Stantinko botnet adds cryptomining to its pool of criminal activities

November 26, 2019

ESET researchers have discovered that the criminals behind the Stantinko botnet are distributing a cryptomining module to the computers they control

Vladislav Hrčka
26 Nov 2019 - 11:30AM

ESET researchers have discovered that the criminals behind the Stantinko botnet are distributing a cryptomining module to the computers they control

The operators of the Stantinko botnet have expanded their toolset with a new means of profiting from the computers under their control. The roughly half-million-strong botnet – known to have been active since at least 2012 and mainly targeting users in Russia, Ukraine, Belarus and Kazakhstan – now distributes a cryptomining module. Mining Monero, a cryptocurrency whose exchange rate has oscillated in 2019 between US$50 and US$110, has been the botnet's monetizing functionality since at least August 2018. Before that, the botnet performed click fraud, ad injection, social network fraud and password stealing attacks.

In this article, we describe Stantinko's cryptomining module and provide an analysis of its functionality.

This module's most notable feature is the way it is obfuscated to thwart analysis and avoid detection. Due to the use of source level obfuscations with a grain of randomness and the fact that Stantinko's operators compile this module for each new victim, each sample of the module is unique.

We will describe the module’s obfuscation techniques and offer, in a separate article for fellow malware analysts, a possible approach to deal with some of them.

Since Stantinko is constantly developing new and improving its existing custom obfuscators and modules, which are heavily obfuscated, it would be backbreaking to track each minor improvement and change that it introduces. Therefore, we decided to mention and describe only what we believe are significant adjustments in comparison with earlier samples relative to the state in which the module is to be described. After all, we intend just to describe the module as it currently is in this article.

Modified open-source cryptominer

Stantinko’s cryptomining module, which exhausts most of the resources of the compromised machine by mining a cryptocurrency, is a highly modified version of the xmr-stak open-source cryptominer. All unnecessary strings and even whole functionalities were removed in attempts to evade detection. The remaining strings and functions are heavily obfuscated. ESET security products detect this malware as Win{32,64}/CoinMiner.Stantinko.

Use of mining proxies

CoinMiner.Stantinko doesn’t communicate with its mining pool directly, but via proxies whose IP addresses are acquired from the description text of YouTube videos. A similar technique to hide data in descriptions of YouTube videos is used by the banking malware Casbaneiro. Casbaneiro uses much more legitimate-looking channels and descriptions, but for much the same purpose: storing encrypted C&Cs.

The description of such a video consists of a string composed of mining proxy IP addresses in hexadecimal format. For example, the YouTube video seen in Figure 1 has the description “03101f1712dec626“, which corresponds to two IP addresses in hexadecimal format – 03101f17 corresponds to 3.16.31[.]23 in decimal dotted-quad format, and 12dec626 is 18.222.198[.]38. As of the time of writing, the format has been slightly adjusted. The IP addresses are currently enclosed in “!!!!”, which simplifies the very process of parsing and prevents possible changes of the YouTube video HTML structure turning the parser dysfunctional.

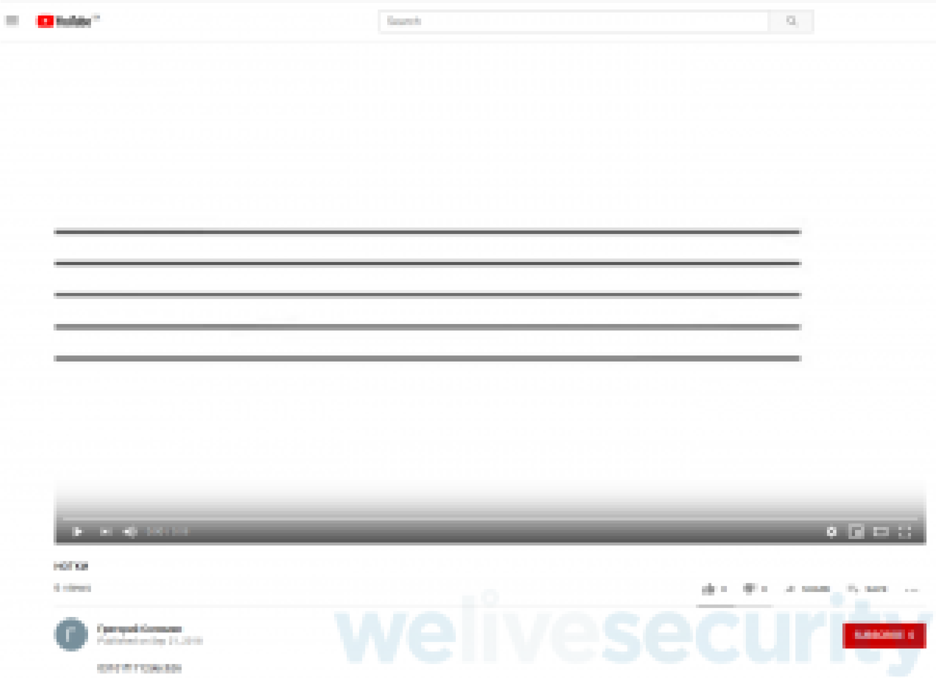

Figure 1. Example YouTube video whose description provides an IP address for the module’s communication with the mining pool

In earlier versions, the YouTube URL was hardcoded in the CoinMiner.Stantinko binary. Currently the module receives a video identifier as a command line parameter instead. This parameter is then used to construct the YouTube URL, in the form https://www.youtube.com/watch?v=%PARAM%. The cryptomining module is executed either by Stantinko’s BEDS component, or by rundll32.exe via a batch file that we have not captured, with the module loaded from a local file system location of the form %TEMP%\%RANDOM%\%RANDOM_GUID%.dll.

We informed YouTube of this abuse; all the channels containing these videos were taken down.

Cryptomining capabilities

We have divided the cryptomining module into four logical parts, which represent distinct sets of capabilities. The main part performs the actual cryptomining; the other parts of the module are responsible for additional functions:

- suspending other (i.e. competing) cryptomining applications
- detecting security software

- suspending the cryptomining function if the PC is on battery power or when a task manager is detected, to prevent being revealed by the user

Cryptomining

At the very core of the cryptomining function lies the process of hashing, and communication with the proxy. The method of obtaining the list of mining proxies is described above; CoinMiner.Stantinko sets the communication with the first mining proxy it finds alive.

Its communication takes place over TCP and is encrypted by RC4 with a key consisting of the first 26 characters of the number pi (including the decimal separator, hardcoded in the string “3,14159265358979323846264З“) and then base64 encoded; the same key is used in all samples we have seen.

The code of the hashing algorithm is downloaded from the mining proxy at the beginning of the communication and loaded into memory – either directly or, in earlier versions, from the library libcr64.dll that is first dropped onto the disk.

Downloading the hashing code with each execution enables the Stantinko group to change this code on the fly. This change makes it possible, for example, to adapt to adjustments of algorithms in existing currencies and to switch to mining other cryptocurrencies in order, perhaps, to mine the most profitable cryptocurrency at the moment of execution. The main benefit of downloading the core part of the module from a remote server and loading it directly into memory is that this part of the code is never stored on disk. This additional adjustment, which is not present in earlier version, is aimed at complicating detection because patterns in these algorithms are trivial for security products to detect.

All instances of Stantinko’s cryptomining module we’ve analyzed mine Monero. We deduced this from the jobs provided by the mining proxy and the hashing algorithm. For example, Figure 2 is a job sent by one of the proxies.

```
{“error”:null,”result”:{“status”:”OK”}}
{“method”:”job”,”params”:”blob”:”0b0bbfdee1e50567042dcfdfe96018227f25672544521f8ee2564cf8b4c3139a6a88c5f0b32664000000a1c8ee5c18
```

Figure 2. Example mining job received from a mining pool proxy

We analyzed the hashing algorithm used and found that it was CryptoNight R. Since there are multiple cryptocurrencies that use this algorithm, its recognition alone isn’t sufficient; it just shortens the list. One can see in the provided job that the height of the blockchain was 1815711 at the time, so we had to find currencies using CryptoNight R with this height on dedicated block explorers which lead us to Monero. Dissecting the string 0b0bbfdee1e50567042dcfdfe96018227f25672544521f8ee2564cf8b4c3139a6a88c5f0b32664000000a1c8ee5c185ed2661daab9d0c454fd40e9f53 reveals that the hash of the previous block (67042dcfdfe96018227f25672544521f8ee2564cf8b4c3139a6a88c5f0b32664) and timestamp (1555590859) indeed fits into Monero’s blockchain at the height of 1815711. One can find the structure of the blob by examining its generator function in the source code of Monero . The generator function exposes another structure called a block header which contains both the hash of the previous block and timestamp.

Unlike the rest of CoinMiner.Stantinko, the hashing algorithm isn’t obfuscated, since obfuscation would significantly impair the speed of hash calculation and hence overall performance and profitability. However, the authors still made sure not to leave any meaningful strings or artifacts behind.

Suspension of other cryptominers

The malware enumerates running processes searching for other cryptominers. If any competitors are found, Stantinko suspends all their threads.

CoinMiner.Stantinko considers a process to be a cryptominer if its command line contains a particular string, or a combination, which vary from sample to sample; for example:

- minerd
- minergate
- xmr
- cpservice
- vidservice and stratum+tcp://
- stratum://
- -u and pool
- “-u and pool
- “-u and xmr
- -u and xmr
- -u and mining
- “-u and mining
- -encodedcommand and exe
- –donate-level
- windows and -c and cfgi
- regsvr32 and /n and /s and /q

- application data and exe
- appdata and exe

These strings refer to the following legitimate cryptominers: https://github.com/pooler/cpuminer, https://minergate.com/, https://github.com/xmrig, and even https://github.com/fireice-uk/xmr-stak – which, interestingly, is the very miner this Stantinko module is based on. The strings also lead to various uninteresting malware samples containing cryptomining functionality.

Of interest is that the Stantinko operators are known to have tried to get rid of competing code in the past. However, they relied on the legitimate AVZ Antiviral Toolkit fed with a script written in its built-in scripting language for this task.

Detection prevention

CoinMiner.Stantinko temporarily suspends mining if it detects there's no power supply connected to the machine. This measure, evidently aimed at portable computers, prevents fast battery draining … which might raise the user's suspicion.

Also, it temporarily suspends mining if a task manager application (a process named procexp64.exe, procexp.exe or taskmgr.exe) is detected running.

The malware also scans running processes to find security software and again task managers. It calculates the CRC-32 of the process's name and then checks it against a hardcoded list of CRC-32 checksums, which is included in the Appendix. In general this technique can help evade detection, since the process names of those security products are not included in the binary – adding a bit more stealth by not containing the process names directly. It also makes it harder for analysts to find out what the malware authors are after because one has to crack these hashes, which is technically the same problem as password cracking. However, using a list of known process names is usually sufficient to determine the exact names.

Should a CRC-32 match be found, the CRC is written to a log file (api-ms-win-crt-io-l1-1-0.dll). The log file is presumably exfiltrated later by some Stantinko component that we have not seen, since there's no other functionality related to it in this module.

Obfuscation

Besides its cryptomining features, CoinMiner.Stantinko is notable also for its obfuscation techniques aimed at avoiding detection and thwarting analysis. Some of those techniques are unique and we will describe them in detail in a follow-up article.

Conclusion

Our discovery shows that the criminals behind Stantinko continue to expand the ways they leverage the botnet they control. Their previous innovations were distributed dictionary-based attacks on Joomla and WordPress web sites aimed at harvesting server credentials, probably with the goal of selling them to other criminals.

This remotely configured cryptomining module, distributed since at least August of 2018 and still active at the time of writing, shows this group continues to innovate and extend its money-making capabilities. Besides its standard cryptomining functionality, the module employs some interesting obfuscation techniques that we will disclose, along with some possible countermeasures, in an upcoming article.

Indicators of Compromise (IoCs)

ESET detection names

Win32/CoinMiner.Stantinko
Win64/CoinMiner.Stantinko

SHA-1

A full list of more than 1,000 hashes is available from our GitHub repository.

00F0AED42011C9DB7807383868AF82EF5454FDD8
01504C2CE8180D3F136DC3C8D6DDDDBD2662A4BF
0177DDD5C60E9A808DB4626AB3161794E08DEF74
01A53BAC150E5727F12E96BE5AAB782CDEF36713
01BFAD430CFA034B039AC9ACC98098EB53A1A703
01FE45376349628ED402D8D74868E463F9047C30

Filenames

api-ms-win-crt-io-l1-1-0.dll
libcr64.dll
C:\Windows\TEMP\%RANDOM%\%RANDOM_GUID%.dll

Mutex name and RC4 key

“3,141592653589793238462643”

YouTube URLs with mining proxy configuration data

IP addresses of mining proxies

- 3.16.150[.]123
- 3.16.152[.]201
- 3.16.152[.]64
- 3.16.167[.]92
- 3.16.30[.]155
- 3.16.31[.]23
- 3.17.167[.]43
- 3.17.23[.]144
- 3.17.25[.]11
- 3.17.59[.]6
- 3.17.61[.]161
- 3.18.108[.]152
- 3.18.223[.]195
- 13.58.182[.]92
- 13.58.22[.]81
- 13.58.77[.]225
- 13.59.31[.]61
- 18.188.122[.]218
- 18.188.126[.]190
- 18.188.249[.]210
- 18.188.47[.]132
- 18.188.93[.]252
- 18.191.104[.]117
- 18.191.173[.]48
- 18.191.216[.]242
- 18.191.230[.]253
- 18.191.241[.]159
- 18.191.47[.]76
- 18.216.127[.]143
- 18.216.37[.]78
- 18.216.55[.]205
- 18.216.71[.]102
- 18.217.146[.]44
- 18.217.177[.]214
- 18.218.20[.]166
- 18.220.29[.]72
- 18.221.25[.]98
- 18.221.46[.]136
- 18.222.10[.]104
- 18.222.187[.]174
- 18.222.198[.]38
- 18.222.213[.]203
- 18.222.253[.]209
- 18.222.56[.]98
- 18.223.111[.]224
- 18.223.112[.]155
- 18.223.131[.]52
- 18.223.136[.]87
- 18.225.31[.]210
- 18.225.32[.]44
- 18.225.7[.]128
- 18.225.8[.]249
- 52.14.103[.]72
- 52.14.221[.]47
- 52.15.184[.]25
- 52.15.222[.]174

MITRE ATT&CK techniques

Tactic	ID	Name	Description
Execution	T1085	Rundll32	The module can be executed by rundll32.exe.
T1035	Service Execution	The malware can be executed as a service.	
Defense Evasion	T1140	Deobfuscate/Decode Files or Information	The module deobfuscates strings in its code during the execution process.
T1027	Obfuscated Files or Information	The module obfuscates its code and strings in an apparent attempt to make analysis and detection difficult.	
T1102	Web Service	The malware acquires configuration data from description of YouTube videos.	
Discovery	T1063	Security Software Discovery	The malware acquires a list of running security products.
Command and Control	T1090	Connection Proxy	The module uses proxies between itself and the mining pool.
T1008	Fallback Channels	The module connects to another mining proxy if the initial one is inaccessible.	
T1095	Standard Non-Application Layer Protocol	The malware uses TCP for its communications.	
T1043	Commonly Used Port	The malware communicates over port 443.	
T1132	Data Encoding	The module encrypts then base64 encodes some network traffic.	
T1032	Standard Cryptographic Protocol	The module encrypts traffic with RC4.	
T1071	Standard Application Layer Protocol	Acquires configuration data from description of YouTube videos via HTTPS.	
Impact	T1496	Resource Hijacking	The module mines cryptocurrency.

Appendix

CRC-32 checksums checked by CoinMiner.Stantinko and the filenames they equate to are listed below.

0xB18362C7	afwserv.exe
0x05838A63	ashdisp.exe
0x36C5019C	ashwebsv.exe
0xB3C17664	aswidsagent.exe
0x648E8307	avastsvc.exe
0x281AC78F	avastui.exe
0xAA0D8BF4	avgcsrva.exe
0x71B621D6	avgcsrvx.exe
0x7D6D668A	avgfws.exe
0x1EF12475	avgidsagent.exe
0x010B6C80	avgmfapx.exe
0x6E691216	avgnsa.exe
0xB5D2B834	avgnsx.exe
0x36602D00	avgnt.exe

0x222EBF57	avgrsa.exe
0xF9951575	avgrsx.exe
0x2377F90C	avgsvc.exe
0x37FAB74F	avgsvca.exe
0xEC411D6D	avgsvcx.exe
0x0BED9FA2	avgtray.exe
0x168022D0	avguard.exe
0x99BA6EAA	avgui.exe
0x7A77BA28	avguix.exe
0x0D22F74A	avgwdsvc.exe
0x98313E09	avira.servicehost.exe
0x507E7C15	avira.systray.exe
0xFF934F08	avp.exe
0x9AC5F806	avpui.exe
0xBD07F203	avshadow.exe
0x64FDC22A	avwebg7.exe
0x0BC69161	avwebgrd.exe
0xBACF2EAC	cureit.exe
0x8FDEA9A9	drwagntd.exe
0xE1856E76	drwagnui.exe
0xF9BF908E	drwcsd.exe
0xC84AB1DA	drwebcom.exe
0x183AA5AC	drwebupw.exe
0xAC255C5E	drwupsrv.exe
0x23B9BE14	dwantispam.exe
0xDAC9F2B7	dwarkdaemon.exe
0x7400E3CB	dwengine.exe
0x73982213	dwnetfilter.exe
0x1C6830BC	dwscanner.exe
0x86D81873	dwservice.exe
0xB1D6E120	dwwatcher.exe
0xD56C1E6F	egui.exe
0x69DD7DB4	ekrn.exe
0xFB1C0526	guardgui.exe
0x5BC1D859	ipmgui.exe
0x07711AAE	ksde.exe
0x479CB9C4	ksdeui.exe
0x6B026A91	nod32cc.exe
0xCFFC2DBB	nod32krn.exe
0x59B8DF4D	nod32kui.exe

0x998B5896	procexp.exe
0xF3EEEFA8	procexp64.exe
0x81C16803	sched.exe
0x31F6B864	spideragent.exe
0x822C2BA2	taskmgr.exe
0x092E6ADA	updrgui.exe
0x09375DFF	wsctool.exe

26 Nov 2019 - 11:30AM

Sign up to receive an email update whenever a new article is published in our Ukraine Crisis – Digital Security Resource Center

Newsletter

Discussion