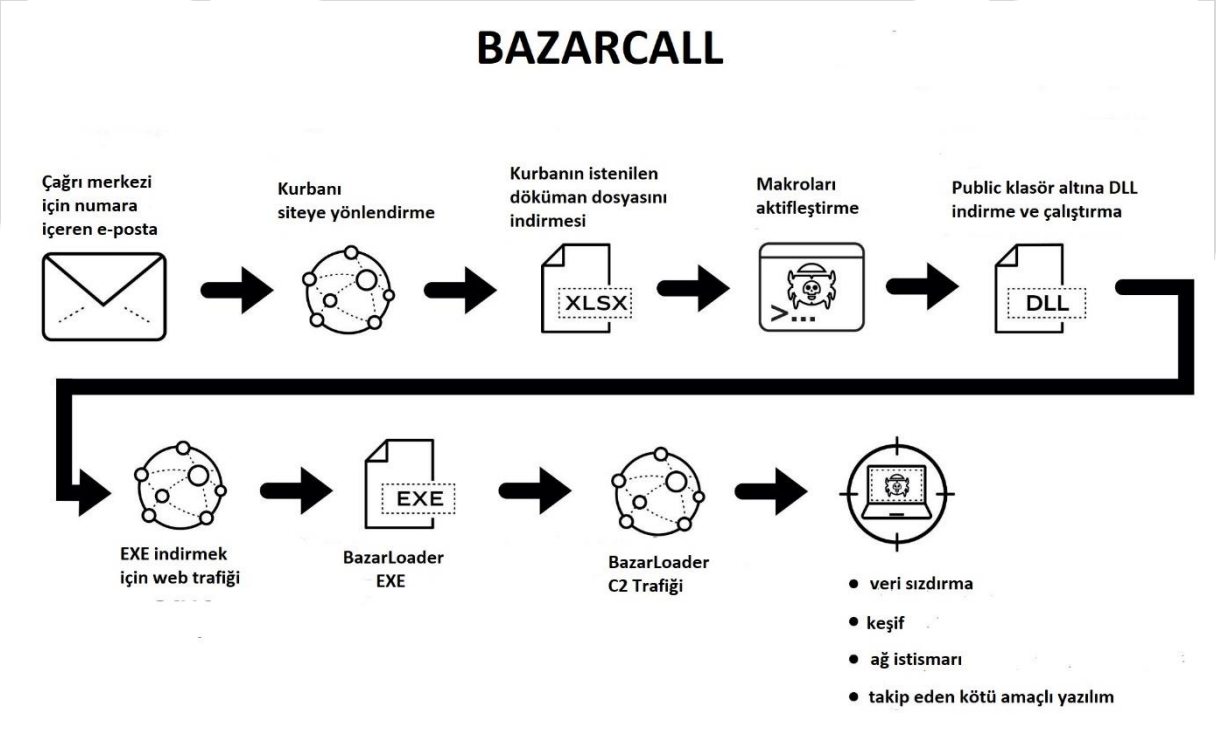
BazarLoader
Teknik Analiz
Raporu

İçindekiler

GİRİŞ

BazarLoader (diğer bilinen ismiyle BazaLoader) bulaştığı Windows host sistemlerine backdoor oluşturan bir zararlı yazılım ailesidir. TA800 tarafından geliştirilmiştir. Backdoor oluşturarak girilen sistemlerdeki zafiyetleri bularak bu zafiyetleri sömürmek üzere zararlı yazılım yükleme ve ağ üzerindeki diğer sistemlere sızmaya çalışmaktadır.

Farklı vektörler ile yayılan BazarLoader genel olarak mail yolu ile kullanıcılara bulaşsa da Şubat 2021'de yapılan araştırmalarda çağrı merkezleri ile yapılan oltalama saldırıları ile kullanıcılara bulaştırılması sağlanmıştır. Bu tarz oltalama çağrılarına da "**BazaCall**" denilmektedir. Kullanıcıları arayıp ücretsiz deneme sürümleri teklif ederek zararlı yazılımı sistemlere enjekte etmeye çalışmaktadırlar.



Dosya İsmi	1f6e8b2f989cc0ce80baa52acc0b3986.dll
MD5	1F6E8B2F989CC0CE80BAA52ACC0B3986
SHA256	bc8407aa092b9b316e72b6082699dd1432521f739eacfb57109bb1d759d89802
SHA1	6fc636cd696a77c590727f512cd4ce02da55d984
İlk Görülme	2021-07-12 06:28:35 UTC

Giriş kısmında bahsedildiği gibi zararlı yazılım çeşitli yollarla kullanıcının cihazına bulaşıp aşağıdaki şekilde çalıştırarak zararlı işlemlerini yapmaya başlamaktadır.

Elimizdeki zararlı yazılım bir DLL olduğu için bir host uygulamaya ihtiyaç duymaktadır, bundan dolayı **rundll32.exe** legal Windows uygulamasına parametre olarak verilip **cmd** tarafından çalıştırılmaktadır. Bu çalıştırmanın ardından birçok teknik ile birlikte yine Windows'un legal bir uygulaması olan **svchost.exe** çalıştırarak içerisine enjekte edilen kodu thread yolu ile çalıştırarak Komuta Kontrol Sunucularına bağlantı kurduğu görülmüştür. Bilindiği üzere **svchost** Windows'ta sistem servislerini çalıştırmak için çalışan legal bir uygulamadır. Zararlı yazılım bu uygulamaya enjekte olarak aynı zamanda kalıcılık da sağlamaktadır. Zararlı yazılımın oluşturduğu işlem ağacını aşağıda görebilirsiniz.

(Sistem Win7 x64)

- cmd.exe (cmdline: cmd.exe /C rundll32.exe 'C:\Users\user\Desktop\1f6e8b2f989cc0ce80baa52acc0b3986.dll',#1 MD5: 4E2ACF4F8A396486AB4268C94A6A245F)
    - rundll32.exe (cmdline: rundll32.exe 'C:\Users\user\Desktop\1f6e8b2f989cc0ce80baa52acc0b3986.dll',#1 MD5: 73C519F050C20580F8A62C849D49215A)
- rundll32.exe (cmdline: C:\Windows\System32\rundll32.exe C:\Users\user\Desktop\1f6e8b2f989cc0ce80baa52acc0b3986.dll,StartW 2791350475 MD5: 73C519F050C20580F8A62C849D49215A)
    - svchost.exe (cmdline: C:\Windows\system32\svchost.exe -k UnistackSvcGroup MD5: 32569E403279B3FD2EDB7EBD036273FA)

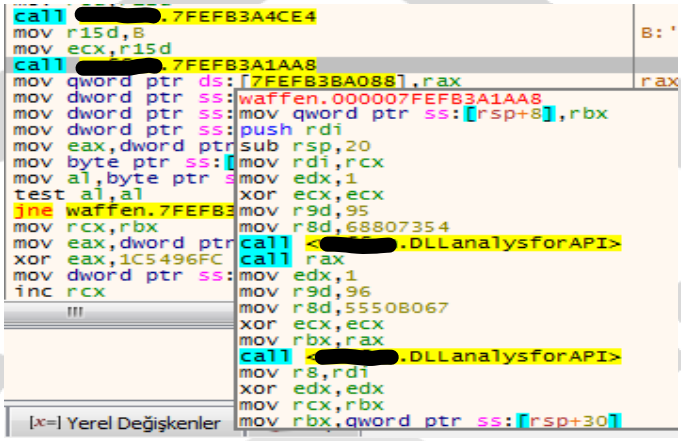
API Hammering

API Hammering, sandbox analizlerini geciktirmek ve yapılan zararlı yazılım teknik analizlerinin kapasitesini düşürmek amacıyla kullanılan bir tekniktir. Değişken olarak belirli API'ların on binlerce kez kullanması yoluyla analizi zorlaştırmaktadır. Sandbox algoritmalarına bakıldığında kayıt tutmak üzerine kurulu olan algoritmalar aşırı yüklenme ile **delay execution** denilen, asıl zararlı kod bloğunun çalıştırılmasını engeller. Örnek vermek gerekirse 2 milyon çağrı yapan bir zararlı yazılım bu çağrılar sonucunda asıl zararlı kod bloğunu çalıştırmak üzere kodlanmış olsun. Belirli bir süre sonra bu kadar fazla çağrının sonucu olarak sandbox'ın tuttuğu kayıtlar tamamen gereksiz verilerle dolacaktır ve asıl zararlı kod çalışmayacaktır.

Bu tekniği kullanılan bir zararlı yazılımdan alınan API çağrılarının sayıları:

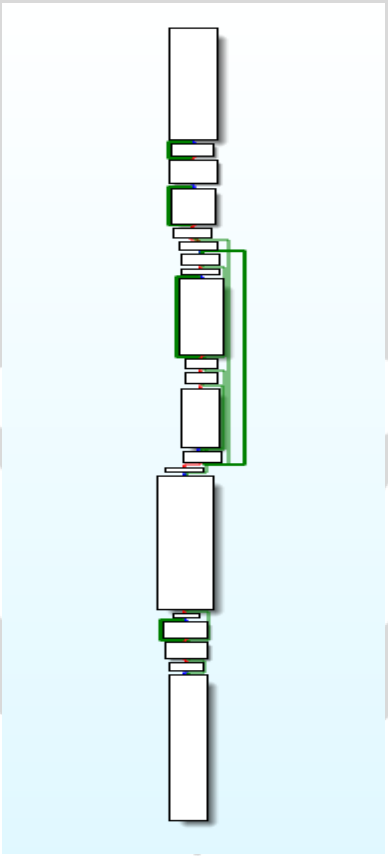
API İsmi	Çağrı Sayısı
KERNEL32.dll.GetLastError	49739
USER32.dll.GetDlgItem	34446
KERNEL32.dll.TlsGetValue	34434
KERNEL32.dll.SetLastError	34434
dbghelp.dll.SymCleanup	30608
USER32.dll.ShowWindow	30608
KERNEL32.dll.GetCurrentProcess	30608
KERNEL32.dll.LeaveCriticalSection	15306
KERNEL32.dll.EnterCriticalSection	15306
KERNEL32.dll.CloseHandle	15305
USER32.dll.FindWindowExA	15304
GDI32.dll.MoveToEx	15304
USER32.dll.GetClassNameA	15304
PSAPI.DLL.GetPerformanceInfo	15304
USER32.dll.SetWindowPlacement	15304
KERNEL32.dll.GlobalMemoryStatusEx	15304
USER32.dll.PostMessageA	15304
PSAPI.DLL.EnumProcesses	15304
KERNEL32.dll.GetVersionExA	15304
dbghelp.dll.SymInitialize	15304…

Bu şekilde manuel şekilde yapılan analizlerde asıl kullanılan zararlı API'ların kullanımlarını ve parametrelerinin analiz edilmesini zorlaştırılmaktadır.

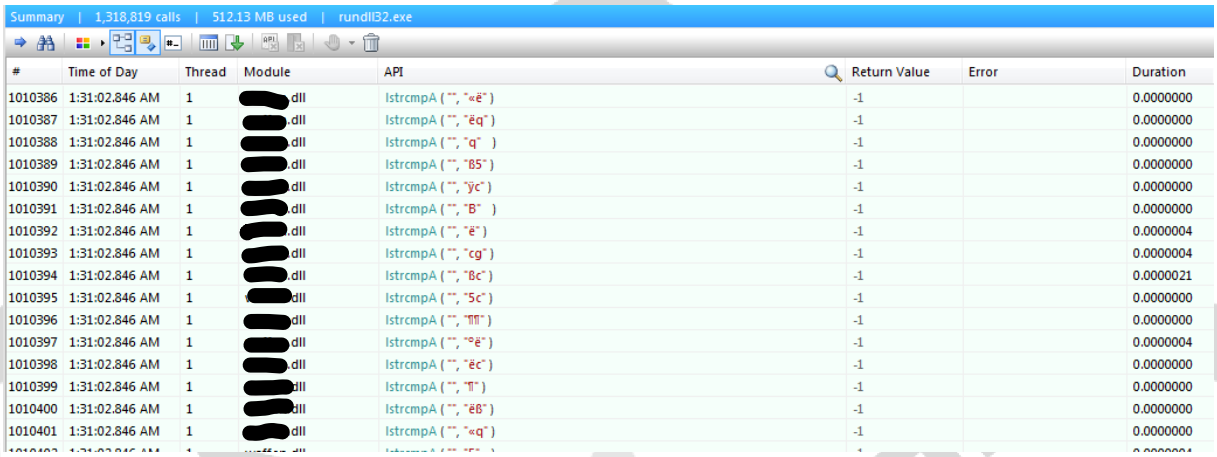


Aynı zamanda zararlı yazılımda dinamik olarak DLL yorumlama ve "parse" işlemi kullanılarak asıl zararlı API'ların hangi kod bloğunda ve ne zaman kullanılacağını analistlerden gizlemiş olmaktadır.

Aşağıda görülen IDA görüntüsündeki gibi bir çok sonsuza yakın döngülerle birlikte yüzbinlerce API çağrısı yapılarak **API Hammering** uygulanmaktadır.



Zararlı yazılımın kısa sürede yaptığı CALL sayısı ve kullanılan hafıza boyutu bu fotoğrafta gözükmektedir:

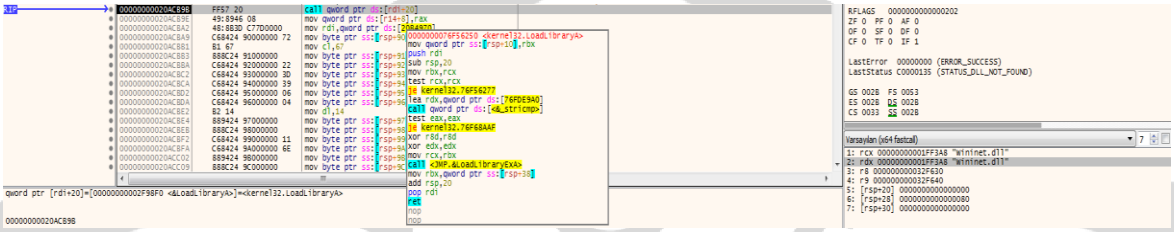


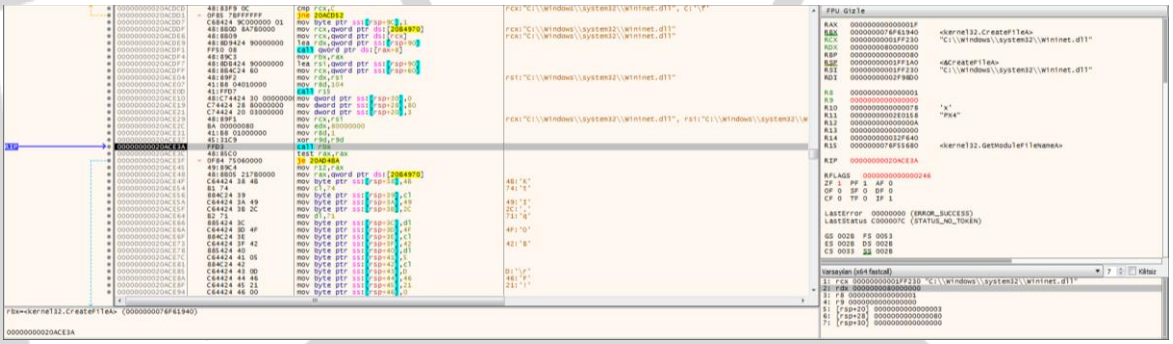
Aşağıdaki tabloda ise zararlı yazılımımız tarafından kendi hafıza bloğuna yüklenen DLL'ler görülmektedir.

Dosya Yolu	API
C:\Windows\System32\kernel32.dll	ReadFile
C:\Windows\System32\wininet.dll	ReadFile
C:\Windows\System32\advapi32.dll	ReadFile
C:\Windows\System32\ole32.dll	ReadFile
C:\Windows\System32\ntdll.dll	ReadFile
C:\Windows\System32\shell32.dll	ReadFile
C:\Windows\System32\bcrypt.dll	ReadFile
C:\Windows\System32\crypt32.dll	ReadFile
C:\Windows\System32\dnsapi.dll	ReadFile
C:\Windows\System32\netapi32.dll	ReadFile
C:\Windows\System32\shlwapi.dll	ReadFile
C:\Windows\System32\user32.dll	ReadFile
C:\Windows\System32\ktmw32.dll	ReadFile

Peki bu DLL'leri nasıl hafızasına yüklemektedir?

Dinamik DLL analizinin ardından **LoadLibrary-CreateFile-ReadFile** sırası ile DLL'leri hafızasına yükleyen zararlı yazılım yukarıda belirtilen DLL'leri hafızasına yükledikten sonra bu DLL'leri parse ederek gerekli API'ların başlangıç adreslerini hafızasında tutmaktadır.





Bu şekilde kullanılacak olan API'ları kendi hafızasında **hard-coded** şekilde tutmayıp statik analizi zorlaştırmaktadır. Aynı zamanda API Hammering tekniği ile birleştirip sonrasında ne zaman kullanacağını da gizlemektedir.

Svchost içerisine enjekte edilip çalıştırılan thread içeriği:

ntdll.dll!ZwWaitForSingleObject
KernelBase.dll!WaitForSingleObjectEx
wininet.dll!InternetSetStatusCallbackW
wininet.dll!GetUrlCacheHeaderData
wininet.dll!InternetSetStatusCallbackA
wininet.dll!HttpOpenDependencyHandle
wininet.dll!InternetCanonicalizeUrlW
wininet.dll!AppCacheGetManifestUrl
wininet.dll!InternetConfirmZoneCrossingW
wininet.dll!HttpSendRequestA
kernel32.dll!BaseThreadInitThunk
ntdll.dll!RtlUserThreadStart
wininet.dll!InternetSetStatusCallbackA

BazarLoader'ın bir backdoor oluşturan zararlı yazılım ailesi olduğu bilinmektedir. Peki bu backdoor nasıl sağlanır?

Svchost'a enjekte edilen kod ise hafızasında sakladığı belirli komuta kontrol sunucularına periyodik olarak **HTTP** isteği atarak komut beklemektedir. Aşağıdaki procmon görüntüsünde görüldüğü üzere thread oluşturarak periyodik olarak istek atmaktadır.

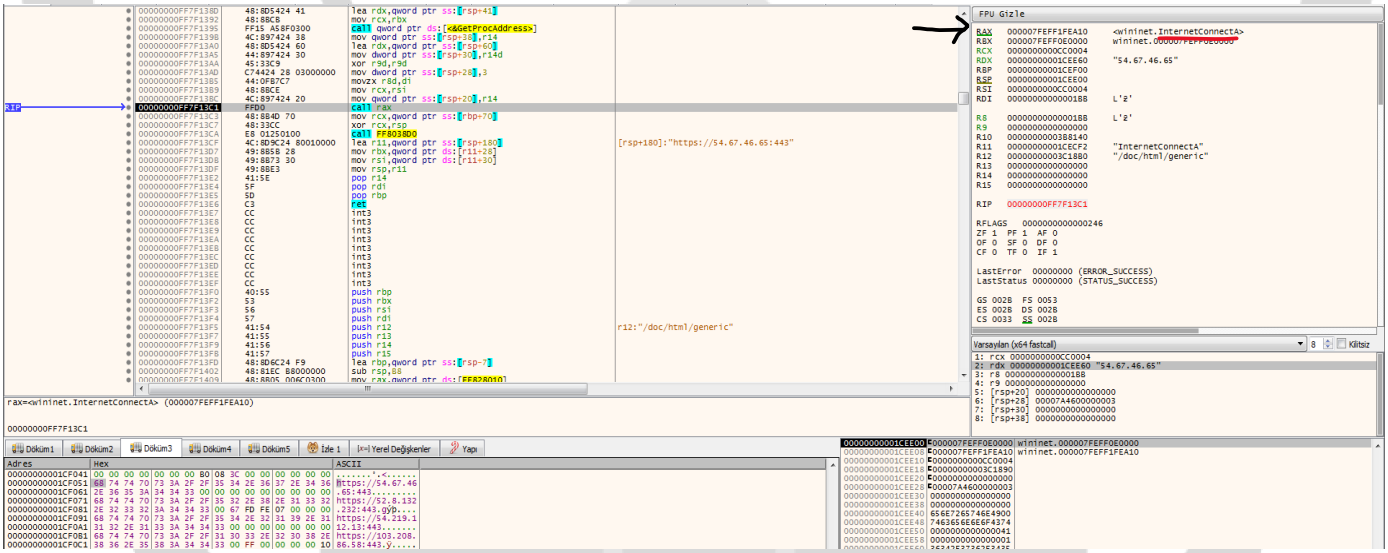
svchost.exe	2216	TCP Reconnect	[redacted]localdomain:62135 -> ec2-54-219-112-13.us-west-1.compute.amazonaws.com:https
svchost.exe	2216	TCP Reconnect	[redacted]localdomain:62135 -> ec2-54-219-112-13.us-west-1.compute.amazonaws.com:https
svchost.exe	2216	TCP Reconnect	[redacted]localdomain:62136 -> ec2-54-219-112-13.us-west-1.compute.amazonaws.com:https
svchost.exe	2216	TCP Reconnect	[redacted]localdomain:62136 -> ec2-54-219-112-13.us-west-1.compute.amazonaws.com:https
svchost.exe	2216	Thread Exit	
svchost.exe	2216	Thread Exit	
svchost.exe	2216	RegQueryKey	HKCU
svchost.exe	2216	RegCreateKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Connections
svchost.exe	2216	RegQueryValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Connections\DefaultConnectionSettings
svchost.exe	2216	RegQueryValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Connections\DefaultConnectionSettings
svchost.exe	2216	RegQueryValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Connections\DefaultConnectionSettings
svchost.exe	2216	RegCloseKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Connections
svchost.exe	2216	RegQueryKey	HKLM
svchost.exe	2216	RegOpenKey	HKLM\SYSTEM\CurrentControlSet\Services\NetBT\Linkage
svchost.exe	2216	RegOpenKey	HKLM\System\CurrentControlSet\Services\NetBT\Linkage
svchost.exe	2216	RegQueryValue	HKLM\System\CurrentControlSet\services\NetBT\Linkage\Export
svchost.exe	2216	RegQueryValue	HKLM\System\CurrentControlSet\services\NetBT\Linkage\Export
svchost.exe	2216	RegQueryValue	HKLM\System\CurrentControlSet\services\NetBT\Linkage\Export
svchost.exe	2216	RegCloseKey	HKLM\System\CurrentControlSet\services\NetBT\Linkage
svchost.exe	2216	TCP Reconnect	[redacted]localdomain:62137 -> ec2-54-219-112-13.us-west-1.compute.amazonaws.com:https
svchost.exe	2216	Thread Create	
svchost.exe	2216	TCP Reconnect	[redacted]localdomain:62137 -> ec2-54-219-112-13.us-west-1.compute.amazonaws.com:https
svchost.exe	2216	TCP Reconnect	[redacted]localdomain:62138 -> ec2-54-219-112-13.us-west-1.compute.amazonaws.com:https

Network Analizi

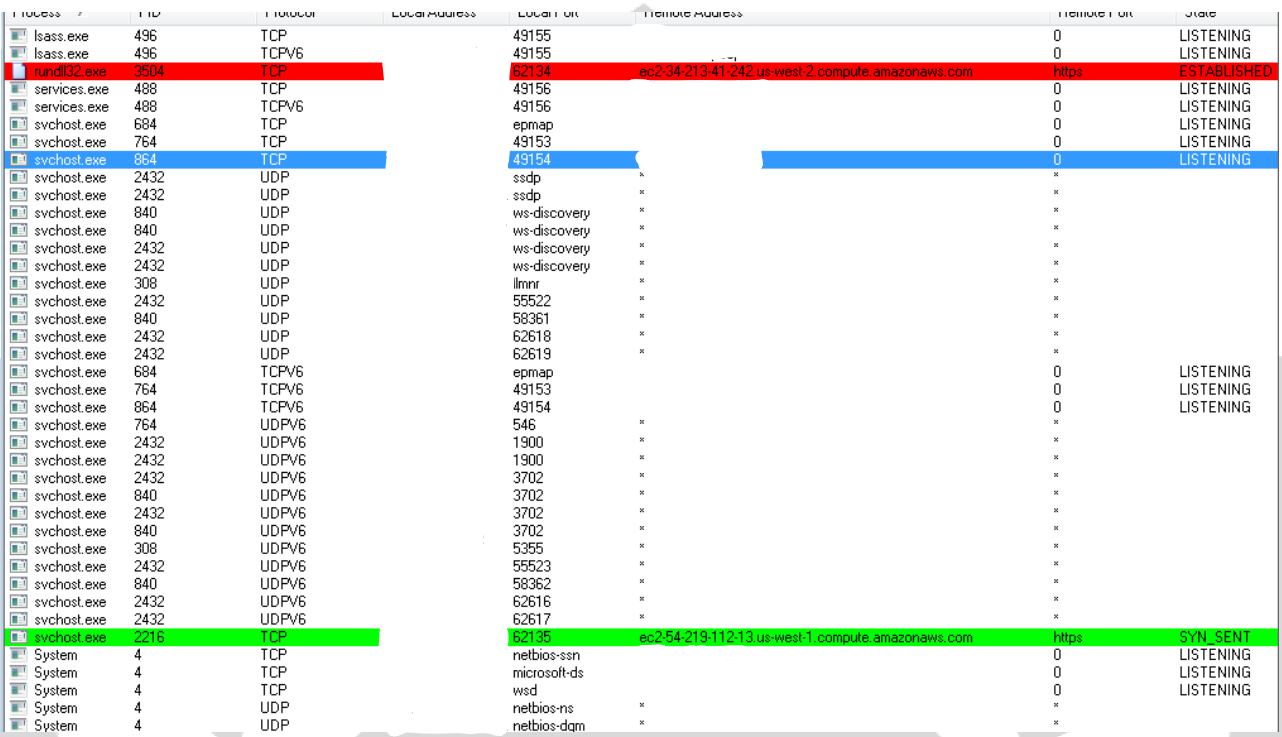
BazarLoader zararlısının bir backdoor sağlayıcı olduğu bilinmektedir. **Bu backdoor nasıl sağlanır ve nedir?**

Sızılan cihazda gelen komut neticesinde kod çalıştırma, sistemden bilgi çalma, sistem hareketlerini izlemek, keylogger gibi bir çok amaç ile kullanılmaktadır. Aslına bakıldığında asıl cihazın bütün trafiği komuta kontrol sunucusu üstünden geçmektedir çünkü bütün hareketleri bu sunucu tarafından izlenebilir, müdahale edilebilir. İstenildiğinde daha büyük zararlar vermek için farklı zararlı yazılımlar indirilip-çalıştırılabilir. Bunun için bağlantı sürekli aktif tutularak komut beklenir.

Backdoor tarafından komut çalıştırmak ve bağlantıyı canlı tutmak için belirli komuta kontrol sunucuları ile **InternetConnectA** API'ı ile bağlantı kurduğu görülmektedir.



Aynı zamanda svchost içerisinde oluşturulan thread ile "xxx.xxx.xxx.xxx.us-west[.]-1[.]compute[.]amazonaws[.]com" domainlerine periyodik olarak bağlantı sağladığı görülmüştür.



Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port	State
lsass.exe	496	TCP		49155		0	LISTENING
lsass.exe	496	TCPV6		49155		0	LISTENING
rundll32.exe	3504	TCP		62134	ec2-34-213-41-242.us-west-2.compute.amazonaws.com	https	ESTABLISHED
services.exe	488	TCP		49156		0	LISTENING
services.exe	488	TCPV6		49156		0	LISTENING
svchost.exe	684	TCP		epmap		0	LISTENING
svchost.exe	764	TCP		49153		0	LISTENING
svchost.exe	864	TCP		49154		0	LISTENING
svchost.exe	2432	UDP		ssdp	*	*	
svchost.exe	2432	UDP		ssdp	*	*	
svchost.exe	840	UDP		ws-discovery	*	*	
svchost.exe	840	UDP		ws-discovery	*	*	
svchost.exe	2432	UDP		ws-discovery	*	*	
svchost.exe	2432	UDP		ws-discovery	*	*	
svchost.exe	308	UDP		llmnr	*	*	
svchost.exe	2432	UDP		55522	*	*	
svchost.exe	840	UDP		58361	*	*	
svchost.exe	2432	UDP		62618	*	*	
svchost.exe	2432	UDP		62619	*	*	
svchost.exe	684	TCPV6		epmap		0	LISTENING
svchost.exe	764	TCPV6		49153		0	LISTENING
svchost.exe	864	TCPV6		49154		0	LISTENING
svchost.exe	764	UDPV6		546	*	*	
svchost.exe	2432	UDPV6		1900	*	*	
svchost.exe	2432	UDPV6		1900	*	*	
svchost.exe	2432	UDPV6		3702	*	*	
svchost.exe	840	UDPV6		3702	*	*	
svchost.exe	2432	UDPV6		3702	*	*	
svchost.exe	840	UDPV6		3702	*	*	
svchost.exe	308	UDPV6		5355	*	*	
svchost.exe	2432	UDPV6		55523	*	*	
svchost.exe	840	UDPV6		58362	*	*	
svchost.exe	2432	UDPV6		62616	*	*	
svchost.exe	2432	UDPV6		62617	*	*	
svchost.exe	2216	TCP		62135	ec2-54-219-112-13.us-west-1.compute.amazonaws.com	https	SYN_SENT
System	4	TCP		netbios-ssn		0	LISTENING
System	4	TCP		microsoft-ds		0	LISTENING
System	4	TCP		wsd		0	LISTENING
System	4	UDP		netbios-ns	*	*	
System	4	UDP		netbios-dgm	*	*	

MITRE ATT&CK Tablosu

Execution	Persistence	Privilege Escalation	Defense Evasion	Discovery	Command and Control	Collection
Shared Modules	Application Shimming	Process Injection	Masquerading	System Time Discovery	Encrypted Channel	Archive Collected Data
		Application Shimming	Virtualization/Sandbox Evasion	Security Software Discovery	Application Layer Protocol	
			Process Injection	Virtualization / Sandbox Evasion		
			Obfuscated Files or Information	Process Discovery		
			Rundll32	File and Directory Discovery		
			Software Packing	System Information Discovery		

Çözüm Önerileri

Backdoor türündeki BazarLoader 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 BazarLoader zararlısının sisteme bulaşmasını engelleyebilmektedir.

YARA Rule

```
import "hash"
import "pe"

rule FirstFile{
        meta:
                description="1f6e8b2f989cc0ce80baa52acc0b3986.dll"
        strings:
                $str1="LoadLibraryW"
                $str2="us-west-1.compute.amazonaws.com"
                $str3="54.67.46.65"
                $str4="52.8.132.232"
                $str5="54.219.112.13"
                $str6="103.208.86.56"
                $str7="InternetConnectA"
                $str8="InternetOpenA"
                $str9="HttpOpenRequestA"
                $str10="CreateMutex"
                $str11=”VirtualAllocA”

        condition:
                hash.md5(0,filesize) =="1F6E8B2F989CC0CE80BAA52ACC0B3986" or all of them
}
```

Fatih YILMAZ

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