

Threat Alert: Mirai/Gafgyt Fork with New DDoS Modules Discovered

cujo.com/mirai-gafgyt-with-new-ddos-modules-discovered/

September 7, 2021



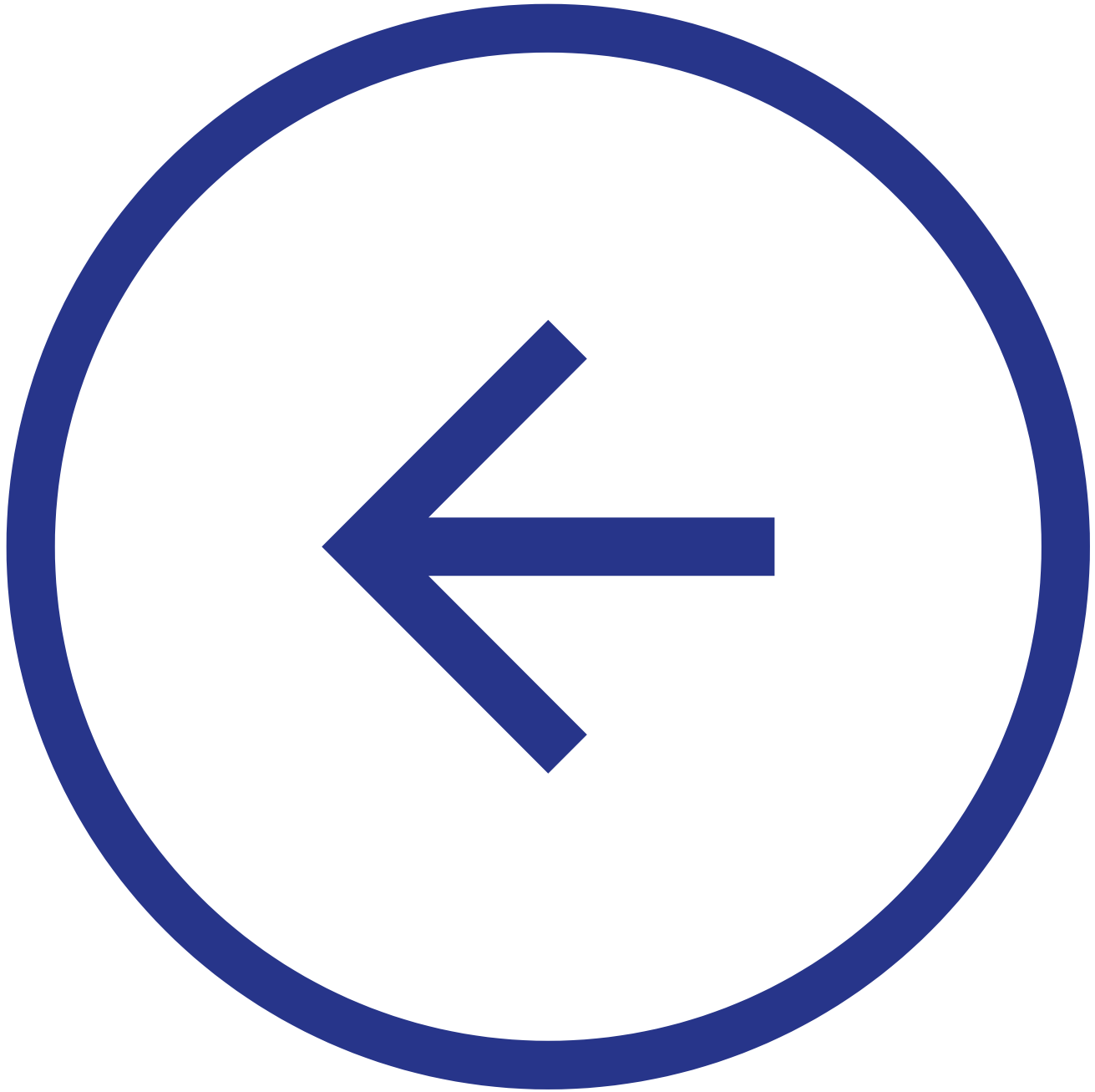
Threat Alert:
**Mirai/Gafgyt Fork with
New DDoS Modules
Discovered**

READ NOW



Albert Zsigovits
Threat Researcher

 **CUJOAI**



All posts

September 7, 2021

On the 27th of August, we have found evidence that an IoT device in one of our customer environments had accessed a malicious software sample. We have investigated the sample and discovered that a Gafgyt fork has been updated and it is now being distributed with **two new Distributed Denial of Service (DDoS) modules** to launch attacks against targeted machines.

Mirai and Gafgyt have been the go-to IoT malware for many years now in cybercrime circles: their versions have successfully infected millions of vulnerable IoT devices over the years. Since their source code have been released publicly, many threat actors use the Mirai or

Gafgyt code as a malware-skeleton and then retrofit it with their unique improvements, creating their own special version of the botnet.

In this short threat alert, we will detail the most important findings related to this new malicious campaign.

Overview of the New Mirai/Gafgyt Fork

Two interesting entries in our logs started the investigation. Previously, we have not observed the name “Korpze” as a campaign tag. The set of numbers at the end of the filename suggests a random keyboard typing with a reference to the l33t internet-slang with “1337”.

```
Date: 27-08-2021 16:31 UTC
URL: http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]sparc

Date: 27-08-2021 15:47 UTC
URL: http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]mpsl
```

Technical Details

All of the investigated malware samples had retained their debug information and symbols, their binaries had not been stripped, which is, as we have observed, standard with these campaigns, as malware operators do not pay much attention to operational security.

```
Korpze1233121337.arm: ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, with debug_info, not stripped
Korpze1233121337.arm4: ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, with debug_info, not stripped
Korpze1233121337.arm5: ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, with debug_info, not stripped
Korpze1233121337.arm6: ELF 32-bit LSB executable, ARM, EABI4 version 1 (SYSV), statically linked, with debug_info, not stripped
Korpze1233121337.i586: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, not stripped
Korpze1233121337.i686: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, not stripped
Korpze1233121337.m68k: ELF 32-bit MSB executable, Motorola m68k, 68020, version 1 (SYSV), statically linked, not stripped
Korpze1233121337.mips: ELF 32-bit MSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, not stripped
Korpze1233121337.mpsl: ELF 32-bit LSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, not stripped
Korpze1233121337.ppc: ELF 32-bit MSB executable, PowerPC or cisco 4500, version 1 (SYSV), statically linked, not stripped
Korpze1233121337.sh4: ELF 32-bit LSB executable, Renesas SH, version 1 (SYSV), statically linked, not stripped
Korpze1233121337.sparc: ELF 32-bit MSB executable, SPARC, version 1 (SYSV), statically linked, with debug_info, not stripped
```

Lack of stripping usually observed in immature campaigns

Origins of the New Botnet

There are two references in the binary to *YakuzaBotnet* and *Scarface*, the developer of a Mirai variant:

```
YakuzaBotnet
```

```
Scarface1337Self Rep Fucking NeTiS and Thisity On Ur FuCkInG FoReHeAd We BiG
L33T HaxErS
```

It suggests that the base of this variant was most likely taken from Yakuza botnet, a Mirai variant leaked to the public:

How the Botnet Gains an Initial Foothold

The function `telnet_scanner_init` is in charge of setting the initial foothold in vulnerable devices. It scans randomly generated IPs and tries to log in with a list of pre-defined, hardcoded credentials on port 23 (Telnet).

These leaked credentials are the default credentials of many poorly secured IoT devices. **Users are strongly advised to change these passwords once they purchase the following appliances:**

Username	Password	Related appliance
admin	admin	–
admin	smcadmin	SMC routers
default	default	–
ftp	ftp	–
guest	12345	–
guest	guest	–
mg3500	merlin	Camtron IP cameras
root	calvin	Dell DRAC/iLO
root	cat1029	HiSilicon IP cameras/DVRs/NVRs
root	gm8182	Grain Media DVR
root	hi3518	HiSilicon IP cameras/DVRs/NVRs
root	icatch99	Lilin DVR
root	pon521	GPON module DFP-34G-2C2
root	root	–
root	root621	SNR-ONU-EPON-1G
root	xc3511	VTA-83170 DVR
root	xmhdipc	HiSilicon IP cameras/DVRs/NVRs
root	vizxv	Dahua IP cameras

Important Functions in this Mirai/Gafgyt Fork

The following table lists all attack modules that were present in the investigated sample. Besides the Telnet dictionary attack module, it uses many different DoS modules. Most of these have already been investigated by other researchers, but the last two modules are quite new:

Function entry	Function name	Description
080490fe	sendCNC	CNC Botnet flood, resource starvation attack
0804b096	sendDOMINATE	DoS attack with random gibberish data
0804b804	sendJUNK	Send junk data as DoS attack
0804b488	sendHTTP	HTTP DoS server resource exhaustion attack
0804b5c3	sendHTTPCloudflare	Attacking a site protected by Cloudflare
080491a0	sendSTD	DoS attack with random strings
08049861	sendSTDHEX	DoS attack with random hexadecimal bytes
08049e39	sendTCP	TCP DoS attack with random TCP packet parameters
0804939d	vseattack1	DoS attack against servers running Valve's Source Engine
08049310	makevsepacket1	DoS attack against servers running Valve's Source Engine
080508d4	telnet_scanner_init	Telnet scanner attacks random IPs with hardcoded creds
0804fe00	add_auth	Wrapper for adding credentials to the auth function
0804fedd	init_auth	Initializing hardcoded credentials
0804b6fb	UDPBYPASS	UDP DoS flood with hardcoded hex bytes
0804a7bf	UDPRAW	UDP DoS flood with raw copied bytes

0804aa43	ovhl7	HTTP DDoS attack on OVH servers with a specific payload
0804c384	attacks_vector_openvpn_swak	New
0804bdd0	attacks_vector_wabba_jack	New

Attacks_vector modules

This is the first time these modules have been observed in Gafgyt variants. The name of the first one suggests a module to DoS OpenVPN servers. The name choice for the second one is curious, as it is the name of a famous modding tool for PC games like Skyrim.

However, the two functions work similarly by building the UDP header via **build_udp_header** and then connecting to the target via **socket_connect_raw_udp**, and launching the UDP flood.

```

0x0804c38f  c745e800000000  mov dword [var_18h], 0
0x0804c396  c745ec00000000  mov dword [fildes], 0
0x0804c39d  6a2d            push 0x2d ; '-' ; 45
0x0804c39f  8d45e4         lea eax, [var_1ch]
0x0804c3a2  50             push eax
0x0804c3a3  ff750c         push dword [arg_ch]
0x0804c3a6  ff7508         push dword [arg_8h]
0x0804c3a9  e8bdfdf7ff    call build_udp_header ; sym.build_udp_header
0x0804c3ae  83c410         add esp, 0x10
0x0804c3b1  8945f0         mov dword [ptr], eax
0x0804c3b4  8b45e4         mov eax, dword [var_1ch]
0x0804c3b7  83c02d         add eax, 0x2d ; 45
0x0804c3ba  8945e4         mov dword [var_1ch], eax
0x0804c3bd  c745ec00000000  mov dword [fildes], 0
0x0804c3c4  eb1f            jmp 0x804c3e5

```

UDP Flood initialized in `openvpn_swak` function

The `wabbajack` function uses **socket_connect_icmp** to launch an ICMP flood at the target.

```

[0x0804beb8]
0x0804beb8  8b5df0         mov ebx, dword [fildes]
0x0804bebb  83ec08         sub esp, 8
0x0804bebe  ff750c         push dword [arg_ch]
0x0804bec1  ff7508         push dword [arg_8h]
0x0804bec4  e86efd77ff    call socket_connect_icmp ; sym.socket_connect_icmp
0x0804bec9  83c410         add esp, 0x10
0x0804becc  89c2           mov edx, eax
0x0804bece  89d8           mov eax, ebx

```

ICMP Flood initiated in `wabba_jack` function

Similar Naming to PBot Modules

Recently, [CN-CERT](#) has released an article on a new, emerging P2P botnet called PBot. PBot consists of 6 interesting DDoS modules that have similar goals to the two DDoS modules we have observed in the Korpze campaign.

- attacks_vector_game_killer
- attacks_vector_nfo_v6
- attacks_vector_plainudp
- attacks_vector_plaintcp
- attacks_vector_l7_ghp
- attacks_vector_ovh_l7

Interestingly, neither PBot, nor the Korpze variant uses each other's DDoS modules, but their naming convention is the same. Most likely these DDoS modules are now disseminated in cybercrime forums, and it is up to the malware developers, which ones they include in their own campaigns.

The less likely assumption is that PBot and this Korpze campaign are related as they share DDoS modules from the same attack corpus, but we cannot really attribute based on a poor string match.

Updating nameservers

There is a specific function called **UpdateNameSrvs** to change nameservers on the infected device. The function is responsible for writing the file `/etc/resolv.conf` with Google's DNS servers.

```
void UpdateNameSrvs() {
    uint16_t fhandler = open("/etc/resolv.conf", O_WRONLY | O_TRUNC);
    if (access("/etc/resolv.conf", F_OK) != -1) {
        const char* resd = "nameserver 8.8.8.8\nnameserver 8.8.4.4\n";
        size_t resl = strlen(resd);
        write(fhandler, resd, resl);
    } else { return; }
    close(fhandler);
}
```

This is most likely to aid malware operators: the developer likely wanted to circumvent any DNS servers that block malicious IPs from reaching users, as Google's 8.8.8.8 DNS does not block malicious IPs:

“Google Public DNS rarely performs blocking or filtering, though it may if we believe this is necessary to protect our users from security threats.”

<https://developers.google.com/speed/public-dns/docs/intro>

User-Agents Used in HTTP DoS Attacks

There are 60 hardcoded User-Agents included in the sample, which are used in the DoS module **ovh17**, **SendHTTP**, and **SendHTTPCloudflare**. Once the DoS module is launched at a target, the function randomly chooses a User-Agents to attack with.

```
[0x0804b4c2]
0x0804b4c2    e8c1aa0000    call rand ; sym.rand ; int rand(void)
0x0804b4c7    89c2         mov edx, eax
0x0804b4c9    b83b000000   mov eax, 0x3b ; ';' ; 59
0x0804b4ce    89c1         mov ecx, eax
0x0804b4d0    89d0         mov eax, edx
0x0804b4d2    c1fa1f      sar edx, 0x1f
0x0804b4d5    f7f9        idiv ecx
0x0804b4d7    89d0         mov eax, edx
0x0804b4d9    8b048560f00508 mov eax, dword [eax*4 + obj.useragents] ; 0x805f060
0x0804b4e0    83ec08      sub esp, 8
0x0804b4e3    50         push eax
0x0804b4e4    ff750c      push dword [arg_ch]
0x0804b4e7    ff7514      push dword [arg_14h]
0x0804b4ea    ff7508      push dword [arg_8h]
0x0804b4ed    68bcc30508  push str.s__s_HTTP_1.1____Host:__s____User_Agent:__s
0x0804b4f2    8d85ecfdfff lea eax, [ptr]
0x0804b4f8    50         push eax ; char *s
0x0804b4f9    e82e770000  call sprintf ; sym.sprintf ; int sprintf(char *s, con
0x0804b4fe    83c420      add esp, 0x20
0x0804b501    e82e720000  call fork ; sym.fork
0x0804b506    85c0        test eax, eax
0x0804b508    0f84a1000000 je 0x804b5af
```

SendHTTPCloudflare DoS module uses randomized User-Agents

This mechanism is in place for evading security countermeasures: victims cannot simply block the attack by denying a single specific User-Agent.
















Here's a small excerpt of User-Agents used:

- FAST-WebCrawler/3.6 (atw-crawler at fast dot no; http://fast.no/support/crawler.asp)
- TheSuBot/0.2 (www.thesubot.de)
- Opera/9.80 (X11; Linux i686; Ubuntu/14.10) Presto/2.12.388 Version/12.16
- BillyBobBot/1.0 (+http://www.billybobbot.com/crawler/)
- FAST-WebCrawler/3.7 (atw-crawler at fast dot no; http://fast.no/support/crawler.asp)
- zspider/0.9-dev http://feedback.redkolibri.com/
- ...

New Campaigns Appearing

Just as we were investigating the Command-and-Control server, we have observed **a new campaign being switched on** and all malicious binaries being exchanged with a new set of binaries for various architectures.

Index of /bins

Name	Last modified	Size	Description
 Parent Directory		-	
 Korpze1233121337.arm	2021-08-29 08:42	158K	
 Korpze1233121337.arm4	2021-08-29 08:42	158K	
 Korpze1233121337.arm5	2021-08-29 08:42	152K	
 Korpze1233121337.arm6	2021-08-29 08:42	171K	
 Korpze1233121337.i586	2021-08-29 08:42	120K	
 Korpze1233121337.i686	2021-08-29 08:42	121K	
 Korpze1233121337.m68k	2021-08-29 08:42	139K	
 Korpze1233121337.mips	2021-08-29 08:42	196K	
 Korpze1233121337.mpsl	2021-08-29 08:42	196K	
 Korpze1233121337.ppc	2021-08-29 08:42	144K	
 Korpze1233121337.sh4	2021-08-29 08:42	138K	
 Korpze1233121337.sparc	2021-08-29 08:42	159K	
 Korpze1233121337.x86	2021-08-29 08:42	145K	
 bins.sh	2021-08-29 08:42	2.4K	

Index of /bins

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
🔙 Parent Directory		-	
? daddyl33t.arm	2021-09-05 05:59	159K	
? daddyl33t.arm5	2021-09-05 05:59	154K	
? daddyl33t.arm6	2021-09-05 05:59	171K	
? daddyl33t.arm7	2021-09-05 05:59	208K	
? daddyl33t.i686	2021-09-05 05:59	124K	
? daddyl33t.mips	2021-09-05 05:59	197K	
? daddyl33t.mpsl	2021-09-05 05:59	197K	
? daddyl33t.ppc	2021-09-05 05:59	145K	
? daddyl33t.sh4	2021-09-05 05:59	138K	
? daddyl33t.x86	2021-09-05 05:59	120K	
? daddyl33t.x86_64	2021-09-05 05:59	138K	

A new campaign just launched with a new set of malicious binaries

This new campaign goes by the tag-name **daddyl33t** as it is revealed by its supposed creator.

```
daddyl33t's back  
[main] bot deployed
```

Daddyl33t, the creator

The campaigns are short lived for many reasons:

- The campaign operator might want to hold on to the surprise element as long as possible: traditional antivirus engines do not usually detect the samples on release, as they need some time to build up detection.
- A compromised Command-and-Control server could be under siege from many different threat actors: as they fight to keep their own ground, new players could come in by exploiting vulnerable servers and overwrite the malicious binaries with their own campaign, distributing a different set of binaries from that point on.
- Also, as new source codes are released on cybercrime or other underground forums, campaign operators adjust and update their malicious tools whenever there is a better malware version with more or better features.

Creating a flavor of Mirai/Gafgyt has never been so easy. The leaked source codes of Mirai and Gafgyt/QBot are all over GitHub and other repositories, and implementing new functions, removing unnecessary features, and adjusting malicious tools with recent exploits (as new vulnerabilities are discovered) is widely practiced by *script-kiddies*.

Coverage

The malicious IPs and URLs related to the Korpze campaign are blocked by [CUJO AI Sentry](#).

Indicators of Compromise

SHA256

2be9013823dbcb7dd4cbcd30e37ffd51ac9b3a0f78d168879c6a59ff1b2704d8
009f8f752458e6bbd340ca3cd34f5ebc520b2846fdbb5339add824d31f195413

Campaign name

“Korpze1233121337”

IP

103[.]161[.]17[.]233 – ASN 135967 – Vietnam

C2

103[.]161[.]17[.]233:1227
103[.]161[.]17[.]233:1228
103[.]161[.]17[.]233:1229

URL

[http://103\[.\]161\[.\]17\[.\]233/bins\[.\]sh](http://103[.]161[.]17[.]233/bins[.]sh)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]arm](http://103[.]161[.]17[.]233/Korpze1233121337[.]arm)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]arm4](http://103[.]161[.]17[.]233/Korpze1233121337[.]arm4)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]arm5](http://103[.]161[.]17[.]233/Korpze1233121337[.]arm5)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]arm6](http://103[.]161[.]17[.]233/Korpze1233121337[.]arm6)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]m68k](http://103[.]161[.]17[.]233/Korpze1233121337[.]m68k)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]mips](http://103[.]161[.]17[.]233/Korpze1233121337[.]mips)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]x86](http://103[.]161[.]17[.]233/Korpze1233121337[.]x86)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]ppc](http://103[.]161[.]17[.]233/Korpze1233121337[.]ppc)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]sparc](http://103[.]161[.]17[.]233/Korpze1233121337[.]sparc)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]i586](http://103[.]161[.]17[.]233/Korpze1233121337[.]i586)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]i686](http://103[.]161[.]17[.]233/Korpze1233121337[.]i686)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]mpsl](http://103[.]161[.]17[.]233/Korpze1233121337[.]mpsl)

[http://103\[.\]161\[.\]17\[.\]233/Korpze1233121337\[.\]sh4](http://103[.]161[.]17[.]233/Korpze1233121337[.]sh4)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]arm](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]arm)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]arm4](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]arm4)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]arm5](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]arm5)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]arm6](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]arm6)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]m68k](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]m68k)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]mips](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]mips)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]x86](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]x86)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]ppc](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]ppc)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]sparc](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]sparc)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]i586](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]i586)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]i686](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]i686)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]mpsl](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]mpsl)

[http://103\[.\]161\[.\]17\[.\]233/bins/Korpze1233121337\[.\]sh4](http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]sh4)



Albert Zsigovits

Malware Researcher



CUJO AI Lens

An AI-powered analytics solution that, for the first time, gives operators an aggregated, dynamic and near real-time view into the way end users utilize their home or business networks

[Learn more](#)



Explorer

Provides complete, programmatic access to granular data via APIs to all the information collected and processed by the CUJO AI Platform

[Learn more](#)



Compass

An advanced service that empowers families and businesses to define and manage how their members' online activity affects their everyday lives

[Learn more](#)

Other posts by Albert Zsigovits

[All posts by Albert Zsigovits](#)

Privacy Overview

This website uses cookies to improve your experience while you navigate through the website. Out of these, the cookies that are categorized as necessary are stored on your browser as they are essential for the working of basic functionalities of the website. We also use third-party cookies that help us analyze and understand how you use this website. These cookies will be stored in your browser only with your consent. You also have the option to opt-out of these cookies. But opting out of some of these cookies may affect your browsing experience.

Necessary cookies are absolutely essential for the website to function properly. These cookies ensure basic functionalities and security features of the website, anonymously.

Cookie	Duration	Description
_GRECAPTCHA	5 months 27 days	This cookie is set by the Google recaptcha service to identify bots to protect the website against malicious spam attacks.
cookielawinfo-checkbox-advertisement	1 year	Set by the GDPR Cookie Consent plugin, this cookie is used to record the user consent for the cookies in the "Advertisement" category .
cookielawinfo-checkbox-analytics	11 months	This cookie is set by GDPR Cookie Consent plugin. The cookie is used to store the user consent for the cookies in the category "Analytics".
cookielawinfo-checkbox-analytics	11 months	This cookie is set by GDPR Cookie Consent plugin. The cookie is used to store the user consent for the cookies in the category "Analytics".

Cookie	Duration	Description
cookielawinfo-checkbox-functional	11 months	The cookie is set by GDPR cookie consent to record the user consent for the cookies in the category "Functional".
cookielawinfo-checkbox-necessary	11 months	This cookie is set by GDPR Cookie Consent plugin. The cookies is used to store the user consent for the cookies in the category "Necessary".
cookielawinfo-checkbox-others	11 months	This cookie is set by GDPR Cookie Consent plugin. The cookie is used to store the user consent for the cookies in the category "Other".
cookielawinfo-checkbox-performance	11 months	This cookie is set by GDPR Cookie Consent plugin. The cookie is used to store the user consent for the cookies in the category "Performance".
cujo_cerber_*	1 day	Secures the website by detecting and mitigating malicious activity.
viewed_cookie_policy	11 months	The cookie is set by the GDPR Cookie Consent plugin and is used to store whether or not user has consented to the use of cookies. It does not store any personal data.

Functional cookies help to perform certain functionalities like sharing the content of the website on social media platforms, collect feedbacks, and other third-party features.

Performance cookies are used to understand and analyze the key performance indexes of the website which helps in delivering a better user experience for the visitors.

Analytical cookies are used to understand how visitors interact with the website. These cookies help provide information on metrics the number of visitors, bounce rate, traffic source, etc.

Cookie	Duration	Description
_ga	session	The _ga cookie, installed by Google Analytics, calculates visitor, session and campaign data and also keeps track of site usage for the site's analytics report. The cookie stores information anonymously and assigns a randomly generated number to recognize unique visitors.
_gat_gtag_UA_128580456_1	session	Set by Google to distinguish users.

Cookie	Duration	Description
_gid	session	Installed by Google Analytics, _gid cookie stores information on how visitors use a website, while also creating an analytics report of the website's performance. Some of the data that are collected include the number of visitors, their source, and the pages they visit anonymously.

Advertisement cookies are used to provide visitors with relevant ads and marketing campaigns. These cookies track visitors across websites and collect information to provide customized ads.

Other uncategorized cookies are those that are being analyzed and have not been classified into a category as yet.