ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 19 / 30 · C2 INFRASTRUCTURE

Evasive C2 and Dual-Use Cracking
Brute Ratel, Overlord and Hashtopolis

How six evasive command-and-control frameworks, 177-port hubs and Hashtopolis cracking servers look from the outside, and which of them to block

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

2,263
Endpoints observed on 1,759 distinct hosts, across 10 C2, cracking and IoT families

1,613
Evasive C2 endpoints across six frameworks, on 1,191 hosts

55
Confirmed hostile C2 endpoints (2.4% of all endpoints)

507
Brute Ratel C4 hits that are commercial VPN exits, not C2

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

How six evasive command-and-control frameworks, 177-port hubs and Hashtopolis cracking servers look from the outside, and which of them to block

01	Executive Summary	04
02	Three Populations and One Analytical Model	07
03	Evasive C2 Fleets	10
04	Multi-Listener Hubs and the Kerberoast Model	14
05	Hashtopolis Servers and Infected IoT Routers	18
06	Telling Real C2 from Look-Alikes	22
07	What Defenders Should Do	26
08	Detection Rules	28
09	Indicators	32
10	Scope and Limits	35
11	Sources	37
A	Indicators of compromise (15)	39
B	Detection rules (4 files)	40

IN SHORT

In a two-week window ending 18 November 2026, Odin by Brandefense observed 1,613 command-and-control (C2) endpoints across six evasive frameworks, 32 Hashtopolis cracking servers and 618 infected IoT routers. Only 55 C2 endpoints are confirmed. Another 507 Brute Ratel C4 hits are commercial VPN exits sharing a Go TLS fingerprint. One Tencent Cloud host runs 177 listeners. Nothing links the three groups. Block the 55 confirmed C2s, watchlist Hashtopolis servers, and alert when one user requests five or more RC4 Kerberos tickets in five minutes.

CHAPTER 01

Executive Summary

IN SHORT

In a two-week window ending 18 November 2026, Odin by Brandefense observed 1,613 command-and-control (C2) endpoints across six evasive frameworks, 32 Hashtopolis cracking servers and 618 infected IoT routers. Only 55 C2 endpoints are confirmed. Another 507 Brute Ratel C4 hits are commercial VPN exits sharing a Go TLS fingerprint. One Tencent Cloud host runs 177 listeners. Nothing links the three groups. Block the 55 confirmed C2s, watchlist Hashtopolis servers, and alert when one user requests five or more RC4 Kerberos tickets in five minutes.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04
The Study in Numbers 05

01 Executive Summary

Modern post-exploitation frameworks are built to stay quiet on monitored workstations. Their implants avoid hooked Windows APIs, hide in memory while they sleep, and talk over ordinary-looking HTTPS. On the other side, stolen password hashes can be cracked offline on rented GPU rigs, where no domain controller sees a failed login. This report maps the internet-facing side of both: the C2 servers, the hubs that stack many listeners on one address, and the Hashtopolis servers that coordinate GPU cracking.

1. **Six evasive C2 frameworks run on 1,613 endpoints; 55 are confirmed.**
   Brute Ratel C4 leads with 690 endpoints on 434 hosts. Cobalt Strike (391), DarkSword (346), Overlord (107), Mythic (54) and Havoc (25) follow. Most Brute Ratel C4 hits are not C2 at all: 507 are commercial VPN exits with the same Go TLS fingerprint.

2. **Thirty-five hubs pack up to 177 C2 listeners onto one cloud address.**
   The largest, Hub-01 on Shenzhen Tencent Cloud, mixes HypeAgent (161), Cobalt Strike (14) and PlugX (2) behind spoofed Chinese bank and security-vendor certificates. In Germany, 77.72.18[.]83 runs nine confirmed Cobalt Strike listeners with one shared certificate.

3. **Thirty-two Hashtopolis servers show the GPU cracking side, but use stays unproven.**
   Hashtopolis splits password-cracking jobs across remote Hashcat GPU workers. It is open-source and widely used by auditors and red teams. No hostile cracking job was observed, so these servers belong on watchlists, not blocklists.

Three Decisions for Leaders

1. **Remove RC4 from Kerberos and watch for ticket bursts.** Move service accounts to Group Managed Service Accounts (gMSA) or random passwords of 28 or more characters, with AES-only encryption. Alert when one user requests five or more RC4 service tickets within five minutes.
2. **Pair memory telemetry on endpoints with precise network rules.** Alert on unbacked memory threads touching LSASS or sacrificial Windows processes. Block the 55 confirmed C2 endpoints, not every Go TLS fingerprint hit or every password-auditing server.
3. **Control who can run Hashtopolis and Hashcat.** Allow them only on named red-team audit hosts. Put the large hubs, unattributed Hashtopolis servers, infected routers and dynamic telecom addresses on short-lived watchlists.

The Study in Numbers

Number	What it is
2,263	Endpoints observed on 1,759 distinct hosts, across 10 C2, cracking and IoT families
1,613	Evasive C2 endpoints across six frameworks, on 1,191 hosts
55	Confirmed hostile C2 endpoints (2.4% of all endpoints)
507	Brute Ratel C4 hits that are commercial VPN exits, not C2
177	Listeners on the single busiest hub (Hub-01, Tencent Cloud CN)
32	Hashtopolis cracking endpoints on 26 hosts, all watchlisted
618	Infected IoT router endpoints on 542 hosts (Mirai 353, Mozi 263, Moobot 2)

How to use this report.

Security leaders can read Chapter 1, Chapter 6 and Chapter 7. Detection engineers and SOC teams should start with Chapters 7 to 9. CTI analysts and hunters will find the infrastructure detail in Chapters 2 to 5.

02

CHAPTER 02

Three Populations and One Analytical Model

IN SHORT

The data holds three separate populations: evasive C2 servers, Hashtopolis cracking servers and infected IoT routers. No shared address, certificate or traffic connects them, so Figure 1 is an analytical model of their roles, not a map of one operation.

INSIDE THIS CHAPTER

What was observed 07

The four roles in the model 08

Neighbours on the same hosts 08

02 Three Populations and One Analytical Model

What was observed

Population	Endpoints	Distinct hosts	Confirmed hostile
Evasive C2 (six frameworks)	1,613	1,191	55
Hashtopolis cracking servers	32	26	0
Infected IoT routers (Mirai, Mozi, Moobot)	618	542	0
Total	**2,263**	**1,759**	**55**

Separately, 35 multi-listener hubs bind 391 sockets in total. Of the 35 hub addresses, 21 also appear in the main fleet. The 391 hub sockets happen to equal the 391 Cobalt Strike endpoints; the two numbers are unrelated.

The three populations share no address. We also observed no traffic, shared certificate or tasking log between them. They appear together in this report because they play complementary roles in documented intrusion tradecraft (MITRE T1558.003 Kerberoasting and T1110.002 Password Cracking).

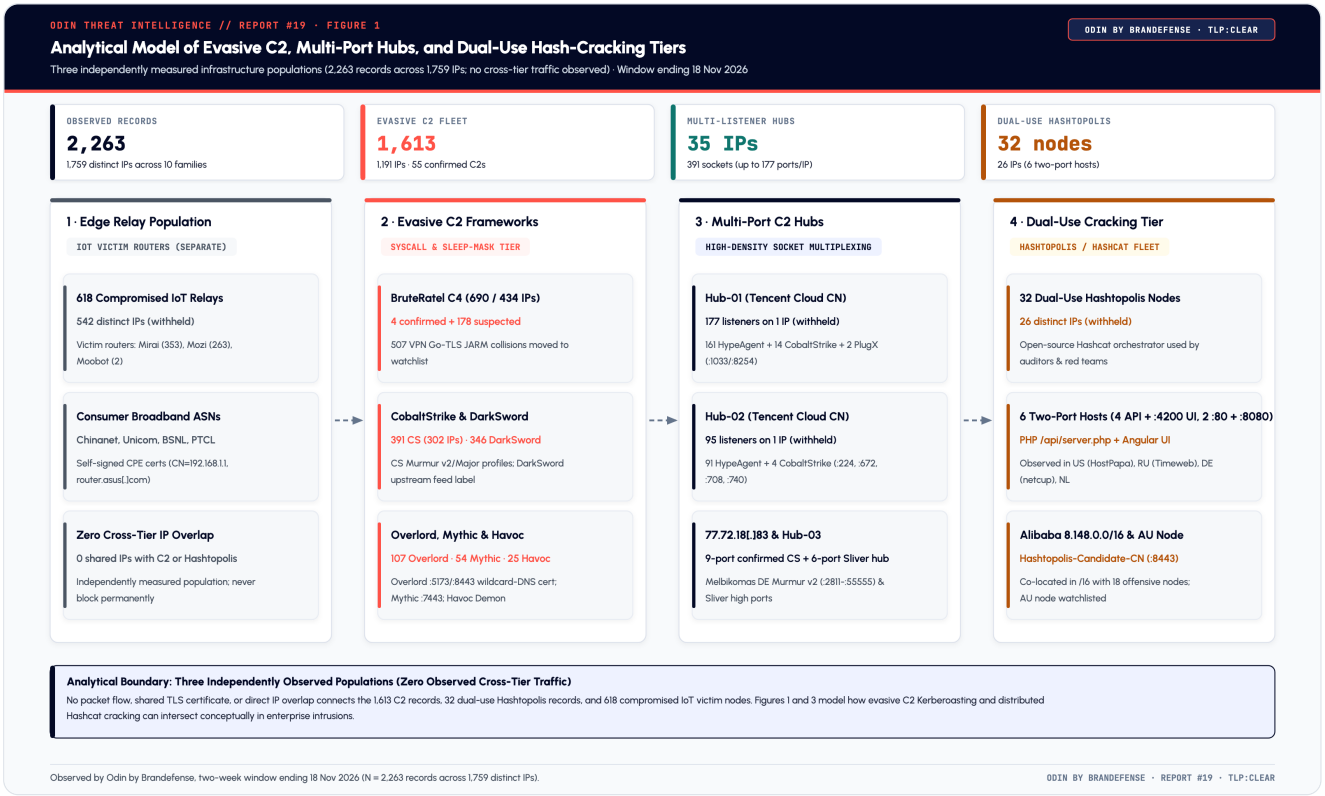


Figure 1: Analytical model of evasive C2, multi-port hubs and dual-use hash-cracking tiers

The four roles in the model

This is an analytical model. It explains why each population needs a different response.

Role	What it is	Defender response
Evasive implant fleet	Brute Ratel C4, DarkSword, Overlord, Mythic and Havoc servers whose implants evade user-mode hooks and use flexible HTTPS or DNS-over-HTTPS profiles	Block confirmed listeners
Multi-implant hubs	Cloud servers that terminate dozens of Cobalt Strike, PlugX, HypeAgent and Sliver sockets	Watchlist and correlate; block only confirmed hubs
Dual-use cracking tier	Hashtopolis servers (ports 8080, 8081, 8443, 4200) that hand out Hashcat jobs through `/api/server.php`	Watchlist, and harden Active Directory
Infected IoT routers	Consumer routers showing `CN=192.168.1.1`, `CN=router.asus[.]com` or `CN=ZXV10`	Victims; never block permanently

Neighbours on the same hosts

Beyond stacking ports within one family, 59 hosts run one of the ten families next to other offensive tooling. Examples include Mythic with HexStrikeAI on confirmed host 138.197.31[.]170. Other pairs, on hosts not published here, include Mythic with AdaptixC2, DarkSword with CyberStrikeAI or mitmproxy, Havoc with HexStrikeAI, Overlord with Sliver, and Cobalt Strike with PentAGI.

03

CHAPTER 03

Evasive C2 Fleets

IN SHORT

Five frameworks other than Cobalt Strike account for 1,222 endpoints. Their implants rely on indirect syscalls, hardware-breakpoint patching and encrypted sleep, and 20 of their endpoints are confirmed after the VPN look-alikes are removed.

INSIDE THIS CHAPTER

How these implants avoid detection 10

Brute Ratel C4: 690 endpoints, three very different groups 11

DarkSword, Overlord, Mythic and Havoc 11

03 Evasive C2 Fleets

How these implants avoid detection

Endpoint detection tools often watch Windows API calls in user mode. The implants in this chapter are documented to step around that.

Technique	What it means
Indirect syscalls	The implant finds system call numbers itself and jumps into a `syscall; ret` instruction inside the real `ntdll.dll`, so hooks never see the call
Hardware-breakpoint patching	It disables `AmsiScanBuffer` and `EtwEventWrite` through CPU debug registers (`Dr0` to `Dr3`) instead of writing to memory
Sleep masking	Between check-ins, it encrypts its own memory and marks it non-executable, then restores it on wake-up
Flexible channels	Check-ins travel over HTTPS, DNS-over-HTTPS (DoH) or SMB named pipes, shaped to look like normal traffic

Figure 2 compares the ten families by volume, distinct hosts and port stacking, next to the top eight hubs.



Figure 2: Ten-family endpoint volume, socket density and top multi-listener hubs

Brute Ratel C4: 690 endpoints, three very different groups

Brute Ratel C4 (MITRE S1063) has a Go teamserver called Ratel and a position-independent implant called Badger. Its 690 endpoints sit on 434 hosts, an average of 1.59 sockets per host. An upstream feed labels 523 of them "confirmed". They split into three main groups, plus one endpoint on a dynamic ISP line.

Group	Endpoints	What it shows	Status
Dedicated Ratel servers with custom TLS	4 confirmed, plus 23 suspected	Operator-set certificates such as `CN=brute[.]bzh[.]ovh` or the default placeholder `CN=FQDN`	Block the 4 confirmed
CTG Server Limited port-80 cluster	155	Plain HTTP on 80 in Singapore and Hong Kong, matching HTTP redirector profiles	Suspected
Commercial VPN exits	507	VPN city-rack certificates and the stock Go TLS fingerprint	Not C2; watchlist only
Dynamic ISP line	1	A match on a dynamic telecom network	72-hour watchlist

The four confirmed Ratel listeners all use custom TLS or custom ports. Two are published:

- **161.35.73[.]220:443** on DigitalOcean (DE), with `CN=brute[.]bzh[.]ovh` and JARM `3fd3fd20d...99b095c`.
- **78.90.248[.]224:7000** on Net-Surf.net (BG), a high-port listener with the same JARM.

The other two sit on hosts that also carry unconfirmed sockets, so their addresses are withheld. A suspected listener on Vultr in Israel shows `CN=FQDN` and JARM `3fd21b20d...c2286f5e` on port 8081. A suspected LUCIDACLOUD host in Hong Kong listens on 9090, 9091 and 9200.

Of the 155 CTG Server Limited listeners, 11 carry an upstream "confirmed" label. They show no certificate subject or JARM, so they stay suspected until there is protocol evidence.

Badger check-ins use JSON keys such as `"b_arch"`, `"c_name"` and `"d_name"`. Badger can also run Beacon Object Files (COFF) in memory.

DarkSword, Overlord, Mythic and Havoc

> **DarkSword is a feed label, not an actor.**
>
> In this report, DarkSword means an upstream threat-feed label for multi-port TLS and HTTP C2 servers. It does not point to one named threat actor or to a known exploit kit.

Framework	Endpoints (hosts)	Confirmed	Main hosting	Main ports
DarkSword	346 (292)	10	CNServer LLC (189, 54.6%), cognetcloud INC (21), CTG Server Limited (12), NetLab Global (12), Alibaba (11)	443 (222), 8888 (62), 80 (23), 2000 (22)
Overlord	107 (94)	4	DE (42) and US (25); Blatant Technologies (16), OVH SAS (13), EggyWall.org (8), AWS (8), Hetzner (7), Private Layer (6)	5173 (51, 47.7%), 443 (26), 8443 (13), 1080 SOCKS (3)
Mythic	54 (46)	2	DigitalOcean (13), AWS (8)	80 (18), 443 (14), 7443 (13), 4443 (3), 8443 (3)
Havoc	25 (23)	0	AWS, DigitalOcean, Oracle, Hostinger	443 (13), 8443 (4)

DarkSword. Over half the fleet (183 endpoints, 52.9%) shares JARM `21d14d00021d21d00042d43d00000091f9827a8676a9d9f27d421962a09b5d`. All 10 confirmed DarkSword rows sit on hosts that also carry unconfirmed sockets, so no DarkSword address is published. New candidates keep appearing on port 8888; one appeared on Alibaba Cloud Hong Kong on 18 November 2026.

Overlord. Confirmed single-socket hosts use hex-generated `.invalid` certificate subjects. Two are published:

Endpoint	Hosting	Certificate subject
194.26.192[.]57:9000	1337 Services GmbH (NL)	`CN=host-43c6ed77a39b7312.invalid`
23.95.217[.]154:8443	DataWagon LLC (US)	`CN=host-21ffa7a17fac9f9a.invalid`

Two three-port Overlord hubs on Private Layer in Switzerland reuse a wildcard-DNS helper certificate. Each has two confirmed sockets on 80 and 8443 and one candidate on 443. Two OVH clusters share another third-party certificate pattern. Those certificate subjects are withheld.

Mythic. Mythic runs its admin and C2 service on port 7443 by default. Two confirmed listeners share JARM `1dd40d40d...9311926e`:

Endpoint	Hosting
5.188.86[.]227:7443	Global Layer B.V. (NL)
138.197.31[.]70:7443	DigitalOcean, LLC (US)

Documented Mythic agents include Apollo (.NET), Poseidon (Go) and Apfell (macOS JXA). They send base64-encoded UUID and JSON messages with actions such as `"checkin"` and `"get_tasking"`.

Havoc. None of the 25 Havoc endpoints is confirmed (5 suspected, 19 candidates, 1 open directory). One suspected server on Adsota Corporation in Vietnam shows `CN=0.0.0.0` on 8443. Havoc's Demon implant is documented to use Ekko and Foliage sleep obfuscation. It queues timer callbacks that switch its memory to read-write, encrypts it with `SystemFunction032` (RC4) while asleep, and restores execute rights on wake-up.

04

CHAPTER 04

Multi-Listener Hubs and the Kerberoast Model

IN SHORT

Thirty-five hubs bind 391 mixed-framework sockets; two Tencent Cloud hosts hold 272 of them. This chapter also models, without observed evidence, how stolen Kerberos tickets could feed offline GPU cracking.

INSIDE THIS CHAPTER

The Cobalt Strike fleet 14

Six flagship hubs 14

Hub-01: three families on one address 14

The Murmur certificate template 15

A workflow model: from Kerberos tickets to offline cracking 15

04 Multi-Listener Hubs and the Kerberoast Model

The Cobalt Strike fleet

Cobalt Strike (MITRE S0154) has 391 endpoints on 302 hosts. Of these, 35 are confirmed, 143 suspected, 207 candidates, 4 candidate open directories and 2 on dynamic ISPs. The top JARM `2ad2ad16d...ed24a4da` appears on 36 endpoints, and the default teamserver port 50050 on 36.

Of the 35 confirmed, 32 sit on hosts where every socket is confirmed. Chapter 9 publishes a selected subset, including 47.239.109[.]126:3389 with `CN=Major Cobalt Strike` and 163.172.45[.]158:8086 with `CN=Murmur Autogenerated Certificate v2`.

Six flagship hubs

The 35 hubs carry 3 to 177 ports each. They split into 7 Cobalt Strike hubs, 14 Sliver hubs, 9 DarkSword hubs, 4 Overlord hubs and 1 Mythic hub.

Hub	Hosting	Sockets	Framework mix	Certificates and status
Hub-01	Shenzhen Tencent (CN)	177	HypeAgent 161, Cobalt Strike 14, PlugX 2	Spoofed Chinese security-vendor, bank and cloud subjects; suspected
Hub-02	Shenzhen Tencent (CN)	95	HypeAgent 91, Cobalt Strike 4	Spoofed Chinese bank, payment and cloud subjects; suspected
77.72.18[.]83	Melbikomas UAB (DE)	9	Cobalt Strike 9	`CN=Murmur Autogenerated Certificate v2`; confirmed, block
Hub-03	Tencent-hosted (HK)	6	Sliver 6	No certificate subject; ports 17443, 25555, 31555, 34443, 34567, 44443; candidate
Hub-04	Microsoft Corp (SG)	5	DarkSword 5	Third-party subject withheld; 2 confirmed, 1 suspected, 2 candidates
Hub-05	OVH SAS (DE)	4	Overlord 4	Third-party subject withheld; ports 80, 443, 1080, 8443; suspected

Hub addresses other than 77.72.18[.]83 are withheld because their sockets are not all confirmed.

Hub-01: three families on one address

Hub-01 binds 161 suspected HypeAgent ports. Their certificates impersonate Chinese commercial banks and security vendors. It also binds 14 suspected Cobalt Strike ports on low ports such as 481, 490, 520, 571 and 749, with certificates that impersonate Chinese cloud and DNS providers.

The last two ports, 1033 and 8254, are suspected PlugX (Korplug, MITRE S0013). Both show a spoofed Chinese security-vendor certificate. In the hub data, PlugX appears only on Hub-01.

Hub-02 binds 95 sockets with the same spoofing template: 91 HypeAgent and 4 Cobalt Strike on 224, 672, 708 and 740. Together, Hub-01 and Hub-02 hold 272 of the 391 hub sockets (69.6%).

The Murmur certificate template

77.72.18[.]83 binds nine confirmed Cobalt Strike listeners on ports 2811, 3323, 4848, 5550, 7007, 8080, 9290, 50500 and 55555. All present `CN=Murmur Autogenerated Certificate v2`.

Across the data, 28 confirmed Cobalt Strike endpoints reuse this exact certificate. They include 15 listeners on 13 OVH France addresses in `92.222.152[.]0`/24, on ports 10001 to 10025.

A workflow model: from Kerberos tickets to offline cracking

> **This section is a model, not an observation.**
>
> We saw no Kerberos tickets, hash lists, C2-to-Hashtopolis transfers, cracking results or service-account logons. Figure 3 is built from MITRE ATT&CK T1558.003 (Kerberoasting) and T1110.002 (Password Cracking).

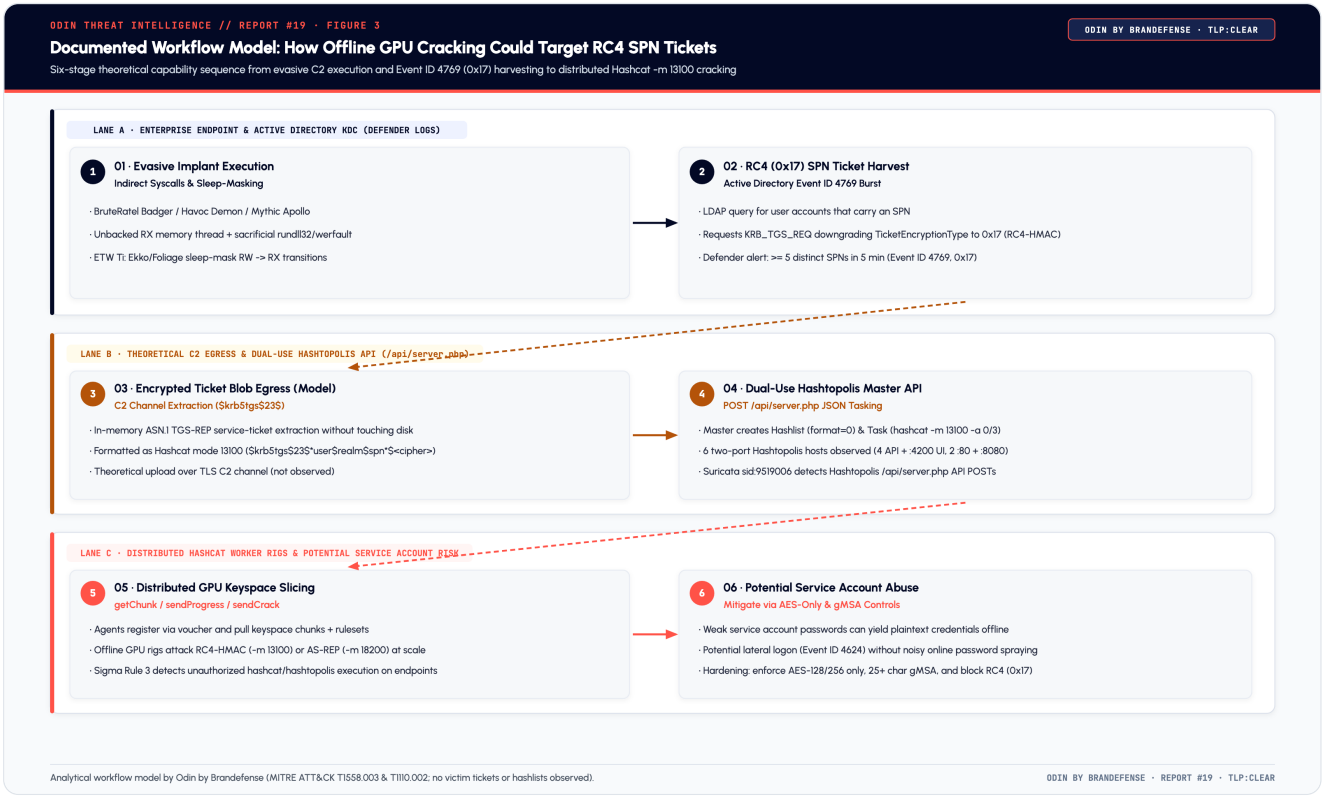


Figure 3: Documented workflow model of how offline GPU cracking could target RC4 SPN tickets

In a well-monitored domain, touching `lsass.exe` trips Credential Guard, Protected Process Light (PPL) or kernel ETW callbacks. Online password spraying trips lockouts. Offline cracking avoids both.

Stage	What could happen	Where defenders can see it
01 Implant runs	An evasive implant runs with any normal domain user token	Unbacked memory threads, sacrificial processes
02 Ticket request	It requests service tickets for accounts with a Service Principal Name (SPN), asking for RC4-HMAC (`etype 0x17`)	Event 4769 bursts with encryption type 0x17
03 Exfiltration	Ticket blobs (`$krb5tgs$23$`) leave over the encrypted C2 channel	C2 egress rules
04 Hashtopolis master	The hashes are loaded into a Hashtopolis server through `/api/server.php`	`/api/server.php` traffic from inside
05 GPU workers	Remote Hashcat workers crack keyspace slices (`-m 13100`)	Hashcat or Hashtopolis processes on endpoints
06 Account abuse	Weak service passwords are recovered with no failed logins on the domain controller	New logons (Event 4624) by service accounts

The same path works for AS-REP roasting (`$krb5asrep$23$` , Hashcat `-m 18200`), Net-NTLMv2 (`-m 5600`), NTLM (`-m 1000`) and Domain Cached Credentials (DCC2, `-m 2100`).

The chain breaks at stage 02. Set SPN accounts to AES only (`msDS-SupportedEncryptionTypes = 24` , AES128 plus AES256, RC4 excluded) and move them to gMSAs.

CHAPTER 05

Hashtopolis Servers and Infected IoT Routers

IN SHORT

The 32 Hashtopolis endpoints expose worker APIs and web consoles on predictable ports. Because the tool is dual-use and no hostile job was seen, every one of them is a watchlist entry, and every address is withheld.

INSIDE THIS CHAPTER

What Hashtopolis is	18
Where the 32 endpoints sit	18
Selected Hashtopolis servers (addresses withheld)	19
How the worker protocol looks on the wire	19
The Alibaba /16 neighbourhood	19
Infected IoT routers: victims, not attackers	20

05 Hashtopolis Servers and Infected IoT Routers

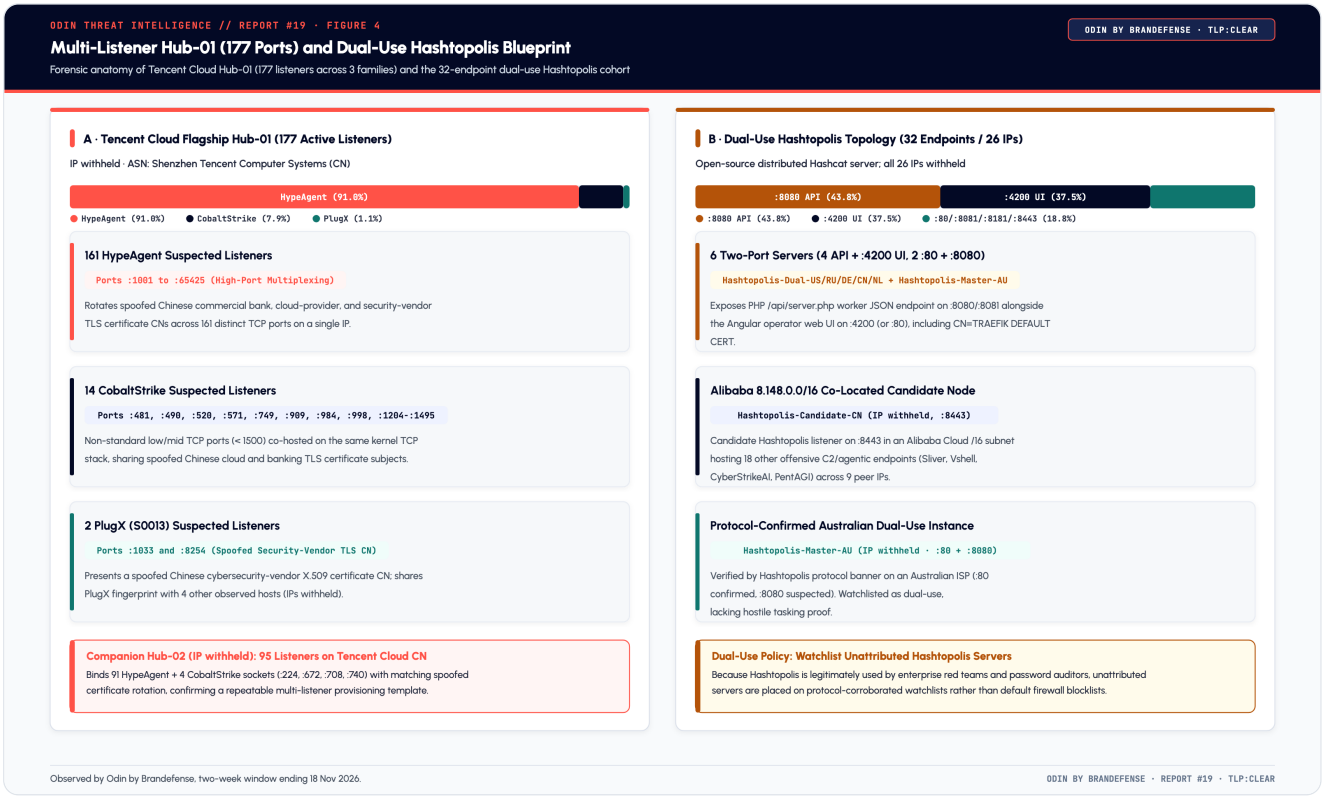


Figure 4: Multi-listener Hub-01 (177 ports) and the dual-use Hashtopolis blueprint

What Hashtopolis is

Hashtopolis is an open-source server that splits Hashcat cracking work across many GPU machines. Auditors use it for authorised password audits. Attackers can use it the same way. From the outside, an auditor's server and an attacker's rig look the same without evidence of their jobs.

Where the 32 endpoints sit

The endpoints span at least 10 countries: Russia 5, Germany 5, Netherlands 5, China 3, United States 3, Finland 3, Australia 2, Singapore 2, Ukraine 1 and Taiwan 1.

Port	Role	Endpoints
8080	Worker API	14
4200	Hashtopolis v2 Angular web console	12
80	Web or API	2
8081	Worker API	2
8181	Worker API	1
8443	Worker API	1

Six hosts expose two Hashtopolis ports at once: three with 8080 plus 4200, one with 8081 plus 4200, and two with 80 plus 8080. The hosts average 1.23 endpoints each.

Selected Hashtopolis servers (addresses withheld)

Server	Ports	Hosting	Notes	Status
Australian master	80, 8080	An Australian ISP	Hashtopolis protocol confirmed on 80; API on 8080	Watchlist (dual-use)
Alibaba candidate	8443	Hangzhou Alibaba Advertising (CN)	Sits in a /16 with other offensive tooling (below)	Watchlist (candidate)
Finnish console	4200	Hetzner Online GmbH (FI)	Third-party Hashtopolis certificate (withheld), JARM `15d3fd...ccbc0b6b`	Watchlist (candidate)
US two-port	8080, 4200	HostPapa (US)	`CN=TRAEFIK DEFAULT CERT`	Watchlist (candidate)
Russian two-port	8080, 4200	JSC "TIMEWEB" (RU)	Third-party certificate withheld	Watchlist (candidate)
German two-port	8081, 4200	netcup GmbH (DE)	Third-party certificate withheld	Watchlist (candidate)
Chinese two-port	80, 8080	Hangzhou Alibaba Advertising (CN)	Web console on 80, API on 8080	Watchlist (candidate)
Taiwan listener	8080	Chunghwa Telecom (TW)	Dynamic subscriber address	72-hour watchlist

How the worker protocol looks on the wire

Remote GPU workers download the Python or C# agent (`hashtopolis.zip`). They enrol once with a voucher through `POST /api/server.php`, then log in. After that they keep polling with the actions `getTask` and `getChunk`.

The server cuts a mask or rule attack into fixed keyspace slices. Workers send results back with `sendProgress` and `sendCrack`. Supported hash modes include Kerberoast RC4 (`-m 13100`), AS-REP (`-m 18200`), Net-NTLMv2 (`-m 5600`), NTLM (`-m 1000`) and DCC2 (`-m 2100`).

Suricata SID 9519006 and the third Sigma rule catch internal machines or GPU workers calling `/api/server.php`. No rule in this report covers the Hashtopolis web console.

The Alibaba /16 neighbourhood

The Alibaba candidate on 8443 was observed on 16 and 17 November 2026. Its /16 allocation, 8.148.0.0/16, holds 19 endpoints across 9 other addresses. Two were MySQL 3306 certificate collisions and are excluded. The other tooling includes:

- CyberStrikeAI (8080) and HexStrikeAI (5000)
- Metasploit Meterpreter (4446 and 443), sharing a host with Sliver (8888)
- VShell (443), mitmproxy, ARTEX (8787) and PentAGI
- DNSLog out-of-band testing (8000, 8443)

Sharing a public cloud /16 is circumstantial. It does not show that one operator controls all of these, so the Hashtopolis candidate stays on a watchlist.

Infected IoT routers: victims, not attackers

The 618 Mirai, Mozi and Moobot endpoints on 542 hosts share no address with the C2 or Hashtopolis populations. We observed no traffic between them. These are third-party consumer routers, so every address is withheld.

Family	Endpoints (hosts)	Where	Fingerprints
Mirai	353 (279)	CN 107 (China Unicom, Chinanet), TW 53 (Taiwan Fixed Network), ET 43 (EthioNet), PH 30 (Globe Telecom), AL 28	`CN=192.168.1.1` (16), `CN=router.asus[.]com` (3), `CN=ZXV10` (2); 19 Taiwan routers share JARM `04b03b000...99e2262e`
Mozi	263 (261)	Consumer gateways, including BSNL India	`CN=192.168.1.1`, JARM `04d02d000...99e2262e`; peer-to-peer DHT mesh
Moobot	2 (2)	Bulgaria and Latvia	Candidates only

In the Ethiopian group, 18 routers show a personal-name certificate subject, which is withheld. Of the Mirai endpoints, 63 are protocol-confirmed infections. They go on a watchlist for infected relays, never on a permanent block list. Infected routers are often reused as proxy or scanning hops in wider internet activity.

CHAPTER 06

Telling Real C2 from Look-Alikes

IN SHORT

Internet-wide C2 hunting collides with VPN exits, password-auditing servers, home routers, database certificates and dynamic broadband. Removing those leaves 55 confirmed C2 endpoints, 2.4% of the 2,263 observed.

INSIDE THIS CHAPTER

Five sources of false positives 22
The status split 23
Two other explanations we weighed 24

06 Telling Real C2 from Look-Alikes

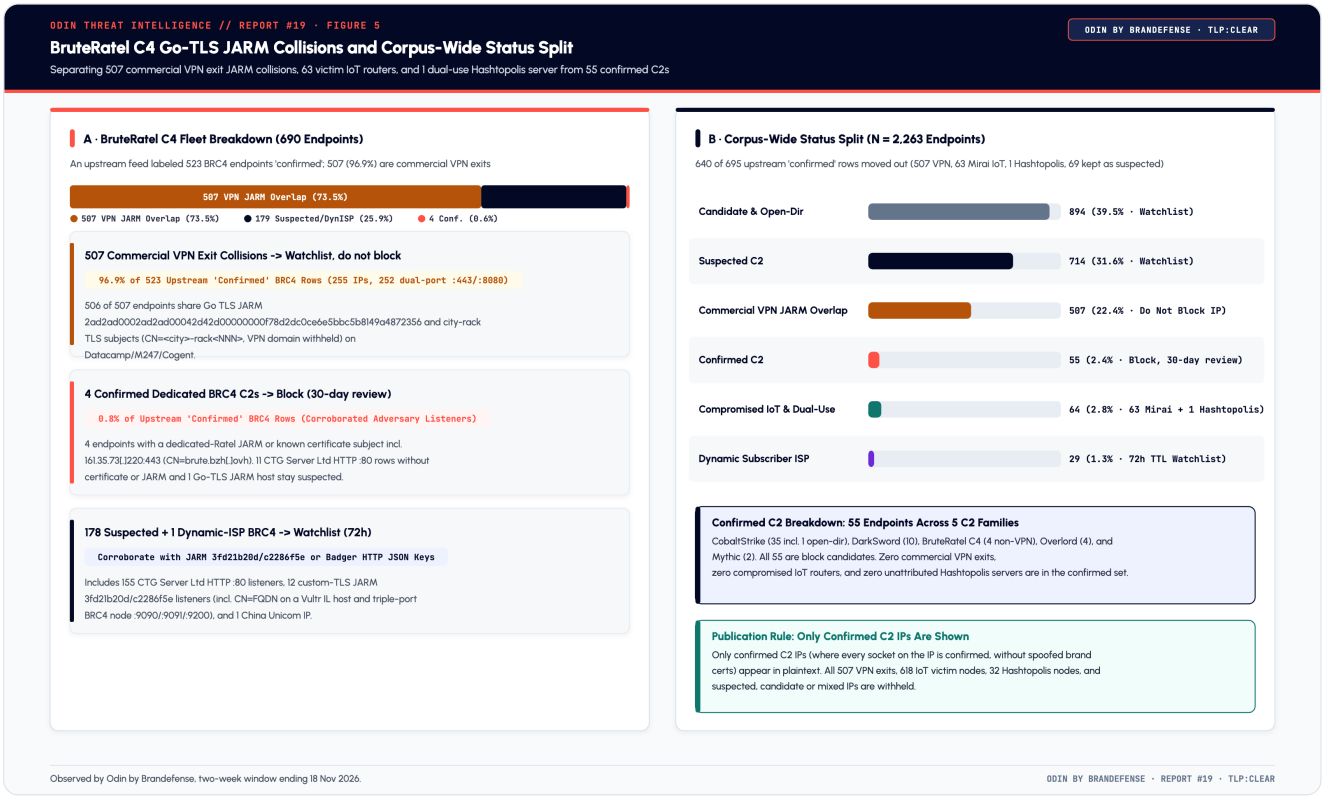


Figure 5: BruteRatel C4 Go TLS JARM collisions and the corpus-wide status split

Five sources of false positives

1 **Commercial VPN exits that look like Brute Ratel C4.**

The Ratel teamserver is written in Go. Its default TLS stack gives the stock Go JARM `2ad2ad0002ad2ad00042d42d00000000f78d2dc0ce6e5bbc5b8149a4872356` and a minimal HTTP 404 reply. Commercial VPN exits give the same JARM and 404 profile.

The 507 VPN rows sit on 255 hosts: 252 with paired 443 and 8080 sockets, 3 with one port. Of the 507, 506 share the Go JARM; one has no JARM. Their certificates follow a VPN naming pattern: 322 use `<city>-rack<NNN>` names under a VPN domain, and 185 use labels such as `sydney439`.

Hosting network	VPN rows
Datacamp Limited	277
M247 Europe SRL	95
Cogent Communications	75
Servers Australia	20
GSL Networks	10
GleSYS	10

Blocking all 523 "confirmed" Brute Ratel C4 rows would cut off 255 VPN exits used by remote staff and customers. Only dedicated Ratel servers with a certificate subject or family JARM belong on a block list.

2 **Dual-use servers and infected routers.**

A Hashtopolis server is blocked only with separate evidence of hostile use. The Australian master lacks that evidence. The 63 confirmed Mirai routers are victims.

3 **Large suspected hubs.**

Every socket on Hub-01 and Hub-02 is suspected, while all nine on 77.72.18[.]83 are confirmed. The two Tencent hubs go on a hub watchlist, paired with the Cobalt Strike certificate rule (SID 9519007).

4 **Dynamic telecom addresses.**

Exactly 29 endpoints sit on subscriber telecom networks: 10 Mirai, 9 Mozi, 4 Mythic, 2 Cobalt Strike, 1 Brute Ratel C4, 1 DarkSword, 1 Overlord and the Taiwan Hashtopolis listener. Their addresses get reassigned, so they get 72-hour watchlists.

5 **Internal audit rigs and port-only matches.**

Internal red teams run Hashtopolis legitimately during password audits. Cloud MySQL servers with auto-generated certificates on 3306 can match port-based C2 rules. The rules exclude named internal audit hosts and ignore port-only matches.

The status split

The upstream feed marked 695 rows "confirmed", leaving out dynamic-ISP rows. Of these, 55 remain confirmed C2. Another 571 move to watchlists as VPN exits, infected routers or a dual-use server. The last 69 stay suspected because they lack a family JARM or known certificate subject (12 Brute Ratel C4, 29 Cobalt Strike, 18 DarkSword, 4 Overlord, 6 Mythic).

Family	Endpoints	Upstream "confirmed"	Confirmed C2	Watchlist (VPN, IoT, dual-use)	Suspected	Candidate and open directory	Dynamic ISP
Brute Ratel C4	690	523	4	507 (VPN)	178	0	1
Cobalt Strike	391	64	35	0	143	211	2
Mirai	353	63	0	63 (IoT)	41	239	10
DarkSword	346	28	10	0	230	105	1
Mozi	263	0	0	0	36	218	9
Overlord	107	8	4	0	31	71	1
Mythic	54	8	2	0	34	14	4
Hashtopolis	32	1	0	1 (dual-use)	16	14	1
Havoc	25	0	0	0	5	20	0
Moobot	2	0	0	0	0	2	0
Total	**2,263**	**695**	**55**	**571**	**714**	**894**	**29**

Two other explanations we weighed

Red teams and hobbyists.

Brute Ratel C4, Mythic, Havoc, Cobalt Strike and Hashtopolis are all used by contracted red teams as well as attackers. Some Hashtopolis consoles on 4200, and some Mythic servers with third-party training certificates, likely belong to trainers, consultants or hobbyists. Blocking only the 55 confirmed endpoints avoids breaking legitimate audit infrastructure.

Decoy or redirector on Hub-01.

A single address with 177 ports spoofing Chinese bank, cloud and security-vendor certificates could be an attacker redirector built with `iptables` or `socat`. It could also be a high-interaction honeypot built to attract scanners. A watchlist entry alerts on any internal connection without claiming who runs it.

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

Within 24 hours

- [ ] Block the 11 published confirmed C2 addresses in the Indicators chapter. Recheck the certificate or JARM before keeping a cloud IP block longer than 30 days, because cloud addresses are reassigned after takedowns.
- [ ] Put Hub-01, Hub-02, unattributed Hashtopolis servers, infected routers and dynamic telecom addresses on time-bound watchlists, not block lists.
- [ ] Do not block on the Go TLS JARM `2ad2ad0002ad...` alone. It matches commercial VPN exits.

Within 72 hours

- [ ] Turn on the domain controller audit policy Audit Kerberos Service Ticket Operations (Event 4769).
- [ ] Alert when one non-machine account requests five or more RC4 (0x17) tickets for different SPNs within five minutes.
- [ ] Deploy the Suricata rules for Brute Ratel C4, DarkSword, Overlord and Cobalt Strike certificates, Mythic check-ins and Hashtopolis polling.

Within 7 to 30 days

- [ ] Inventory every user account with a `ServicePrincipalName`. Move eligible services to gMSAs or random passwords of 28 or more characters, and set AES-only encryption (`msDS-SupportedEncryptionTypes = 24`).
- [ ] Turn on kernel ETW Threat Intelligence call-stack telemetry, Credential Guard and LSA RunAsPPL. Alert on Sysmon Event 10 with an `UNKNOWN` call trace touching `lsass.exe`, `rundll32.exe`, `dllhost.exe` or `werfault.exe`.
- [ ] Use application control so Hashcat (`-m 13100`, `-m 18200`, `-m 1000`, `-m 2100`) and Hashtopolis agents run only on named, isolated password-audit hosts.
- [ ] Add the six JARM values in the Indicators chapter to external attack surface hunting, always paired with a certificate subject check.

CHAPTER 08

Detection Rules

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

INSIDE THIS CHAPTER

How the Suricata signatures work 28

How the KQL query works 29

How the Sigma and YARA rules work 29

How to sort a host 30

08 Detection Rules

The full rules are in Appendix B. Every rule file loads without errors in its engine (Suricata 7.0.12, YARA-X, pySigma, Kusto.Language); the rules were not replayed against live traffic. Test each rule in a lab before production use.

Rule file	What it catches
`odin_bruteratel_mythic_plugx_hashtopolis.rules`	Suricata 7: seven signatures for C2 certificates, Badger and Mythic check-ins, and Hashtopolis worker polling
`sentinel_evasive_c2_callstack_and_kerberoast_burst.kql`	Microsoft Sentinel and Defender XDR: contact with confirmed C2s, injection followed by C2 egress, RC4 ticket bursts, and unauthorised Hashcat or Hashtopolis runs
`sigma_unbacked_memory_syscall_and_kerberoast_export.yml`	Sigma (3 rules): unbacked call stacks touching LSASS, RC4 service-ticket requests, and Hashtopolis or Hashcat execution
`odin_brc4_havoc_mythic_and_hashtopolis_artifacts.yar`	YARA (4 rules): Badger configs and syscall stubs, Havoc sleep-mask code, Mythic agent stubs, and Hashtopolis agents

How the Suricata signatures work

SID	Looks for	Why
9519001	Server certificate `CN=brute[.]bzh[.]ovh` or `CN=FQDN`	Dedicated or default Brute Ratel C4 teamservers (S1063)
9519002	Outbound `POST` whose body holds `"b_arch"` plus `c_name`, `d_name`, `p_id` or `b_ver`	Badger check-ins; needs TLS inspection when wrapped in HTTPS
9519003	Server certificate for one of three confirmed DarkSword domains on 443, 8443 or 8888	Confirmed DarkSword listeners (T1071.001)
9519004	Server certificate `CN=host-<16 hex>.invalid` on Overlord ports	Overlord's generated certificates
9519005	Outbound `POST /agent_message` with a base64 body longer than 47 bytes	Mythic agent check-ins; needs TLS inspection on 7443 and similar
9519006	Outbound `POST /api/server.php` with actions `register`, `login`, `getTask`, `getChunk`, `sendProgress` or `sendCrack`	Hashtopolis workers (T1110.002); flagged as policy violation
9519007	Server certificate `CN=Murmur Autogenerated Certificate v2` or `CN=Major Cobalt Strike`	Multi-port Cobalt Strike templates (S0154)

The certificate rules need `tls.cert_subject` inspection on server handshakes. Suppress SID 9519006 only for named internal GPU audit hosts talking to an approved internal Hashtopolis server. If you run self-hosted Mumble voice servers (Murmur, default port 64738), exclude those exact internal IPs from SID 9519007 rather than disabling it.

How the KQL query works

The query looks back 14 days and has four branches.

Branch	Fires when
C2 egress	A device connects to one of the 11 confirmed C2 addresses, or to a watchlist you add from your own CTI feed
Injection plus egress	A process injects into `rundll32.exe`, `dllhost.exe`, `werfault.exe` or similar from a temp or profile folder, and the same process reaches a listed address within 10 minutes
Kerberoast burst	Event 4769 shows one account requesting RC4 (0x17) tickets for five or more distinct SPNs in five minutes, excluding `krbtgt` and machine accounts
Cracking tools	`hashcat` or `hashtopolis` runs, or a command line holds `/api/server.php`, a Kerberos-related Hashcat mode or `hashtopolis.zip`

Fill the approved audit-host and scanner-account lists with your own verified assets. In older forests where legacy SPNs still use RC4, raise the burst threshold from 5 to 8, or exclude exact service accounts while you migrate them to AES.

How the Sigma and YARA rules work

Rule	Fires when
Sigma 1: unbacked memory call stack	Sysmon Event 10 shows a call trace containing `UNKNOWN(` while touching LSASS or a sacrificial host process
Sigma 2: Kerberoast RC4 request	Event 4769 with ticket encryption 0x17, excluding machine accounts and `krbtgt`
Sigma 3: Hashtopolis or Hashcat	Command lines with `/api/server.php`, `hashtopolis.zip`, `--voucher`, Kerberos-related Hashcat modes or `$krb5tgs$23$`, outside named audit hosts
YARA: Brute Ratel C4 Badger	At least two Badger JSON keys (`"b_arch"`, `"c_name"`, `"d_name"`) appear with two evasion strings such as `NtAllocateVirtualMemory`, `EtwEventWrite`, `AmsiScanBuffer` or `coffee`
YARA: Havoc Demon sleep mask	`SystemFunction032`, `CreateTimerQueueTimer` and `NtContinue` appear together with a Havoc or Demon marker
YARA: Mythic agent stub	Mythic JSON actions (`checkin`, `get_tasking`, `post_response`) appear with Apollo, Poseidon or `agent_message` markers
YARA: Hashtopolis agent	`/api/server.php` appears with two worker actions and a `hashcat` or `clientSignature` string

Each YARA rule has a strict file size bound.

How to sort a host

What the host shows	Action
Dedicated C2 with a family certificate or JARM, every socket confirmed	Block
Multi-family hub with suspected sockets only	Hub watchlist
Hashtopolis server with no evidence of hostile use	Dual-use watchlist
Hashtopolis candidate inside an offensive cloud neighbourhood	Candidate watchlist
VPN city-rack certificate with the stock Go JARM on a VPN provider network	VPN watchlist, do not block
Any match on a dynamic telecom network	72-hour watchlist
A named internal audit host	Allowlist
A MySQL or SSH port match with no family signal	Ignore

09

CHAPTER 09

Indicators

IN SHORT

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

INSIDE THIS CHAPTER

Confirmed C2 addresses to block 32

JARM values for hunting 33

MITRE ATT&CK mapping 33

09 Indicators

Appendix A holds 15 published indicators: 11 confirmed C2 addresses and the 4 JARM values tied to confirmed C2. The Brute Ratel 8081 and DarkSword JARM values, listed under JARM values for hunting in this chapter, are for hunting only.

Type	Count
Confirmed C2 addresses to block	11
Watchlist hubs, Hashtopolis servers and cluster candidates (addresses withheld)	11
JARM values for TLS hunting	6 (4 in Appendix A)

No commercial VPN exit is in the set.

Confirmed C2 addresses to block

Address and port	Framework	Hosting	Key fingerprint
161.35.73[.]220:443	Brute Ratel C4	DigitalOcean (DE)	`CN=brute[.]bzh[.]ovh`
191.96.31[.]13:443	Cobalt Strike	Hostinger (US)	Dominant Cobalt Strike JARM
78.90.248[.]224:7000	Brute Ratel C4	Net-Surf (BG)	JARM `3fd3fd20d...99b095c`
163.172.45[.]158:8086	Cobalt Strike	Scaleway (FR)	`CN=Murmur Autogenerated Certificate v2`
200.105.111[.]26:50000	Cobalt Strike	AR	`CN=Murmur Autogenerated Certificate v2`
194.26.192[.]57:9000	Overlord	NL	`CN=host-43c6ed77a39b7312.invalid`
23.95.217[.]154:8443	Overlord	US	`CN=host-21ffa7a17fac9f9a.invalid`
5.188.86[.]227:7443	Mythic	Global Layer (NL)	Default 7443 TLS
138.197.31[.]70:7443	Mythic	DigitalOcean (US)	Default 7443 TLS
77.72.18[.]83:50500	Cobalt Strike	Melbikomas (DE)	Nine ports 2811 to 55555, Murmur certificate
47.239.109[.]26:3389	Cobalt Strike	Alibaba (HK)	`CN=Major Cobalt Strike`

JARM values for hunting

Use these with a certificate subject or issuer check. Never block on a JARM value alone.

Family	Port	JARM	Seen on
Brute Ratel C4	8081	`3fd21b20d00000021c43d21b21b43de0a012c76cf078b8d06f4620c2286f5e`	12 suspected custom Ratel nodes
Brute Ratel C4	443	`3fd3fd20d00000021c43d43d00043d15e5079913504d6ff89229f1f99b095c`	Both published Ratel C2s
DarkSword	443	`21d14d00021d21d00042d43d00000091f9827a8676a9d9f27d421962a09b5d`	183 of 346 endpoints (52.9%)
Overlord	8443	`21d14d00021d21d00042d43d00041df8e4b7c874a766a0a2344c0afd986207`	11 Overlord endpoints on 8443 and 9000, including both published Overlord C2s
Mythic	7443	`1dd40d40d00040d00042d43d000000831b6af40378e2dd35eeac4e9311926e`	Both published Mythic C2s
Cobalt Strike	10010	`29d08d00029d00000000000000000001f93725095f606ed113a9a32b2392e62`	Confirmed Cobalt Strike cluster on 13 OVH addresses in `92.222.152[.]0` /24

Withheld: all hub, Hashtopolis, cluster-candidate, suspected and candidate addresses; the 507 VPN exits and their VPN domain; all 618 router addresses; and every third-party or personal certificate subject. For watchlist entries, correlate outbound traffic with process lineage, the certificate rules or Kerberos anomalies over 72 to 168 hours.

MITRE ATT&CK mapping

Tactic	Technique or software	How it appears here	Detection
Command and Control	T1071.001 Web Protocols	HTTPS and HTTP C2 on DarkSword, Overlord and Mythic	SIDs 9519003 to 9519005
Command and Control	S1063 Brute Ratel C4	Ratel teamservers and Badger check-ins	SIDs 9519001, 9519002; YARA Badger rule
Command and Control	S0154 Cobalt Strike, S0013 PlugX	Multi-port hubs and Murmur certificate templates	SID 9519007
Defense Evasion	T1055 Process Injection	Injection into sacrificial host processes; indirect syscalls	KQL injection branch; Sigma 1
Credential Access	T1558.003 Kerberoasting	Modeled RC4 SPN ticket requests	KQL burst branch; Sigma 2
Credential Access	T1110.002 Password Cracking	Hashtopolis distributes offline Hashcat jobs	SID 9519006; Sigma 3; KQL tool branch

10

CHAPTER 10

Scope and Limits

IN SHORT

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

10 Scope and Limits

- **Window.** The findings cover a two-week observation window ending 18 November 2026. Most endpoints were last seen between 13 and 18 November. Some records carry earliest first-seen dates back to 2022 from other sources; those are not a tracking start.
- **Outside view only.** The data shows internet-facing servers. It cannot show what happened inside any enterprise. No Kerberos ticket theft, cracked password or service-account logon was observed.
- **The Kerberoast chain is a model.** Chapter 4 and Figure 3 describe documented tradecraft. They are not a reconstruction of a real intrusion.
- **No links between populations.** The C2, Hashtopolis and IoT populations share no address, certificate or observed traffic. Nothing here ties them to one operator.
- **Confirmed is not the same as attributed.** Several frameworks are dual-use. A confirmed listener runs the framework; it may still belong to an authorised red team.
- **Feed labels are labels.** DarkSword is an upstream feed label. It does not attribute activity to a named actor.
- **Rule testing.** Every published rule file loads without errors in its engine (Suricata 7.0.12, YARA-X, pySigma and Microsoft's Kusto.Language analyzer, checked on 20 November 2026). The rules were not replayed against live traffic, so test and tune them in your own environment before production use.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. Palo Alto Networks Unit 42. When Pentest Tools Go Brutal: Red-Teaming Tool Being Abused by Malicious Actors. https://unit42.paloaltonetworks.com/brute-ratel-c4-tool/ (verified 18 November 2026)
2. MITRE ATT&CK. Brute Ratel C4, S1063. https://attack.mitre.org/software/S1063/ (verified 18 November 2026)
3. Mythic documentation. https://docs.mythic-c2.net/ (verified 18 November 2026)
4. Havoc Framework documentation. https://havocframework.com/docs/welcome (verified 18 November 2026)
5. MITRE ATT&CK. PlugX, S0013. https://attack.mitre.org/software/S0013/ (verified 19 November 2026)
6. Hashtopolis project. Server repository. https://github.com/hashtopolis/server (verified 18 November 2026)
7. MITRE ATT&CK. Steal or Forge Kerberos Tickets: Kerberoasting, T1558.003. https://attack.mitre.org/techniques/T1558/003/ (verified 18 November 2026)
8. Salesforce Engineering. Easily Identify Malicious Servers on the Internet with JARM. https://engineering.salesforce.com/easily-identify-malicious-servers-on-the-internet-with-jarm-e095edac525a/ (verified 18 November 2026)

APPENDICES

Indicators and Detection Rules

FOR DEFENDERS

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

INSIDE

A. Indicators of compromise (15) 39

B. Detection rules (4 files) 40

A Indicators of Compromise

Values are defanged for display. The CSV, JSON and STIX files on the online edition carry the raw values. Values the report marks as withheld are not listed.

ID	Type	Indicator	Port	Context	Action
R19-IOC-01	IPv4	161.35.73[.]220	443	bruteratelc4	BLOCK
R19-IOC-02	IPv4	191.96.31[.]13	443	cobaltstrike	BLOCK
R19-IOC-03	IPv4	78.90.248[.]224	7000	bruteratelc4	BLOCK
R19-IOC-04	IPv4	163.172.45[.]158	8086	cobaltstrike	BLOCK
R19-IOC-05	IPv4	200.105.111[.]26	50000	cobaltstrike	BLOCK
R19-IOC-06	IPv4	194.26.192[.]57	9000	overlord	BLOCK
R19-IOC-07	IPv4	23.95.217[.]154	8443	overlord	BLOCK
R19-IOC-08	IPv4	5.188.86[.]227	7443	mythic	BLOCK
R19-IOC-09	IPv4	138.197.31[.]70	7443	mythic	BLOCK
R19-IOC-10	IPv4	77.72.18[.]83	50500	cobaltstrike	BLOCK
R19-IOC-11	IPv4	47.239.109[.]26	3389	cobaltstrike	BLOCK
R19-IOC-12	JARM	3fd3fd20d00000021c43d43d00043d15e50799 13504d6ff89229f1f99b095c	443	bruteratelc4	HUNT
R19-IOC-13	JARM	21d14d00021d21d00042d43d00041df8e4b7c8 74a766a0a2344c0afd986207	8443	overlord	HUNT
R19-IOC-14	JARM	1dd40d40d00040d00042d43d000000831b6af4 0378e2dd35eeac4e9311926e	7443	mythic	HUNT
R19-IOC-15	JARM	29d08d00029d00000000000000000001f937250 95f606ed113a9a32b2392e62	10010	cobaltstrike	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_brc4_havoc_mythic_and_hashtopolis_artifacts.yar

YARA

```
/*
   Odin by Brandefense - Report #19 Deterministic YARA Suite (TLP:CLEAR)
   Covers: BruteRatel C4 Badger Syscall & Malleable Config, Havoc Demon Ekko Sleep-Mask,
           Mythic Apollo/Poseidon Agent Check-In Stubs, and Hashtopolis Distributed GPU Agents.
*/

rule ODIN_BruteRatel_C4_Badger_Config_And_Syscall_Stub
{
    meta:
        author = "Odin by Brandefense"
        description = "Detects BruteRatel C4 Badger telemetry JSON keys paired with indirect-syscall / AMSI-ETW
hardware-breakpoint strings"
        reference = "Odin Monograph #19 (19 / 30)"
        classification = "TLP:CLEAR"
        mitre_software = "S1063"
    strings:
        $mz = { 4D 5A }
        $b1 = "\"b_arch\"" ascii wide
        $b2 = "\"c_name\"" ascii wide
        $b3 = "\"d_name\"" ascii wide
        $s1 = "NtAllocateVirtualMemory" ascii wide
        $s2 = "EtwEventWrite" ascii wide
        $s3 = "AmsiScanBuffer" ascii wide
        $s4 = "coffee" ascii wide
    condition:
        filesize < 15MB and (
            ($mz at 0 and 2 of ($b*) and 2 of ($s*)) or
            (all of ($b*) and 2 of ($s*))
        )
}

rule ODIN_Havoc_Demon_Ekko_SleepMask_Implant
{
    meta:
        author = "Odin by Brandefense"
        description = "Detects Havoc Demon implant artifacts chaining SystemFunction032 RC4 memory encryption
with NtContinue ROP timer sleep-masking"
        reference = "Odin Monograph #19 (19 / 30)"
        classification = "TLP:CLEAR"
        mitre_technique = "T1027"
    strings:
        $mz = { 4D 5A }
        $e1 = "SystemFunction032" ascii wide
        $e2 = "CreateTimerQueueTimer" ascii wide
        $e3 = "NtContinue" ascii wide
        $h1 = "Demon" ascii wide
        $h2 = "Havoc" ascii wide
        $h3 = "SpoofCallStack" ascii wide
    condition:
        filesize < 12MB and (
            ($mz at 0 and all of ($e*) and 1 of ($h*)) or
            (all of ($e*) and 2 of ($h*))
        )
}

rule ODIN_Mythic_Apollo_Poseidon_Agent_Checkin_Stub
{
    meta:
        author = "Odin by Brandefense"
        description = "Detects Mythic C2 modular agent (Apollo .NET or Poseidon Go) check-in and tasking JSON
```

```
serialization markers"
        reference = "Odin Monograph #19 (19 / 30)"
        classification = "TLP:CLEAR"
        mitre_technique = "T1071.001"
    strings:
        $m1 = "\"action\":\"checkin\"" ascii wide nocase
        $m2 = "\"action\":\"get_tasking\"" ascii wide nocase
        $m3 = "\"action\":\"post_response\"" ascii wide nocase
        $u1 = "/api/v1.4/agent_message" ascii wide
        $u2 = "ApolloInterop" ascii wide
        $u3 = "poseidon" ascii wide nocase
        $u4 = "mythic_encrypt" ascii wide
    condition:
        filesize < 25MB and (
            (2 of ($m*) and 1 of ($u*)) or
            (1 of ($m*) and 2 of ($u*))
        )
}

rule ODIN_Hashtopolis_Agent_And_Hashcat_Orchestrator
{
    meta:
        author = "Odin by Brandefense"
        description = "Detects Hashtopolis distributed cracking agent scripts or binaries polling /api/server.php
for Hashcat task chunks"
        reference = "Odin Monograph #19 (19 / 30)"
        classification = "TLP:CLEAR"
        mitre_technique = "T1110.002"
    strings:
        $api = "/api/server.php" ascii wide
        $a1 = "\"action\":\"login\"" ascii wide
        $a2 = "\"action\":\"getTask\"" ascii wide
        $a3 = "\"action\":\"sendCrack\"" ascii wide
        $a4 = "\"action\":\"sendProgress\"" ascii wide
        $hc1 = "hashcat" ascii wide nocase
        $hc2 = "clientSignature" ascii wide
    condition:
        filesize < 20MB and $api and 2 of ($a*) and 1 of ($hc*)
}
```

odin_bruteratel_mythic_plugx_hashtopolis.rules

SURICATA

```
# Odin by Brandefense - Report #19 Suricata 7.x Evasive C2, Multi-Port Hubs & Hashtopolis Suite
# Classification: TLP:CLEAR
# Scope Note: Covers BruteRatel C4 TLS/HTTP profiles, DarkSword & Overlord TLS C2 certificates,
#             Mythic modular agent check-ins, Hashtopolis /api/server.php JSON GPU worker protocol,
#             and multi-port CobaltStrike Murmur/Major certificates.

# 1. BruteRatel C4 Dedicated TLS Certificate or Default Ratel FQDN Placeholder
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-C2 BruteRatel C4 TLS Server Certificate CN=brute.bzh.ovh
or CN=FQDN"; flow:established,from_server; tls.cert_subject; content:"CN="; fast_pattern; pcre:"/
CN=(brute\.bzh\.ovh|FQDN)(,|$)/"; metadata:created_at 2026_11_18, mitre_software_id S1063; classtype:command-and-
control; sid:9519001; rev:1;)

# 2. BruteRatel C4 Badger Malleable HTTP Telemetry Check-In JSON Keys (behind TLS inspection or HTTP)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-C2 BruteRatel C4 Badger HTTP Check-In JSON Framing";
flow:established,to_server; http.method; content:"POST"; http.request_body; content:"\"b_arch\""; fast_pattern;
pcre:"/\"(c_name|d_name|p_id|b_ver)\"\s*:/"; metadata:created_at 2026_11_18, mitre_software_id S1063;
classtype:command-and-control; sid:9519002; rev:1;)

# 3. DarkSword Multi-Port C2 Confirmed TLS Server Certificate Cluster
alert tls $EXTERNAL_NET [443,8443,8888] -> $HOME_NET any (msg:"ODIN-C2 DarkSword Confirmed C2 TLS Certificate
Observed"; flow:established,from_server; tls.cert_subject; content:"CN="; fast_pattern; pcre:"/CN=(shidiqi8\.top|
shidiqi9\.top|qizaiqizai\.icu)(,|$)/"; metadata:created_at 2026_11_18, mitre_technique_id T1071_001;
classtype:command-and-control; sid:9519003; rev:1;)

# 4. Overlord C2 Confirmed .invalid Self-Signed TLS Server Certificates
alert tls $EXTERNAL_NET [443,1080,5173,8443,9000] -> $HOME_NET any (msg:"ODIN-C2 Overlord TLS Server Certificate
```

```
host-hex.invalid"; flow:established,from_server; tls.cert_subject; content:"CN="; fast_pattern; pcre:"/CN=host-
[0-9a-f]{16}\.invalid(,|$)/"; metadata:created_at 2026_11_18, mitre_technique_id T1071_001; classtype:command-
and-control; sid:9519004; rev:1;)

# 5. Mythic Modular C2 Agent (Apollo / Poseidon / Apfell) HTTP API Check-In Pattern
alert http $HOME_NET any -> $EXTERNAL_NET [80,443,4443,7443,8443] (msg:"ODIN-C2 Mythic Modular Agent HTTP Check-
In Message"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/agent_message";
fast_pattern; http.request_body; bsize:>47; pcre:"/^[A-Za-z0-9+\/=]{48,}$/"; metadata:created_at 2026_11_18,
mitre_technique_id T1071_001; classtype:command-and-control; sid:9519005; rev:1;)

# 6. Hashtopolis Distributed GPU Hash-Cracking Worker API Protocol (/api/server.php)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-CRACK Hashtopolis Distributed GPU Agent API Tasking (/
api/server.php)"; flow:established,to_server; http.method; content:"POST"; http.uri; content:"/api/server.php";
fast_pattern; http.request_body; pcre:"/\"(action|request)\"\s*:\s*\"(register|login|getTask|getChunk|
sendProgress|sendCrack)\"/"; metadata:created_at 2026_11_18, mitre_technique_id T1110_002; classtype:policy-
violation; sid:9519006; rev:1;)

# 7. Multi-Port CobaltStrike Consolidation TLS Certificate (Major Cobalt Strike or Murmur v2)
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-C2 CobaltStrike Default or Multi-Port Murmur v2 TLS
Certificate"; flow:established,from_server; tls.cert_subject; content:"CN="; fast_pattern; pcre:"/CN=(Murmur
Autogenerated Certificate v2|Major Cobalt Strike)(,|$)/"; metadata:created_at 2026_11_18, mitre_software_id
S0154; classtype:command-and-control; sid:9519007; rev:1;)
```

sentinel_evasive_c2_callstack_and_kerberoast_burst.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Report #19 Microsoft Sentinel & Defender XDR Hunting Query
// Correlates Confirmed/Watchlist Evasive C2 & Hashtopolis IPs, Unbacked Memory / Remote Thread
// Injection Anomalies (DeviceEvents API telemetry only; Sysmon call-trace coverage is in the Sigma rule), Active
Directory Kerberoast RC4 (0x17) Ticket Bursts, and Unauthorized
// Hashtopolis / Hashcat Distributed GPU Cracking Activity.
let Lookback = 14d;
let ConfirmedEvasiveIPs = dynamic([
    "161.35.73.220", "191.96.31.13", "78.90.248.224", "163.172.45.158",
    "200.105.111.26", "194.26.192.57", "23.95.217.154", "5.188.86.227",
    "138.197.31.70", "77.72.18.83", "47.239.109.26"
]);
// Populate WatchlistHubAndClusterIPs from your own CTI watchlist feed (these addresses are withheld in the
report)
let WatchlistHubAndClusterIPs = dynamic([]);
let SacrificialHostBinaries = dynamic(["rundll32.exe", "dllhost.exe", "werfault.exe", "svchost.exe",
"notepad.exe", "regsvr32.exe"]);
let ApprovedAuditRigHosts = dynamic(["SRV-PENTEST-GPU01", "SRV-REDTEAM-SIM01"]);
let ApprovedVulnScanAccounts = dynamic(["SVC-VULNSCAN01", "SVC-AD-AUDIT01"]);
union isfuzzy=true
(
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback) and ActionType == "ConnectionSuccess"
    | where RemoteIP in (ConfirmedEvasiveIPs) or RemoteIP in (WatchlistHubAndClusterIPs)
    | extend HuntSignal = iff(RemoteIP in (ConfirmedEvasiveIPs), "Confirmed_Evasive_C2_Or_Hashtopolis_Egress",
"Watchlist_MultiListener_Hub_Or_Hashtopolis_Cluster_Egress"),
             ArtifactDetail = strcat(RemoteIP, ":", tostring(RemotePort), " (", InitiatingProcessFolderPath, ")")
    | project Timestamp, DeviceName, HuntSignal, ActorOrProcess = InitiatingProcessFileName, ArtifactDetail
),
(
    DeviceEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAuditRigHosts))
    | where ActionType in ("CreateRemoteThreadApiCall", "QueueUserApcRemoteApiCall",
"NtAllocateVirtualMemoryApiCall")
    | where FileName in~ (SacrificialHostBinaries)
        and (InitiatingProcessFolderPath has_any (@"\AppData\Local\Temp\", @"\AppData\Roaming\",
@"\ProgramData\", @"\Windows\Temp\") or isempty(InitiatingProcessVersionInfoCompanyName))
    | join kind=inner (
        DeviceNetworkEvents
        | where Timestamp >= ago(Lookback) and ActionType == "ConnectionSuccess"
        | where RemoteIP in (ConfirmedEvasiveIPs) or RemoteIP in (WatchlistHubAndClusterIPs)
        | project NetTime = Timestamp, DeviceId, InitiatingProcessId, RemoteIP, RemotePort, NetProc =
InitiatingProcessFileName
```

```
    ) on DeviceId, InitiatingProcessId
    | where abs(datetime_diff("minute", Timestamp, NetTime)) <= 10
    | extend HuntSignal = "Evasive_Implant_Sacrificial_Injection_With_C2_Beacon",
             ArtifactDetail = strcat(InitiatingProcessFileName, " -> ", FileName, " | Egress ", RemoteIP, ":",
tostring(RemotePort))
    | project Timestamp, DeviceName, HuntSignal, ActorOrProcess = InitiatingProcessFileName, ArtifactDetail
),
(
    SecurityEvent
    | where TimeGenerated >= ago(Lookback) and EventID == 4769
    | extend Timestamp = TimeGenerated
    | where TicketEncryptionType == "0x17"
    | where ServiceName != "krbtgt" and not(ServiceName endswith "$")
    | where not(TargetUserName endswith "$") and not(TargetUserName in~ (ApprovedVulnScanAccounts))
    | summarize DistinctSPNs = dcount(ServiceName), SampleSPNs = make_set(ServiceName, 6), FirstReq =
min(Timestamp)
        by bin(Timestamp, 5m), DeviceName = Computer, TargetUserName, IpAddress
    | where DistinctSPNs >= 5
    | extend HuntSignal = "Kerberoast_RC4_0x17_SPN_Harvest_Burst",
             ArtifactDetail = strcat("User=", TargetUserName, " SrcIP=", IpAddress, " DistinctSPNs=",
tostring(DistinctSPNs), " Sample=", tostring(SampleSPNs))
    | project Timestamp = FirstReq, DeviceName, HuntSignal, ActorOrProcess = TargetUserName, ArtifactDetail
),
(
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAuditRigHosts))
    | where (FileName has_any ("hashcat", "hashtopolis") or ProcessCommandLine has_any ("/api/server.php", "-m
13100", "-m 18200", "-m 1000", "-m 2100", "hashtopolis.zip"))
    | extend HuntSignal = "Unauthorized_Hashtopolis_Worker_Or_Hashcat_Kerberoast_Run",
             ArtifactDetail = strcat(FolderPath, " | ", ProcessCommandLine)
    | project Timestamp, DeviceName, HuntSignal, ActorOrProcess = FileName, ArtifactDetail
)
| summarize
    FirstSeen = min(Timestamp),
    LastSeen = max(Timestamp),
    EventCount = count(),
    SampleDetails = make_set(ArtifactDetail, 5)
    by DeviceName, HuntSignal, ActorOrProcess
| order by EventCount desc
```

sigma_unbacked_memory_syscall_and_kerberoast_export.yml

SIGMA

```
title: Unbacked Memory CallStack Accessing LSASS or Sacrificial Host Process
id: 9136ad79-1901-47ef-94e0-fd924bceb901
status: stable
description: Detects Sysmon ProcessAccess (EventID 10) where the CallTrace originates from unbacked RWX/RX memory
(UNKNOWN module), characteristic of BruteRatel C4 Badger, Havoc Demon, and Mythic Apollo indirect-syscall and
Ekko sleep-masking implants.
author: Odin by Brandefense
date: 2026-11-18
tags:
  - attack.defense-evasion
  - attack.credential-access
  - attack.t1055
  - attack.t1003.001
logsource:
  category: process_access
  product: windows
detection:
  selection_target:
    TargetImage|endswith:
      - '\lsass.exe'
      - '\rundll32.exe'
      - '\dllhost.exe'
      - '\werfault.exe'
  selection_unbacked_trace:
    CallTrace|contains: 'UNKNOWN('
```

```
    GrantedAccess|contains:
      - '0x1FFFFF'
      - '0x1F3FFF'
      - '0x143A'
      - '0x1010'
  # No directory-based exclusion: allowlist exact, signed SourceImage paths of approved profilers or agents here.
  filter_approved_agents:
    SourceImage:
      - 'C:\Program Files\ExampleVendor\Profiler\profiler-agent.exe'
  condition: selection_target and selection_unbacked_trace and not filter_approved_agents
falsepositives:
  - Custom .NET or JVM administrative agents injecting JIT-compiled profiler stubs; allowlist explicit signed
SourceImage paths.
level: critical
---
title: Active Directory Kerberoast RC4-HMAC Service Ticket Extraction
id: 9136ad79-1902-47ef-94e0-fd924bceb902
status: stable
description: Detects Kerberos TGS ticket requests (EventID 4769) requesting legacy RC4-HMAC encryption (0x17) for
non-machine Active Directory service accounts, feeding offline Hashtopolis GPU cracking clusters.
author: Odin by Brandefense
date: 2026-11-18
tags:
  - attack.credential-access
  - attack.t1558.003
logsource:
  product: windows
  service: security
detection:
  selection_tgs_rc4:
    EventID: 4769
    TicketEncryptionType: '0x17'
    TicketOptions:
      - '0x40810000'
      - '0x40800000'
      - '0x40810010'
  filter_machine_and_krbtgt:
    ServiceName|endswith: '$'
  filter_krbtgt_exact:
    ServiceName: 'krbtgt'
  filter_approved_scanners:
    TargetUserName:
      - 'SVC-VULNSCAN01'
      - 'SVC-AD-AUDIT01'
  condition: selection_tgs_rc4 and not filter_machine_and_krbtgt and not filter_krbtgt_exact and not
filter_approved_scanners
falsepositives:
  - Legacy ERP or SQL service accounts explicitly configured without AES256 Kerberos support; migrate SPNs to
gMSA with AES-only encryption.
level: high
---
title: Unauthorized Hashtopolis Distributed GPU Worker or Hashcat Ticket Cracking Execution
id: 9136ad79-1903-47ef-94e0-fd924bceb903
status: stable
description: Detects execution of the dual-use Hashtopolis Python/C# distributed agent polling /api/server.php or
Hashcat invoked against Kerberoast (-m 13100), AS-REP (-m 18200), NTLM (-m 1000), or DCC2 (-m 2100) hash modes on
non-audit hosts.
author: Odin by Brandefense
date: 2026-11-18
tags:
  - attack.credential-access
  - attack.t1110.002
logsource:
  category: process_creation
  product: windows
detection:
  selection_hashtopolis_agent:
    CommandLine|contains:
      - '/api/server.php'
      - 'hashtopolis.zip'
      - '--voucher'
  selection_hashcat_ad_modes:
```

```
    Image|endswith:
      - '\hashcat.exe'
      - '\hashcat64.exe'
    CommandLine|contains:
      - '-m 13100'
      - '-m 18200'
      - '-m 1000'
      - '-m 2100'
      - '$krb5tgs$23$'
  filter_authorized_rig:
    ComputerName:
      - 'SRV-PENTEST-GPU01'
      - 'SRV-REDTEAM-SIM01'
  condition: (selection_hashtopolis_agent or selection_hashcat_ad_modes) and not filter_authorized_rig
falsepositives:
  - Authorized red-team password-auditing GPU servers; restrict exceptions strictly to named audit hosts such as
SRV-PENTEST-GPU01.
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/evasive-c2-and-dual-use-cracking/
Short Read · brandefense.io/odin-research/blog/most-brute-ratel-c4-hits-are-vpn-exits/
Defense Kit · brandefense.io/odin-research/defense-kit/