

Destruction in Plain Sight:

How Modern DDoS Botnets Weaponize Publicly Available
Cryptocurrency Blockchains for Immutable C2
Infrastructure

·||· Recorded Future®



·||· The Most Common Malware Seen

Top 10 families by unique samples. Eight of ten are remote-access trojans.

UNIQUE SAMPLES IDENTIFIED (H1 2026)



Recorded Future Malware Intelligence · H1 2026



Remote-access trojans

Attacker controls the PC as if sitting at it. Eight of the ten families here.



Cobalt Strike

Pirated pen-test tool. Its presence means a live intruder, not just malware.



Stealc

Stealware. Grabs saved passwords and crypto wallets in seconds.



Five Malware Types and What Each Does

Two families defined each category in H1 2026.



Remote-access trojans

Attacker controls the PC as if sitting at it.

AsyncRAT

XWorm

In via phishing & fake packages



Stealware

Steals saved passwords & crypto wallets in seconds.

Vidar

MacSync

In via fake downloads & ads



Loaders

Smuggles the real payload past defences.

HijackLoader

DonutLoader

In via phishing pages & files



Backdoors

Quiet, long-term access for espionage.

Cobalt Strike

BRICKSTORM

In via unpatched servers & MSP keys



Ransomware

Encrypts data, deletes backups, extorts.

Interlock

The Gentlemen

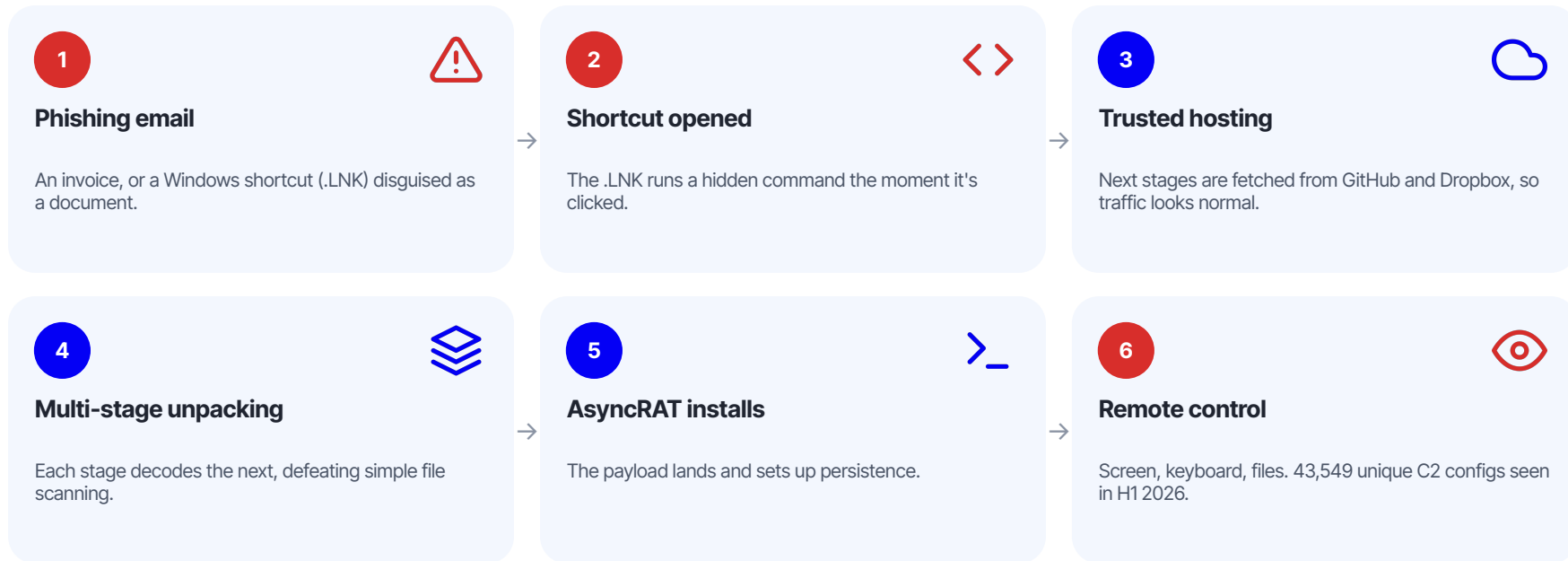
In via firewall zero-day & AnyDesk



·||· Step by Step: Phishing → AsyncRAT

Case study · Insikt Group reporting

The #1 malware family of H1 2026 still starts with an email.



Seen in H1 2026 Kimsuky's multi-stage .LNK chain hosted on GitHub & Dropbox · invoice-themed phishing against German manufacturers exploiting CVE-2024-4345 · 59,507 unique samples, the #1 family

Stops it Block .LNK and script attachments at the mail gateway · restrict script hosts · alert on outbound traffic to code- and file-hosting your organisation doesn't use



·||· Seven Places Attackers Hit Networks

Critical domain: outside-in

Six come from the H1 2026 report. The seventh is what we found ourselves.



Firewalls & VPN

60 flaws give full control, no login



IT provider access

One stolen key rooted firewalls & storage



Staff inboxes

Phishing & fake pop-ups start many attacks



Backups

Ransomware deletes them before encrypting



Developer tools

Poisoned packages steal cloud secrets



Customer phones

Tap-to-pay card fraud up 188% (Jan-Apr 2026)



Routers & IoT

Turned into DDoS botnets against others (Section 04)

None of these needs a new product to defend.

Only attention to the basics.

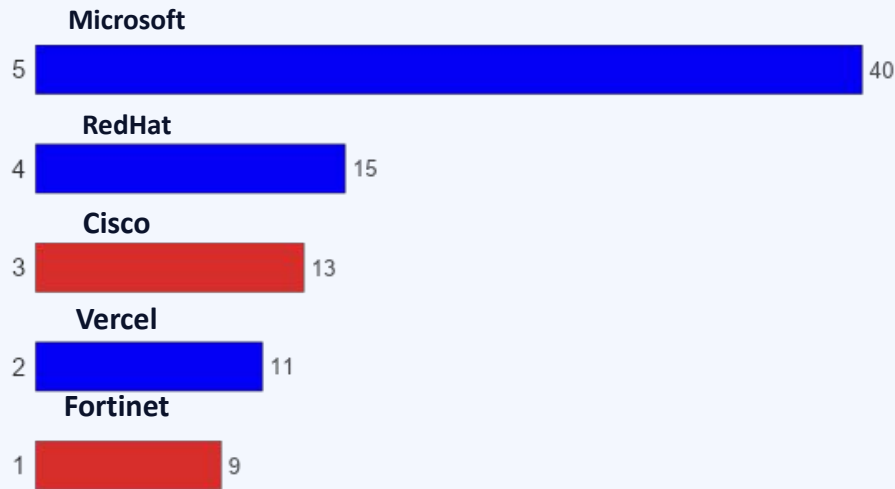


Which Systems Attackers Went After

Source · Recorded Future, Figure 2

Many of these sit on your network edge right now.

EXPLOITED FLAWS BY VENDOR (TOP 5 OF 98)



● Network-edge vendors. Plus 93 other vendors with at least one exploited flaw; 67 vendors had exactly one.

60

Flaws give full control from the internet with no password. Fix these first.

31 days

Median gap from disclosure to attack. 38% were hit within a week.

17

Exploited flaws dated 2020 or older. Old bugs still work.



·||· Patch Priority: What to Fix First

Work the list from the top.

Two deadlines: disclosure day and the backlog

1

Edge gear with no-login remote takeover

Cisco, Fortinet, Ivanti, Exchange. Patch in days.

60 flaws

2

Anything with a public exploit out

Assume it's in use. Can't patch? Restrict & segment.

66 flaws

3

The long tail and the old bugs

17 exploited flaws date from 2020 or earlier. 67 vendors had a single flaw each.

98 vendors

THE CLOCK

38%

exploited within 7 days of disclosure

31 days

median time to attack

11%

hit before the flaw was public

Can't patch yet? Block macros · phishing-resistant MFA · restrict remote tools · WAF/IPS rules and segmentation as interim controls



Dysphoria / Drunkasfk

DDoS Botnet

What We Observed

Our research · 63 samples

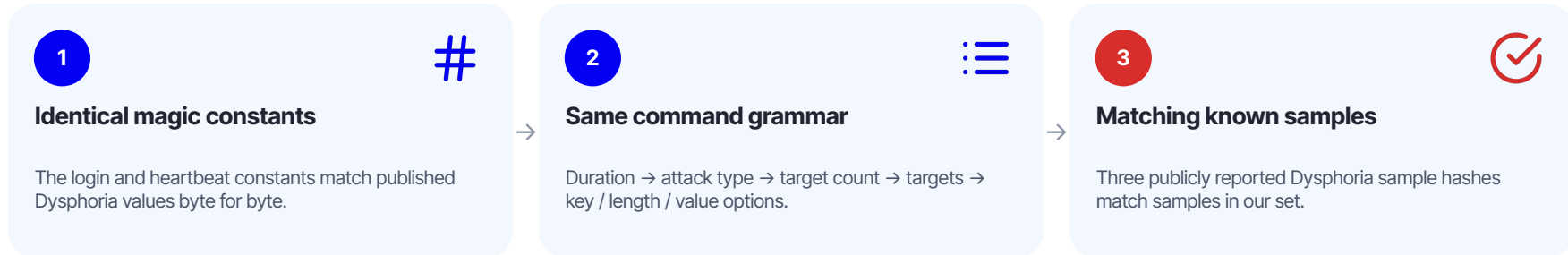
Every sample carried the same plaintext lock string.

Every sample carried the same plaintext lock string:

`drunkasfk`

← our internal working name: *DrunkAsFk*

HOW WE KNEW IT WAS DYSPHORIA



BUILT FOR FIVE PROCESSOR FAMILIES

ARM32

ARM64

MIPS (big-endian)

MIPS (little-endian)

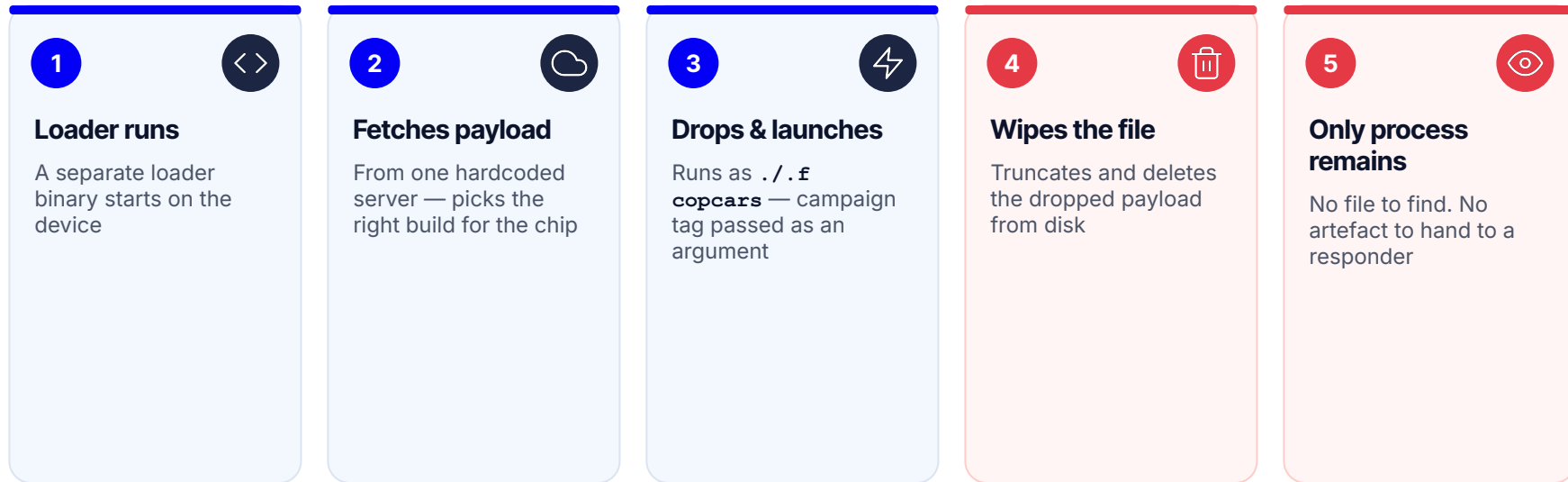
x86

Runs on almost any embedded device: routers, cameras, NAS.



How It Lands — and Vanishes

EXECUTION PIPELINE & FILELESS PERSISTENCE MECHANISM



We confirmed it still live on 16 Sep 2026 — pulled `http /arm5` directly and matched its hash to our reference sample.

Server: 162.249.125.145 **Paths:** /arm /arm5 /arm7 /mips /mipsel **Loader:** 5ba55001...99a15e

What It Does Once Inside



Once inside

The payload is running on the device

Deletes its files

Wipes its own install files so forensics find nothing

Kills rival bots

Walks the process list and terminates competing malware

Blocks admin tools

Kills wget, curl, tftp, telnet, sh, ps, reboot

Locks itself in

Edits its memory priority so the OS never kills it

Result: an invisible, unkillable resident. The usual clean-up commands don't work — because it killed them.

How It Finds Its Command Server



Fallback: if a node fails, it resolves via DNS-over-HTTPS (Google, Cloudflare). The address lives on a public ledger — no takedown can reach it.

What It Can Actually Do

19 attack slots. 16 work. 3 are dead leftovers from a shared code template.



UDP floods

Random, custom or fixed-prefix payloads, CIDR-aware



Raw TCP floods

SYN, ACK, FIN/PSH with spoofed headers & real checksums



Connection floods

Up to 2,000 concurrent connects per target



Slow exhaustion

Connections held open with minimal keepalive pokes



Browser mimic

128 connections posing as real desktop & mobile browsers

badge = number of attack types in the family

Standouts Type 7 rotates 14 real User-Agents and randomises headers & cookies to pass as organic traffic · Type 17 does a real handshake to harvest the target's own TCP options, then spoofs SYNs that reuse them — defeating anti-spoof checks · Type 19 randomises socket options for a realistic client fingerprint.

Dead slots types 6, 10 and 16 are broken pointers into shared helper code — template residue, not capabilities.

Dysphoria Scale and Infrastructure

Public reporting · broader family

Publicly reported figures for the wider Dysphoria family; our 63-sample cluster is one branch of it.

740,000

Peak daily C2 access requests

Recorded during a monitored weekly window, demonstrating extremely high-frequency command-and-control activity.

239,000

Peak active bot devices recorded outside mainland China through active monitoring.

1,801

Active monitored devices recorded within mainland China during the same observation window.

~200k

Estimated total infected devices, based on leaked criminal infrastructure claims.

22

Verified C2 servers across the family. Our cluster resolves one at a time via a single ENS record.

Source: XLab public reporting on the Dysphoria family. Figures describe the family-wide botnet, not only the cluster analysed in this deck. Verify against the cited report before external distribution.



Verdict: one operator's branch of a family still under active development.



Where it's Dysphoria

- Login & heartbeat magic constants are byte-for-byte identical to XLab's published values
- Command grammar matches structurally: duration, type, targets, then key + length + value options
- Three of XLab's own published sample hashes match samples in our corpus
- All confirmed live against the running protocol, not just by reading strings



Where this build differs

- Different ENS name and TEXT key from the ones XLab documented
- Simpler C2-decode transform
- 80-byte frame instead of 78
- No relay, proxy or UPnP code anywhere in 63 samples. Treat this cluster as DDoS-only

Detection lesson One matching sample arrived UPX-packed and only showed the fingerprint after unpacking. A string-only rule would have missed it; our rules now require unpacking first.

·||· Following the Money On-Chain

Our research · on-chain tracking

One wallet. Eight blockchains. Every move on the public record.

THE CONTROLLING WALLET

0x435Cf9676ea8a906A51E97012dF2A947435F2d7f

Registered the botnet's ENS name and still controls it. Same public resolver (0xF291...) found in the malware's code.

TRANSACTIONS WE'RE TRACKING

356

Ethereum

writes to the ENS resolver, each one a C2 update. Latest: Sep 20, 2026, under TEXT key "woah"

223

Solana

transactions from the same operator on the second-busiest chain

ACTIVE ON 8 BLOCKCHAINS

Ethereum

Solana

BNB Chain

Optimism

Polygon

Avalanche

Base

Gnosis

Nearly all volume sits on Ethereum and Solana.

ENS renamed Mar 26, 2026: [2busydrinkingcodeine.eth](#) → [ukrainianhorseriding.eth](#)



·||· See It for Yourself, Live

Live demo · if time allows

The operator's wallet is public. Open it in a block explorer and watch the C2 updates in real time.



Ethereum · Etherscan

etherscan.io/address/0x435Cf9676ea8a906A51E97012dF2A947435F2d7f



Solana · Solscan

solscan.io/tx/5kX6Y1ogEE97Mvmmx2e8idMP6GAJQYicMxQVAgZ6hQbbKChFbnxi6kEgLrk haz xpyFopiPyK9teCUKXCDXbR7vZc

WHAT TO LOOK FOR



The transaction list

Hundreds of writes to the ENS resolver. Each one is a C2 update.



The "woah" TEXT record

Under the resolver's event logs: the obfuscated address string.



The cross-chain activity

The same operator's footprint across 8 blockchains.

Both links are repeated on the closing slide.



••• The Operator Left a Calling Card

Our research · on-chain OSINT

The same blockchain record that hides the C2 also leaked the operator's Telegram handle.



On the same ENS resolver the botnet reads for its server address, the operator stored a second record: a contact handle. A public ledger cuts both ways.

ON-CHAIN EVENT LOG (ENS PUBLIC RESOLVER 0xF291...)

Address 0xf29100983e058b709f3d539b0c765937b804ac15 (ENS: Public Resolver 3)

Name TextChanged (index_topic_1 bytes32 node, index_topic_2 string indexedKey, string ke

Topics

- 0 0x448bc014f1536726cf8d54ff3d6481ed3cbc683c2591ca204274009afa09b1a1
- 1: node Dec → 9AD95A97E82232BD5068D5403048CCA1472A8FE023CBE648600ACE0870589C9C
- 2 → 0xc798f577086e38b080cfcf1b2551913f4c793c9b0869e7fbc3e476a4ef647518

Data

key (string): telegram main
value (string): @dysphoriaprivate



Leaked handle

@dysphoriaprivate

TEXT key "telegram main", on the same resolver the botnet uses for C2.

Note: a self-reported handle is an investigative lead, not proof of identity. It can be faked or reused.

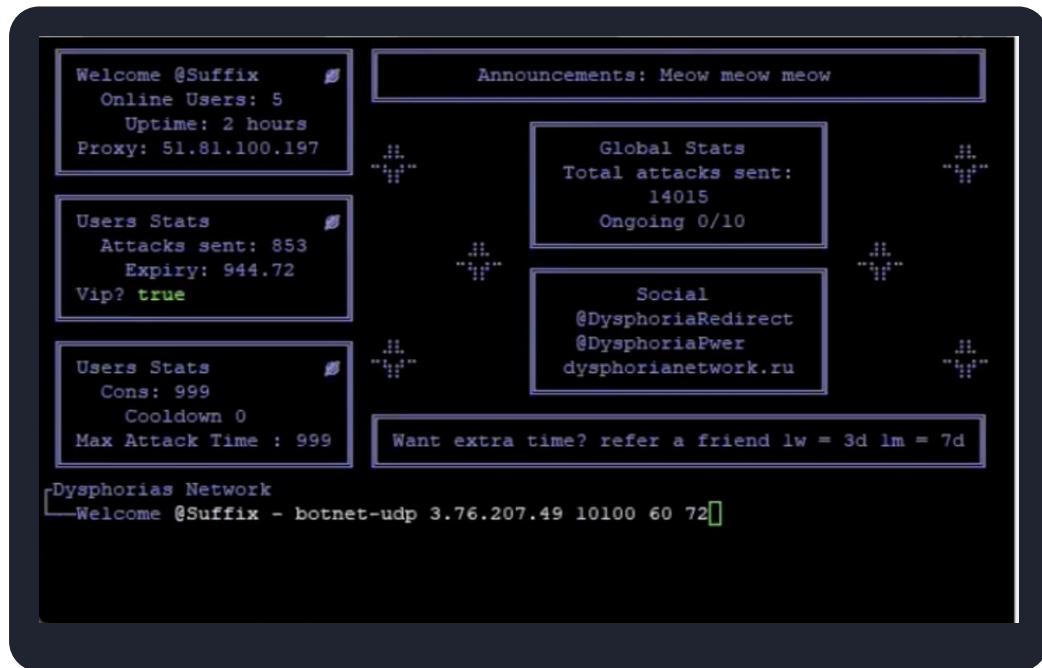


Infrastructure and Channels

Our research · social footprint

Attribution and social footprint of the operator group.

C2 PANEL INTERFACE



KNOWN COMMUNICATION CHANNELS



Discord

[discord\[.\]com/invite/x2xTc9EhcC](https://discord[.]com/invite/x2xTc9EhcC)



YouTube

[youtube\[.\]com/DysphoriaServices](https://youtube[.]com/DysphoriaServices)

Demonstrated DDoS-for-hire attacks against online games including Rainbow Six Siege and Apex Legends.



Telegram

[t\[.\]me/dysphoriacar](https://t[.]me/dysphoriacar)

Main channel.



Defending Against IoT Botnets

For network operators

Harden the edge, disable what you don't use, and watch for traffic that shouldn't be there.



Harden edge & routers

Disable UPnP/PnP to stop port-mapping abuse. Enforce strong admin credentials on every Telnet and SSH interface, or disable them outright.



Watch the traffic

Flag outbound TLS 1.3 on 443 with no SNI, and connections going straight to a raw external IP. Baseline first: some CPE firmware does this legitimately, so tune before alerting.



Watch for blockchain calls

Track outbound Ethereum and Solana RPC traffic from IoT and CPE subnets, e.g. cloudflare-eth.com, drpc.org, and public Solana mainnet RPC endpoints.



Detect on the device

Deploy YARA rules matching the drunksfk lock string and registration magic. Unpack UPX first or you will miss samples.



·||· Three Things to Take Back

Summary

Attackers are hiding in plain sight: in your trusted tools, logins, and update channels.

1

Watch behaviour, not just signatures.

Attackers hide in trusted tools, logins and update channels. Correlate sequences of activity rather than alerting on single events.

2

Patch the edge first, and disable UPnP/PnP on edge-facing devices.

Sixty flaws give full control with no login, 38% of exploited CVEs were hit within a week of disclosure, and the median attacker moves in 31 days.

3

Your CPE is somebody's weapon.

Blockchain-based botnets are visible at ISP scale if you're looking: no-SNI TLS to raw IPs, and Ethereum/Solana RPC calls from subscriber subnets.

LIVE WALLET

```
etherscan.io/address/  
0x433Cf9676ea8a906A51E97012dF2A947435  
F2d7f
```

SOURCES



YARA Rule: Dysphoria/Drunkasfk Botnet Hunt

Detection · hunt rule

Hunts unpacked Linux ELF payloads that resolve C2 via Ethereum JSON-RPC. Run UPX unpacking first.

●●● HUNT_MAL_Dysphoria_Botnet.yar

```
rule HUNT_MAL_Dysphoria_Botnet {
  meta:
    description = "Hunts Linux ELF bot payloads resolving C2 via Ethereum JSON-RPC"
  strings:
    $ens_rpc_template    = "{\"jsonrpc\":\"2.0\", \"id\":1, \"method\":\"eth_call\"}"  ascii
    $ens_public_resolver = "0xf29100983e058b709f3d539b0c765937b804ac15"  nocase
    $eth_call            = "\"method\":\"eth_call\""  ascii
    $rpc_latest          = "}, \"latest\"}"  ascii
    $rpc_result          = /"result": "0x[0-9a-fA-F]{2,}"/  ascii
  condition:
    uint32(0) == 0x464c457f and filesize < 20MB and (
      all of ($ens_rpc_template, $ens_public_resolver) or
      2 of ($eth_call, $rpc_latest, $rpc_result)
    )
}
```

Fix from the draft: YARA text strings do not support * wildcards, so the earlier \$eth_call / \$rpc_latest / \$rpc_result strings would only have matched a literal asterisk. Wildcard matching now uses a regex string.



Questions?

CONTACT

Chase Clowser

chase.clowser@recordedfuture.com