Amadey

TEKNİK ANALİZ RAPORU

ZAYOTEM
ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

İçindekiler

Ön Bakış

Botnet olarak ortaya çıkan Amadey, Rus Hack forumlarında ilk kez Kasım 2018 tarihlerinde görülmüştür. Düzenli olarak sistem hakkındaki bilgilerini ve antivirüs yazılımlarını komuta kontrol sunucusuna aktarmaktadır. Asıl işlevi ise, ele geçirilen sistemlerin hepsine veya **özel olarak hedeflenenlere** başka payloadlar yüklemektir.

Enfekte edilen cihazlardan elde edilen bilgilerden bazıları şunlardır;

- Kayıt defteri manipülasyonu,
- Cihaz özellikleri ve dosya bilgileri,
- Güvenlik uygulamaları,
- İşletim sistemi bilgisi

Nbveek.exe Analizi

Adı	Nbveek.exe
MD5	77e0a0a90e0231493bd421f4cdab0668
SHA256	75520c76a4051b2be15db8625f35d4c1c63d93686bf849e6fc67f4e62d2fd000
Dosya Türü	PE32 / EXE

Statik Analiz

İlk bakışta zararlı dosyada stringler içerisinde kullanılan API'ler dışında **base64** değerler göze çarpmaktadır.

```
.rdata:00433430  00000011  C  KQKkIcN JIhBGC==
.rdata:00433444  00000011  C  9gy4UM0wHo06SS==
.rdata:00433458  0000000D  C  RUCKQuKYO3R=
.rdata:00433468  00000029  C  IUCAUMK5VUFxN2DdTYaKNWMHMyZJKABAFwYWO9F=
.rdata:00433494  00000009  C  FwYWP9Fn
.rdata:004334A0  00000009  C  FcpxM7==
.rdata:004334B0  00000045  C  RUYIQviGQm0eMYcgcqYv43U21B4I3oSyaXCeLT0D9o0w7HQicrCI430eYjLqK46mWK==
.rdata:004334F8  0000005D  C  RUYIQviGQm0eMYcgcqYv43U21B4I3oSyaXCeLT0D9o0w7HQicrCI430eVT8s3I BWRyeQNSq9kGV...
.rdata:00433558  0000000D  C  RXGj8dW69C==
.rdata:00433570  00000019  C  VWQmDsdIEHOPKGcPIGYvD4zi
.rdata:0043358C  0000003D  C  RUYIQviGQm0eMYcgcqYv43U21B4I3oSyaXCeLT0D9o0w7HQicrCI430eYjLq
.rdata:004335CC  0000000D  C   hKwUwCxIBNi
.rdata:004335DC  00000015  C  FwYGUMCq IZiE3ILIGx=
.rdata:004335F4  0000000D  C  RByxUTOm8ZR=
.rdata:00433608  00000059  C  RUYIQviGQm0eMYcgcqYv43U21B4I3oSyaXCeLT0D9o0w7HQicrCI430eVT8s3I BWRyePSmq8IBiK...
.rdata:00433664  00000015  C  GPKVMLOVQmdILWoCJU==
.rdata:0043367C  00000009  C  Uyuy8q==
```

Görsel 1- IDA strings

Base64 olduğu düşünülen değerlerin bazıları decode edildiğinde "**)p^G\ė=" , "'"黸Q{+d*" , "&#" , ")!IF**" şeklinde anlamsız değerler çıkmaktadır. Bu değerler belirlenmiş bir anahtar kullanılarak **runtime anında anlamlandırılmaktadır**.

```
v25 = (const CHAR *)lpFileName;
if ( v60 >= 0x10 )
  v25 = lpFileName[0];
v26 = GetFileAttributesA(v25);
if ( v26 == -1 || (v26 & 0x10) == 0 )
{
  v27 = (const CHAR *)lpFileName;
  v45 = 0;
  if ( v60 >= 0x10 )
    v27 = lpFileName[0];
  CreateDirectoryA(v27, (LPSECURITY_ATTRIBUTES)v45);
}
v28 = (const CHAR *)lpFileName;
if ( v60 >= 0x10 )
  v28 = lpFileName[0];
v29 = GetFileAttributesA(v28);
if ( v29 != -1 && (v29 & 0x10) != 0 )
{
  v44 = (LPCSTR)sub_415850(FileName);
  v37 = (const CHAR *)sub_415850(v61);
  CopyFileA(v37, v44, 1);
  if ( (unsigned __int8)sub_405020((char *)FileName) )
  {
    v58 = &v40;
```

Görsel 2- IDA pseudo kod parçası

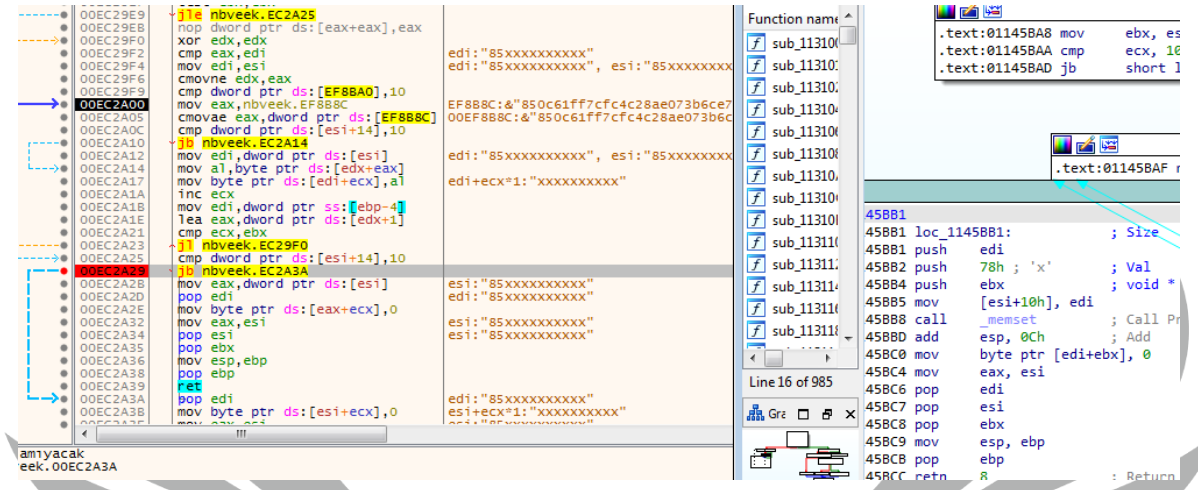
Burada bir dizin içerisinde önce klasörü **arayıp**, yoksa **oluşturulduğunu** ardından o klasör altında bir dosyayı **arayıp** yoksa içerisine elde tutulan **dosyanın kopyalandığı** görülmektedir.

```
:00403A4A push    offset a1       ; "1"
:00403A4F call    sub_416B10      ; Call Procedure
:00403A54 add     esp, 8          ; Add
:00403A57 xor     edx, edx        ; Logical Exclusive OR
:00403A59 test    al, al          ; Logical Compare
:00403A5B mov     ecx, 5
:00403A60 cmovz   ecx, edx        ; Move if Zero (ZF=1)
:00403A63 mov     edx, offset aRunas ; "runas"
:00403A68 push    ecx             ; Size
:00403A69 mov     ecx, offset Src
:00403A6E cmovnz  ecx, edx        ; Move if Not Zero (ZF=0)
:00403A71 push    ecx             ; Src
:00403A72 lea     ecx, [ebp+lpOperation] ; void *
:00403A75 call    sub_415CD0      ; Call Procedure
:00403A7A cmp     [ebp+arg_44], 10h ; Compare Two Operands
:00403A7E lea     edx, [ebp+lpParameters] ; Load Effective Address
:00403A81 push    0               ; nShowCmd
:00403A83 cmovnb  edx, [ebp+lpParameters] ; Move if Not Below (CF=0)
:00403A87 lea     ecx, [ebp+lpFile] ; Load Effective Address
:00403A8A cmp     [ebp+arg_2C], 10h ; Compare Two Operands
:00403A8E lea     eax, [ebp+lpOperation] ; Load Effective Address
:00403A91 push    0               ; lpDirectory
:00403A93 cmovnb  ecx, [ebp+lpFile] ; Move if Not Below (CF=0)
:00403A97 cmp     [ebp+arg_14], 10h ; Compare Two Operands
:00403A9B push    edx             ; lpParameters
:00403A9C cmovnb  eax, [ebp+lpOperation] ; Move if Not Below (CF=0)
:00403AA0 push    ecx             ; lpFile
:00403AA1 push    eax             ; lpOperation
:00403AA2 push    0               ; hwnd
:00403AA4 call    ds:ShellExecuteA ; Indirect Call Near Procedure
```

Görsel 3- IDA görünümü

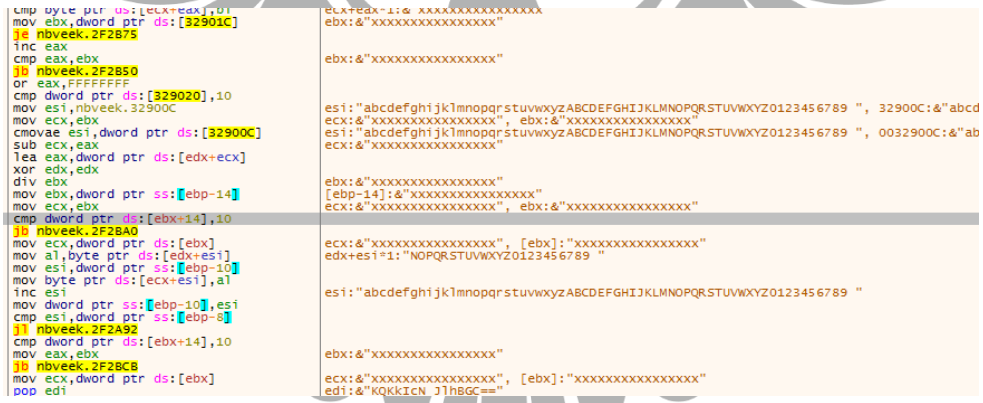
Dosya kopyalama işleminin hemen ardından kopyalanan dosyanın burada “**runas”** parametresi ile çalıştırıldığı açıkça görülmektedir.

Dinamik Analiz



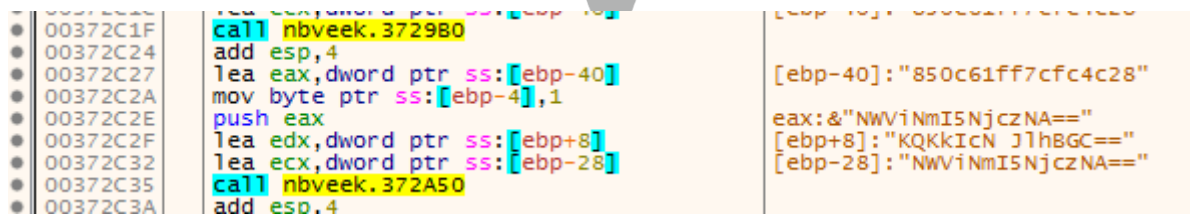
Görsel 3- x32dbg

Anahtar görevi gören değer, **çözülecek değerin boyutuna** ayarlanmaktadır.



Görsel 4- x32dbg metin çözümleme işlemi

Elde edilen değer ile çözülecek değer bir takım işlemlerin ardından **anlamlı base64** değerlerini oluşturmaktadır.

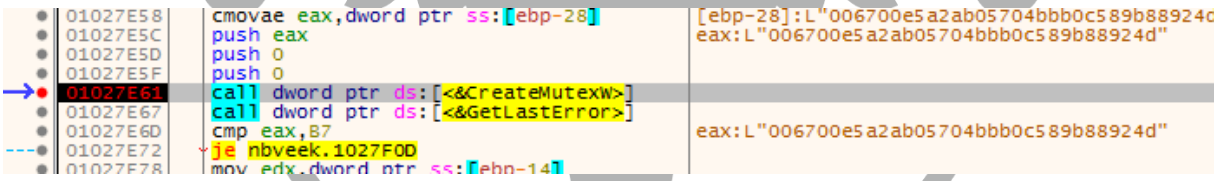


Görsel 5- x32dbg çözümlenmiş örnek

KQKkIcN JlhBGC== // NWViNmI5NjczNA== // **5eb6b96734 (klasör adı)**

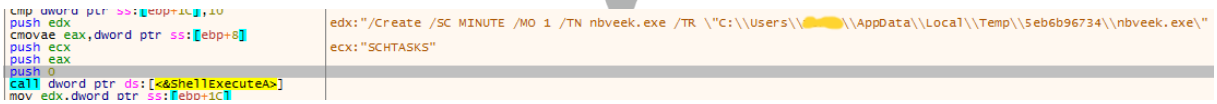
Görsel 6- AntiDebug

GetModuleFileNameA Api'si kullanılarak çalışan dosya konumu almaktadır. Öncesinde kendini kopyaladığı yer ile de bunu kıyaslamaktadır. Eğer aynı yerde çalışmıyorsa kendini kapatmaktadır.



Görsel 7- x32dbg Mutex

"006700e5a2ab05704bbb0c589b88924d" adıyla bir **mutex** oluşturulmakta ve **GetLastError** ile "ERROR_ALREADY_EXISTS" (**0xB7**) kontrolü yapılmaktadır. Şayet mutex oluşturulmuşsa **zaten çalıştığı kabul edilerek** program kapanmaktadır.



Görsel 8- Cmd SCHTASKS persistence

Görseldeki script ile programı **her dakika başında** çalıştırmaktadır ve **"/F" (Force)** parametresi ile de uyarıları göz ardı etmektedir.

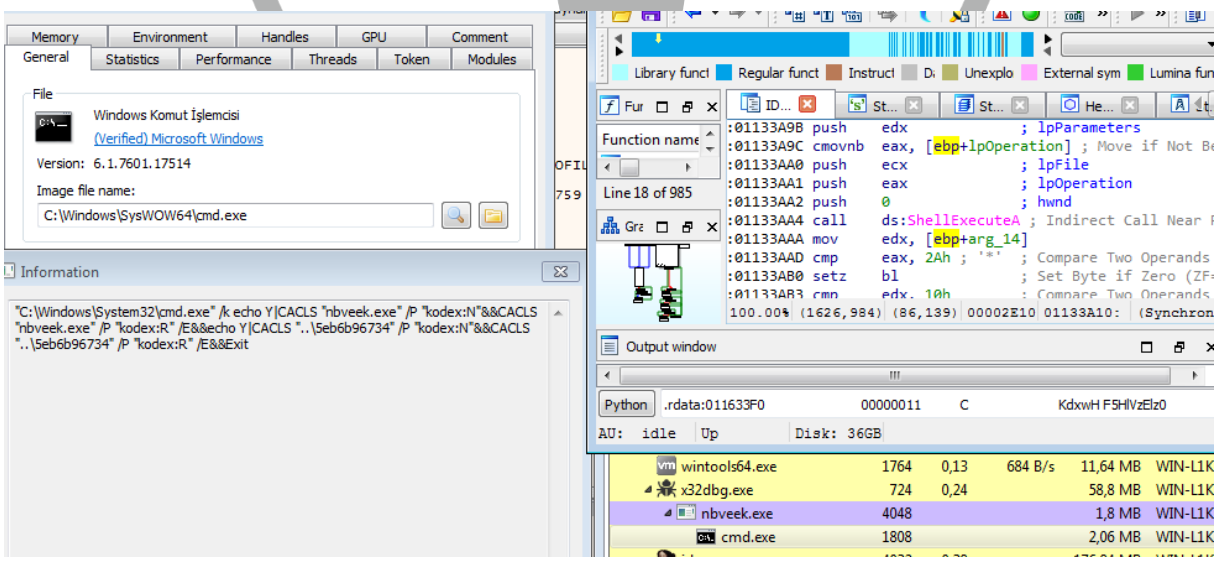
```
"kodex"
":N\""
"&&"
"CACLS \""
"nbveek.exe"
"\" /P \""
"kodex"
":R\" /E"
"&&"
"echo Y|CACLS \""
"..\\5eb6b96734"
"\" /P \""
"kodex"
":N\""
"&&"
"CACLS \""
"..\\5eb6b96734"
"\" /P \""
"kodex"
":R\" /E"
"&&Exit"
```

Görsel 9- Cmd scripti

Parça parça çözülen değerler **stack içinde** tutulup çalıştırılmadan önce birleştirilmektedir.

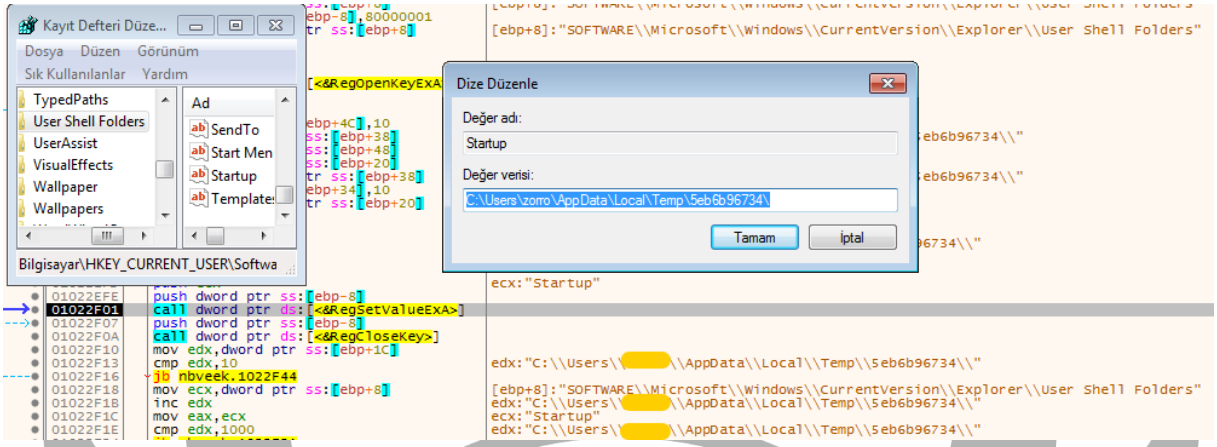
```
"C:\Windows\System32\cmd.exe" /k echo Y|CACLS "nbveek.exe" /P "kodex:N"&&CACLS "nbveek.exe" /P "kodex:R" /E&&echo Y|CACLS "..\5eb6b96734" /P "kodex:N"&&CACLS "..\5eb6b96734" /P "kodex:R" /E&&Exit
```

*Kodex = %USERNAME%



Görsel 10- Cmd scripti_2

“Cmd.exe” içinde çalışan bu scripti **erişim listesini** kontrol etmeyi sağlamaktadır.”echo Y|” komutuyla, “Are you sure?” sorusunun yanıtını “YES” olarak girmektedir. “Kodex:” **kullanıcının girmesi gereken** cevapları girmeyi sağlamaktadır, burada ise ”nbveek.exe” dosyasına önce “**None**” izni sonra “**Read**” izini verilmektedir. Ardından aynı işlemi **zararlının bulunduğu klasör** için de uygulanmaktadır.



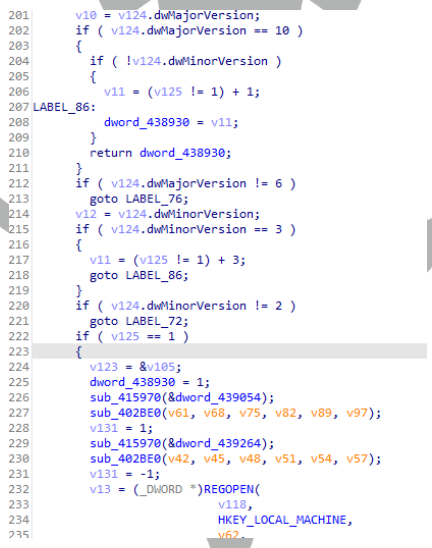
Görsel 11- Kayıt defteri ile kalıcılık

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\
User Shell Folders

Kayıt defterinde yer alan bu yolun **varsayılan değeri** aşağıdaki dosya yoludur.

"%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup"

Fakat zararlı bu işlem ile **varsayılan başlangıç yolunu** kendi dizinine ayarlamaktadır.



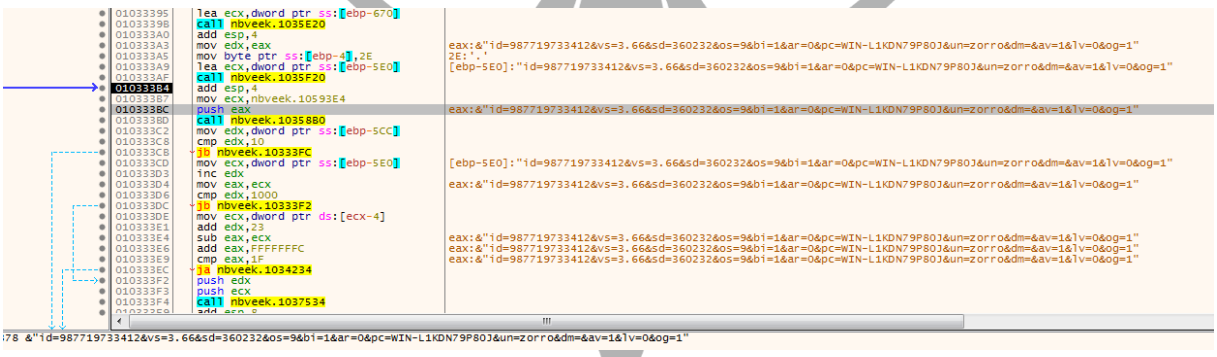
Görsel 12- Versiyon kontrolü

Daha sonra loglanmak için belleğe alınan "**OSVERSIONINFOEXA**" yapısı içerisinden versiyon kontrolü yapılmaktadır. Eğer işletim sistemi **Windows Server 2012** veya **Windows 8** ise registery üzerinden kontrol yapılmaktadır. Ayrıca **GetNativeSystemInfo** API'si ile kontrol edilen veriye göre de **SOFTWARE\Microsoft\Windows NT\CurrentVersion** içindeki **ProductName** değerinde **2016, 2019, 2022** değerlerini aramaktadır. Bu sonuçlara göre "**OS**" loglaması değişmektedir.

Görsel 13- AV kontrolü

ProgramData klasörü altında seçilen güvenlik yazılımlarının varlığı kontrol edilmektedir. ProgramData klasörü altında belirtilen **güvenlik uygulamalarından** herhangi birinin adı ile oluşturulmuş (**boş bile olsa**) bir klasör varsa **"antivirüs var"** olarak algılamaktadır. (&av=1)

AVAST Software	Avira	Kaspersky Lab	ESET
Panda Security	Doctor Web	AVG	360TotalSecurity
Bitdefender	Norton	Sophos	Comod

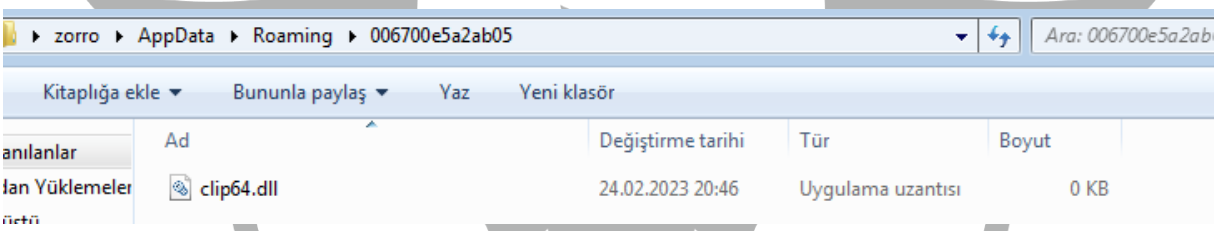


Görsel 14- LSAPolicyLookup

Yukarıdaki görselde sistem hakkında toplanan bilgiler **komuta kontrol sunucusuna** gönderilmek üzere tek metin halinde birleştirilmektedir.

Görsel 15- Zararlı URL

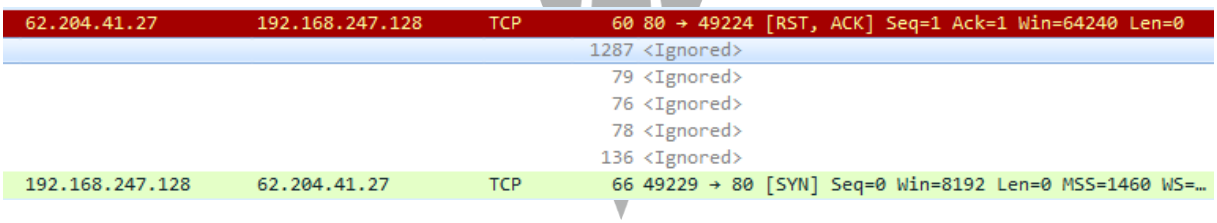
Görseldeki URL'yi açıp **dosya okumayı denemekte** fakat **kapalı** olduğu için bunu gerçekleştirememektedir.



Görsel 16- İndirilecek dosya

Okunduktan sonra yazılması beklenen dosya dizini de bu şekildedir. Öncesinde **Mutex** burada oluşturulmaktadır.

```
C:\Users\%USERNAME%\AppData\Roaming\006700e5a2ab05\
```



Görsel 17- Wireshark görünümü

Oluşturulan **thread** ile program her defasında komuta kontrol sunucusuna **TCP paketi** göndermektedir. Fakat sunucu kapalı olduğu için **RST paket** geldiğinden dolayı herhangi bir işlem devam etmemektedir.

```
end = 0x433E59      # Here is for end of the encoded string's
start = 0x4333F0    # Start of encoded string's address

value = idaapi.get_bytes(start,end-start)
stringValue = value.decode("utf-8")
listedValue = stringValue.split('\x00')

def yaz(final):
        if(final != ""):
                text_file = open("Decrypted.txt", "a")
                text_file.write(final+"\n")
                text_file.close()

def decodeToBase64(listedValue):
    alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 "
    key = "850c61ff7cfc4c28ae073b6ce7172cbd850c61ff7cfc4c28ae073b6ce7172cbd"
    alphabet1 = alphabet[::-1]

    for line in listedValue:
        final = ""
        counter=0
        while(len(line)!= counter and line[counter]!="=" and line[counter]!="+"):
            indexAlp= 0
            while(alphabet1[indexAlp] != key[counter]):
                indexAlp+=1
            indexAlp2 = 0
            #print(line[counter])
            while(alphabet[indexAlp2] != line[counter]):
                indexAlp2+=1

            final +=alphabet[((indexAlp+1) + (indexAlp2+1)) % 63 -1]
            counter +=1
        yaz(final)
decodeToBase64(listedValue)
```

Şifreli metinler yukarıdaki **IdaPython** kodu ile **Base64 değerlerine** döndürülmüş halde metin dosyasına yazılmaktadır.

YARA Kuralı

```
import "pe"

rule Amadey{

meta:

        author="enessakircolak"

        date= "01.03.2023"

strings:

        $a = "Amadey.pdb"

        $b = {83 3D ?? ?? ?? ?? 10 BE ?? ?? ?? ?? 8B CB 0F 43 35 ?? ?? ?? ?? 2B C8 8D 04 0A 33 D2 F7 F3 8B 5D EC 8B CB 83 7B 14 10 72 02 8B 0B 8A 04 32 8B 75 F0 88 04 31 46 89 75 F0 3B 75 F8}

        $mutex = "006700e5a2ab05704bbb0c589b88924d"

        $key = "850c61ff7cfc4c28ae073b6ce7172cbd"

        $enc1 = "KdxwH F5HlVzElz0"

        $enc2 = "RUYIQviGQm0eMYcgcqYv43U21B4l3oSyaXCeLT0D9o0w7HQicrCl430 eVT8s3I BWRyeQNSq9kGV4IMpbGqC43sm3Tzv"

        $enc3 = "OWK2OcK57Z4nN5cwdKKpKX0ofs=="

        $enc4 = "AKeF7S 5VY 2EWImc7qr53g2eSZqFkCp9XyvGMWm IJ9BIWeb0J5AnMjhCueFACp8Qmn7cKyVV5k"

        $api1 = "CreateMutex"

        $api2 = "GetVersionEx"
```

```
        $api3 = "CreateThread"
        $api4 = "ShellExecute"
        $api5 = "HttpOpenRequest"
        $api6 = "InternetOpenUrl"
        $api7 = "CopyFile"
        $api8 = "LoadLibraryEx"
        $api9 = "CreateDirectory"
        $api10 = "RemoveDirectory"
        $api11 = "GetFileAttributes"
        $api12 = "RegCloseKey"

    condition:
        uint16(0) == 0x5a4d
        and filesize <= 1MB
        and pe.imports("WININET.dll")
        and(
        any of ($a,$b,$api*) or all of ($enc*,$mutex,$key)
        )
}
```

MITRE ATTACK TABLE

Reconnaissance	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	C&C	Exfliration
Gather Victim Host Information (T1592)	**Windows Command Shell (T1059)**	**Scheduled Tasks/Job (T1053)**	**Manipulate System Process (T1053)**	**Obfuscated Files or Information (T1027)**	**OS Credential Dumping (T1003)**	**Remote Acces Software (T1219)**	Scheduled Transfer (T1029)
	Scheduled Task (T1053)	**Startup Folder (T1547)**	**Registery Run Keys (T1547)**	**Modify Registry (T1112)**	**Credentials in Registry (T1552.002)**		
	Startup Folder (T1547)						

Çözüm Önerileri

1. Sistem güncel tutulmalıdır.
2. Her işlem çalışma anında denetlenmelidir.
3. Güvenilir anti-virüs yazılımı kullanılmalıdır(kullanılmasa bile o isimde dosya ProgramData altında bulundurulmalıdır).
4. Her türlü doküman teftiş edilerek kullanılmalıdır.

HAZIRLAYAN

Enes Şakir ÇOLAK LinkedIn