using `NimPlant C2 Client` on `/register`, `/task`, or `/result`.

Windows endpoints and .NET runtime

- [ ] Enable Windows AMSI, PowerShell Script Block Logging (Event ID 4104), and .NET CLR ETW assembly-load logging on workstations and servers to catch in-memory Donut, SharpSploit, Mimikatz, and NimPlant `execute-assembly` loads.
- [ ] Alert on PowerShell or Python processes invoking sRDI reflective DLL shellcode conversion functions (`ConvertTo-Shellcode`, `Invoke-Shellcode`, `Out-Shellcode`, `ShellcodeRDI`, or `SRDI_CLEARHEADER`) outside registered red-team hosts.

Active Directory and Kerberos identity

- [ ] Harden service accounts against Covenant Kerberos ticket and NTLM hash capture workflows (T1558) by requiring 256-bit AES Kerberos encryption and disabling RC4 where supported. Rotate `krbtgt` twice on schedule, put privileged admins in `Protected Users`, and enable Windows Credential Guard.

Cloud platforms and threat intelligence

- [ ] Audit internal red-team, purple-team, and DLP-testing servers to make sure Covenant, sRDI, and exfiltration test folders are never exposed on unauthenticated `0.0.0.0:80` HTTP listeners.
- [ ] Keep shared web-hosting, FTP port 21, DNS port 53, Cloudflare/Kubernetes default certificates, and residential telecom ISP feed hits out of automatic firewall blocks.

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 26

MITRE ATT&CK mapping 26

08 Detection Rules

The full rules are in Appendix B. The Suricata file passes a static syntax lint (`0 problems`). These rules were checked statically. They were not loaded on live Suricata, Sentinel, Sigma, or YARA engines, so test and tune them before production enforcement.

Rule file	What it catches
`odin_covenant_poshc2_nimplant_and_pikabot_loaders.rules`	Suricata 7.x (3 signatures): PoshC2 (`CN=P18055077`) and Covenant (`CN=Covenant`) TLS certificates and NimPlant default HTTP check-ins
`sentinel_postex_dotnet_nim_c2_and_loader_hunt.kql`	Microsoft Sentinel and Defender XDR: estimated distinct `IP:port` pairs on port 7443 or NimPlant URIs and independent sRDI or Covenant process-token hits
`sigma_covenant_poshc2_srdi_and_loader_relays.yml`	Windows Sigma (2 rules): PowerShell or Python sRDI shellcode conversion functions and Covenant or SharpSploit credential class activity
`odin_covenant_credential_models_srdi_and_poshc2.yar`	YARA (4 rules): Covenant C# ticket and hash credential classes, SharpSploit and HTTP profile classes, sRDI converter flags, and NimPlant or PoshC2 configs

How the Suricata signatures work

SID	Looks for	Why
9545001	Server certificate subject `CN=P18055077`	PoshC2's built-in self-signed TLS certificate (`O=Pajfds` , `OU=Jethpro`)
9545002	Server certificate subject `CN=Covenant` on ports 443, 7443, 8443, 9443, or 27443	Covenant's default Kestrel HTTPS listener certificate
9545008	HTTP `User-Agent: NimPlant C2 Client` paired with `/register` , `/task` , or `/result`	NimPlant's default HTTP implant check-in and tasking profile

TLS 1.3 encrypts server certificates on the wire, so SIDs 9545001 and 9545002 trigger on TLS 1.2 handshakes or behind a TLS-decrypting forward proxy. Signatures for the five unconfirmed relay candidates and the Japanese open-directory paths are withheld until independent host verification is established.

How the KQL, Sigma, and YARA rules work

The Sentinel KQL query looks back seven days across `DeviceProcessEvents` and `DeviceNetworkEvents`. It estimates distinct remote IP and port pairs for port 7443 and NimPlant HTTP URI candidates. It reports network hits and post-exploitation command-line hits independently; either signal can appear without the other. Exact host allowlists (`REDTEAM-SIM-WIN-01`, `PURPLE-DOTNET-LAB-01`, `DLP-EXFIL-TEST-SRV-01`) tune lab hosts.

Sigma or YARA rule	Scope	Fires when
Sigma 1: sRDI shellcode converter	High (Windows)	PowerShell or Python runs `ConvertTo-Shellcode`, `Invoke-Shellcode`, `Out-Shellcode`, `ShellcodeRDI`, or `SRDI_CLEARHEADER` outside exact lab hosts
Sigma 2: Covenant or SharpSploit activity	High (Windows)	Command line references `Covenant.API.Models`, `CapturedTicketCredential`, `CapturedHashCredential`, `SharpSploit.Credentials`, or `MimikatzCommand` outside exact lab hosts
`ODIN_Covenant_Captured_Credential_and_Ticket_Model_Source`	YARA (< 256 KB)	Covenant namespace pairs with `Captured*Credential` and ticket or hash fields
`ODIN_Covenant_SharpSploit_Credentials_and_HttpProfile_Config`	YARA (< 512 KB)	`Covenant` pairs with `SharpSploit.*` and `powerkatz_x64` / `MimikatzCommand`, or `HttpProfileHeader`
`ODIN_sRDI_Reflective_DLL_Shellcode_Converter_Suite`	YARA (< 256 KB)	`ConvertTo-Shellcode`, `ShellcodeRDI`, or `ConvertToShellcode` pairs with `SRDI_CLEARHEADER` or function hashing
`ODIN_NimPlant_and_PoshC2_Listener_Config_Indicators`	YARA (< 256 KB)	`NimPlant C2 Client` pairs with `registerPath` / `riskyMode`, or `P18055077` pairs with `Pajfds` and `Jethpro`

MITRE ATT&CK mapping

Tactic	Technique	How it shows up in this study	Detection
Resource Development	T1583.003 Acquire Infrastructure: Virtual Private Server	Cloud/VPS C2 relay candidates and the open directory Covenant/sRDI workbench	SID 9545002; KQL; Sigma rule 2
Execution	T1059.001 Command and Scripting Interpreter: PowerShell	PoshC2 implants and PowerShell sRDI shellcode conversion and loader functions	SID 9545001; KQL; Sigma rule 1; YARA
Defense Evasion	T1620 Reflective Code Loading	sRDI PE-to-shellcode conversion, Donut assembly-to-shellcode library, and NimPlant `execute-assembly` / `shinject`	KQL; Sigma rule 1; YARA
Credential Access	T1558 Steal or Forge Kerberos Tickets	Covenant Kerberos ticket credential models, SharpSploit credential tasks, and embedded Mimikatz DLLs	KQL; Sigma rule 2; YARA
Command and Control	T1071.001 Application Layer Protocol: Web Protocols	Pikabot, PoshC2, Covenant HTTP profile header models, and NimPlant HTTP/HTTPS C2 channels	SIDs 9545001, 9545002, and 9545008; KQL

09

CHAPTER 09

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Relay candidates withheld pending independent confirmation 28

JARM fingerprints to hunt (in Appendix A) 28

Text-only certificate, HTTP, and workbench markers 28

09 Indicators

Appendix A holds 3 published indicators: 3 JARM fingerprints for TLS correlation hunting. All five relay candidate IPv4 addresses and all 308 sampled workbench file hashes are withheld. Fourteen more indicators and generalized file-class descriptions appear in this report only: two default TLS certificate names, one HTTP User-Agent string, and eleven Covenant and sRDI workbench file classes (paths withheld).

Type	Count	Where to use it
Confirmed IPv4 addresses published	0	All five relay candidates are withheld pending independent confirmation
Candidate-relay JARM fingerprints (in Appendix A)	3	TLS hunting and correlation only, never block on a JARM alone
Default TLS certificate names and HTTP User-Agent (text-only)	3	Suricata, Zeek, and proxy hunting
Generalized Covenant and sRDI workbench file classes (text-only)	11	Endpoint command-line hunting and YARA scans (all paths withheld)

Relay candidates withheld pending independent confirmation

Relay candidate and port	Family	Hosting	Note
Withheld IPv4 (port 80)	Pikabot	Germany, Amazon.com, Inc.	Plain HTTP relay candidate; withhold block until verified
Withheld IPv4 (port 443)	Pikabot	Germany, Amazon.com, Inc.	HTTPS relay candidate; withhold block until verified
Withheld IPv4 (port 443)	Pikabot	Ireland, Microsoft Corporation	HTTPS relay candidate; hunt JARM, withhold block
Withheld IPv4 (port 8443)	PoshC2	Germany, Hetzner Online GmbH	HTTPS (`CN=P18055077`); hunt certificate, withhold block
Withheld IPv4 (port 443)	PoshC2	Australia, Amazon.com, Inc.	HTTPS (`CN=P18055077`); hunt certificate, withhold block

JARM fingerprints to hunt (in Appendix A)

JARM	Family	Context
`2ad2ad16d2ad2ad22c2ad2ad2ad2ad47321614530b94a96fa03d06e666d6d6`	Pikabot	Unconfirmed cloud relay candidate in Ireland (port 443, address withheld)
`2ad2ad0002ad2ad00042d42d000000ad9bf51cc3f5a1e29eecb81d0c7b06eb`	PoshC2	Six candidate PoshC2 endpoints including a German VPS relay candidate (addresses withheld)
`2ad2ad0002ad2ad0002ad2ad2ad2adf05f3b95c2927ff44a7871306bfc5465`	PoshC2	Unconfirmed cloud relay candidate in Australia (port 443, address withheld)

Text-only certificate, HTTP, and workbench markers

Indicator	Type	Context and use
`CN=P18055077`	Default TLS certificate name	PoshC2 default self-signed subject seen on 13 endpoints including two relay candidates

Indicator	Type	Context and use
`CN=Covenant`	Default TLS certificate name	Covenant default self-signed subject seen on 14 endpoints
`NimPlant C2 Client`	HTTP User-Agent	NimPlant default implant User-Agent paired with `/register`, `/task`, and `/result`
Covenant base credential model	File class (path withheld)	1,882-byte C# source
Covenant Kerberos ticket credential model	File class (path withheld)	2,787-byte Kerberos ticket model C# source
Covenant NTLM hash credential model	File class (path withheld)	2,541-byte NTLM hash model C# source
Covenant password credential model	File class (path withheld)	2,131-byte password model C# source
Covenant HTTP profile header model	File class (path withheld)	1,279-byte malleable HTTP listener header C# source
Covenant SharpSploit credential task definition	File class (path withheld)	47,957-byte credential-harvesting task YAML
Covenant SharpSploit lateral-movement task definition	File class (path withheld)	24,316-byte lateral-movement task YAML
Covenant embedded 64-bit Mimikatz resource	File class (path withheld)	986,624-byte embedded Mimikatz DLL (hash withheld)
Covenant .NET-to-shellcode reference library	File class (path withheld)	50,688-byte Donut assembly-to-shellcode DLL (hash withheld)
sRDI Python reflective DLL shellcode converter	File class (path withheld)	29,707-byte Python reflective DLL shellcode script
sRDI PowerShell reflective DLL shellcode converter	File class (path withheld)	40,563-byte PowerShell reflective DLL shellcode script

Withheld: all 2,385 unconfirmed loader and post-exploitation endpoints across 2,294 hosts, including five Pikabot and PoshC2 relay candidates. We also withhold the Japanese workbench address and paths, unverified certificate domains, institutional and individual registrant network names, and 308 sampled file hashes.

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

- **Historical feed review.** Findings cover a historical feed review completed 20 November 2026. Earlier dates on 2,346 of 2,385 endpoint records (before 7 November 2026) reflect upstream record metadata, not Odin monitoring duration. Workbench first-listed dates (3 to 12 August 2026 on 70 retained rows and 18 November 2026 on 150 rows) are also upstream directory metadata.
- **Analytical model, not one actor.** Zero shared IPv4 addresses link the 1,495 loader hosts, the 799 post-exploitation hosts, and the Japanese workbench. Figures 1 and 3 show a functional model rather than one campaign.
- **Feed labels are not attribution.** Upstream labels, especially the nine suspected Coruna endpoints across five hosts, reflect feed heuristics without captured malware samples or actor attribution.
- **Dual-use and testing servers.** Covenant, PoshC2, NimPlant, BYOB, and sRDI are used by corporate red teams, and the open directory includes synthetic Base64 and ROT13 files consistent with data-loss validation. All unconfirmed hosts, all workbench file paths, and all 308 sampled workbench hashes stay withheld.
- **Unconfirmed relay candidates and cloud IP rotation.** Even the five dedicated Pikabot and PoshC2 relay candidates lack an independent protocol response, beacon, or workbench link in the package, and four sit on AWS or Microsoft Azure where addresses rotate quickly. Their addresses and block recommendations stay withheld until each address is independently verified.
- **Static rule checks.** Published rules were linted and reviewed statically without replay on live network traffic or endpoint engines.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. MITRE ATT&CK. S1155 Covenant. https://attack.mitre.org/software/S1155/ (verified 21 November 2026)
2. MITRE ATT&CK. S0378 PoshC2. https://attack.mitre.org/software/S0378/ (verified 21 November 2026)
3. Cas van Cooten. NimPlant: lightweight first-stage C2 implant written in Nim. https://github.com/chvancooten/NimPlant (verified 21 November 2026)
4. MITRE ATT&CK. S1145 Pikabot. https://attack.mitre.org/software/S1145/ (verified 21 November 2026)
5. MITRE ATT&CK. S1160 Latrodectus. https://attack.mitre.org/software/S1160/ (verified 21 November 2026)
6. MITRE ATT&CK. S1025 Amadey. https://attack.mitre.org/software/S1025/ (verified 21 November 2026)
7. MITRE ATT&CK. T1558 Steal or Forge Kerberos Tickets. https://attack.mitre.org/techniques/T1558/ (verified 21 November 2026)
8. Salesforce Engineering. JARM: active TLS server fingerprinting. https://github.com/salesforce/jarm (verified 21 November 2026)

APPENDICES

Indicators and Detection Rules

FOR DEFENDERS

Appendix A lists the indicators published for this study. Appendix B carries every published detection rule in full.

INSIDE

A. Indicators of compromise (3) 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
R45-IOC-01	JARM	2ad2ad16d2ad2ad22c2ad2ad2ad2ad47321614530b94a96fa03d06e666d6d6	443	Pikabot · TLS fingerprint observed on an unconfirmed cloud relay candidate in Ireland (address withheld)	HUNT
R45-IOC-02	JARM	2ad2ad0002ad2ad00042d42d000000ad9bf51cc3f5a1e29eecb81d0c7b06eb	8443	PoshC2 · TLS fingerprint observed on six candidate PoshC2 endpoints including a German VPS relay candidate (addresses withheld)	HUNT
R45-IOC-03	JARM	2ad2ad0002ad2ad0002ad2ad2ad2adf05f3b95c2927ff44a7871306bfc5465	443	PoshC2 · TLS fingerprint observed on an unconfirmed cloud relay candidate in Australia (address withheld)	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_covenant_credential_models_srdi_and_poshc2.yar

YARA

```
/*
 * Odin by Brandefense - Monograph 45 / 46 Bounded YARA Suite (Public Edition v1.1)
 * Coverage: Covenant C# credential/ticket models, Covenant SharpSploit & HTTP profile
 *           definitions, sRDI reflective DLL shellcode converters, and NimPlant/PoshC2
 *           listener configuration files.
 * Classification: TLP:CLEAR
 */

rule ODIN_Covenant_Captured_Credential_and_Ticket_Model_Source
{
    meta:
        author = "Odin by Brandefense"
        monograph = "45 / 46"
        description = "Detects Covenant C# credential-harvesting API model classes (CapturedTicketCredential,
CapturedHashCredential, CapturedPasswordCredential)"
        tlp = "CLEAR"
        action = "hunt_credential_model_source"
    strings:
        $ns1 = "namespace Covenant.API.Models" ascii wide
        $ns2 = "namespace Covenant.Models.Covenant" ascii wide
        $cls1 = "CapturedTicketCredential" ascii wide
        $cls2 = "CapturedHashCredential" ascii wide
        $cls3 = "CapturedPasswordCredential" ascii wide
        $fld1 = "ServicePrincipalName" ascii wide
        $fld2 = "HashCredentialType" ascii wide
        $fld3 = "TicketByteString" ascii wide
    condition:
        filesize < 256KB and (1 of ($ns*)) and (1 of ($cls*)) and (1 of ($fld*))
}

rule ODIN_Covenant_SharpSploit_Credentials_and_HttpProfile_Config
{
    meta:
        author = "Odin by Brandefense"
        monograph = "45 / 46"
        description = "Detects Covenant SharpSploit credential/lateral-movement task definitions or malleable
HttpProfileHeader classes"
        tlp = "CLEAR"
        action = "hunt_covenant_profile_or_task"
    strings:
        $task1 = "SharpSploit.Credentials" ascii wide
        $task2 = "SharpSploit.LateralMovement" ascii wide
        $katz1 = "powerkatz_x64" ascii wide
        $katz2 = "MimikatzCommand" ascii wide
        $prof1 = "class HttpProfileHeader" ascii wide
        $prof2 = "EditHttpProfileHeaderList" ascii wide
        $cov = "Covenant" ascii wide
    condition:
        filesize < 512KB and $cov and ((1 of ($task*) and 1 of ($katz*)) or (1 of ($prof*)))
}

rule ODIN_sRDI_Reflective_DLL_Shellcode_Converter_Suite
{
    meta:
        author = "Odin by Brandefense"
        monograph = "45 / 46"
        description = "Detects sRDI Python and PowerShell reflective DLL shellcode converter function and header
flags"
        tlp = "CLEAR"
        action = "hunt_shellcode_converter"
```

```
    strings:
        $fn1 = "ConvertTo-Shellcode" ascii wide
        $fn2 = "ShellcodeRDI" ascii wide
        $fn3 = "ConvertToShellcode" ascii wide
        $flag1 = "SRDI_CLEARHEADER" ascii wide
        $flag2 = "HashFunctionName" ascii wide
        $flag3 = "FunctionName" ascii wide
    condition:
        filesize < 256KB and (1 of ($fn*)) and (1 of ($flag*))
}

rule ODIN_NimPlant_and_PoshC2_Listener_Config_Indicators
{
    meta:
        author = "Odin by Brandefense"
        monograph = "45 / 46"
        description = "Detects NimPlant config.toml listener profiles (NimPlant C2 Client) or PoshC2 default
self-signed certificate generator parameters (CN=P18055077)"
        tlp = "CLEAR"
        action = "hunt_c2_listener_config"
    strings:
        $nim_ua = "NimPlant C2 Client" ascii wide
        $nim_reg = "registerPath = \"/register\"" ascii wide
        $nim_risk = "riskyMode" ascii wide
        $posh_cn = "P18055077" ascii wide
        $posh_o = "Pajfds" ascii wide
        $posh_ou = "Jethpro" ascii wide
    condition:
        filesize < 256KB and (($nim_ua and ($nim_reg or $nim_risk)) or ($posh_cn and $posh_o and $posh_ou))
}
```

odin_covenant_poshc2_nimplant_and_pikabot_loaders.rules

SURICATA

```
# Odin by Brandefense - Monograph 45 / 46 Suricata 7.x Detection Suite (Public Edition v1.1)
# Coverage: PoshC2 (CN=P18055077) and Covenant (CN=Covenant) default TLS server certificates
#           and NimPlant default HTTP C2 implant check-ins (/register, /task, /result)
# Classification: TLP:CLEAR

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R45] PoshC2 Default Self-Signed TLS Server Certificate
(CN=P18055077)"; flow:established,to_client; tls.cert_subject; bsize:>11; content:"CN=P18055077"; fast_pattern;
classtype:trojan-activity; sid:9545001; rev:1; metadata:created_at 2026_11_20, deployment perimeter_and_proxy,
tactic command_and_control;)

alert tls $EXTERNAL_NET [443,7443,8443,9443,27443] -> $HOME_NET any (msg:"ODIN [R45] Covenant .NET C2 Default TLS
Server Certificate (CN=Covenant)"; flow:established,to_client; tls.cert_subject; bsize:>10;
content:"CN=Covenant"; fast_pattern; classtype:trojan-activity; sid:9545002; rev:1; metadata:created_at
2026_11_20, deployment perimeter_and_proxy, tactic command_and_control;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R45] NimPlant Default HTTP C2 User-Agent and Implant
URI"; flow:established,to_server; http.user_agent; bsize:18; content:"NimPlant C2 Client"; fast_pattern;
http.uri; bsize:>4; pcre:"/^\/(register|task|result)(?:\/|$)/"; classtype:trojan-activity; sid:9545008; rev:1;
metadata:created_at 2026_11_20, deployment perimeter_egress, tactic command_and_control;)
```

sentinel_postex_dotnet_nim_c2_and_loader_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 45 / 46 Microsoft Sentinel & Defender XDR Hunt (v1.1)
// Estimates distinct remote IP and port pairs for port 7443 and NimPlant (/register, /task,
// /result) HTTP URI candidates and reports network hits and post-exploitation command-line
// hits (sRDI, Covenant, SharpSploit, PoshC2, NimPlant) independently via fullouter join.
// Classification: TLP:CLEAR

let Lookback = 7d;
let ApprovedRedTeamHosts = dynamic(["REDTEAM-SIM-WIN-01", "PURPLE-DOTNET-LAB-01"]);
let ApprovedDlpValidationHosts = dynamic(["DLP-EXFIL-TEST-SRV-01"]);
```

```
let PostExFrameworkAndShellcodeTokens = dynamic(["SRDI_CLEARHEADER", "ConvertTo-Shellcode", "Invoke-Shellcode",
"ShellcodeRDI", "CapturedTicketCredential", "CapturedHashCredential", "SharpSploit", "MimikatzCommand",
"NimPlant", "PoshC2"]);
let PostExToolProcs = DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedRedTeamHosts)) and not(DeviceName in~ (ApprovedDlpValidationHosts))
    | where FileName in~ ("powershell.exe", "pwsh.exe", "python.exe", "python3", "dotnet.exe", "dotnet")
        and ProcessCommandLine has_any (PostExFrameworkAndShellcodeTokens)
    | project ProcTime = Timestamp, DeviceId, DeviceName, InitiatingProcessFileName, FileName, FolderPath,
ProcessCommandLine;
DeviceNetworkEvents
| where Timestamp >= ago(Lookback)
| where not(DeviceName in~ (ApprovedRedTeamHosts)) and not(DeviceName in~ (ApprovedDlpValidationHosts))
| where isnotempty(RemoteIP) and (
    RemotePort == 7443
    or (RemotePort == 80 and RemoteUrl has_any ("/register", "/task", "/result"))
)
| summarize
    NetFirstSeen = min(Timestamp),
    NetLastSeen = max(Timestamp),
    DistinctRemoteIps = dcount(RemoteIP),
    Port7443PairCount = dcountif(strcat(RemoteIP, ":", tostring(RemotePort)), RemotePort == 7443),
    NimPlantUriPairCount = dcountif(strcat(RemoteIP, ":", tostring(RemotePort)), RemotePort == 80 and RemoteUrl
has_any ("/register", "/task", "/result")),
    SampleEndpoints = make_set(strcat(RemoteIP, ":", tostring(RemotePort)), 12),
    SampleUrls = make_set(RemoteUrl, 10),
    ConnectionCount = count()
    by DeviceId, DeviceName, InitiatingProcessFileName
| join kind=fullouter (PostExToolProcs) on DeviceId, DeviceName
| extend ResolvedDevice = coalesce(DeviceName, DeviceName1)
| where isnotempty(ProcessCommandLine) or Port7443PairCount > 0 or NimPlantUriPairCount > 0
| extend AlertAction = case(
    ProcessCommandLine has_any ("SRDI_CLEARHEADER", "ShellcodeRDI", "ConvertTo-Shellcode", "Invoke-Shellcode"),
"CONTAIN_SRDI_OR_DOTNET_SHELLCODE_CONVERSION",
    ProcessCommandLine has_any ("CapturedTicketCredential", "CapturedHashCredential", "SharpSploit",
"MimikatzCommand"), "HUNT_COVENANT_OR_SHARPSPLOIT_CREDENTIAL_ACTIVITY",
    Port7443PairCount > 0, "HUNT_PORT_7443_OR_POSTEX_C2_ANOMALY",
    "HUNT_NIMPLANT_OR_POSTEX_HTTP_CHECKIN"
)
| project NetFirstSeen, NetLastSeen, ProcTime, ResolvedDevice, AlertAction, InitiatingProcessFileName, FileName,
FolderPath, ProcessCommandLine, Port7443PairCount, NimPlantUriPairCount, SampleEndpoints, SampleUrls,
ConnectionCount
| sort by coalesce(ProcTime, NetLastSeen) desc
```

sigma_covenant_poshc2_srdi_and_loader_relays.yml

SIGMA

```
title: Execution of sRDI Reflective DLL Shellcode Converter or Loader Function
id: 45b9c823-ad42-4e53-982a-045002d3b822
status: experimental
description: Detects PowerShell or Python processes invoking sRDI reflective DLL shellcode conversion functions
or header-clearing flags (ConvertTo-Shellcode, Invoke-Shellcode, Out-Shellcode, ShellcodeRDI, SRDI_CLEARHEADER)
used to wrap native DLLs for in-memory post-exploitation loading.
author: Odin by Brandefense
date: 2026/11/20
modified: 2026/11/20
references:
  - https://attack.mitre.org/techniques/T1620/
  - https://github.com/monoxgas/sRDI
tags:
  - attack.defense-evasion
  - attack.t1620
  - attack.execution
  - attack.t1059.001
logsource:
  category: process_creation
  product: windows
detection:
  selection_interp:
```

```
    Image|endswith:
      - '\powershell.exe'
      - '\pwsh.exe'
      - '\python.exe'
      - '\python3.exe'
  selection_srdi_cli:
    CommandLine|contains:
      - 'ConvertTo-Shellcode'
      - 'Invoke-Shellcode'
      - 'Out-Shellcode'
      - 'ShellcodeRDI'
      - 'SRDI_CLEARHEADER'
  filter_approved_lab:
    ComputerName:
      - 'REDTEAM-SIM-WIN-01'
      - 'PURPLE-DOTNET-LAB-01'
  condition: selection_interp and selection_srdi_cli and not filter_approved_lab
falsepositives:
  - Authorized red-team or purple-team hosts registered by exact ComputerName in filter_approved_lab
level: high
---
title: Covenant .NET C2 Credential Model or SharpSploit Task Invocation
id: 45c0d934-be53-4f64-a93b-045003e4c733
status: experimental
description: Detects dotnet, MSBuild, or script processes referencing Covenant credential-harvesting model
classes, SharpSploit credential namespaces, or MimikatzCommand task parameters outside named simulation hosts.
author: Odin by Brandefense
date: 2026/11/20
modified: 2026/11/20
references:
  - https://attack.mitre.org/software/S1155/
  - https://attack.mitre.org/techniques/T1558/
tags:
  - attack.credential-access
  - attack.t1558
  - attack.resource-development
  - attack.t1583.003
logsource:
  category: process_creation
  product: windows
detection:
  selection_postex_artifacts:
    CommandLine|contains:
      - 'Covenant.API.Models'
      - 'CapturedTicketCredential'
      - 'CapturedHashCredential'
      - 'SharpSploit.Credentials'
      - 'MimikatzCommand'
  filter_approved_lab:
    ComputerName:
      - 'REDTEAM-SIM-WIN-01'
      - 'PURPLE-DOTNET-LAB-01'
      - 'DLP-EXFIL-TEST-SRV-01'
  condition: selection_postex_artifacts and not filter_approved_lab
falsepositives:
  - Authorized red-team or DLP-validation build servers listed by exact hostname in filter_approved_lab
level: high
```

ODIN

About Odin Research

Odin Research is the public research series of Odin by Brandefense. It documents adversary infrastructure seen on the internet and turns each finding into indicators and detection rules that defenders can use. Victims and third parties stay anonymous, addresses appear only for confirmed adversary infrastructure, and no working attack code is published.

Odin is the agentic analyst inside the Brandefense threat intelligence platform, which unifies Cyber Threat Intelligence, Digital Risk Protection and External Attack Surface Management.

Online edition, indicators and rules · brandefense.io/odin-research/reports/loader-and-net-c2-infrastructure/
Short Read · brandefense.io/odin-research/blog/pikabot-to-nim-c2/
Defense Kit · brandefense.io/odin-research/defense-kit/