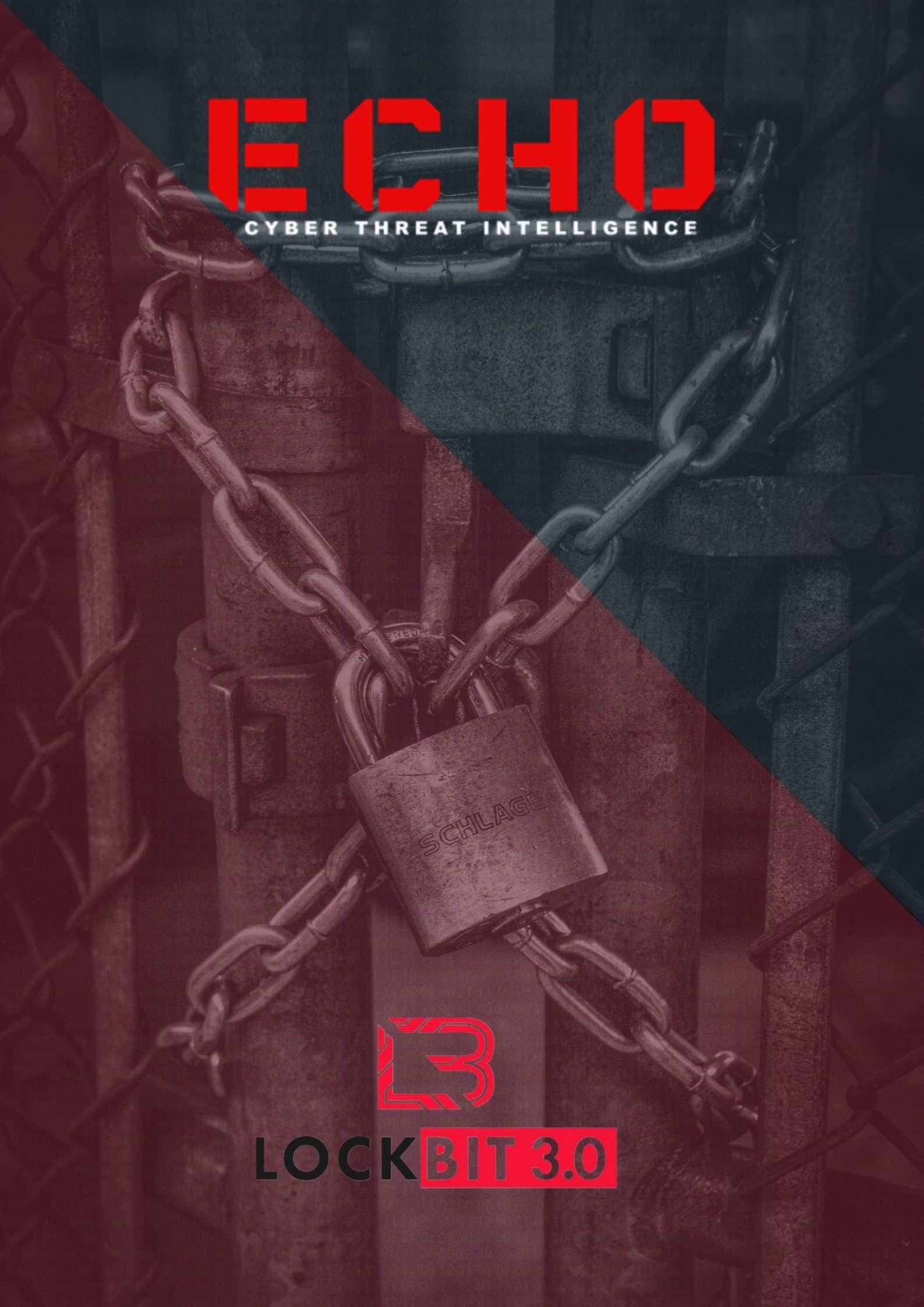

ECHO
CYBER THREAT INTELLIGENCE
SCHLAGE
LOCKBIT 3.0

İçindekiler

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 çözümlenmesi 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. Fidye, 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:

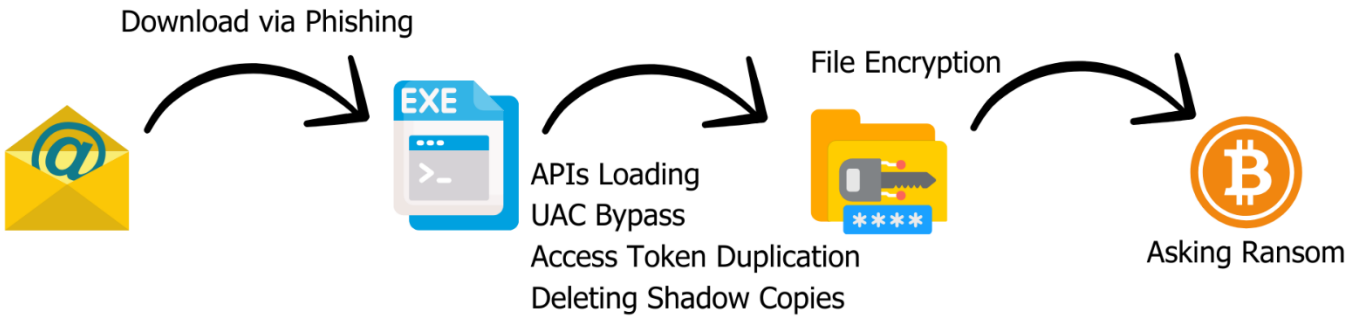
- Rusya: LockBit 3.0, Rusya'daki kuruluşları sık sık hedef alır. Bu, Rusya'daki büyük ve küçük işletmeleri, devlet kurumlarını ve bireyleri etkileyebilir.
- Amerika Birleşik Devletleri: ABD, dünyanın en büyük ekonomilerinden biri olduğu için LockBit 3.0 için cazip bir hedef olabilir. Finans, sağlık, 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 tehlikesi 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

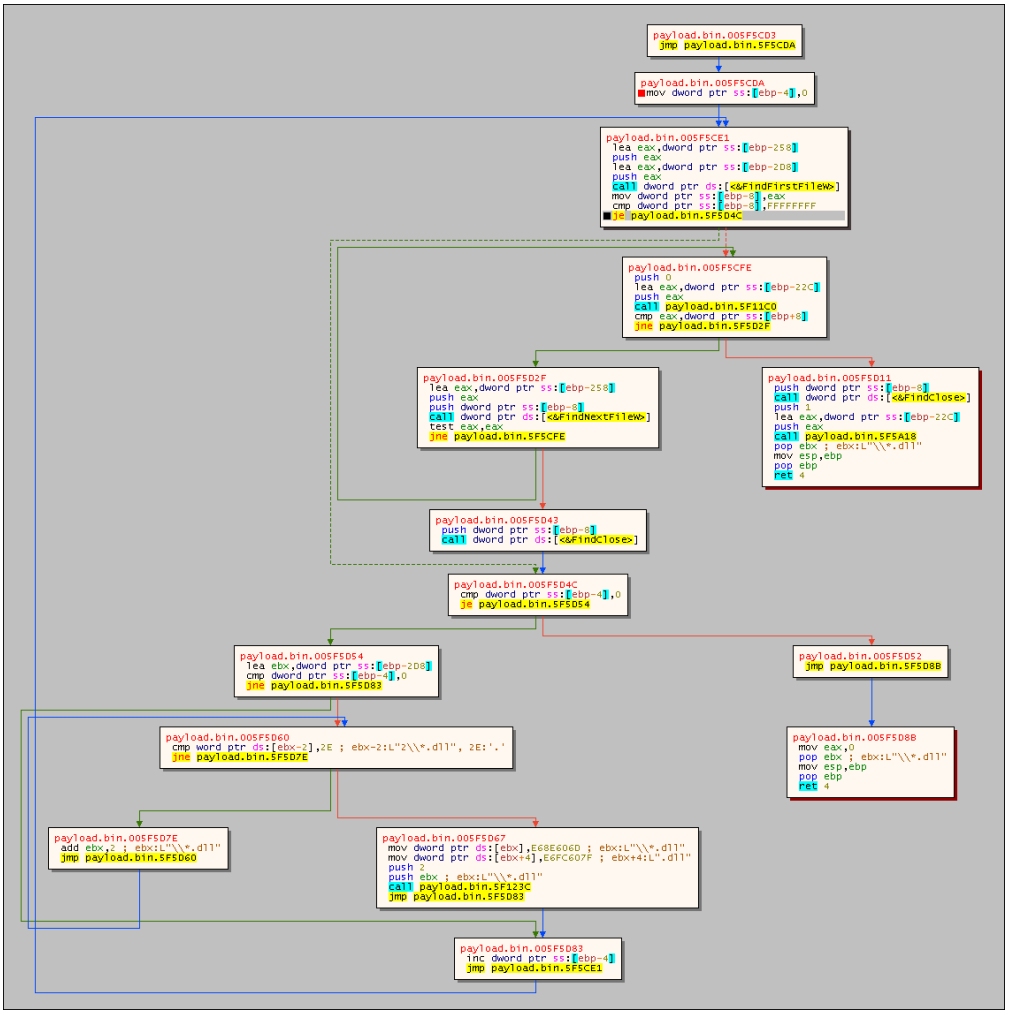
MD5	bbe63d8efc8d8dc7f387b08ee07721ba
SHA256	2e8aaa6338cbf95d8d268559fb8afac64e1c0dfc9ded4bb2de63a9db634e354d
File Type	PE32/EXE

Şekil 2 General File Information

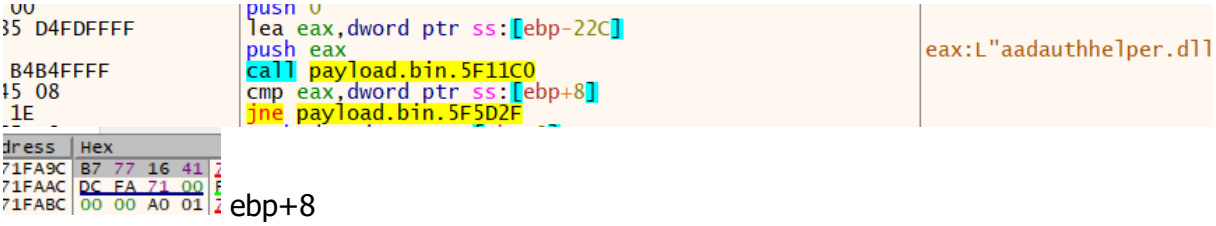
```
lea eax,dword ptr ss:[ebp-258]          eax:L"C:\\Windows\\System32\\*.dll"
push eax
lea eax,dword ptr ss:[ebp-2D8]          eax:L"C:\\Windows\\System32\\*.dll"
push eax
call dword ptr ds:[<&FindFirstFileW>]
mov dword ptr ss:[ebp-8],eax
cmp dword ptr ss:[ebp-8],FFFFFFFF
je payload.bin.5F5D4C
push 0
```

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

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

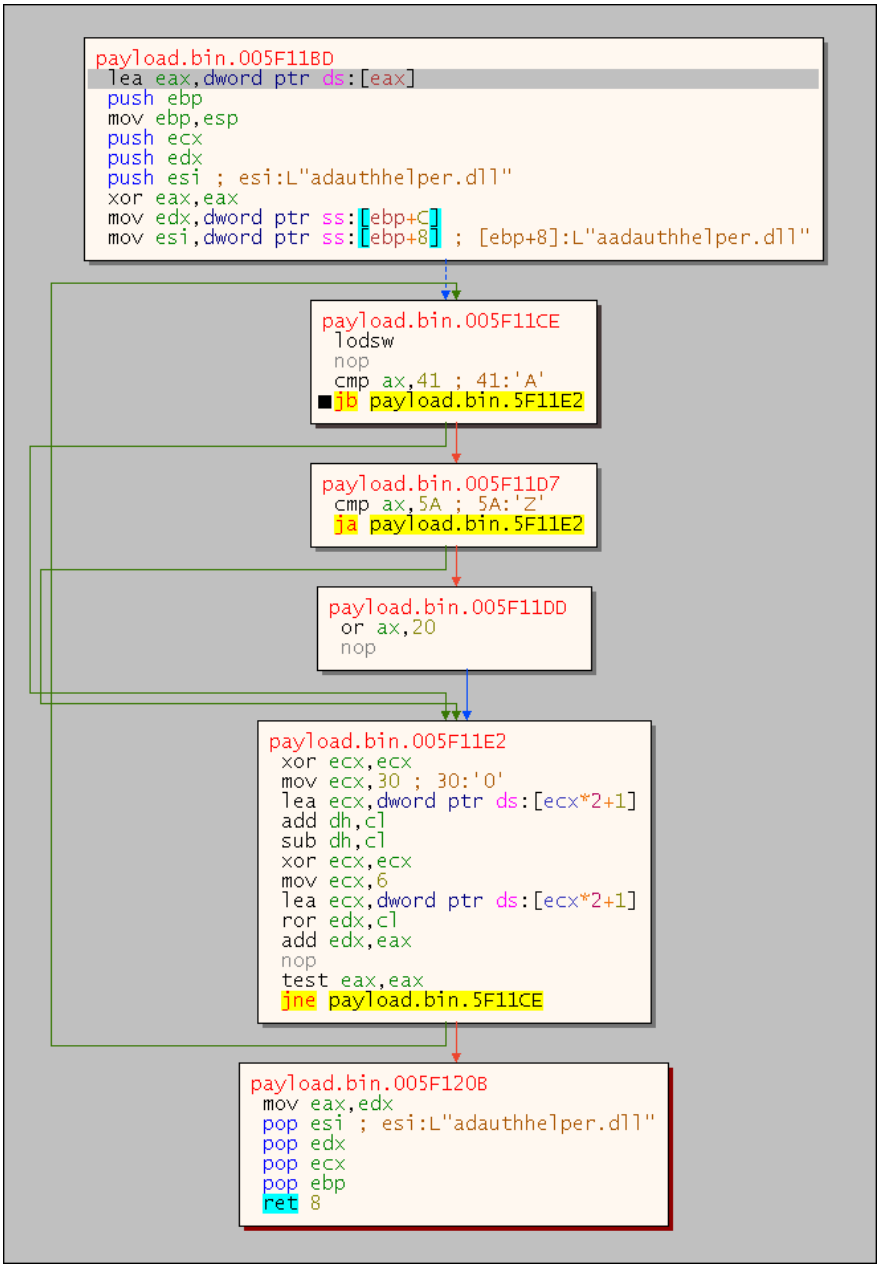


Şekil 4 Traversal Algorithm



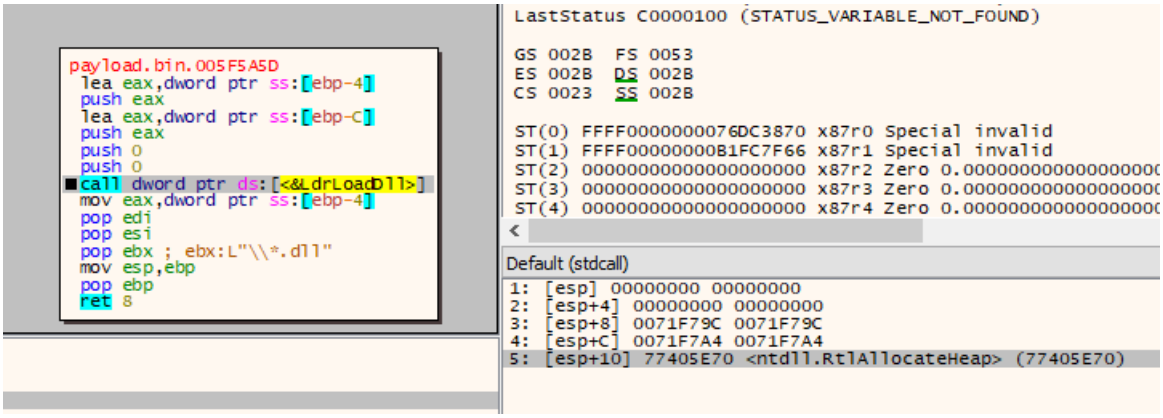
Şekil 5 Hash Generating and Comparing

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

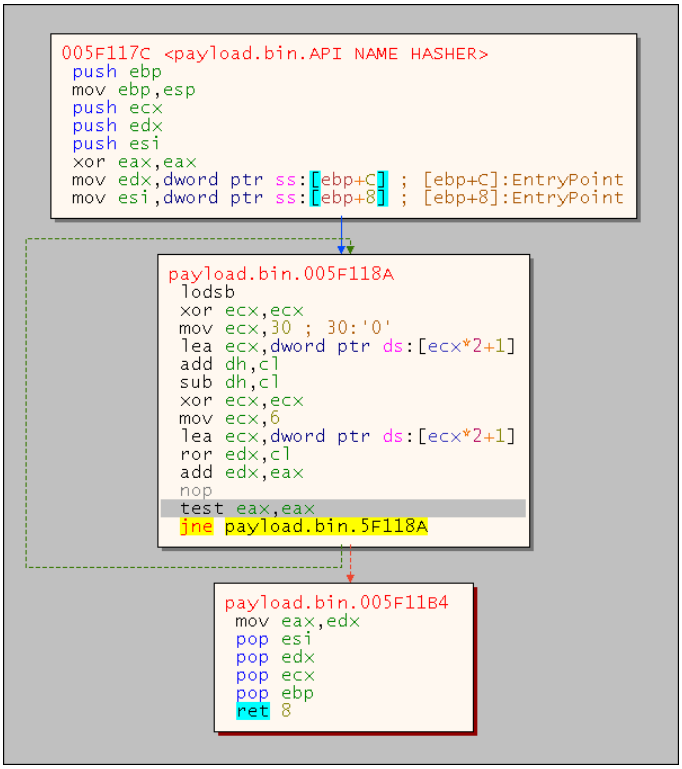
Aranılan DLL dosyasına ait hash bilgisinin **"41 16 77 B7"** olduğu ve bunun **ntdll.dll** dosyasına ait olduğu tespit edilmiştir.



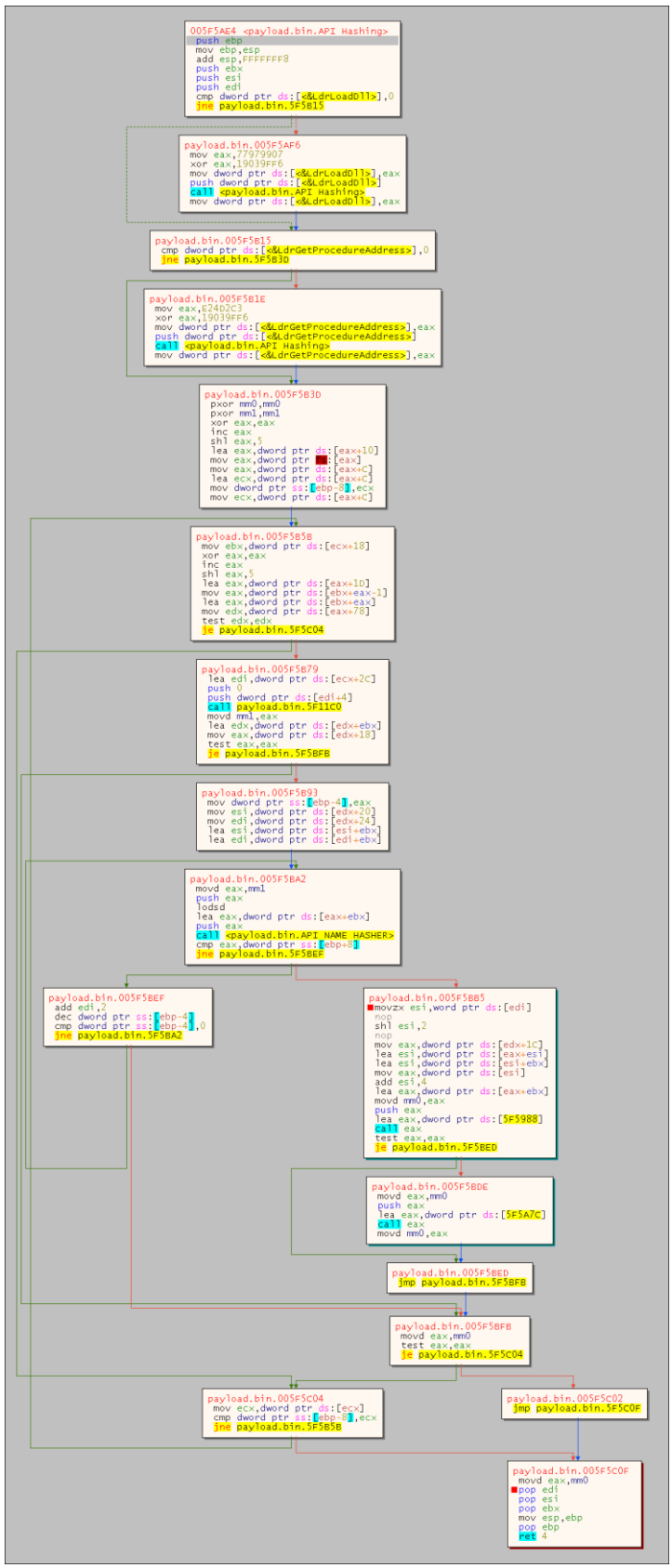
Şekil 7 LdrLoadDll

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.

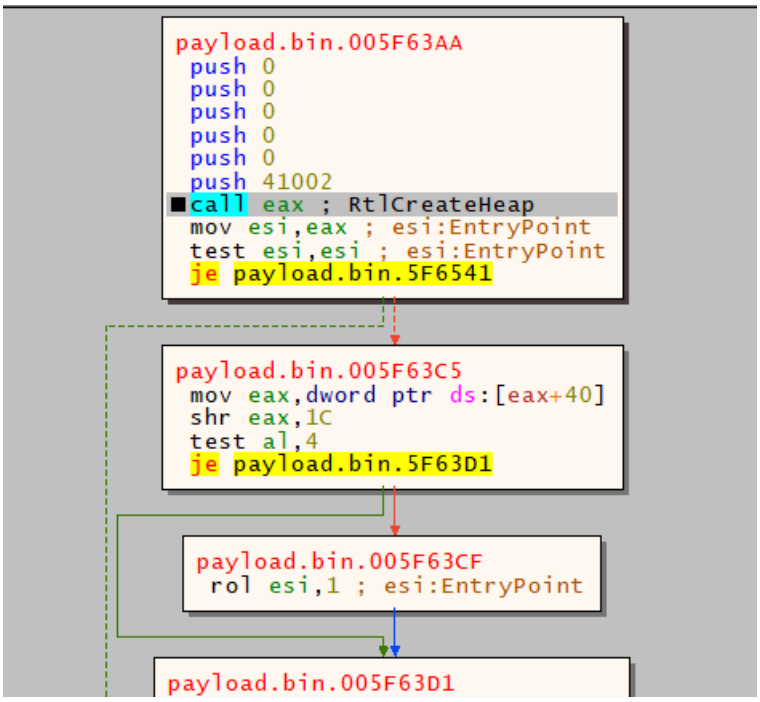


Şekil 8 API Name Hashing Algorithm

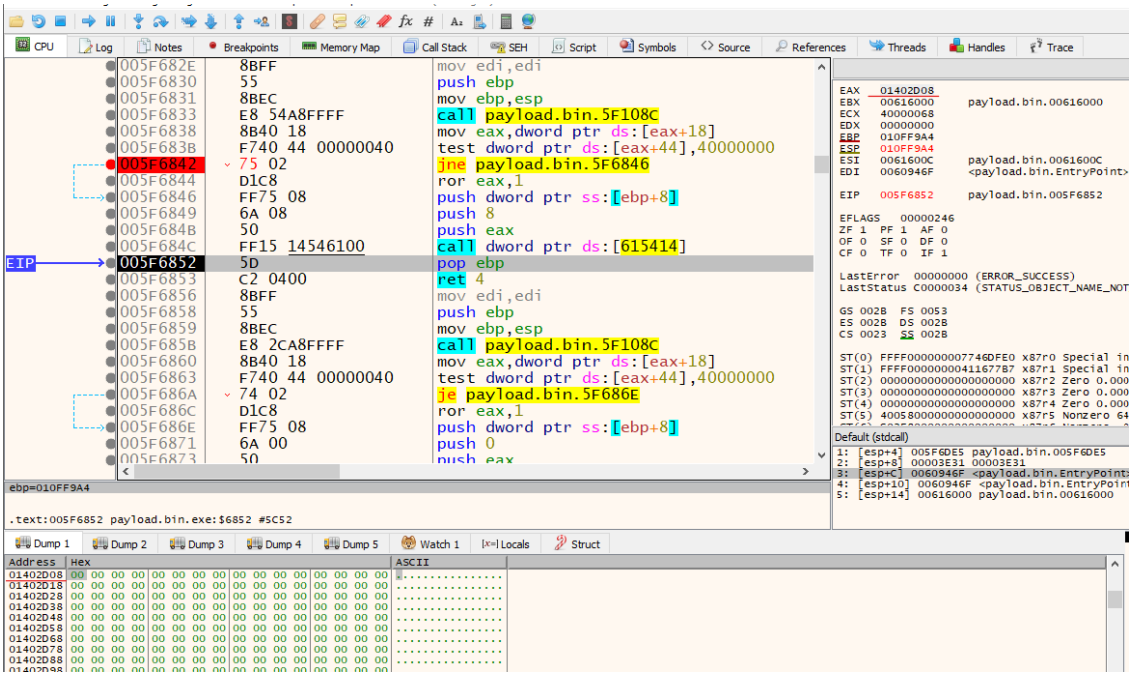


Şekil 9 API Hashing Algorithm

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

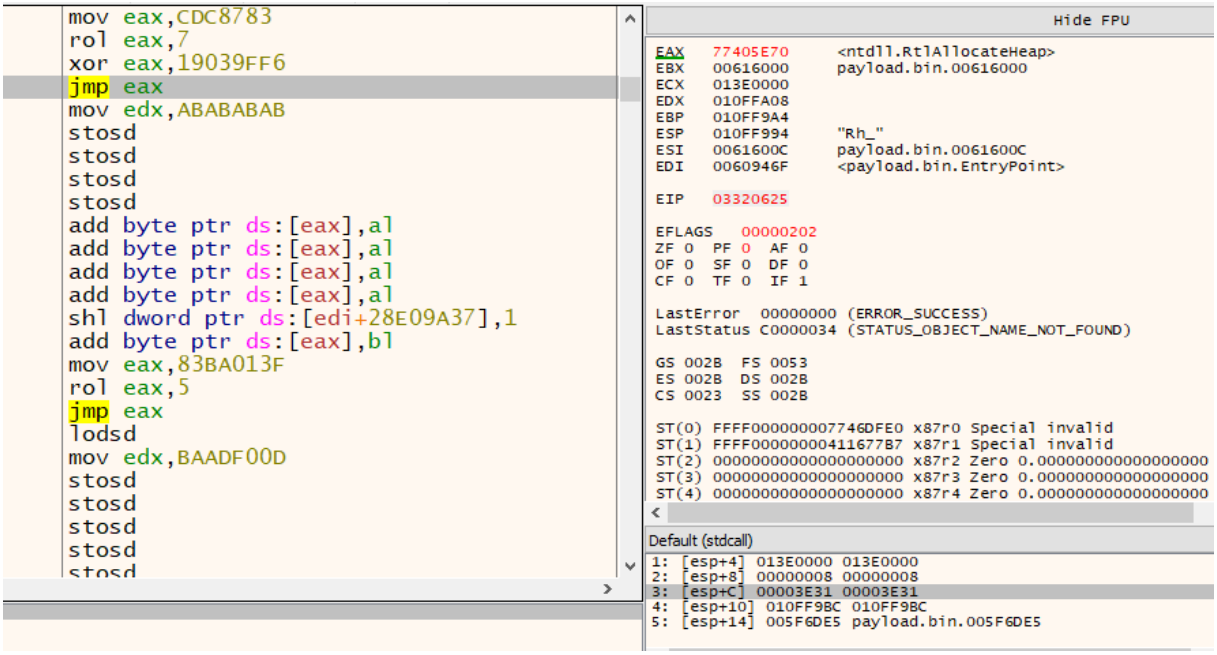


Şekil 10 Anti-Debug: Heap Based



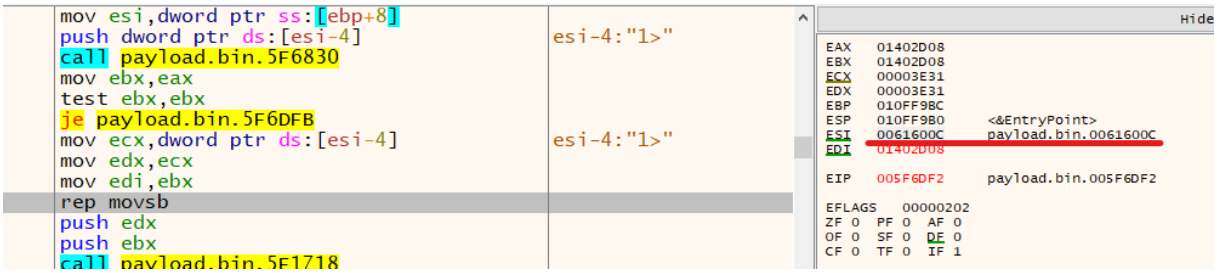
Şekil 11 Anti-Debug: Heap Based

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



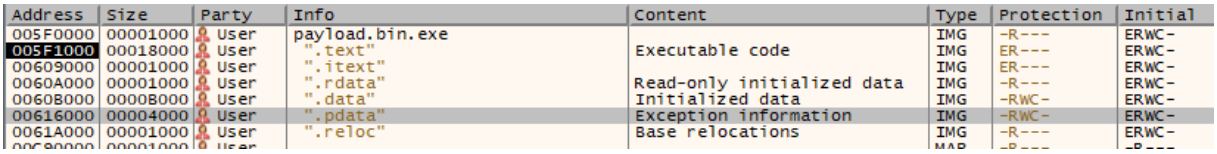
Şekil 12 RtAllocateHeap: 15921 byte

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



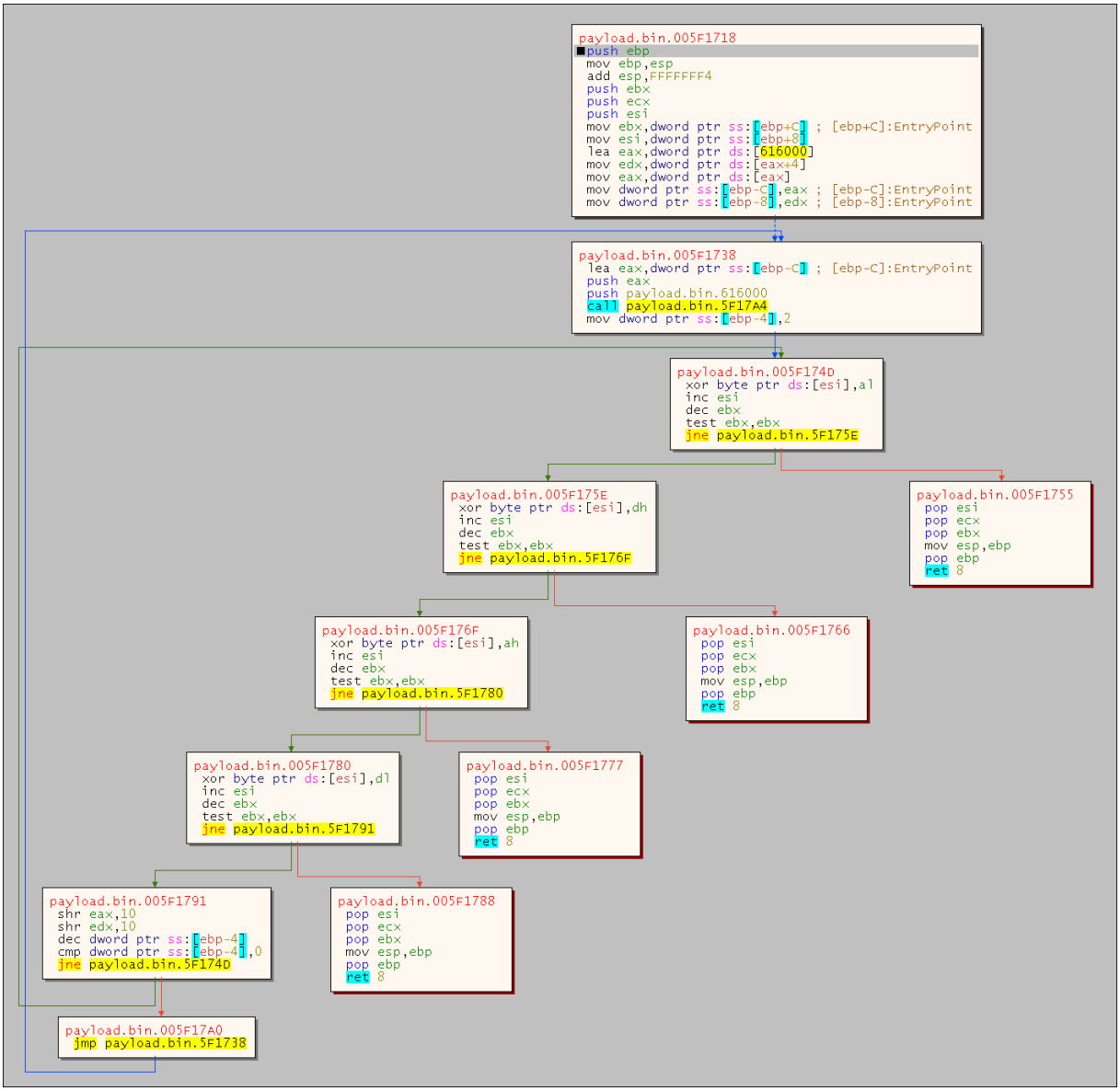
Şekil 13 Writing .pdata section

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



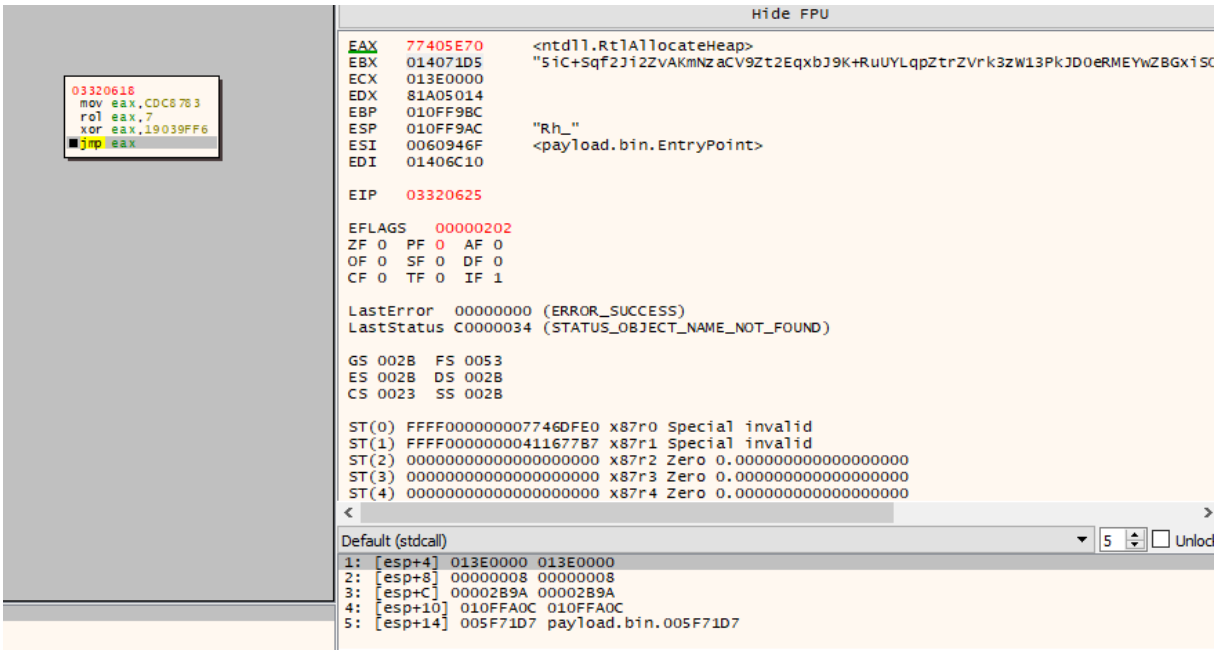
Şekil 14 .pdata section

Söz konusu section incelendiğinde şifrelenmiş olduğu görülmektedir.



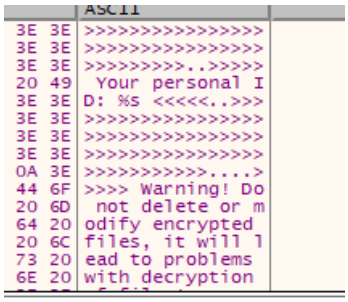
Şekil 15 Decryption of .pdata

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



Şekil 16 RtlAllocateHeap

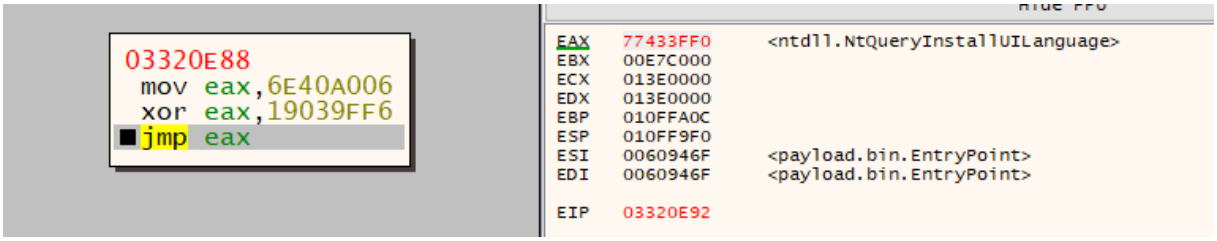
11KB'lik bir alan ayrıldığı ve ayrılan alan içerisine çözümlenen verilerin yazıldığı tespit edilmiştir.



Şekil 17 README.txt Content Decryption

Çözümlenen veri incelendiğinde, sonradan oluşturulacak README.txt dosyasının içeriği olduğu görülmüştür. Zararlının her bilgisayar için ID oluşturduğu tespit edilmiştir.
VictimID yapısı: "BD23223ABCFA78BC"+<randomly_generated_16_character>

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

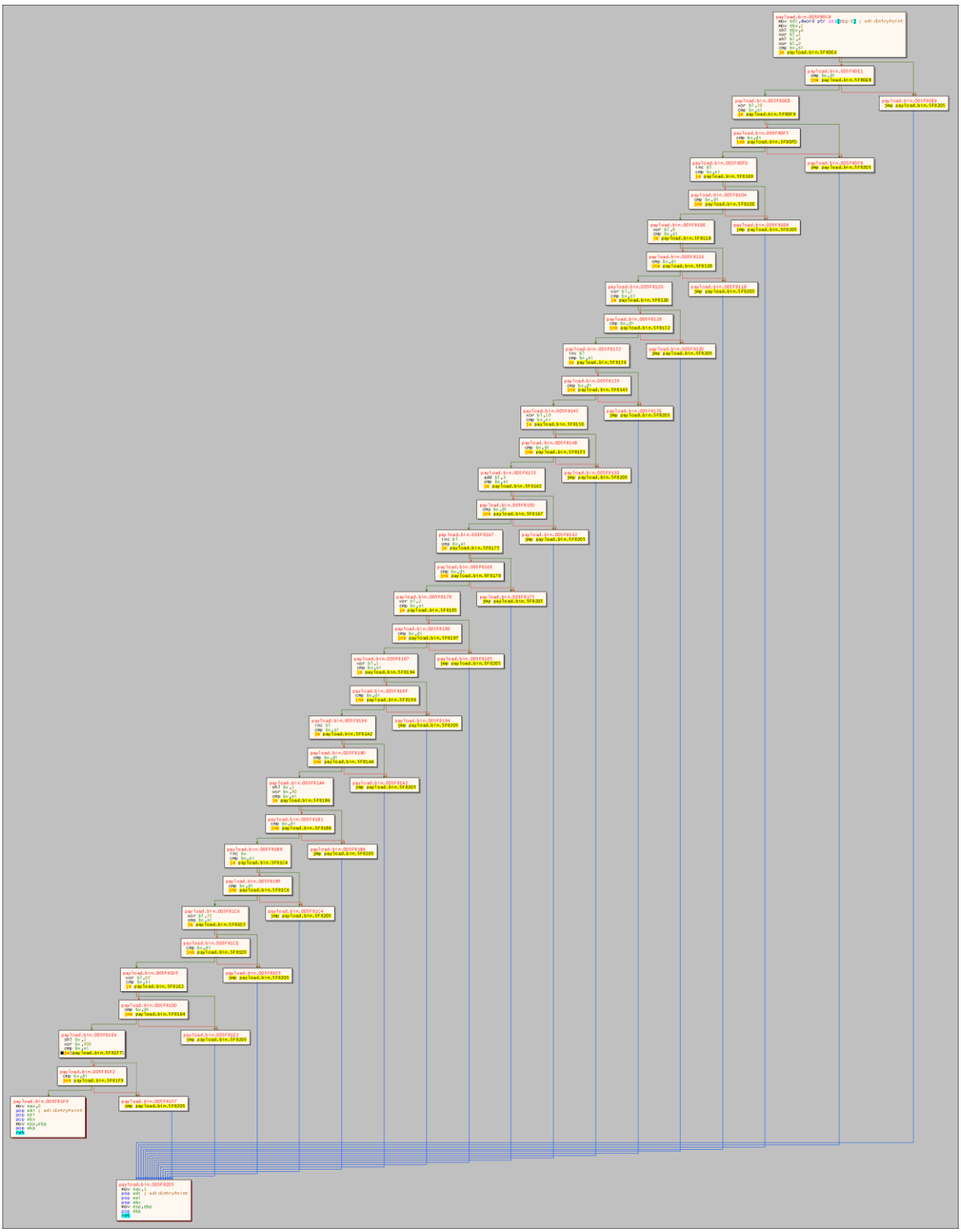


Şekil 18 NtQueryInstallUILanguage

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

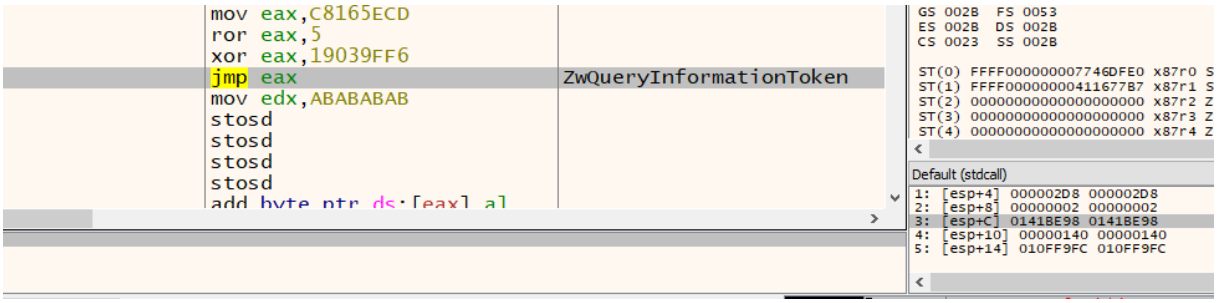


Şekil 19 Country Checking

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

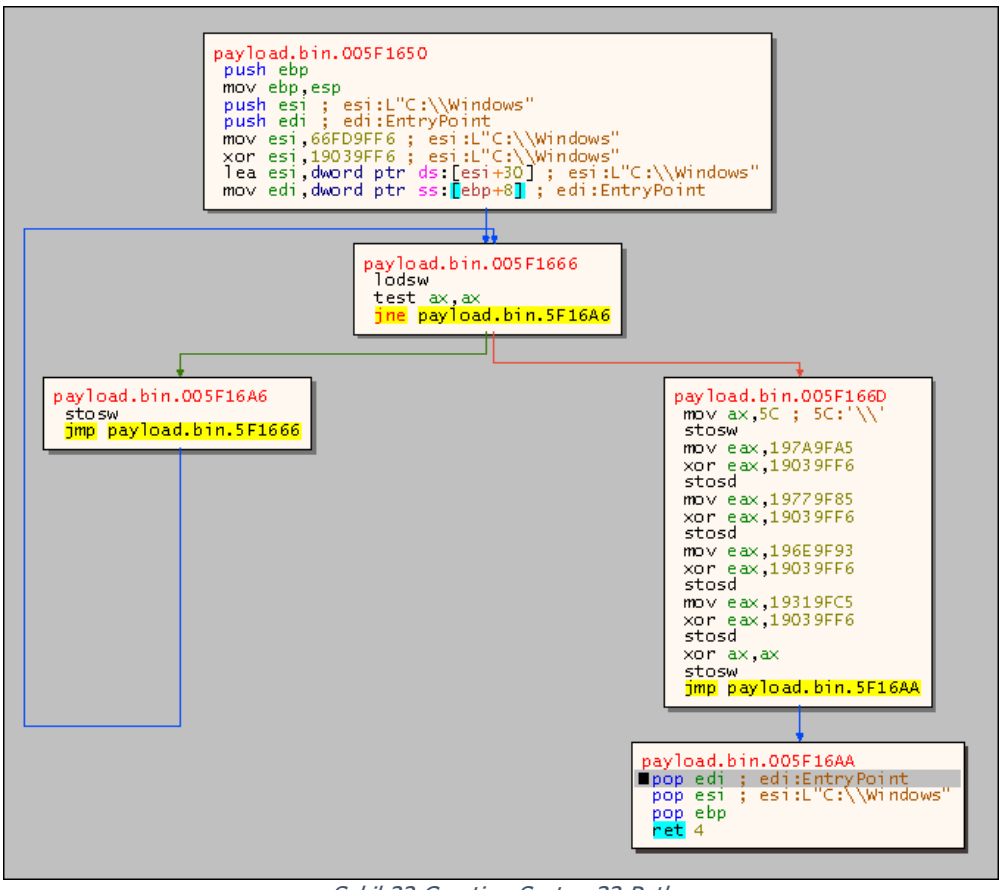
Şekil 20 NtOpenProcessToken

Sürecin kendisine ait Access token tutamacının alındığı tespit edildi.



Şekil 21 ZwqueryInformationToken

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



Şekil 22 Creation System32 Path

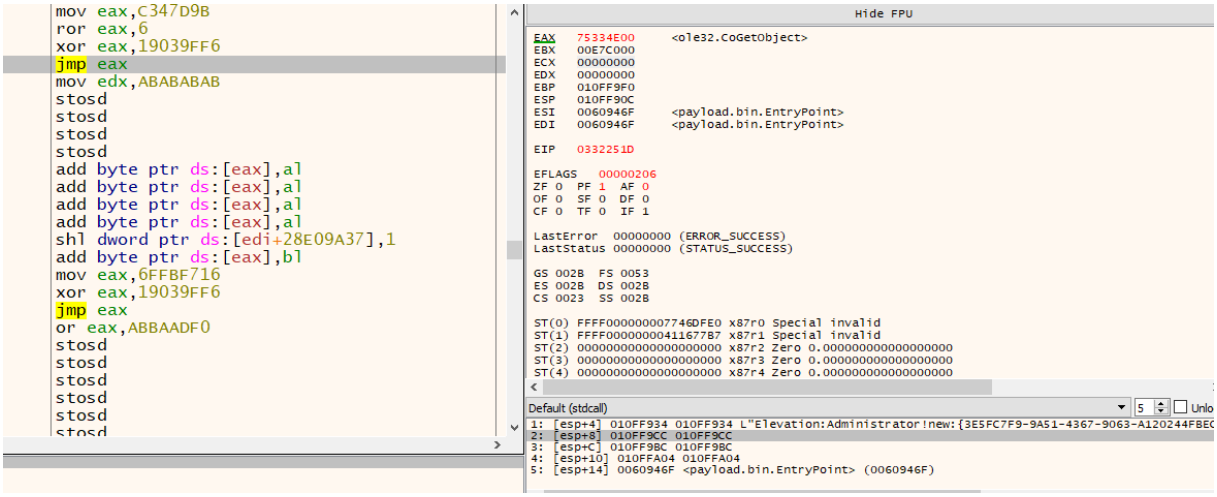
Şekil 22'te **"C:\\Windows\\System32"** ifadesi oluşturulmaktadır. XOR yöntemi kullanılarak güvenlik ürünlerinden kaçınıldığı görülmektedir.

Şekil 23 Decryption Algorithm

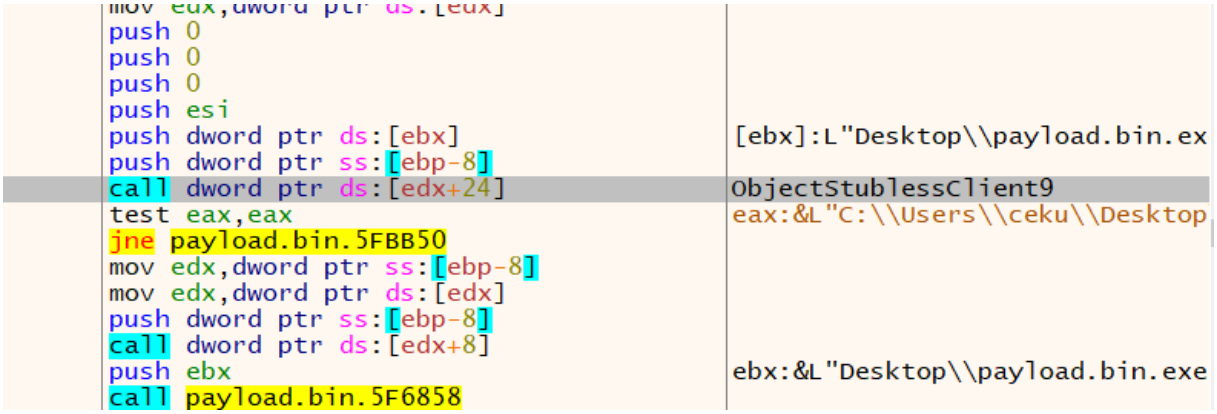
Şekil 23'te gösterilen algoritma ile çözümlenen ifadeler şunlardır:

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

```
DWORD* decryption_function(DWORD *arry,size_t size) {

	for (int i = 0; i < size; i++) {

		arry[i] = arry[i] ^ 0x19039ff6;
		arry[i] = ~(arry[i]);

		std::cout << std::hex<< arry[i]<<" ";

	}
	return arry;
}
```



Şekil 24 CoGetObject



Şekil 25 ObjectStublessClient9

UAC Bypass yaparak dllhost.exe altında child process şeklinde çalıştığı tespit edilmiştir.

Önceden çözümlenen verilerin ayrılan bir parçasının MD5 yöntemi ile hash'lendiği görülmüştür.

MD5 ile hash'lenen veri şu şekildedir:

```
3a2223bd  bc78fabc  bb04ea7f  3286dcc7
0860c7ff  c03edf06  3e570a04  9c55aaee
6a051e98  96cd73c8  d17595fc  a1ad958b
fa52e8cc  8b65411c  587767a2  fd5a5db2
809964a8  0cf2a551  be0e3392  1b07e687
ec4c1f53  605e4e11  293dbfd1  5540bb91
b186938a  9c496dae  d13d64ea  a6577138
4b9adcb3  c985d873  11100549  5892daf0
```

Hemen ardından yapılan bir başka çözümleme sonucunda ortaya çıkan ifade şu şekildedir:
"{{%08X-%04X-%04X-%02X%02X-%02X%02X%02X%02X%02X%02X}}"

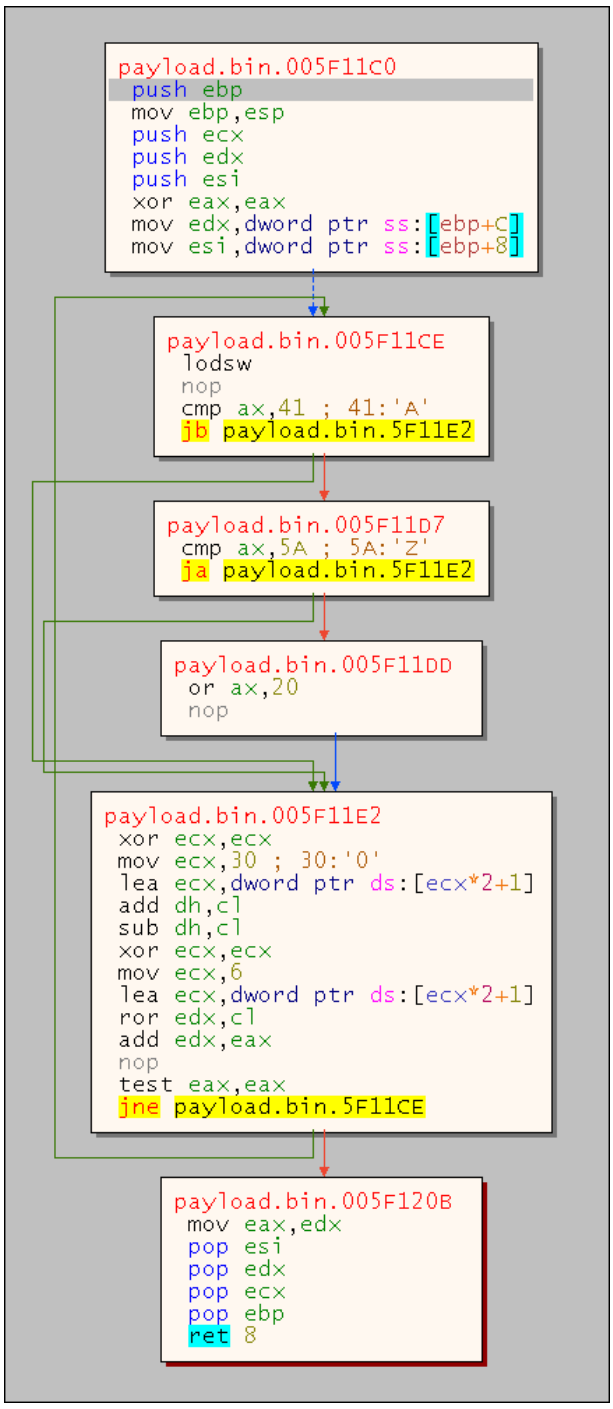
MD5 hash belirtilen formatta yukarıdaki metne entegre edilmektedir.

```
jmp eax
or eax,ABBAADF0
stosd
stosd
stosd
stosd
```

```
Hide FPU
EAX 77432E50 <ntdll.ZwQuerySystemInformation>
EBX 00000000
ECX 40000068
EDX 00000000
EBP 0362F8E8
ESP 0362F8C0
```

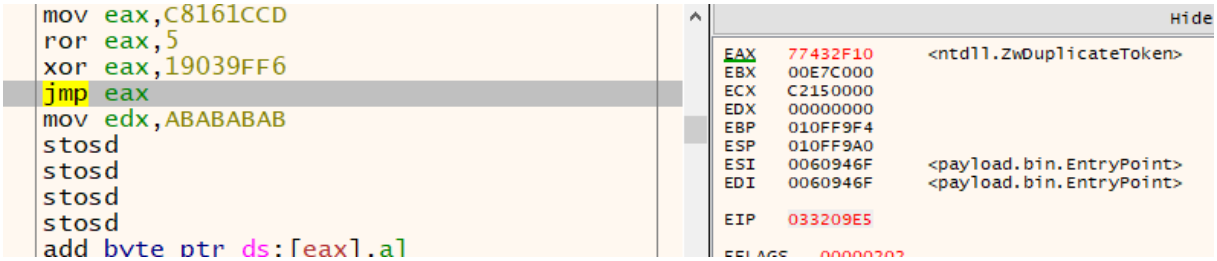
Şekil 26 ZwQuerySystemInformation

Sistem üzerinde çalışmakta olan process bilgilerinin çekildiği tespit edilmiştir.



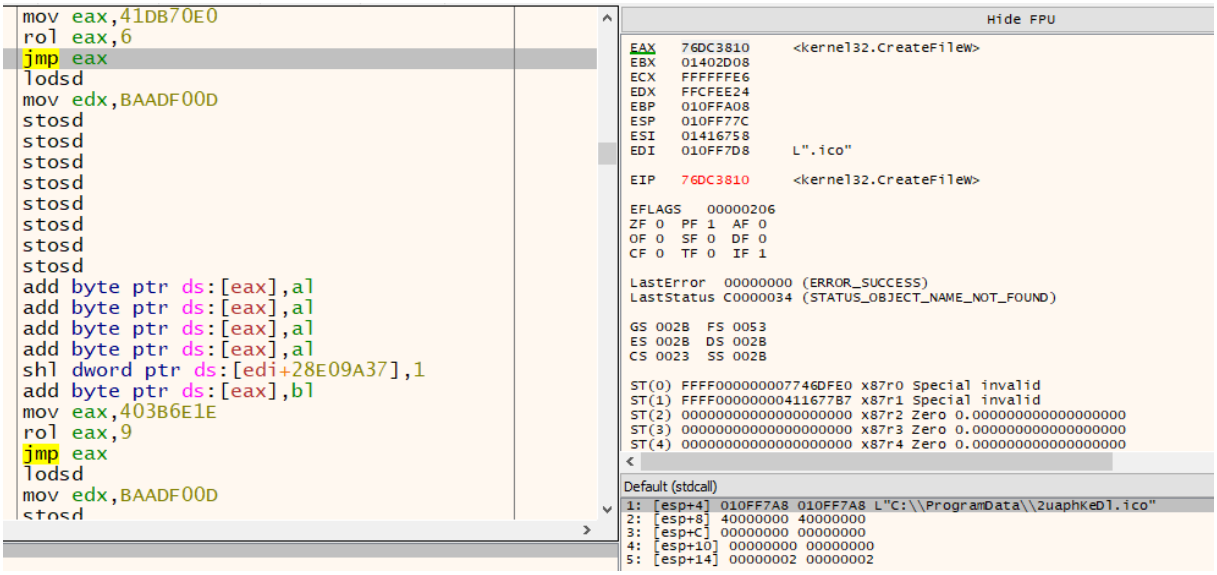
Şekil 27 Process Name Hashing Algorithm

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.



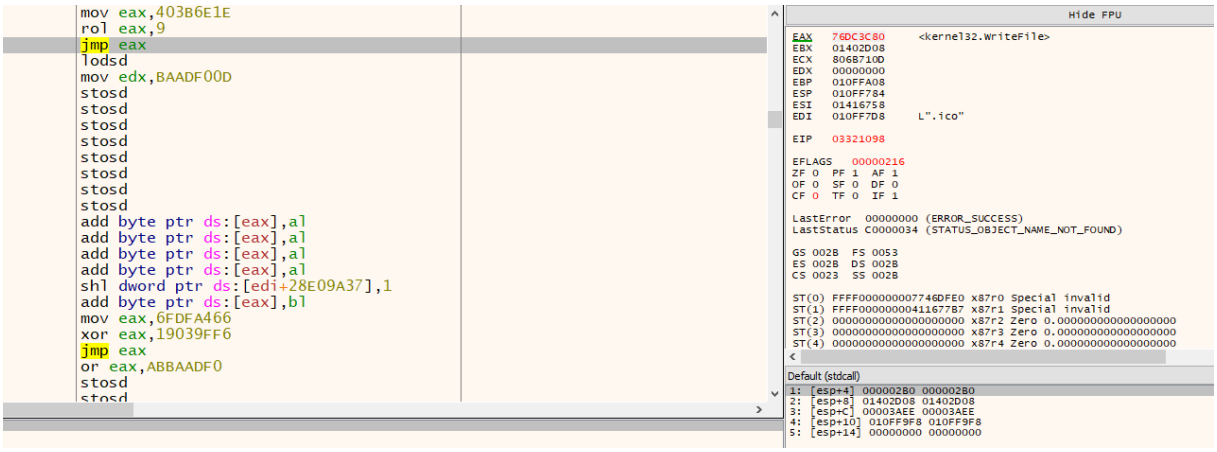
Şekil 28 DuplicateToken

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



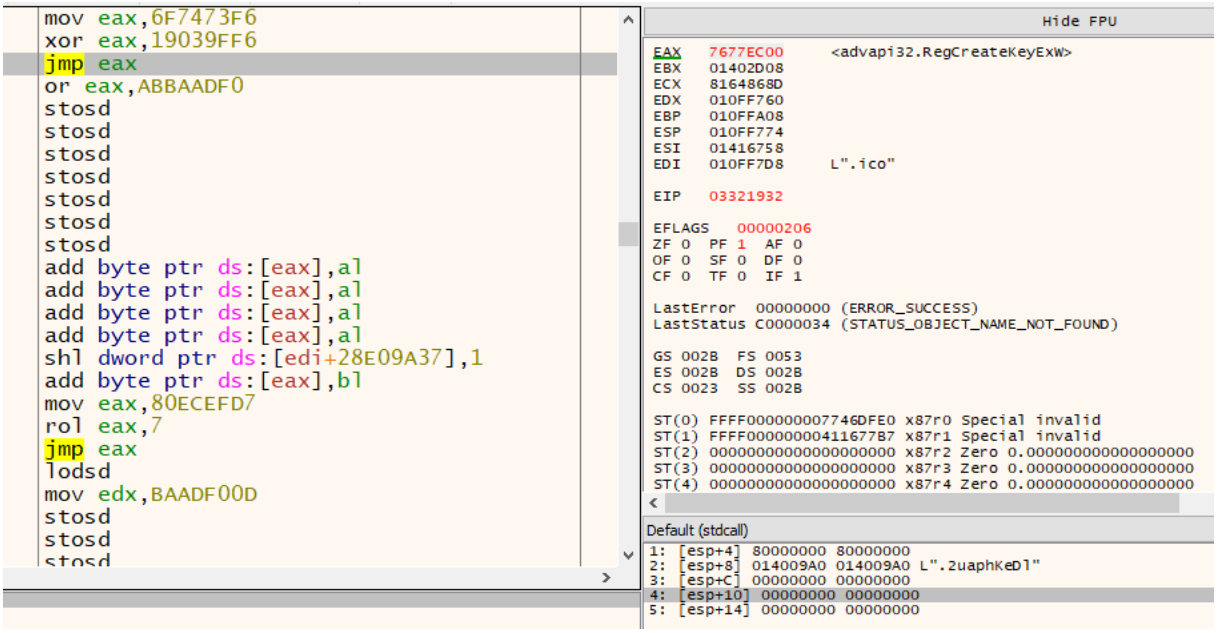
Şekil 29 CreateFile: Creation LockBit Icon File

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



Şekil 30 WriteFile: LockBit Icon

Meşhur LockBit dosya ikonunu içeriğinin yazıldığı tespit edildi.

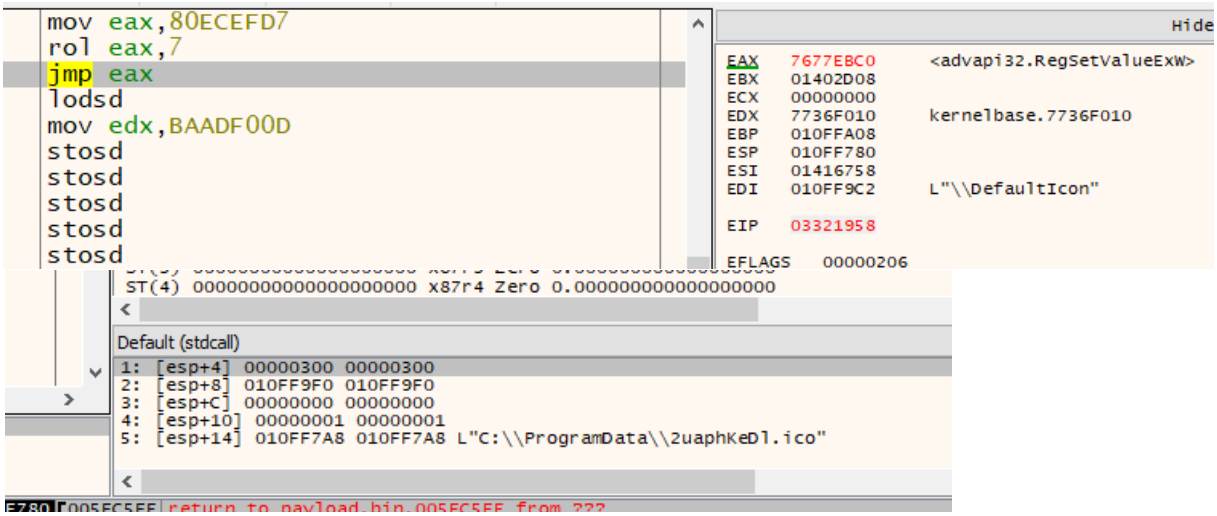


Şekil 31 RegCreateKeyExW

HKEY_CLASSES_ROOT anahtarının altında **".2uaphKeDl"** adında alt anahtarı oluşturulduğu tespit edildi.

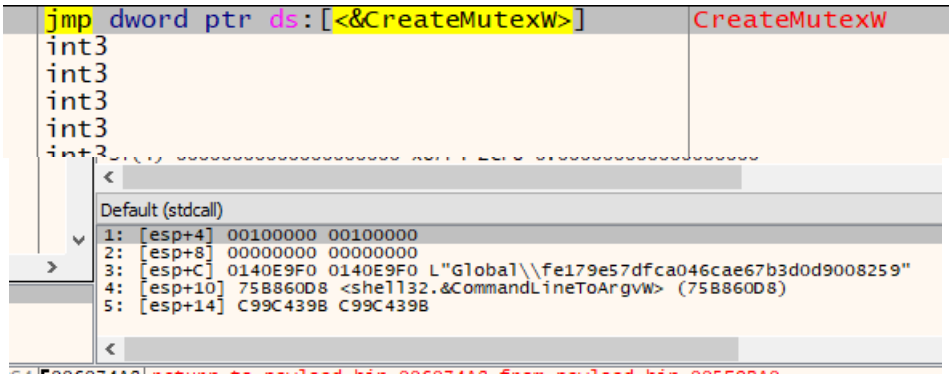
Şekil 32 RegCreateKeyExW: DefaultIcon

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



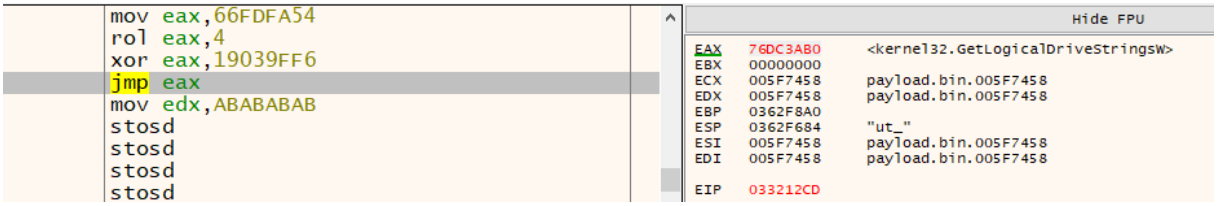
Şekil 33 RegSetValueExW: Setting Icon File Path

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



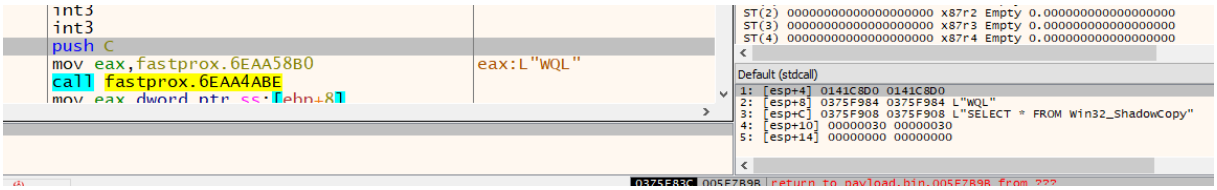
Şekil 34 CreateMutex

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



Şekil 35 GetLogicalDriveStringsW

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

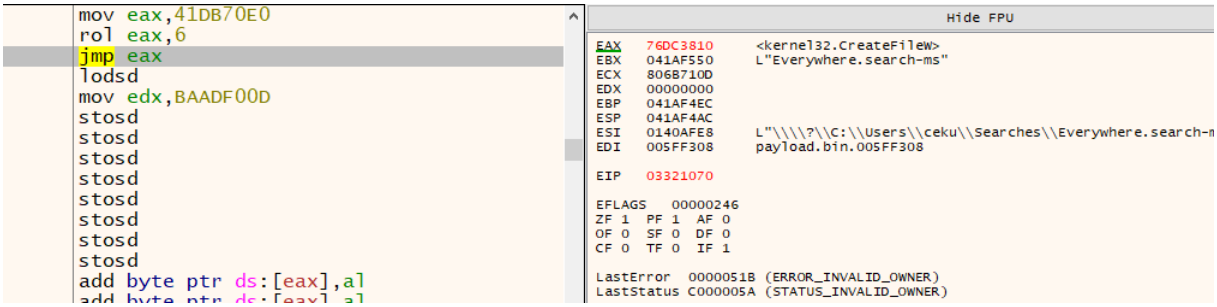


Şekil 36 Delete Shadow Copies

Shadow kopyalarını toplamak için WMI sorgusu çalıştırıldığı tespit edildi.

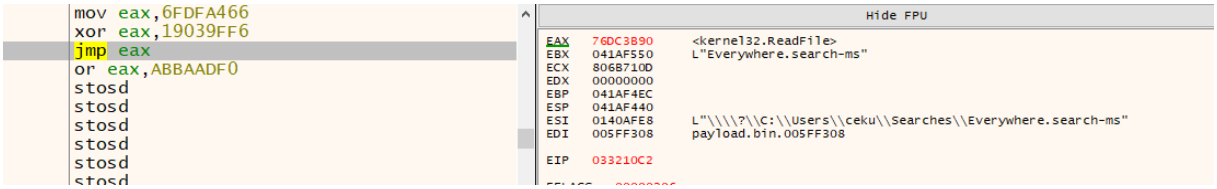
Ardından, **"C:\\"** dizininden başlayarak özellikle users dizini altındaki dizinleri gezerek dosyaları şifrelemektedir.

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

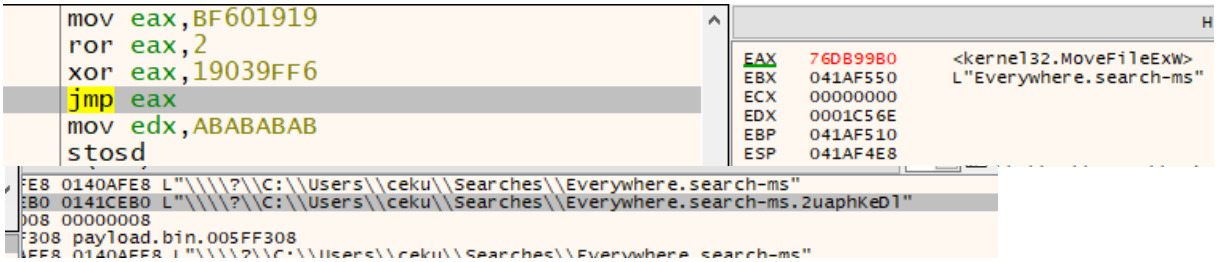


Şekil 37 Opening File that will Encrypt

Tespit edilen dosyanın açılması,

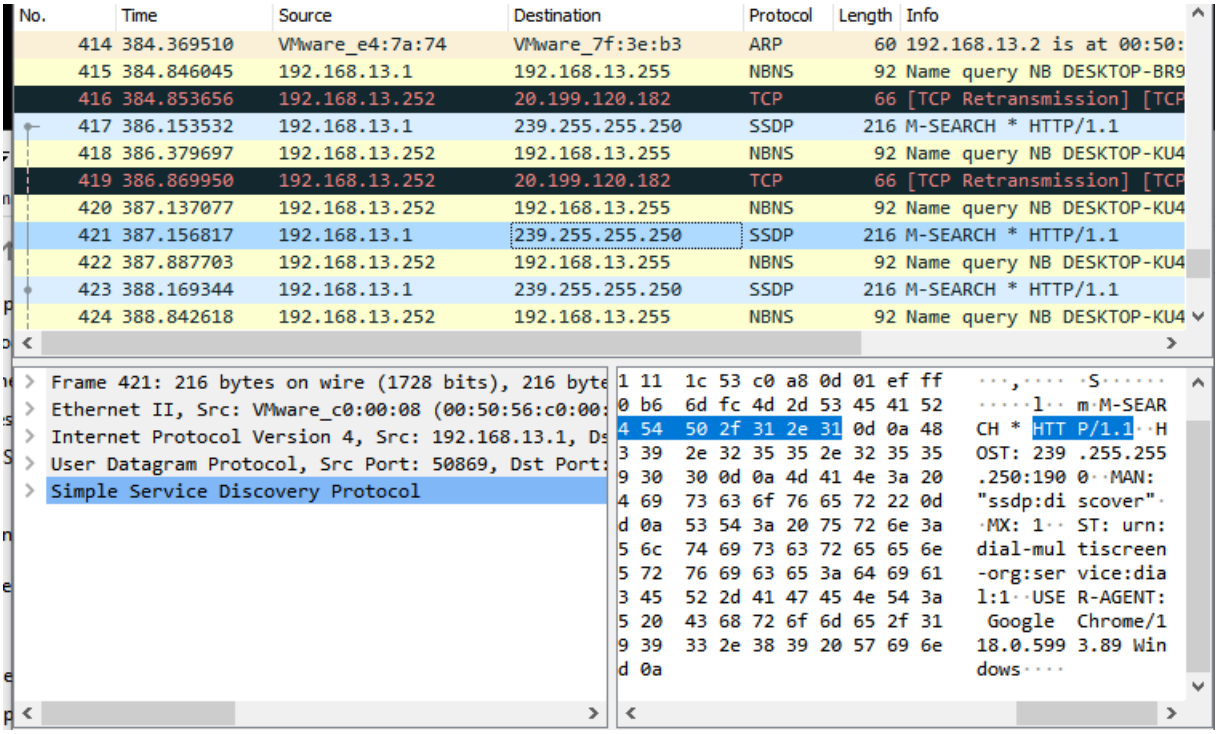


Okunması,



Şekil 38 MoveFileExW

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



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

```yaml
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
Privilege Escalation	T1548.002	Abuse Elevation Control Mechanism: Bypass User Account Control
Privilege Escalation	T1134	Access Token Manipulation
Discovery	T1083	File and Directory Discovery
Discovery	T1069.002	Permission Groups Discovery: Domain Groups
Discovery	T1082	System Information Discovery
Execution	T1047	Windows Management Instrumentation


ECHO
CYBER THREAT INTELLIGENCE