Network Security
-- developing attacks

Xu Zhang
University of New Mexico

Contents

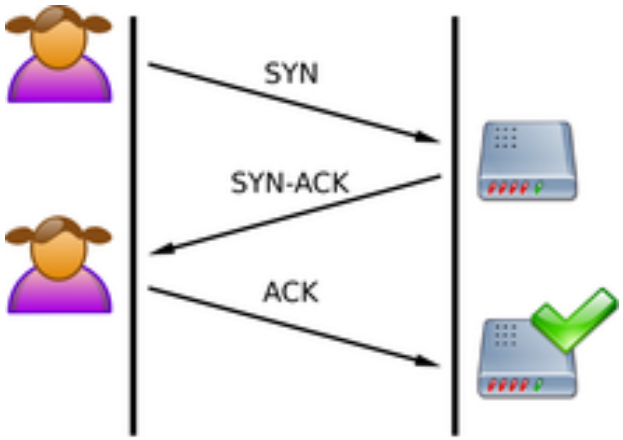
- SYN flood attack
- Evading IDS
- Network RE

Contents

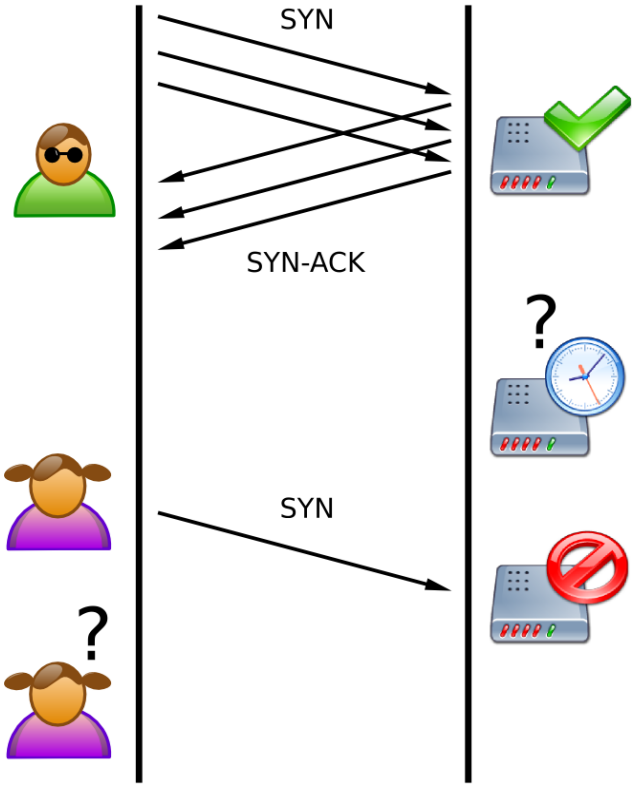
- SYN flood attack
- Evading IDS
- Network RE

How SYN flood attack works?

Normal connection



SYN flood attack



Countermeasures

- SYN backlog
- SYN cache
- SYN cookies

SYN backlog (linux)

- “Linux backlog size is the maximum number of queued connection requests which have still not received an acknowledgement from the connecting client.”[1]
- “If this number is exceeded, the kernel will begin dropping requests.” [1]
- “The default value is 1024 when the memory in the system is adequate or greater (>= 128Mb), and reduced to 128 for those systems with very low memory (<= 32Mb).”[1]
- In ***/proc/sys/net/ipv4/tcp_max_syn_backlog***, can be changed to other value

SYN cache (freeBSD)

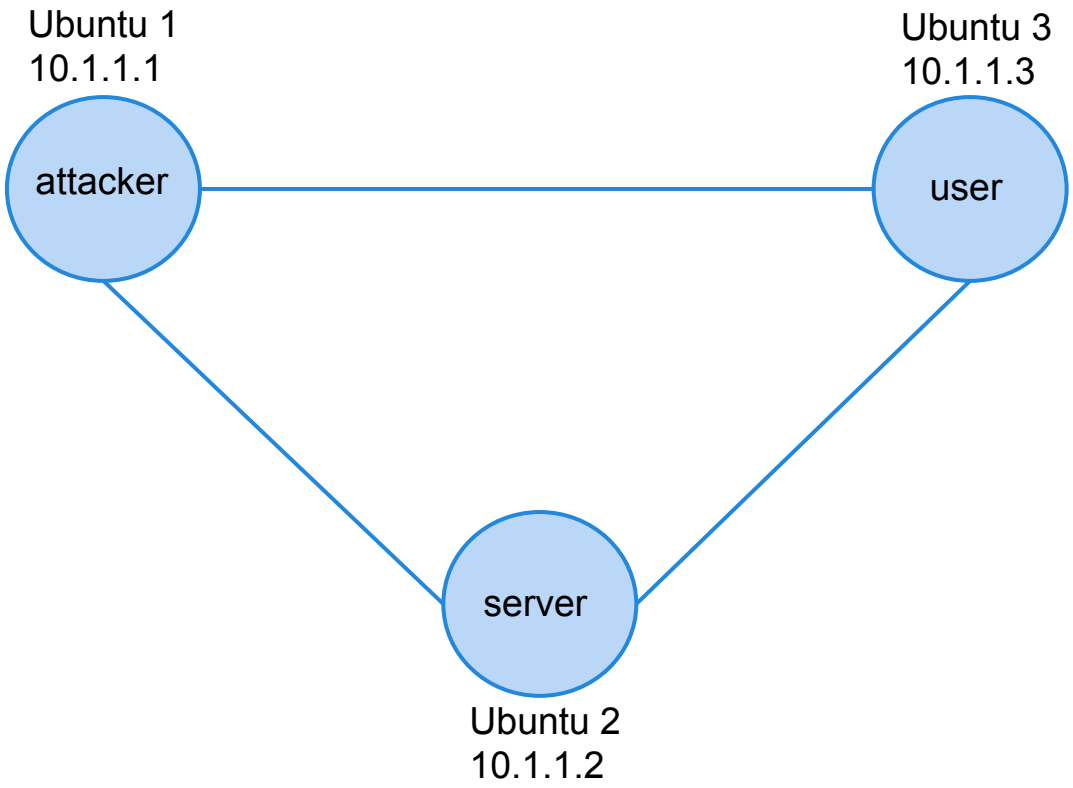
- “The syncache implementation replaces the per-socket linear chain of incomplete queued connections with a global hash table.”[2]
- “FreeBSD first selects secret bits from incoming SYN segment. Then the secret bits are hashed along with the IP addresses and TCP ports of a segment, and the hash value determines the location in a global hash table where the incomplete TCB is stored.”[2]
- ***net.inet.tcp.syncache.hashsize = 512***
- ***net.inet.tcp.syncache.bucketlimit = 30***

SYN cookies

- “SYN cookies go a step further and allocate no state at all for connections in SYN-RECEIVED. Instead, they encode most of the state (and all of the strictly required) state that they would normally keep into the sequence number transmitted on the SYN-ACK.” [2]
- to enable syn cookies in linux, go to /etc/sysctl.conf and uncomment ***net.ipv4.tcp.syncookies=1***
- to enable syn cookies in freeBSD, do:

  sysctl net.inet.tcp.syncookies=1

Demo -- SYN flood attack



Contents

- SYN flood
- Evading IDS
- Network RE

What is IDS by the way?

“An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station.”[3]

Type of attacks to NIDS

- Insertion
- Evasion
- Denial of Service

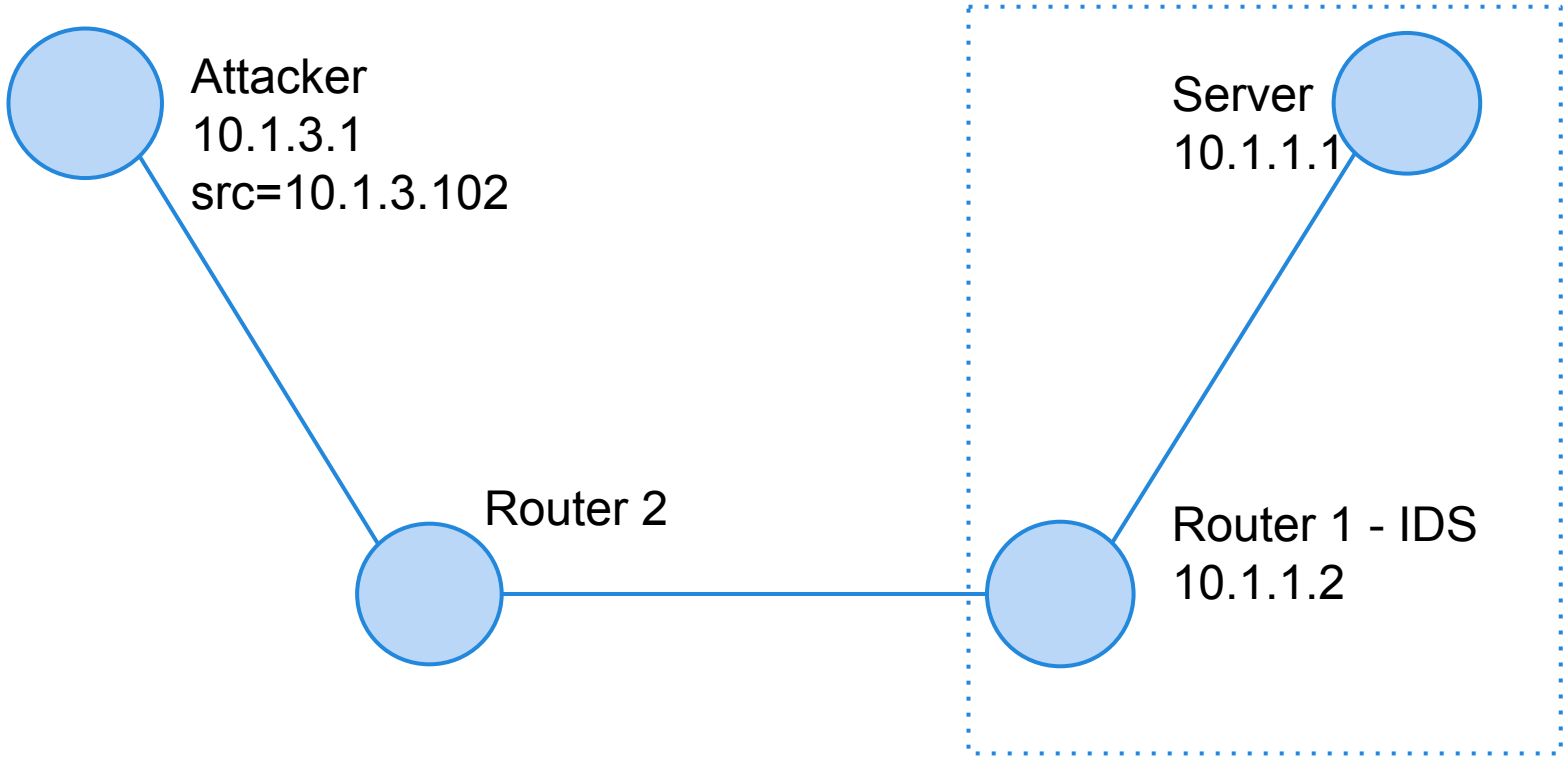
A Evasion demo to NIDS -- bro

1. Open Source, Unix based NIDS, passively monitors network traffic

```
root@ubuntu2:/usr/local/bro/logs/2013-04-09# ls
communication.03:00:00-04:00:00.log.gz  communication.20:00:00-21:00:00.log.gz
communication.04:00:00-05:00:00.log.gz  communication.21:00:00-22:00:00.log.gz
communication.05:00:00-06:00:00.log.gz  communication.22:00:00-23:00:00.log.gz
communication.06:00:00-07:00:00.log.gz  communication.23:00:00-00:00:00.log.gz
communication.07:00:00-08:00:00.log.gz  conn.03:30:16-04:00:00.log.gz
communication.08:00:00-09:00:00.log.gz  conn.04:00:00-05:00:00.log.gz
communication.09:00:00-10:00:00.log.gz  conn.05:00:00-06:00:00.log.gz
communication.10:00:00-11:00:00.log.gz  conn.06:00:00-07:00:00.log.gz
communication.11:00:00-12:00:00.log.gz  conn-summary.03:30:16-04:00:00.log.gz
communication.12:00:00-13:00:00.log.gz  conn-summary.04:00:00-05:00:00.log.gz
communication.13:00:00-14:00:00.log.gz  conn-summary.05:00:00-06:00:00.log.gz
communication.14:00:00-15:00:00.log.gz  conn-summary.06:00:00-07:00:00.log.gz
communication.15:00:00-16:00:00.log.gz  known_hosts.04:02:08-05:00:00.log.gz
communication.16:00:00-17:00:00.log.gz  known_services.04:23:14-05:00:00.log.gz
communication.17:00:00-18:00:00.log.gz  weird.04:18:17-05:00:00.log.gz
communication.18:00:00-19:00:00.log.gz  weird.05:00:00-06:00:00.log.gz
communication.19:00:00-20:00:00.log.gz  weird.06:00:00-07:00:00.log.gz
```

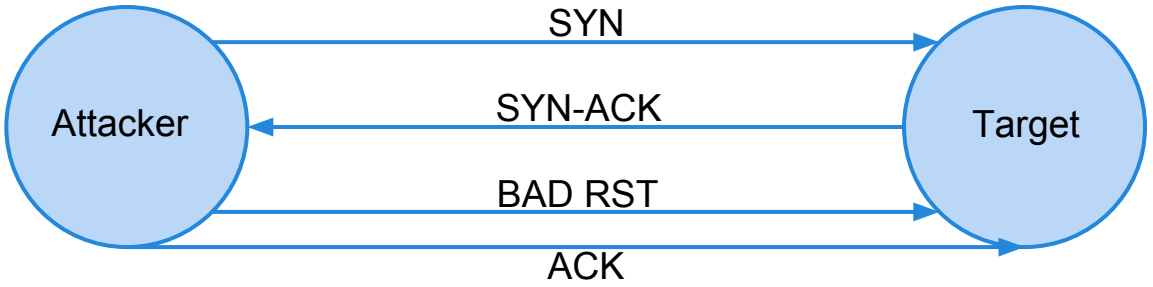
```
path   weird
open   2013-04-09-04-18-17
fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp
       name    addl    notice  peer
types  time    string  addr    port    addr    port    string  string  bool
tring
1365502697.421689       MH6E1HW52G1     10.1.3.102      1234    10.1.1.1
1080    SYN_after_reset -       F       bro
1365502697.569992       MH6E1HW52G1     10.1.3.102      1234    10.1.1.1
1080    RST_storm       -       F       bro
```

Network Environment



What we did

Method 1:



Method 2:

What IDS saw?

```
8080    tcp     -       0.004457        0       0       RSTO    F       0
ShR     2       92      1       56      (empty)
1367289164.197982       tePB3BBNNVk     10.1.3.102      1234    10.1.1.1
8080    tcp     -       0.007333        0       0       RSTO    F       0
ShR     2       92      1       56      (empty)
1367289164.210937       HvKOJ468oqj     10.1.3.102      1234    10.1.1.1
8080    tcp     -       0.006390        0       0       RSTO    F       0
ShR     2       92      1       56      (empty)
1367289164.223139       gNqWlSb0nva     10.1.3.102      1234    10.1.1.1
8080    tcp     -       0.006182        0       0       RSTO    F       0
ShR     2       92      1       56      (empty)
1367289164.237878       Ca1YTNsKVf9     10.1.3.102      1234    10.1.1.1
8080    tcp     -       0.006695        0       0       RSTO    F       0
ShR     2       80      1       44      (empty)
1367289164.248621       OePDXC1ExL3     10.1.3.102      1234    10.1.1.1
8080    tcp     -       3.360510        0       0       RSTR    F       0
ShArrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr    707
28280   707     28288   (empty)
~
                                                              445,1         Bot
```

How IDS (bro) works?

1. IDS set flag in conn.log based on SYN and SYN_ACK
2. That is to say after it saw syn, syn_ack
   + rst from sender, (RSTO)
   + rst from receiver,(RSTR)

   It thinks connection has already been closed.

Contents

- SYN flood attack
- Evading IDS
- Network RE

Demo-- (Tracer Fire problems)

1. What's Alice's email address?(250)

2. What sort of plants does this farm grow? (300)

3. What is Domo in now?(400)

Thank you!

References

[1]http://unixhelp.ed.ac.uk/CGI/man-cgi?tcp

[2]http://www.ietf.org/rfc/rfc4987.txt

[3]http://en.wikipedia.org/wiki/Intrusion_detection_system