SECTRIO

MALWARE REPORT

Mirai-Backdoor

Date: **19/06/2020**

Sanjuktasree Chatterjee

The main intention of gaining access to any system is to conduct an attack. The attacker can gain persistence into the compromised system through some of the different mechanism including backdoor. There is nothing new in the malware family backdoor, it is targeting most of the operating system over the years. Backdoor is also installed into the IoT devices. In this blog, it will be discussed about a new version of Mirai Backdoor installing onto the target machine to do the further attack.

HISTORY

- In 2003, there was a Linux Backdoor attempt was failed because of the code that was in CVS Repository.
- In Nov 2013, Security researchers had discovered a Linux Backdoor that had been used SSH protocol to disguise its presence on compromised system.
- In Aug 2016, OpenSSH backdoor was used to compromise the Linux Server.
- The backdoor is used to steal the credentials and maintain access to a compromised server. There is a shared library that is loaded by all OpenSSH executable file such as ssh, sshd and ssh-agent.
- On July 2019, Trend Micro detected Mirai Backdoor by using their product and also deleted it after detection.
- From the past few infections, it suggests that Mirai Backdoor are capable of infecting wide variety of devices including x64, x86, ARM, ARM64 architectures and many other architectures.
- This version of Mirai Backdoor was intercepted by Subex's honeypot on May 17th, 2020.

FEATURES

The process of infection of this malware is started with installing into a machine. After that it started a network communication as the process '`sshd`' and send multiple '`SYN`' packet at a short period of time to a domain which is malicious and which has a connection with multiple `elf` files and `bash` scripts which are also malicious.

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      1 Linux-Malware:47950     painstress.online:38221 SYN_SENT
tcp        0      1 Linux-Malware:47952     painstress.online:38221 SYN_SENT
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
udp        0      0 localhost:domain        0.0.0.0:*
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp        0      0 0.0.0.0:40471           0.0.0.0:*
udp6       0      0 [::]:mdns               [::]:*
udp6       0      0 [::]:57631              [::]:*
raw6       0      0 [::]:ipv6-icmp          [::]:*                  7
Active UNIX domain sockets (servers and established)
```

Fig 1: Network Connection made by Backdoor

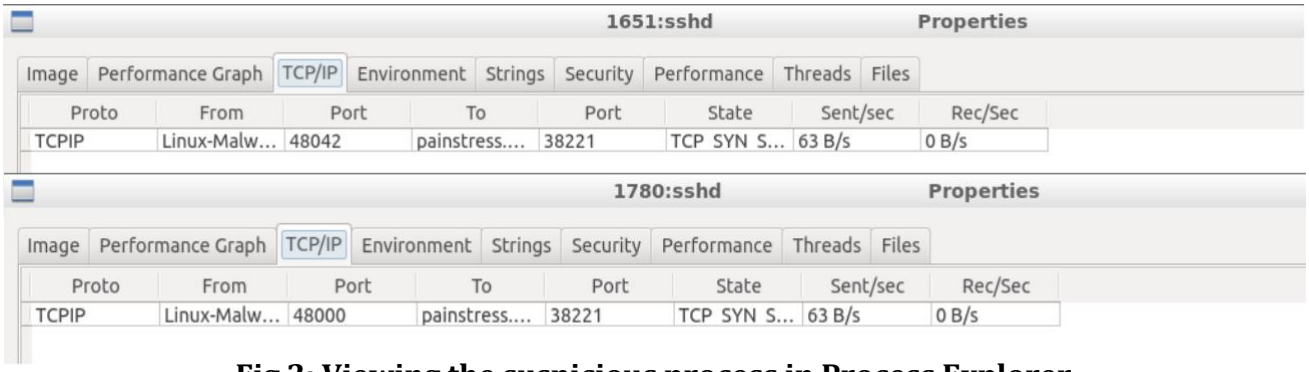


Fig 2: Viewing the suspicious process in Process Explorer

The domain that is communicated by the malware is:

- `painstress[.]online`

The domain is assigned to an IP which is marked as malicious by various antivirus vendor. By executing the file, it will create multiple process as '`sshd`'. The process is communicating with the domain with TCP connection via an unassigned port number 38221.

The file gives this message when we execute it. This message was also shown by previous version of Mirai botnet and DDOS Malware.

Fig 3: Message Given by Execu

Now the port number had checked that creates `sshd` communication with different PID. So, it is clear that there was multiple `sshd` communication with the same domain.

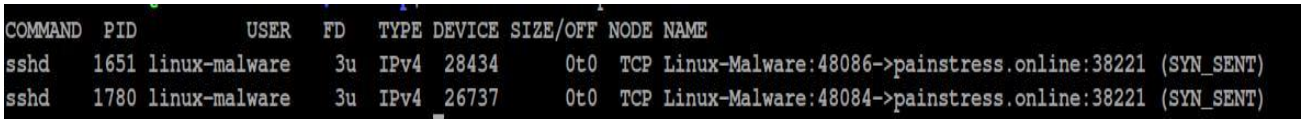


Fig 4: Checking PID using Port Number

This is a directory that was created '`/dev/pts`' when there is any telnet or ssh connection made remotely. We can see the changes that when the process will be finished the values will be deleted.

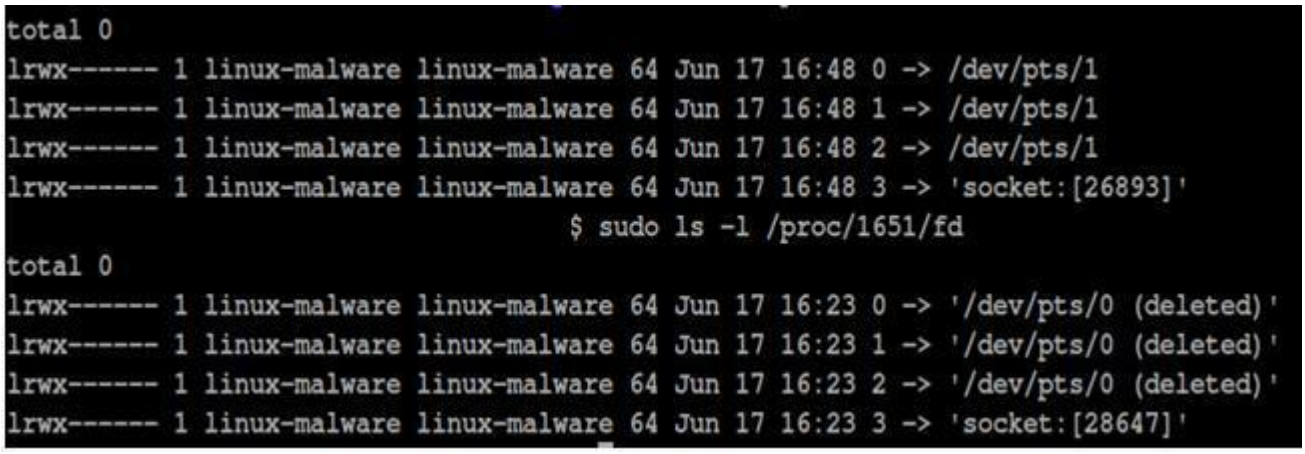


Fig 5: Files that are accessed by these PIDs

The sample has been collected and analyzed. It has '`Intel`' architecture.

Fig 6: Information about architecture

The file is an executable and 32 bits, containing the ELF header. The information is shown below:

```
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 03 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - GNU
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Intel 80386
  Version:                           0x1
  Entry point address:               0x804f580
  Start of program headers:          52 (bytes into file)
  Start of section headers:          0 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         3
  Size of section headers:           40 (bytes)
  Number of section headers:         0
  Section header string table index: 0

There are no sections in this file.

There are no sections to group in this file.

Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  LOAD           0x000000 0x08048000 0x08048000 0x08774 0x08774 R E 0x1000
  LOAD           0x000000 0x08051000 0x08051000 0x00000 0x0fb84 RW  0x1000
  GNU_STACK      0x000000 0x00000000 0x00000000 0x00000 0x00000 RW  0x4

There is no dynamic section in this file.

There are no relocations in this file.

The decoding of unwind sections for machine type Intel 80386 is not currently supported.

Dynamic symbol information is not available for displaying symbols.

No version information found in this file.
```

Fig 7: ELF Header

The file is opened in ELF Parser and it is found that it does not contain any section headers and also found that the sections have been obfuscated. In the ELF file these headers must be present, because the dynamic linker uses them for sanity checking. Some creator tries to strip them in an attempt to obfuscate the binary and prevent reverse engineering by using some packer to pack the file.

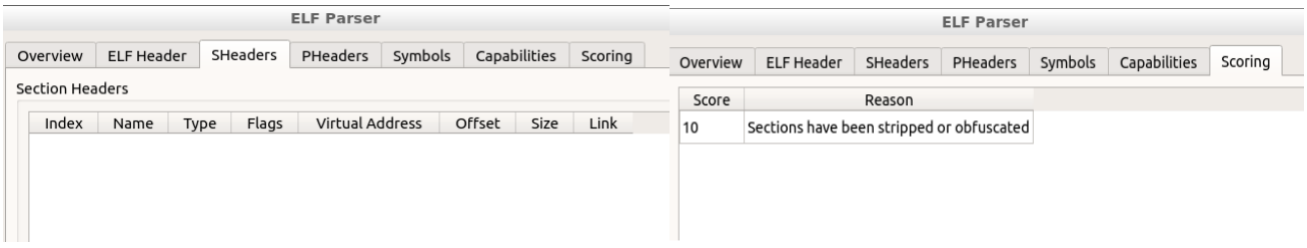


Fig 8: Internal structure of ELF

Analysis on Ghidra shows that the sample is having data in the code. There are only few codes that are visible. It means the sample is packed.

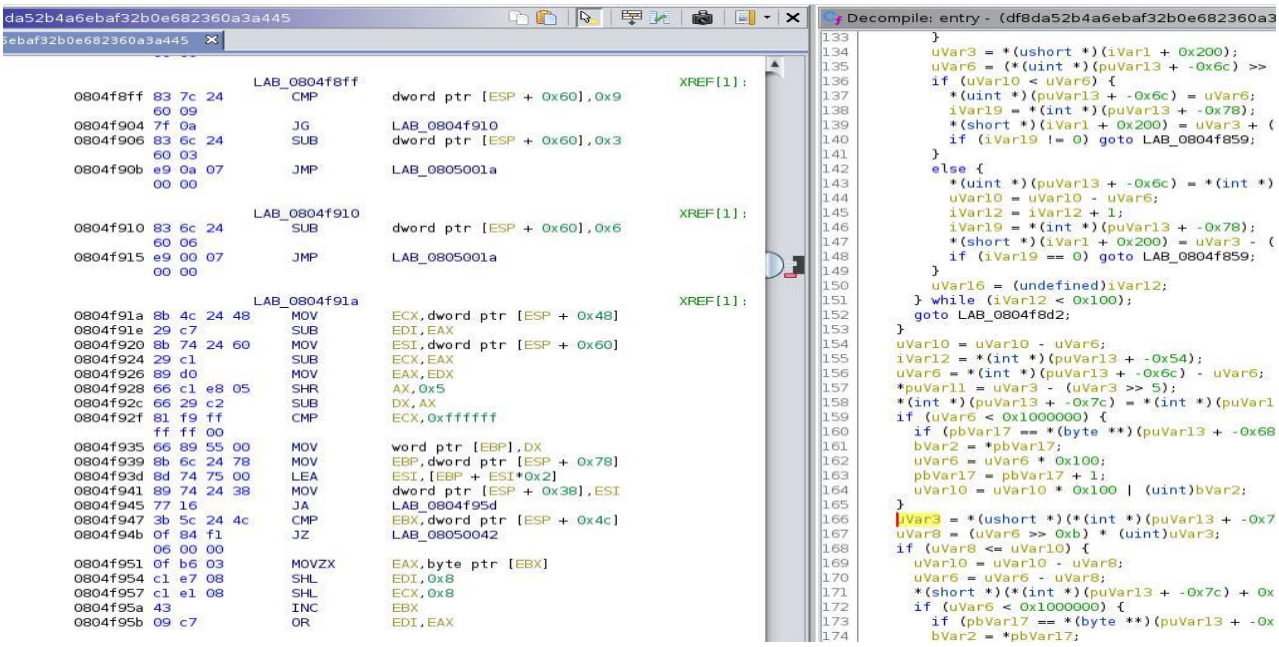


Fig 9: Analyzing Code

It was found that the process ID which had been used by the executable has shown:

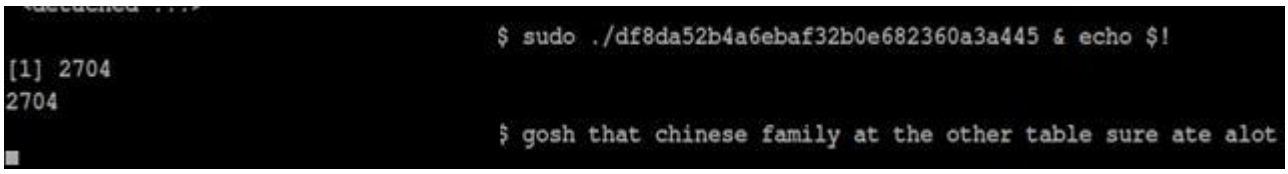


Fig 10: PID of Executable

The malware is executed in the '`/home`' directory that is shown below. It means no other process is created by this malware to do the malicious activity to any other directory in the targeted system. The changes and the malicious activity that has been done in the /home.

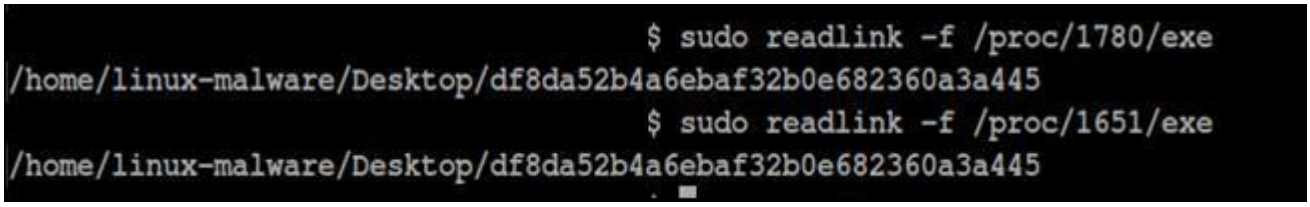
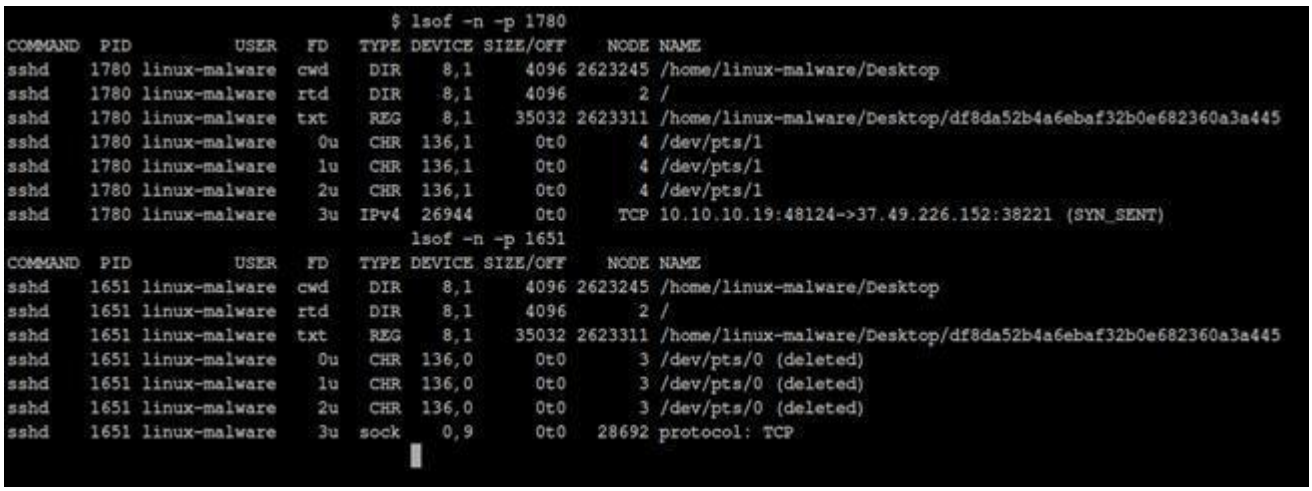


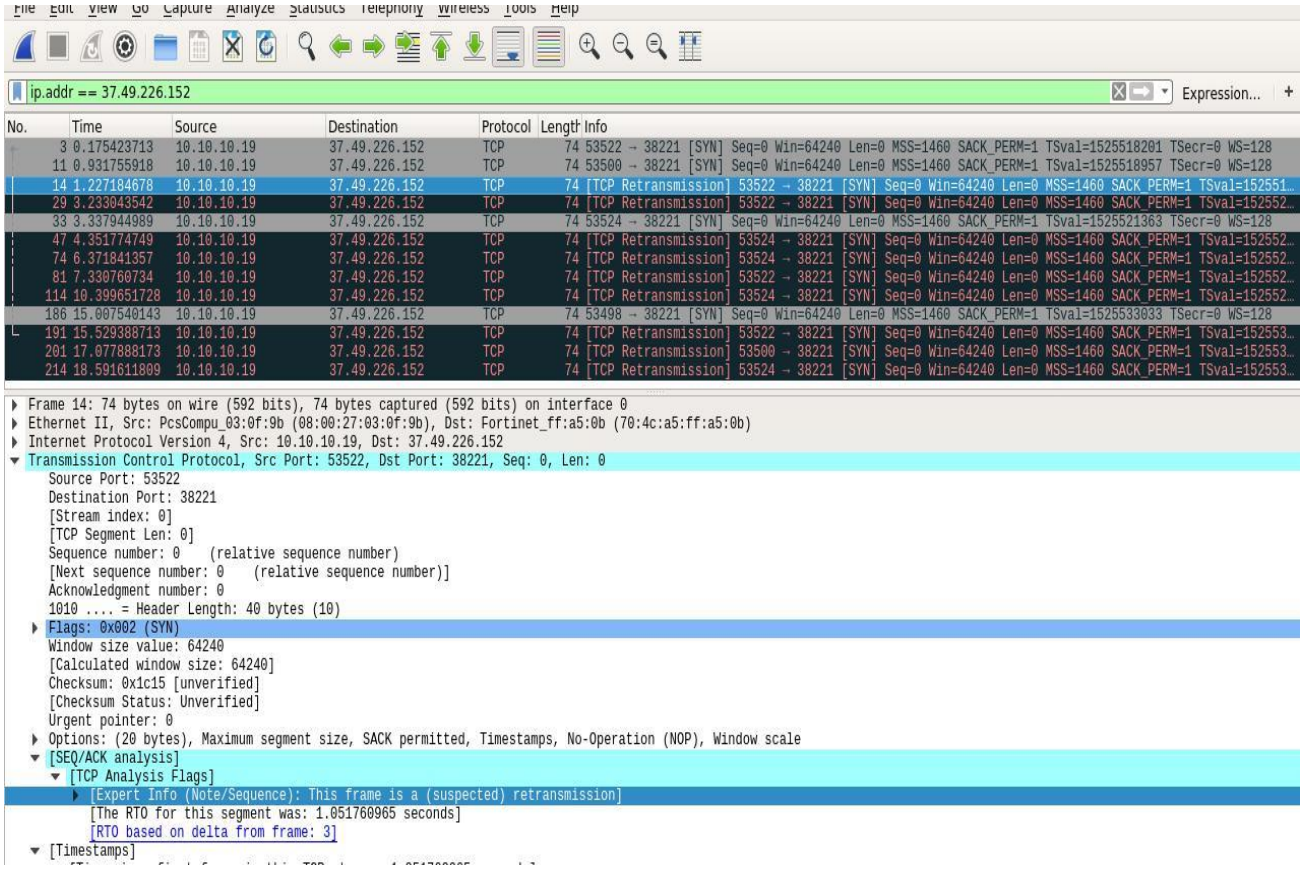
Fig 11: Directory where it Executes



```
                                $ lsof -n -p 1780
COMMAND  PID          USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
sshd    1780 linux-malware  cwd    DIR    8,1     4096 2623245 /home/linux-malware/Desktop
sshd    1780 linux-malware  rtd    DIR    8,1     4096       2 /
sshd    1780 linux-malware  txt    REG    8,1    35032 2623311 /home/linux-malware/Desktop/df8da52b4a6ebaf32b0e682360a3a445
sshd    1780 linux-malware    0u   CHR  136,1      0t0       4 /dev/pts/1
sshd    1780 linux-malware    1u   CHR  136,1      0t0       4 /dev/pts/1
sshd    1780 linux-malware    2u   CHR  136,1      0t0       4 /dev/pts/1
sshd    1780 linux-malware    3u  IPv4  26944      0t0     TCP 10.10.10.19:48124->37.49.226.152:38221 (SYN_SENT)
                                lsof -n -p 1651
COMMAND  PID          USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
sshd    1651 linux-malware  cwd    DIR    8,1     4096 2623245 /home/linux-malware/Desktop
sshd    1651 linux-malware  rtd    DIR    8,1     4096       2 /
sshd    1651 linux-malware  txt    REG    8,1    35032 2623311 /home/linux-malware/Desktop/df8da52b4a6ebaf32b0e682360a3a445
sshd    1651 linux-malware    0u   CHR  136,0      0t0       3 /dev/pts/0 (deleted)
sshd    1651 linux-malware    1u   CHR  136,0      0t0       3 /dev/pts/0 (deleted)
sshd    1651 linux-malware    2u   CHR  136,0      0t0       3 /dev/pts/0 (deleted)
sshd    1651 linux-malware    3u  sock    0,9      0t0   28692 protocol: TCP
```

Fig 12: Files that are accessed by PIDs

These are the list of files that have been used by those PIDs. The socket is used because there is network connection that has been made by the malware to communicate with various IPs and domains. In the below screenshot it has been shown that there was an IP in which the SYN packets are send. The IP is further checked and in result it is found malicious.



```
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 37.49.226.152                                                    Expression...  +
No.  Time          Source          Destination     Protocol Length Info
   3 0.175423713   10.10.10.19     37.49.226.152   TCP        74 53522 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1525518201 TSecr=0 WS=128
  11 0.931755918   10.10.10.19     37.49.226.152   TCP        74 53500 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1525518957 TSecr=0 WS=128
  14 1.227184678   10.10.10.19     37.49.226.152   TCP        74 [TCP Retransmission] 53522 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=152551…
  29 3.233043542   10.10.10.19     37.49.226.152   TCP        74 [TCP Retransmission] 53522 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=152552…
  33 3.337944989   10.10.10.19     37.49.226.152   TCP        74 53524 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1525521363 TSecr=0 WS=128
  47 4.351774749   10.10.10.19     37.49.226.152   TCP        74 [TCP Retransmission] 53524 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=152552…
  74 6.371841357   10.10.10.19     37.49.226.152   TCP        74 [TCP Retransmission] 53524 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=152552…
  81 7.330760734   10.10.10.19     37.49.226.152   TCP        74 [TCP Retransmission] 53522 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=152552…
 114 10.399651728  10.10.10.19     37.49.226.152   TCP        74 [TCP Retransmission] 53524 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=152552…
 186 15.007540143  10.10.10.19     37.49.226.152   TCP        74 53498 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1525533033 TSecr=0 WS=128
 191 15.529388713  10.10.10.19     37.49.226.152   TCP        74 [TCP Retransmission] 53522 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=152553…
 201 17.077888173  10.10.10.19     37.49.226.152   TCP        74 [TCP Retransmission] 53500 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=152553…
 214 18.591611809  10.10.10.19     37.49.226.152   TCP        74 [TCP Retransmission] 53524 → 38221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=152553…
▸ Frame 14: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_03:0f:9b (08:00:27:03:0f:9b), Dst: Fortinet_ff:a5:0b (70:4c:a5:ff:a5:0b)
▸ Internet Protocol Version 4, Src: 10.10.10.19, Dst: 37.49.226.152
▾ Transmission Control Protocol, Src Port: 53522, Dst Port: 38221, Seq: 0, Len: 0
    Source Port: 53522
    Destination Port: 38221
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    1010 .... = Header Length: 40 bytes (10)
  ▸ Flags: 0x002 (SYN)
    Window size value: 64240
    [Calculated window size: 64240]
    Checksum: 0x1c15 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
  ▾ [SEQ/ACK analysis]
    ▾ [TCP Analysis Flags]
      ▸ [Expert Info (Note/Sequence): This frame is a (suspected) retransmission]
        [The RTO for this segment was: 1.051760965 seconds]
        [RTO based on delta from frame: 3]
  ▾ [Timestamps]
```

Fig 13: Network Analysis by WireShark

The network analysis is done to show the result related to the IP. As in the given screenshot we can see that there are multiple SYN packets are sent to the malicious IP. And TCP Retransmission happens multiple time. In detail analysis of that packet it has shown there is suspicious retransmission happened to the particular IP.

This network connection is also shown via strace. The connection is closed again and again and the backdoor is trying to connect to the IP by sending `SYN` packet.

```
connect(3, {sa_family=AF_INET, sin_port=htons(38221), sin_addr=inet_addr("37.49.226.152")}, 16) = -1 EINPROGRESS (Operation no
w in progress)
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
connect(3, {sa_family=AF_INET, sin_port=htons(38221), sin_addr=inet_addr("37.49.226.152")}, 16) = -1 EINPROGRESS (Operation no
w in progress)
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
connect(3, {sa_family=AF_INET, sin_port=htons(38221), sin_addr=inet_addr("37.49.226.152")}, 16) = -1 EINPROGRESS (Operation no
w in progress)
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
close(3)                                = 0
socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3
connect(3, {sa_family=AF_INET, sin_port=htons(38221), sin_addr=inet_addr("37.49.226.152")}, 16) = -1 EINPROGRESS (Operation no
```

Fig 14: Result of strace

In the analysis we had checked the parent PID of those PIDs and these PIDs are also analyzed. There are multiple files are accessed by PID. In which the /dev/pts is used for establishing remote connections.

The `/proc/self/exe` command along with the info, it suggest that the information linked with the current process which are currently running on the system that will be discovered. When a process tries to access /proc/self/, it will generate its content dynamically by reading data from that process. The /proc/self/ behaves as a symlink, that means a file contains a reference to another file in a form of absolute or relative path.

The file /dev/watchdog is opened which is detected by analyzing the syscall by strace. This file is used in remote location equipment to do the automatic hardware reset which can also be used by backdoor. It is an attempt to document the existing usage which will allow future driver writers to use this as a reference.

FILES OPENED:

/proc/self/exe	/proc/net/route
/dev/watchdog	/dev/misc/watchdog
/etc/ld.so.nohwcap	/etc/ld.so.preload
/usr/lib/locale/locale-archive	/etc/ld.so.cache
/usr/share/locale/locale.alias	/dev/pts
/lib/x86_64-linux-gnu/libc.so.6	/usr/share/locale-langpack/en_IN/LC_MESSAGES/coreutils.mo
/proc/self/mountinfo	/dev/sda
/usr/share/locale/en_IN/LC_MESSAGES/coreutils.mo	/usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache
/lib/x86_64-linux-gnu/libnss_files-2.27.so	/lib/x86_64-linux-gnu/libnss_nis-2.27.so
/lib/x86_64-linux-gnu/libnsl-2.27.so	/lib/x86_64-linux-gnu/libnss_compat-2.27.so
/lib/x86_64-linux-gnu/libtinfo.so.5.9	/lib/x86_64-linux-gnu/libc-2.27.so
/lib/x86_64-linux-gnu/libdl-2.27.so	/lib/x86_64-linux-gnu/ld-2.27.so

VULNERABILITIES TARGETED:

CVE-2006-3470	CVE-2006-6302
CVE-2007-2791	CVE-2010-5107
CVE-2011-2294	CVE-2014-2721

CVE-2014-3348	CVE-2015-2897
CVE-2015-0924	CVE-2015-4236
CVE-2015-6565	CVE-2016-10009
CVE-2017-5803	CVE-2018-12336
CVE-2018-12338	CVE-2019-12147
CVE-2019-9160	CVE-2020-10364

The IPs and domains that are found to be malicious by the analysis are:

- 185[.]255[.]130[.]202
- 37[.]49[.]226[.]152
- painstress[.]online

The hash of the file that has been analyzed:

- df8da52b4a6ebaf32b0e682360a3a445

Sectrio Protection

Sectrio detects this malware as **'**SS_Checksum_SSHDoor_B**'.**

Our Honeypot Network

This report has been prepared from the threat intelligence gathered by our honeypot network. This honeypot network is today operational in 72 cities across the world. These cities have at least one of the following attributes:

- **Are landing centers for submarine cables**
- **Are internet traffic hotspots**
- **House multiple IoT projects with a high number of connected endpoints**

- **House multiple connected critical infrastructure projects**
- **Have academic and research centers focusing on IoT**
- **Have the potential to host multiple IoT projects across domains in the future**

Over 12 million attacks a day is being registered across this network of individual honeypots. These attacks are studied, analyzed, categorized, and marked according to a threat rank index, a priority assessment framework that we have developed within Sectrio. The honeypot network includes over 4000 physical and virtual devices covering over 400 device architectures and varied connectivity mediums globally. These devices are grouped based on the sectors they belong to for purposes of understanding sectoral attacks. Thus, a layered flow of threat intelligence is made possible.