

SYN-Cookies Exploration Lab

Copyright © 2006 Wenliang Du, Syracuse University.

The development of this document is funded by the National Science Foundation's Course, Curriculum, and Laboratory Improvement (CCLI) program under Award No. 0618680 and 0231122. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation. A copy of the license can be found at <http://www.gnu.org/licenses/fdl.html>.

1 Lab Description

The learning objective of this lab is for students to explore the mechanism of SYN cookies in Linux system. SYN flooding is a type of Denial of Service (DoS) attack. When a SYN packet is received by a server, the server allocates some memory in its SYN queue, so the SYN information can be stored. Then, the server generates an ISN (Initial Sequence Number) and sends an acknowledgment to the client, hoping to receive an acknowledgment back from the client to complete the three-way handshake protocol. The server will hold the allocated memory for a period of time. If the expected acknowledge does not come, the memory will be freed after timeout. In a SYN flooding attack, the expected acknowledge never comes; instead the attacker fakes a large number of SYN packets. Because the server has to allocate memory from its SYN queue for each of these faked SYN packets, it can eventually hit exhaust its memory in the SYN queue. As results, any further SYN packet will be dropped due to the lack of memory.

To resist against SYN flooding attacks, a technique called **SYN cookies** was proposed. SYN cookies are used to distinguish an authentic SYN packet from a faked SYN packet. When the server sees a possibility of SYN flooding on a port, it generates a *syn cookie* in place of an ISN, which is transparent to the client. Actually, SYN cookies can be defined as “particular choices of initial TCP sequence numbers by TCP servers”. SYN cookies have the following properties:

1. They are generated when the SYN queue hits the upper limit. The server behaves as if the SYN queue has been enlarged.
2. The generated SYN cookie is used in place of the ISN. The system sends back SYN+ACK response to the client and discards the SYN queue entry.
3. If the server receives a subsequent ACK response from the client, server is able to reconstruct the SYN queue entry using the information encoded in the TCP sequence number.

2 Lab Tasks

2.1 Task 1: SYN Flooding Attacks

You will have to try establishing a legitimate TCP connection once the system is SYN flooded. You should describe your observation with SYN cookies enabled and disabled.

1. *SYN cookies disabled*: Conduct a SYN flooding attack on the Linux System with SYN cookies disabled and describe how the system behaved. You can disable SYN cookies using the following command:

```
# sysctl -w net.ipv4.tcp_syncookies = 0
```

2. *SYN cookies enabled*: Conduct a SYN flooding attack on the Linux System with SYN cookies enabled and describe how the system behaved. You can enable SYN cookies using the following command:

```
# sysctl -w net.ipv4.tcp_syncookies = 1
```

The following guidelines may help conduct the attacks: (This is tested on Fedora Core 4 and 5).

1. Netwag tool 76 can be used to SYN flood a system with a specific destination port and IP address.
2. Firewall may be enabled on the system by default, it has to be disabled using:

```
# /sbin/service iptables stop
```

3. Status of the firewall can be found using:

```
# /sbin/service iptables status
```

4. You can use the following command to check the SYN cookies status:

```
# sysctl net.ipv4.tcp_syncookies
```

5. The following commands may help in checking the status of SYN flooding attacks:

```
# netstat -ant (This may behave differently on vmware  
in showing the open connections)  
# dmesg
```

2.2 Task 2: Exploring the SYN Cookies Implementation

The main goal of this task is to come up with an effective SYN cookies design. The challenge is design a way for the server to generate its ISN, such that SYN flooding attacks will not work.

1. Consider to have a SYN cookie generation equation as follows :

cookie = hash(saddr, daddr, sport, dport) + sseq

where

saddr : Source IP Address

daddr : Destination IP Address

sport : Source Port

dport : Destination Port

sseq : Source Sequence Number.

The “cookie” generated would be the new ISN. This would satisfy the SYN cookie requirements of generating a unique ISN for a unique combination of above parameters. Moreover, it is possible to recalculate the cookie once an ACK is received back and hence regard it as authenticate SYN.

Can you discover if this method introduces any new problems to the system ?

2. Consider a different SynCookie generation equation as follows :
cookie = hash(saddr, daddr, sport, dport, random) + seq
where random : a random number generated at the boot time.
Can you discover if the above equation may introduce any new problems to the system ?
3. Consider one more equation of SynCookie generation:
cookie = hash(saddr, daddr, sport, dport, random) + sseq + count
Consider count to be a number that gets incremented every minute or so.
Do you think the above equation may still be a threat to the sytem at any given point of time ?
4. If you think the third equation may still be a threat, can you come up with a new equation to satisfy all the requirements of SynCookies ? You also need to elaborate as to how to recalculate the cookie once an ACK is received back to regard the connection to be authentic.

3 Helpful Materials

Here are some links that might help you discover answers for the above questions:

1. Current implementation of SYN cookies in Linux system can be found in the Linux source code at *net/ipv4/syncookies.c*.
2. <http://cr.yip.to/syncookies.html>
3. <http://cr.yip.to/syncookies/archive>
4. www.cs.colorado.edu/~jrblack/class/csci4830/f03/syncookies.pdf

4 Submission

You need to submit a detailed lab report to describe what you have done and what you have observed; you also need to provide explanation to the observations that are interesting or surprising.