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

By Albert Zsigovits

Published: 2021-09-07 · Archived: 2026-04-05 13:16:38 UTC

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.


new botnet gafgyt mirai unstripped binary
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 0n 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:

- https://github.com/m1lw0rm/Yakuza

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
telnetadmin	telnetadmin	–

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.


udp flood function botnet
UDP Flood initialized in openvpn_swak function

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


icmp flood funtion botnet
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 8.8.8.8nnameserver 8.8.4.4n";
 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 no 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 **ovhl7**, **SendHTTP,** and **SendHTTPCloudflare**. Once the DoS module is launched at a target, the function randomly chooses a User-Agents to attack with.


DoS module using randomized user agents
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.


old botnet campaign binary


new botnet campaign binary
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 new botnet campaignbinary
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

> *2be9013823dbcb7dd4cbed30e37ffd51ac9b3a0f78d168879c6a59ff1b2704d8*
>
> *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/Korpze1233121337[.]arm*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]arm4*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]arm5*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]arm6*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]m68k*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]mips*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]x86*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]ppc*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]sparc*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]i586*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]i686*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]mpsl*
>
> *http://103[.]161[.]17[.]233/Korpze1233121337[.]sh4*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]arm*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]arm4*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]arm5*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]arm6*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]m68k*

> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]mips*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]x86*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]ppc*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]sparc*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]i586*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]i686*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]mpsl*
>
> *http://103[.]161[.]17[.]233/bins/Korpze1233121337[.]sh4*
