ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 44 / 30 · COMMODITY MALWARE

# IoT Botnet Infrastructure

## Separating Kaiji Candidates from Router Peers

How 2,851 examined endpoints across five botnet families split into 1,696 withheld telecom peers, 195 commercial-VPS Kaiji candidates, and deployable network and Linux hunts

Prepared November 2026
Full Report · Public edition · Version 1.1

**2,587**
Distinct physical hosts behind the 2,851 examined endpoints, with zero hosts shared across families

**0 / 1,677**
Central C2 servers found across 840 Mozi and 837 Hajime peer-to-peer DHT endpoints

**661 / 892**
Mirai endpoints on dynamic broadband (598) and regional fixed or cable telecom carriers (63)

**0 / 195**
Confirmed Kaiji C2 addresses to block, with all 195 commercial-VPS and ISP Kaiji endpoints kept for hunting

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

How 2,851 examined endpoints across five botnet families split into 1,696 withheld telecom peers, 195 commercial-VPS Kaiji candidates, and deployable network and Linux hunts


| | | |
|---|---|---|
| 01 | Executive Summary | 04 |
| 02 | Four Infrastructure Groups, One Analytical Model | 07 |
| 03 | Mozi and Hajime Peer-to-Peer DHT Swarms | 10 |
| 04 | Mirai and Gafgyt Edge Routers on Telecom Networks | 13 |
| 05 | Kaiji Golang SSH Botnet on Commercial VPS Hosting | 16 |
| 06 | Feed Labels Against Confirmed C2 Infrastructure | 19 |
| 07 | What Defenders Should Do | 22 |
| 08 | Detection Rules | 24 |
| 09 | Indicators | 27 |
| 10 | Scope and Limits | 30 |
| 11 | Sources | 32 |
| A | Indicators of compromise (7) | 34 |
| B | Detection rules (4 files) | 35 |



IN SHORT

Odin by Brandefense reviewed an inventory of 2,851 endpoints on 2,587 hosts through 20 November 2026 across five botnet families: Mozi, Mirai, Hajime, Kaiji, and Gafgyt. Upstream feeds linked these families to 280,027 global sightings, and 1,831 examined records date before 6 November 2026. Nearly three in five endpoints (1,696, or 59.5%) sit on subscriber or regional telecom networks, where feed labels alone do not confirm infection. Zero endpoints met the confirmation bar for IP blocking. Enforce key-only SSH and keep telecom peer lists out of firewall blocks.

01

CHAPTER 01

# Executive Summary

IN SHORT

Odin by Brandefense reviewed an inventory of 2,851 endpoints on 2,587 hosts through 20 November 2026 across five botnet families: Mozi, Mirai, Hajime, Kaiji, and Gafgyt. Upstream feeds linked these families to 280,027 global sightings, and 1,831 examined records date before 6 November 2026. Nearly three in five endpoints (1,696, or 59.5%) sit on subscriber or regional telecom networks, where feed labels alone do not confirm infection. Zero endpoints met the confirmation bar for IP blocking. Enforce key-only SSH and keep telecom peer lists out of firewall blocks.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

# 01 Executive Summary

IoT and Linux DDoS botnets do not all run on the same infrastructure. Peer-to-peer router worms turn home and small-office gateways into temporary lookup nodes. Linux SSH botnets rent commercial virtual private servers (VPS) to stage binaries and send attack commands. Defenders can use cyber threat intelligence to review feed matches before permanent firewall blocks cut off broadband customers while missing internal peer traffic.

1. **Mozi and Hajime peer-to-peer meshes have no central C2 server to block.**
   Mozi accounts for 215,742 feed-labelled sightings and 840 examined endpoints, with 84.3% sitting on dynamic telecom networks and zero dedicated C2 servers. Hajime adds 16,292 feed sightings and 837 examined endpoints, and all 20 feed-confirmed entries sit on Spanish fiber and Jordanian telecom networks as decentralized peers.

2. **Regional and dynamic telecom networks explain all 66 feed-confirmed Mirai endpoints.**
   Out of 47,711 Mirai feed sightings, we examined 892 endpoints across 680 hosts. Dynamic broadband and mobile networks hold 598 of them, including three marked confirmed by upstream feeds. The other 63 of the 66 feed-confirmed endpoints sit on Ethiopian, Taiwanese, and Indian fixed or cable broadband carriers rather than commercial cloud servers.

3. **Go-based Kaiji favors commercial VPS hosting over consumer broadband.**
   Unlike router swarms, Kaiji places 186 of its 195 examined endpoints outside dynamic subscriber networks and clusters 63.6% of them on ports 10081, 808, and 8080. One Kaiji endpoint on a commercial VPS remains uncorroborated as C2, as do all 87 Gafgyt entries.

## Three Decisions for Leaders

1. **Keep uncorroborated feed lists off automatic firewall blocks.** Zero endpoints in this set met the confirmation bar for permanent IP blocking. Route Mozi, Mirai, Hajime, Kaiji, and Gafgyt feed matches to network and host hunting instead of automated perimeter blocks.
2. **Isolate IoT segments and alert on peer-to-peer UDP traffic.** Place IP cameras, routers, and building gateways on dedicated VLANs. Alert when any host on those segments sends BitTorrent Distributed Hash Table (DHT) queries or fetches BusyBox shell scripts.
3. **Require key-only SSH sign-in on every Linux server.** Disable password login for `root` across cloud and on-premises Linux hosts. Monitor for binaries running from `/usr/bin/lib/` and for SSH traffic on non-standard port 24.

## The Study in Numbers

| Number | What it is |
|---|---|
| **280,027 / 2,851** | Global feed-labelled sightings across five botnet families, and the 2,851 endpoints we examined |
| **2,587** | Distinct physical hosts behind the 2,851 examined endpoints, with zero hosts shared across families |
| **1,696 (59.5%)** | Examined endpoints on dynamic subscriber ISPs (1,613) and regional telecom or carrier networks (83) |
| **0 / 1,677** | Central C2 servers found across 840 Mozi and 837 Hajime peer-to-peer DHT endpoints |
| **661 / 892** | Mirai endpoints on dynamic broadband (598) and regional fixed or cable telecom carriers (63) |
| **0 / 195** | Confirmed Kaiji C2 addresses to block, with all 195 commercial-VPS and ISP Kaiji endpoints kept for hunting |
| **7** | Published indicators in Appendix A: 7 JARM fingerprints to hunt (0 IPv4 addresses published) |

**How to use this report.**

Security leaders can read Chapters 1, 6, and 7. Network and Linux teams should read Chapters 3, 4, 5, and 7. SOC and detection engineers should start with Chapters 5, 8, and 9. CTI analysts will find hosting and TLS detail in Chapters 2 through 6.

02

CHAPTER 02

# Four Infrastructure Groups, One Analytical Model

IN SHORT

None of the 2,587 hosts appears in more than one botnet family. We group the five families into four functional roles as an analytical model for defenders, not as a single operator.

INSIDE THIS CHAPTER

Where the endpoints sit 08

# 02 Four Infrastructure Groups, One Analytical Model

Feed lists often mix decentralized router meshes, edge loader fleets, commercial Linux C2 servers, and benign port collisions into one flat table. Blocking every row treats a subscriber home router like a rented cloud server, cutting off the next broadband customer after DHCP lease rotation.

Comparing hosts across the 2,851-endpoint set shows zero shared hosts between Mozi, Mirai, Hajime, Kaiji, and Gafgyt. Shared TLS fingerprints appear across Mozi, Mirai, and Hajime; common JARMs alone do not confirm infection or establish the cause of TLS similarity. Figure 1 presents the four groups as an analytical model of infrastructure roles and defender actions.

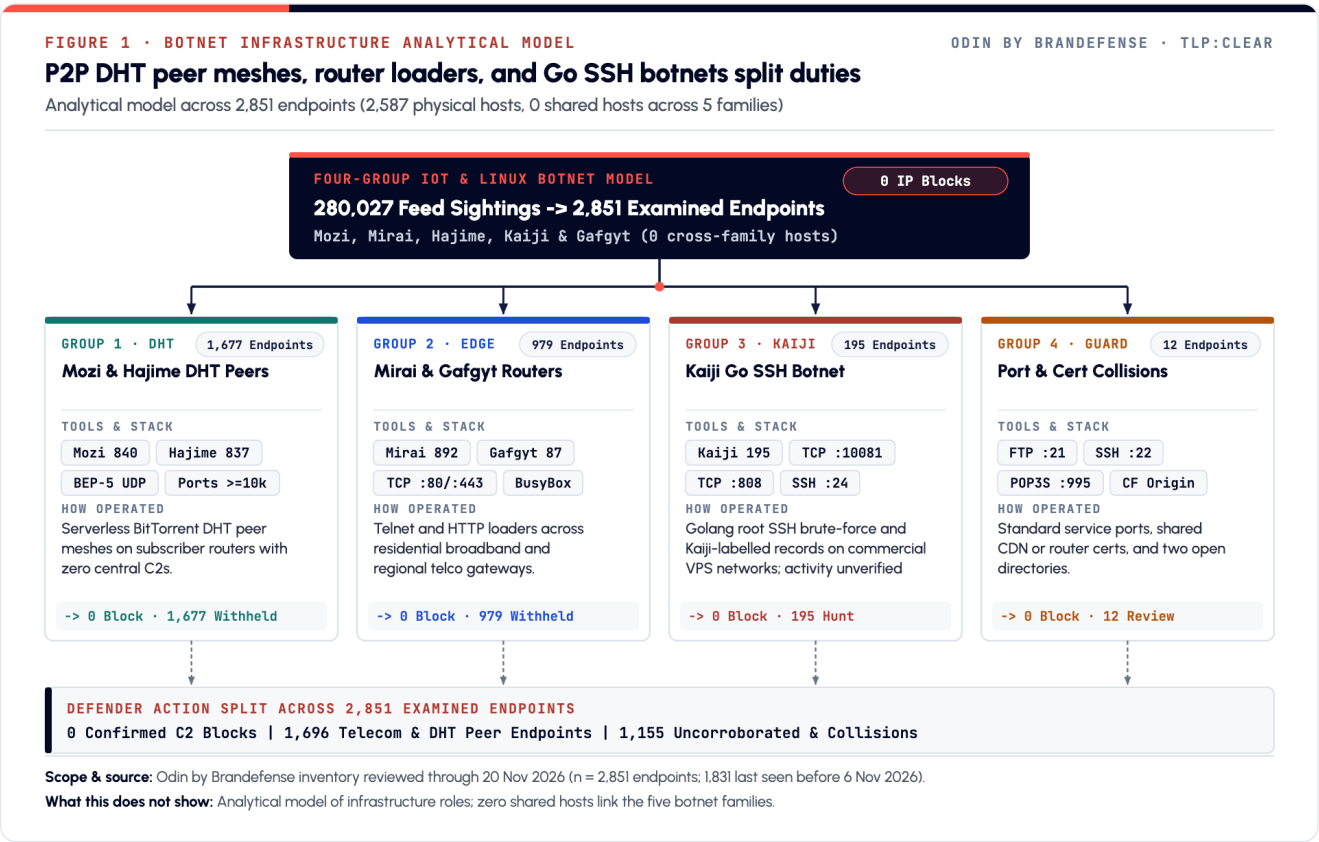



Figure 1: P2P DHT peer meshes, router loaders, and Go SSH botnets split duties (analytical model)

| Group | Families | Examined endpoints | Physical hosts | Published addresses | Defender action |
|---|---|---|---|---|---|
| Peer-to-peer DHT meshes | Mozi, Hajime | 1,677 | 1,648 | 0 | Hunt UDP DHT and verify port 5555 on IoT VLANs |
| Edge router loader fleets | Mirai, Gafgyt | 979 | 757 | 0 | Hunt `/bins/` and BusyBox fetches; patch edge devices |
| Commercial-VPS Linux SSH botnet | Kaiji | 195 | 182 | 0 | Enforce key-only SSH; hunt ports 10081, 808, and 24 |
| Port and certificate collisions (subset) | Mozi, Mirai, Hajime, Kaiji | 12 | 12 | 0 | Keep FTP, SSH, POP3S, and CDN origin certs out of blocks |

The 12 collision and open-directory rows are counted inside the family totals above (`1,677 + 979 + 195 = 2,851` endpoints across `1,648 + 757 + 182 = 2,587` physical hosts).

## Where the endpoints sit

Across all 2,851 endpoints, China leads with 1,027 endpoints (36.0%), followed by India (140, 4.9%) and the United States (129, 4.5%). Russia (128, 4.5%), Indonesia (115, 4.0%), Taiwan (104, 3.6%), and the Philippines (98, 3.4%) come next. Brazil (68, 2.4%), Ethiopia (66, 2.3%), and Hong Kong (52, 1.8%) round out the top ten.

| Network provider | Role | Endpoints | Main families |
|---|---|---|---|
| CHINA UNICOM China169 Backbone | Broadband and mobile ISP | 833 | Mozi (639), Mirai (179), Hajime (10), Gafgyt (5) |
| CHINANET BACKBONE | Broadband and mobile ISP | 129 | Mirai (124), Mozi and Hajime (5) |
| Globe Telecom Inc. | Mobile and broadband ISP (Philippines) | 94 | Mirai (94) |
| Taiwan Fixed Network, Telco and Network Service Provider. | Fixed-line telco (Taiwan) | 73 | Mirai (73) |
| EthioNet and Ethio Telecom | National telecom networks (Ethiopia) | 66 | Mirai (34 on EthioNet, 32 on Ethio Telecom) |
| PT Telekomunikasi Indonesia | National broadband ISP (Indonesia) | 53 | Mirai (34), Hajime (19) |
| Spartan Host Ltd | Commercial VPS hosting (United States) | 38 | Kaiji (38) |
| NetLab Global | Commercial VPS hosting (United States, Hong Kong) | 14 | Kaiji (14, including one feed-confirmed US VPS candidate and historical records for ports 24, 60000, and 60001 across three host tokens at different times) |

Consumer broadband networks dominate Mozi, Mirai, and Hajime, while commercial VPS providers such as Spartan Host Ltd and NetLab Global appear almost exclusively under Kaiji.

CHAPTER 03

# Mozi and Hajime Peer-to-Peer DHT Swarms

IN SHORT

Mozi (840 examined endpoints) and Hajime (837 examined endpoints) use serverless BitTorrent DHT peer meshes, and feed labels on telecom endpoints do not by themselves confirm router infection. Neither family offers a central C2 address to block, so defenders must watch for bencoded UDP peer traffic and verify exposed ports instead.

INSIDE THIS CHAPTER

Mozi: 93.8% ephemeral high ports on subscriber broadband 10

Hajime: decentralized DHT peers and port 5555 listings 11

# 03 Mozi and Hajime Peer-to-Peer DHT Swarms

Mozi and Hajime avoid central command servers by building on the BitTorrent Mainline DHT protocol (BEP 5). In public analyses of these families, peer routers, cameras, or digital video recorders join a Kademlia swarm over UDP (T1584.005) and relay signed configuration updates. Figure 2 maps all five families to their transport profiles and matching detections.

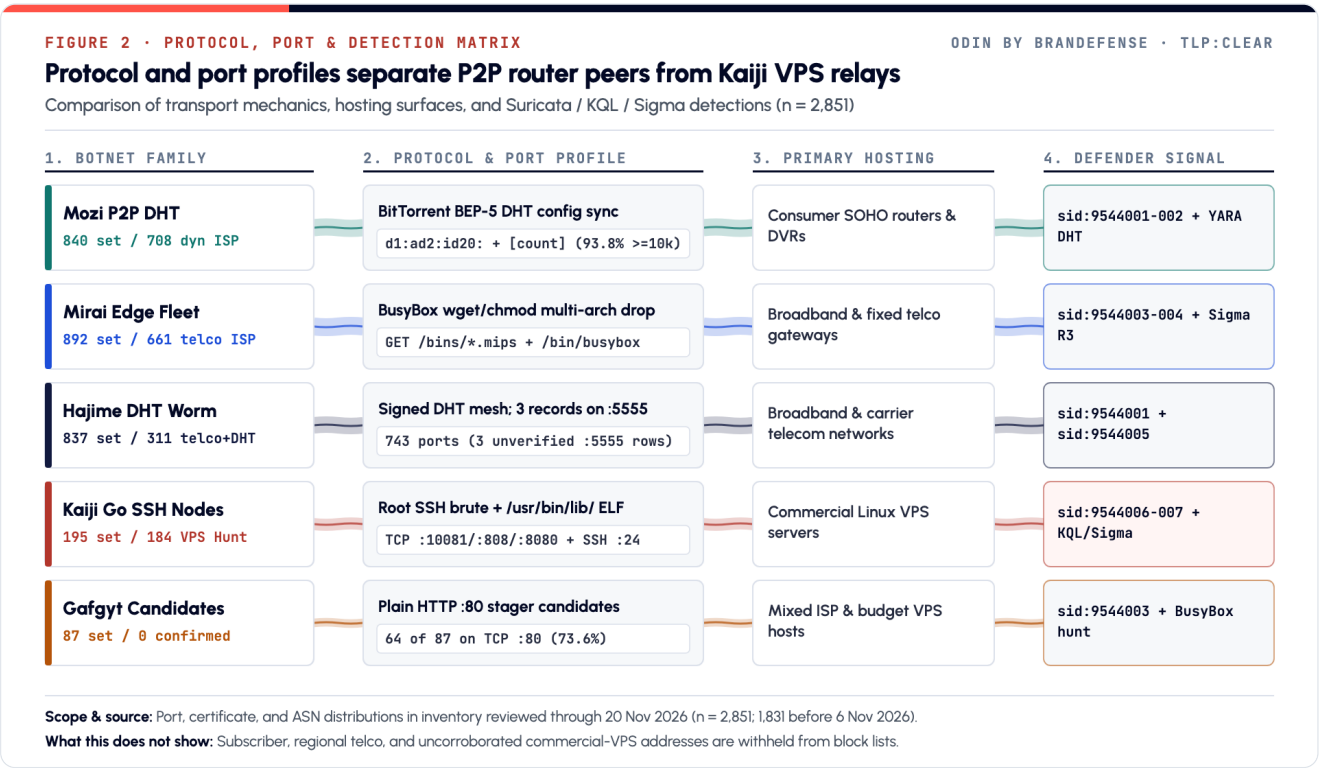



Figure 2: Protocol and port profiles separate P2P router peers from Kaiji VPS relays

## Mozi: 93.8% ephemeral high ports on subscriber broadband

Mozi accounts for 215,742 global feed sightings across 89 countries (77.0% of all sightings in this study). Across our 840 examined endpoints on 825 hosts, 788 endpoints (93.8%) use ports 10000 or higher across 712 distinct port numbers (T1571). Port 443 holds 38 endpoints, port 80 holds 7, and port 51769 holds 5.

Dynamic residential and mobile ISPs host 708 of the 840 Mozi endpoints (84.3%). China leads with 649 endpoints, including 639 on CHINA UNICOM China169 Backbone. India follows with 71 endpoints, including 34 on National Internet Backbone and 32 on Bharat Sanchar Nigam Ltd. Sweden (29), Pakistan (20), and Morocco (11) follow next. Seven Mozi endpoints on port 80 carried a confirmed label in upstream feeds on 13 November 2026, yet all seven sit on Pakistan Telecommuication company limited subscriber lines. All 825 Mozi host addresses are withheld.

On the wire, Mozi nodes send bencoded KRPC `find_node` queries ( `d1:ad2:id20:` ) and return signed config payloads carrying `[count]` and `[ss]` tags. On server or IoT VLANs where BitTorrent is prohibited, an outbound `d1:ad2:id20:` UDP packet flags DHT traffic for investigation; it does not by itself confirm infection.

## Hajime: decentralized DHT peers and port 5555 listings

Hajime accounts for 16,292 global sightings across 149 countries and 837 examined endpoints on 823 hosts. Public analyses establish that Hajime has no central C2 server; its author pushes signed modules directly into the BitTorrent Mainline DHT swarm.

| Feature | Hajime count and share |
|---|---|
| Endpoints on ports 10000 and above | 649 of 837 (77.5%), across 743 distinct port numbers |
| Plain HTTP port 80 / TLS 443 / HTTP 8080 | 35 on 80, 3 on 443, 3 on 8080 |
| Unverified TCP port 5555 listings | 3 Hajime-labelled records (port 5555 is commonly associated with ADB, but service banners were not captured) |
| Leading countries | Russia (116), Indonesia (71), India (58), Bangladesh (39), Ukraine (37), South Korea (35), Spain (33), Brazil (28) |
| Dynamic and carrier telecom endpoints | 291 dynamic ISP (34.8%) plus 20 carrier and telecom DHT peers |

Upstream feeds marked 20 Hajime endpoints as confirmed on 13 and 14 November 2026. Nineteen listen on port 80 inside one Spanish subnet of LYNTIA NETWORKS S.A., a wholesale fiber carrier. The twentieth listens on port 443 on Batelco Jordan. It presents the same self-signed device certificate and JARM (`04b06b00004b04b04b04b06b04b04be3b0c44298fc1c149afbf4c8996fb924`) seen on two dynamic-ISP endpoints on Iran Telecommunication Company PJS. Because Hajime uses no central C2 and all 20 hosts belong to carrier or telecom networks, all 20 addresses are withheld.

CHAPTER 04

# Mirai and Gafgyt Edge Routers on Telecom Networks

IN SHORT

Mirai (892 examined endpoints) and Gafgyt (87 examined endpoints) spread BusyBox shell stagers across exposed routers and gateways. Of 66 Mirai endpoints marked confirmed by upstream feeds, 63 sit on regional fixed, cable, or fiber carriers and three sit on dynamic telecom networks. Isolate IoT VLANs and keep telecom peers off block lists.

INSIDE THIS CHAPTER

Why 66 feed-confirmed Mirai endpoints belong on watchlists, not block lists 13

Gafgyt candidates 14

# 04 Mirai and Gafgyt Edge Routers on Telecom Networks

Mirai and Gafgyt scan for exposed Telnet, SSH, and router web interfaces that defenders can find with external attack surface management (T1110.001, T1071.001). A shell one-liner invokes `/bin/busybox` with `wget` or `tftp` and `chmod 777` to pull an architecture-specific ELF binary (`/bins/*.mips`, `.mpsl`, `.arm7`, `.sh4`). Figure 3 contrasts DHT gossip and BusyBox router staging with Kaiji's VPS-backed SSH workflow.

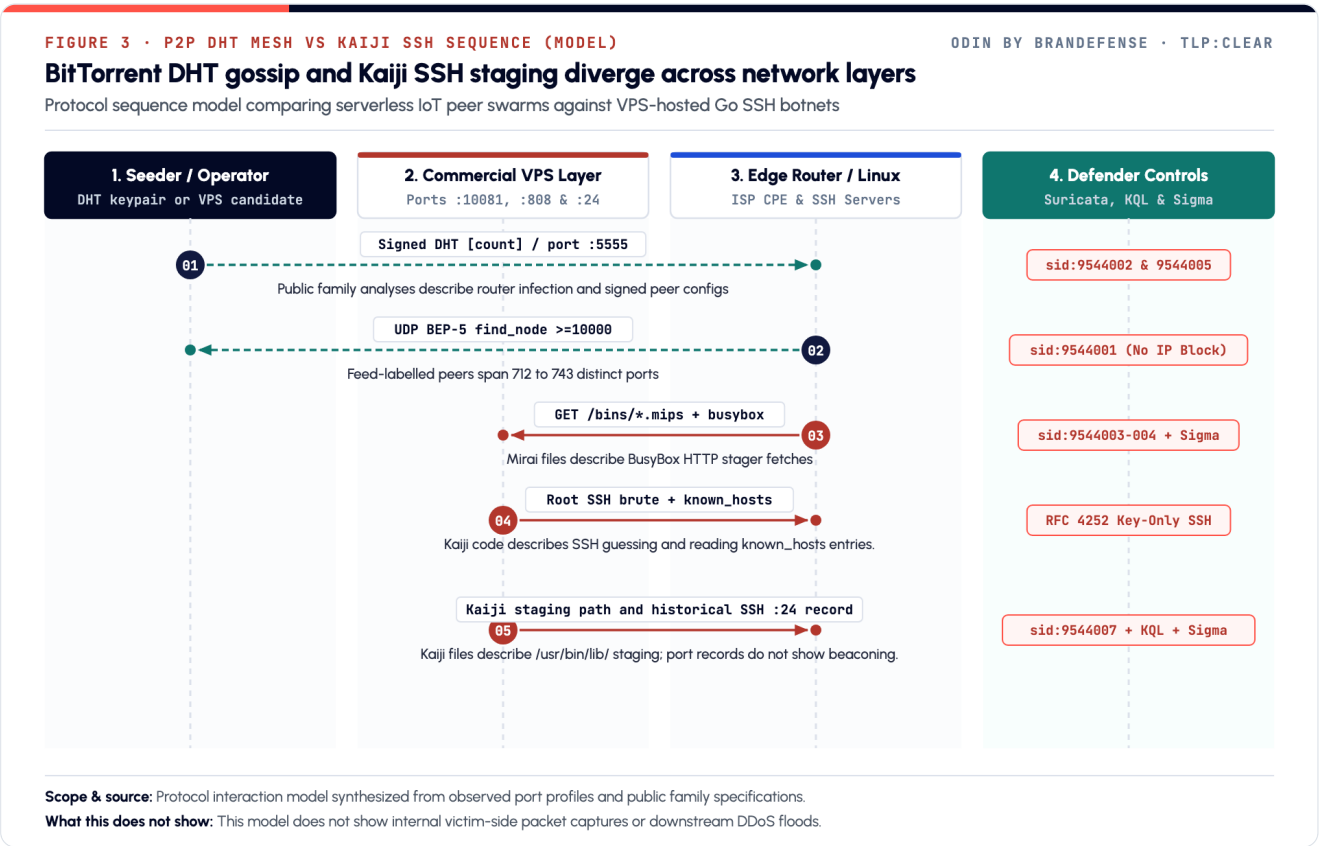



Figure 3: BitTorrent DHT gossip and Kaiji SSH staging diverge across network layers

## Why 66 feed-confirmed Mirai endpoints belong on watchlists, not block lists

Across 47,711 global Mirai sightings in 115 countries, we examined 892 endpoints on 680 hosts. High ports 10000 and above hold 762 endpoints (85.4%) across 714 distinct ports, while ports 443 and 80 hold 56 each. Leading countries are China (340), the Philippines (96), Taiwan (94), Ethiopia (66), Indonesia (42), Brazil (40), Albania (37), and South Africa (35).

Dynamic mobile and residential broadband backbones hold 598 Mirai endpoints (67.0%, including three marked confirmed by upstream feeds), and 231 more are uncorroborated or collision leads. Checking the six networks behind the other 63 of the 66 feed-confirmed Mirai endpoints (48 hosts) showed that all six are national fixed, cable, or fiber broadband carriers:

| Country | Telecom or broadband network | Endpoints | Hosts | Ports and TLS detail |
|---|---|---|---|---|
| Ethiopia | EthioNet ( `AS24757` ) | 34 | 19 | 18 on 443 and 16 on 80; all 18 TLS ports use the same CPE certificate seen on 4 Mozi-labelled endpoints, and 16 share JARM `21d19d00021d…` |
| Taiwan | Taiwan Fixed Network, Telco and Network Service Provider. | 19 | 19 | All 19 on 443, sharing JARM `04b03b00004b…` |
| Taiwan | Hoshin Multimedia Center Inc. | 6 | 6 | Plain HTTP port 80 on a cable broadband ISP |
| India | Atria Convergence Technologies Pvt. Ltd., | 2 | 2 | Plain HTTP port 80 on ACT Fibernet consumer fiber |
| India | Quadrant Televentures Limited | 1 | 1 | Plain HTTP port 80 on a regional broadband carrier |
| India | RailTel Corporation of India Ltd. | 1 | 1 | Plain HTTP port 80 on a national broadband carrier |

EthioNet is Ethiopia's national telecom carrier, sibling to the 32 Ethio Telecom rows already treated as subscriber broadband. Its JARM ( `21d19d00021d21d21c42d43d00000079c40eb3c1447424b28b311ae50d7d52` ) also appears on cable ISPs in Belize and Canada. Combined telecom networks account for 661 of the 892 Mirai endpoints (74.1%), and all 680 host addresses are withheld.

## Gafgyt candidates

All 87 Gafgyt feed sightings span 16 countries across 77 hosts, with 64 listening on port 80 (73.6%). Seven sit on dynamic ISPs, while 80 remain uncorroborated candidates. Those candidates sit in the United States (18), the Netherlands (10), Cambodia (9 on OpenNet ISP Cambodia), and the United Kingdom (8). Their upstream feed record dates end in September 2026, before the November 2026 review cutoff, and all 77 host addresses are withheld.

05

CHAPTER 05

# Kaiji Golang SSH Botnet on Commercial VPS Hosting

IN SHORT

Kaiji is a Go-based Linux SSH brute-force and DDoS botnet whose infrastructure looks nothing like Mozi or Mirai. Commercial VPS providers host the bulk of its 195 endpoints, and 63.6% cluster on ports 10081, 808, and 8080. One upstream-confirmed NetLab Global endpoint lacked a second C2 signal and is withheld, so defenders should enforce key-only SSH and hunt Kaiji ports and host artifacts.

INSIDE THIS CHAPTER

NetLab Global VPS candidates and structured port clusters 16

# 05 Kaiji Golang SSH Botnet on Commercial VPS Hosting

Written from scratch in Go, Kaiji guesses `root` SSH passwords on Linux servers (T1110.001, T1021.004). It copies disguised ELF binaries into `/usr/bin/lib/` under names such as `netstat` or `sshd` and persists via `/etc/init.d/`. It also reads `.ssh/known_hosts` for lateral SSH movement and opens raw sockets for spoofed `SYN_ACK`, TCP, and UDP floods (T1498). Figure 4 compares Kaiji's VPS port records against shared router TLS profiles.




Figure 4: Commercial Kaiji VPS port records contrast with embedded router TLS stacks

## NetLab Global VPS candidates and structured port clusters

Across 195 Kaiji endpoints on 182 physical hosts in 17 countries, only 39 distinct ports appear and just 9 endpoints (4.6%) sit on dynamic ISPs. Commercial VPS hubs lead in the United States (73 endpoints, 37.4%), Hong Kong (39, 20.0%), and unattributed blocks (24, 12.3%). China (19, 9.7%), Japan (11, 5.6%), and Singapore (9, 4.6%) follow next:

| Hosting provider | Kaiji endpoints | Detail |
|---|---|---|
| Spartan Host Ltd | 38 | Largest commercial VPS cluster (19.5% of Kaiji endpoints) |
| NetLab Global | 14 | 1 feed-confirmed port-80 VPS candidate plus 3 host tokens with historical records for ports 24, 60000, and 60001 at different times |
| FranTech Solutions, Oracle, Cogent, Cloudflare London | 24 | 6 endpoints each across cloud and VPS hosting |
| CTG Server Limited and CNSERVERS LLC | 10 | 5 endpoints each on commercial server hosting |

Three findings guide defense:

- **Upstream-confirmed VPS candidate (withheld).** Recorded on 13 November 2026 on NetLab Global in the United States, one Kaiji-labelled commercial-VPS record lists port 80 (T1583.003). Because it showed no independent C2 behavior or second adversary signal beyond the upstream feed label, its address is withheld from block lists.
- **Core operator ports 10081, 808, and 8080.** Three ports hold 124 of the 195 endpoints (63.6%): 10081 (59, 30.3%), 808 (49, 25.1%), and 8080 (16, 8.2%). Smaller clusters use 8888 (7), 60001 (6), 8087 (6), 23812 (5), 24 (4), 9527 (4), and 60000 (3). Ten Kaiji VPS endpoints share JARM `29d29d20d29d29d22c29d29d29d29d8e917db1de9d33002d70770504a55ef7`.
- **Historical port 24, 60000, and 60001 records.** Three Kaiji-labelled host tokens have historical records for ports 24, 60000, and 60001 at different times; the records do not establish a simultaneous port triad or bot control. Another NetLab Global host in Hong Kong has historical records for 8088 and 8089 at different times; they do not show both ports open together. Candidate addresses stay withheld for port and TLS hunting.

06

CHAPTER 06

# Feed Labels Against Confirmed C2 Infrastructure

IN SHORT

The 2,851 examined endpoints were drawn from 280,027 feed sightings; none yet meets the publication bar for an automatic IP block. Blocking raw IoT feed lists punishes broadband subscribers and breaks shared cloud services.

INSIDE THIS CHAPTER

Four reasons raw IoT botnet feeds misfire in firewalls 19

# 06 Feed Labels Against Confirmed C2 Infrastructure

Figure 5 compares upstream feed-labelled records and examined endpoints with corroborated C2 block counts across all five families.

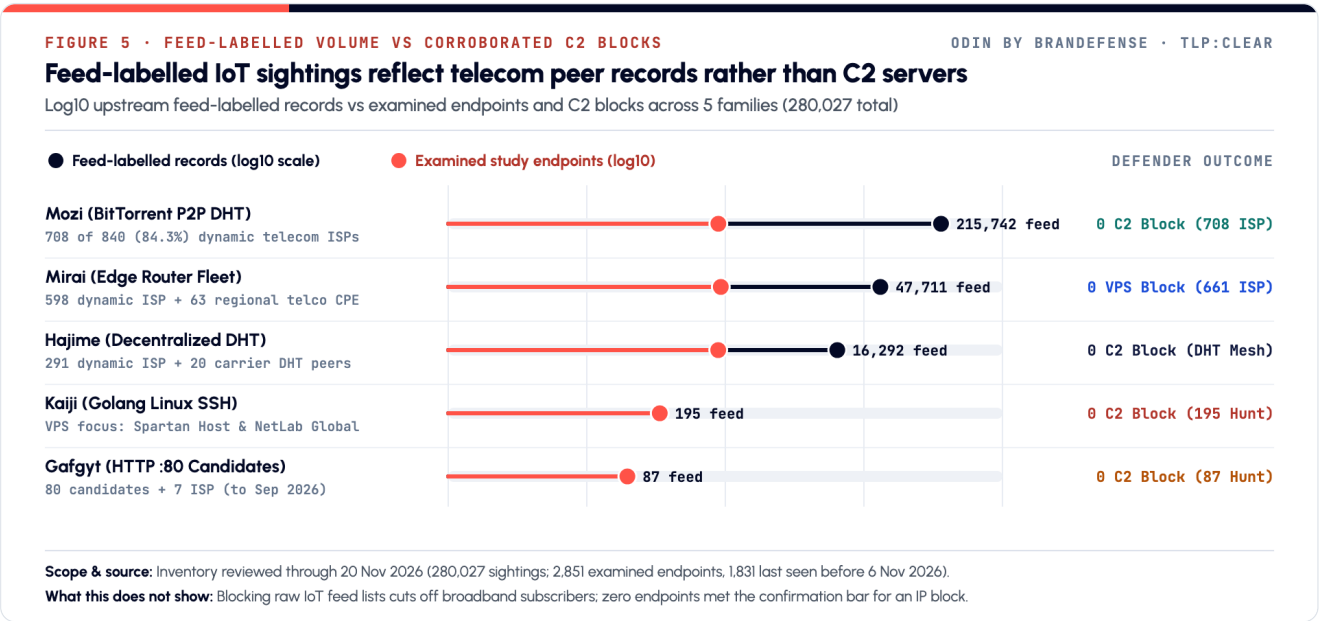



Figure 5: Feed-labelled IoT sightings reflect telecom peer records rather than C2 servers

| Family | Feed sightings | Examined | Physical hosts | Confirmed C2 (block) | Withheld peers, candidates, and collisions |
|---|---|---|---|---|---|
| Mozi | 215,742 | 840 | 825 | 0 | 840 (708 dynamic ISP, 36 suspected, 93 candidates, 3 collisions) |
| Mirai | 47,711 | 892 | 680 | 0 | 892 (598 dynamic ISP, 63 regional telco, 42 suspected, 185 candidates, 4 collisions or open dirs) |
| Hajime | 16,292 | 837 | 823 | 0 | 837 (291 dynamic ISP, 20 carrier or telco DHT peers, 17 suspected, 506 candidates, 3 collisions) |
| Kaiji | 195 | 195 | 182 | 0 | 195 (184 commercial VPS candidates, 9 dynamic ISP, 2 CDN collisions) |
| Gafgyt | 87 | 87 | 77 | 0 | 87 (80 historical candidates, 7 dynamic ISP) |

## Four reasons raw IoT botnet feeds misfire in firewalls

1. **Subscriber DHCP churn and regional carrier aliases.** Dynamic ISPs hold 1,613 endpoints (56.6%), and regional carriers hold 63 more Mirai rows. Blocking ISP leases cuts off downstream broadband subscribers after reassignment.
2. **Serverless DHT peer protocols.** Mozi and Hajime use signed BitTorrent DHT updates rather than central servers, so IP blocks on 20 Spanish or Jordanian peers cannot stop swarm propagation.
3. **Shared TLS fingerprints across families (120 endpoints across three JARMs).** Three JARM hashes span Mozi, Mirai, and Hajime despite zero shared hosts. Hash `3fd3fd0003fd3fd00042d42d000000020120996177a65431cde640fa58d2e8` appears on 52 endpoints (32 Mozi, 14

Mirai, 6 Hajime). Hash `04d02d00004d04d04c04d02d04d04d9674c6b4e623ae36cc2d998e99e2262e` appears on 44 endpoints (30 Mozi, 7 Mirai, 7 Hajime), and `29d14d00029d29d00029d14d29d29df26cd45df722aff084b72395f8d18d1b` appears on 24 (18 Mirai, 5 Hajime, 1 Mozi). Common JARMs alone do not confirm infection, prove why TLS fingerprints match, or identify C2 software.

4. **Service-port, CDN, and open-directory collisions (12 endpoints).** Collisions include FTP port 21 in Japan, SSH port 22 in Germany, and POP3S port 995 in China. Seven more match shared certificates (three `CloudFlare Origin Certificate` servers and four router vendor certs), while two Mirai candidates on 8080 and 8082 expose open HTTP directories.

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

- [ ] Monitor and restrict unexpected outbound traffic from Linux servers to primary Kaiji VPS operator ports (10081, 808, 8080, 8087, 8888) and relocated SSH port 24 at perimeter firewalls and proxies.
- [ ] Block inbound WAN access to Telnet (23), Android Debug Bridge (5555), and router management ports (7547, 8080, 8443).

### Linux servers and SSH

- [ ] Enforce key-only SSH sign-in (`PasswordAuthentication no` and `PermitRootLogin no`) in `/etc/ssh/sshd_config` across all Linux servers (RFC 4252), restricted to jump hosts or VPNs.
- [ ] Audit Linux hosts for binaries under `/usr/bin/lib/`, modified scripts in `/etc/init.d/`, and outbound SSH connections to port 24.

### IoT and edge segments

- [ ] Isolate IP cameras, DVRs, and branch SOHO routers on dedicated VLANs blocked from reaching internal server subnets.
- [ ] Alert on outbound UDP packets to ports 10000 and above carrying BitTorrent Mainline DHT queries from server or IoT VLANs.

### Detection and threat intelligence

- [ ] Deploy the Suricata, Sentinel KQL, Sigma, and YARA rules in Chapter 8 with exact-hostname allowlists for dedicated lab or jump hosts.
- [ ] Check autonomous-system ownership before blocking IoT feeds, keeping broadband ISPs, regional telcos, DHT peer lists, FTP/SSH/POP3S ports, and shared CDN certificates in hunting workflows.

CHAPTER 08

# Detection Rules

IN SHORT

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

INSIDE THIS CHAPTER

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

# 08 Detection Rules

The full rules are in Appendix B. During preparation, Suricata `7.0.12` (`suricata -T`) loaded all 7 signatures with 0 failures, and `lint_suricata.py` found 0 problems. The suite also passed `YARA-X` compilation (3 rules), `pySigma` validation (2 rules), and `Kusto.Language` semantic checks. Test and tune them against local traffic before automated isolation.

| Rule file | What it catches |
|---|---|
| `odin_iot_p2p_dht_mirai_and_kaiji_ssh_botnets.rules` | Suricata 7: seven signatures for BitTorrent DHT queries, Mozi `[count]` responses, Mirai `/bins/` and BusyBox stagers, ADB port 5555 shell commands, Kaiji `.sh` fetches, and SSH port 24 |
| `sentinel_iot_p2p_dht_and_kaiji_ssh_botnet_hunt.kql` | Microsoft Sentinel and Defender XDR: connections to Kaiji operator ports, high-port UDP peer fan-out, `/usr/bin/lib/` binaries, and BusyBox stagers |
| `sigma_linux_kaiji_ssh_persistence_and_iot_stager.yml` | Sigma (2 rules): Kaiji `/usr/bin/lib/` execution or `/etc/init.d/` staging, and BusyBox multi-architecture fetches |
| `odin_kaiji_golang_ssh_mirai_and_mozi_dht_artifacts.yar` | YARA (3 rules): Golang Kaiji ELF implants, Mozi DHT config binaries, and Mirai or Gafgyt BusyBox shell droppers |

## How the Suricata signatures work

| SID | Looks for | Why |
|---|---|---|
| 9544001 | Outbound UDP packet starting with `d1:ad2:id20:` with `1:q` and `1:y1:qe` on ports 1024:65535 | BitTorrent Mainline DHT lookup from IoT or server VLANs |
| 9544002 | Inbound UDP DHT response starting with `d1:rd2:id20:` and carrying `[count]` | Mozi signed peer configuration sync payload |
| 9544003 | Outbound HTTP `GET` for `/bins/` ending in `.arm`, `.arm5`, `.arm7`, `.mips`, `.mpsl`, `.x86`, `.spc`, `.sh4`, `.ppc`, or `.m68k` | Mirai and Gafgyt multi-architecture ELF stager fetch |
| 9544004 | HTTP request body containing `/bin/busybox` followed by `chmod` | Router command-injection exploit staging a binary |
| 9544005 | Inbound TCP port 5555 packet carrying `OPEN` and `shell:` | Unauthenticated ADB remote shell bootstrap |
| 9544006 | Outbound HTTP `GET` for `.sh` on ports 808, 8080, 8087, 8888, or 10081 with `Wget/`, `curl/`, or `Go-http-client/` | Kaiji shell stager fetch on primary VPS operator ports |
| 9544007 | Outbound SSH 2.0 handshake on TCP port 24 | Relocated SSH port seen in Kaiji-labelled records |

## How the other rules work

The Sentinel KQL query correlates seven days of `DeviceNetworkEvents` and `DeviceProcessEvents`. It flags connections to Kaiji ports (24, 808, 8087, 10081, 60000, 60001) and high-port UDP fan-out to 25 or more public peers on ports 10000 and above. That network threshold measures high-port UDP fan-out rather than confirmed

DHT payload traffic. The query also flags `/usr/bin/lib/` execution and `/bin/busybox` stagers, with exact-hostname allowlists (`CORP-SSH-BASTION-01`, `CORP-NETOPS-JUMP-01`, `IOT-FIRMWARE-LAB-01`, `SEC-SANDBOX-LNX-01`) for known bastions and labs.

| Sigma rule | Level | Fires when |
|---|---|---|
| Kaiji Linux Botnet Masqueraded Binary Persistence in `/usr/bin/lib` | High | A process runs from `/usr/bin/lib/` or stages there with `chmod` and `/etc/init.d/` (exact lab hostnames excluded) |
| Mirai Mozi and Gafgyt BusyBox Multi-Architecture Stager Execution | High | `/bin/busybox` runs with `wget http`, `tftp -g`, or `chmod 777` and `/bins/`, `.mips`, or `.arm7` |

| YARA rule | Fires when |
|---|---|
| `ODIN_Kaiji_Golang_Linux_SSH_DDoS_Implant` | An ELF under 15 MB contains `Go build`, `/usr/bin/lib/` or `/etc/init.d/`, and `.ssh/known_hosts` or `SYN_ACK` |
| `ODIN_Mozi_P2P_BitTorrent_DHT_Config_Node` | A file under 2 MB combines `d1:ad2:id20:` or `find_node` with `[count]` or `[ss]` and `/bin/busybox` |
| `ODIN_Mirai_Gafgyt_MultiArch_Shell_Dropper` | A script under 512 KB combines `/bin/busybox` and `chmod 777` with two architecture suffixes |

## MITRE ATT&CK mapping

| Tactic | Technique | How it shows up | Detection |
|---|---|---|---|
| Resource Development | T1583.003 Acquire Infrastructure: Virtual Private Server | Kaiji port records on commercial VPS hosts across ports 10081, 808, and 8080 | SID 9544006; KQL |
| Resource Development | T1584.005 Compromise Infrastructure: Botnet | Broadband and regional-telecom routers in Mozi, Hajime, and Mirai | SIDs 9544001 to 9544005; KQL UDP fan-out |
| Credential Access | T1110.001 Brute Force: Password Guessing | Root SSH password guessing and router credential attacks | Key-only SSH policy; SSH authentication logs |
| Lateral Movement | T1021.004 Remote Services: SSH | Kaiji root SSH login, `.ssh/known_hosts` harvesting, and port 24 management | SID 9544007; Sigma rule 1; YARA |
| Command and Control | T1071.001 Application Layer Protocol: Web Protocols | Plain HTTP binary staging on Kaiji operator ports and `/bins/` paths | SIDs 9544003 and 9544006 |
| Command and Control | T1571 Non-Standard Port | Ephemeral DHT ports `>= 10000` and Kaiji ports 10081, 808, 24, 60000, 60001 | SIDs 9544001, 9544006, and 9544007; KQL |
| Impact | T1498 Network Denial of Service | Raw-socket `SYN_ACK`, TCP, and UDP flood routines | Static YARA implant indicator; no shipped rule detects flood traffic |

09

CHAPTER 09

# Indicators

IN SHORT

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

INSIDE THIS CHAPTER

JARM fingerprints to hunt 27

Other indicators (text only) 28

# 09 Indicators

Appendix A holds 7 published indicators: 7 JARM fingerprints for TLS hunting. There are no published IPv4 addresses or file hashes in this report. Ten more protocol, path, port, and certificate markers appear in this report only.

| Type | Count | Where to use it |
|---|---|---|
| JARM fingerprints | 7 | TLS hunting only; never block on a JARM alone |
| Wire, path, port, and certificate markers (text only) | 10 | Suricata, Sentinel KQL, Sigma, and YARA hunting |

## JARM fingerprints to hunt

| JARM | Family | Context |
|---|---|---|
| `29d29d20d29d29d22c29d29d29d29d8e917db1de9d33002d70770504a55ef7` | Kaiji | Shared by 10 commercial-VPS Kaiji candidate endpoints |
| `3fd3fd0003fd3fd00042d42d000000020120996177a65431cde640fa58d2e8` | Mozi, Mirai, Hajime | Shared TLS fingerprint across 52 feed-labelled endpoints (32 Mozi, 14 Mirai, 6 Hajime) |
| `04d02d00004d04d04c04d02d04d04d9674c6b4e623ae36cc2d998e99e2262e` | Mozi, Mirai, Hajime | Shared TLS fingerprint across 44 feed-labelled endpoints (30 Mozi, 7 Mirai, 7 Hajime) |
| `29d14d00029d29d00029d14d29d29df26cd45df722aff084b72395f8d18d1b` | Mirai, Hajime, Mozi | Shared TLS fingerprint across 24 feed-labelled endpoints (18 Mirai, 5 Hajime, 1 Mozi) |
| `21d19d00021d21d21c42d43d00000079c40eb3c1447424b28b311ae50d7d52` | Mirai | Telecom CPE self-signed TLS stack across 21 Mirai-labelled endpoints |
| `04b03b00004b04b04b04b03b04b04b9674c6b4e623ae36cc2d998e99e2262e` | Mirai | Fixed-telco gateway TLS cluster across 19 endpoints in Taiwan |
| `3fd3fd16d3fd3fd22c3fd3fd3fd3fdf20014c17cd0943e6d9e2fb9cd59862b` | Hajime | Leading TLS fingerprint across 12 Hajime-labelled edge peer endpoints |

## Other indicators (text only)

| Indicator | Type | Where to hunt |
|---|---|---|
| `d1:ad2:id20:` | Bencoded BEP 5 KRPC node query | Suricata SID 9544001 on IoT and server VLANs |
| `[count]` | Mozi signed DHT config sync tag | Suricata SID 9544002 and YARA |
| `/bins/` | Multi-architecture ELF HTTP stager URI prefix | Suricata SID 9544003 and Sigma rule 2 |
| `/bin/busybox` | Embedded shell dropper command marker | Suricata SID 9544004, KQL, Sigma rule 2, and YARA |
| `OPEN` + `shell:` on TCP 5555 | Unauthenticated Android Debug Bridge shell bootstrap | Suricata SID 9544005 |
| `/usr/bin/lib/` | Kaiji masqueraded Linux persistence directory | Sentinel KQL, Sigma rule 1, and YARA |
| TCP 10081, 808, 8080 | Primary Kaiji VPS listener ports (124 of 195 endpoints) | Suricata SID 9544006 and Sentinel KQL |
| TCP 24, 60000, 60001 | Historical Kaiji-labelled ports recorded at different times across three host tokens | Suricata SID 9544007 and Sentinel KQL |
| `CN=localhost` | Default self-signed TLS subject (6 Mirai, 4 Kaiji) | TLS hunting context only |
| `CN=CloudFlare Origin Certificate` | Shared CDN origin certificate on 3 collision endpoints | Exclude from standalone certificate blocks |

Withheld: all 2,587 physical host addresses across the study, plus hardware-identifying device certificate strings. The withheld set includes 1,613 dynamic subscriber ISP rows and 68 hosts behind 83 regional-telecom and carrier DHT peer rows. It also includes all 182 Kaiji hosts, other uncorroborated candidates, and collision or open-directory hosts.

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 inventory was reviewed through 20 November 2026, but its upstream records span earlier years: 1,831 of 2,851 were last seen before 6 November, including 189 Kaiji rows and all 87 Gafgyt rows.
- **Feed labels are not attribution.** Family names (Mozi, Mirai, Hajime, Kaiji, Gafgyt) come from upstream feeds. Zero endpoints in this dataset met the multi-signal confirmation bar for published adversary C2 infrastructure.
- **The model is an analytical model.** Zero hosts overlap across the five families. Figures 1 and 3 show how router peer worms and VPS-hosted SSH botnets divide roles, not one coordinated actor.
- **Telecom and subscriber hosts are third parties.** Consumer broadband gateways, regional telecom CPE hosts, and DHT peer records belong to subscribers and carriers, and feed labels alone do not confirm infection, so their addresses and device strings are withheld.
- **No downstream flood visibility.** External records do not show which targets a botnet attacked or capture internal packet logs from target Linux servers.
- **VPS address rotation.** Commercial VPS addresses change hands when leases expire, so uncorroborated single-feed VPS listings should stay in hunting workflows rather than permanent firewall block lists.
- **Rule validation.** All four rule files passed syntax and semantic checks on local Suricata `7.0.12`, `YARA-X`, `pySigma`, and `Kusto.Language` tooling, without live traffic replay. Validate them against local baselines before enabling automated isolation.

11

CHAPTER 11

# Sources

IN SHORT

This chapter lists the public references used in the report.

# 11 Sources


1. MITRE ATT&CK. T1584.005 Compromise Infrastructure: Botnet. https://attack.mitre.org/techniques/T1584/005/ (verified 21 November 2026)
2. BitTorrent.org. BEP 5: DHT Protocol Specification. https://www.bittorrent.org/beps/bep_0005.html (verified 21 November 2026)
3. MITRE ATT&CK. T1583.003 Acquire Infrastructure: Virtual Private Server. https://attack.mitre.org/techniques/T1583/003/ (verified 21 November 2026)
4. MITRE ATT&CK. T1110.001 Brute Force: Password Guessing. https://attack.mitre.org/techniques/T1110/001/ (verified 21 November 2026)
5. MITRE ATT&CK. T1021.004 Remote Services: SSH. https://attack.mitre.org/techniques/T1021/004/ (verified 21 November 2026)
6. IETF. RFC 4252: The Secure Shell (SSH) Authentication Protocol. https://www.rfc-editor.org/rfc/rfc4252 (verified 21 November 2026)
7. MITRE ATT&CK. T1571 Non-Standard Port. https://attack.mitre.org/techniques/T1571/ (verified 21 November 2026)
8. MITRE ATT&CK. T1071.001 Application Layer Protocol: Web Protocols. https://attack.mitre.org/techniques/T1071/001/ (verified 21 November 2026)
9. MITRE ATT&CK. T1498 Network Denial of Service. https://attack.mitre.org/techniques/T1498/ (verified 21 November 2026)
10. Salesforce Engineering. JARM: An Active Transport Layer Security (TLS) Server Fingerprinting Tool. 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 (7) 34

B. Detection rules (4 files) 35

# 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 |
|---|---|---|---|---|---|
| R44-IOC-01 | JARM | 29d29d20d29d29d22c29d29d29d29d8e917db1de9d33002d70770504a55ef7 | 808 | kaiji · shared by ten commercial VPS endpoints | HUNT |
| R44-IOC-02 | JARM | 3fd3fd0003fd3fd00042d42d00000002012099 6177a65431cde640fa58d2e8 | 443 | mozi, mirai and hajime · shared TLS fingerprint across 52 feed-labelled endpoints | HUNT |
| R44-IOC-03 | JARM | 04d02d00004d04d04c04d02d04d04d9674c6b4e623ae36cc2d998e99e2262e | 443 | mozi, mirai and hajime · shared TLS fingerprint across 44 feed-labelled endpoints | HUNT |
| R44-IOC-04 | JARM | 29d14d00029d29d00029d14d29d29df26cd45df722aff084b72395f8d18d1b | 443 | mirai, hajime and mozi · shared TLS fingerprint across 24 feed-labelled endpoints | HUNT |
| R44-IOC-05 | JARM | 21d19d00021d21d21c42d43d00000079c40eb3c1447424b28b311ae50d7d52 | 443 | mirai · telecom CPE self-signed TLS stack across 21 endpoints | HUNT |
| R44-IOC-06 | JARM | 04b03b00004b04b04b04b03b04b04b9674c6b4e623ae36cc2d998e99e2262e | 443 | mirai · fixed-telco gateway TLS cluster across 19 endpoints in Taiwan | HUNT |
| R44-IOC-07 | JARM | 3fd3fd16d3fd3fd22c3fd3fd3fd3fdf20014c17cd0943e6d9e2fb9cd59862b | 443 | hajime · leading TLS fingerprint across 12 edge peer endpoints | HUNT |

# B Detection Rules

Each rule file loads in its own engine (Suricata 7, YARA-X, 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_iot_p2p_dht_mirai_and_kaiji_ssh_botnets.rules

SURICATA

```
# Odin by Brandefense - Monograph 44 / 46 Suricata 7.x Detection Suite
# Scope: Mozi & Hajime BitTorrent DHT peer queries, Mirai/Gafgyt BusyBox & /bins/ stagers,
#        ADB TCP 5555 shell bootstrap, and Kaiji Linux SSH/HTTP operator port behavior.
# Classification: TLP:CLEAR

alert udp $HOME_NET any -> $EXTERNAL_NET 1024:65535 (msg:"ODIN R44 P2P Botnet - Outbound BitTorrent Mainline DHT
KRPC Node Query"; content:"d1:ad2:id20:"; depth:12; fast_pattern; content:"1:q"; distance:20; within:64;
content:"1:y1:qe"; distance:0; classtype:trojan-activity; metadata:affected_product Linux_IoT, attack_target
Networking_Equipment, deployment Perimeter, signature_severity Major, created_at 2026_11_21; sid:9544001; rev:1;)
alert udp any 1024:65535 -> $HOME_NET any (msg:"ODIN R44 Mozi DHT - Inbound Bencoded Peer Config Sync Payload";
content:"d1:rd2:id20:"; depth:12; content:"[count]"; distance:20; fast_pattern; classtype:trojan-activity;
metadata:malware_family Mozi, deployment Perimeter, signature_severity Critical, created_at 2026_11_21; sid:
9544002; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN R44 Mirai/Gafgyt - Outbound Multi-Arch /bins/ ELF Stager
HTTP GET"; flow:to_server,established; http.method; content:"GET"; bsize:3; http.uri; content:"/bins/";
startswith; fast_pattern; pcre:"/\.(?:arm[57]?|mips|mpsl|x86|spc|sh4|ppc|m68k)$/i"; classtype:trojan-activity;
metadata:malware_family Mirai, deployment Perimeter, signature_severity Critical, created_at 2026_11_21; sid:
9544003; rev:1;)
alert http any any -> any any (msg:"ODIN R44 IoT Router Exploit - BusyBox chmod/wget Dropper Command Injection";
flow:to_server,established; http.request_body; content:"/bin/busybox"; fast_pattern; content:"chmod "; distance:
0; classtype:attempted-admin; metadata:malware_family Mirai_Mozi, deployment Perimeter, signature_severity
Critical, created_at 2026_11_21; sid:9544004; rev:1;)
alert tcp any any -> $HOME_NET 5555 (msg:"ODIN R44 Hajime/Mirai - Unauthenticated ADB Port 5555 Remote Shell
Bootstrap"; flow:to_server,established; content:"OPEN"; fast_pattern; content:"shell:"; distance:0; within:64;
classtype:attempted-admin; metadata:malware_family Hajime, deployment Perimeter, signature_severity High,
created_at 2026_11_21; sid:9544005; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET [808,8080,8087,8888,10081] (msg:"ODIN R44 Kaiji Botnet - Outbound Linux
Shell Stager GET on Operator Port"; flow:to_server,established; http.method; content:"GET"; bsize:3; http.uri;
content:".sh"; endswith; fast_pattern; http.user_agent; pcre:"/^(?:Wget|curl|Go-http-client)\//";
classtype:trojan-activity; metadata:malware_family Kaiji, deployment Perimeter, signature_severity High,
created_at 2026_11_21; sid:9544006; rev:1;)
alert ssh $HOME_NET any -> $EXTERNAL_NET 24 (msg:"ODIN R44 Kaiji Hunt - Outbound SSH Handshake on Relocated Port
24"; flow:to_server,established; ssh.proto; content:"2.0"; bsize:3; fast_pattern; classtype:policy-violation;
metadata:malware_family Kaiji, policy behavioral_hunt, signature_severity Medium, created_at 2026_11_21; sid:
9544007; rev:1;)
```

## odin_kaiji_golang_ssh_mirai_and_mozi_dht_artifacts.yar

YARA

```
/*
 * Odin by Brandefense - Monograph 44 / 46 YARA Detection Suite
 * Scope: Kaiji Golang Linux SSH/DDoS implants, Mozi BitTorrent DHT peer payloads,
 *        and Mirai/Gafgyt multi-architecture BusyBox droppers.
 * Classification: TLP:CLEAR
 */

rule ODIN_Kaiji_Golang_Linux_SSH_DDoS_Implant
{
    meta:
        author = "Odin by Brandefense"
        monograph = "44 / 46"
        family = "Kaiji"
        action = "alert_and_isolate"
        tlp = "CLEAR"
    strings:
        $go_id = "Go build" ascii
        $p1 = "/usr/bin/lib/" ascii
```

```
        $p2 = "/etc/init.d/" ascii
        $s1 = ".ssh/known_hosts" ascii
        $s2 = "SYN_ACK" ascii nocase
    condition:
        uint32(0) == 0x464c457f and filesize < 15360KB and $go_id and ($p1 or $p2) and ($s1 or $s2)
}

rule ODIN_Mozi_P2P_BitTorrent_DHT_Config_Node
{
    meta:
        author = "Odin by Brandefense"
        monograph = "44 / 46"
        family = "Mozi"
        action = "alert_and_isolate"
        tlp = "CLEAR"
    strings:
        $dht1 = "d1:ad2:id20:" ascii
        $dht2 = "find_node" ascii
        $cfg1 = "[count]" ascii
        $cfg2 = "[ss]" ascii
        $bb = "/bin/busybox" ascii
    condition:
        filesize < 2048KB and ($dht1 or $dht2) and ($cfg1 or $cfg2) and $bb
}

rule ODIN_Mirai_Gafgyt_MultiArch_Shell_Dropper
{
    meta:
        author = "Odin by Brandefense"
        monograph = "44 / 46"
        family = "Mirai_Gafgyt"
        action = "alert_and_contain"
        tlp = "CLEAR"
    strings:
        $bb = "/bin/busybox" ascii
        $ch = "chmod 777" ascii
        $a1 = ".mips" ascii
        $a2 = ".mpsl" ascii
        $a3 = ".arm7" ascii
        $a4 = ".sh4" ascii
    condition:
        filesize < 512KB and $bb and $ch and 2 of ($a1,$a2,$a3,$a4)
}
```

## sentinel_iot_p2p_dht_and_kaiji_ssh_botnet_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 44 / 46 Microsoft Sentinel & Defender XDR Hunt
// Correlates Kaiji /usr/bin/lib/ masqueraded persistence, relocated SSH port 24 / operator
// port (10081, 808, 8087, 60000, 60001) activity, high-port UDP peer fan-out, and BusyBox IoT stagers.
// Exact device-name allowlists exclude known SSH bastions and firmware lab gateways.

let Lookback = 7d;
let KaijiOperatorPorts = dynamic([24, 808, 8087, 10081, 60000, 60001]);
let ApprovedSshBastions = dynamic(["CORP-SSH-BASTION-01", "CORP-NETOPS-JUMP-01"]);
let ApprovedLabGateways = dynamic(["IOT-FIRMWARE-LAB-01", "SEC-SANDBOX-LNX-01"]);
let NetSignals =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where ActionType in ("ConnectionSuccess", "ConnectionAttempt")
    | where isnotempty(RemoteIP) and RemoteIPType == "Public"
    | extend IsKaijiPortBehavior = (not(DeviceName in~ (ApprovedSshBastions)) and not(DeviceName in~
(ApprovedLabGateways)))
        and RemotePort in (KaijiOperatorPorts)
    | extend IsHighUdpFanout = (not(DeviceName in~ (ApprovedLabGateways)))
        and Protocol =~ "Udp" and RemotePort >= 10000
    | where IsKaijiPortBehavior or IsHighUdpFanout
    | summarize
        FirstNetSeen = min(Timestamp),
```

```
        LastNetSeen = max(Timestamp),
        KaijiPortHits = countif(IsKaijiPortBehavior),
        HighUdpRemotePeers = dcountif(RemoteIP, IsHighUdpFanout),
        SampleRemoteEndpoints = make_set(strcat(RemoteIP, ":", tostring(RemotePort)), 10)
      by DeviceId, DeviceName;
let ProcSignals =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedLabGateways))
    | extend IsKaijiMasquerade = FolderPath startswith "/usr/bin/lib/"
        or (ProcessCommandLine contains "/usr/bin/lib/" and (ProcessCommandLine contains "/etc/init.d/" or
ProcessCommandLine has_any ("cp", "chmod", "netstat", "sshd")))
    | extend IsBusyBoxStager = ProcessCommandLine contains "/bin/busybox"
        and (ProcessCommandLine contains "/bins/" or ProcessCommandLine has_any ("wget", "tftp", "chmod 777"))
    | where IsKaijiMasquerade or IsBusyBoxStager
    | summarize
        FirstProcSeen = min(Timestamp),
        KaijiProcHits = countif(IsKaijiMasquerade),
        BusyBoxProcHits = countif(IsBusyBoxStager),
        ObservedCommands = make_set(ProcessCommandLine, 5)
      by DeviceId, DeviceName;
NetSignals
| join kind=fullouter ProcSignals on DeviceId
| extend HostName = coalesce(DeviceName, DeviceName1)
| extend KaijiPortHits = coalesce(KaijiPortHits, 0),
         HighUdpRemotePeers = coalesce(HighUdpRemotePeers, 0),
         KaijiProcHits = coalesce(KaijiProcHits, 0),
         BusyBoxProcHits = coalesce(BusyBoxProcHits, 0)
| where KaijiProcHits > 0 or BusyBoxProcHits > 0 or KaijiPortHits >= 3 or HighUdpRemotePeers >= 25
| extend RecommendedAction = case(
    KaijiProcHits > 0 and KaijiPortHits > 0, "ISOLATE_KAIJI_SSH_PERSISTENCE_AND_PORT_CLUSTER",
    KaijiProcHits > 0 or BusyBoxProcHits > 0, "CONTAIN_LINUX_MASQUERADE_OR_BUSYBOX_STAGER",
    HighUdpRemotePeers >= 25, "HUNT_HIGH_PORT_UDP_FANOUT_ON_VLAN",
    "HUNT_RELOCATED_SSH_OR_KAIJI_OPERATOR_PORTS"
  )
| project
    HostName,
    RecommendedAction,
    KaijiProcHits,
    BusyBoxProcHits,
    KaijiPortHits,
    HighUdpRemotePeers,
    FirstNetSeen,
    LastNetSeen,
    SampleRemoteEndpoints,
    ObservedCommands
| order by KaijiProcHits desc, HighUdpRemotePeers desc
```

## sigma_linux_kaiji_ssh_persistence_and_iot_stager.yml

SIGMA

```
title: Kaiji Linux Botnet Masqueraded Binary Persistence in /usr/bin/lib
id: 7c44a102-8d2e-4f19-9b31-440000000002
status: experimental
description: Detects Linux process creation from or staging into /usr/bin/lib/ paired with system-utility
masquerading or /etc/init.d/ persistence characteristic of Golang Kaiji implants.
author: Odin by Brandefense
date: 2026-11-21
modified: 2026-11-21
references:
    - https://attack.mitre.org/techniques/T1110/001/
    - https://attack.mitre.org/techniques/T1021/004/
logsource:
    category: process_creation
    product: linux
detection:
    selection_exec_dir:
        Image|startswith: '/usr/bin/lib/'
    selection_stage_path:
```

```
        CommandLine|contains: '/usr/bin/lib/'
    selection_stage_chmod:
        CommandLine|contains: 'chmod '
    selection_stage_init:
        CommandLine|contains: '/etc/init.d/'
    filter_approved_sandbox:
        ComputerName:
            - 'SEC-SANDBOX-LNX-01'
            - 'IOT-FIRMWARE-LAB-01'
    condition: (selection_exec_dir or (selection_stage_path and selection_stage_chmod and selection_stage_init))
and not filter_approved_sandbox
falsepositives:
    - Dedicated malware detonation sandboxes listed by exact hostname
level: high
tags:
    - attack.persistence
    - attack.defense-evasion
    - attack.lateral-movement
    - attack.t1021.004
---
title: Mirai Mozi and Gafgyt BusyBox Multi-Architecture Stager Execution
id: 7c44a103-8d2e-4f19-9b31-440000000003
status: experimental
description: Detects BusyBox shell dropper command lines retrieving and marking executable multi-architecture IoT
botnet binaries (.mips, .mpsl, .arm7, .sh4, or /bins/ paths).
author: Odin by Brandefense
date: 2026-11-21
modified: 2026-11-21
references:
    - https://attack.mitre.org/techniques/T1584/005/
    - https://attack.mitre.org/techniques/T1498/
logsource:
    category: process_creation
    product: linux
detection:
    selection_busybox:
        CommandLine|contains: '/bin/busybox'
    selection_fetch:
        CommandLine|contains:
            - 'wget http'
            - 'tftp -g'
            - 'chmod 777'
    selection_arch:
        CommandLine|contains:
            - '/bins/'
            - '.mips'
            - '.mpsl'
            - '.arm7'
            - '.sh4'
    filter_approved_lab:
        ComputerName:
            - 'IOT-FIRMWARE-LAB-01'
            - 'SEC-SANDBOX-LNX-01'
    condition: selection_busybox and selection_fetch and selection_arch and not filter_approved_lab
falsepositives:
    - Embedded Linux firmware build or QA test benches listed by exact hostname
level: high
tags:
    - attack.execution
    - attack.resource-development
    - attack.t1584.005
    - attack.impact
    - attack.t1498
```

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/iot-botnet-infrastructure/
Short Read · brandefense.io/odin-research/blog/iot-botnet-feeds/
Defense Kit · brandefense.io/odin-research/defense-kit/