

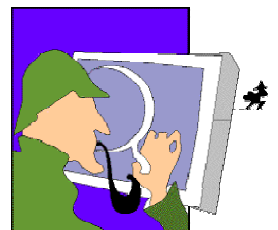
Anatomy of Denial of Service Attack and Defense in a Lab Environment

Dongqing Yuan

**Department of Information Technology Management
University of Wisconsin-Stout
Yuanh@uwstout.edu**

Dr. Jiling Zhong

**Department of Computer Science
Troy University
Jzhong@troy.edu**



Overview

- **Introduction of DoS attack**
- **Attack 1 – Target is the host**
- **Attack 2 – Target is the network**
- **Summary**



What is Denial of Service Attack?

- “Attack in which the primary goal is to deny the victim(s) access to a particular resource.” (CERT/CC)
- The definition covers many types of DoS
- Three basic types of DoS– Smurf, Fraggle, SYN Flood Attack.
- This study only focuses on SYN Flood Attack
 - SYN Flooding DoS attacks are the most popular DoS attacks

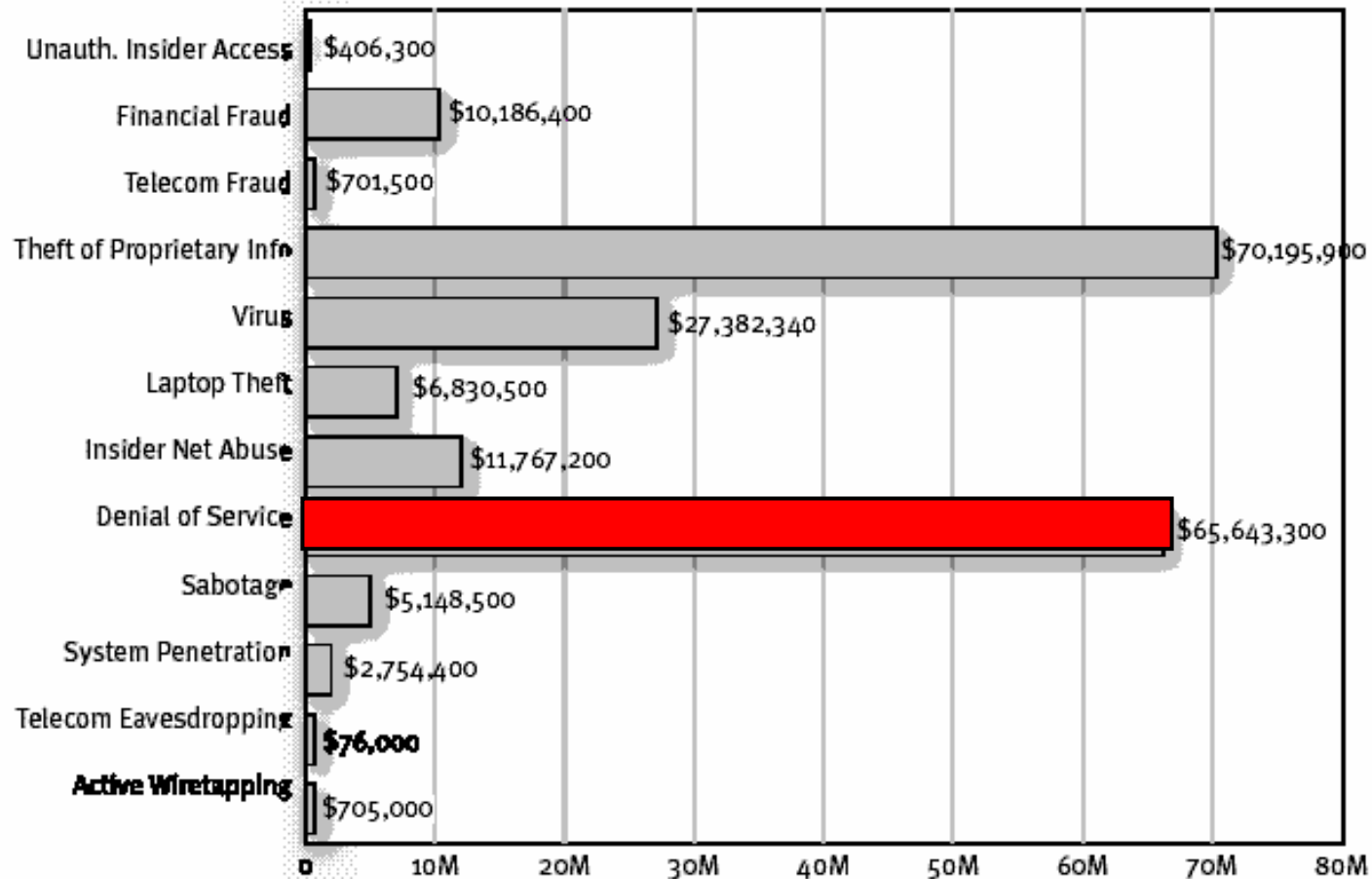


Why it is important to exam this attack?

- Easier to launch the attack
- Many incentives for attackers: unauthorized use, ego, hate, disrupt competitor...
- The design of the Internet
- There is no universal solution to the attack



Dollar Amount of Losses by Type

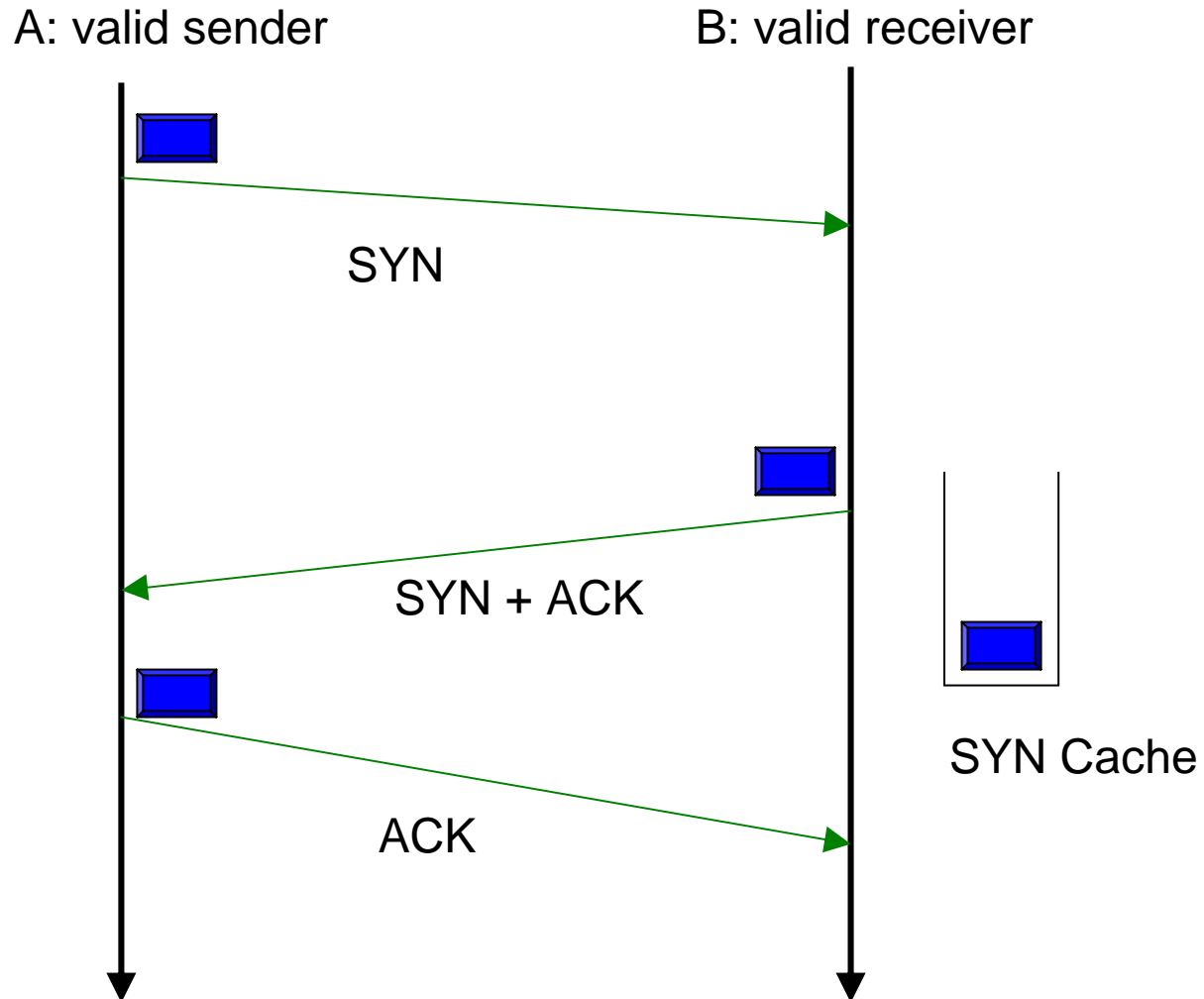


CSI/FBI 2003 Computer Crime and Security Survey
Source: Computer Security Institute

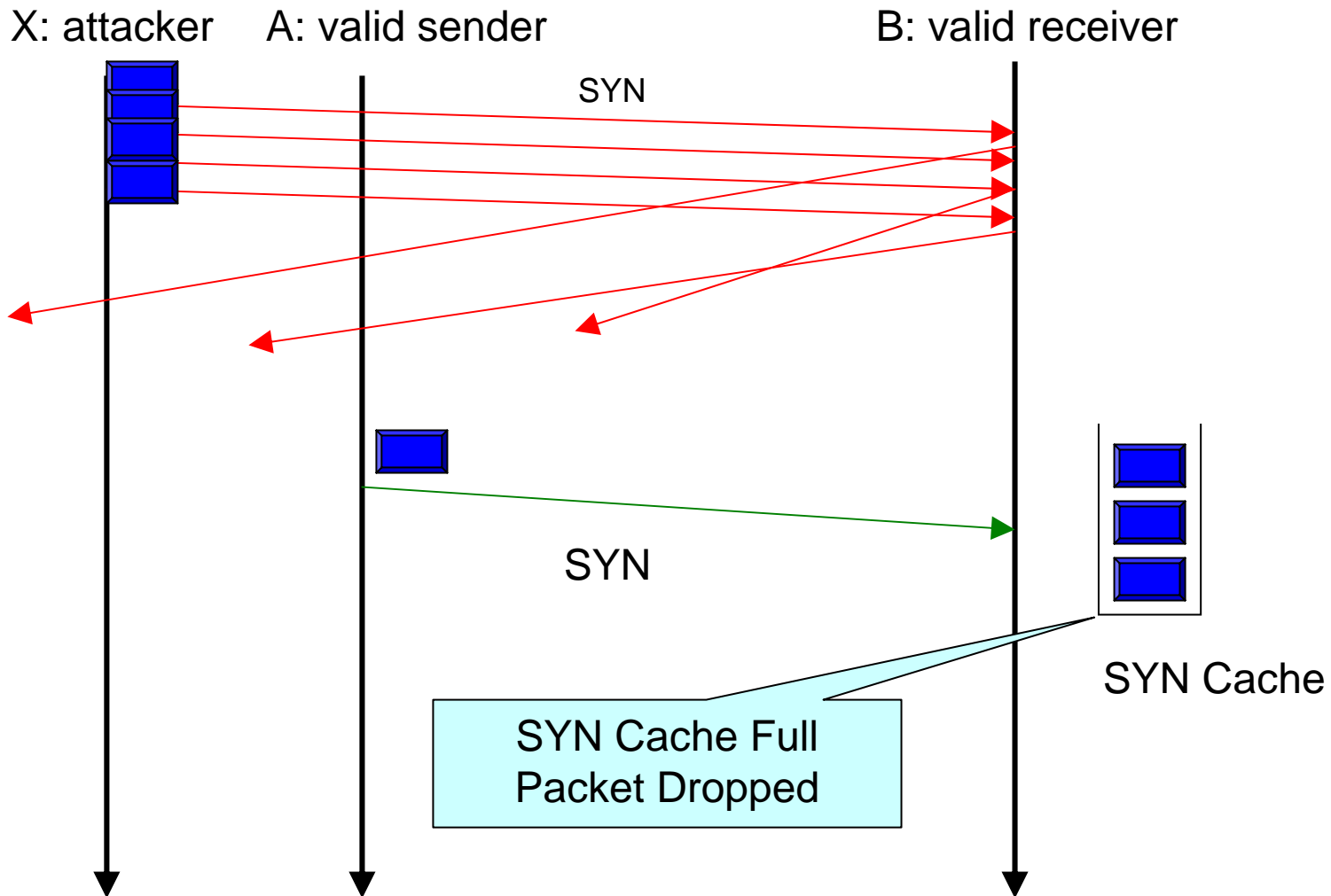
2003: 251 Respondents/ 47%



TCP is susceptible to DoS attacks



TCP is Susceptible to DoS Attacks



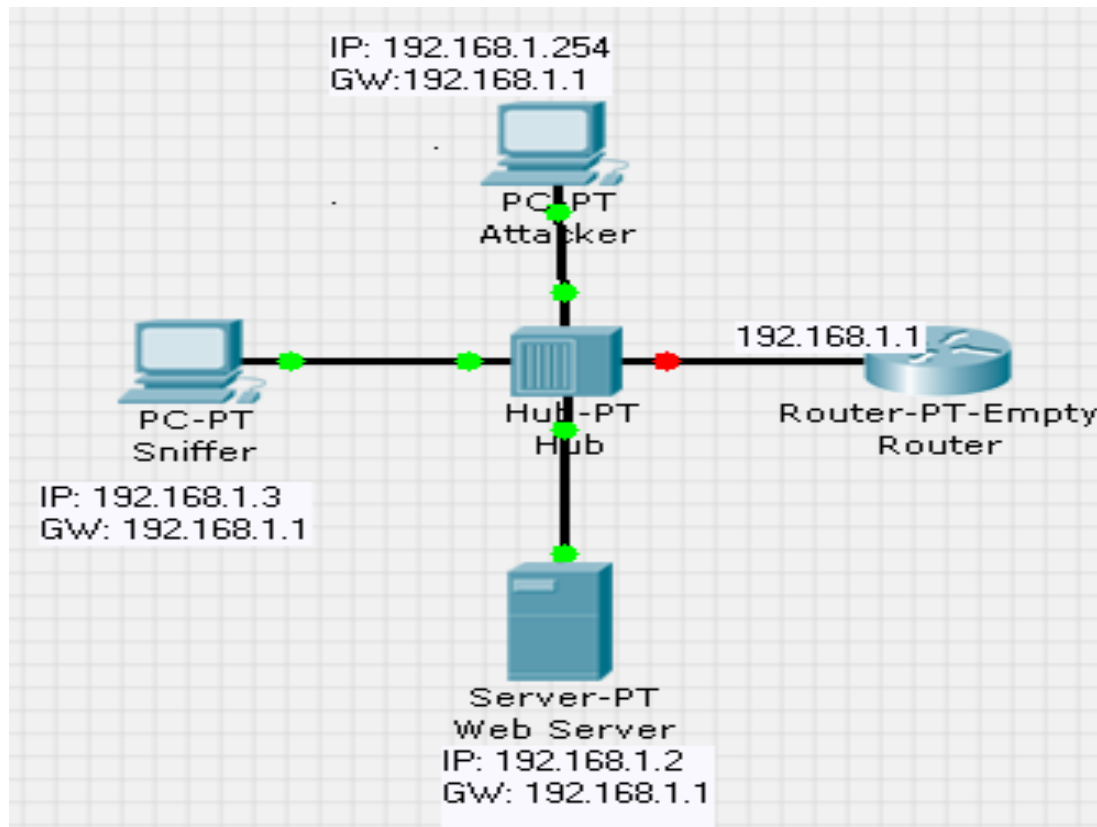
DoS Tools

- There are lots of DoS tools.
- In our simulation, we use Datapool. Datapool is a powerful DoS tool that includes 106 DoS attacks.
- <http://packetstormsecurity.org/DoS/datapool2.0.tar.gz>



Attack 1– Target is the End Node

- Topology: A hub connect web server, sniffer and attacker.



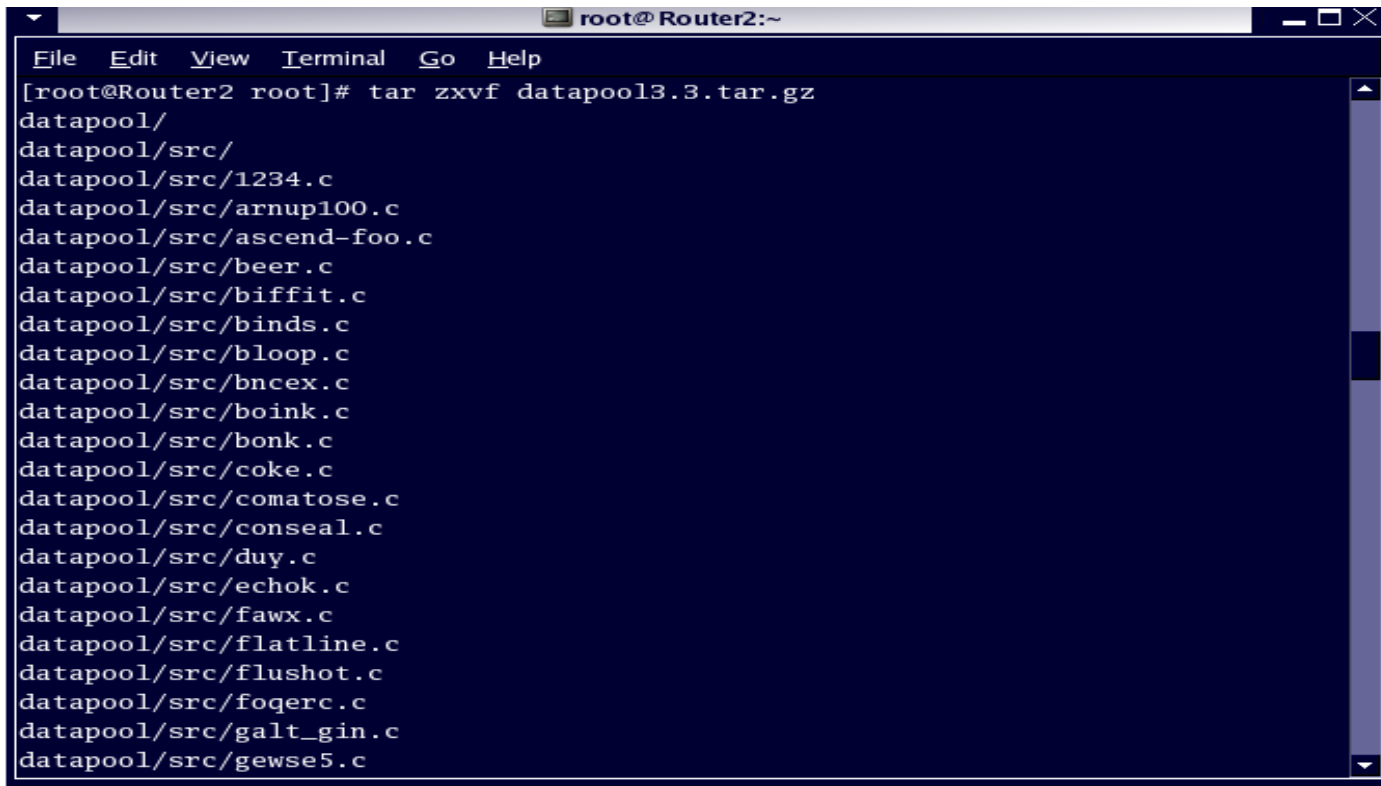
Lab Requirement for Attack 1

- A Linux machine is set up as an HTTP Server, the IP address of which is 192.168.1.2.
- A Windows XP computer is set up as a Sniffer running Ethereal, which is a program that turns a computer's NIC card into promiscuous mode to gather all packets on the wire. The Sniffer's IP address is 192.168.1.3.
- Another Linux machine is set up as an Attacker, running Datapool. The attacker's IP address is 192.168.1.254.



Extract the DoS tool

Download the Datapool and extract the file.



```
root@Router2:~  
File Edit View Terminal Go Help  
[root@Router2 root]# tar zxvf datapool3.3.tar.gz  
datapool/  
datapool/src/  
datapool/src/1234.c  
datapool/src/arnup100.c  
datapool/src/ascend-foo.c  
datapool/src/beer.c  
datapool/src/biffit.c  
datapool/src/binds.c  
datapool/src/bloop.c  
datapool/src/bncex.c  
datapool/src/boink.c  
datapool/src/bonk.c  
datapool/src/coke.c  
datapool/src/comatose.c  
datapool/src/conseal.c  
datapool/src/duy.c  
datapool/src/echok.c  
datapool/src/fawx.c  
datapool/src/flatline.c  
datapool/src/flushot.c  
datapool/src/foqerc.c  
datapool/src/galt_gin.c  
datapool/src/gewse5.c
```



Attacking...

```
root@Router2:~  
File Edit View Terminal Go Help  
-----  
|Option          |Setting  
-----  
Destination Host: 192.168.1.2  
Source IP:        13.31.16.15  
Port Range:      80-80  
Logging:         OFF  
Scan Only:       OFF  
Line Speed:      Modem  
Continuous Attack: ON  
"Don't stop till they drop": OFF  
Wait for online host: OFF  
# of simultaneous attacks: 1  
Attacks in initial list: synful  
  
Starting portstan...  
192.168.1.2 resolved to 192.168.1.2  
Linux host detected...  
1 TCP port(s) were found open:  
80/http  
Launching 1 attack(s) at 192.168.1.2 on port: 80  
Running SYN flooder (synful)...  
Launching 1 attack(s) at 192.168.1.2 on port: 80  
Running SYN flooder (synful)...  
Launching 1 attack(s) at 192.168.1.2 on port: 80  
Running SYN flooder (synful)...  
Launching 1 attack(s) at 192.168.1.2 on port: 80  
Running SYN flooder (synful)...  
Launching 1 attack(s) at 192.168.1.2 on port: 80  
Running SYN flooder (synful)...
```



Sniffer Shows a Normal Three-way Handshake

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.3	192.168.1.2	TCP	1096 > http [SYN] Seq=0 Len=0 MSS=1460
2	0.000379	192.168.1.2	192.168.1.3	TCP	http > 1096 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1460
3	0.000418	192.168.1.3	192.168.1.2	TCP	1096 > http [ACK] Seq=1 Ack=1 win=17520 Len=0
4	0.000642	192.168.1.3	192.168.1.2	HTTP	GET / HTTP/1.1
5	0.001249	192.168.1.2	192.168.1.3	TCP	http > 1096 [ACK] Seq=1 Ack=282 win=6432 Len=0
6	0.001941	192.168.1.2	192.168.1.3	HTTP	HTTP/1.1 304 Not Modified
7	0.001962	192.168.1.2	192.168.1.3	TCP	http > 1096 [FIN, ACK] Seq=147 Ack=282 win=6432 Len=0
8	0.001994	192.168.1.3	192.168.1.2	TCP	1096 > http [ACK] Seq=282 Ack=148 win=17374 Len=0
9	0.002229	192.168.1.3	192.168.1.2	TCP	1096 > http [FIN, ACK] Seq=282 Ack=148 win=17374 Len=0
10	0.002486	192.168.1.2	192.168.1.3	TCP	http > 1096 [ACK] Seq=148 Ack=283 win=6432 Len=0
11	25.900774	3com_c0:6e:9f	Broadcast	ARP	who has 192.168.1.2? Tell 192.168.1.254
12	25.900948	de11_79:22:64	3com_c0:6e:9f	ARP	192.168.1.2 is at 00:14:22:79:22:64
13	25.901055	192.168.1.254	192.168.1.2	TCP	46952 > http [SYN] Seq=0 Len=0
14	25.901203	192.168.1.2	192.168.1.254	TCP	http > 46952 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1460
15	25.901329	192.168.1.254	192.168.1.2	TCP	46952 > http [RST] Seq=1 Len=0
16	25.901886	192.168.1.254	192.168.1.2	TCP	46959 > http [SYN, ECN] Seq=0 Len=0 WS=10 MSS=265 TSV=1061109567
17	25.901964	192.168.1.254	192.168.1.2	TCP	46960 > http [] Seq=0 Len=0 WS=10 MSS=265 TSV=1061109567 TSER=0
18	25.902043	192.168.1.254	192.168.1.2	TCP	46961 > http [FIN, SYN, PSH, URG] Seq=0 Urg=0 Len=0 WS=10 MSS=265
19	25.902145	192.168.1.254	192.168.1.2	TCP	46962 > http [ACK] Seq=0 Ack=0 win=3072 Len=0 WS=10 MSS=265 TSV=1061109567
20	25.902285	192.168.1.254	192.168.1.2	TCP	46963 > 34446 [SYN] Seq=0 Len=0 WS=10 MSS=265 TSV=1061109567 TSER=0
21	25.902385	192.168.1.254	192.168.1.2	TCP	46964 > 34446 [ACK] Seq=0 Ack=0 win=3072 Len=0 WS=10 MSS=265 TSV=1061109567
22	25.902465	192.168.1.254	192.168.1.2	TCP	46965 > 34446 [FIN, PSH, URG] Seq=0 Urg=0 Len=0 WS=10 MSS=265 TSV=1061109567
23	25.902840	192.168.1.254	192.168.1.2	UDP	Source port: 46952 Destination port: 34446
24	25.902846	192.168.1.2	192.168.1.254	TCP	http > 46959 [SYN, ACK] Seq=0 Ack=1 win=5792 Len=0 MSS=1460 TSV=1061109567
25	25.902918	192.168.1.2	192.168.1.254	TCP	http > 46961 [SYN, ACK] Seq=0 Ack=1 win=5792 Len=0 MSS=1460 TSV=1061109567
26	25.903009	192.168.1.2	192.168.1.254	TCP	http > 46962 [RST] Seq=0 Len=0
27	25.903100	192.168.1.2	192.168.1.254	TCP	34446 > 46963 [RST, ACK] Seq=0 Ack=1 win=0 Len=0
28	25.903168	192.168.1.2	192.168.1.254	TCP	34446 > 46964 [RST] Seq=0 Len=0

Frame 74 (60 bytes on wire, 60 bytes captured)

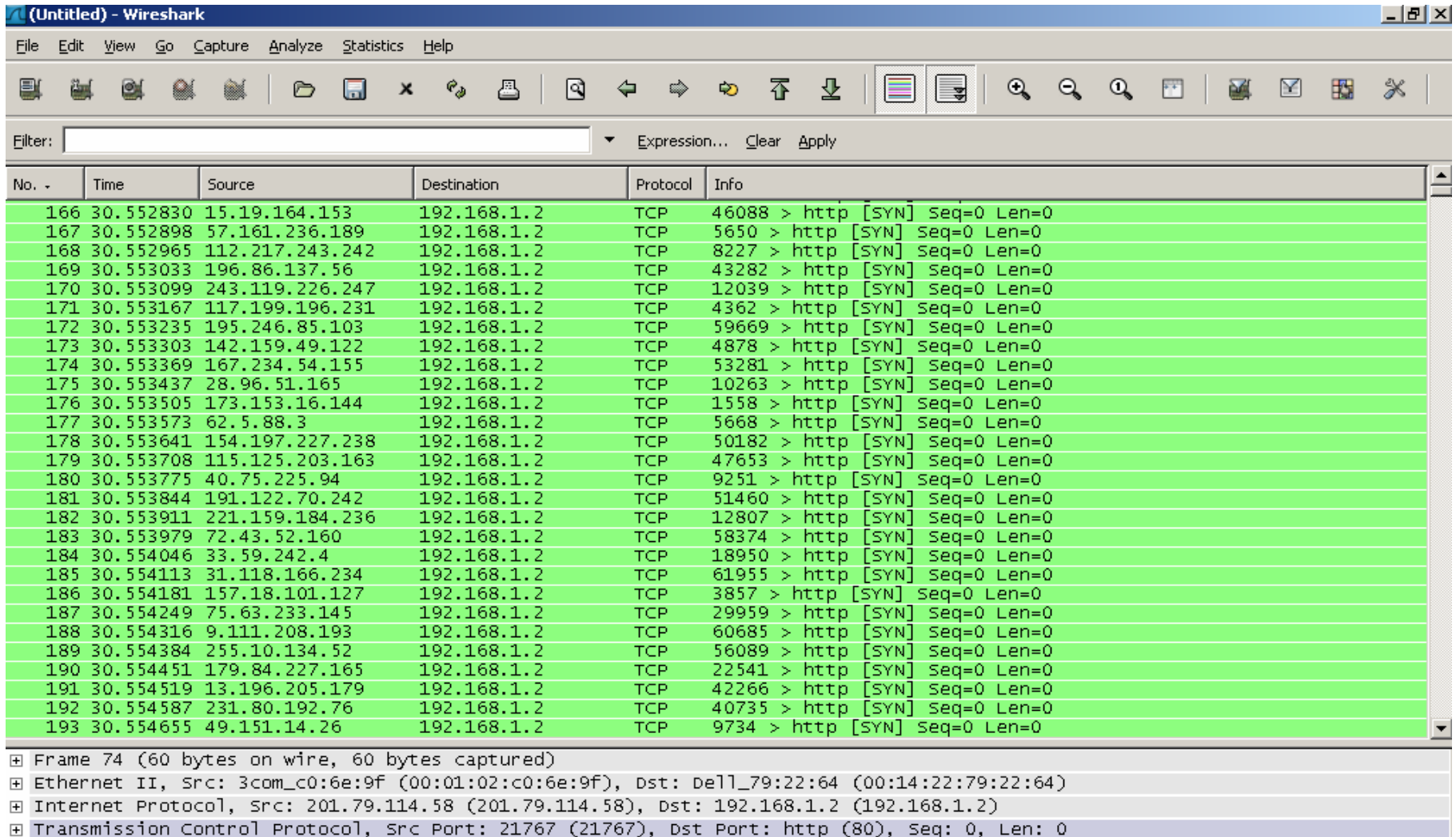
Ethernet II, Src: 3com_c0:6e:9f (00:01:02:c0:6e:9f), Dst: de11_79:22:64 (00:14:22:79:22:64)

Internet Protocol, Src: 201.79.114.58 (201.79.114.58), Dst: 192.168.1.2 (192.168.1.2)

Transmission Control Protocol, Src Port: 21767 (21767), Dst Port: http (80), Seq: 0, Len: 0



Sniffer Shows SYN Flooding Packets



The image shows a Wireshark network traffic capture window titled "(Untitled) - Wireshark". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Help), a toolbar with various icons, and a filter field. The main display area shows a list of captured packets, all of which are SYN packets from various source IP addresses to the destination 192.168.1.2. The packets are highlighted in green. Below the packet list, the details pane shows the structure of the selected packet (No. 74): Ethernet II, Internet Protocol, and Transmission Control Protocol (TCP) with Seq=0 and Len=0.

No.	Time	Source	Destination	Protocol	Info
166	30.552830	15.19.164.153	192.168.1.2	TCP	46088 > http [SYN] Seq=0 Len=0
167	30.552898	57.161.236.189	192.168.1.2	TCP	5650 > http [SYN] Seq=0 Len=0
168	30.552965	112.217.243.242	192.168.1.2	TCP	8227 > http [SYN] Seq=0 Len=0
169	30.553033	196.86.137.56	192.168.1.2	TCP	43282 > http [SYN] Seq=0 Len=0
170	30.553099	243.119.226.247	192.168.1.2	TCP	12039 > http [SYN] Seq=0 Len=0
171	30.553167	117.199.196.231	192.168.1.2	TCP	4362 > http [SYN] Seq=0 Len=0
172	30.553235	195.246.85.103	192.168.1.2	TCP	59669 > http [SYN] Seq=0 Len=0
173	30.553303	142.159.49.122	192.168.1.2	TCP	4878 > http [SYN] Seq=0 Len=0
174	30.553369	167.234.54.155	192.168.1.2	TCP	53281 > http [SYN] Seq=0 Len=0
175	30.553437	28.96.51.165	192.168.1.2	TCP	10263 > http [SYN] Seq=0 Len=0
176	30.553505	173.153.16.144	192.168.1.2	TCP	1558 > http [SYN] Seq=0 Len=0
177	30.553573	62.5.88.3	192.168.1.2	TCP	5668 > http [SYN] Seq=0 Len=0
178	30.553641	154.197.227.238	192.168.1.2	TCP	50182 > http [SYN] Seq=0 Len=0
179	30.553708	115.125.203.163	192.168.1.2	TCP	47653 > http [SYN] Seq=0 Len=0
180	30.553775	40.75.225.94	192.168.1.2	TCP	9251 > http [SYN] Seq=0 Len=0
181	30.553844	191.122.70.242	192.168.1.2	TCP	51460 > http [SYN] Seq=0 Len=0
182	30.553911	221.159.184.236	192.168.1.2	TCP	12807 > http [SYN] Seq=0 Len=0
183	30.553979	72.43.52.160	192.168.1.2	TCP	58374 > http [SYN] Seq=0 Len=0
184	30.554046	33.59.242.4	192.168.1.2	TCP	18950 > http [SYN] Seq=0 Len=0
185	30.554113	31.118.166.234	192.168.1.2	TCP	61955 > http [SYN] Seq=0 Len=0
186	30.554181	157.18.101.127	192.168.1.2	TCP	3857 > http [SYN] Seq=0 Len=0
187	30.554249	75.63.233.145	192.168.1.2	TCP	29959 > http [SYN] Seq=0 Len=0
188	30.554316	9.111.208.193	192.168.1.2	TCP	60685 > http [SYN] Seq=0 Len=0
189	30.554384	255.10.134.52	192.168.1.2	TCP	56089 > http [SYN] Seq=0 Len=0
190	30.554451	179.84.227.165	192.168.1.2	TCP	22541 > http [SYN] Seq=0 Len=0
191	30.554519	13.196.205.179	192.168.1.2	TCP	42266 > http [SYN] Seq=0 Len=0
192	30.554587	231.80.192.76	192.168.1.2	TCP	40735 > http [SYN] Seq=0 Len=0
193	30.554655	49.151.14.26	192.168.1.2	TCP	9734 > http [SYN] Seq=0 Len=0

⊞ Frame 74 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 3com_c0:6e:9f (00:01:02:c0:6e:9f), Dst: dell_79:22:64 (00:14:22:79:22:64)
⊞ Internet Protocol, Src: 201.79.114.58 (201.79.114.58), Dst: 192.168.1.2 (192.168.1.2)
⊞ Transmission Control Protocol, Src Port: 21767 (21767), Dst Port: http (80), Seq: 0, Len: 0



Analyzing

- Upon analyzing the data captured, we find that the attacker sends packets at a rate of 13568/s, with the size of each packet being 60 bytes.
- It takes approximately 21 packets to consume a 10 Mbps line, causing our server to stop answering any requests. This attack would theoretically have accomplished this at 0.0015 seconds;
- However, due to processing time and propagation delay, our client does not receive notification of the crash until 0.0029 seconds.

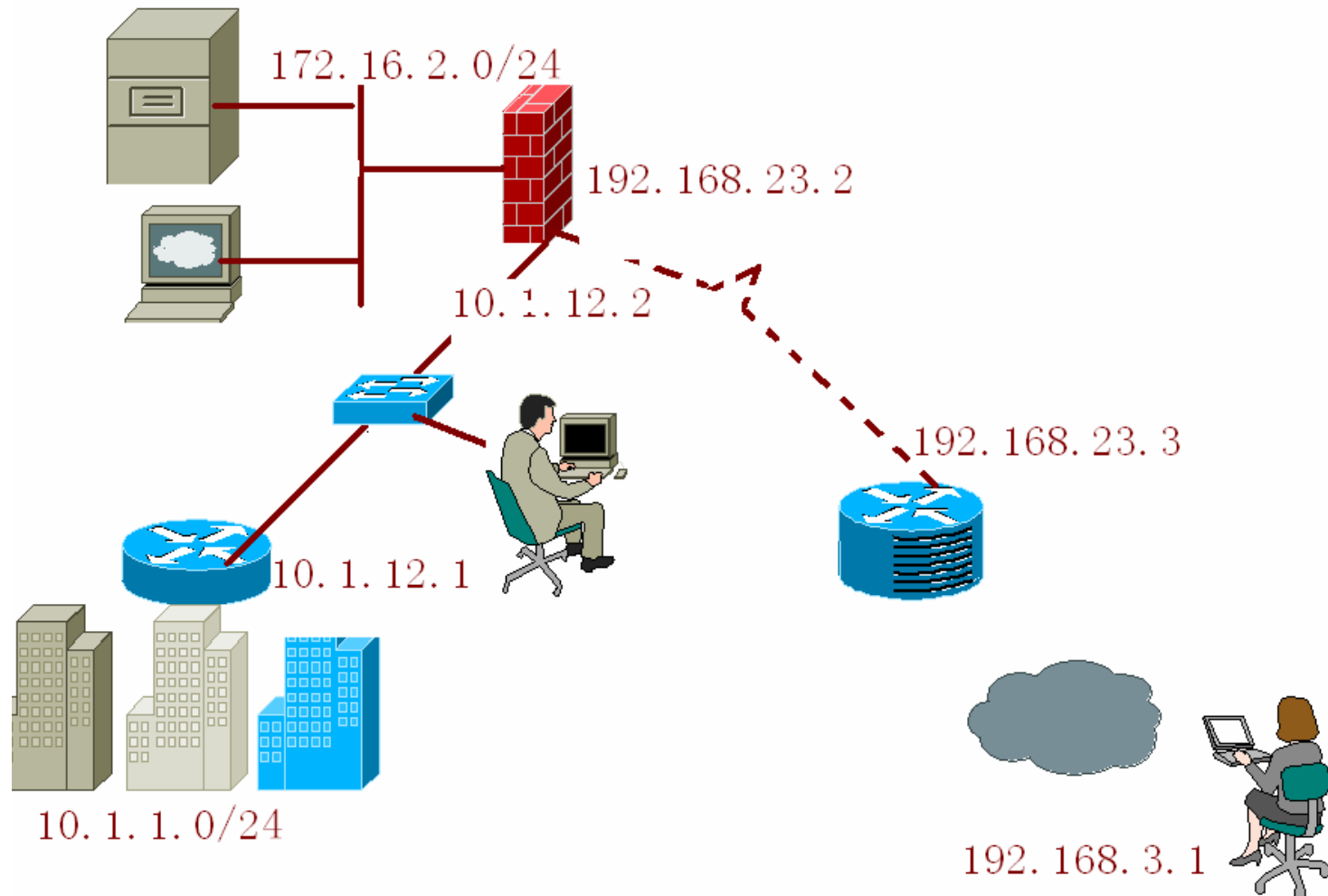


Defend Solution 2--SYN Cookies

- *Shipped with Linux and FreeBSD, but unfortunately not enabled by default*
- *Accepts SYN even if table is full, simply don't keep state -> reconstruct using cookie(seq#)*
- *# echo
1 > /proc/sys/net/ipv4/tcp_syncookies*



Attack 2—Target is on the Network



Lab Requirement for Attack 2

- There are three segments of network— Inside, outside, and DMZ.
- Inside network is the network we need protect.
- DMZ has web server and other services that can be reached both from inside and outside.
- We use CISCO routers 7200 running IOS 12.4 for this attack.



Solution 1--CBAC Firewall

- CBAC will check the access control list first, if the packets don't match the list, the packets are dropped.
- If match, CBAC inspects all the outgoing packets and maintains state information for every session. CBAC create temporary openings for outbound traffic at the firewall interface.
- The return traffic is allowed in only if it is the part of the original outgoing traffic.



Solution 1--CBAC Firewall

```
Telnet localhost
no ip http secure-server

access-list 100 remark auto generated by SDM firewall configuration
access-list 100 remark SDM_ACL Category=1
access-list 100 deny ip 172.16.2.0 0.0.0.255 any
access-list 100 deny ip host 255.255.255.255 any
access-list 100 deny ip 127.0.0.0 0.255.255.255 any
access-list 100 permit ip any any
access-list 101 remark auto generated by SDM firewall configuration
access-list 101 remark SDM_ACL Category=1
access-list 101 deny ip any any log
access-list 102 remark auto generated by SDM firewall configuration
access-list 102 remark SDM_ACL Category=1
access-list 102 deny ip 172.16.2.0 0.0.0.255 any
access-list 102 deny ip 10.1.12.0 0.0.0.255 any
access-list 102 permit icmp any host 192.168.23.2 echo-reply
access-list 102 permit icmp any host 192.168.23.2 time-exceeded
access-list 102 permit icmp any host 192.168.23.2 unreachable
access-list 102 permit tcp any host 172.16.2.10 eq www
access-list 102 permit tcp any host 172.16.2.11 eq www
access-list 102 permit tcp any host 172.16.2.12 eq www
access-list 102 permit tcp any host 172.16.2.13 eq www
access-list 102 permit tcp any host 172.16.2.14 eq www
access-list 102 permit tcp any host 172.16.2.15 eq www
access-list 102 permit tcp any host 172.16.2.16 eq www
access-list 102 permit tcp any host 172.16.2.17 eq www
access-list 102 permit tcp any host 172.16.2.18 eq www
access-list 102 permit tcp any host 172.16.2.19 eq www
access-list 102 permit tcp any host 172.16.2.20 eq www
access-list 102 deny ip 10.0.0.0 0.255.255.255 any
access-list 102 deny ip 172.16.0.0 0.15.255.255 any
access-list 102 deny ip 192.168.0.0 0.0.255.255 any
access-list 102 deny ip 127.0.0.0 0.255.255.255 any
access-list 102 deny ip host 255.255.255.255 any
access-list 102 deny ip host 0.0.0.0 any
access-list 102 deny ip any any log
```



Solution 1--CBAC Firewall

```
Telnet localhost
?
:
:
interface Loopback0
description $FW_DMZ$
ip address 172.16.2.1 255.255.255.0
ip access-group 101 in
ip inspect dmzinspect out
?
interface FastEthernet0/0
description $FW_INSIDE$
ip address 10.1.12.2 255.255.255.0
ip access-group 100 in
ip inspect appfw_100 in
duplex half
?
interface Serial1/0
description $FW_OUTSIDE$
ip address 192.168.23.2 255.255.255.0
ip access-group 102 in
ip verify unicast reverse-path
serial restart-delay 0
clock rate 64000
?
interface Serial1/1
no ip address
shutdown
serial restart-delay 0
?
interface Serial1/2
no ip address
shutdown
serial restart-delay 0
?
interface Serial1/3
no ip address
shutdown
serial restart-delay 0
?
```



Solution 2– Intrusion Prevention System(IPS)

- The Intrusion Detection system is an add-on module to the IOS Firewall Feature Set. It has 59 of the most common attack signatures to detect intrusion. When IPS detects suspicious activity, it logs the event and can either shut down the port or send an alarm before network security is compromised.



Solution 2– Intrusion Prevention System (IPS)

```
Telnet localhost
?
?
interface FastEthernet0/0
 ip address 192.168.12.2 255.255.255.0
 ip virtual-reassembly
 duplex half
?
interface Serial1/0
 ip address 192.168.23.2 255.255.255.0
 ip ips sdm_ips_rule in
 ip virtual-reassembly
 serial restart-delay 0
 clock rate 64000
?
interface Serial1/1
 no ip address
 ip virtual-reassembly
 shutdown
 serial restart-delay 0
?
interface Serial1/2
 no ip address
 ip virtual-reassembly
 shutdown
 serial restart-delay 0
```



Signature is triggered

```
Telnet localhost
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.063: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.095: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.279: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.331: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.387: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.435: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.483: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.519: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.623: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.727: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.875: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
*Dec  2 19:13:46.911: %IPS-4-SIGNATURE: Sig:2004 Subsig:0 Sev:2 ICMP Echo Req [1
192.168.23.3:0 -> 192.168.12.1:0]
```



Attacking is failing...

```
root@Router2:~/datapool
File Edit View Terminal Go Help
Running SYN flooder (synful)...
Continuing attack...he's not dead yet...
Launching 1 attack(s) at 192.168.1.2 on port: 80
Running SYN flooder (synful)...
Continuing attack...he's not dead yet...
Launching 1 attack(s) at 192.168.1.2 on port: 80
Running SYN flooder (synful)...
Continuing attack...he's not dead yet...
Launching 1 attack(s) at 192.168.1.2 on port: 80
Running SYN flooder (synful)...
Continuing attack...he's not dead yet...
Launching 1 attack(s) at 192.168.1.2 on port: 80
Running SYN flooder (synful)...
Continuing attack...he's not dead yet...
Launching 1 attack(s) at 192.168.1.2 on port: 80
Running SYN flooder (synful)...
Continuing attack...he's not dead yet...
Launching 1 attack(s) at 192.168.1.2 on port: 80
Running SYN flooder (synful)...
Continuing attack...he's not dead yet...
Launching 1 attack(s) at 192.168.1.2 on port: 80
Running SYN flooder (synful)...
Continuing attack...he's not dead yet...
Launching 1 attack(s) at 192.168.1.2 on port: 80
Running SYN flooder (synful)...
Continuing attack...he's not dead yet...
Killed
[root@Router2 datapool]#
```



Build A free DoS Attack World

- Customer side—Be a good citizen. How? Using Egress Filtering: Authenticate Source IP of locally generated packets.
- ISP side-Using Ingress Filtering: Authenticate source IP of packets from customer.
- Host—updated OS, patches.
- Stateful Firewall inspect incoming and outgoing packets and create temporary hole in the firewall.
- IPS-An ounce of prevention is worth a pound of cure.



Summary

- Denial of Service attacks represent a fundamental threat to today's Internet
- DoS attacks cost significant losses
- Rate-limiting
- SYN cookies
- Firewall
- IPS



Reference

- [1] <http://www.ethereal.com>
- [2] <http://packetstormsecurity.org/DoS/datapool2.0.tar.gz>
- [3] *TCP-LP: A Distributed Algorithm for Low Priority Data Transfer*, In *IEEE INFOCOM 2003*.
- [4] A. Kuzmanovic and E. Knightly. Low-Rate TCP-Targeted Denial of Service Attacks. In Proceedings of ACM SIGCOMM '03, Karlsruhe, Germany, August 2003.
- [5] <http://www.cisco.com>
- [6] <http://www.cert.org/>
- [7] <ftp://ftp.isi.edu/in-notes/rfc2267.txt>

