Take a “NetWalk” on the Wild Side

mcafee.com/blogs/other-blogs/mcafee-labs/take-a-netwalk-on-the-wild-side/

August 3, 2020

Executive Summary

The NetWalker ransomware, initially known as Mailto, was first detected in August 2019. Since then, new variants were discovered throughout 2019 and the beginning of 2020, with a strong uptick noticed in March of this year.

NetWalker has noticeably evolved to a more stable and robust ransomware-as-a-service (RaaS) model, and our research suggests that the malware operators are targeting and attracting a broader range of technically advanced and enterprising criminal affiliates.

McAfee Advanced Threat Research (ATR) discovered a large sum of bitcoins linked to NetWalker which suggest its extortion efforts are effective and that many victims have had no option other than to succumb to its criminal demands.

We approached our investigation of NetWalker with some possible ideas about the threat actor behind it, only to later disprove our own hypothesis. We believe the inclusion of our thinking, and the means with which we debunked our own theory, highlight the importance of thorough research and we welcome further discussion on this topic. We believe it starts valuable discussions and helps avoid duplicate research efforts by others. We also encourage our peers in the industry to share information with us in case you have more evidence.

McAfee protects its customers against the malware covered in this blog in all its products, including personal antivirus, endpoint and gateway. To learn more about how McAfee products can defend against these types of attacks, visit our blog on Building Adaptable Security Architecture Against NetWalker.

Check out McAfee Insights to stay on top of NetWalker's latest developments and intelligence on other cyber threats, all curated by the McAfee ATR team. Not only that, Insights will also help you prioritize threats, predict if your countermeasures will work and prescribe corrective actions.

Introduction

Since 2019, NetWalker ransomware has reached a vast number of different targets, mostly based in western European countries and the US. Since the end of 2019, the NetWalker gang has indicated a preference for larger organisations rather than individuals. During the COVID-19 pandemic, the adversaries behind NetWalker clearly stated that hospitals will not be targeted; whether they keep to their word remains to be seen.

The ransomware appends a random extension to infected files and uses Salsa20 encryption. It uses some tricks to avoid detection, such as a new defence evasion technique, known as reflective DLL loading, to inject a DLL from memory.

The NetWalker collective, much like those behind Maze, REvil and other ransomware, threatens to publish victims' data if ransoms are not paid.

As mentioned earlier, NetWalker RaaS prioritizes quality over quantity and is looking for people who are Russian-speaking and have experience with large networks. People who already have a foothold in a potential victim's network and can exfiltrate data with ease are especially sought after. This is not surprising, considering that publishing a victims' data is part of NetWalker's model.

The following sections are dedicated to introducing the NetWalker malware and displaying the telemetry status before moving on to the technical malware analysis of the ransomware's behaviour. We will explain how the decryptor works and show some interactions between NetWalker's operators and their victims. After this, we discuss the changes in modus operandi since September 2019, especially regarding payment behaviour. Then we show our

attempts, unfruitful as they were, at discovering a link between NetWalker and previous, seemingly unrelated ransomware variants. Finally, we deliver an overview of IOCs related to NetWalker and its MITRE ATT&CK techniques.

Telemetry

Using McAfee's billion Insights sensors, we can show the global prevalence of the NetWalker ransomware.

Figure 1. McAfee MVISION Insights shows global prevalence of the NetWalker ransomware

Technical Analysis

Ransom note (pre-March 2020)

Before March 2020, the NetWalker ransom note indicated how to contact the adversary directly using anonymous email account services with random names (such as kkeessnnkkaa@cock.li and hhaaxxhhaaxx@tuta.io):

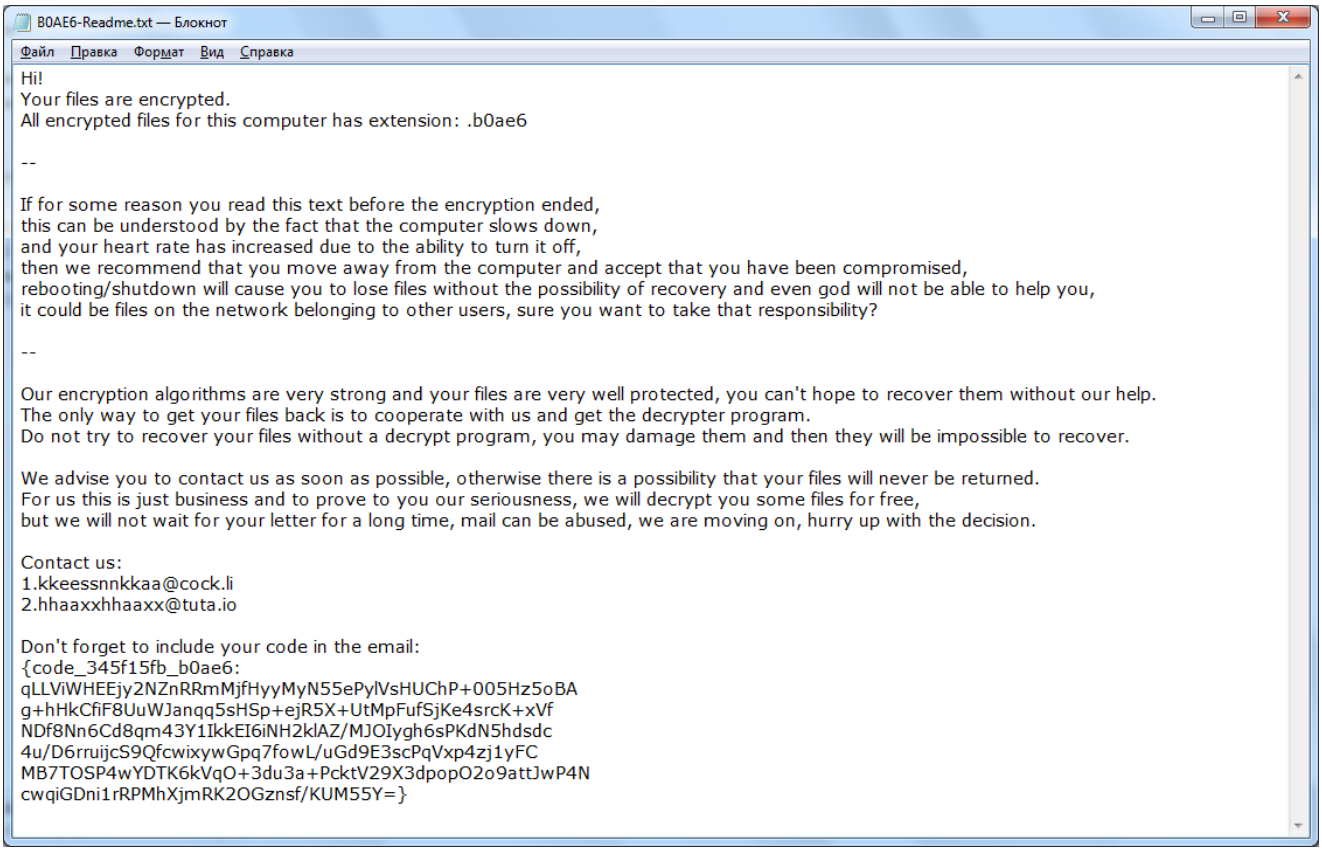
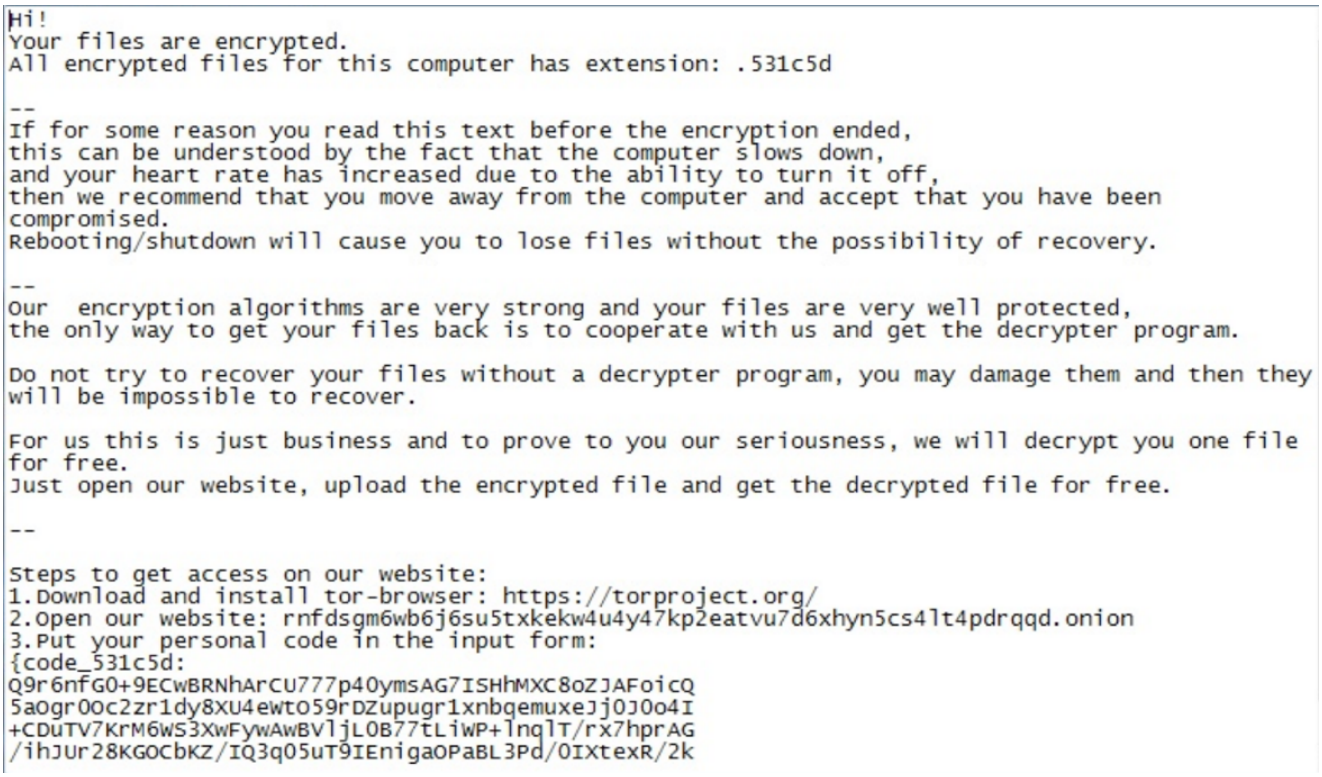


Figure 2. Example of ransom note prior to March 2020

Ransom Note (Post-March 2020)

On 12 March 2020, a researcher shared a screenshot of a new NetWalker ransom note in a tweet and we can see that the attackers have changed the contact method significantly. Email communication has been dropped completely with victims now required to make contact through the NetWalker Tor interface where, after submitting their user key, they will then be redirected to a chat with NetWalker technical support. This change in contact method coincides with underground forum postings where NetWalker revealed it was opening its RaaS up for new affiliates. The Tor page was not the only noticeable change we will highlight in this blog.



```
Hi!
Your files are encrypted.
All encrypted files for this computer has extension: .531c5d

--
If for some reason you read this text before the encryption ended,
this can be understood by the fact that the computer slows down,
and your heart rate has increased due to the ability to turn it off,
then we recommend that you move away from the computer and accept that you have been
compromised.
Rebooting/shutdown will cause you to lose files without the possibility of recovery.

--
Our  encryption algorithms are very strong and your files are very well protected,
the only way to get your files back is to cooperate with us and get the decrypter program.

Do not try to recover your files without a decrypter program, you may damage them and then they
will be impossible to recover.

For us this is just business and to prove to you our seriousness, we will decrypt you one file
for free.
Just open our website, upload the encrypted file and get the decrypted file for free.

--

Steps to get access on our website:
1.Download and install tor-browser: https://torproject.org/
2.Open our website: rnfdsgm6wb6j6su5txkekw4u4y47kp2eatvu7d6xhyn5cs4lt4pdrqqd.onion
3.Put your personal code in the input form:
{code_531c5d:
Q9r6nfGO+9ECwBRNhArCU777p4OymsAG7ISHhMXC8oZJAFoicQ
5aOgrOOc2zr1dy8XU4eWtO59rDZupugr1xnbqemuxeJj0JOo4I
+CDuTV7KrM6WS3XwFywAwBVljLOB77tLiWP+lnqlT/rx7hprAG
/ihJUr28KGOCbKZ/IQ3qO5uT9IEnigaOPaBL3Pd/OIXtexR/2k
```

Figure 3. Example of ransom note after March 2020

NetWalker Analysis

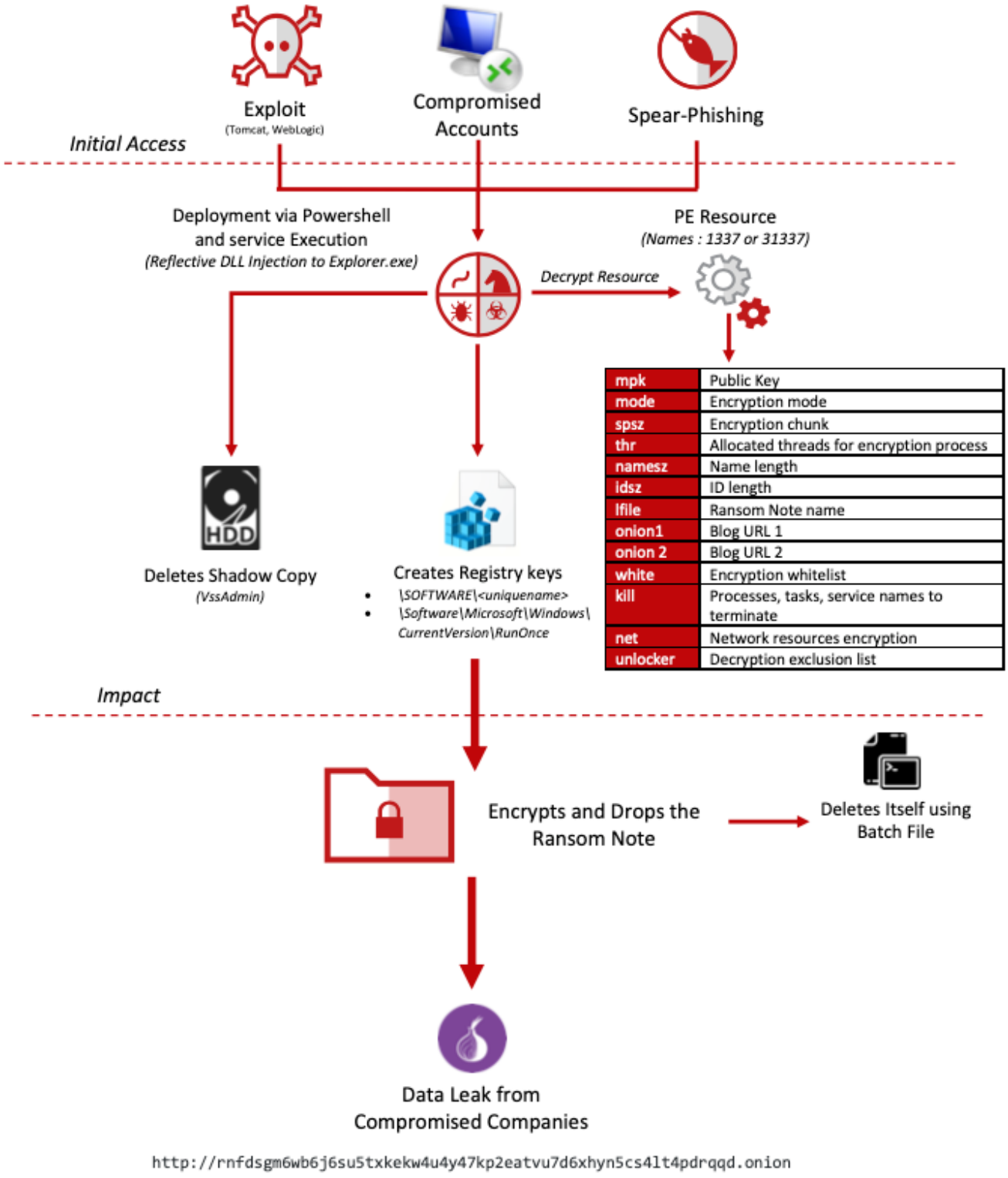


Figure 4. NetWalker behavior

NetWalker Resource Analysis (Pre-March 2020)

The NetWalker malware uses a custom resource type (1337 or 31337) containing its entire configuration. This file is extracted to memory and decrypted using the RC4 algorithm with a hard-coded key in the resource.

Before 12 March 2020, NetWalker used the email contact process between its support operation and the victims to proceed with payment and send the decryption program. To do this, NetWalker used its configuration file in the resource to set its encryption mode, the name of the ransom note, etc., and email contacts.

Name	wwllww.exe
Size	96256 bytes
File-Type	EXE
SHA 256	58e923ff158fb5aecd293b7a0e0d305296110b83c6e270786edcc4fea1c8404c
Compile time	6 December 2019

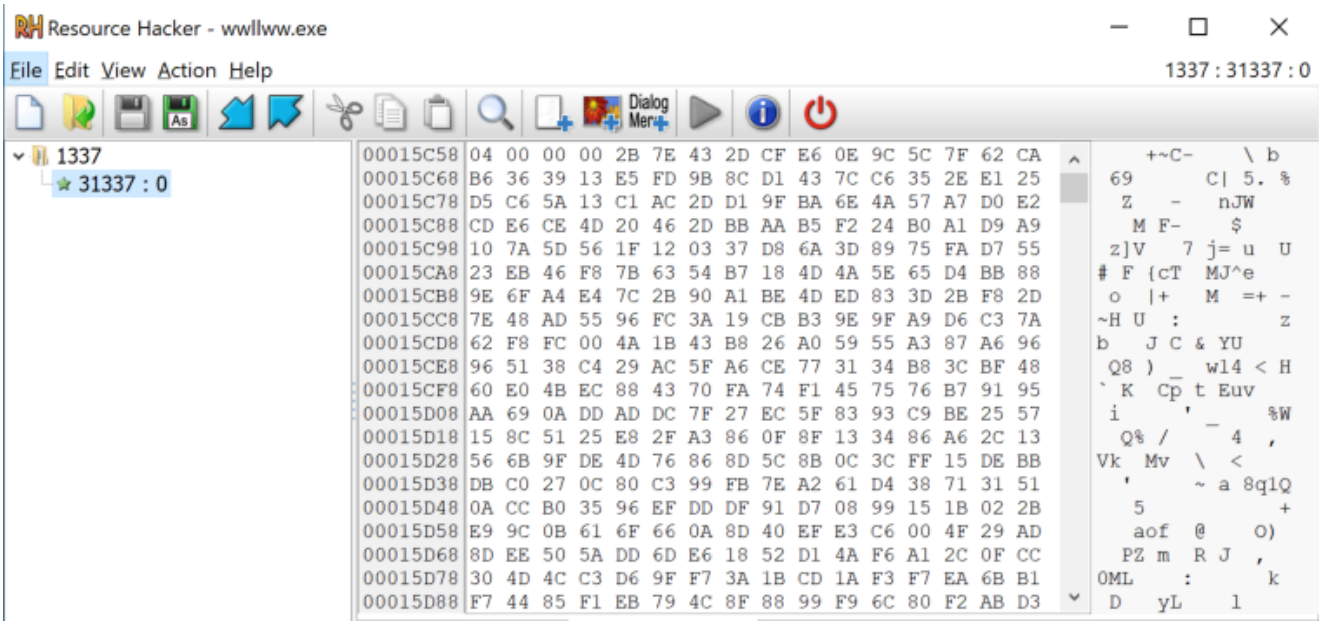


Figure 5. NetWalker resource from wwllww.exe

Once decrypted, the configuration file reveals several parameters, allowing us to understand how it works (how it constitutes the ransom note, the number of threads allocated for encryption, etc.):

mpk	Public key
mode	Encryption mode

thr	Allocated threads for encryption process
spsz	Encryption chunk
namesz	Name length
idsz	ID length
crmask	.mailto[email].{ID}
mail	Contact mail
lfile	Ransom Note name
lend	B64 encoded ransom note
white	Encryption whitelist
kill	Processes, tasks, service names to terminate
unlocker	Decryption exclusion list

NetWalker Resource Analysis (Post-March 2020)

When Netwalker changed its contact mode and switched from email to the submission of the user key directly on the web portal of the group's blog, the configuration file in the resource also changed. We found changes in the configuration file, such as the disappearance of the contact "mail" and "crmask" fields (previously set as XXX@cock.li,XXX@tuta.io, etc., and .mailto[email].{ID}). This field was replaced by "onion1" and "onion2", and these fields are set with the NetWalker blog URL/payment page (hxxp://rnfdsgm<snip>drqqd.onion/). We also noticed that the NetWalker developers complemented their "unlocker" field with some specific values (e.g. "psexec.exe, system, forti*.exe, fmon.exe*, etc").

Name	cnt.ex
Size	70656 bytes
File-Type	EXE
SHA 256	26dfa8512e892dc8397c4ccbbe10efbcf85029bc2ad7b6b6fe17d26f946a01bb
Compile time	2 May 2020

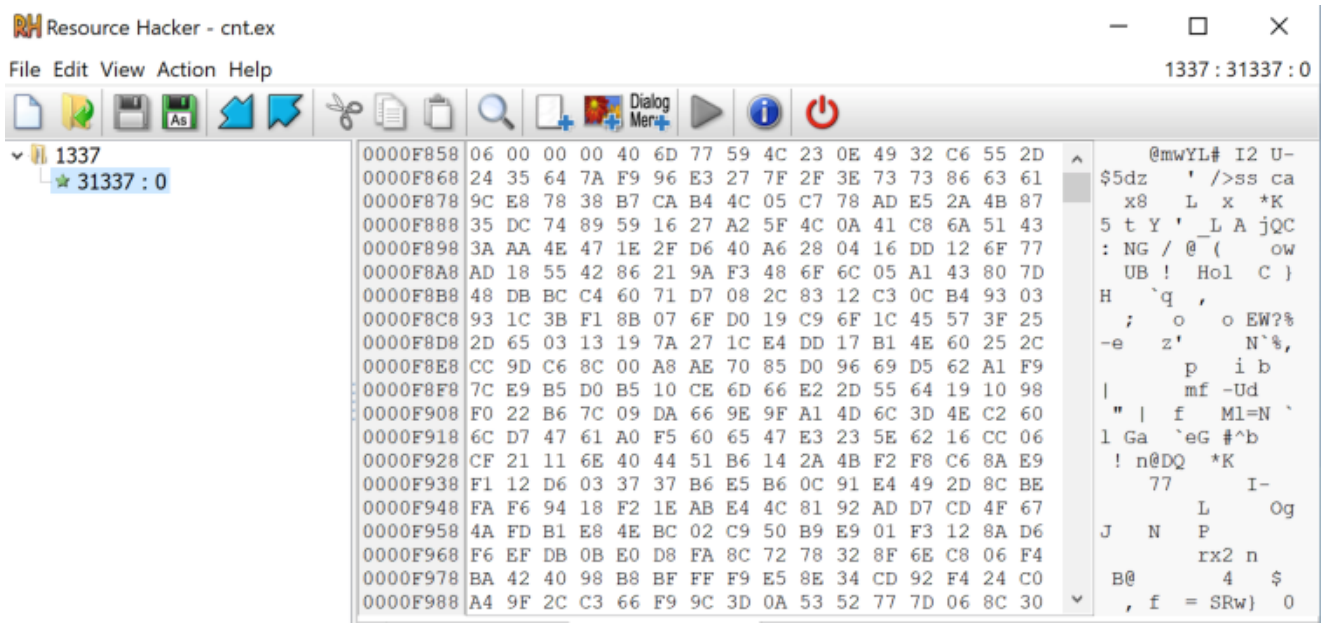


Figure 6. NetWalker resource from cnt.ex

Usually, attackers use RC_DATA or a malicious BITMAP. The latter can, for example, be a regular Bitmap (open matrix image format used by Windows) that can be used by malware to execute code or as a payload dropper. The image's pixels are an actual binary representation of the payload. This process can be summarized as Exe -> Resources -> BMP with embedded data in pixels fetched and decrypted by, e.g. a DLL -> Payload), etc. However, in this case, they use this special custom type to increase obfuscation. The NetWalker developers chose custom types by using 1337 or 31337 structs, so the resource format does not change. However, as we said, several values have changed or been replaced:

mpk	Public Key
mode	Encryption mode
spsz	Encryption chunk
thr	Allocated threads for encryption process
namesz	Name length
idsz	ID length
lfile	Ransom Note name
onion1	Blog URL 1
onion 2	Blog URL 2
white	Encryption whitelist

kill	Processes, tasks, service names to terminate
net	Network resources encryption
unlocker	Decryption exclusion list
lend	B64 encoded ransom note

NetWalker Executable Analysis (Post-March 2020)

The malware sample used for this blog post has the same information:

Name	c21ecd18f0bbb28112240013ad42dad5c01d20927791239ada5b61e1c6f5f010
Size	70656 bytes
File-Type	EXE
SHA 256	c21ecd18f0bbb28112240013ad42dad5c01d20927791239ada5b61e1c6f5f010
Compile time	2 May 2020

The unpacked malware is a binary file of 32 bits that can be found as an EXE file.

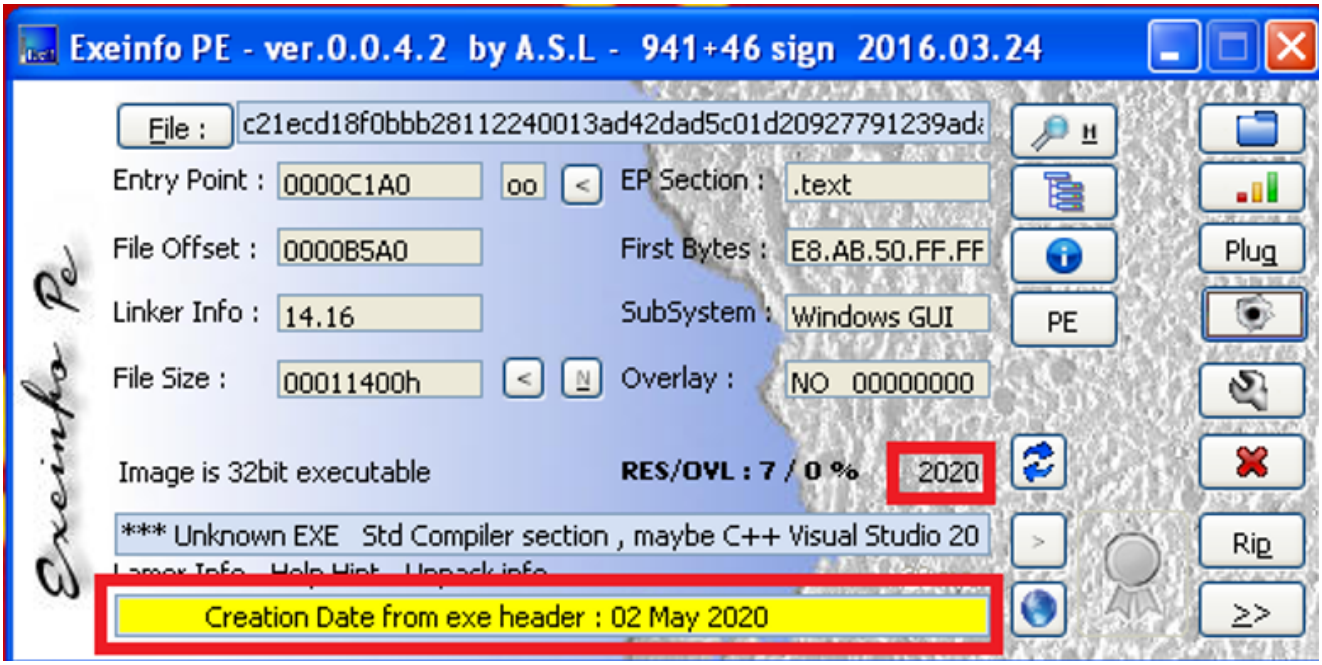


Figure 7. Information sample of the malware

The malware's first action is to combine all the required functions it needs into one large function, combining the modules already loaded in Windows with additional DLLs as described below.

Instead of searching for the function in the usual way, the malware makes a CRC32 hash of the name of each function and compares with hardcoded values. Additionally, instead of using the function "GetProcAddress", the malware uses the Process Environment Block (PEB) to make analysis harder.

```
NetwalkerGetModuleAddressByCRC32HashFromModuleNameFunction proc near
                                        ; CODE XREF: NetwalkerGetModulesByCRC32AndFunctionsFunction+10↓p
                                        ; NetwalkerGetModulesByCRC32AndFunctionsFunction+4FD↓p ...

arg_0           = dword ptr  4

                push    ebx
                push    esi
                push    edi
                call    NetwalkerGetPEBAddressFunction
                test    eax, eax
                jz      short _exit
                mov     edi, [eax+0Ch]
                add     edi, 14h
                mov     esi, [edi]
                cmp     esi, edi
                jz      short _exit
                mov     ebx, [esp+0Ch+arg_0]
                nop     dword ptr [eax+00h]

_loop_found_module:                     ; CODE XREF: NetwalkerGetModuleAddressByCRC32HashFromModuleNameFunction+3C↓j
                movzx   eax, word ptr [esi+24h]
                push    1
                shr     eax, 1
                push    eax
                push    dword ptr [esi+28h]
                call    NetwalkerCRC32PolinomialFunction
                add     esp, 0Ch
                cmp     eax, ebx
                jz      short _exit_return_module_address
                mov     esi, [esi]
                cmp     esi, edi
                jnz     short _loop_found_module

_exit:                                  ; CODE XREF: NetwalkerGetModuleAddressByCRC32HashFromModuleNameFunction+A↑j
                                        ; NetwalkerGetModuleAddressByCRC32HashFromModuleNameFunction+16↑j
                pop     edi
                pop     esi
                xor     eax, eax        ; return FALSE
                pop     ebx
                retn
; ---------------------------------------------------------------------------

_exit_return_module_address:            ; CODE XREF: NetwalkerGetModuleAddressByCRC32HashFromModuleNameFunction+36↑j
                mov     eax, [esi+10h]
                pop     edi
                pop     esi
                pop     ebx
                retn
NetwalkerGetModuleAddressByCRC32HashFromModuleNameFunction endp
```

Figure 8. Get the module accessing the PEB and using a CRC32

If the module cannot be discovered, it will load with "LdrLoadDll", a native function of Windows, to try avoiding hooks in the usual functions, e.g. "LoadLibraryW":

```
00401D56   85F6            test    esi, esi
00401D58   75 42           jnz     short <_get_advapi_functions>
00401D5A   68 1C104100     push    offset <aAdvapi32_dll>                  UNICODE "advapi32.dll"
00401D5F   8D4424 14       lea     eax, [esp+14]
00401D63   50              push    eax
00401D64   A1 64124100     mov     eax, [<NetwalkerGlobalVarMemoryAddressOfBufferReserved>]
00401D69   8B40 50         mov     eax, [eax+50]
00401D6C   FFD0            call    eax
00401D6E   8D4424 0C       lea     eax, [esp+C]
00401D72   897424 0C       mov     [esp+C], esi
00401D76   50              push    eax
00401D77   8D4424 14       lea     eax, [esp+14]
00401D7B   50              push    eax
00401D7C   A1 64124100     mov     eax, [<NetwalkerGlobalVarMemoryAddressOfBufferReserved>]
00401D81   56              push    esi
00401D82   56              push    esi
00401D83   8B40 64         mov     eax, [eax+64]
00401D86   FFD0            call    eax                                     ntdll.LdrLoadDll
00401D88   85C0            test    eax, eax
00401D8A   0F88 6E020000   js      00401FFE
00401D90   8B7424 0C       mov     esi, [esp+C]
00401D94   85F6            test    esi, esi
00401D96   0F84 62020000   je      00401FFE
```

Figure 9. Load library using LdrLoadDll

```
.text:004017A2                 push    esi
.text:004017A3                 call    NetwalkerGetFunctionFromModuleByCRC32HashFromFunctionNameFunction
.text:004017A8                 mov     ecx, NetwalkerGlobalVarMemoryAddressOfBufferReserved
.text:004017AE                 push    0CCE95612h
.text:004017B3                 push    esi
.text:004017B4                 mov     [ecx+0DCh], eax
.text:004017BA                 call    NetwalkerGetFunctionFromModuleByCRC32HashFromFunctionNameFunction
.text:004017BF                 mov     ecx, NetwalkerGlobalVarMemoryAddressOfBufferReserved
.text:004017C5                 push    95C03D0h
.text:004017CA                 push    esi
.text:004017CB                 mov     [ecx+0E0h], eax
.text:004017D1                 call    NetwalkerGetFunctionFromModuleByCRC32HashFromFunctionNameFunction
.text:004017D6                 mov     ecx, NetwalkerGlobalVarMemoryAddressOfBufferReserved
.text:004017DC                 push    0A7FB4165h
.text:004017E1                 push    esi
.text:004017E2                 mov     [ecx+0E4h], eax
.text:004017E8                 call    NetwalkerGetFunctionFromModuleByCRC32HashFromFunctionNameFunction
.text:004017ED                 mov     ecx, NetwalkerGlobalVarMemoryAddressOfBufferReserved
.text:004017F3                 push    8B5819AEh
.text:004017F8                 push    esi
.text:004017F9                 mov     [ecx+0E8h], eax
.text:004017FF                 call    NetwalkerGetFunctionFromModuleByCRC32HashFromFunctionNameFunction
.text:00401804                 mov     ecx, NetwalkerGlobalVarMemoryAddressOfBufferReserved
.text:0040180A                 push    998508E2h
.text:0040180F                 push    esi
.text:00401810                 mov     [ecx+0ECh], eax
.text:00401816                 call    NetwalkerGetFunctionFromModuleByCRC32HashFromFunctionNameFunction
.text:0040181B                 mov     ecx, NetwalkerGlobalVarMemoryAddressOfBufferReserved
.text:00401821                 push    2519B15Ah
.text:00401826                 push    esi
.text:00401827                 mov     [ecx+0F0h], eax
.text:0040182D                 call    NetwalkerGetFunctionFromModuleByCRC32HashFromFunctionNameFunction
.text:00401832                 mov     ecx, NetwalkerGlobalVarMemoryAddressOfBufferReserved
.text:00401838                 push    0D2E536B7h
.text:0040183D                 push    esi
.text:0040183E                 mov     [ecx+0F4h], eax
.text:00401844                 call    NetwalkerGetFunctionFromModuleByCRC32HashFromFunctionNameFunction
.text:00401849                 mov     ecx, NetwalkerGlobalVarMemoryAddressOfBufferReserved
.text:0040184F                 add     esp, 40h
.text:00401852                 push    0F54D69C8h
```

Figure 10. Get functions from module, e.g. using a CRC32 hash

If the malware fails to get a function, it will go to a “sleep” call and terminate itself.

Later, the malware extracts the configuration file from a resource with a custom type and a custom name using the functions “FindResourceA”, “LockResource”, “LoadResource” and “SizeOfResource”. The file extracted in memory is decrypted using the RC4 algorithm with a hardcoded key in the resource.

The struct of the resource is:

- 4 bytes -> The size of the hardcoded key to decrypt the configuration file.
- Variable size -> the hardcoded key to decrypt the configuration file.
- Variable size -> the configuration file encrypted.

The malware reads the first 4 bytes and reserves memory with the size of the password and reserves memory of the resource minus 4 bytes and the size of the password. Finally, it decrypts the configuration file:

```
.text:00403AA2                test    eax, eax
.text:00403AA4                jz      _exit___
.text:00403AAA                push    ebp
.text:00403AAB                call    NetwalkerGetMemoryAddressToPointerStructWithFunctionsAddressesFunction
.text:00403AB0                push    ebx
.text:00403AB1                push    edi
.text:00403AB2                mov     ecx, [eax+14Ch] ; SizeOfResource
.text:00403AB8                call    ecx             ; SizeOfResource
.text:00403ABA                mov     esi, eax
.text:00403ABC                push    esi
.text:00403ABD                call    NetwalkerReserveMemoryHeapFunction
.text:00403AC2                mov     ebp, eax
.text:00403AC4                add     esp, 4
.text:00403AC7                test    ebp, ebp
.text:00403AC9                jz      short _return_flag_and_exit
.text:00403ACB                push    esi
.text:00403ACC                push    [esp+18h+var_4]
.text:00403AD0                push    ebp
.text:00403AD1                call    NetwalkerMemcpyWrapperFunction
.text:00403AD6                mov     ebx, [ebp+0]
.text:00403AD9                lea     edi, [ebp+4]
.text:00403ADC                push    ebx
.text:00403ADD                call    NetwalkerRtlAllocateHeapWrapperFunction
.text:00403AE2                add     esp, 10h
.text:00403AE5                mov     [esp+14h+var_4], eax
.text:00403AE9                test    eax, eax
.text:00403AEB                jz      short _check_critical_flag
.text:00403AED                push    ebx
.text:00403AEE                push    edi
.text:00403AEF                push    eax
.text:00403AF0                call    NetwalkerMemcpyWrapperFunction
.text:00403AF5                sub     esi, ebx
.text:00403AF7                add     edi, ebx
.text:00403AF9                sub     esi, 4
.text:00403AFC                push    esi
.text:00403AFD                mov     esi, [esp+24h+var_4]
.text:00403B01                push    edi
.text:00403B02                push    ebx
.text:00403B03                push    esi
.text:00403B04                call    NtwalkerRC4DecryptFunction
```

Figure 11. Get configuration file and decrypt it

If the malware fails to get the configuration file, it will terminate itself.

After getting the configuration file, the malware will parse it and save the fields in memory and write in the registry information to encrypt the files in the machine. The malware will try first to write in the registry-hive “HKEY_LOCAL_MACHINE” but if it cannot create it, it will use the registry-hive “HKEY_CURRENT_USER”:

```
.text:00402E58                 push    1
.text:00402E5A                 mov     eax, [eax+1C8h]
.text:00402E60                 push    0
.text:00402E62                 call    eax
.text:00402E64                 mov     [esi+10h], eax
.text:00402E67
.text:00402E67 _manage_registry:                       ; CODE XREF: NetwalkerWriteDataInRegistryFunction+1A3↑j
.text:00402E67                                         ; NetwalkerWriteDataInRegistryFunction+1B1↑j ...
.text:00402E67                 push    0
.text:00402E69                 push    80000002h       ; HKEY_LOCAL_MACHINE
.text:00402E6E                 call    NetwalkerManageRegistryFunction
.text:00402E73                 add     esp, 8
.text:00402E76                 test    eax, eax
.text:00402E78                 jnz     _get_memory_pointer_address
.text:00402E7E                 push    eax
.text:00402E7F                 push    80000001h       ; HKEY_CURRENT_USER
.text:00402E84                 call    NetwalkerManageRegistryFunction
.text:00402E89                 add     esp, 8
.text:00402E8C                 test    eax, eax
.text:00402E8E                 jnz     _get_memory_pointer_address
.text:00402E94                 mov     ecx, NetwalkerGlobalVarReserveMemory
.text:00402E9A                 lea     eax, [ecx+4Ch]
.text:00402E9D                 push    eax
.text:00402E9E                 lea     eax, [ecx+50h]
.text:00402EA1                 push    eax
.text:00402EA2                 lea     eax, [ecx+44h]
.text:00402EA5                 push    eax
.text:00402EA6                 lea     eax, [ecx+48h]
.text:00402EA9                 push    eax
.text:00402EAA                 push    dword ptr [ecx+40h]
.text:00402EAD                 call    sub_406130
.text:00402EB2                 add     esp, 14h
```

Figure 12. Write in the registry

After the writing in the registry has been completed, it will get some privileges using a token as SE_DEBUG_PRIVILEGE and SE_IMPERSONATE_PRIVILEGE:

```
.text:0040DA53                 mov     [esp+74h+var_3C], 65006Ch
.text:0040DA5B                 mov     [esp+74h+var_38], 650067h ; SeDebugPrivilege
.text:0040DA63                 mov     [esp+74h+var_34], ax
.text:0040DA68                 jmp     short _memset
.text:0040DA6A ; ---------------------------------------------------------------------------
.text:0040DA6A
.text:0040DA6A _check_eax:                             ; CODE XREF: NtwalkerGetSpecialPrivilegesInTokenFunction+D↑j
.text:0040DA6A                 cmp     eax, 1
.text:0040DA6D                 jnz     _exit
.text:0040DA73                 xor     eax, eax        ; set at NULL to end string
.text:0040DA75                 mov     [esp+74h+var_30], 650053h
.text:0040DA7D                 mov     [esp+74h+var_2C], 6D0049h
.text:0040DA85                 lea     edi, [esp+74h+var_30]
.text:0040DA89                 mov     [esp+74h+var_28], 650070h
.text:0040DA91                 mov     [esp+74h+var_24], 730072h
.text:0040DA99                 mov     [esp+74h+var_20], 6E006Fh
.text:0040DAA1                 mov     [esp+74h+var_1C], 740061h
.text:0040DAA9                 mov     [esp+74h+var_18], 500065h
.text:0040DAB1                 mov     [esp+74h+var_14], 690072h
.text:0040DAB9                 mov     [esp+74h+var_10], 690076h
.text:0040DAC1                 mov     [esp+74h+var_C], 65006Ch
.text:0040DAC9                 mov     [esp+74h+var_8], 650067h ; SeImpersonatePrivilege
.text:0040DAD1                 mov     word ptr [esp+74h+var_4], ax
.text:0040DAD6
.text:0040DAD6 _memset:                                ; CODE XREF: NtwalkerGetSpecialPrivilegesInTokenFunction+58↑j
.text:0040DAD6                 push    10h
.text:0040DAD8                 lea     eax, [esp+78h+var_64]
.text:0040DADC                 push    0
.text:0040DADE                 push    eax
.text:0040DADF                 call    NetwalkerWrapperMemsetFunction
.text:0040DAE4                 add     esp, 0Ch
.text:0040DAE7                 call    NetwalkerGetMemoryAddressToPointerStructWithFunctionsAddressesFunction
.text:0040DAEC                 lea     ecx, [esp+74h+var_6C]
.text:0040DAF0                 push    ecx
.text:0040DAF1                 push    edi
.text:0040DAF2                 mov     eax, [eax+218h]
.text:0040DAF8                 push    0
.text:0040DAFA                 call    eax             ; LookupPrivilegeValueW
.text:0040DAFC                 test    eax, eax
.text:0040DAFE                 jnz     short _prepare_open_process_token
```

Figure 13. Get some special privileges in the token

Later, the malware creates three threads, one to get information about the machine, such as the operating system version, one to get processes and the last one to get services in the system.

After this step, it will get the system directory and use “VSSadmin” to delete the Volume Shadow copies of the system. Volume Shadow copies can contain copies of the encrypted files and would be an option to restore from if no backup exists.

```
.text:0040D0E3
.text:0040D0E3 _get_system_directory:              ; CODE XREF: NetwalkerGeySystemDirectoryWAndDeleteShadowVolumesWithVssadminFunction+2F↑j
.text:0040D0E3                 call    NetwalkerGetMemoryAddressToPointerStructWithFunctionsAddressesFunction
.text:0040D0E8                 push    104h
.text:0040D0ED                 push    esi
.text:0040D0EE                 mov     eax, [eax+130h]
.text:0040D0F4                 call    eax             ; GetSystemDirectoyW
.text:0040D0F6                 test    eax, eax
.text:0040D0F8                 jz      _get_peb
.text:0040D0FE                 push    ebx
.text:0040D0FF                 push    ebp
.text:0040D100                 push    edi
.text:0040D101                 mov     [esp+0BCh+var_A8], 76005Ch
.text:0040D109                 mov     [esp+0BCh+var_A4], 730073h
.text:0040D111                 mov     [esp+0BCh+var_A0], 640061h
.text:0040D119                 mov     [esp+0BCh+var_9C], 69006Dh
.text:0040D121                 mov     [esp+0BCh+var_98], 2E006Eh
.text:0040D129                 mov     [esp+0BCh+var_94], 780065h
.text:0040D131                 mov     [esp+0BCh+var_90], 65h
.text:0040D139                 mov     [esp+0BCh+var_8C], 640020h
.text:0040D141                 mov     [esp+0BCh+var_88], 6C0065h
.text:0040D149                 mov     [esp+0BCh+var_84], 740065h
.text:0040D151                 mov     [esp+0BCh+var_80], 200065h
.text:0040D159                 mov     [esp+0BCh+var_7C], 680073h
.text:0040D161                 mov     [esp+0BCh+var_78], 640061h
.text:0040D169                 mov     [esp+0BCh+var_74], 77006Fh
.text:0040D171                 mov     [esp+0BCh+var_70], 200073h
.text:0040D179                 mov     [esp+0BCh+var_6C], 61002Fh
.text:0040D181                 mov     [esp+0BCh+var_68], 6C006Ch
.text:0040D189                 mov     [esp+0BCh+var_64], 2F0020h
.text:0040D191                 mov     dword ptr [esp+0BCh+var_60], 750071h
.text:0040D199                 mov     [esp+0BCh+var_5C], 650069h
.text:0040D1A1                 mov     [esp+0BCh+var_58], 74h ; vssadmin.exe delete shadows /all /quiet      |
.text:0040D1A9                 call    NetwalkerGetMemoryAddressToPointerStructWithFunctionsAddressesFunction
```

Figure 14. Delete the shadow volumes

Later, the malware will enumerate the logical units, prepare the new extension for the future encrypted files, based on the size that is defined in the ransomware config with a random extension, and encrypt all files in the fixed type units and remote units with the new extension.

```
.text:0040D560
.text:0040D560 _wcslen:                           ; CODE XREF: NetwalkerStartMainCriticalPart+13F↓j
.text:0040D560                 lea     ebx, ds:0[esi*2]
.text:0040D567                 add     ebx, ebp
.text:0040D569                 push    ebx
.text:0040D56A                 call    NetwalkerWCSLENFunction
.text:0040D56F                 add     esp, 4
.text:0040D572                 test    eax, eax
.text:0040D574                 jz      short _more_wcslen
.text:0040D576                 push    ebx
.text:0040D577                 call    NetwalkerReserveMemoryAndCopyInsideDataFunction
.text:0040D57C                 mov     edi, eax
.text:0040D57E                 add     esp, 4
.text:0040D581                 test    edi, edi
.text:0040D583                 jz      short _more_wcslen
.text:0040D585                 call    NetwalkerGetMemoryAddressToPointerStructWithFunctionsAddressesFunction
.text:0040D58A                 push    ebx
.text:0040D58B                 mov     ecx, [eax+0FCh]
.text:0040D591                 call    ecx             ; GetDriveTypeW
.text:0040D593                 cmp     eax, 4          ; DRIVE_REMOTE              |
.text:0040D596                 jnz     short _crypt_remote
.text:0040D598                 push    0Ch
.text:0040D59A                 call    NetwalkerReserveMemoryHeapFunction
.text:0040D59F                 add     esp, 4
.text:0040D5A2                 test    eax, eax
.text:0040D5A4                 jz      short _more_wcslen
.text:0040D5A6                 push    eax
.text:0040D5A7                 mov     [eax+4], edi
.text:0040D5AA                 push    offset NetwalkerStartCryptProcedureFunction
.text:0040D5AF                 jmp     short _call_to_the_function_to_will_use_the_crypto_function_needed
.text:0040D5B1 ; ---------------------------------------------------------------------------
```

Figure 15. Crypt the files

After all these steps have been completed, it will create the ransom note on the desktop using the functions “SHGetFolderPathlW” and “CreateFileW”. Subsequently, it will write the ransom note from the memory into a new file with the function “WriteFile”. The malware will create the ransom note in the root folder (for example “c:\”) of each logical unit. Next, it will launch “notepad.exe” with an argument to the ransom note file to show the user what happened on the system:

```
.text:00403442                push    0
.text:00403444                push    edi
.text:00403445                mov     eax, [eax+258h]
.text:0040344B                push    10h
.text:0040344D                push    0
.text:0040344F                call    eax             ; SHGetFolderPathW (Desktop)
.text:00403451                test    eax, eax
.text:00403453                jnz     _release_memory
.text:00403459                mov     dword ptr [esp+8Ch+var_7C], eax
.text:0040345D                lea     eax, [esp+8Ch+var_7C]
.text:00403461                push    eax
.text:00403462                mov     eax, NetwalkerGlobalVarReserveMemory
.text:00403467                push    dword ptr [eax+68h]
.text:0040346A                push    ebp
.text:0040346B                call    NetwalkerJoinDesktopPathWithRansomNoteNameFunction
.text:00403470                add     esp, 0Ch
.text:00403473                test    eax, eax
.text:00403475                jz      _release_memory
.text:0040347B                mov     eax, NetwalkerGlobalVarReserveMemory
.text:00403480                push    dword ptr [eax+70h]
.text:00403483                push    dword ptr [eax+74h]
.text:00403486                push    dword ptr [esp+94h+var_7C]
.text:0040348A                call    NetwalkerCreateFileRansomNoteAndWriteItAndCloseHandleFunction
.text:0040348F                add     esp, 0Ch
.text:00403492                test    eax, eax
.text:00403494                jz      _release_memory_
.text:0040349A                cmp     [esp+8Ch+arg_4], 0
.text:004034A2                jz      _release_memory_
.text:004034A8                xor     eax, eax
.text:004034AA                mov     [esp+8Ch+var_60], 6E005Ch
.text:004034B2                push    104h
.text:004034B7                mov     dword ptr [esp+90h+var_5C], 74006Fh
.text:004034BF                mov     [esp+90h+var_58], 700065h
.text:004034C7                mov     [esp+90h+var_54], 640061h
.text:004034CF                mov     [esp+90h+var_50], 65002Eh
.text:004034D7                mov     [esp+90h+var_4C], 650078h
.text:004034DF                mov     [esp+90h+var_48], ax ; notepad.exe
.text:004034E4                call    NetwalkerReserveMemoryFunctionForUnicodeFunction
```

Figure 16. Creation of the ransom note in the desktop and root units

Finally, after the encryption of the files and creation of the ransom note, the malware creates a bat file in the %temp% folder of the machine with a temporary name and writes the content to destroy itself using the program “taskkill”. The batch script will delete the malware sample with its path using the command “del” and finally delete the bat file with the command “del %0%”. Of course, as the malware uses the “del” command without destroying itself before the deletion, it can be recovered with some forensic tools with luck (the same can also be said for the bat file).

This way the malware tries to remove itself from the machine to avoid being detected and analyzed by security researchers:

```
.text:00403896          call    ecx             ; GetTempFileNameW
.text:00403898          test    eax, eax
.text:0040389A          jz      _release_memory____
.text:004038A0          xor     eax, eax
.text:004038A2          mov     dword ptr [esp+0A0h+var_78], 62002Eh
.text:004038AA          mov     [esp+0A0h+var_74], 740061h
.text:004038B2          mov     [esp+0A0h+var_70], ax ; .bat
.text:004038B7          call    NetwalkerGetMenoryAddressToPointerStructWithFunctionsAddressesFunction
.text:004038BC          lea     ecx, [esp+0A0h+var_78]
.text:004038C0          push    ecx
.text:004038C1          push    esi
.text:004038C2          mov     eax, [eax+30h]
.text:004038C5          call    eax             ; wcscat
.text:004038C7          push    ebx
.text:004038C8          call    NetwalkerStrlenFunction
.text:004038CD          push    eax
.text:004038CE          push    ebx
.text:004038CF          push    esi
.text:004038D0          call    NetwalkerCreateFileRansomNoteAndWriteItAndCloseHandleFunction
.text:004038D5          add     esp, 18h
.text:004038D8          test    eax, eax
.text:004038DA          jz      short _release_memory____
.text:004038DC          push    44h
.text:004038DE          lea     eax, [esp+0A4h+var_44]
.text:004038E2          push    0
.text:004038E4          push    eax
.text:004038E5          call    NetwalkerWrapperMemsetFunction
.text:004038EA          push    10h
.text:004038EC          lea     eax, [esp+0B0h+var_54]
.text:004038F0          push    0
.text:004038F2          push    eax
.text:004038F3          call    NetwalkerWrapperMemsetFunction
.text:004038F8          add     esp, 18h
.text:004038FB          mov     [esp+0A0h+var_18], 1
.text:00403906          xor     eax, eax
.text:00403908          mov     [esp+0A0h+var_14], ax
.text:00403910          call    NetwalkerGetMenoryAddressToPointerStructWithFunctionsAddressesFunction
.text:00403915          lea     ecx, [esp+0A0h+var_54]
.text:00403919          push    ecx
.text:0040391A          lea     ecx, [esp+0A4h+var_44]
```

Figure 17. Get Temp path and make a temporary file as a bat and launch it

Finally, the malware will finish with “ExitProcess”.

Decryptor

When a NetWalker victim goes through technical support (see an example of this below) and pays the ransom demanded by the group they will be able to download the decryptor to clean up their environment.

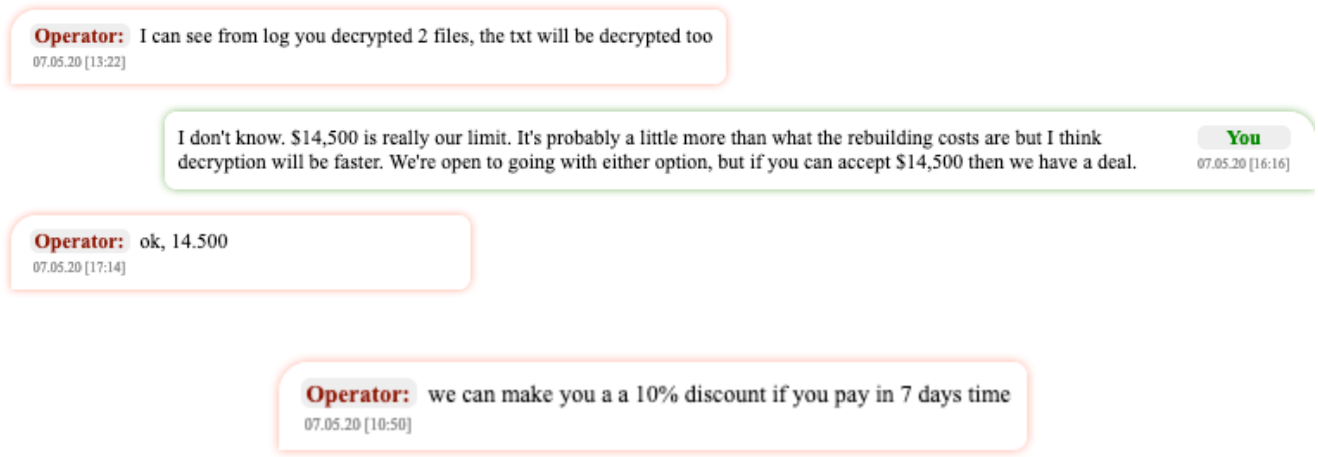


Figure 18. Conversation with NetWalker operators

The download is done directly from the NetWalker Tor site, where the payment page switches to a download page certifying that the payment was made and received:

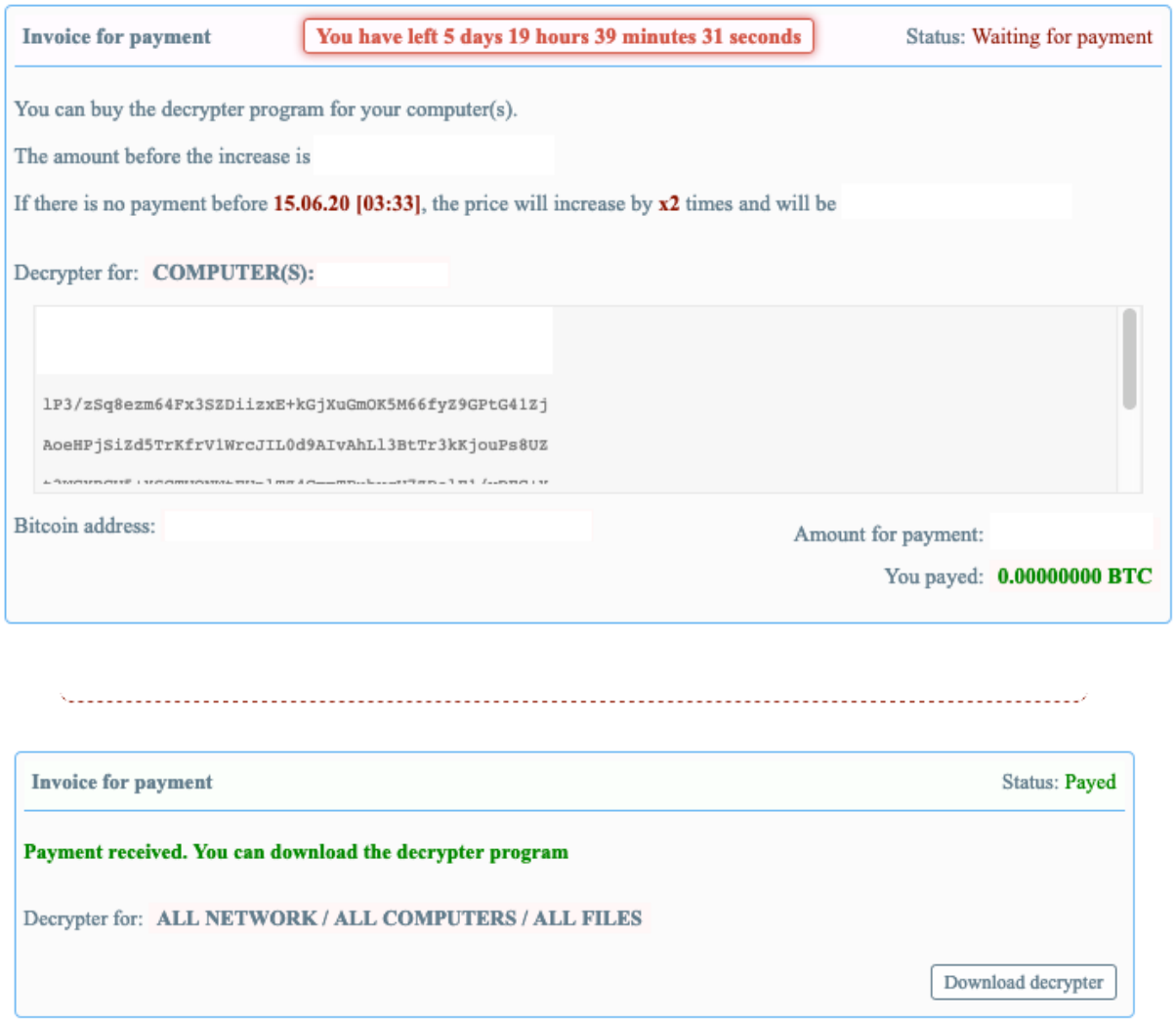


Figure 19. Decryptor download

The decryptor is delivered in a zip archive containing the decryptor executable and a note explaining how to run the program correctly:

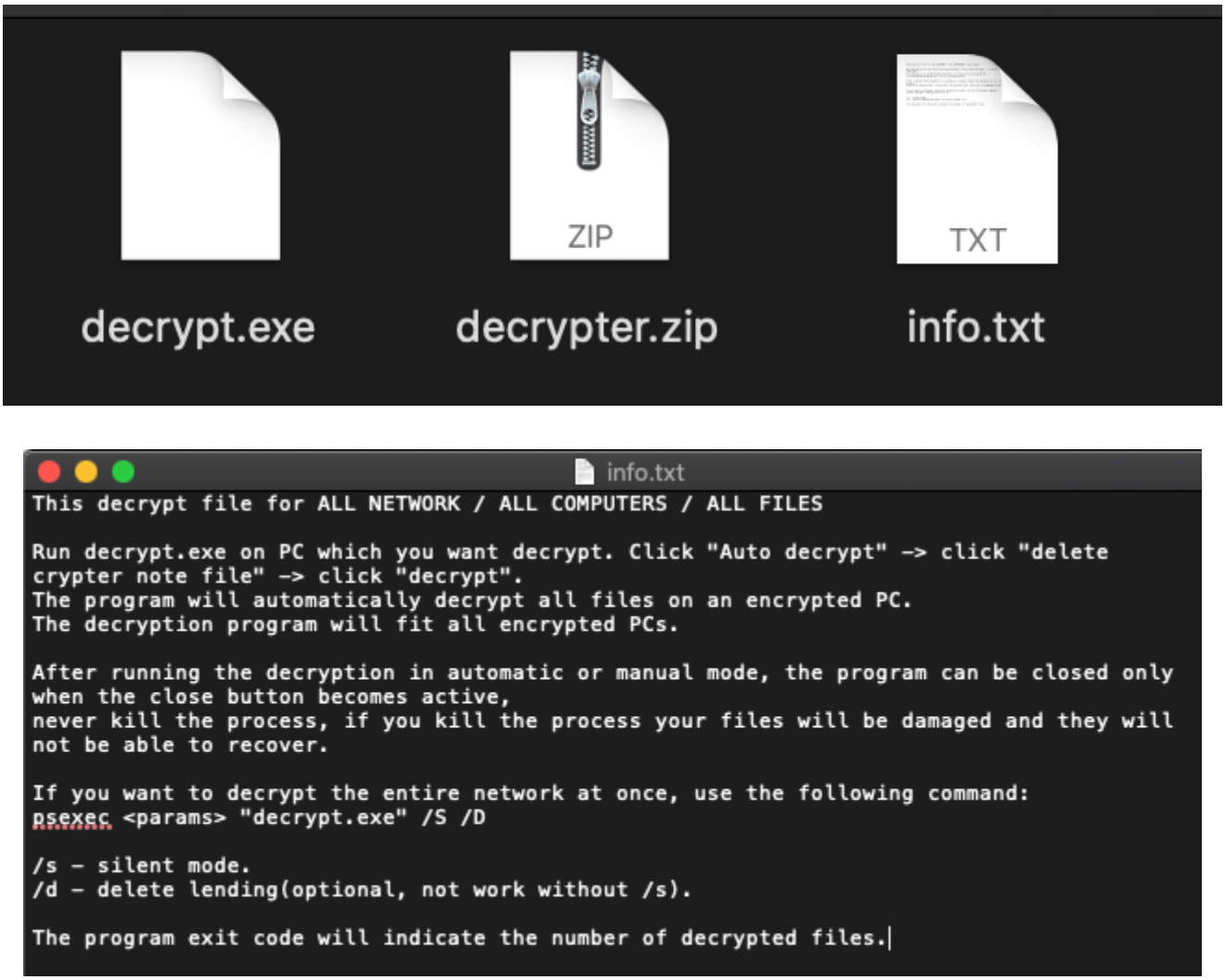


Figure 20. Decryptor delivery

The program launches a graphical interface allowing the user to decipher their workstation automatically or manually:

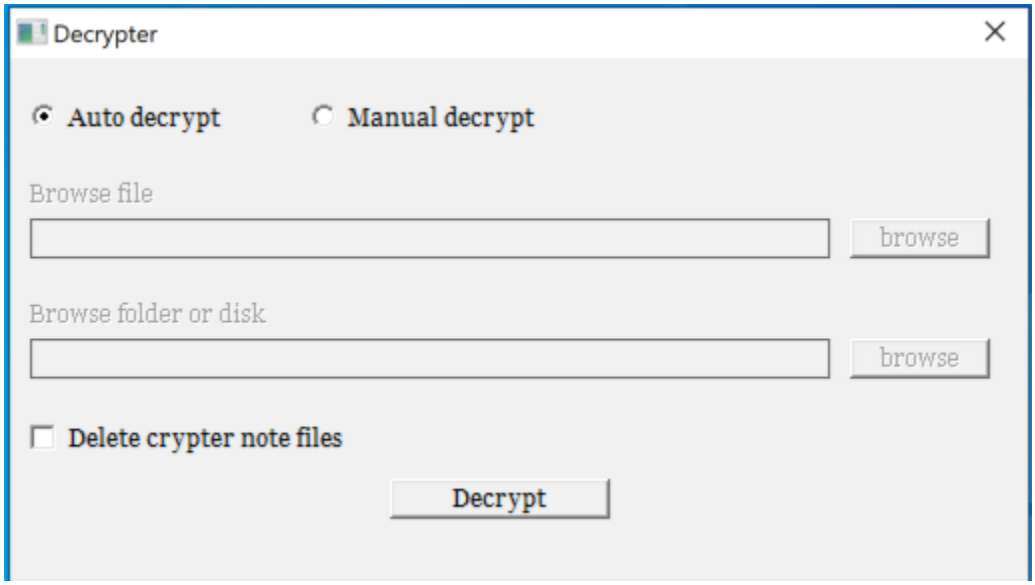


Figure 21. Decryptor execution

At the end of the decryption process, the program indicates the number of decrypted files, deletes the ransom note if the user has checked that option, and terminates, leaving the user to resume their work peacefully:

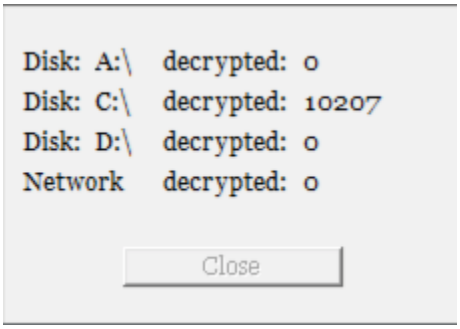


Figure 22. Decryptor has finished the decryption process

The decryptor program appears unique and is linked to one victim specifically. In our example, it only decrypts the files belonging to the victim who made the payment from the user key specified in the ransom note.

Underground Advertising

In March 2020, the moniker Bugatti began actively advertising the NetWalker Ransomware-as-a-Service on two popular underground fora. Bugatti seems to have joined the underground scene in February 2020 but claims to have been active with NetWalker ransomware since September 2019. We have seen NetWalker activity before March but there has been a noticeable uptick in larger victims since their advertisement. For a relatively new ransomware it has been well received and respected among other cybercriminals as compared to, for instance, Nemty ransomware. The strength of NetWalker's reputation is such that our current hypothesis is that the individual behind Bugatti is most likely a well-respected and experienced cybercriminal, even though it is a new moniker.

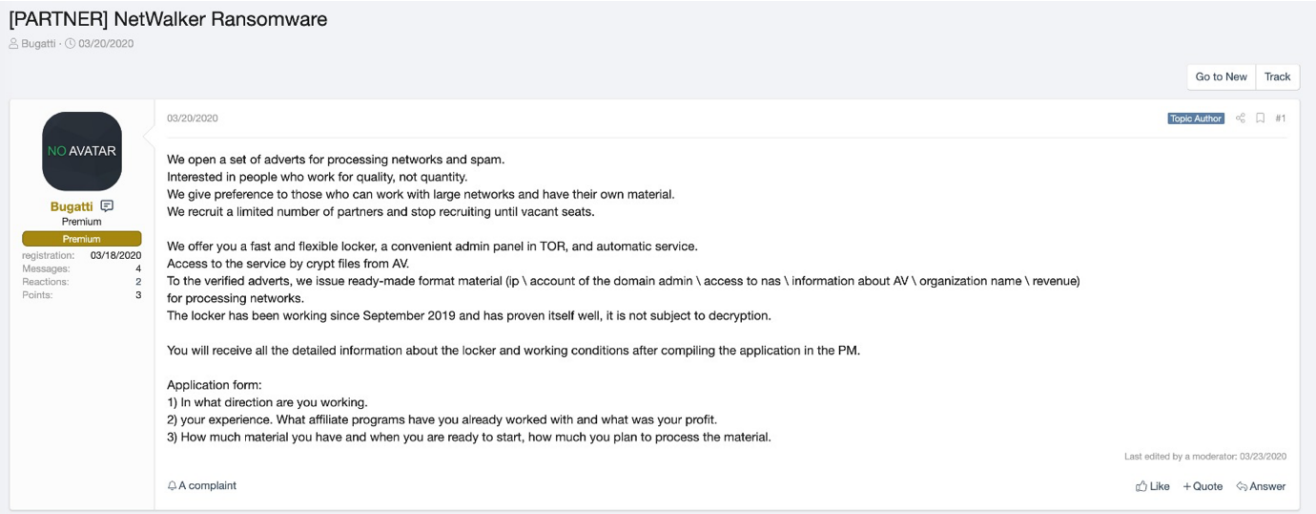


Figure 23. Bugatti advertising NetWalker on an underground forum

Bugatti provides regular updates on the improvements in the ransomware, such as the popular Invoke-ReflectivePEInjection method, also commonly used by Sodinokibi. In addition to the improvements in the ransomware, open slots for new affiliates are advertised. Bugatti strongly emphasized that they are primarily looking for experienced affiliates that focus on compromising the complete networks of organizations as opposed to end users. NetWalker is clearly following in the footsteps of its illustrious targeted ransomware peers like Sodinokibi, Maze and Ryuk.

One forum message in particular caught our attention as it included screenshots of several partial bitcoin addresses and USD amounts. This was most likely done to showcase the financial success of the ransomware. We have seen a similar posting in the past with the influential Sodinokibi affiliate Lalartu, so we decided to follow the money once more.

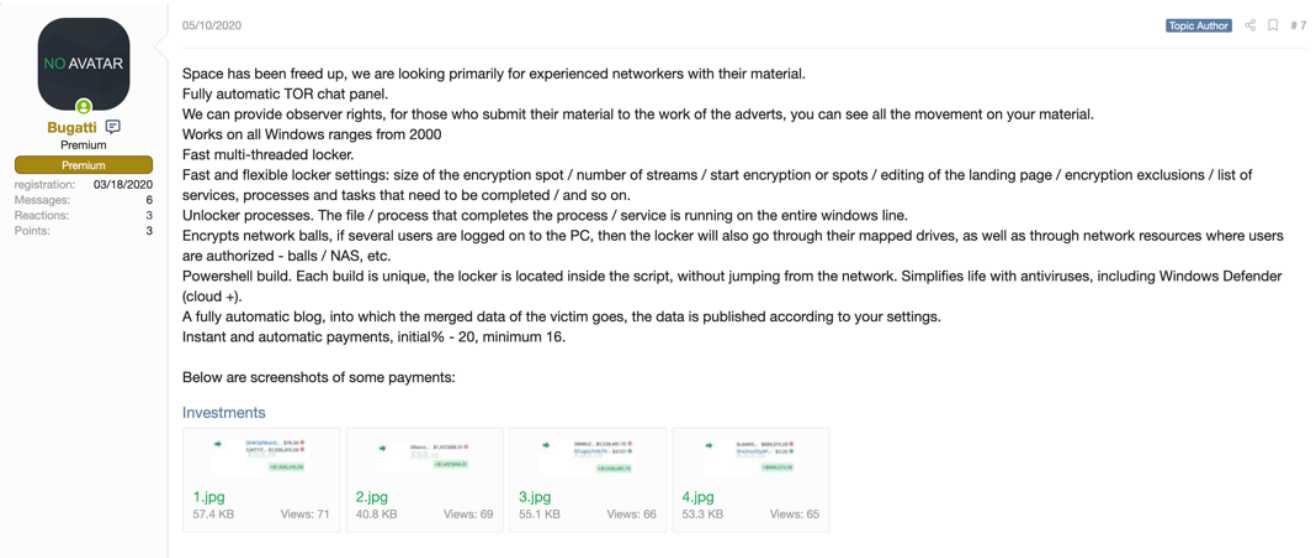


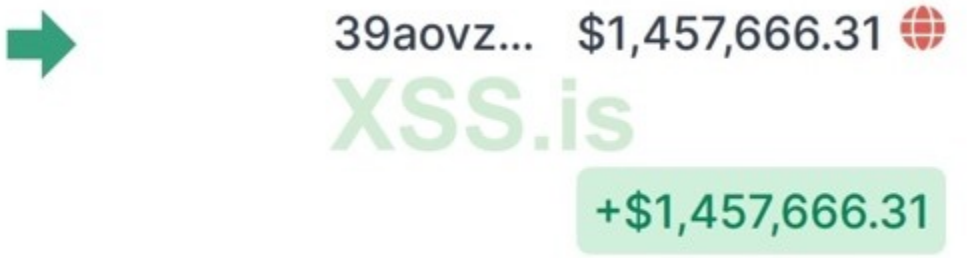
Figure 24. Bugatti is looking for advanced affiliates and shows samples of BTC payments

With the help of CipherTrace software we were able to find the complete BTC addresses from the screenshot and investigate the ledger further:

Screenshot 1

3JHTYZhRmMcq7WCKRzFN98vWvAZk792w9J

Screenshot 2



39aovzbz5rGoQdKjDm6JiybkSu1uGdVJ2V

Screenshot 3

39NRnZtgACDVhhmc7RwmvH9ZDUKTNwwaeB

Screenshot 4

3L4AW5kHnUCZBBjg2j1LBFCUN1RsHPLxCs

Following the Money

In the transactions mentioned in the underground forum post, the ransom amount payed by the victims is presumably shown. Since the bitcoin blockchain is a publicly accessible ledger, we can follow the money and see where the ransomware actors are transferring it to. In the case of the four posted transactions above, the full amount payed by the victim was transferred to two addresses (these addresses begin with bc1q98 and 1DgLhG respectively). It is safe to say that these two bitcoin addresses are under control of the NetWalker actors. We then proceeded on to analyze all incoming transactions to these two addresses and we were able to make the following observations:

- The first incoming transaction occurs on 1 March 2020.
- On 30 March 2020 the first incoming transaction appears where the amount is split between 4 different bitcoin addresses. A split like this is typically seen in Ransomware-as-a-Service, where the ransom payment is split between the RaaS operators and the affiliate who caused the infection. In this first transaction, the split is 80%, 10% and two 5% portions. This split matches the advertisement on the underground forum (80% – 20%).
- The two 5% portions of the ransom payments that are split, seem to be consistently transferred to the two bitcoin addresses we revealed earlier (bc1q98 and 1DgLhG).
- While the beneficiaries of the 5% cuts remain the same, the beneficiary of the 10% cut seems to change over time. Based on the forum post we assume these addresses also belong to the NetWalker actors.
- Payments to the bc1q98 and 1DgLhG addresses that are not being split continue up until the end of May. Possibly the initial NetWalker operators added a RaaS operation, while continuing to cause NetWalker infections themselves.

- While analyzing the bitcoin addresses that received 80% or more of the transaction amount, we noticed that there are some addresses that receive payments multiple times. A possible explanation could be that the address is configured as payout addresses for a certain campaign or affiliate. We identified 30 unique bitcoin addresses that seem to be the beneficiary of this larger portion of the ransom transaction. Some of these only received one payment but there are several that received multiple payments.
- In the two addresses uncovered by tracing the transactions a total of 641 bitcoin is held on 27 July 2020. Which at the current market value of bitcoin is worth well over 7 million USD.

Amounts Extorted

Working under the hypothesis that all the incoming transactions are ransomware payments; we can make the following observations:

- We found 23 transactions where the ransom payments were not split up and the beneficiaries are the two bitcoin addresses found by following the transactions mentioned in the underground forum post. The total amount of bitcoin extorted this way between 1 March 2020 and 27 July 2020 is 677 BTC. Additionally, the amount received from remaining transactions following the Ransomware-as-a-Service scheme by these addresses between 1 March 2020 and 27 July 2020 is 188 BTC.
- In the transactions that are split, the largest amount (usually 80% to 90% of the total transaction value) is presumably transferred to the affiliate that caused the infection. When we summed up these largest portions, we saw a total of 1723 BTC being transferred to affiliates.
- The total amount of extorted bitcoin that has been uncovered by tracing transactions to these NetWalker related addresses is 2795 BTC between 1 March 2020 and 27 July 2020. By using historic bitcoin to USD exchange rates, we estimate a total of 25 million USD was extorted with these NetWalker related transactions.

Even though we do not have complete visibility into the BTC flow before NetWalker started ramping up, one thing is certain, this quarter alone it has been highly successful at extorting organisations for large amounts of money. All this at a time when many sectors are struggling because people are sheltering in place and governments are trying to keep businesses from going bankrupt. NetWalker is making millions off the backs of legitimate companies.

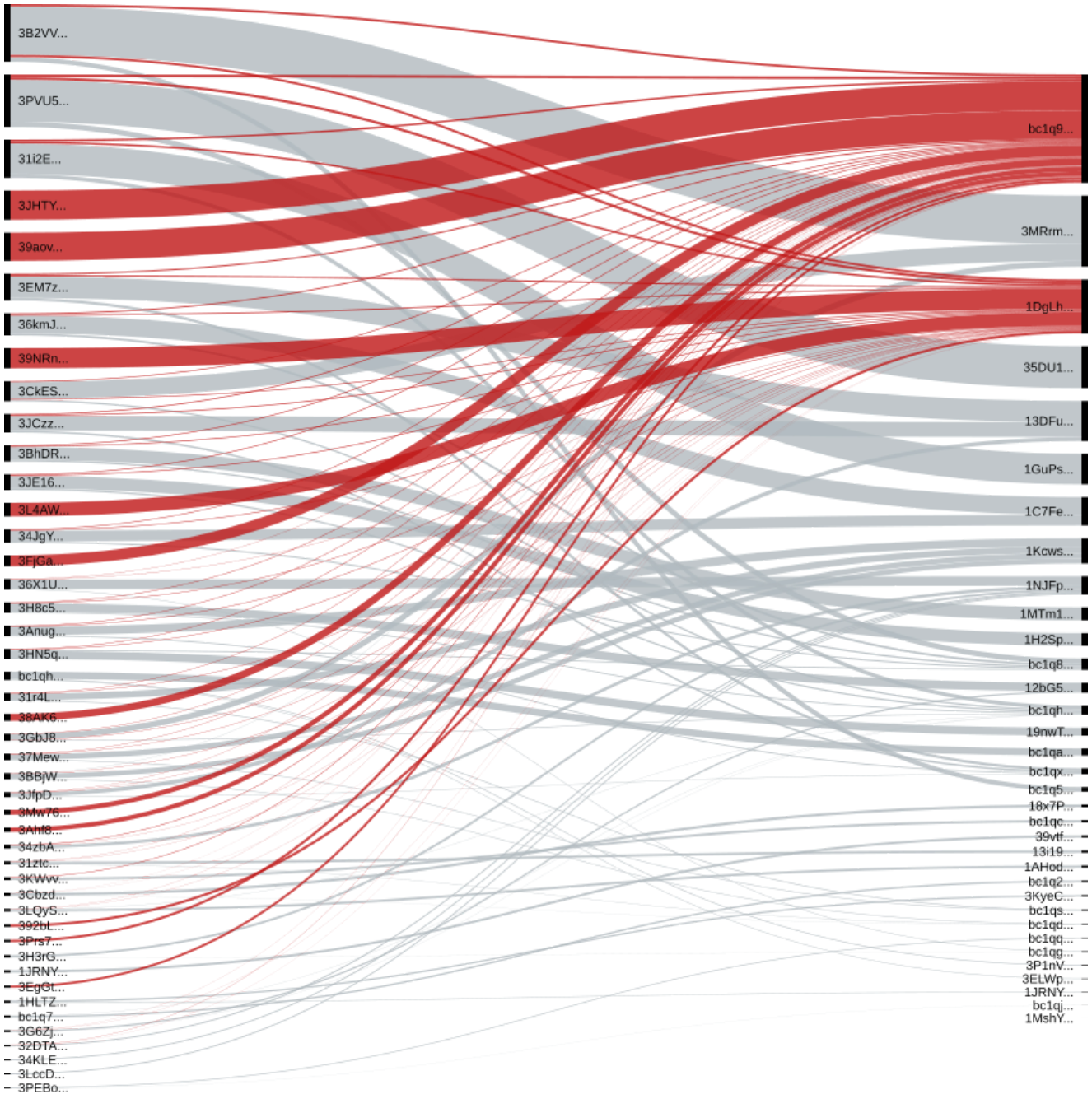


Figure 25. Overview of uncovered bitcoin transactions, highlighting the two identified actor addresses.

Observed Changes

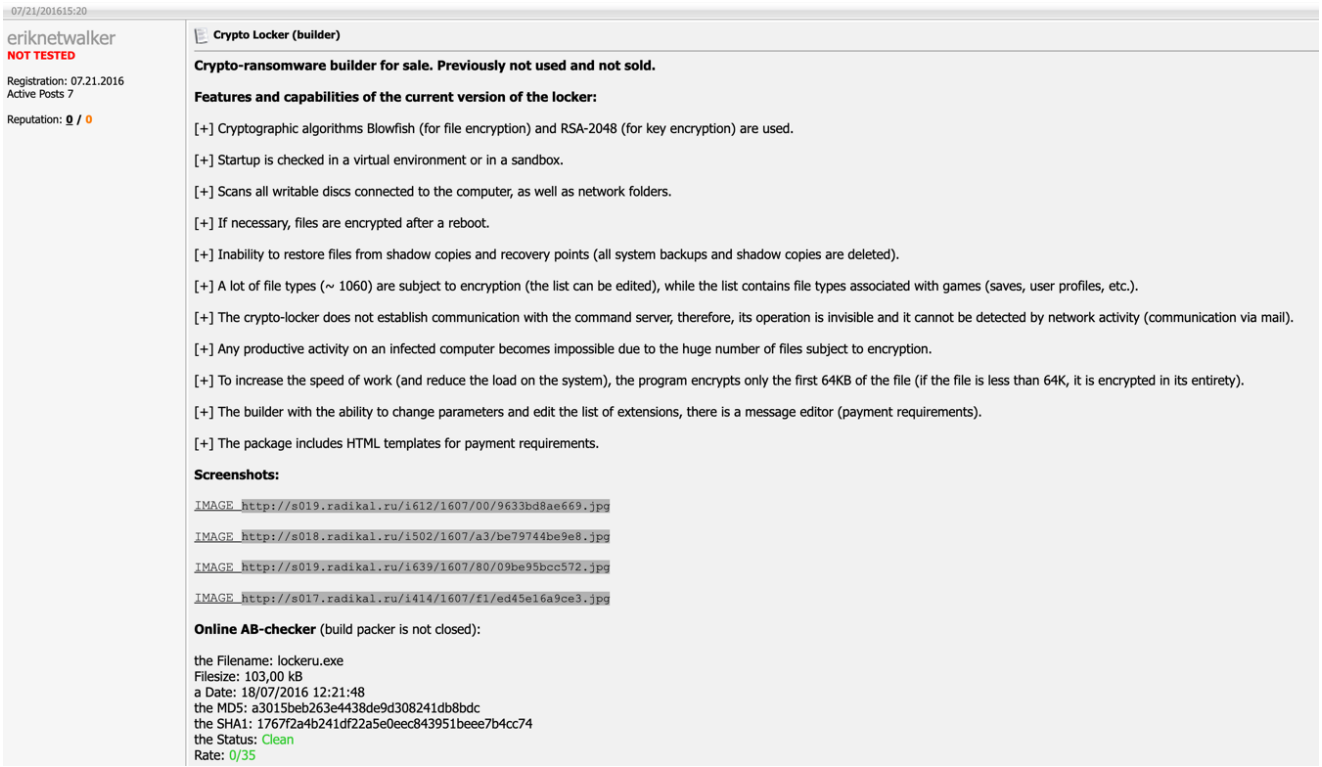
While talking about the impact of NetWalker with our partners, we learned that the change in modus operandi not only affected the way the actors communicate with their victims. When there was a change from email communication to a dedicated Tor hidden service, the actors also moved away from using legacy bitcoin addresses to SegWit addresses. The benefits of using the newer SegWit addresses include faster transaction time and lower transaction cost. The NetWalker advertisement on the underground forum mentions instant and fully

automatic payments around the time of this observed change. This makes us believe the ransomware actors were professionalizing their operation just before expanding to the Ransomware-as-a-Service model.

Comparison with Previous Ransomware

Given the sudden appearance of NetWalker ransomware and the associated threat actor, it suggests that some prior knowledge on ransomware development or underground presence had to be in place. Armed with this hypothesis, we searched for possible links to underground actors and other ransomware strains that might fit the bill. We came across one threat actor offering ransomware that caught our attention. It was the use of the name NetWalker, in combination with a strong ransomware connection, that sparked our interest.

Some years ago, a threat actor using the moniker Eriknetwalker was advertising ransomware on several underground forums. We found posts from 2016 and the latest public activity was around June 2019, several months before NetWalker ransomware made its appearance.


07/21/201615:20

eriknetwalker
NOT TESTED

Registration: 07.21.2016
Active Posts 7

Reputation: 0 / 0

Crypto Locker (builder)

Crypto-ransomware builder for sale. Previously not used and not sold.

Features and capabilities of the current version of the locker:

[+] Cryptographic algorithms Blowfish (for file encryption) and RSA-2048 (for key encryption) are used.

[+] Startup is checked in a virtual environment or in a sandbox.

[+] Scans all writable discs connected to the computer, as well as network folders.

[+] If necessary, files are encrypted after a reboot.

[+] Inability to restore files from shadow copies and recovery points (all system backups and shadow copies are deleted).

[+] A lot of file types (~ 1060) are subject to encryption (the list can be edited), while the list contains file types associated with games (saves, user profiles, etc.).

[+] The crypto-locker does not establish communication with the command server, therefore, its operation is invisible and it cannot be detected by network activity (communication via mail).

[+] Any productive activity on an infected computer becomes impossible due to the huge number of files subject to encryption.

[+] To increase the speed of work (and reduce the load on the system), the program encrypts only the first 64KB of the file (if the file is less than 64K, it is encrypted in its entirety).

[+] The builder with the ability to change parameters and edit the list of extensions, there is a message editor (payment requirements).

[+] The package includes HTML templates for payment requirements.

Screenshots:

IMAGE http://s019.radikal.ru/i612/1607/00/9633bd8ae669.jpg

IMAGE http://s018.radikal.ru/i502/1607/a3/be79744be9e8.jpg

IMAGE http://s019.radikal.ru/i639/1607/80/09be95bcc572.jpg

IMAGE http://s017.radikal.ru/i414/1607/f1/ed45e16a9ce3.jpg

Online AB-checker (build packer is not closed):

the Filename: lockeru.exe
Filesize: 103,00 kB
a Date: 18/07/2016 12:21:48
the MD5: a3015beb263e4438de9d308241db8bdc
the SHA1: 1767f2a4b241df22a5e0eec843951beee7b4cc74
the Status: Clean
Rate: 0/35

Figure 26. Eriknetwalker began advertising their ransomware in 2016 (Google-translated from Russian)

Based on our underground research, we linked the moniker Eriknetwalker to the development and/or distribution of Amnesia, Bomber and Scarab ransomware. Eriknetwalker stopped advertising ransomware around June 2019. Therefore, we decided to perform a comparative analysis between the different ransomware strains linked to Eriknetwalker and some of the earliest versions of NetWalker we could find.

The goal of this comparative analysis was to identify whether there was an overlap between source codes. Such overlap could suggest a stronger link between the current NetWalker version and the other ransomware versions from Eriknetwalker, possibly even explaining the name overlap.

To execute the analysis, we used several tools one of which was the binary visualization tool Veles, which dynamically translates binary information into an abstract visualization that allows us to identify and compare patterns.

The different types of ransomware we began analyzing were the variants of Amnesia, Scarab, and NetWalker.

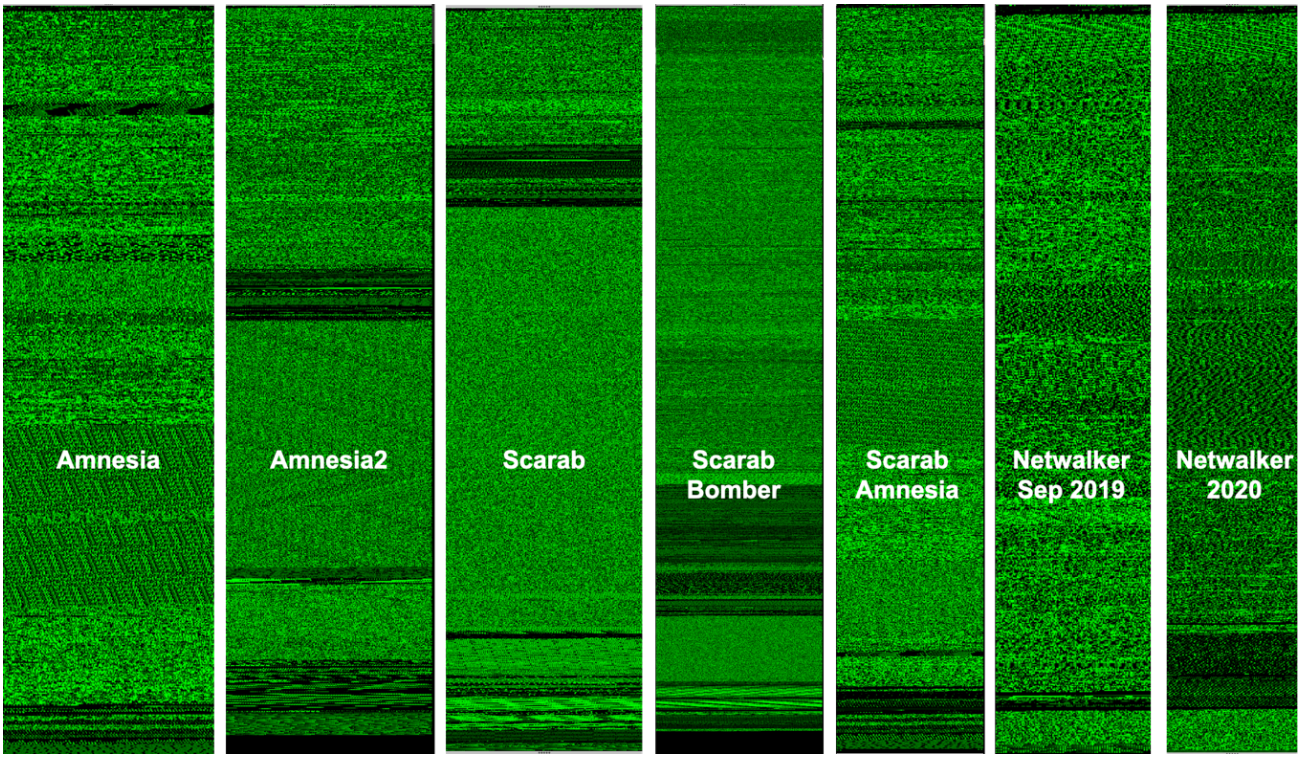


Figure 27. Flat visualization of binary data of the different ransomware variants

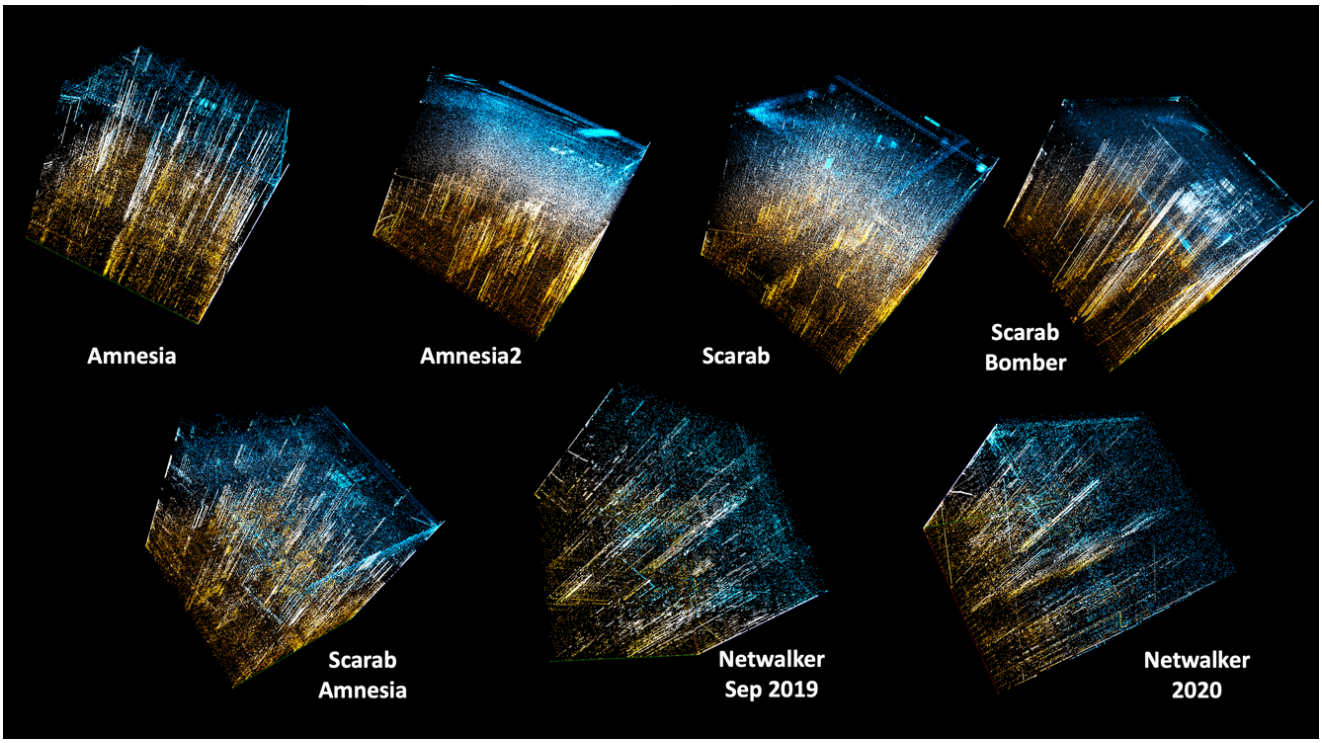


Figure 28. 3D visualization of binary data of the different ransomware variants

Visualizing data in such a manner is a way to use the human brain to quickly identify patterns and be able to draw comparisons between objects. In our case, we see that, based on the binary data visualized in Figures 27 and 28, the ransomware binaries do yield differences that we cannot ignore.

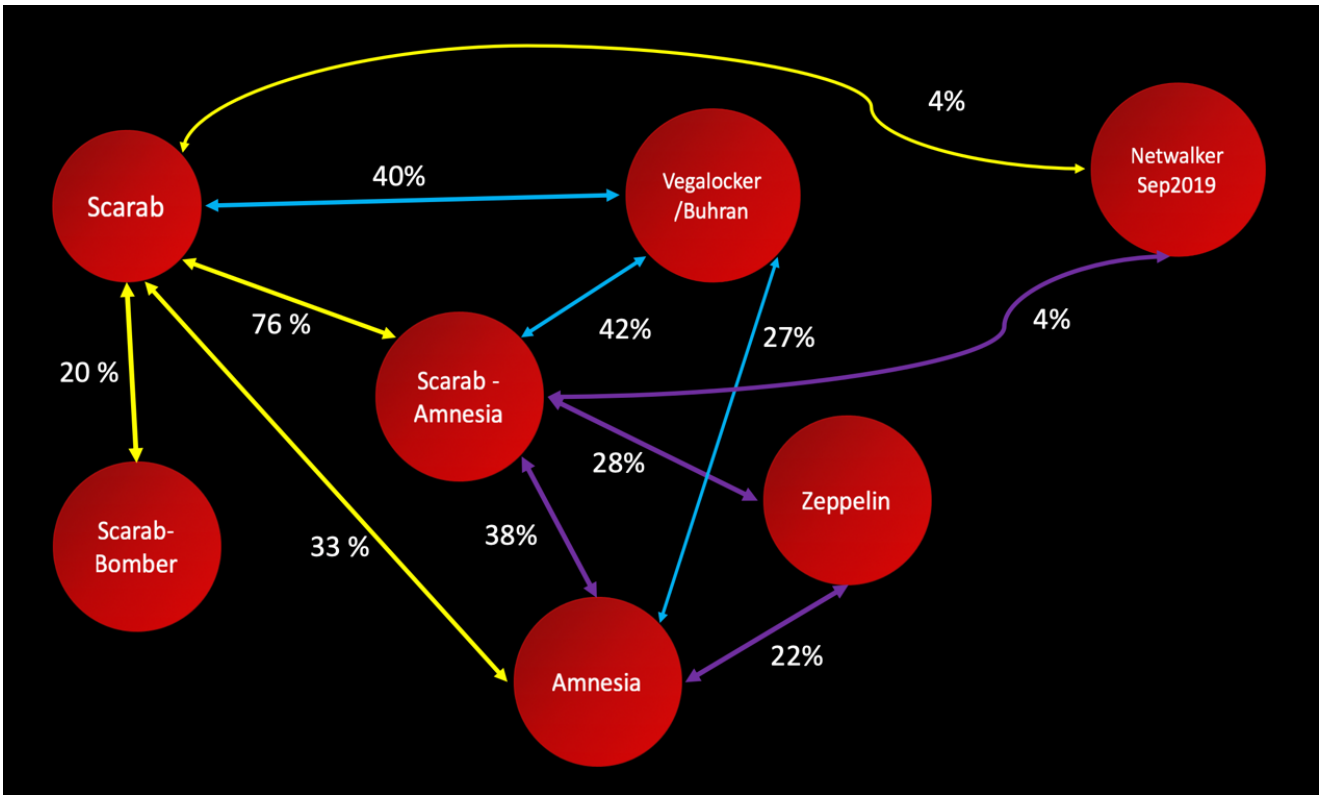


Figure 29. Comparison of source code results

Figure 29 shows the results of a source code similarity analysis led on the different variants of ransomware named in the figure itself. Interesting enough, Scarab and Amnesia show a higher overlap with Buran and Zeppelin than the early NetWalker samples. The percentages shown are the amount of code that is similar between two variants.

As illustrated in the overview, the September 2019 NetWalker version has a different codebase from the ErikNetWalker-linked ransomware variants. This finding disproves our earlier hypothesis that NetWalker is linked to the older Amnesia variants based on code overlap.

Often, research teams do not publish their results when it disproves their own hypothesis. However, for the sake of transparency, we decided to include our research efforts.

YARA Rules

We uploaded a YARA rule to detect almost all the samples observed in the wild to date.

Indicators of Compromise

During our investigation we have observed numerous IoCs linked to NetWalker ransomware. To obtain them please visit our McAfee ATR GitHub site, or get the latest NetWalker IoCs and intelligence on many other threats with McAfee Insights.

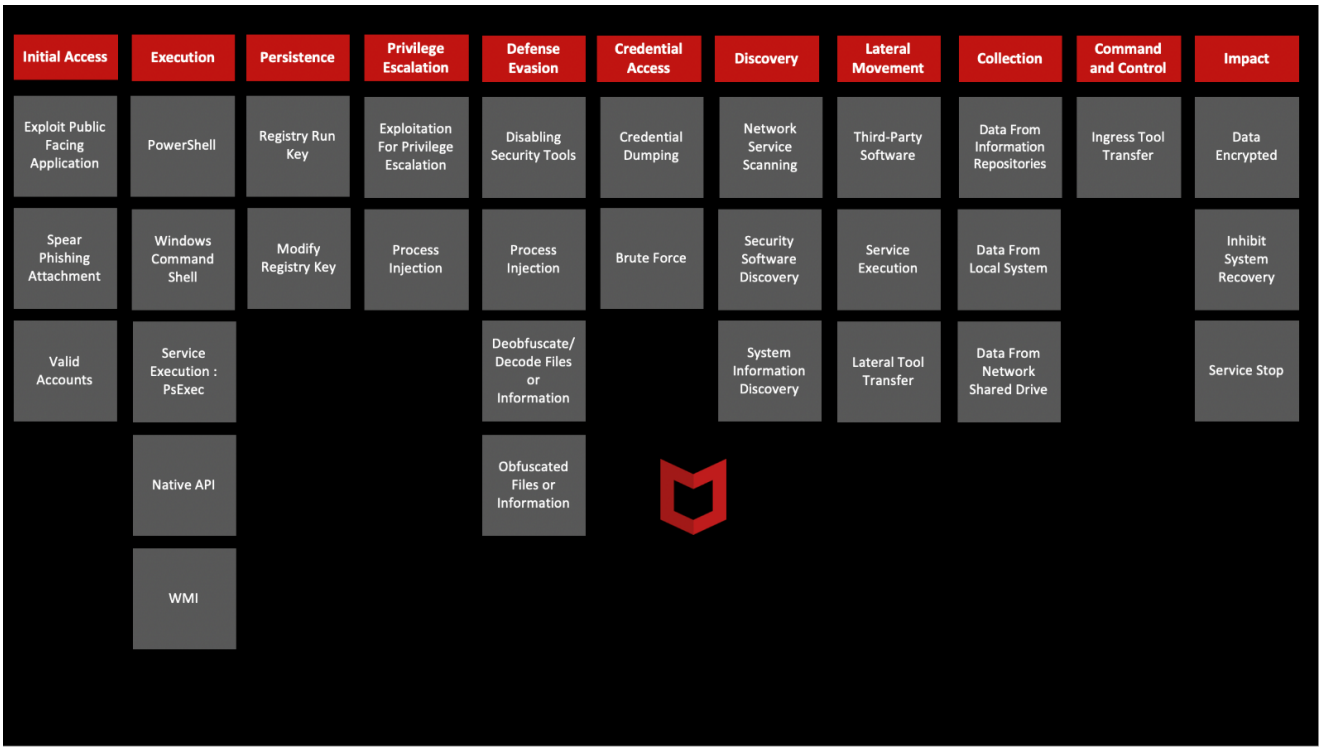
MITRE ATT&CK Techniques

The below techniques were based on our research and complemented with research from industry peers.

- **Initial Access**
  - Exploit Public-Facing Application (T1190) : Exploit Tomcat, Exploit WebLogic
  - Spear phishing Attachment (T1566.001): Phishing email
  - Valid Accounts (T1078): RDP compromised
- **Execution**
  - PowerShell (T1059.001): PowerShell Script
  - Command and Scripting Interpreter: Windows Command Shell (003)
  - Service Execution (T1569.002): PsExec
  - Native API (T1106): Use Windows API functions to inject DLL
  - Windows Management Instrumentation (T1047)
- **Persistence**
  - Registry Run Key (T1547.001): Place a value on RunOnce key
  - Modify Registry key (T1112): Create its own registry key in \SOFTWARE\ <uniquename>

- **Privilege Escalation**
  - Exploitation for Privilege Escalation (T1068): CVE-2020-0796, CVE-2019-1458, CVE-2017-0213, CVE-2015-1701
  - Process Injection (T1055.001): Reflective DLL Injection
- **Defense Evasion**
  - Disabling Security Tools (T1562.001): ESET AV Remover, Trend Micro's Security Agent Uninstall Tool, Microsoft Security Client Uninstall
  - Process Injection (T1055.001): Reflective DLL Injection
  - Deobfuscate/Decode Files or Information (T1140)
  - Obfuscated Files or Information (T1027): PowerShell Script uses Base64 and hexadecimal encoding and XOR-encryption
- **Credential Access**
  - Credential Dumping (T1003): Mimikatz, Mimidogz, Mimikittenz, Windows Credentials Editor, Pwdump, LaZagne
  - Brute Force (T1110.001): NLBrute
- **Discovery**
  - Network Service Scanning (T1046): SoftPerfect Network Scanner
  - Security Software Discovery (T1518.001)
  - System Information Discovery (T1082)
- **Lateral Movement**
  - Third-Party Software (T1072): TeamViewer, Anydesk
  - Service Execution (T 1569.002): PsExec
  - Lateral Tool Transfer (T1570)
- **Collection**
  - Data from information repositories (T1213)
  - Data from local system (T1005)
  - Data from network shared drive (T1039)
- **Command and Control**

  Ingress Tool Transfer (T1105)
- **Impact**
  - Data Encrypted (T1486): NetWalker Ransomware
  - Inhibit System Recovery (T1490): Shadow Copies Deleted
  - Service Stop (T1489)



Conclusion

Ransomware has evolved into a lucrative business for threat actors, from underground forums selling ransomware, to offering services such as support portals to guide victims through acquiring crypto currency for payment, to the negotiation of the ransom. McAfee's Advanced Threat Research team has analysed the NetWalker ransomware and have been following its evolution from the initial sighting of the Mailto ransomware to its redevelopment into the NetWalker ransomware. The recent shift to a business-centric model of Ransomware-as-a-Service is a clear sign that it is stepping up, so it seems that the NetWalker group is following in the footsteps of REvil and other successful RaaS groups. The ransomware developers have proven the ability to refocus and capitalize on current world events and develop lures to help ensure the effectiveness of the ransomware, which has allowed them to become selective of their affiliates by limiting access to the ransomware to only those with vetted access to large organizations. As development of the ransomware continues, we have witnessed recent shifts in activity that closely follow in the footsteps of other ransomware developments, including threatening victims with the release of confidential information if the ransom is not met.

McAfee ATR is actively monitoring ransomware threats and will continue to update McAfee MVISION Insights and its social networking channels with new and current information. MVISION Insights is the only proactive endpoint security solution that simultaneously prioritizes and predicts threats that matter to our customers while offering prescriptive guidance on what to do in their local environment. Want to stay ahead of the adversaries?

Check out McAfee MVISION Insights for more information. If you want to experience some of the MVISION Insights capabilities, go the Preview of MVISION Insights where you can select the top threat information that is available.

Authored by: Thibault Seret, Valentine Mairet, Jeffrey Sman, Alfred Alvarado, Tim Hux, Alexandre Mundo, John Fokker, Marc Rivero Lopez and Thomas Roccia.

ATR Operational Intelligence Team
McAfee's Advanced Threat Research Operational Intelligence team operates globally around the clock, keeping watch of the latest cyber campaigns and actively tracking the most impactful cyber threats.