

ECHO

CYBER THREAT INTELLIGENCE



LOCKBIT 3.0

İçindekiler

Giriş	2
Attack Chain.....	3
Teknik Analiz.....	3
Payload.bin Analizi.....	3
Kurallar.....	23
YARA.....	23
SIGMA – 1	24
SIGMA – 2	25
MITRE ATT&CK Tablosu	26

Giriş

LockBit 3.0, ilk ortaya çıktığı 2019 yılından beri fidye yazılımı ailelerinin arasında son derece tehlikeli bir üye olmuştur. Bu nedenle, Dünya genelinde birçok kuruluş için ciddi bir siber güvenlik tehdidi oluşturmaktadır. LockBit, kurban sistemlerdeki verileri şifreleyerek çalışmakta ve ardından verilerin çözülmesi karşılığında fidye talep etmektedir. Ancak LockBit 3.0, sadece verileri şifrelemekle kalmamakta, aynı zamanda çevrimiçi olarak bu verilerin yayınlanması tehdidi ile kurbanları zorlamaktadır, bu da organizasyonların itibarını ve güvenilirliğini zedelemektedir. LockBit 3.0, kurbanın sistemlerine dağıtıldığında oldukça gelişmiş şifreleme algoritmaları kullanmaktadır. Bu durum, verileri şifrelerinin kırılmasını son derece zorlaştırmakta ve kurbanları fidye ödemeye zorlamaktadır. Fidyeye, genellikle kripto para birimleriyle ödenmekte, dolayısıyla ödenen fidyenin izlenmesi imkânsız olabilmektedir.

LockBit 3.0, dünya genelinde birçok ülkeyi hedef almaktadır. Ancak bazı ülkeler, bu kötü niyetli fidye yazılımının daha fazla etkin olduğu veya yoğun bir şekilde hedeflendiği ülkeler olarak öne çıkmaktadır (CISA,2023). Söz konusu ülkeler:

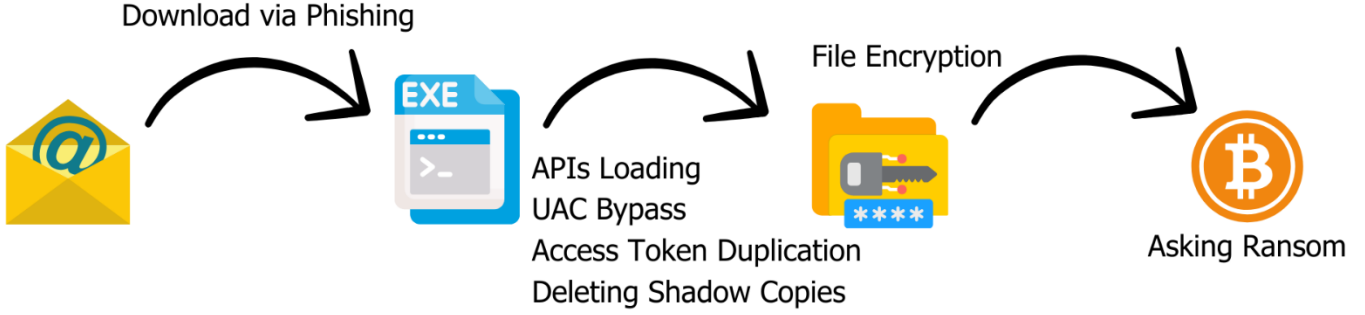
- Rusya: LockBit 3.0, Rusya'daki kuruluşları sık sık hedef alır. Bu, Rusya'daki büyük ve küçük işletmeleri, devlet kurumlarını ve bireyleri etkileyebilir.
- Amerika Birleşik Devletleri: ABD, dünyanın en büyük ekonomilerinden biri olduğu için LockBit 3.0 için cazip bir hedef olabilir. Finans, sağlık, üretim ve teknoloji sektörleri ABD'de sıklıkla hedeflenir.
- Kanada: Kanada, LockBit 3.0 tarafından hedeflenen diğer bir ülkedir. Kanada'daki çeşitli sektörler, bu fidye yazılımının etkilerini hissedebilir.
- Birleşik Krallık: Birleşik Krallık, Avrupa'nın en büyük ekonomilerinden birine sahiptir ve bu nedenle LockBit 3.0 için cazip bir hedeftir. Finans ve sağlık sektörleri sıklıkla hedeflenir.
- Almanya: Almanya, teknoloji, üretim ve diğer sektörlerdeki kuruluşlarıyla LockBit 3.0 tarafından sıkça hedef alınır.

LockBit 3.0, farklı sektörlerde faaliyet gösteren kuruluşları hedef alır ve bu sektörlerin birçoğu fidye yazılımının etkilerini deneyimlemektedirler (BleepingComputer,2023). Hedef alınan sektörler:

- Sağlık Sektörü: Sağlık kuruluşları, hassas hasta bilgilerini sakladıkları için fidye yazılımlarının sıkça hedefi olurlar. LockBit 3.0, hastane, klinik ve sağlık sigortası şirketlerini hedef alır.
- Finans Sektörü: LockBit 3.0, bankalar, finansal danışmanlık şirketleri ve finansal kuruluşları hedef alarak finans sektöründe büyük zararlara neden olabilir.
- Üretim Sektörü: Üretim tesisleri ve endüstriyel işletmeler, üretim süreçleri ve tedarik zinciri yönetimi açısından kritik öneme sahiptir. LockBit 3.0, üretim sektörünü hedef alarak üretim aksamalarına yol açabilir.
- Teknoloji Sektörü: Teknoloji şirketleri, müşteri bilgilerini ve fikri mülkiyeti sakladıkları için LockBit 3.0'ın hedefi olabilir. Bu, teknoloji şirketlerinin itibarını ve rekabet gücünü ciddi şekilde etkileyebilir.
- Diğer Sektörler: LockBit 3.0 ayrıca eğitim, perakende, enerji ve diğer birçok sektörde faaliyet gösteren kuruluşları hedef alabilir.

Bu tehditlerin tümü, organizasyonlar için LockBit 3.0'ı ciddi bir siber güvenlik tehdidi haline getirmektedir. Kuruluşların, güçlü güvenlik önlemleri almaları ve fidye yazılımlarına karşı savunma stratejileri geliştirmeleri kritik öneme sahiptir. Bu rapor, LockBit 3.0'ın ayrıntılı analizi ile organizasyonlara, bu tehdide karşı nasıl korunabilecekleri konusunda önemli bilgiler sunmaktadır. İlgili güvenlik önlemlerini almak, organizasyonların verilerini ve itibarlarını korumak için kritik bir adımdır.

Attack Chain



Şekil 1 Attack Chain

Teknik Analiz

Payload.bin Analizi

MD5	bbe63d8efc8d8dc7f387b08ee07721ba
SHA256	2e8aaa6338cbf95d8d268559fb8afac64e1c0dfc9ded4bb2de63a9db634e354d
File Type	PE32/EXE

Şekil 2 General File Information

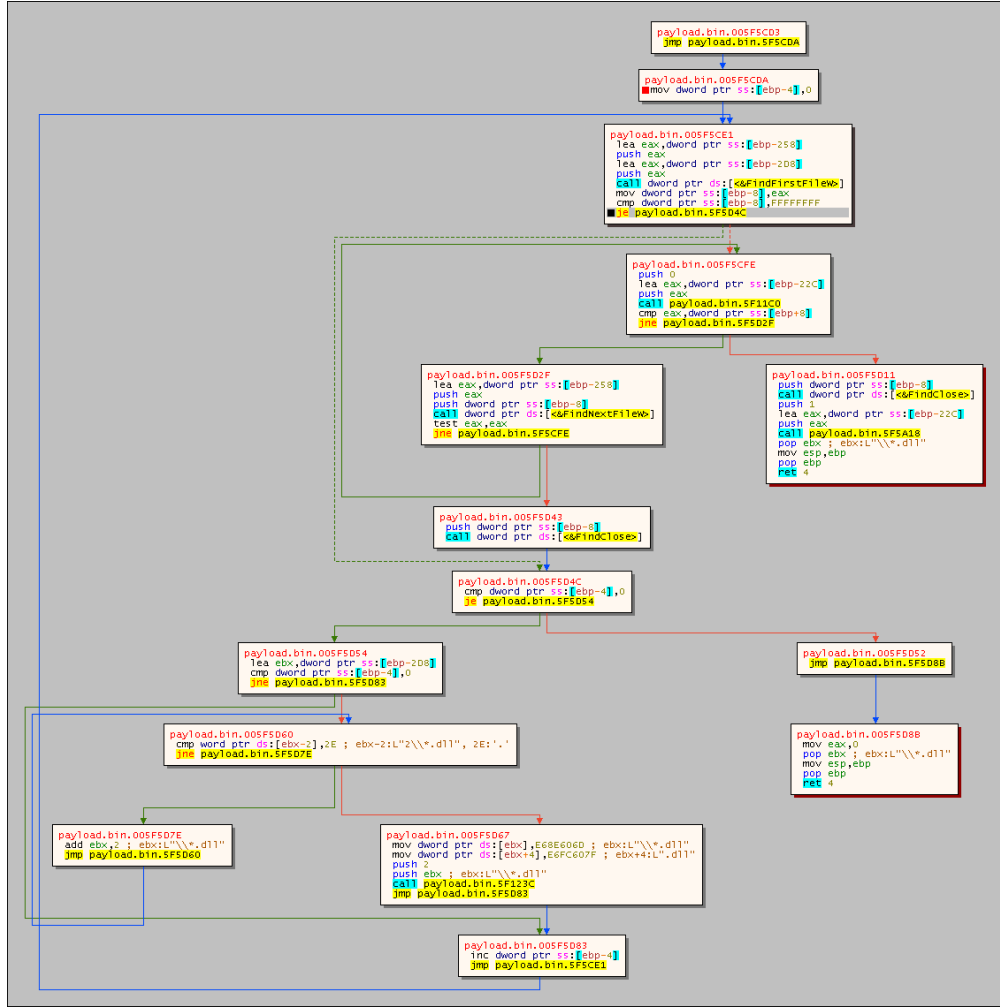
```
lea eax, dword ptr ss:[ebp-258]
push eax
lea eax, dword ptr ss:[ebp-2D8]
push eax
call dword ptr ds:[<&FindFirstFile>]
mov dword ptr ss:[ebp-8], eax
cmp dword ptr ss:[ebp-8], FFFFFFFF
je payload.bin.5F5D4C
push 0
```

eax: L"C:\\windows\\System32*.dll"

eax: L"C:\\windows\\System32*.dll"

Şekil 3 FindFirstFile: C:\\Windows\\System32*.dll

Zararlının "**System32**" dizini içerisindeki DLL dosyalarını sırayla gezdiği tespit edildi.

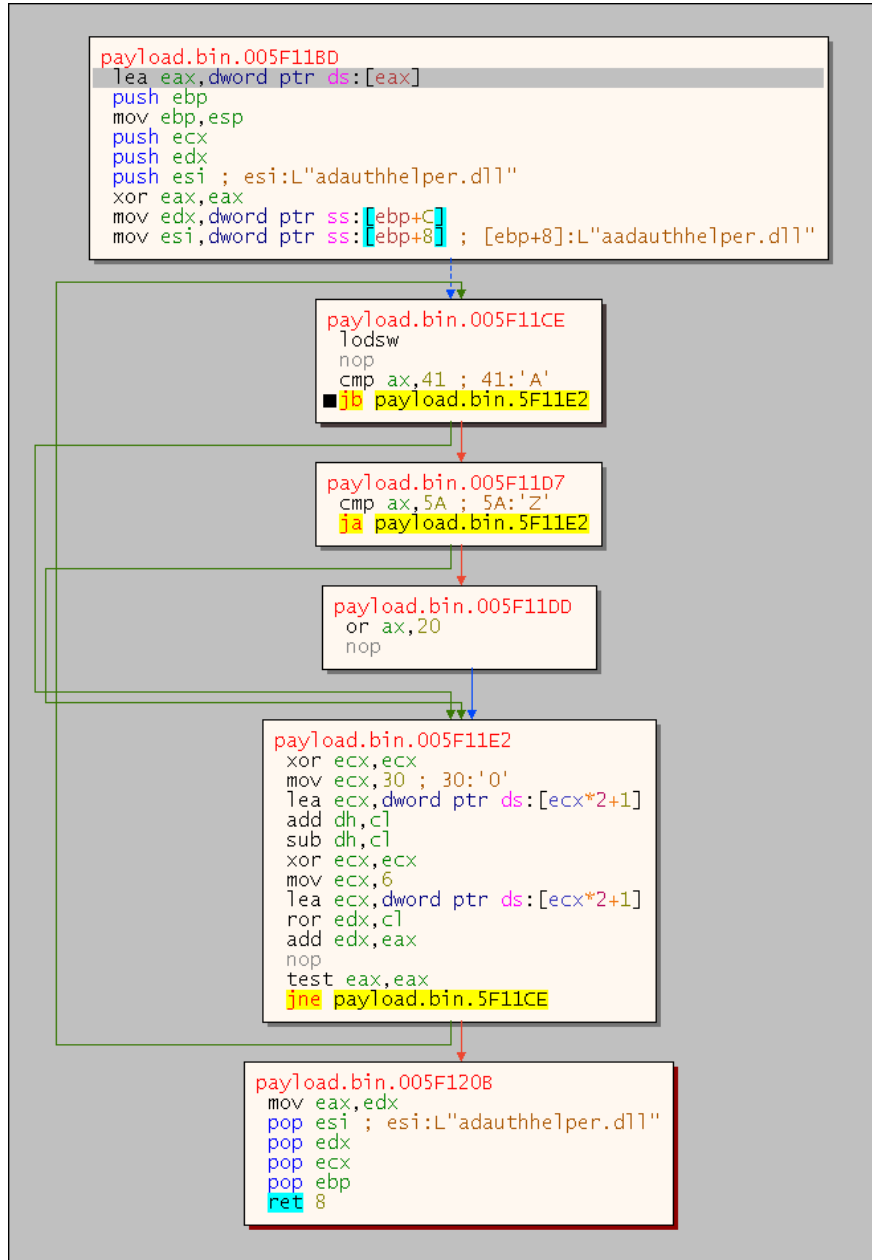


Şekil 4 Traversal Algorithm

00		push 0	
35	D4FDFFFF	lea eax,dword ptr ss:[ebp-22C]	
		push eax	
B4B4FFFF		call payload.bin.5F11C0	eax:L"aadauthhe]per.d11"
15 08		cmp eax,dword ptr ss:[ebp+8]	
1E		jne payload.bin.5F5D2F	
dress	Hex		
71FA9C	87 77 16 41		
71FAAC	DC FA 71 00		
71FABC	00 00 A0 01	ebp+8	

Şekil 5 Hash Generating and Comparing

Gezinme sürecinde yapılan bir işlem göze çarpmaktadır: DLL dosyalarına ait (hash) karma oluşturma. Zararlı API Hashing yöntemine benzer bir şekilde DLL adlarının karmalarını (hash) oluşturup, ardından karşılaştırarak istenilen dll dosyasına ulaşmaktadır.



Şekil 6 DLL Name Hashing Algorithm

Aranılan DLL dosyasına ait hash bilgisinin **"41 16 77 B7"** olduğu ve bunun **ntdll.dll** dosyasına ait olduğu tespit edilmiştir.


```

payload.bin.005F5A5D
lea eax,dword ptr ss:[ebp-4]
push eax
lea eax,dword ptr ss:[ebp-C]
push eax
push 0
push 0
call dword ptr ds:[<&LdrLoadDll>]
mov eax,dword ptr ss:[ebp-4]
pop edi
pop esi
pop ebx ; ebx:L"\\*.dll"
mov esp,ebp
pop ebp
ret 8

LastStatus C0000100 (STATUS_VARIABLE_NOT_FOUND)
GS 0028 FS 0053
ES 0028 DS 002B
CS 0023 SS 002B

ST(0) FFFF0000000076DC3870 x87r0 Special invalid
ST(1) FFFF00000000081FC7F66 x87r1 Special invalid
ST(2) 000000000000000000000000 x87r2 Zero 0.000000000000000000000000
ST(3) 000000000000000000000000 x87r3 Zero 0.000000000000000000000000
ST(4) 000000000000000000000000 x87r4 Zero 0.000000000000000000000000

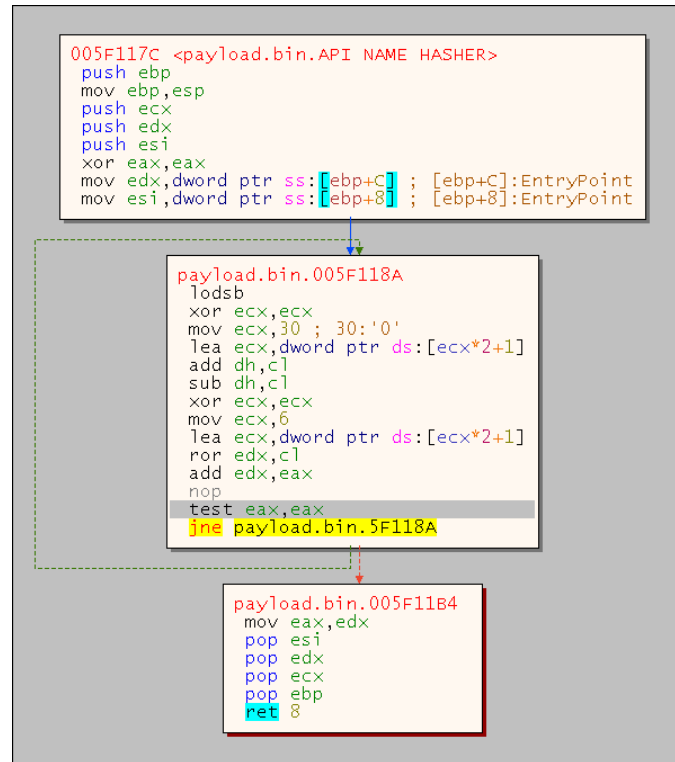
Default (stdcall)
1: [esp] 00000000 00000000
2: [esp+4] 00000000 00000000
3: [esp+8] 0071F79C 0071F79C
4: [esp+C] 0071F7A4 0071F7A4
5: [esp+10] 77405E70 <ntdll.RtlAllocateHeap> (77405E70)

```

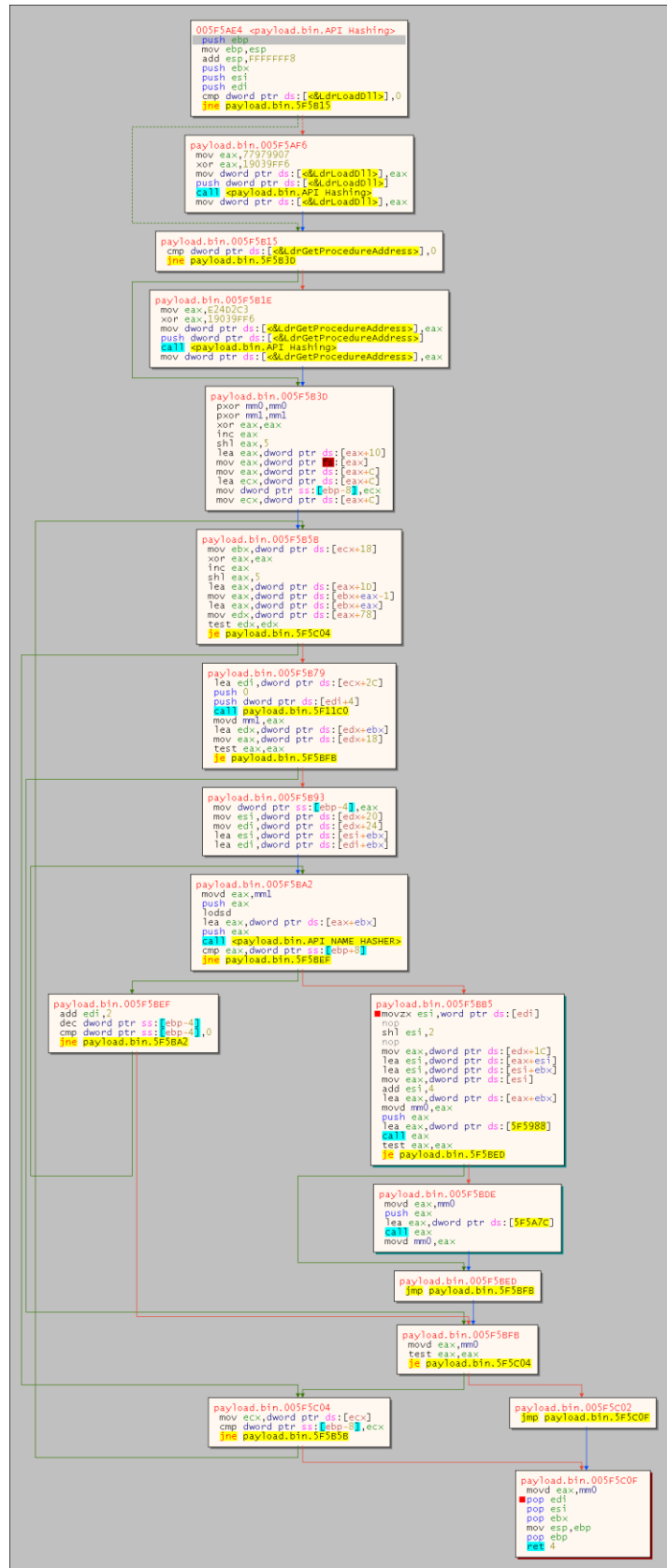
Şekil 7 LdrLoadDll

LdrLoadDll API'si ile dll dosyalarının yüklendiği görülmektedir. Zararlının **LoadLibrary** API'sini kullanmaktan kaçındığı görülmektedir.

Ayrıca, zararlının API Hashing tekniği kullandığı da tespit edilmiştir. Gelenksel API Hashing tekniğinden farklı olarak **LoadLibrary** ve **GetProcAddress** fonksiyonları yerine **LdrLoadDll** ve **LdrGetProcedureAddress** fonksiyonlarının kullanıldığı görülmektedir.

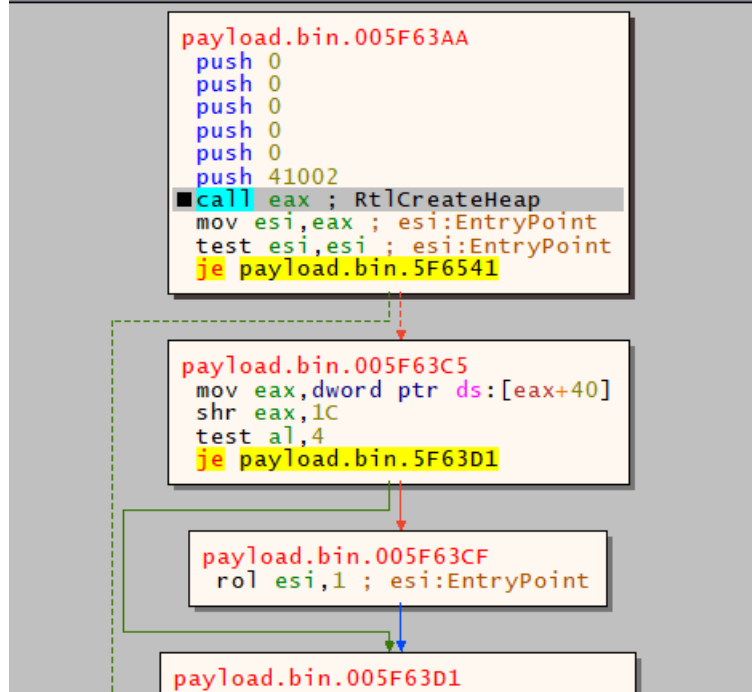


Şekil 8 API Name Hashing Algorithm

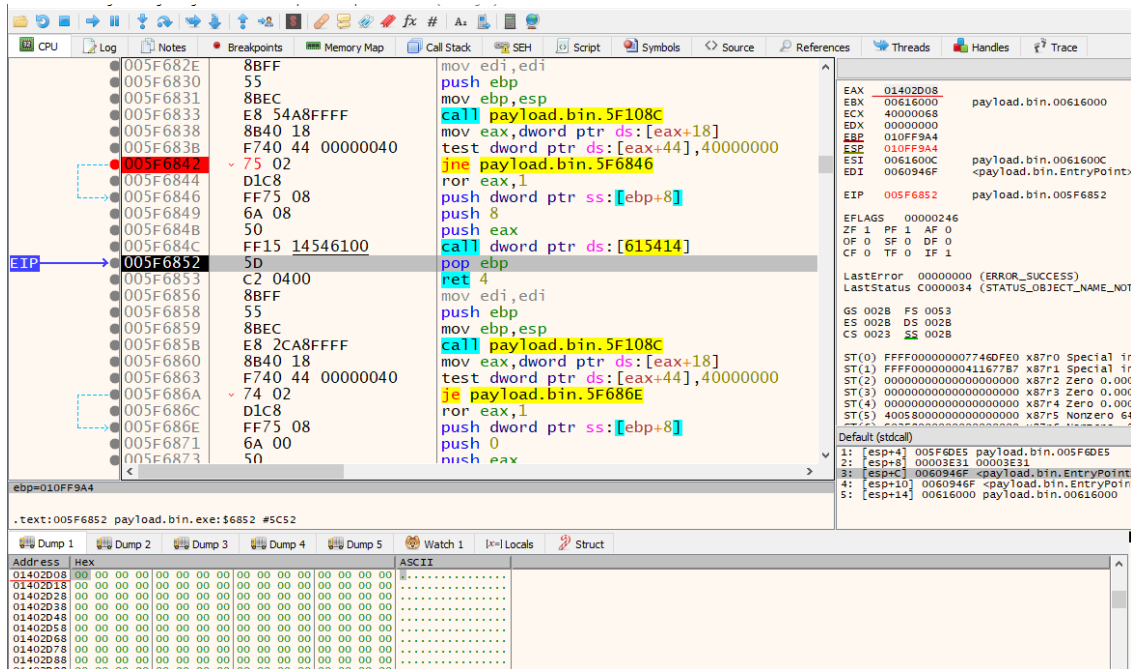


Şekil 9 API Hashing Algorithm

Zararlıının bazı anti-debug tekniklerinden yararlandığı da ayrıca tespit edilmiştir. Debug durumundaki bir heap yapısının normal durumdakinden farklarından yararlanarak bazı anti-debug teknikleri kullanmaktadır.



Şekil 10 Anti-Debug: Heap Based



Şekil 11 Anti-Debug: Heap Based

Bir diğer **heap based** anti-debug tekniği tespit edilmiştir. Şekil 11'de patched durumdaki kod bulunmaktadır.

```

mov eax,CDC8783
rol eax,7
xor eax,19039FF6
jmp eax
mov edx,ABABABAB
stosd
stosd
stosd
stosd
add byte ptr ds:[eax],al
add byte ptr ds:[eax],al
add byte ptr ds:[eax],al
add byte ptr ds:[eax],al
shl dword ptr ds:[edi+28E09A37],1
add byte ptr ds:[eax],bl
mov eax,83BA013F
rol eax,5
jmp eax
lodsd
mov edx,BAADF00D
stosd
stosd
stosd
stosd
stosd

```

Hide FPU

EAX 77405E70 <ntdll.RtlAllocateHeap>
EBX 00616000 payload.bin.00616000
ECX 013E0000
EDX 010FFA08
EBP 010FF9A4
ESP 010FF994 "Rh_"
ESI 0061600C payload.bin.0061600C
EDI 0060946F <payload.bin.EntryPoint>
EIP 03320625

EFLAGS 00000202
ZF 0 PF 0 AF 0
OF 0 SF 0 DF 0
CF 0 TF 0 IF 1

LastError 00000000 (ERROR_SUCCESS)
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)

GS 0028 FS 0053
ES 0028 DS 0028
CS 0023 SS 0028

ST(0) FFFF000000007746DFE0 x87r0 Special invalid
ST(1) FFFF00000000411677B7 x87r1 Special invalid
ST(2) 000000000000000000000000 x87r2 Zero 0.00000000000000000000
ST(3) 000000000000000000000000 x87r3 Zero 0.00000000000000000000
ST(4) 000000000000000000000000 x87r4 Zero 0.00000000000000000000

Default (stdcall)

1: [esp+4] 013E0000 013E0000
2: [esp+8] 00000008 00000008
3: [esp+C] 00003E31 00003E31
4: [esp+10] 010FF9BC 010FF9BC
5: [esp+14] 005F6DE5 payload.bin.005F6DE5

Şekil 12 RtlAllocateHeap: 15921 byte

Anti-debug teknikleri atlatıldıktan hemen sonra **15.921** byte alan ayrıldığı tespit edildi.

```

mov esi,dword ptr ss:[ebp+8]
push dword ptr ds:[esi-4]
call payload.bin.5F6830
mov ebx,eax
test ebx,ebx
je payload.bin.5F6DFB
mov ecx,dword ptr ds:[esi-4]
mov edx,ecx
mov edi,ebx
rep movsb
push edx
push ebx
call payload.bin.5F1718

```

esi-4:"1>"

esi-4:"1>"

Hide

EAX 01402D08
EBX 01402D08
ECX 00003E31
EDX 00003E31
EBP 010FF9BC
ESP 010FF980 <&EntryPoint>
ESI 0061600C payload.bin.0061600C
EDI 01402D08
EIP 005F6DF2 payload.bin.005F6DF2

EFLAGS 00000202
ZF 0 PF 0 AF 0
OF 0 SF 0 DF 0
CF 0 TF 0 IF 1

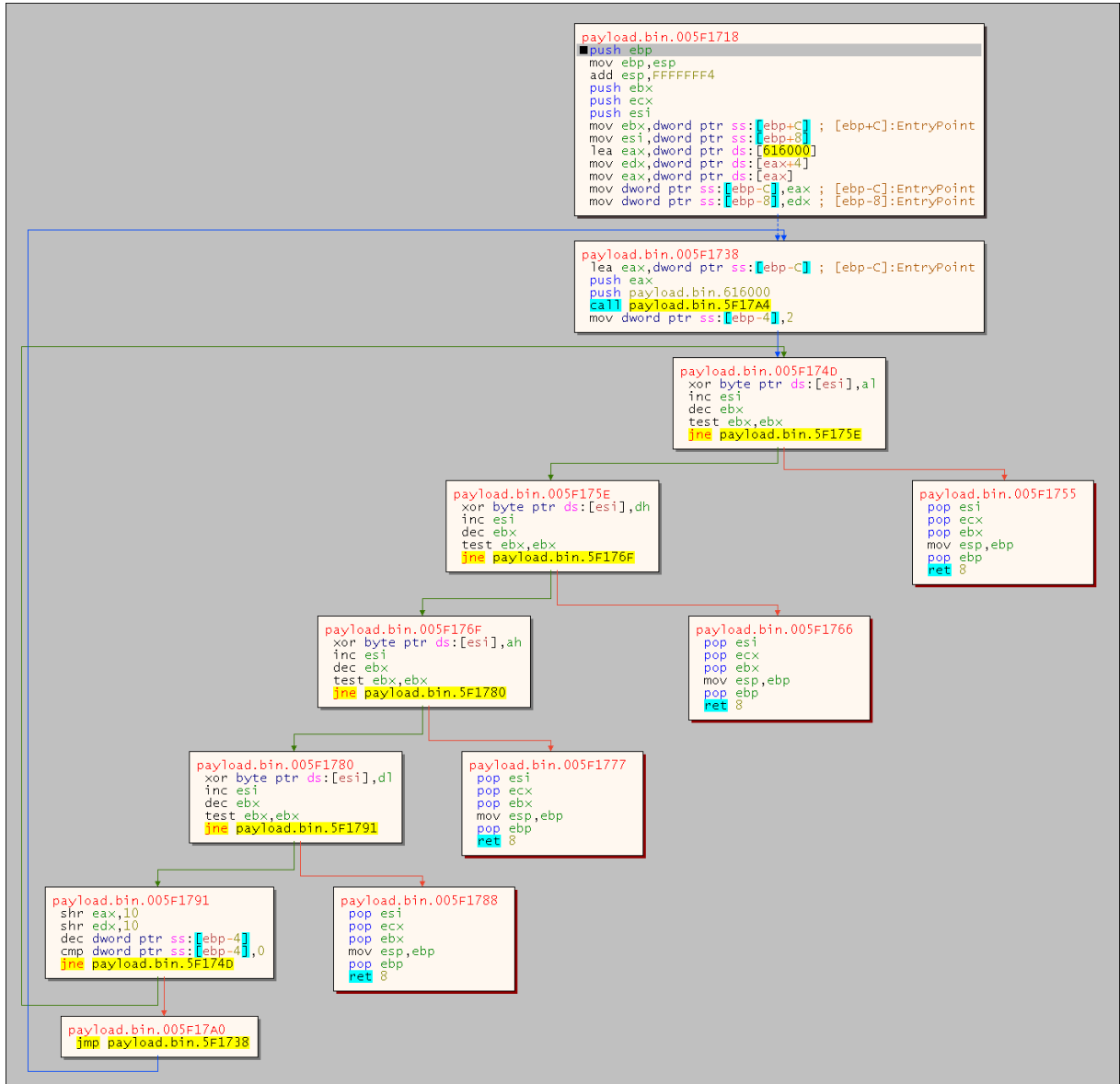
Şekil 13 Writing .pdata section

Ayrılan alana yazılacak veri takip edildiğinde, **.pdata** section bölümünün başlangıç adresi olduğu tespit edildi

Address	Size	Party	Info	Content	Type	Protection	Initial
005F0000	00001000	User	payload.bin.exe		IMG	-R---	ERWC-
005F1000	00018000	User	".text"	Executable code	IMG	ER---	ERWC-
00609000	00001000	User	".itext"		IMG	ER---	ERWC-
0060A000	00001000	User	".rdata"	Read-only initialized data	IMG	-R---	ERWC-
0060B000	00008000	User	".data"	Initialized data	IMG	-RWC-	ERWC-
00616000	00004000	User	".pdata"	Exception information	IMG	-RWC-	ERWC-
0061A000	00001000	User	".reloc"	Base relocations	IMG	-R---	ERWC-
0061B000	00001000	User			MAP	-P---	-P---

Şekil 14 .pdata section

Söz konusu section incelendiğinde şifrelenmiş olduğu görülmektedir.



Şekil 15 Decryption of .pdata

.pdata bölümünde şifreli halde bulunan verinin çözümlendiği algoritma Şekil 15'deki gibidir.

```
Hide FPU

EAX 77405E70 <ntdll.RtlAllocateHeap>
EBX 014071D5 "5iC+Sqf2j12ZvAKmNz aCV9Zt2EqxbJ9K+RuUYLqPzTrVrk3zW13PkJD0eRMEYwZBGxiSC
ECX 013E0000
EDX 81A05014
EBP 010FF9BC
ESP 010FF9AC "Rh_"
ESI 0060946F <payload.bin.EntryPoint>
EDI 01406C10

EIP 03320625

EFLAGS 00000202
ZF 0 PF 0 AF 0
OF 0 SF 0 DF 0
CF 0 TF 0 IF 1

LastError 00000000 (ERROR_SUCCESS)
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)

GS 002B FS 0053
ES 002B DS 002B
CS 0023 SS 002B

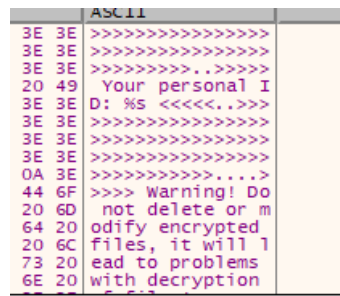
ST(0) FFFF000000007746DFE0 x87r0 Special invalid
ST(1) FFFF00000000411677B7 x87r1 Special invalid
ST(2) 00000000000000000000 x87r2 Zero 0.00000000000000000000
ST(3) 00000000000000000000 x87r3 Zero 0.00000000000000000000
ST(4) 00000000000000000000 x87r4 Zero 0.00000000000000000000

<

Default (stdcall) 5 Unlod
1: [esp+4] 013E0000 013E0000
2: [esp+8] 00000008 00000008
3: [esp+C] 00002B9A 00002B9A
4: [esp+10] 010FFA0C 010FFA0C
5: [esp+14] 005F71D7 payload.bin.005F71D7
```

Şekil 16 Rtl/AllocateHeap

11KB'lık bir alan ayrıldığı ve ayrılan alan içerisine çözümlenen verilerin yazıldığı tespit edilmiştir.



Şekil 17 README.txt Content Decryption

Çözümlenen veri incelendiğinde, sonradan oluşturulacak README.txt dosyasının içeriği olduğu görülmüştür. Zararının her bilgisayar için ID oluşturduğu tespit edilmiştir.

VictimID yapısı: "BD23223ABCFA78BC"+<randomly_generated_16_character>

Oluşturulan VictimID README.txt dosyasının içeriğinde ilgili alana entegre edilmektedir.

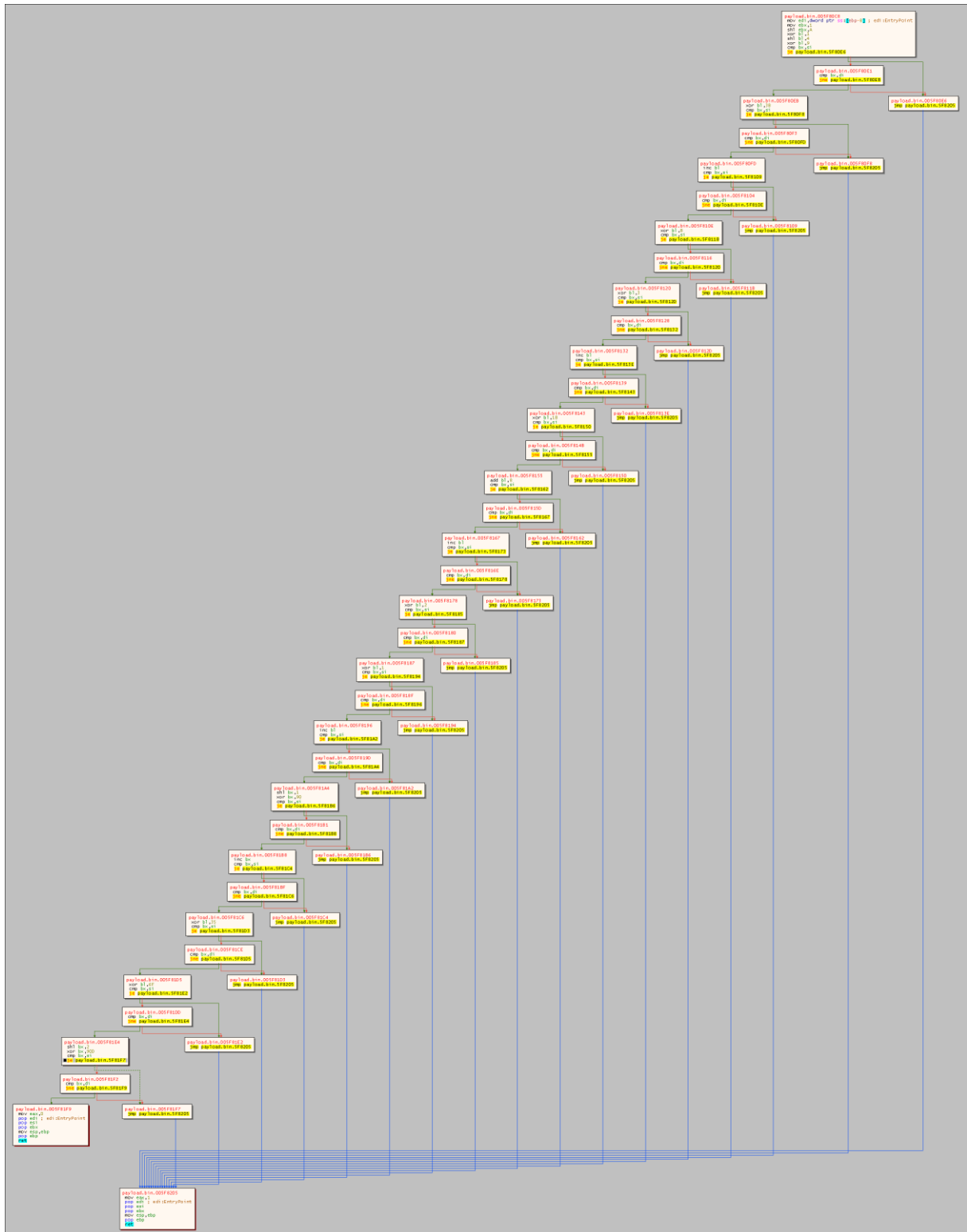
<pre> 00320E88 mov eax,6E40A006 xor eax,19039FF6 jmp eax </pre>	<table> <tr> <td>EAX</td><td>77433FF0</td><td><ntd11.NtQueryInstallUILanguage></td></tr> <tr> <td>EBX</td><td>00E7C000</td><td></td></tr> <tr> <td>ECX</td><td>013E0000</td><td></td></tr> <tr> <td>EDX</td><td>013E0000</td><td></td></tr> <tr> <td>EBP</td><td>010FFA0C</td><td></td></tr> <tr> <td>ESP</td><td>010FF9F0</td><td></td></tr> <tr> <td>ESI</td><td>0060946F</td><td><payload.bin.EntryPoint></td></tr> <tr> <td>EDI</td><td>0060946F</td><td><payload.bin.EntryPoint></td></tr> <tr> <td>EIP</td><td>00320E92</td><td></td></tr> </table>	EAX	77433FF0	<ntd11.NtQueryInstallUILanguage>	EBX	00E7C000		ECX	013E0000		EDX	013E0000		EBP	010FFA0C		ESP	010FF9F0		ESI	0060946F	<payload.bin.EntryPoint>	EDI	0060946F	<payload.bin.EntryPoint>	EIP	00320E92	
EAX	77433FF0	<ntd11.NtQueryInstallUILanguage>																										
EBX	00E7C000																											
ECX	013E0000																											
EDX	013E0000																											
EBP	010FFA0C																											
ESP	010FF9F0																											
ESI	0060946F	<payload.bin.EntryPoint>																										
EDI	0060946F	<payload.bin.EntryPoint>																										
EIP	00320E92																											

Şekil 18 NtQueryInstallUILanguage

Sisteme ait kullanılan dil bilgisi alındığı görülmüştür.

LockBit 3.0 Ailesinin Çalışmadığı Ülkeler:

- Ukrayna
- Belarus
- Tacikistan
- Ermenistan
- Azerbaycan
- Gürcistan
- Kazakistan
- Kırgızistan
- Türkmenistan
- Özbekistan
- Tataristan
- Romanya
- Rusya
- Moldova
- Arabistan
- Suriye



Şekil 19 Country Checking

mov eax,E0EE867A	
rol eax,7	
jmp eax	NtOpenProcessToken
lodsd	
mov edx,BAADF00D	

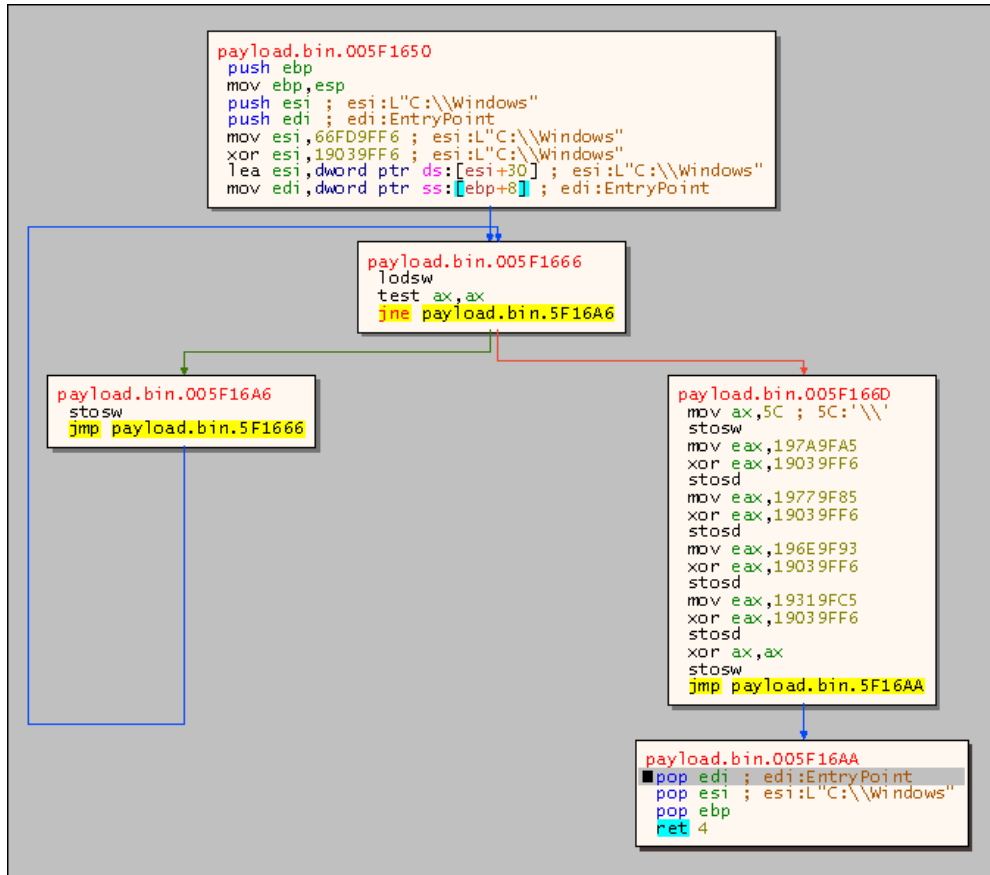
Şekil 20 NtOpenProcessToken

Sürecin kendisine ait Access token tutamacının alındığı tespit edildi.

mov eax,C8165ECD		GS 002B FS 0053
ror eax,5		ES 002B DS 002B
xor eax,19039FF6		CS 0023 SS 002B
jmp eax	ZwQueryInformationToken	ST(0) FFFF0000000007746DFE0 x87r0 S
mov edx,ABABABAB		ST(1) FFFF000000000411677B7 x87r1 S
stosd		ST(2) 000000000000000000000000 x87r2 Z
stosd		ST(3) 000000000000000000000000 x87r3 Z
stosd		ST(4) 000000000000000000000000 x87r4 Z
add byte ptr ds:[eax],al		<
		Default (stdcall)
		1: [esp+4] 000002D8 000002D8
		2: [esp+8] 00000002 00000002
		3: [esp+C] 01418E98 01418E98
		4: [esp+10] 00000140 00000140
		5: [esp+14] 010FF9FC 010FF9FC

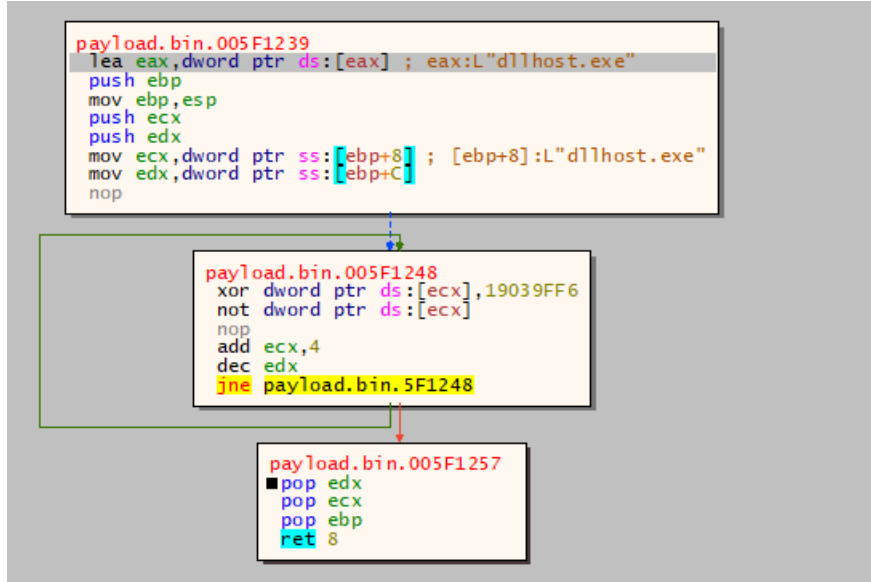
Şekil 21 ZwqueryInformationToken

Sürecin çalıştığı Access token yapısını kullanarak kullanıcı grup bilgisinin çekildiği tespit edilmiştir.



Şekil 22 Creation System32 Path

Şekil 22’te “C:\\Windows\\System32” ifadesi oluşturulmaktadır. XOR yöntemi kullanılarak güvenlik ürünlerinden kaçınıldığı görülmektedir.



Şekil 23 Decryption Algorithm

Şekil 23’te gösterilen algoritma ile çözümlenen ifadeler şunlardır:

- dllhost.exe
- Elevation:Administrator!new:{{3E5FC7F9-9A51-4367-9063-A120244FBEC7}}

```

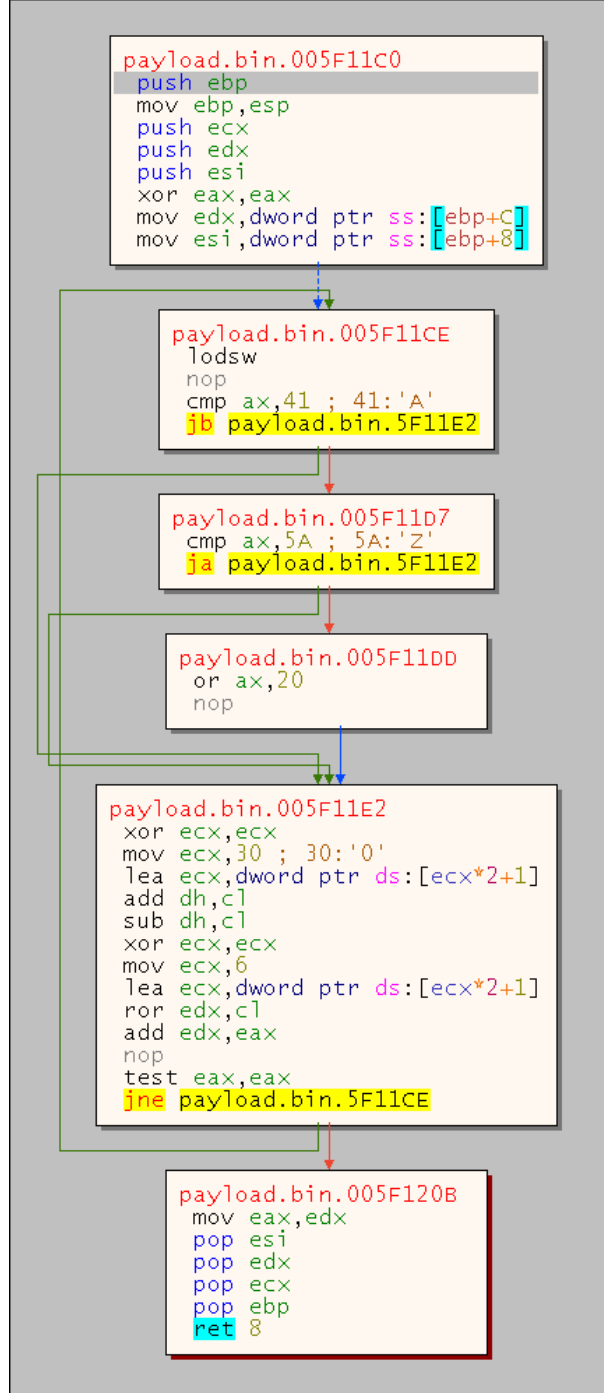
DWORD* decryption_function(DWORD *array,size_t size) {
    for (int i = 0; i < size; i++) {
        array[i] = array[i] ^ 0x19039ff6;
        array[i] = ~(array[i]);

        std::cout << std::hex<< array[i]<<" ";
    }
    return array;
}
    
```


<pre>jmp eax or eax,ABBAADF0 stosd stosd stosd stosd</pre>	<div>Hide FPU</div> <table><tr><td>EAX</td><td>77432E50</td><td><ntdll.ZwQuerySystemInformation></td></tr><tr><td>EBX</td><td>00000000</td><td></td></tr><tr><td>ECX</td><td>40000068</td><td></td></tr><tr><td>EDX</td><td>00000000</td><td></td></tr><tr><td>EBP</td><td>0362F8E8</td><td></td></tr><tr><td>ESP</td><td>0362F8C0</td><td></td></tr></table>	EAX	77432E50	<ntdll.ZwQuerySystemInformation>	EBX	00000000		ECX	40000068		EDX	00000000		EBP	0362F8E8		ESP	0362F8C0	
EAX	77432E50	<ntdll.ZwQuerySystemInformation>																	
EBX	00000000																		
ECX	40000068																		
EDX	00000000																		
EBP	0362F8E8																		
ESP	0362F8C0																		

Şekil 26 ZwQuerySystemInformation

Sistem üzerinde çalışmakta olan process bilgilerinin çekildiği tespit edilmiştir.



Şekil 27 Process Name Hashing Algorithm

DLL dosyalarında gezinmeye benzer bir şekilde process isimlerine ait karma (hash) oluşturulduğu ve istenilen bir process adına ait hash ile karşılaştırıldığı tespit edildi. Ulaşılmak istenilen process adının explorer.exe olduğu tespit edildi.

<pre> mov eax,C8161CCD ror eax,5 xor eax,19039FF6 jmp eax mov edx,ABABABAB stosd stosd stosd stosd add byte ptr ds:[eax].a] </pre>	<table border="1"> <tr><th colspan="3">Hide</th></tr> <tr><td>EAX</td><td>77432F10</td><td><ntdll.ZwDuplicateToken></td></tr> <tr><td>EBX</td><td>00E7C000</td><td></td></tr> <tr><td>ECX</td><td>C2150000</td><td></td></tr> <tr><td>EDX</td><td>00000000</td><td></td></tr> <tr><td>EBP</td><td>010FF9F4</td><td></td></tr> <tr><td>ESP</td><td>010FF9A0</td><td></td></tr> <tr><td>ESI</td><td>0060946F</td><td><payload.bin.EntryPoint></td></tr> <tr><td>EDI</td><td>0060946F</td><td><payload.bin.EntryPoint></td></tr> <tr><td>EIP</td><td>033209E5</td><td></td></tr> </table>	Hide			EAX	77432F10	<ntdll.ZwDuplicateToken>	EBX	00E7C000		ECX	C2150000		EDX	00000000		EBP	010FF9F4		ESP	010FF9A0		ESI	0060946F	<payload.bin.EntryPoint>	EDI	0060946F	<payload.bin.EntryPoint>	EIP	033209E5	
Hide																															
EAX	77432F10	<ntdll.ZwDuplicateToken>																													
EBX	00E7C000																														
ECX	C2150000																														
EDX	00000000																														
EBP	010FF9F4																														
ESP	010FF9A0																														
ESI	0060946F	<payload.bin.EntryPoint>																													
EDI	0060946F	<payload.bin.EntryPoint>																													
EIP	033209E5																														

Şekil 28 DuplicateToken

explorer.exe sürecine ait process access token bilgisinin kopyalandığı tespit edildi.

<pre> mov eax,41DB70E0 rol eax,6 jmp eax lodsd mov edx,BAADF00D stosd stosd stosd stosd stosd stosd stosd stosd add byte ptr ds:[eax],a] add byte ptr ds:[eax],a] add byte ptr ds:[eax],a] add byte ptr ds:[eax],a] shl dword ptr ds:[edi+28E09A37],1 add byte ptr ds:[eax],b] mov eax,403B6E1E rol eax,9 jmp eax lodsd mov edx,BAADF00D stosd </pre>	<table border="1"> <tr><th colspan="3">Hide FPU</th></tr> <tr><td>EAX</td><td>76DC3810</td><td><kernel32.CreateFile></td></tr> <tr><td>EBX</td><td>01402D08</td><td></td></tr> <tr><td>ECX</td><td>FFFFFFFFE6</td><td></td></tr> <tr><td>EDX</td><td>FFCFEE24</td><td></td></tr> <tr><td>EBP</td><td>010FFA08</td><td></td></tr> <tr><td>ESP</td><td>010FF77C</td><td></td></tr> <tr><td>ESI</td><td>01416758</td><td></td></tr> <tr><td>EDI</td><td>010FF7D8</td><td>L".ico"</td></tr> <tr><td>EIP</td><td>76DC3810</td><td><kernel32.CreateFile></td></tr> <tr><td colspan="3">EFLAGS 00000206</td></tr> <tr><td>ZF</td><td>0</td><td>PF 1 AF 0</td></tr> <tr><td>OF</td><td>0</td><td>SF 0 DF 0</td></tr> <tr><td>CF</td><td>0</td><td>TF 0 IF 1</td></tr> <tr><td colspan="3">LastError 00000000 (ERROR_SUCCESS)</td></tr> <tr><td colspan="3">LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)</td></tr> <tr><td colspan="3">GS 0028 FS 0053</td></tr> <tr><td colspan="3">ES 0028 DS 0028</td></tr> <tr><td colspan="3">CS 0023 SS 0028</td></tr> <tr><td colspan="3">ST(0) FFFF0000000007746DFE0 x87r0 Special invalid</td></tr> <tr><td colspan="3">ST(1) FFFF000000000411677B7 x87r1 Special invalid</td></tr> <tr><td colspan="3">ST(2) 00000000000000000000 x87r2 Zero 0.00000000000000000000</td></tr> <tr><td colspan="3">ST(3) 00000000000000000000 x87r3 Zero 0.00000000000000000000</td></tr> <tr><td colspan="3">ST(4) 00000000000000000000 x87r4 Zero 0.00000000000000000000</td></tr> <tr><td colspan="3"><</td></tr> <tr><td colspan="3">Default (stdcall)</td></tr> <tr><td>1:</td><td>[esp+4]</td><td>010FF7A8 010FF7A8 L"C:\\ProgramData\\2uaphKeDI.ico"</td></tr> <tr><td>2:</td><td>[esp+8]</td><td>40000000 40000000</td></tr> <tr><td>3:</td><td>[esp+C]</td><td>00000000 00000000</td></tr> <tr><td>4:</td><td>[esp+10]</td><td>00000000 00000000</td></tr> <tr><td>5:</td><td>[esp+14]</td><td>00000002 00000002</td></tr> </table>	Hide FPU			EAX	76DC3810	<kernel32.CreateFile>	EBX	01402D08		ECX	FFFFFFFFE6		EDX	FFCFEE24		EBP	010FFA08		ESP	010FF77C		ESI	01416758		EDI	010FF7D8	L".ico"	EIP	76DC3810	<kernel32.CreateFile>	EFLAGS 00000206			ZF	0	PF 1 AF 0	OF	0	SF 0 DF 0	CF	0	TF 0 IF 1	LastError 00000000 (ERROR_SUCCESS)			LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)			GS 0028 FS 0053			ES 0028 DS 0028			CS 0023 SS 0028			ST(0) FFFF0000000007746DFE0 x87r0 Special invalid			ST(1) FFFF000000000411677B7 x87r1 Special invalid			ST(2) 00000000000000000000 x87r2 Zero 0.00000000000000000000			ST(3) 00000000000000000000 x87r3 Zero 0.00000000000000000000			ST(4) 00000000000000000000 x87r4 Zero 0.00000000000000000000			<			Default (stdcall)			1:	[esp+4]	010FF7A8 010FF7A8 L"C:\\ProgramData\\2uaphKeDI.ico"	2:	[esp+8]	40000000 40000000	3:	[esp+C]	00000000 00000000	4:	[esp+10]	00000000 00000000	5:	[esp+14]	00000002 00000002
Hide FPU																																																																																														
EAX	76DC3810	<kernel32.CreateFile>																																																																																												
EBX	01402D08																																																																																													
ECX	FFFFFFFFE6																																																																																													
EDX	FFCFEE24																																																																																													
EBP	010FFA08																																																																																													
ESP	010FF77C																																																																																													
ESI	01416758																																																																																													
EDI	010FF7D8	L".ico"																																																																																												
EIP	76DC3810	<kernel32.CreateFile>																																																																																												
EFLAGS 00000206																																																																																														
ZF	0	PF 1 AF 0																																																																																												
OF	0	SF 0 DF 0																																																																																												
CF	0	TF 0 IF 1																																																																																												
LastError 00000000 (ERROR_SUCCESS)																																																																																														
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)																																																																																														
GS 0028 FS 0053																																																																																														
ES 0028 DS 0028																																																																																														
CS 0023 SS 0028																																																																																														
ST(0) FFFF0000000007746DFE0 x87r0 Special invalid																																																																																														
ST(1) FFFF000000000411677B7 x87r1 Special invalid																																																																																														
ST(2) 00000000000000000000 x87r2 Zero 0.00000000000000000000																																																																																														
ST(3) 00000000000000000000 x87r3 Zero 0.00000000000000000000																																																																																														
ST(4) 00000000000000000000 x87r4 Zero 0.00000000000000000000																																																																																														
<																																																																																														
Default (stdcall)																																																																																														
1:	[esp+4]	010FF7A8 010FF7A8 L"C:\\ProgramData\\2uaphKeDI.ico"																																																																																												
2:	[esp+8]	40000000 40000000																																																																																												
3:	[esp+C]	00000000 00000000																																																																																												
4:	[esp+10]	00000000 00000000																																																																																												
5:	[esp+14]	00000002 00000002																																																																																												

Şekil 29 CreateFile: Creation LockBit Icon File

"C:\\ProgramData\\2uaphKeDI.ico" dosyasının oluşturulduğu tespit edildi.

<pre> mov eax,403B6E1E rol eax,9 jmp eax lodsd mov edx,BAADF00D stosd stosd stosd stosd stosd stosd stosd stosd add byte ptr ds:[eax],a] add byte ptr ds:[eax],a] add byte ptr ds:[eax],a] add byte ptr ds:[eax],a] shl dword ptr ds:[edi+28E09A37],1 add byte ptr ds:[eax],b] mov eax,6FDA466 xor eax,19039FF6 jmp eax or eax,ABBAADF0 stosd stosd </pre>	<table border="1"> <tr><th colspan="3">Hide FPU</th></tr> <tr><td>EAX</td><td>76DC3C80</td><td><kernel32.WriteFile></td></tr> <tr><td>EBX</td><td>01402D08</td><td></td></tr> <tr><td>ECX</td><td>806B7100</td><td></td></tr> <tr><td>EDX</td><td>00000000</td><td></td></tr> <tr><td>EBP</td><td>010FFA08</td><td></td></tr> <tr><td>ESP</td><td>010FF784</td><td></td></tr> <tr><td>ESI</td><td>01416758</td><td></td></tr> <tr><td>EDI</td><td>010FF7D8</td><td>L".ico"</td></tr> <tr><td>EIP</td><td>03321098</td><td></td></tr> <tr><td colspan="3">EFLAGS 00000216</td></tr> <tr><td>ZF</td><td>0</td><td>PF 1 AF 1</td></tr> <tr><td>OF</td><td>0</td><td>SF 0 DF 0</td></tr> <tr><td>CF</td><td>0</td><td>TF 0 IF 1</td></tr> <tr><td colspan="3">LastError 00000000 (ERROR_SUCCESS)</td></tr> <tr><td colspan="3">LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)</td></tr> <tr><td colspan="3">GS 0028 FS 0053</td></tr> <tr><td colspan="3">ES 0028 DS 0028</td></tr> <tr><td colspan="3">CS 0023 SS 0028</td></tr> <tr><td colspan="3">ST(0) FFFF0000000007746DFE0 x87r0 Special invalid</td></tr> <tr><td colspan="3">ST(1) FFFF000000000411677B7 x87r1 Special invalid</td></tr> <tr><td colspan="3">ST(2) 00000000000000000000 x87r2 Zero 0.00000000000000000000</td></tr> <tr><td colspan="3">ST(3) 00000000000000000000 x87r3 Zero 0.00000000000000000000</td></tr> <tr><td colspan="3">ST(4) 00000000000000000000 x87r4 Zero 0.00000000000000000000</td></tr> <tr><td colspan="3"><</td></tr> <tr><td colspan="3">Default (stdcall)</td></tr> <tr><td>1:</td><td>[esp+4]</td><td>000002B0 000002B0</td></tr> <tr><td>2:</td><td>[esp+8]</td><td>01402D08 01402D08</td></tr> <tr><td>3:</td><td>[esp+C]</td><td>00003AEE 00003AEE</td></tr> <tr><td>4:</td><td>[esp+10]</td><td>010FF9F8 010FF9F8</td></tr> <tr><td>5:</td><td>[esp+14]</td><td>00000000 00000000</td></tr> </table>	Hide FPU			EAX	76DC3C80	<kernel32.WriteFile>	EBX	01402D08		ECX	806B7100		EDX	00000000		EBP	010FFA08		ESP	010FF784		ESI	01416758		EDI	010FF7D8	L".ico"	EIP	03321098		EFLAGS 00000216			ZF	0	PF 1 AF 1	OF	0	SF 0 DF 0	CF	0	TF 0 IF 1	LastError 00000000 (ERROR_SUCCESS)			LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)			GS 0028 FS 0053			ES 0028 DS 0028			CS 0023 SS 0028			ST(0) FFFF0000000007746DFE0 x87r0 Special invalid			ST(1) FFFF000000000411677B7 x87r1 Special invalid			ST(2) 00000000000000000000 x87r2 Zero 0.00000000000000000000			ST(3) 00000000000000000000 x87r3 Zero 0.00000000000000000000			ST(4) 00000000000000000000 x87r4 Zero 0.00000000000000000000			<			Default (stdcall)			1:	[esp+4]	000002B0 000002B0	2:	[esp+8]	01402D08 01402D08	3:	[esp+C]	00003AEE 00003AEE	4:	[esp+10]	010FF9F8 010FF9F8	5:	[esp+14]	00000000 00000000
Hide FPU																																																																																														
EAX	76DC3C80	<kernel32.WriteFile>																																																																																												
EBX	01402D08																																																																																													
ECX	806B7100																																																																																													
EDX	00000000																																																																																													
EBP	010FFA08																																																																																													
ESP	010FF784																																																																																													
ESI	01416758																																																																																													
EDI	010FF7D8	L".ico"																																																																																												
EIP	03321098																																																																																													
EFLAGS 00000216																																																																																														
ZF	0	PF 1 AF 1																																																																																												
OF	0	SF 0 DF 0																																																																																												
CF	0	TF 0 IF 1																																																																																												
LastError 00000000 (ERROR_SUCCESS)																																																																																														
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)																																																																																														
GS 0028 FS 0053																																																																																														
ES 0028 DS 0028																																																																																														
CS 0023 SS 0028																																																																																														
ST(0) FFFF0000000007746DFE0 x87r0 Special invalid																																																																																														
ST(1) FFFF000000000411677B7 x87r1 Special invalid																																																																																														
ST(2) 00000000000000000000 x87r2 Zero 0.00000000000000000000																																																																																														
ST(3) 00000000000000000000 x87r3 Zero 0.00000000000000000000																																																																																														
ST(4) 00000000000000000000 x87r4 Zero 0.00000000000000000000																																																																																														
<																																																																																														
Default (stdcall)																																																																																														
1:	[esp+4]	000002B0 000002B0																																																																																												
2:	[esp+8]	01402D08 01402D08																																																																																												
3:	[esp+C]	00003AEE 00003AEE																																																																																												
4:	[esp+10]	010FF9F8 010FF9F8																																																																																												
5:	[esp+14]	00000000 00000000																																																																																												

Şekil 30 WriteFile: LockBit Icon

Meşhur LockBit dosya ikonunu içeriğinin yazıldığı tespit edildi.

The screenshot shows a debugger window with assembly code on the left and register values on the right. The assembly code includes instructions like `mov eax, 6F7473F6`, `xor eax, 19039FF6`, `jmp eax`, `or eax, ABBAADF0`, `stosd`, `add byte ptr ds:[eax], al`, `shl dword ptr ds:[edi+28E09A37], 1`, `add byte ptr ds:[eax], bl`, `mov eax, 80ECEFD7`, `rol eax, 7`, `jmp eax`, `lodsd`, `mov edx, BAADF00D`, `stosd`, and `stosd`. The register window on the right shows `EAX: 7677EC00 <advapi32.RegCreateKeyExW>`, `ECX: 8164868D`, `EDX: 010FF760`, `EBP: 010FFA08`, `ESP: 010FF774`, `ESI: 01416758`, `EDI: 010FF7D8 L".ico"`, `EIP: 03321932`, `EFLAGS: 00000206`, `LastError: 00000000 (ERROR_SUCCESS)`, `LastStatus: C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)`, `GS: 002B FS: 0053`, `ES: 002B DS: 002B`, `CS: 0023 SS: 002B`, and a stack dump showing `ST(0) FFFF000000007746DFE0 x87r0 Special invalid`, `ST(1) FFFF00000000411677B7 x87r1 Special invalid`, `ST(2) 00000000000000000000 x87r2 Zero 0.00000000000000000000`, `ST(3) 00000000000000000000 x87r3 Zero 0.00000000000000000000`, and `ST(4) 00000000000000000000 x87r4 Zero 0.00000000000000000000`. The stack dump also shows `Default (stdcall)` with values like `1: [esp+4] 80000000 80000000`, `2: [esp+8] 014009A0 014009A0 L".2uaphKeD1"`, `3: [esp+C] 00000000 00000000`, `4: [esp+10] 00000000 00000000`, and `5: [esp+14] 00000000 00000000`.

Şekil 31 RegCreateKeyExW

HKEY_CLASSES_ROOT anahtarının altında ".2uaphKeDI" adında alt anahtar oluşturulduğu tespit edildi.

The screenshot shows a debugger window with assembly code on the left and register values on the right. The assembly code includes instructions like `mov eax, 6F7473F6`, `xor eax, 19039FF6`, `jmp eax`, `or eax, ABBAADF0`, `stosd`, `stosd`, `stosd`, and `stosd`. The register window on the right shows `EAX: 7677EC00 <advapi32.RegCreateKeyExW>`, `EBX: 01402D08`, `ECX: FFFFFFFF`, `EDX: 01400986`, `EBP: 010FFA08`, `ESP: 010FF774`, `ESI: 01416758`, `EDI: 010FF9C2 L"\\DefaultIcon"`, `EIP: 03321932`, and `EFLAGS: 00000206`.

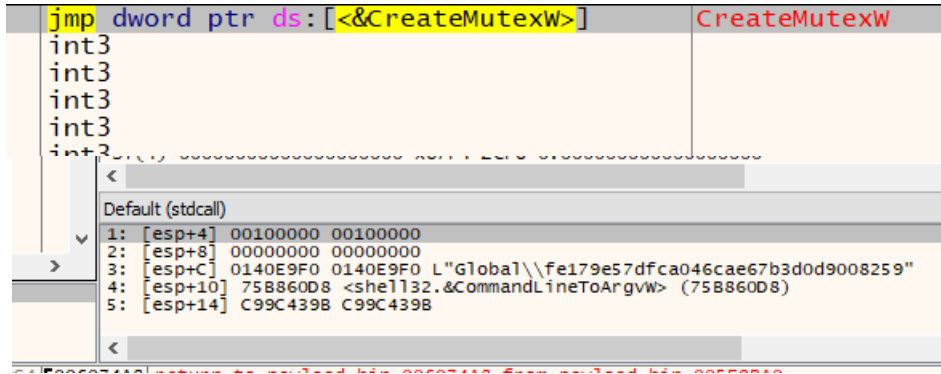
Şekil 32 RegCreateKeyExW: DefaultIcon

".2uaphKeDI" alt anahtarının altına da "DefaultIcon" isimli bir alt anahtar açıldığı tespit edildi.

The screenshot shows a debugger window with assembly code on the left and register values on the right. The assembly code includes instructions like `mov eax, 80ECEFD7`, `rol eax, 7`, `jmp eax`, `lodsd`, `mov edx, BAADF00D`, `stosd`, `stosd`, `stosd`, and `stosd`. The register window on the right shows `EAX: 7677EBC0 <advapi32.RegSetValueExW>`, `EBX: 01402D08`, `ECX: 00000000`, `EDX: 7736F010 kernelbase.7736F010`, `EBP: 010FFA08`, `ESP: 010FF780`, `ESI: 01416758`, `EDI: 010FF9C2 L"\\DefaultIcon"`, `EIP: 03321958`, and `EFLAGS: 00000206`. The stack dump shows `ST(4) 00000000000000000000 x87r4 Zero 0.00000000000000000000` and `Default (stdcall)` with values like `1: [esp+4] 00000300 00000300`, `2: [esp+8] 010FF9F0 010FF9F0`, `3: [esp+C] 00000000 00000000`, `4: [esp+10] 00000001 00000001`, and `5: [esp+14] 010FF7A8 010FF7A8 L"C:\\ProgramData\\2uaphKeD1.ico"`.

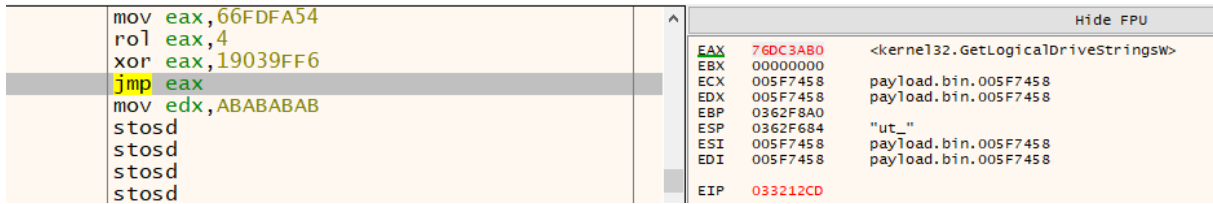
Şekil 33 RegSetValueExW: Setting Icon File Path

Oluşturulan alt anahtara ait değer içerisinde oluşturulan ikon dosyasının dizini yazılmaktadır.



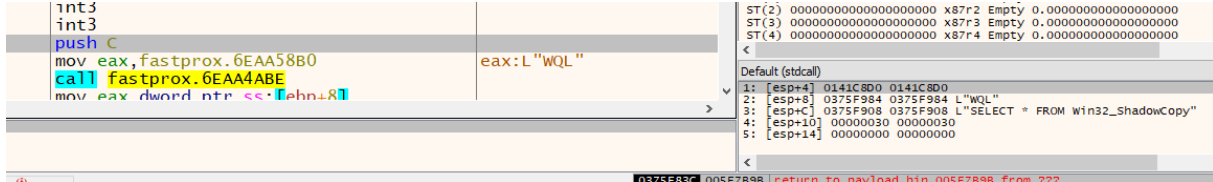
Şekil 34 CreateMutex

"Global\\fe179e57dfca046cae67b3d0d9008259" isimli bir Mutex oluşturulduğu tespit edildi.



Şekil 35 GetLogicalDriveStringsW

Cihazda bulunan sürücülere ait izin bilgileri çekilmektedir. Çekilen sürücülerin hard disk gibi bir depolama birimi olup olmadığının kontrolü yapılmaktadır.



Şekil 36 Delete Shadow Copies

Shadow kopyalarını toplamak için WMI sorgusu çalıştırıldığı tespit edildi.

Ardından, "C:\\\\" dizininden başlayarak özellikle users dizini altındaki izinleri gezerek dosyaları şifrelemektedir.

Geleneksel fidye yazılımlarından farklı olarak, tek bir thread kullanarak izinler arasında gezinme yaptığı ve dosya şifrelediği görülmüştür. Tek bir thread kullanılmasına rağmen bu denli hızlı olmasının en büyük nedenleri ise şu şekildedir: gezinilen izinlerin önemliliğinin kontrol edilmesi ve dosyalar şifrelenirken hazır fonksiyonlar değil, özel olarak yazılan şifreleme fonksiyonları kullanılmaktadır.

<pre> mov eax,41DB70E0 rol eax,6 jmp eax lodsd mov edx,BAADF00D stosd stosd stosd stosd stosd stosd stosd add byte ptr ds:[eax],al add byte ptr ds:[eax],al </pre>	<p>Hide FPU</p> <p>EAX 76DC3810 <kernel32.CreateFile> EBX 041AF550 L"Everywhere.search-ms" ECX 806B710D EDX 00000000 EBP 041AF4EC ESP 041AF4AC ESI 0140AFE8 L"\\\\?\\C:\\Users\\ceku\\Searches\\Everywhere.search-ms" EDI 005FF308 payload.bin.005FF308 EIP 03321070</p> <p>EFLAGS 00000246 ZF 1 PF 1 AF 0 OF 0 SF 0 DF 0 CF 0 TF 0 IF 1</p> <p>LastError 0000051B (ERROR_INVALID_OWNER) LastStatus C000005A (STATUS_INVALID_OWNER)</p>
--	---

Şekil 37 Opening File that will Encrypt

Tespit edilen dosyanın açılması,

<pre> mov eax,6FDA466 xor eax,19039FF6 jmp eax or eax,ABBAADF0 stosd stosd stosd stosd stosd stosd stosd </pre>	<p>Hide FPU</p> <p>EAX 76DC3890 <kernel32.ReadFile> EBX 041AF550 L"Everywhere.search-ms" ECX 806B710D EDX 00000000 EBP 041AF4EC ESP 041AF440 ESI 0140AFE8 L"\\\\?\\C:\\Users\\ceku\\Searches\\Everywhere.search-ms" EDI 005FF308 payload.bin.005FF308 EIP 033210C2</p>
---	--

Okunması,

<pre> mov eax,BF601919 ror eax,2 xor eax,19039FF6 jmp eax mov edx,ABABABAB stosd </pre>	<p>Hide FPU</p> <p>EAX 76DB9980 <kernel32.MoveFileExW> EBX 041AF550 L"Everywhere.search-ms" ECX 00000000 EDX 0001C56E EBP 041AF510 ESP 041AF4E8</p>
---	--

Şekil 38 MoveFileExW

Farklı uzantı ile kopyası oluşturulur. Ardından oluşturulan yeni dosya açılmakta ve içerik şifrelenerek tekrar aynı dosya üzerine yazılmaktadır.

No.	Time	Source	Destination	Protocol	Length	Info
414	384.369510	VMware_e4:7a:74	VMware_7f:3e:b3	ARP	60	192.168.13.2 is at 00:50:
415	384.846045	192.168.13.1	192.168.13.255	NBNS	92	Name query NB DESKTOP-BR9
416	384.853656	192.168.13.252	20.199.120.182	TCP	66	[TCP Retransmission] [TCP
417	386.153532	192.168.13.1	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
418	386.379697	192.168.13.252	192.168.13.255	NBNS	92	Name query NB DESKTOP-KU4
419	386.869950	192.168.13.252	20.199.120.182	TCP	66	[TCP Retransmission] [TCP
420	387.137077	192.168.13.252	192.168.13.255	NBNS	92	Name query NB DESKTOP-KU4
421	387.156817	192.168.13.1	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
422	387.887703	192.168.13.252	192.168.13.255	NBNS	92	Name query NB DESKTOP-KU4
423	388.169344	192.168.13.1	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
424	388.842618	192.168.13.252	192.168.13.255	NBNS	92	Name query NB DESKTOP-KU4

> Frame 421: 216 bytes on wire (1728 bits), 216 byte	1 11 1c 53 c0 a8 0d 01 ef ff	...
> Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:	0 b6 6d fc 4d 2d 53 45 41 521...m-M-SEAR
> Internet Protocol Version 4, Src: 192.168.13.1, Ds	4 54 50 2f 31 2e 31 0d 0a 48	CH * HTTP/1.1...H
> User Datagram Protocol, Src Port: 50869, Dst Port:	3 39 2e 32 35 35 2e 32 35 35	OST: 239 .255.255
> Simple Service Discovery Protocol	9 30 30 0d 0a 4d 41 4e 3a 20	.250:190 0...MAN:
	4 69 73 63 6f 76 65 72 22 0d	"ssdp:discover"
	d 0a 53 54 3a 20 75 72 6e 3a	-MX: 1...ST: urn:
	5 6c 74 69 73 63 72 65 65 6e	dial-multiscreen
	5 72 76 69 63 65 3a 64 69 61	-org:service:dia
	3 45 52 2d 41 47 45 4e 54 3a	l:1...USE R-AGENT:
	5 20 43 68 72 6f 6d 65 2f 31	Google Chrome/1
	9 39 33 2e 38 39 20 57 69 6e	18.0.599 3.89 Win
	d 0a	dows....

Yapılan incelemeler sonucunda zararlı yazılım ile ilişkilendirilebilecek bazı IP bilgileri tespit edilmiştir. Bunlar:

- 239.255.255.250
- 224.0.0.252

Kurallar

YARA

```
rule LockBit_3_0{

meta:
    date = "2023-10-26"
    description = "Detects LockBit 3.0"
    author = "Bilal BAKARTEPE"
    hash = "bbe63d8efc8d8dc7f387b08ee07721ba"
    verdict = "dangerous"
    platform = "windows"

strings:
    $hash1={2D D8 63 77} //ntdll RtlAllocateHeap
    $hash2={54 31 19 c3} //FindFirstFile
    $hash3={23 56 69 4e} //FindNextFile
    $hash4={8a a5 43 61} //FindClose
    $hash5={f6 9f 03 19} //MD4Init

    $xorkey={f6 9f 03 19} //xor key for hashed API's

    $opc1={55 8B EC 51 52 56 33 C0 8B 55 0C 8B 75 08 AC 33 C9 B9 30 00 00 00 8D 0C 4D 01 00 00 00 02
F1 2A F1 33 C9 B9 06 00 00 00 8D 0C 4D 01 00 00 00 D3 CA 03 D0 90 85 C0 75 D6 8B C2 5E 5A 59 5D} //API
name hasher algorithm
    $opc2={55 8B EC 56 57 BE F6 9F FD 66 81 F6 F6 9F 03 19 8D 76 30 8B 7D 08 66 AD 66 85 C0 75 39 66
B8 5C 00 66 AB B8 A5 9F 7A 19 35 F6 9F 03 19 AB B8 85 9F 77 19 35 F6 9F 03 19 AB B8 93 9F 6E 19 35
F6 9F 03 19 AB B8 C5 9F 31 19 35 F6 9F 03 19 AB 66 33 C0 66 AB EB 04 66 AB EB BC 5F 5E 5D C2 04 00}
//deobfuscating "C:\\windows\\system32" string
    $opc3={C7 03 55 60 D6 E6 C7 43 04 27 60 98 E6 C7 43 08 65 60 90 E6 C7 43 0C 09 60 FC
E6} //deobfuscating "*.dll" string
    $opc4={55 8B EC 51 52 8B 4D 08 8B 55 0C 90 81 31 F6 9F 03 19 F7 11 90 83 C1 04 4A 75 F1 5A 59 5D}
//deobfuscating "*.dll" string together
    $opc5={66 83 F8 41 72 0B 66 83 F8 5A 77 05 66 83 C8 20 90 33 C9 B9 30 00 00 00 8D 0C 4D 01 00 00
00 02 F1 2A F1 33 C9 B9 06 00 00 00 8D 0C 4D 01 00 00 00 D3 CA 03 D0 90 85 C0 75 C3} //Dll name
hashing
    $opc6={8B 40 18 F7 40 44 00 00 00 40 74 02 D1 C8} //Heap-based Anti-debug
    $opc7={B9 5D 34 A8 B2 81 F1 F6 9F 03 19 39 48 10 74 01 AB C6 00 B8} //Heap-based Anti-debug

condition:
    any of ($opc*) or (any of ($hash*)and $xorkey)
}
```

SIGMA – 1

```
title: LockBit 3.0 Registry Operation
status: experimental
description: Detects LockBit 3.0 icon file definition
author: Bilal BAKARTEPE
date: 2023/10/26
logsource:
  category: registry_set
  product: windows
detection:
  selection:
    CommandLine|contains:
      - HKEY_CLASSES_ROOT
      - .2uaphKeDl
    TargetObject|endswith: reg.exe
  condition: selection
falsepositives:
  - Unknown
level: high
```

SIGMA – 2

```
title: Win32_ShadowCopy Query Alert
description: Detects a query for Win32_ShadowCopy class in WMI.
author: Bilal BAKARTEPE
date: 2023-10-26
logsource:
  product: windows
  service: security
detection:
  selection:
    EventID: 10 # Event ID for WMI Queries (Adjust this if needed)
    Query: "*FROM Win32_ShadowCopy*"
  condition: selection
level: high
tags:
  - wmi
  - windows
  - alert
falsepositives:
  - Legitimate use of WMI for querying Win32_ShadowCopy
fields:
  - Query
  - EventID
  - ComputerName
  - User
  - ProcessName
  - ParentProcessName
  - ParentProcessID
  - CommandLine
```


MITRE ATT&CK Tablosu

Tactic	ID	Technic Name
<u>Privilege Escalation</u>	<u>T1548.002</u>	<u>Abuse Elevation Control Mechanism: Bypass User Account Control</u>
<u>Privilege Escalation</u>	<u>T1134</u>	<u>Access Token Manipulation</u>
<u>Discovery</u>	<u>T1083</u>	<u>File and Directory Discovery</u>
<u>Discovery</u>	<u>T1069.002</u>	<u>Permission Groups Discovery: Domain Groups</u>
<u>Discovery</u>	<u>T1082</u>	<u>System Information Discovery</u>
<u>Execution</u>	<u>T1047</u>	<u>Windows Management Instrumentation</u>



ECHO

CYBER THREAT INTELLIGENCE