The 2019 Resurgence of Smokeloader

research.checkpoint.com/2019-resurgence-of-smokeloader/

July 9, 2019

July 9, 2019

A View into New Nasty Tricks and Actor Activity

Research By: Israel Gubi

Background

Smokeloader is a popular bot and a veteran in its field – being sold on underground cybercriminal markets since 2011, this piece of malware is used mainly for loading other malicious software, usually obtained from a third party. At the same time, it has the capability of loading its own modules, allowing it to conduct a variety of actions without the usage of external components. The seller of Smokeloader (which is known by the handle SmokeLdr) is active in providing this malware as a service to this date, and from what we can tell, restricts access to it to only Russian speaking users.

On the same note, we can tell that the author of Smokeloader has kept changing the malware throughout the years, and added multiple novel features to it. As an example, it was the only malware to incorporate the Propagate DLL injection method at the time it was released as a PoC by researchers.

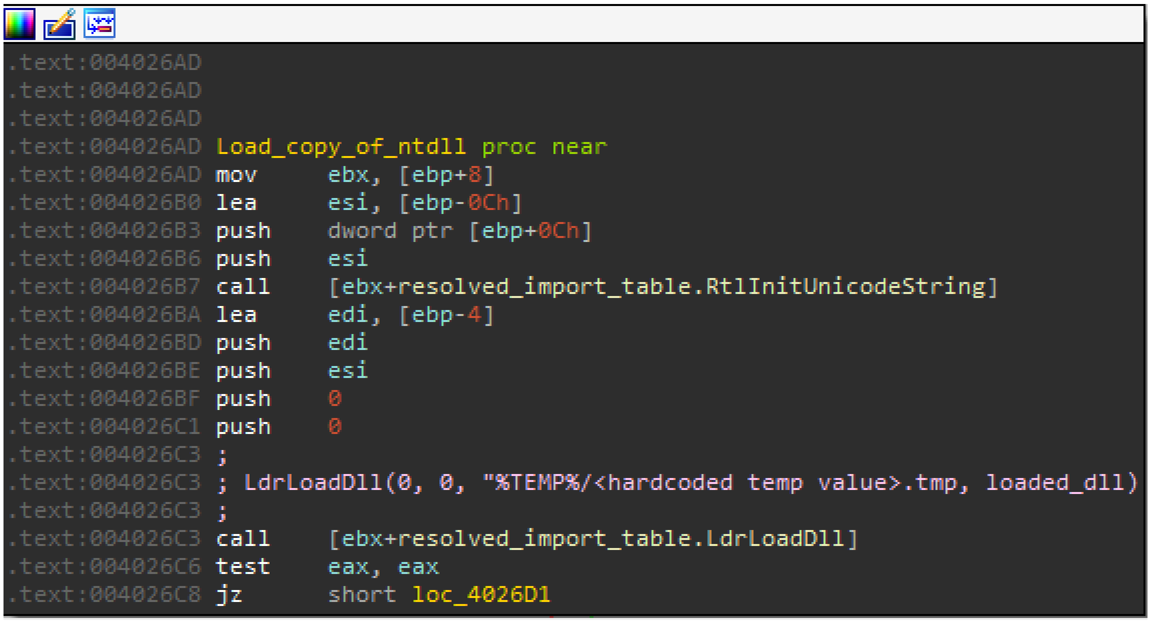
As a part of this constant renovation, we were able to spot another new version of the malware a couple of weeks ago. This version employs new tricks for deception and self-protection, which we will outline in the upcoming sections. Additionally, we will give some insight into the activity of one of the actors that makes use of this recent version and shed light on the campaigns it was involved in.

Updates from 2018’s Version

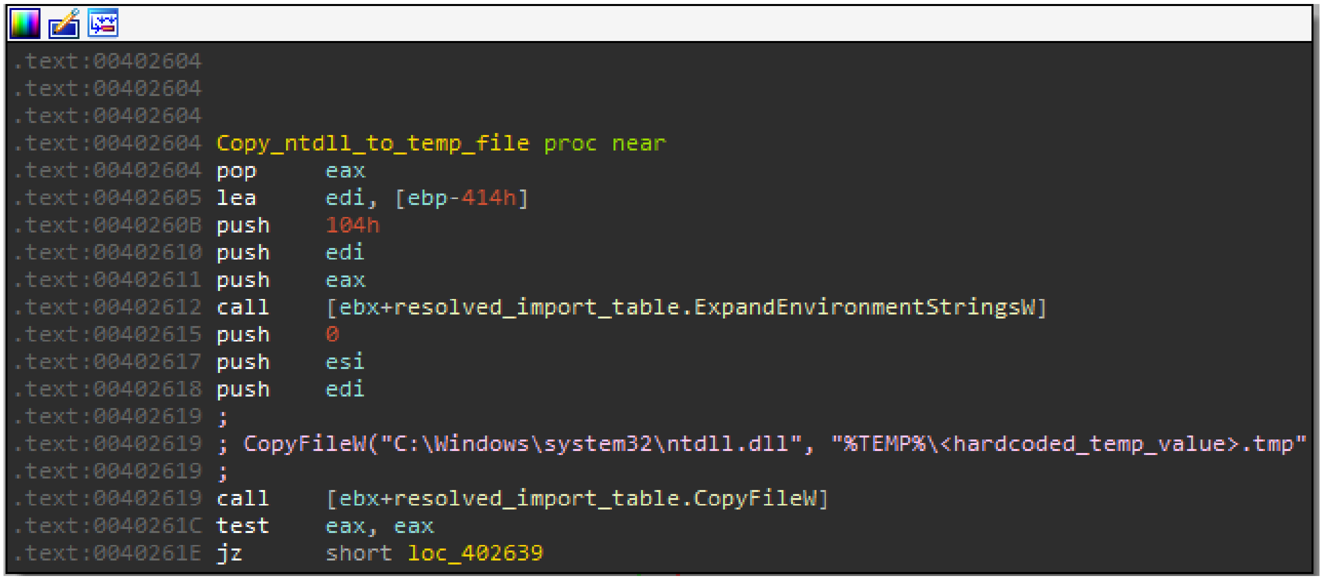
New anti-hooking and anti-VM methods

Sandboxes and other security solutions frequently use user-land hooking of ntdll functions, so as to trace all of the system calls invoked by an inspected sample (Cuckoo sandbox is just one example that employs this technique). One of the main goals of a generic malware loader is to remain undetected by such products, and evade this type of monitoring.

In order to do so, Smokeloader first creates a new copy of ntdll.dll as a temporary file with a hardcoded name in the %APPDATA%\Local\Temp directory and then loads it using LdrLoadDll. Following this, it resolves all the functions it requires for its own usage and invokes them from the new copy of ntdll in its memory.



```
.text:004026AD
.text:004026AD
.text:004026AD
.text:004026AD Load_copy_of_ntdll proc near
.text:004026AD mov     ebx, [ebp+8]
.text:004026B0 lea     esi, [ebp-0Ch]
.text:004026B3 push    dword ptr [ebp+0Ch]
.text:004026B6 push    esi
.text:004026B7 call    [ebx+resolved_import_table.RtlInitUnicodeString]
.text:004026BA lea     edi, [ebp-4]
.text:004026BD push    edi
.text:004026BE push    esi
.text:004026BF push    0
.text:004026C1 push    0
.text:004026C3 ;
.text:004026C3 ; LdrLoadDll(0, 0, "%TEMP%/<hardcoded temp value>.tmp, loaded_dll)
.text:004026C3 ;
.text:004026C3 call    [ebx+resolved_import_table.LdrLoadDll]
.text:004026C6 test    eax, eax
.text:004026C8 jz      short loc_4026D1
```



```
.text:00402604
.text:00402604
.text:00402604
.text:00402604 Copy_ntdll_to_temp_file proc near
.text:00402604 pop     eax
.text:00402605 lea     edi, [ebp-414h]
.text:0040260B push    104h
.text:00402610 push    edi
.text:00402611 push    eax
.text:00402612 call    [ebx+resolved_import_table.ExpandEnvironmentStringsW]
.text:00402615 push    0
.text:00402617 push    esi
.text:00402618 push    edi
.text:00402619 ;
.text:00402619 ; CopyFileW("C:\Windows\system32\ntdll.dll", "%TEMP%\<hardcoded_temp_value>.tmp"
.text:00402619 ;
.text:00402619 call    [ebx+resolved_import_table.CopyFileW]
.text:0040261C test    eax, eax
.text:0040261E jz      short loc_402639
```

[Copying and loading ntdll.dll]

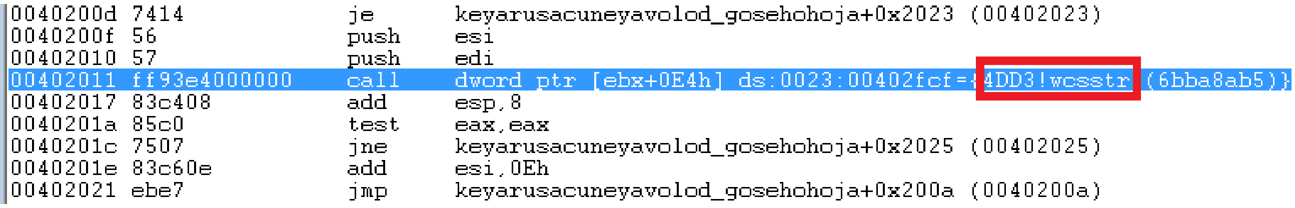
Considering that the monitoring hooks were set on the original ntdll module loaded by the operating system, invoking the functions from the memory duplicate of it will not report the behaviour of the malware to a third party security product, thus allowing Smokeloader to conduct code injection to explorer.exe that goes unnoticed. A similar evasion method was observed in usage by Hancitor, as previously outlined by MalwareBytes.

Moreover, Smokeloader conducts checks to determine if it runs in a virtual machine by reading the values of the following registry keys:

System\CurrentControlSet\Services\Disk\Enum\IDE

System\CurrentControlSet\Services\Disk\Enum\SCSI

It would use the wcsstr function from the untraced ntdll copy to find an instance of the following substrings in the values of the keys above: *qemu*, *virtio*, *vmware vbox* or *xen*, and in the presence of either one would terminate its own execution.

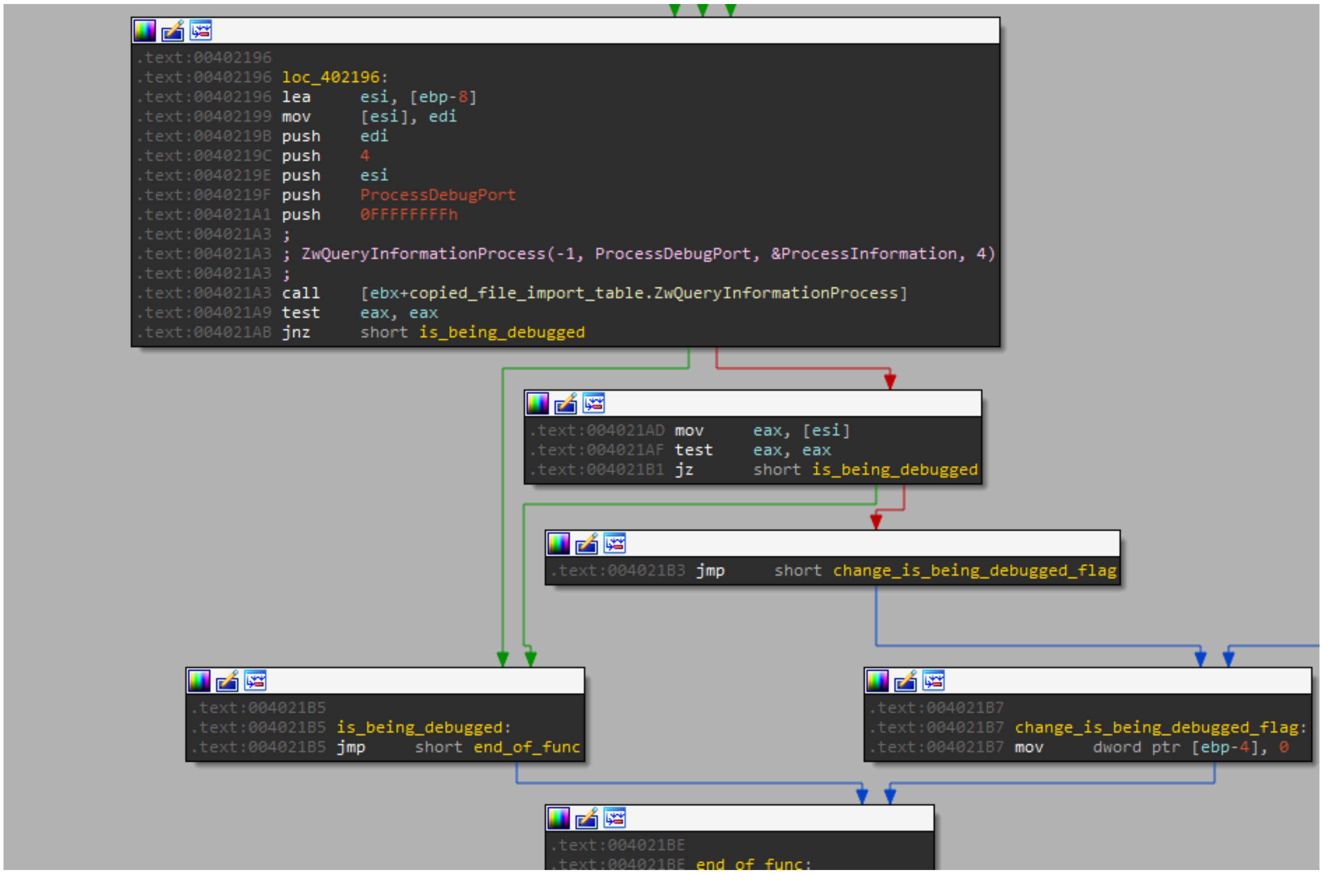


```
0040200d 7414           je      keyarusacuneyavolod_gosehohoja+0x2023 (00402023)
0040200f 56             push    esi
00402010 57             push    edi
00402011 ff93e4000000   call    dword ptr [ebx+0E4h] ds:0023:00402fcf={4DD3!wcsstr (6bba8ab5)}
00402017 83c408         add     esp,8
0040201a 85c0           test    eax,eax
0040201c 7507           jne     keyarusacuneyavolod_gosehohoja+0x2025 (00402025)
0040201e 83c60e         add     esi,0Eh
00402021 ebe7           jmp     keyarusacuneyavolod_gosehohoja+0x200a (0040200a)
```

[Calling wcsstr from the copied ntdll.dll file]

New Anti-Debug Method

In addition to the anti-debug checks used in the older version of Smokeloader, the author added another method, which is rather well known. He made the malware call the API function NtQueryInformationProcess from the copy of ntdll, with an information class argument called ProcessDebugPort. The result provided by the function indicates if the debug port is used in the malware's process, i.e. a debugger is attached to it. In the case that a non-null value is retrieved by this function, Smokeloader determines that it is indeed run by a debugger (and likely by a researcher), hence aborts its execution.



```
.text:00402196
.text:00402196 loc_402196:
.text:00402196 lea     esi, [ebp-8]
.text:00402199 mov     [esi], edi
.text:0040219B push    edi
.text:0040219C push    4
.text:0040219E push    esi
.text:0040219F push    ProcessDebugPort
.text:004021A1 push    0FFFFFFFFh
.text:004021A3 ;
.text:004021A3 ; ZwQueryInformationProcess(-1, ProcessDebugPort, &ProcessInformation, 4)
.text:004021A3 ;
.text:004021A3 call    [ebx+copied_file_import_table.ZwQueryInformationProcess]
.text:004021A9 test    eax, eax
.text:004021AB jnz     short is_being_debugged
```

```
.text:004021AD mov     eax, [esi]
.text:004021AF test    eax, eax
.text:004021B1 jz      short is_being_debugged
```

```
.text:004021B3 jmp     short change_is_being_debugged_flag
```

```
.text:004021B5
.text:004021B5 is_being_debugged:
.text:004021B5 jmp     short end_of_func
```

```
.text:004021B7
.text:004021B7 change_is_being_debugged_flag:
.text:004021B7 mov     dword ptr [ebp-4], 0
```

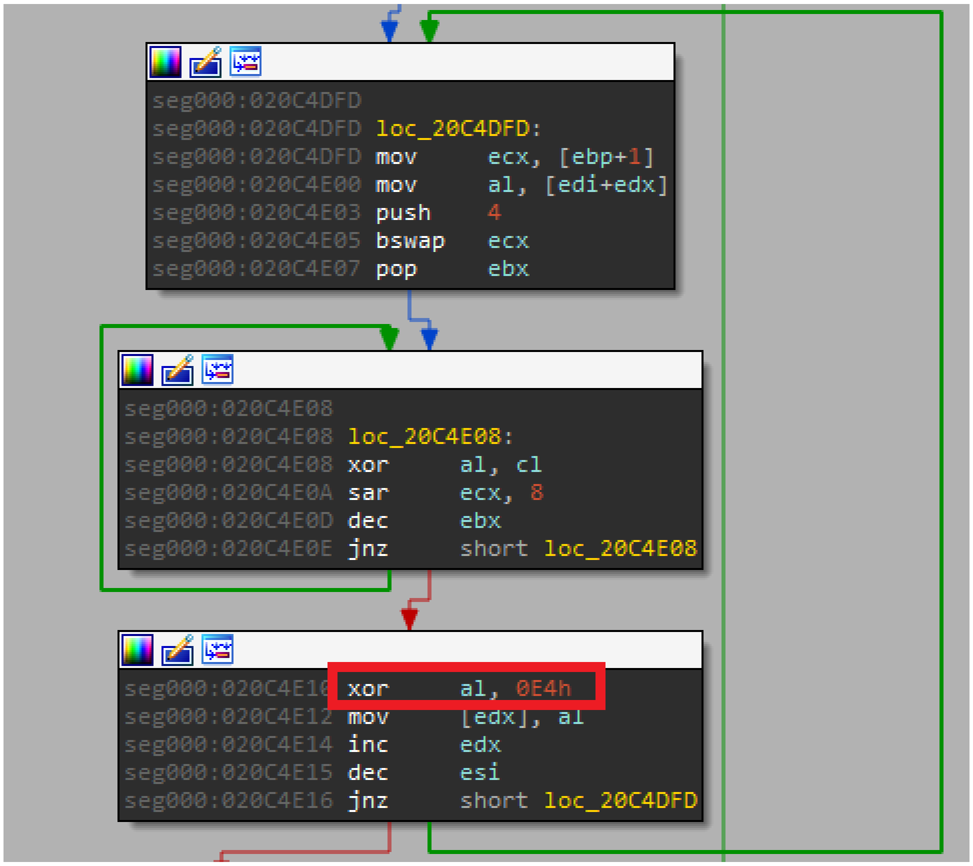
```
.text:004021BE
.text:004021BE end_of_func:
```

[NtQueryInformationProcess checking for ProcessDebugPort information]

Changed URL Decode Method

Smokeloader's C2 domains are encoded using an algorithm based on a custom sequence of arithmetic-logic operations.

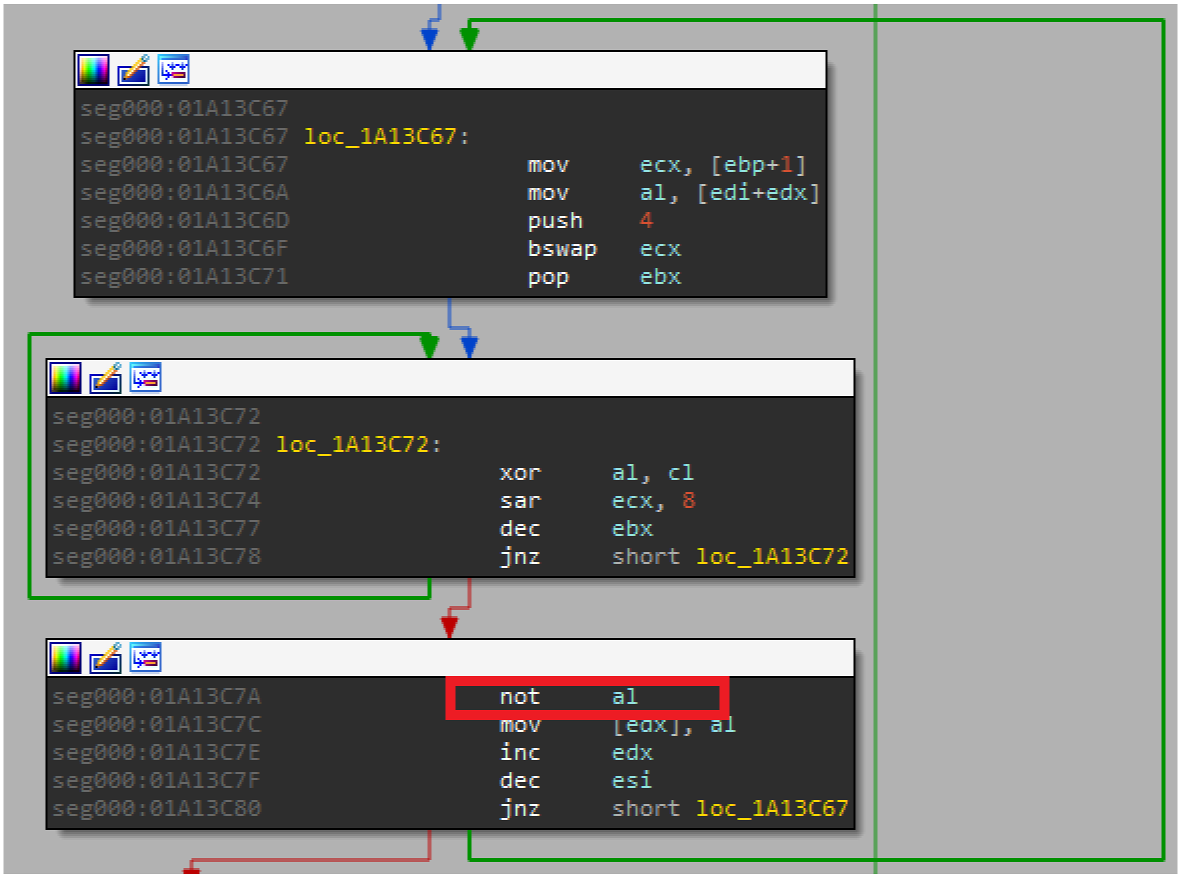
In the new version, the malware authors changed the method by modifying a single instruction in the sequence, replacing a 'not' operation with a 'xor 0xe4'. This single modification causes failure to automatic tools intended to extract the configuration of Smokeloader that relied on the old sequence for this purpose.



```
seg000:020C4DFD
seg000:020C4DFD loc_20C4DFD:
seg000:020C4DFD mov     ecx, [ebp+1]
seg000:020C4E00 mov     al, [edi+edx]
seg000:020C4E03 push    4
seg000:020C4E05 bswap   ecx
seg000:020C4E07 pop     ebx
```

```
seg000:020C4E08
seg000:020C4E08 loc_20C4E08:
seg000:020C4E08 xor     al, cl
seg000:020C4E0A sar     ecx, 8
seg000:020C4E0D dec     ebx
seg000:020C4E0E jnz     short loc_20C4E08
```

```
seg000:020C4E10 xor     al, 0E4h
seg000:020C4E12 mov     [edx], al
seg000:020C4E14 inc     edx
seg000:020C4E15 dec     esi
seg000:020C4E16 jnz     short loc_20C4DFD
```



```
seg000:01A13C67
seg000:01A13C67 loc_1A13C67:
seg000:01A13C67                 mov     ecx, [ebp+1]
seg000:01A13C6A                 mov     al, [edi+edx]
seg000:01A13C6D                 push    4
seg000:01A13C6F                 bswap   ecx
seg000:01A13C71                 pop     ebx
```

```
seg000:01A13C72
seg000:01A13C72 loc_1A13C72:
seg000:01A13C72                 xor     al, cl
seg000:01A13C74                 sar     ecx, 8
seg000:01A13C77                 dec     ebx
seg000:01A13C78                 jnz     short loc_1A13C72
```

```
seg000:01A13C7A                 not     al
seg000:01A13C7C                 mov     [edx], al
seg000:01A13C7E                 inc     edx
seg000:01A13C7F                 dec     esi
seg000:01A13C80                 jnz     short loc_1A13C67
```

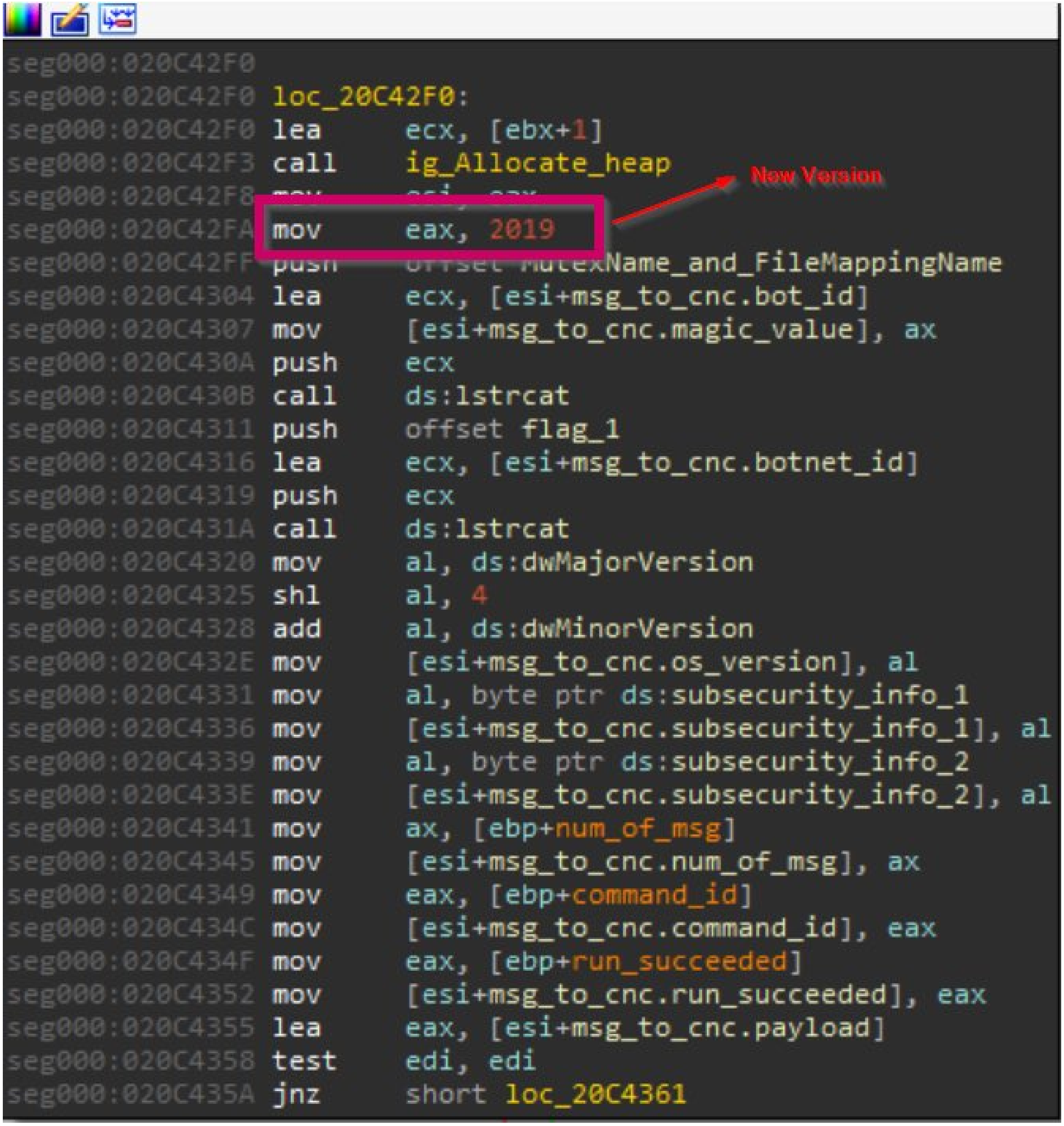
[The changed Url decryption method – by one instruction]

Changed Connection Method

Smokeloader uses a particular struct (which we'll refer to as the connection struct) for the purpose of conveying information on the victim machine to the attacker. This struct has remained mostly the same in the latest version, except for 2 changes:

1. The magic value (2 bytes at the very beginning of it that identify the start of a message sent to the C&C) has now changed to 0x7e3(2019) from 0x7e2(2018), suggesting that the latest version was released this year.

2. The malware concatenates a random-size buffer (of at least 0x1f bytes) with random data to the connection struct, which is likely done in order to make it harder to uniquely sign its communication and avoid its interception by IDS/IPS products.



```
seg000:020C42F0
seg000:020C42F0 loc_20C42F0:
seg000:020C42F0 lea     ecx, [ebx+1]
seg000:020C42F3 call    ig_Allocate_heap
seg000:020C42F8 mov     esi, eax
seg000:020C42FA mov     eax, 2019
seg000:020C42FF push    offset MutexName_and_FileMappingName
seg000:020C4304 lea     ecx, [esi+msg_to_cnc.bot_id]
seg000:020C4307 mov     [esi+msg_to_cnc.magic_value], ax
seg000:020C430A push    ecx
seg000:020C430B call    ds:lstrcat
seg000:020C4311 push    offset flag_1
seg000:020C4316 lea     ecx, [esi+msg_to_cnc.botnet_id]
seg000:020C4319 push    ecx
seg000:020C431A call    ds:lstrcat
seg000:020C4320 mov     al, ds:dwMajorVersion
seg000:020C4325 shl     al, 4
seg000:020C4328 add     al, ds:dwMinorVersion
seg000:020C432E mov     [esi+msg_to_cnc.os_version], al
seg000:020C4331 mov     al, byte ptr ds:subsecurity_info_1
seg000:020C4336 mov     [esi+msg_to_cnc.subsecurity_info_1], al
seg000:020C4339 mov     al, byte ptr ds:subsecurity_info_2
seg000:020C433E mov     [esi+msg_to_cnc.subsecurity_info_2], al
seg000:020C4341 mov     ax, [ebp+num_of_msg]
seg000:020C4345 mov     [esi+msg_to_cnc.num_of_msg], ax
seg000:020C4349 mov     eax, [ebp+command_id]
seg000:020C434C mov     [esi+msg_to_cnc.command_id], eax
seg000:020C434F mov     eax, [ebp+run_succeeded]
seg000:020C4352 mov     [esi+msg_to_cnc.run_succeeded], eax
seg000:020C4355 lea     eax, [esi+msg_to_cnc.payload]
seg000:020C4358 test    edi, edi
seg000:020C435A jnz     short loc_20C4361
```

[Changed magic value in the new version of Smokeloader]

New Persistence Methods

As part of Smokeloader’s behaviour, it generates a unique ID for each victim machine, which is based on concatenation of the computer name, a hard coded static number (that differs between campaigns) and the volume serial number of the system drive. The ID is then generated as an MD5 hash of the concatenated string and appended again with the MD5 of the volume serial number.

The malware uses this unique ID for several purposes, namely creating random file names for 2 dropped files – the first is a copy of Smokeloader's executable, and the second is an lnk which is invoked as a scheduled task. The latter is used just to run the former, thus allowing the malware to persist on the machine after reboot using this pair of files.

In older versions, the random name of the copied malware executable was based on the last eight characters of the ID described above. Those were all dependent on the volume serial number and would create the same file artifact for a single machine. In the new version, however, the name is generated from seven letters starting from the 30th letter, allowing it to also depend on the hardcoded static value. As a result, samples with different hardcoded values will generate different file names on the same machine, allowing the malware to be less detectable by AV products.

In addition to the above, a few other modifications to persistence mechanisms were witnessed in the new version. One of them is the creation of the aforementioned lnk file in the *%startup%* folder with the name "***Opera Scheduled Autoupdate <random_number>***". The name of the scheduled task that invokes it is "***NvNgxUpdateCheckDaily_{%08X-%04X-%04X-%04X-%08X%04X}***" (the hex values are hardcoded in the binary) and the task executed via the lnk runs the following script embedded in it:

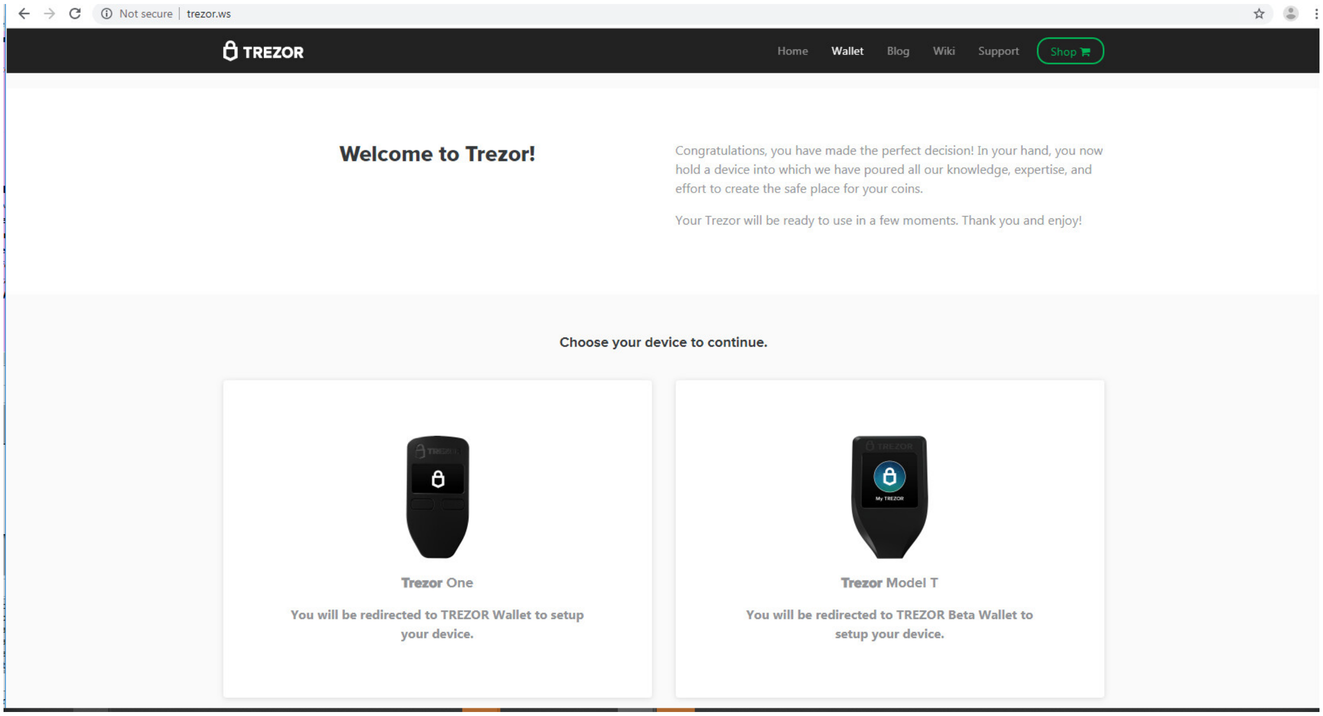
"<?xml version="1.0"?><scriptlet><registration classid="{00000000-0000-0000-0000-00000000%04X}"><script language="jscript"><![CDATA[GetObject("winmgmts:Win32_Process").Create("%ls",null,null,null);]]></script></registration></scriptlet>"

Apart from this, there is a change in the way scheduled task is created, which is now registered only when the explorer.exe process receives a ***WM_QUERYENDSESSION*** or ***WM_ENDSESSION*** window message. This allows the malware to conduct task scheduling only when the computer is turned off (which is when the window messages are received). Once again, this technique provides Smokeloader with the ability to evade AV solutions and remain under the radar. To better understand how this works you may reference the explanation given here.

The Actor Leveraging the New Smokeloader Variant

The sample we analyzed is utilized in part by an actor that is using Smokeloader for a long time.

The payload provided by our sample is using Smokloader's FakeDNS and DDoS plugins to attack trezor.io (the site of a cryptocurrency hardware wallet product). The former causes the redirection of the site on a victim's host to the IP ***31.210.170[.]195***, which seems to look like a fake website mimicking the original trezor.io.



Welcome to Trezor!

Congratulations, you have made the perfect decision! In your hand, you now hold a device into which we have poured all our knowledge, expertise, and effort to create the safe place for your coins.

Your Trezor will be ready to use in a few moments. Thank you and enjoy!

Choose your device to continue.

Trezor One

You will be redirected to TREZOR Wallet to setup your device.

Trezor Model T

You will be redirected to TREZOR Beta Wallet to setup your device.

[Trezor fake website main page]

Other than that, the new variant of Smokeloader is downloading one more malware from the url: ***fileboard[.]live/upd.exe***.

This downloaded payload (D83F3025BA5B41775423A456BC4C19EF) turns out to be the Azorult infostealer, which in turn communicates to a URL under the same domain – ***fileboard[.]live/index.php***.

The campaign described above is connected to an actor we previously witnessed using Smokeloader, which was involved in several notorious campaigns. Those included a mass campaign spreading Amadey Loader (8b1b2dee404f274e90bd87ff6983d2162abee16c4d9868a10b802bd9bcbdbec6), the AveMaria info stealer (88c47899f49dd25e5799fdcf892b990320c645475b612ac5324e635e2acf89dd) and most interestingly ServHelper – a backdoor vastly used by TA505 (20dd61fae49972323bb9c38a46ca4c93). The latter may suggest that in reality, the actor using this new variant is in fact TA505.

Our attribution to this actor is based on three clues that we were able to obtain from investigating the current campaign:

1. Usage of a similar format for C2 domain names – (e.g. protest-0124.tk vs. protest-01242505.tk in former activities)
2. Usage of the same RC4 keys for encrypting communication and decrypting headers – these keys are 0xaf03e678 and 0x78821544.

3. No presence of an advertisement for the new version of the malware in the underground forums in which it is sold. We believe this may indicate that so far the seller is distributing the new variant among known buyers so as to test and evaluate its quality, before another stable release.

We will keep monitoring Smokeloader's development and threat actor activity and intend to update on any new variants of the malware as soon as they emerge in the wild.

Check Point protects against all variants of Smokeloader, both of previous versions and the one described in this publication.

The relevant protections carry the names Smokeloader.TC.* and Trojan-Downloader.Win32.Smokeloader.TC.*

IOCs

MD5s

5FC6F24D43BC7CA45A81D159291955D1 – New Smokeloader variant
20DD61FAE49972323BB9C38A46CA4C93 – ServHelper
E7680155F86AEAC74B65DA38143F7E9F – Ave Maria Info Stealer
AF93FD5C7810669D125EC9B0D6E28509 – Amadey Loader

Smokeloader C2s:

hxxp://protest-01242505[.]tk/
hxxp://test-service012505[.]ru.com/
hxxp://test-service012505[.]pw/
hxxp://test-service012505[.]com/
hxxp://test-service012505[.]site/
hxxp://test-service012505[.]store/
hxxp://test-service01242505[.]ru/
hxxp://mytest-service012505[.]ru/
hxxp://test-service012505[.]su/
hxxp://test-service012505[.]info/
hxxp://test-service012505[.]net/
hxxp://test-service012505[.]tech/
hxxp://test-service012505[.]online/
hxxp://rutest-service012505[.]ru/
hxxp://test-service01dom2505[.]ru/
hxxp://test-service012505[.]website/
hxxp://test-service012505[.]xyz/
hxxp://test-service01pro2505[.]ru/
hxxp://test-service01rus2505[.]ru/
hxxp://test-service012505[.]eu/

hxxp://test-service012505[.]press/
hxxp://protest-service012505[.]ru/
hxxp://rustest-service012505[.]ru/
hxxp://test-service012505[.]net2505[.]ru/
hxxp://test-service012505[.]space/
hxxp://domtest-service012505[.]ru/
hxxp://mirtest-service012505[.]ru/
hxxp://test-service012505[.]org2505[.]ru/
hxxp://test-service012505[.]pp2505[.]ru/
hxxp://test-service012505[.]pro/
hxxp://test-service012505[.]host/
hxxp://test-service012505[.]fun/
hxxp://mostest-service012505[.]ru/
hxxp://toptest-service012505[.]ru/
hxxp://alltest-service012505[.]ru/
hxxp://vsetest-service012505[.]ru/
hxxp://newtest-service012505[.]ru/
hxxp://biotest-service012505[.]ru/
hxxp://test-service01shop2505[.]ru/
hxxp://test-service01info2505[.]ru/
hxxp://test-service01plus2505[.]ru/
hxxp://test-service01club2505[.]ru/
hxxp://test-service01torg2505[.]ru/
hxxp://test-service01land2505[.]ru/
hxxp://test-service01life2505[.]ru/
hxxp://test-service01blog2505[.]ru/
hxxp://megatest-service012505[.]ru/
hxxp://infotest-service012505[.]ru/
hxxp://besttest-service012505[.]ru/
hxxp://shoptest-service012505[.]ru/
hxxp://kupitest-service012505[.]ru/
hxxp://proftest-service012505[.]ru/
hxxp://clubtest-service012505[.]ru/
hxxp://mytest-service01242505[.]ru/
hxxp://rutest-service01242505[.]ru/
hxxp://test-service01stroy2505[.]ru/
hxxp://test-service01forum2505[.]ru/
hxxp://supertest-service012505[.]ru/
hxxp://protest-service01242505[.]ru/
hxxp://protest-01252505[.]ml/
hxxp://protest-01262505[.]ga/
hxxp://protest-01272505[.]cf/

hxxp://protest-01282505[.]gq/
hxxp://protest-01292505[.]com/
hxxp://protest-01302505[.]net/
hxxp://protest-01312505[.]org/
hxxp://protest-01322505[.]biz/
hxxp://protest-01332505[.]info/
hxxp://protest-01342505[.]eu/
hxxp://protest-01352505[.]nl/
hxxp://protest-01362505[.]mobi/
hxxp://protest-01372505[.]name/
hxxp://protest-01382505[.]me/
hxxp://protest-01392505[.]garden/
hxxp://protest-01402505[.]art/
hxxp://protest-01412505[.]band/
hxxp://protest-01422505[.]bargains/
hxxp://protest-01432505[.]bet/
hxxp://protest-01442505[.]blue/
hxxp://protest-01452505[.]business/
hxxp://protest-01462505[.]casa/
hxxp://protest-01472505[.]city/
hxxp://protest-01482505[.]click/
hxxp://protest-01492505[.]company/
hxxp://protest-01502505[.]futbol/
hxxp://protest-01512505[.]gallery/
hxxp://protest-01522505[.]game/
hxxp://protest-01532505[.]games/
hxxp://protest-01542505[.]graphics/
hxxp://protest-01552505[.]group/
hxxp://protest-02252505[.]ml/
hxxp://protest-02262505[.]ga/
hxxp://protest-02272505[.]cf/
hxxp://protest-02282505[.]gq/
hxxp://protest-03252505[.]ml/
hxxp://protest-03262505[.]ga/
hxxp://protest-03272505[.]cf/
hxxp://protest-03282505[.]gq/
hxxp://protest-05242505[.]tk/
hxxp://protest-06242505[.]tk/

Trezor fake website:
hxxp://31.210.170[.]195

AZORult IOCs:

hxxp://fileboard[.]live/index.php
hxxp://fileboard[.]live/upd.exe

Smokeloader DropZones:

hxxp://vinomag.pw/nsis.exe
hxxp://mypromo.online/parapara.exe
hxxps://babolgum.icu/cobal.exe

Amadey IOCs:

skcalladhellormi.xyz

Ave Maria IOCs:

hxxps://paste.ee/r/2zmfq/0

ServHelper IOCs:

hxxp://esupdate.icu/js/s.php