



CYBER THREAT INTELLIGENCE



RACCOONSTEALER V2.0

TECHNICAL ANALYSIS REPORT

Contents

File.exe Analysis	2
General Review	2
Stage 2 Analysis	3
DLL Detection	3
Process Detection.....	4
Computer Name Detection	5
Username Detection	6
InstallUtil.exe Analysis.....	7
General Review	7
Dynamic Analysis.....	7
Getting API Function Address.....	7
String Resolving Algorithm	8
Process Access Detection	8
Creation of Request Contents	11
Network Analysis.....	12
Request Analysis	12
After Response	14
Gathering Device Information.....	14
DLL Loading	18
Database Operations	18
File Traversal Algorithm	19
Additional Analysis	20
SQL Queries	20
YARA Rule	21
MITRE ATTACK TABLE	22
Mitigations	22

File.exe Analysis

General Review

SHA 256	1976859574585aac13a24b6696cec26479029a92334c721ec71492094a7edec3
Name	file.exe
File Type	PE32-EXE

Table 1 file.exe file information

The screenshot shows the OllyDbg debugger interface. The assembly window displays the following code:

```

    push r11
    mov edx,dword ptr ds:[55C4B0]
    push edx
    call dword ptr ds:[<&GetProcAddress>]
    mov dword ptr ds:[<&VirtualProtect>],eax
    lea eax,dword ptr ss:[ebp-4]
    push eax
    push 40
    mov ecx,dword ptr ss:[ebp+C]
    push ecx
    mov edx,dword ptr ss:[ebp+8]
    push edx
    call dword ptr ds:[<&VirtualProtect>]
    mov esp,ebp
    pop ebp
    ret

```

The instruction at address 0054BA62 is highlighted in red. The CPU tab shows the registers and memory dump tabs are visible.

Figure 1 Obtaining executable permission for the area where the analysed code is written

The screenshot shows the OllyDbg debugger interface. The assembly window displays the following code:

```

    stc
    push ds
    jmp file.549D083
    pushfd
    les edx,fword ptr ss:[ebp]
    mov dword ptr ss:[ebp-4],eax
    call dword ptr ss:[ebp-4]

```

The instruction at address 00549DE9 is highlighted in red. The CPU tab shows the registers and memory dump tabs are visible.

Figure 2 Call to the starting address of the decoded code

Malware has been detected and unpacked.

Stage 2 Analysis

At this stage, it was determined that malware applied analysis detection techniques.

DLL Detection

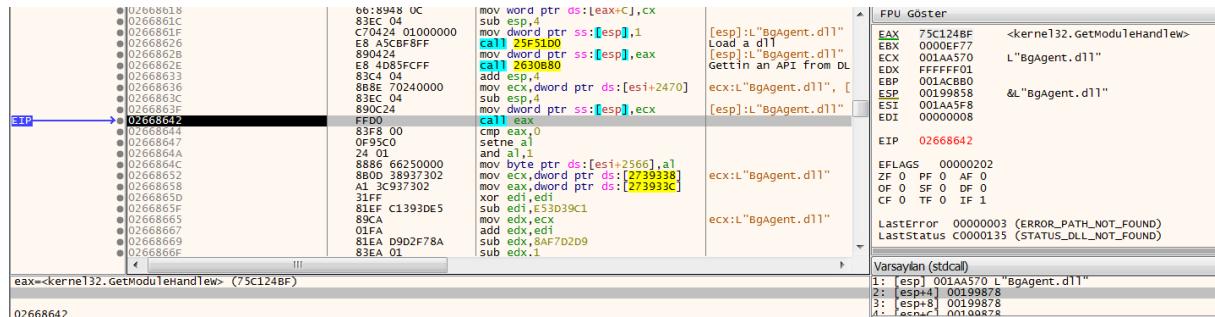


Figure 3 Determination of the existence of the dynamic library names analysed at runtime on the computer

It has been observed that the malware tries to detect DLL files belonging to some security products and systems. The DLLs it tries to detect and the systems they belong to are as follows:

CWSandbox	api_log.dll
	dir_watch.dll
	pstorec.dll
Sandboxie	sbieDII.dll
ThreatExpert	dbghelp.dll
Comodo	cmdvrt32.dll /cmdvrt64.dll
BullGuard	BgAgent.dll

Table 2 Names of the checked dynamic library files and the systems they belong to

ECHO

Process Detection

Figure 4 shows a screenshot of the Immunity Debugger interface. The CPU tab is selected, displaying assembly code. A red arrow points to the instruction at address 0265981F, which is part of the kernel32.createToolhelp32Snapshot function. The assembly code includes various mov, add, and call instructions, with some labels like 'Z31E1F0' and '25F510'. The right side of the screen shows registers (EAX-EIP) and memory dump tabs.

Figure 4 Taking a snapshot of the processes running in the background

Figure 5 shows a screenshot of the Immunity Debugger interface. A blue arrow points to the instruction at address 0265A433, which is part of the 32.lstrcmpiW function. The assembly code includes cmp, add, and mov instructions, with labels like '25F5100' and '265BE80'. The right side of the screen shows registers and memory dump tabs.

Figure 5 Process Blacklist Control

Figure 6 shows a screenshot of the Immunity Debugger interface. A blue arrow points to the instruction at address 0265AEBC, which is part of the kernel32.lstrcmpiW function. The assembly code includes cmp, add, and mov instructions, with labels like '25F5100' and '265BE80'. The right side of the screen shows registers and memory dump tabs.

Figure 6 Process Blacklist Control

It was observed that the malware compares the processes running in the background with its own blacklist. The process list compared is as follows:

- fmon.exe
- WRSA.exe
- PSUAService.exe
- BullGuardCore.exe

ECHO

Computer Name Detection

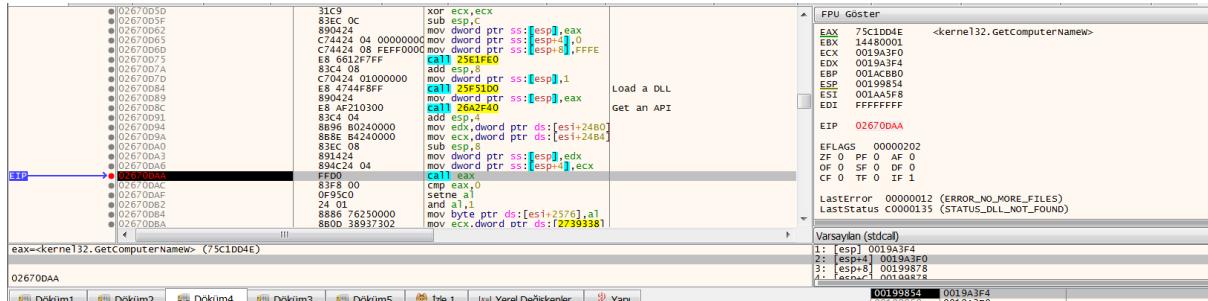


Figure 7 Withdrawing the victim's computer name

0267782A		Döküm1	Döküm2	Döküm4	Döküm3	Döküm5	İzle 1	İç=1 Yerel Değişken
Adres	Hex	ASCII						
0019A2A4	B4 A2 19 00	B4 A2 19 00	B4 A2 19 00	B4 A2 19 00	B4 A2 19 00	B4 A2 19 00	B4 A2 19 00	B4 A2 19 00
0019A2B4	37 00 53 00	49 00 4C 00	56 00 49 00	41 00 00 00	73 .I..C..	73 .I..C..	73 .I..C..	73 .I..C..
0019A2C4	6B 00 6C 00	6F 00 66 00	65 00 5F 00	78 00 36 00	K.L.o.n.e._X.6	K.L.o.n.e._X.6	K.L.o.n.e._X.6	K.L.o.n.e._X.6
0019A2D4	34 00 2D 00	70 00 63 00	00 00 00 00	49 00 6E 00	4..-p.c..	4..-p.c..	4..-p.c..	4..-p.c..
0019A2E4	73 00 69 00	64 00 65 00	54 00 6D 00	00 00 00 00	s.i.d.e.T.m..	s.i.d.e.T.m..	s.i.d.e.T.m..	s.i.d.e.T.m..
0019A2F4	54 00 55 00	2D 00 34 00	4E 00 48 00	30 00 39 00	T.U.-4.N.H.O.9	T.U.-4.N.H.O.9	T.U.-4.N.H.O.9	T.U.-4.N.H.O.9
0019A304	53 00 4D 00	43 00 47 00	31 00 48 00	43 00 00 00	S.M.C.G.1.H.C..	S.M.C.G.1.H.C..	S.M.C.G.1.H.C..	S.M.C.G.1.H.C..
0019A314	54 00 45 00	51 00 55 00	49 00 4C 00	41 00 42 00	T.E.Q.U.I.L.A.B	T.E.Q.U.I.L.A.B	T.E.Q.U.I.L.A.B	T.E.Q.U.I.L.A.B
0019A324	4F 00 4F 00	4D 00 42 00	4F 00 4F 00	4D 00 00 00	O.O.M.B.O.O.M..	O.O.M.B.O.O.M..	O.O.M.B.O.O.M..	O.O.M.B.O.O.M..
0019A334	46 00 4F 00	52 00 54 00	49 00 4E 00	45 00 54 00	E.O.R.T.I.N.E.T	E.O.R.T.I.N.E.T	E.O.R.T.I.N.E.T	E.O.R.T.I.N.E.T
0019A344	00 00 00 00	57 00 49 00	4E 00 37 00	2D 00 54 00W.I.N.7.-TW.I.N.7.-TW.I.N.7.-TW.I.N.7.-T
0019A354	52 00 41 00	50 00 53 00	00 00 00 00	4D 00 55 00	R.A.P.S....M.U	R.A.P.S....M.U	R.A.P.S....M.U	R.A.P.S....M.U
0019A364	45 00 4C 00	4C 00 45 00	52 00 2D 00	50 00 43 00	E.L.L.E.R.-P.C	E.L.L.E.R.-P.C	E.L.L.E.R.-P.C	E.L.L.E.R.-P.C
0019A374	00 00 00 00	48 00 41 00	4E 00 53 00	50 00 45 00H.A.N.S.P.EH.A.N.S.P.EH.A.N.S.P.EH.A.N.S.P.E
0019A384	54 00 45 00	52 00 2D 00	50 00 43 00	00 00 00 00	T.E.R.-P.C..	T.E.R.-P.C..	T.E.R.-P.C..	T.E.R.-P.C..
0019A394	1A 00 4F 00	48 00 4E 00	2D 00 50 00	43 00 00 00	J.O.H.N.-P.C..	J.O.H.N.-P.C..	J.O.H.N.-P.C..	J.O.H.N.-P.C..
0019A3A4	53 00 41 00	4E 00 44 00	42 00 4F 00	58 00 00 00	S.A.N.D.B.O.X..	S.A.N.D.B.O.X..	S.A.N.D.B.O.X..	S.A.N.D.B.O.X..
0019A3B4	74 00 7A 00	00 00 00 00	4E 00 66 00	5A 00 74 00	t.z....N.F.Z.t	t.z....N.F.Z.t	t.z....N.F.Z.t	t.z....N.F.Z.t
0019A3C4	46 00 62 00	50 00 66 00	48 00 00 00	68 00 66 00	F.b.P.F.H..h.f	F.b.P.F.H..h.f	F.b.P.F.H..h.f	F.b.P.F.H..h.f
0019A3D4	76 00 64 00	68 00 78 00	00 00 00 00	45 00 4C 00	v.d.h.x....E.L	v.d.h.x....E.L	v.d.h.x....E.L	v.d.h.x....E.L
0019A3E4	49 00 43 00	5A 00 00 00	FO A3 19 00	OF 00 00 00	I.C.Z....ðf..	I.C.Z....ðf..	I.C.Z....ðf..	I.C.Z....ðf..
0019A3F4	57 00 49 00	4E 00 2D 00	4C 00 31 00	4B 00 44 00	W.I.N.-L.1.K.D	W.I.N.-L.1.K.D	W.I.N.-L.1.K.D	W.I.N.-L.1.K.D

Figure 8 Black list of computer names after analyses

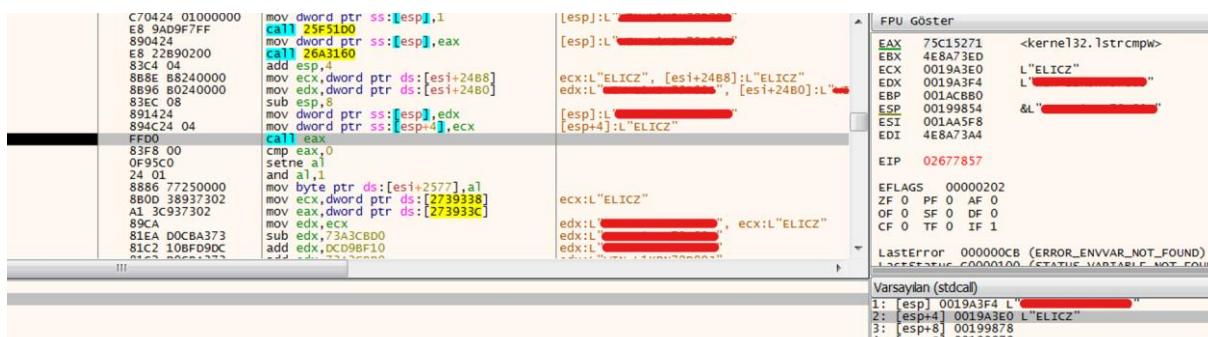


Figure 9 Comparing the computer names in the list with the victim computer name in order

It was observed that malware analyses a list of names and compares it with the name of the computer on which it was found. Analysed computer names:

SANDBOX	JOHN-PC	HANSPETER-PC	MUELLER-PC
WIN7-TRAPS	FORTINET	TEQUILABOOMBOOM	TU-4NH09SMCG1HC
InsideTm	klone_x64-pc	7SILVIA	tz
NfZt	FbPfH	hfvdhx	ELICZ

Table 3 Analysed computer names

ECHO

Username Detection

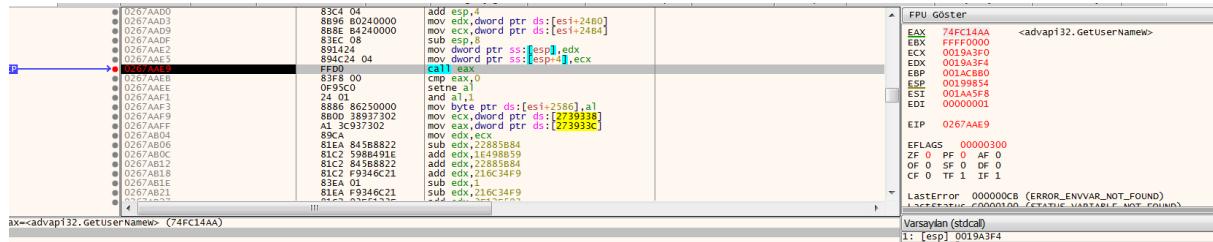


Figure 10 User name retrieval process

0019A0D8	E4 A0 19 00	E4 A0 19 00	E4 A0 19 00	54 00 45 00	ā . . ā . . T. E.
0019A0E8	51 00 55 00	49 00 4C 00	41 00 42 00	4F 00 4F 00	Q. U. I. L. A. B. O. O.
0019A0F8	4D 00 42 00	4F 00 4F 00	4D 00 00 00	73 00 61 00	M. B. O. M. S. A.
0019A108	6E 00 64 00	62 00 6F 00	78 00 00 00	74 00 69 00	n. d. b. o. x. t. i.
0019A118	5D 00 6D 00	79 00 00 00	4A 00 6F 00	68 00 6E 00	m. m. y. . J. o. h. n.
0019A128	20 00 44 00	6F 00 65 00	00 00 00 00	77 00 69 00	. D. o. e. . . . w. i.
0019A138	6C 00 62 00	65 00 72 00	74 00 00 00	76 00 69 00	T. b. e. r. t. . . v. i.
0019A148	72 00 75 00	73 00 63 00	6C 00 6F 00	6E 00 65 00	r. u. s. c. l. o. n. e.
0019A158	00 00 00 00	73 00 6E 00	6F 00 72 00	74 00 00 00	. . . s. n. o. r. t.
0019A168	41 00 6E 00	64 00 79 00	00 00 00 00	76 00 69 00	A. n. d. y. . . . v. i.
0019A178	72 00 75 00	73 00 65 00	74 00 65 00	73 00 74 00	r. u. s. t. e. s. t.
0019A188	20 00 75 00	73 00 65 00	72 00 00 00	6D 00 61 00	u. s. e. r. . . m. a.
0019A198	6C 00 74 00	65 00 73 00	74 00 00 00	6D 00 61 00	l. t. e. s. t. . . m. a.
0019A1A8	6C 00 77 00	61 00 72 00	65 00 00 00	73 00 61 00	l. w. a. r. e. . . s. a.
0019A1B8	5E 00 64 00	20 00 62 00	6F 00 78 00	00 00 00 00	n. d. . b. o. x. . . .
0019A1C8	50 00 65 00	74 00 65 00	72 00 20 00	57 00 69 00	P. e. t. e. r. . . w. i.
0019A1D8	6C 00 73 00	6F 00 6E 00	00 00 00 00	6D 00 69 00	l. s. o. n. . . m. i.
0019A1E8	6C 00 6F 00	7A 00 73 00	00 00 00 00	4D 00 69 00	l. o. z. s. . . M. i.
0019A1F8	6C 00 6C 00	65 00 72 00	00 00 00 00	4A 00 6F 00	l. l. e. r. . . . j. o.
0019A208	68 00 6E 00	73 00 6F 00	6E 00 00 00	49 00 54 00	h. n. s. o. n. . I. T.
0019A218	2D 00 41 00	44 00 4D 00	49 00 4E 00	00 00 00 00	- A. D. M. I. N. . . .
0019A228	48 00 6F 00	6E 00 67 00	20 00 4C 00	65 00 65 00	H. o. n. g. . . L. e. e.
0019A238	00 00 00 00	48 00 41 00	50 00 55 00	42 00 57 00	.. . H. A. P. U. B. W.
0019A248	53 00 00 00	45 00 6D 00	69 00 6C 00	79 00 00 00	S. . . E. m. i. l. y. . .
0019A258	43 00 75 00	72 00 72 00	65 00 6E 00	74 00 55 00	C. u. r. r. e. n. t. U.
0019A268	73 00 65 00	72 00 00 00	B4 A2 19 00	B4 A2 19 00	s. e. r. . . . C. . .
0019A278	B4 A2 19 00	B4 A2 19 00	B4 A2 19 00	B4 A2 19 00	C. . . . C. . . .

Figure 11 Post-analysis username blacklist

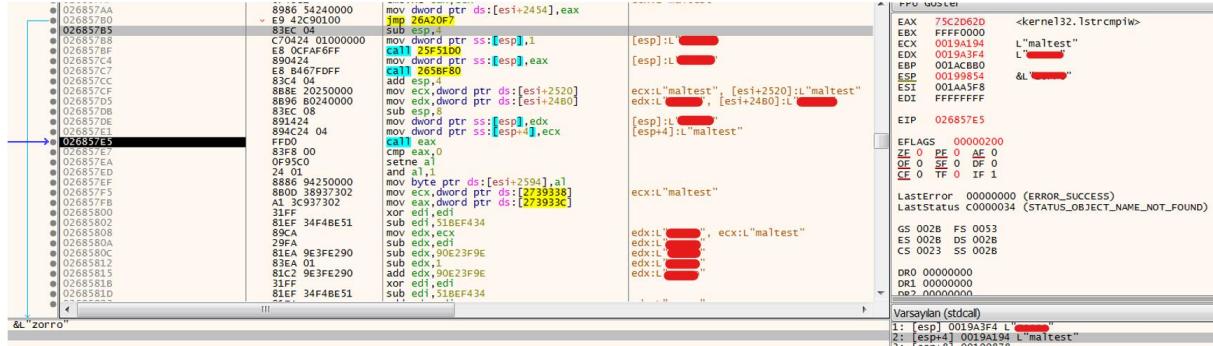


Figure 12 The process of comparing the user names in the list

The malware was also found to analyse the username list and compare it with the username of the computer it was found on. Compared usernames:

CurrentUser	sandbox	Emily	HAPUBWS
Hong Lee	IT-ADMIN	Johnson	Miller
TEQUILABOOMBOOM	milozs	Peter Wilson	sand box
malware	maltest	test user	virus
Andy	snort	virusclone	wilbert
virusClone	John Doe	timmy	

Table 4 Checked user names

It was detected that the malware executes malicious code with the ProcessHollowing technique in the executable file "C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe".

InstallUtil.exe Analysis

General Review

SHA256	6052F7D7832F6EDDF1BA8309F189FCCCB9917128D216FC1C181327B3DEBDEDAC
Name	InstallUtil.exe
File Type	PE32-EXE

Table 5 InstallUtil.exe file information

Dynamic Analysis

Getting API Function Address

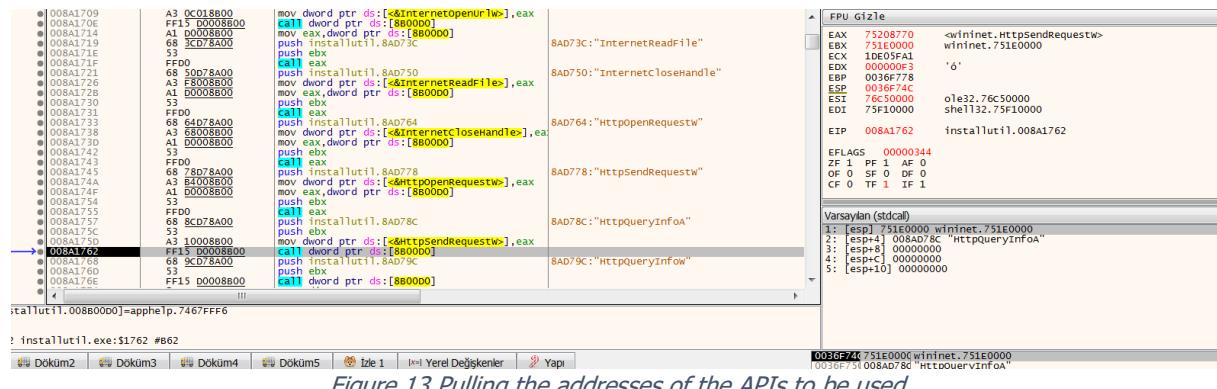


Figure 13 Pulling the addresses of the APIs to be used

It has been determined that it has received the addresses of the API Functions it will use. The functions whose addresses he received are as follows:

GetFileSize	GetDriveType	GetFileSize	GetDriveType
GetModuleFileNameW	GetSystemInfo	GetModuleFileNameW	GetSystemInfo
wideCharToMultiByte	ShellExecuteW	wideCharToMultiByte	ShellExecuteW
PathMatchSpecW	InternetReadFile	PathMatchSpecW	InternetReadFile
HttpSendRequestW	HttpQueryInfoA	HttpSendRequestW	HttpQueryInfoA

Table 6 APIs whose addresses were retrieved

ECHO



Figure 14 Mutex Creation

It was also detected that malware created a mutex named "**264782971_qJ5tS2bD5fD1nZ5kD2kV**".

String Resolving Algorithm

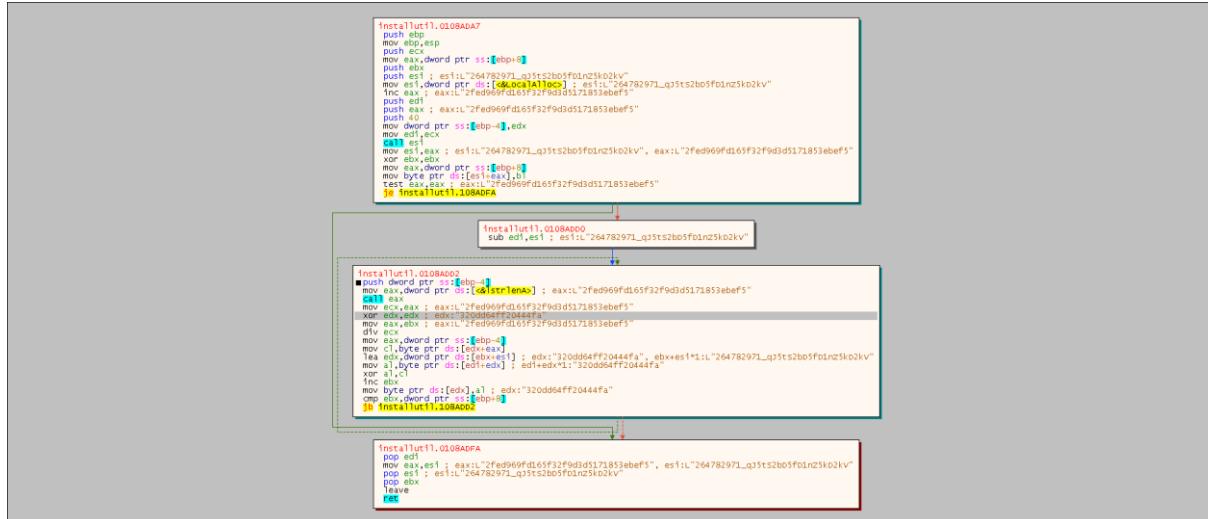


Figure 15 Algorithm for analysing String expressions

Process Access Detection



Figure 16 Access token information of the current process

ECHO

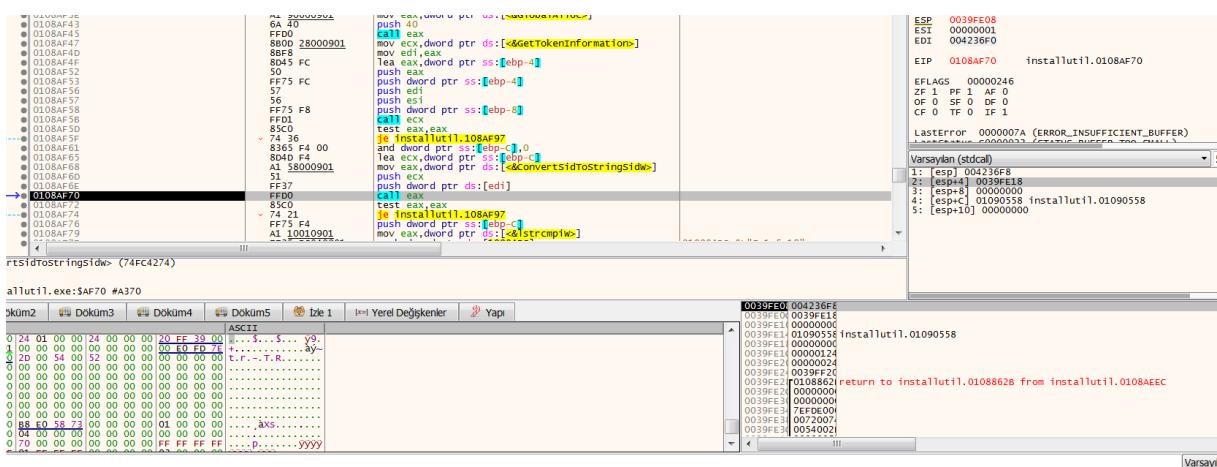


Figure 17 Retrieval of SID information for comparison

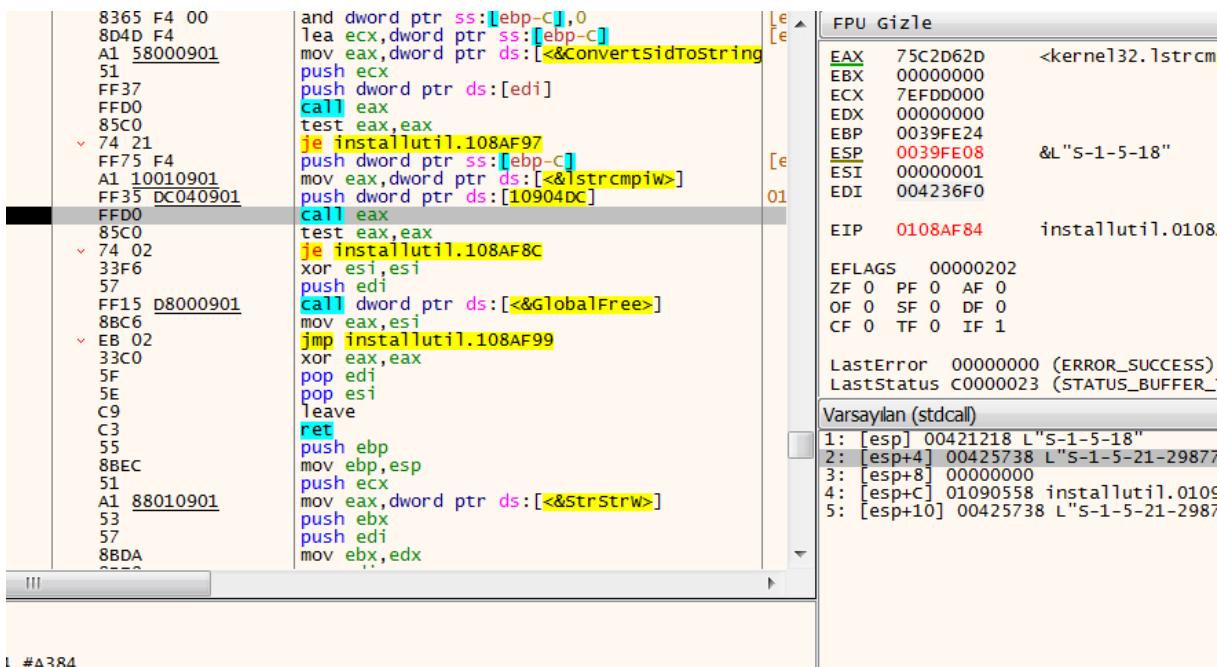


Figure 18 Admin authorisation comparison with SID information

It has been determined that the malware checks whether it has Admin authorisation. If it does not have Admin authorisation, it copies the Access Token of explorer.exe and restarts itself.

The Duplication Algorithm of Access Token of explorer.exe

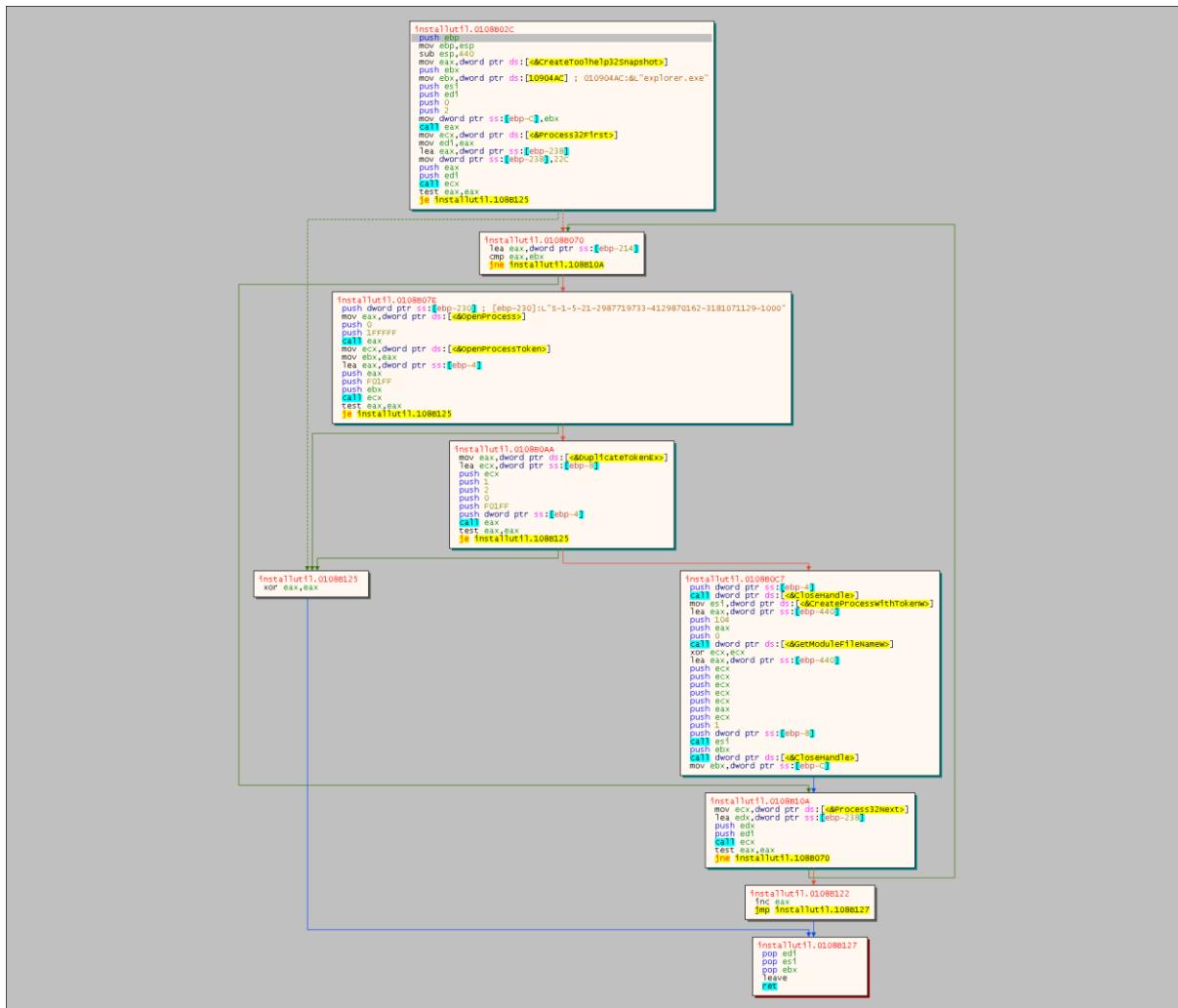


Figure 19 Algorithm to be applied if the process detects that it does not have admin authorisation

ECHO

Creation of Request Contents

```

installutil!1.0108840C
push ebp
mov esp,ebp
sub esp,C
mov eax,dword ptr ds:[<&LocalAlloc>]
push esi
push edi
push ed1
push 40
push 0
push 0
mov ecx,dword ptr ds:[0x09009B]
mov eax,esi
lea eax,dword ptr ss:[ebp-4]
mov dword ptr ss:[ebp-4]
push 20119
push 0
push dword ptr ds:[0x090344] ; 01090344:dL"SOFTWARE\Microsoft\Cryptography"
mov dword ptr ss:[ebp-4]
push 0
push 0
call ec
mov dword ptr ds:[<>GetQueryValueExW]
lea eax,dword ptr ss:[ebp-4]
push 0
push edi
lea eax,dword ptr ss:[ebp-4]
push 0
push 0
push dword ptr ds:[0x0902A4] ; 010902A4:dL"MachineGuid"
push dword ptr ss:[ebp-4]
call GetMachineGuid
test esi,esi
int installutil!1.01088546

installutil!1.01088542
test eax,eax
je installutil!1.0108854F

installutil!1.01088548
push dword ptr ss:[ebp-4]
call dword ptr ss:[<>CloseKey]
call dword ptr ss:[<>CloseKey]

installutil!1.0108854F
pop eax
pop edi
pop esi
leave
ret

```

Figure 20 Obtaining Machine GUID information

01088555 SS	push ebp	
01088556 88EC	mov ebp,esp	
01088559 51	push	
01088559 A1 44000901	mov eax,dword ptr ds:[<>LocalAlloc]	
0108855E 56	push esi	esi:L"_____"
0108855F 60 08020000	push	
01088564 6A 40	push 40	
01088566 C745 FC 01010000	call eax	
0108856D FF00	mov dword ptr ss:[ebp-4],101	
0108856E 8000 0000	lea eax,eax	esi:L"_____"
01088571 8045 FC	lea eax,dword ptr ss:[ebp-4]	esi:L"_____"
01088575 50	push eax	
01088576 FF15 78010901	call dword ptr ds:[<>GetUserNameW]	esi:L"_____"
0108857C 88C6	mov eax,esi	esi:L"_____"
0108857E 5E	pop esi	
0108857F C9	leave	
01088580 C3	ret	
01088581 55	push ebp	
01088582 88EC	mov ebp,esp	
01088584 51	push ecx	
01088585 A1 44000901	mov eax,dword ptr ds:[<>LocalAlloc]	
01088586 52	push ebx	
01088588 56	push edi	
0108858C 57	push edi	
0108858D BE 08020000	mov esi,1028	
0108858E 8000 0000	mov edx,edx	
01088594 56	push esi	
01088595 6A 40	push 40	
01088597 C3	ret	

FPU Gizle

EAX 00000001
EBX 00000000
ECX 00000000
EDX 00000003
EBP 0039FEC
ESP 0039FEC
ESI 00437648 L"266792b0-6193-4b67-adfd-f1404573300a"
EDI 00437438 EIP 0108857C installutil.0108857C
EFLAGS 00000244
ZF 1 PF 1 AF 0
OF 0 SF 0 DE 0
CF 0 TF 0 IF 1
LastError 00000000 (ERROR_SUCCESS)

Varsayılan (stdcall)

1: [esp+4] 00000006
2: [esp+8] 0039FEC
3: [esp+12] 0039FEC
4: [esp+10] 00000000
5: [esp+14] 00000000

Figure 21 Pulling username information to generate victim ID information

It has been detected that it tries to create a uniq machine ID with MachineGuID and username. The malware will use this machineID and the configID variables it analyses in the http request it will send in the future. The created machineID is as follows:

machineId= <MachineGuID> |<username>&configId=2fed969fd165f32f9d3d5171853ebef5

ECHO

Network Analysis

Request Analysis

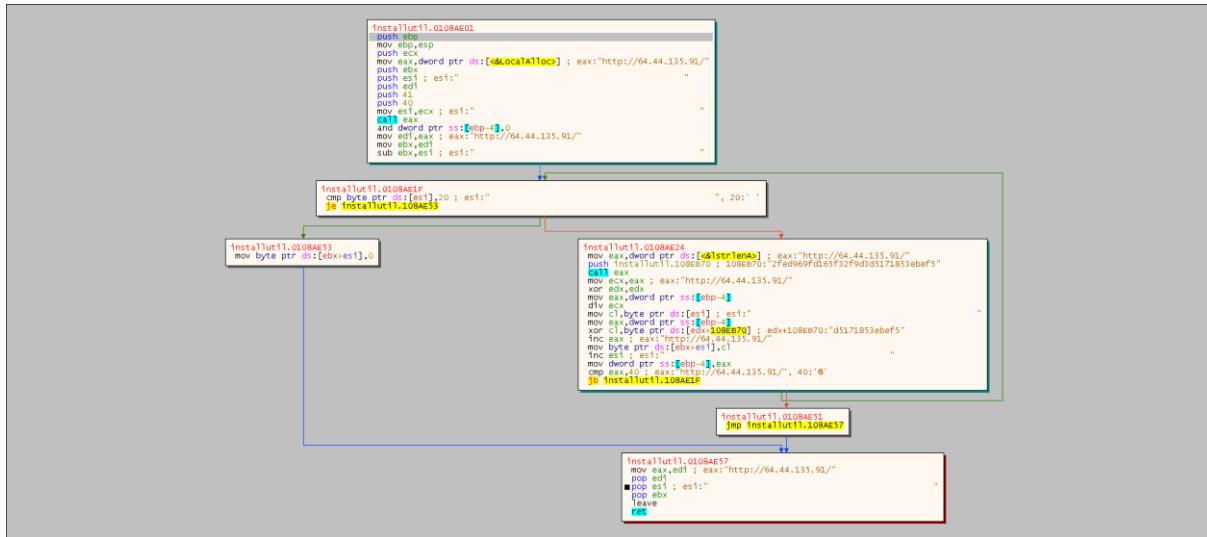


Figure 22 IP resolving

As a result of the analysis, it was determined that the IP "<http://64.44.135.91/>" appeared.

Request İçeriği

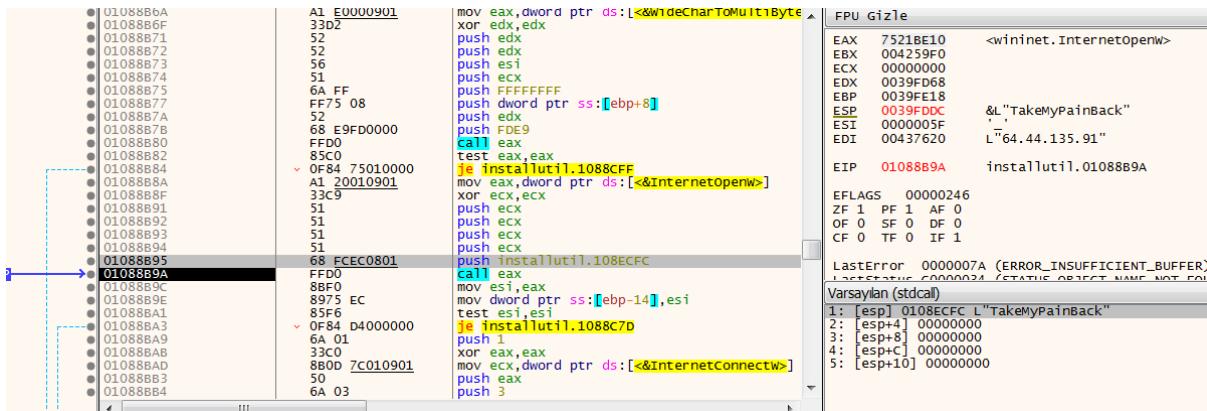


Figure 23 Detection of Agent information used in C2 communication

ECHO

Figure 24 shows the assembly code for the installutil.exe process. The code is comparing memory at address 0F84 D4000000 with the value 0F84 97000000. It then calls ECX (01088BD0) which leads to another comparison with 0F84 1088C76. The assembly code includes various pushes, moves, and comparisons. Registers and stack frames are visible on the right.

Figure 24

Figure 25 Determining the method of the request to be sent. The assembly code for net.HttpOpenRequestW is shown, comparing memory at address 01088CD0 with the value 01090284. It then calls ECX (01088CD0) which leads to another comparison with 01090284. The assembly code includes various pushes, moves, and comparisons. Registers and stack frames are visible on the right.

Figure 25 Determining the method of the request to be sent

Figure 26 shows the assembly code for net.HttpOpenRequestW, comparing memory at address 01088C39 with the value 01090284. It then calls ECX (01088C39) which leads to another comparison with 01090284. The assembly code includes various pushes, moves, and comparisons. Registers and stack frames are visible on the right.

Figure 26

Figure 27 Sending the edited request to the C2 server. The assembly code for net.HttpOpenRequestW is shown, comparing memory at address 01088C39 with the value 01090284. It then calls ECX (01088C39) which leads to another comparison with 01090284. The assembly code includes various pushes, moves, and comparisons. Registers and stack frames are visible on the right.

Figure 27 Sending the edited request to the C2 server

It has been determined that the agent information is "TakeMyPainBack" and the Request method is POST.

Http request content:

- Content-Type: application/x-www-form-urlencoded; charset=utf-8\r\n\r\n\r\n\r\n\r\n
- machineId=2e6792b0-6193-4b67-adfd-f1404573300a|zorro&configId=2fed969fd165f32f9d3d5171853ebef5

Expected Variables in Response

token
wlts_
grbr_
tlgrm_
dr_

Table 7

When the behaviour of malware after the response was examined, it was determined that it tried to use some data from the responses. The detected data variables are shown in Table 7.

After Response

Gathering Device Information

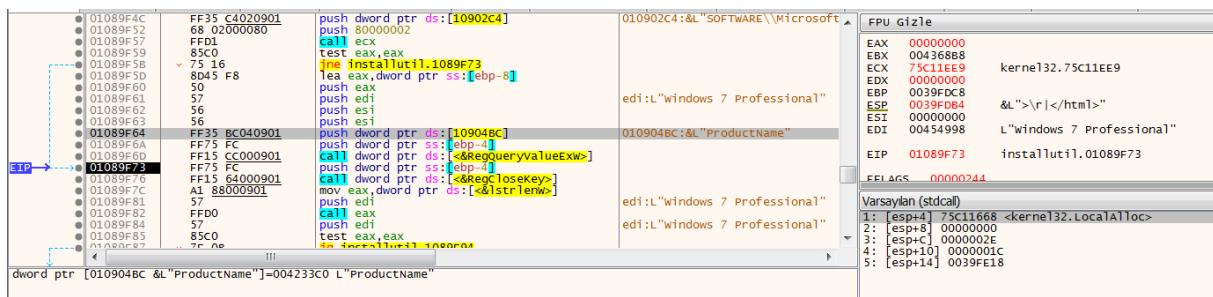


Figure 28 Detection of collecting information through the Registry

It has been found that the malware collects some information from the victim device. This information includes:

Locale	Time Zone	OS	Architecture	System Informations
Memory Information	Display Size	Display Devices	Application Information	

Table 1 3 Registry üzerinden aldığı bazı bilgiler

It has been found that malware collects this information in the following format:

- Locale: %s \n\t- Time zone: %c%ld minutes from GMT \n\t- OS: %s\n\t- CPU: %s (%d cores)) - \n\t- Architecture: x%d\n\t- RAM: %d MB \n\t- Display size: %dx%d\n\t- Display Devices: %s

It generates a random file name for the generated collection of information and is transmitted to the server with the POST method along with the file name.

Request içeriği: "Content-Type: multipart/form-data; boundary=bF0xB2TEnUcQ7DfR\r\n\r\n\r\n\r\n"

bF0xB2TEnUcQ7DfR = <verilerin toplandığı dosyanın adı>

Figure 29 Sending the collected information back to the C2 server with the token information returned by the server

It was found that the token information received from the server was also used in sending requests to the server.

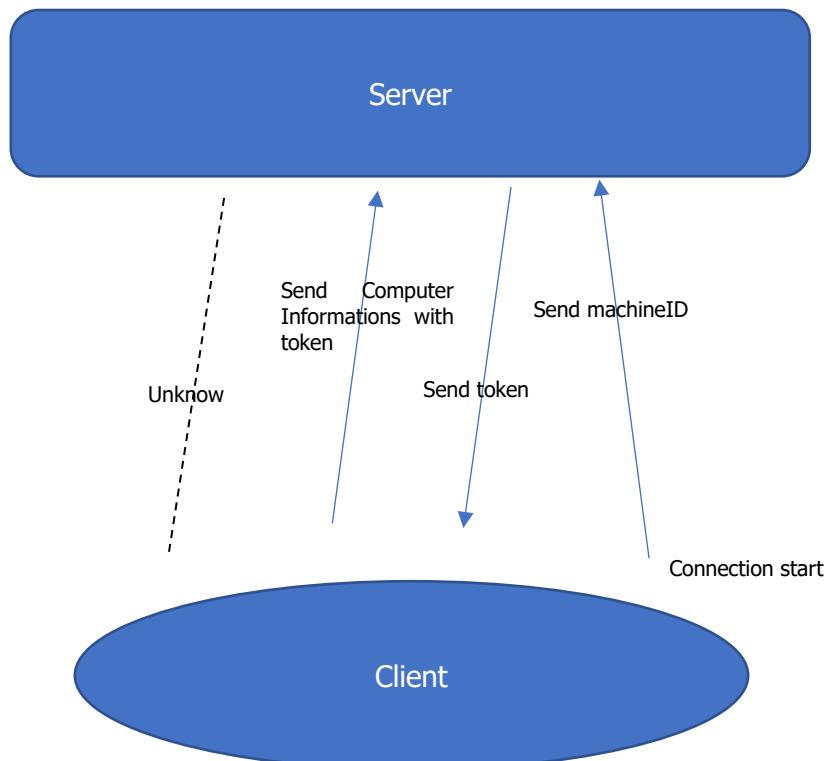


Figure 30 C2 Server and victim device communication

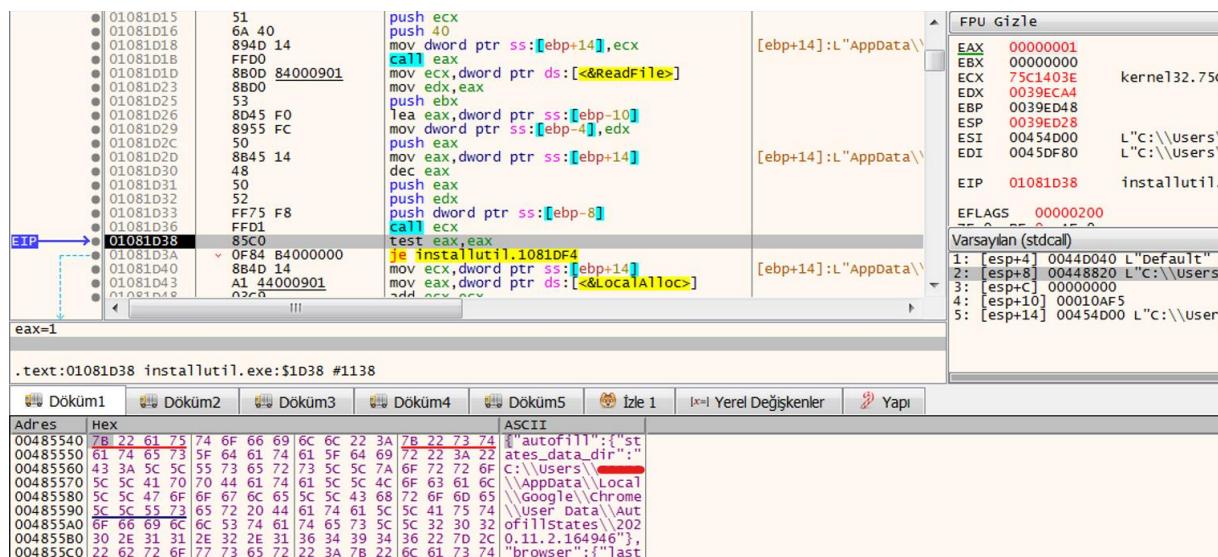


Figure 31

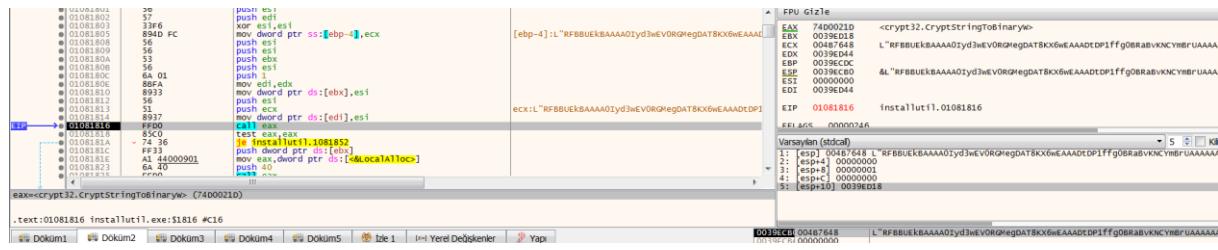


Figure 32 Detection of encrypted_key information encrypted with base64 algorithm and used as byte

It was found that the malware extracted the information in the "**\"AppData\\Local\\Chrome\\User Data\\Default\\Login"** file and the "**\"AppData\\Local\\Google\\Chrome\\User Data\\AutofillStates\\"** folder.

It was observed that encrypted_key data was encrypted with base64 encryption algorithm.

ECHO

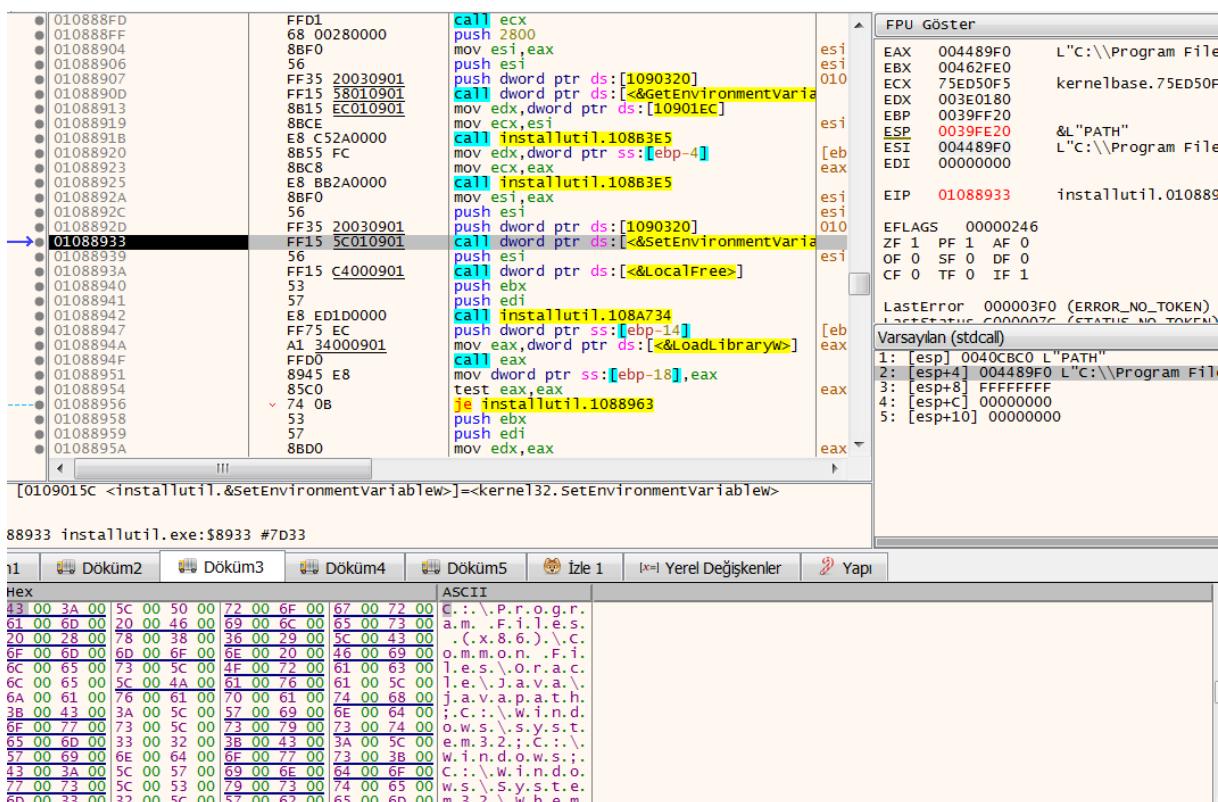


Figure 33 Change detection on environment variables

It was determined that malware made additions on the PATH variable among the environment variables.

```
C:\Program Files (x86)\Common
Files\Oracle\Java\javapath;C:\Wi
ndows\system32;C:\Windows;C:\
Windows\System32\Wbem;C:\Win
dows\System32\WindowsPowerShe
ll\v1.0\;C:\Users\<user>
\LocalAppData\Local\Programs\Python
\Python37\Scripts\;C:\Users\<us
er>\AppData\Local\Programs\Pyt
hon\Python37\;C:\Users\<user>\A
ppData\LocalLow
```

ECHO

DLL Loading

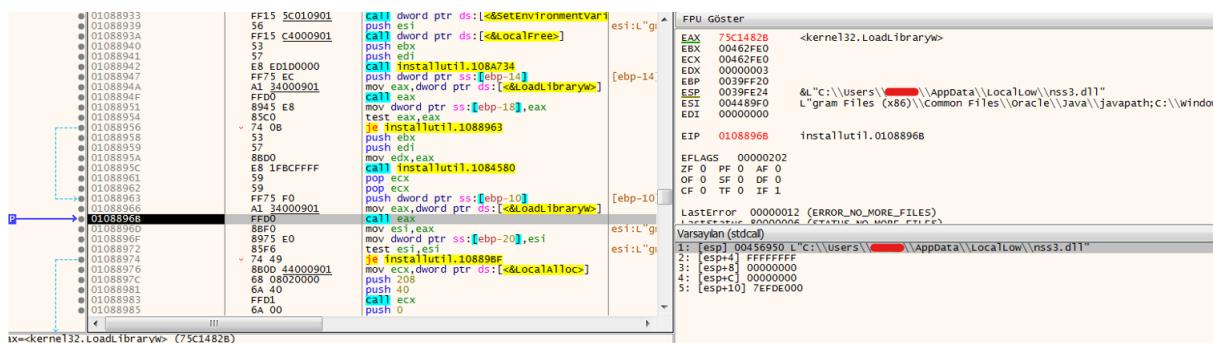


Figure 34 nss3.dll installation process

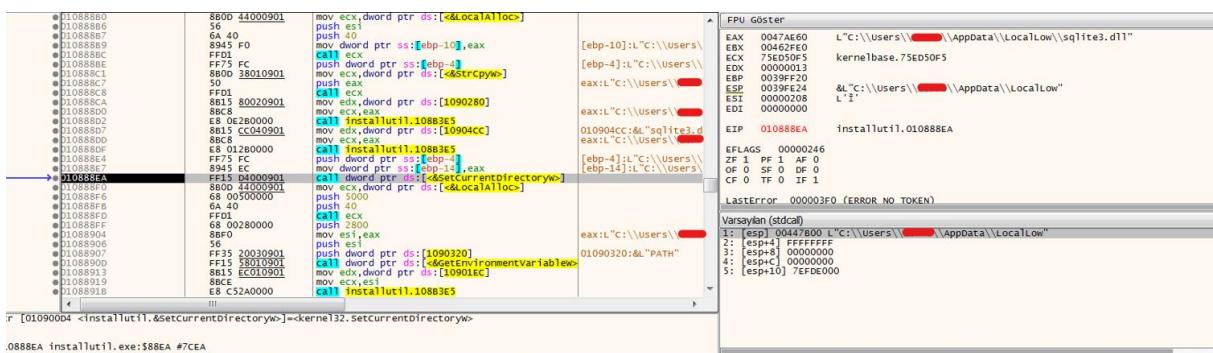


Figure 35 sqlite3.dll installation process

It has been detected that the malware installs sqlite3.dll and nss3.dll created in the LocalLow directory.

Database Operations

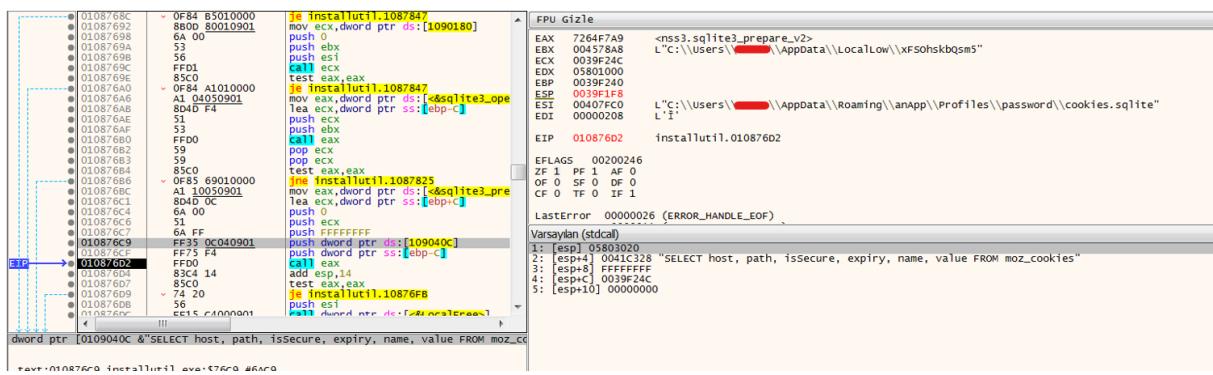


Figure 36

It has been detected that the malware creates a database file with a random name in the AppData\LocalLow\ directory. It has been determined that it pulls data from the following files with SQL queries to the created database.

coocies.sqlite
formhistory.sqlite
password.txt
storage\default
wallet.dat
logins.json

Table 8 Some targeted file and directory names

File Traversal Algorithm

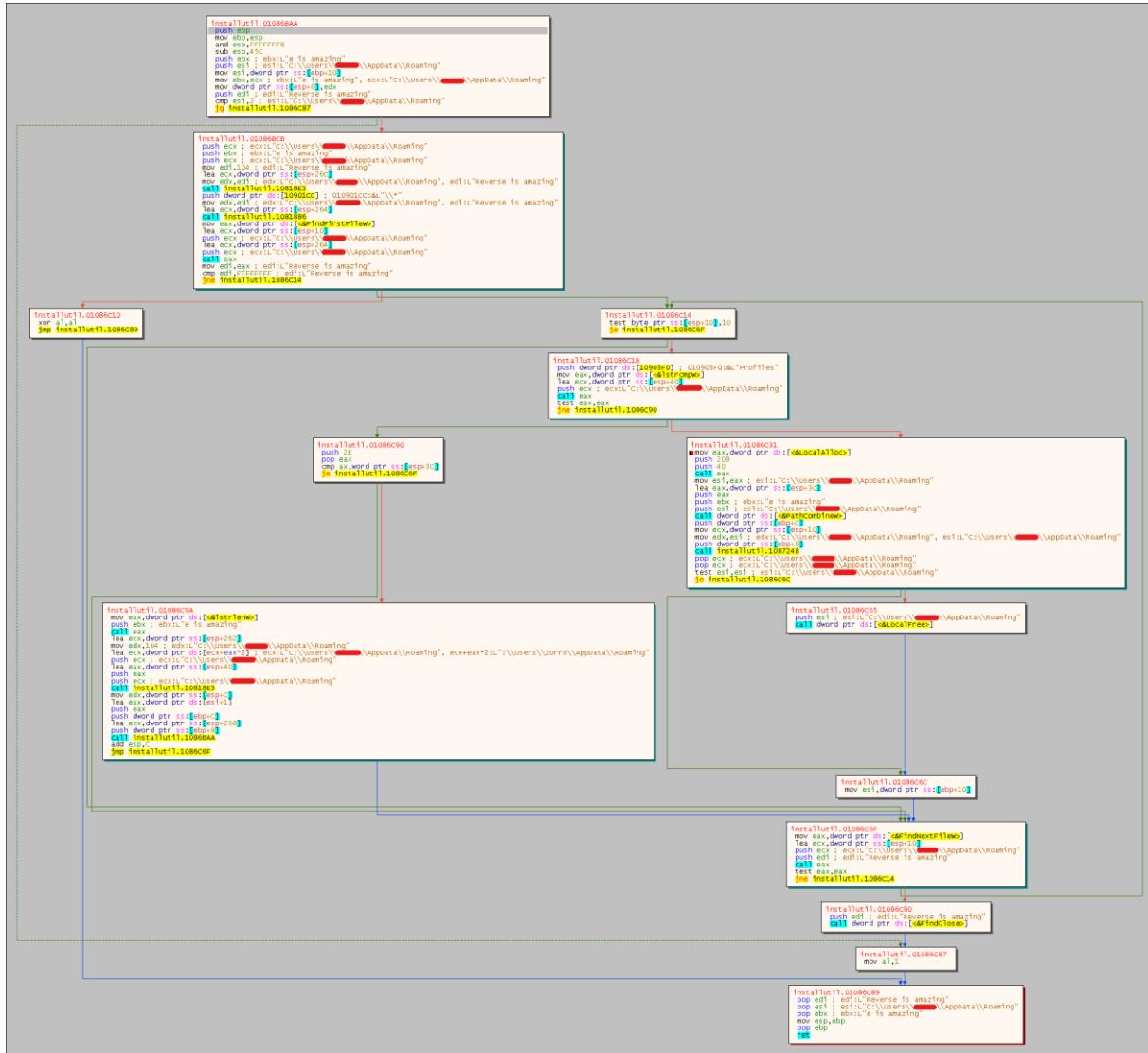


Figure 37 Index scanning algorithm

It was observed that it scanned the directory until it found the Profiles and User Data folder.

ECHO

Additional Analysis

It was found that malware takes a ScreenShot and saves it as a recording.

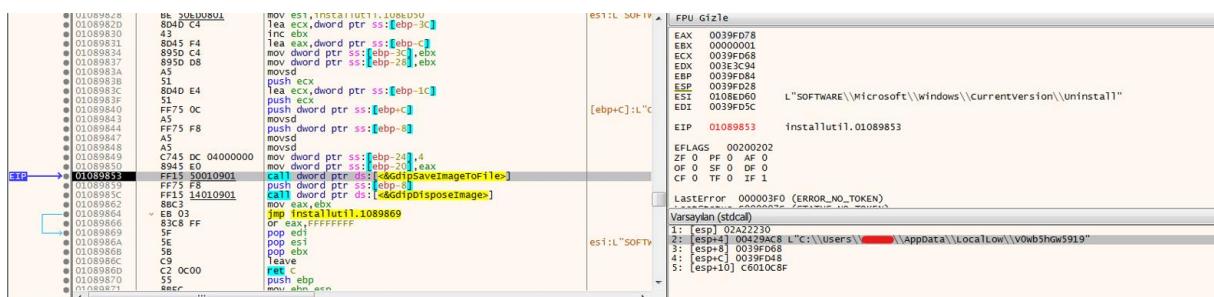


Figure 38 Saving the screen photo taken by Malware

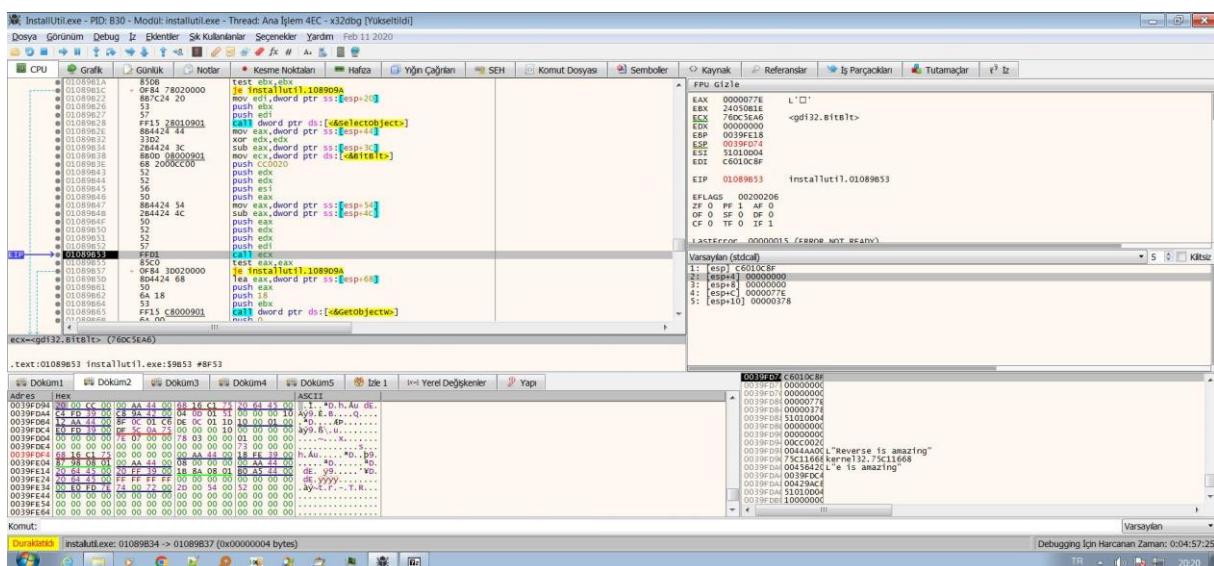


Figure 39 Screen Photo taken by Malware

SQL Queries

SELECT origin_url, username_value, password_value FROM logins
SELECT origin_url, username_value, password_value FROM logins
SELECT host_key, path, is_secure , expires_utc, name, encrypted_value FROM cookies
SELECT name, value FROM autofill
SELECT host, path, isSecure, expiry, name, value FROM moz_cookies
SELECT fieldname, value FROM moz_formhistory
SELECT name_on_card, card_number_encrypted, expiration_month, expiration_year FROM credit_cards

Table 9 Some of the enquiries made

YARA Rule

```

rule Rule_InstallUtil
{
meta:
    author = "Bilal BAKARTEPE (EchoCTI Team)"
    site = "https://github.com/bixploit"
    description = "RaccoonStealler v2.0 second stage PE file"
    hash= "d69ee30203430d1404a2890268bb04e9"
strings:
    $sql1 = "SELECT origin_url, username_value, password_value FROM logins"
    $sql2 = "SELECT host_key, path, is_secure , expires_utc, name, encrypted_value FROM cookies"
    $sql3 = "SELECT name, value FROM autofill"
    $sql4 = "SELECT host, path, isSecure, expiry, name, value FROM moz_cookies"
    $sql5 = "SELECT fieldname, value FROM moz_formhistory"
    $sql6 = "SELECT name_on_card, card_number_encrypted, expiration_month, expiration_year FROM credit_cards"

    $dir_name1 = "profiles"
    $dir_name2 = "712006f6e7da2882" //User Data
    $dir_name3 = "Default"
    $dir_name4 = "Login Data"
    $file_name1= "password.txt"
    $file_name2= "cookies.sqlite"
    $file_name3= "Cookies"

    $agent="TakeMyPainBack"

    $ip_clear="http://64.44.135.91"
    $ip_enc="d5171853ebef5"

    $enc_str1="aa0bb6f89e4fc28e"
    $enc_str2="ba0c5f9d6a984fdd" //encrypted_values
    $enc_str3="587a51bde849292f"
    $enc_str4="ca82e1c9d5793376"
    $configurationID="2fed969fd165f32f9d3d5171853ebef5"
    $mutex_name="264782971_qJ5tS2bD5fD1nZ5kD2kV"

    $respons_variable1="320dd64ff20444fa" //tlgrm
    $respons_variable2="d286b66a2753e1b1" //wlts
    $respons_variable3="bee04f3449ba713e" //sstmnfo
    $respons_variable4="6ef9561122a8649a" //token
    $respons_variable5="877e12dc4d066d8c" //nss3.dll
    $respons_variable6="274f2fd9bfa77a7c" //sqlite.dll

    $opc1 = {53 56 57 6A 41 6A 40 8B F1 FF D0 83 65 FC 00 8B F8 8B DF 2B DE 80 3E 20 74 2F A1
             94 01 41 00 68 70 EB 40 00 FF D0 8B C8 33 D2 8B 45 FC F7 F1 8A 0E 8B 45 FC 32 8A 70 EB 40
             00 40 88 0C 33 46 89 45 FC 83 F8 40 72 CE}// allocation and deobfuscation
    $opc2 = {55 8B EC 51 53 56 57 8B 3D 88 00 41 00 8B DA 53 89 4D FC FF D7 FF 75 FC 8B F0 FF D7
             8B 0D 44 00 41 00 8D B8 80 00 00 00 03 FE 8D
             04 3F 50 6A 40 FF D1 FF 75 FC 8B F0 8B D7 8B CE E8 34 64 FF FF 53 8B D7 8B CE E8 57 64 FF
             FF FF 75 FC FF 15 D8 00 41 00 5F 8B C6 5E 5B C9 C3}//deobfuscation and ascii to unicode transition

condition:
    (any of ($opc*)) and (2 of ($sql*, $dir_name*, $file_name*, $enc_str*, $respons_variable*) or any
    of ($ip_clear, $ip_enc, $agent, $configurationID, $mutex_name))
}

```

MITRE ATTACK TABLE

Reconnaissance	Execution	Discovery	Collection	Defense Evasion	Credential Access	Command and Control	Exfiltration
T1592 Gather Victim Host Information: <u>Hardware</u>	T1559 Inter-Process Communication: <u>Component Object Model</u>	T1012 Query Registry	T1005 Data from Local System	T1070 Indicator Removal on Host: File Deletion	T1539 Steal Web Session Cookie	T1071 Application Layer Protocol: <u>Web Protocols</u>	T1041 Exfiltration Over C2 Channel
T1589 Gather Victim Identity Information: <u>Credentials</u>		T1082 System Information Discovery	T1113 Screen Capture	T1140 Deobfuscate/Decode Files or Information		T1105 Ingress Tool Transfer	T1020 Automated Exfiltration
T1592 Gather Victim Host Information: <u>Software</u>		T1614 System Location Discovery: <u>System Language Discovery</u>					

Mitigations

1. Emails and senders should be carefully checked to ensure they are genuine before opening any attachments.
2. Do not download any resources from unsafe websites.
3. Use a reliable, high quality and always updated antivirus software.
4. Keep your operating system and applications up to date with the latest security patches.
5. End user training is vital for your organisation, make sure you inform your employees about the dos and don'ts of online security.



ECHO

CYBER THREAT INTELLIGENCE