Benefits and Drawbacks of SYN Cookies in Linux Kernel

Ali Ghaffarian

Proposal for evaluating SYN cookies in 2024

Contents

Summary

In an estimated three-week time period, we will examine the costs and benefits of SYN cookies as well as the effects of SYN floods on a modern GNU/Linux host with SYN cookies disabled and the viability of previously known attacks against SYN cookies, such as connection spoofing.

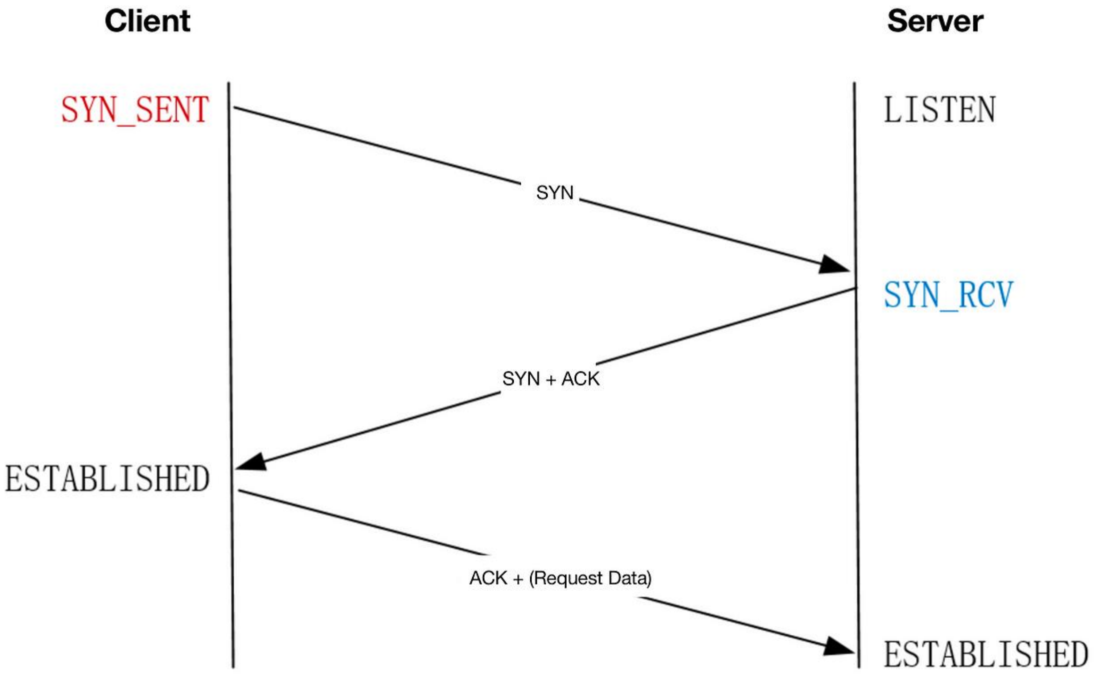
Introduction

TCP

Transmission Control Protocol is the most used reliable data transfer protocol in TCP/IP networks and considered a complex one.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             SPORT             |             DPORT             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              SEQ                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ACK                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|DATAOFS|RESER|      FLAGS      |             WINDOW            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             CHKSUM            |             URGPTR            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            OPTIONS            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

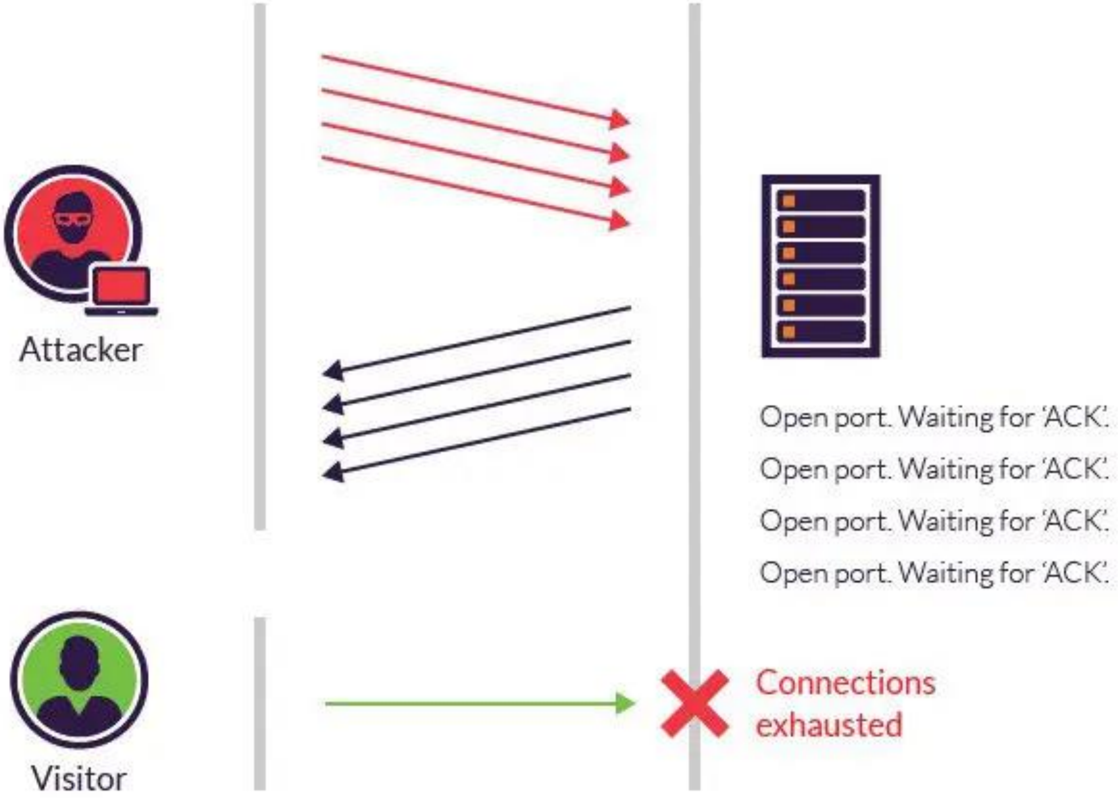
Three Way Handshake



[1]

The two ends of the TCP connection must know each other's initial sequence number before the actual reliable data transfer can begin, one of the responsibilities of TCP three way handshake is exactly this.

SYN Floods


Attacker
Open port. Waiting for 'ACK'.
Open port. Waiting for 'ACK'.
Open port. Waiting for 'ACK'.
Open port. Waiting for 'ACK'.
Visitor
Connections exhausted

[2]

TCP SYN flooding is a DoS attack that leverages the reserved resources for pending connections.

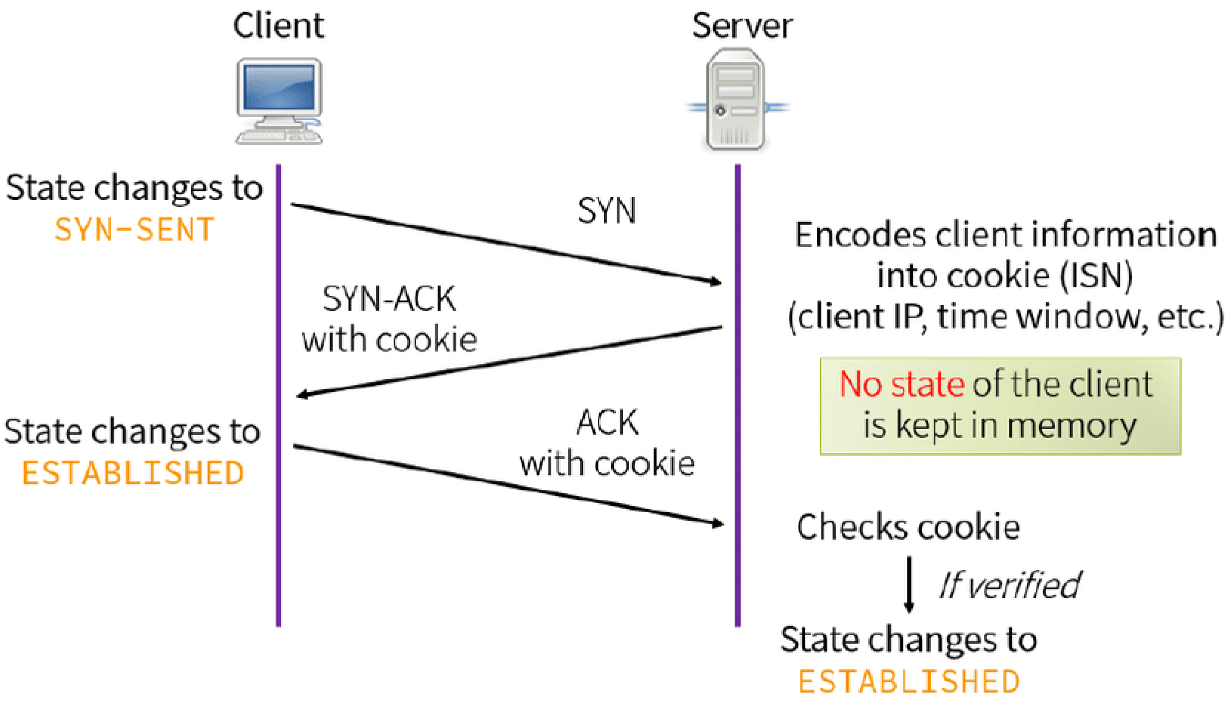
Because TCP servers normally reserve such resources only when a connection is initiated, the attacker can flood the server with SYN requests that include spoofed source IP address.

This way the backlog of the TCP server will eventually fill up, making the server always waiting for the non-existent clients to complete their connections.

When using the server's timeout and backlog as the attack vector, the cost of the attack drops dramatically, as a typical TCP server will wait several seconds until it determines that the client's not going to complete the connection.

In some circumstances, the server's memory will get full if the backlog is big enough.

SYN Cookies



[3]

SYN floods were so successful because of the stateful nature of the TCP protocol.

One way to stop SYN floods from being so cheap and effective, is to handle the three-way handshake differently, while preserving the rest of the protocol.

In other words, handle the handshake statelessly without breaking the protocol, that is, to forget the client ever initiated a connection, but having a method to recover all connection's crucial data when the client completes the connection.

SYN cookies do this by carefully encoding the connection's data in the server's initial sequence number (ISN) when acknowledging the client's SYN request.

This way when the client sends the final packet of the three-way handshake all of the required data to reconstruct the connection will be recoverable.

TCP Crucial Fields

The data a TCP server needs to be able to reconstruct the connection includes:

- client’s ISN
- server’s ISN
- client’s IP address
- server’s IP address
- client’s port number
- server’s port number
- max segment size (MSS)

Client’s ISN and server’s MSS are encoded into server’s ISN; The rest can already be recovered from client’s ACK packet.

Encoding TCP options

TCP is an old transport layer protocol, it was designed for networks of its time that tended to lose packets frequently and have lower bandwidth, in which devices reside with small buffers to dedicate to the connection, because of this certain TCP options are needed to be negotiated so that modern devices can use all of their offered load while using TCP as the transport layer protocol.

The most important of which includes:

- Window Scale: each device advertises a window scale value, indicating each declared window size needs to be multiplied by 2^window scale.
- Selective Acknowledgement (SACK): when using SACK as the loss recovery each device can declare the sequence number of the lost data and continue the data transfer without needing to slow down.

Linux encodes these two options in the timestamp field of the SYN-ACK packet only if both devices support timestamps [4].

Research Goals

We try to answer the following questions:

1. How does SYN floods effect a SYN cookie disabled 2024 GNU/Linux server?
2. Are the known attacks against SYN cookies still viable?
3. What are the Drawbacks of SYN cookies in a 2024 GNU/Linux host and how much they impact the performance of the system?

Testing Environment

Rule of the device will probably vary based on the experiment.

Device 1:

```
OS: Arch Linux x86_64
Host: ASUS TUF Gaming F17 FX706HC_FX706HC 1.0
Kernel: 6.11.5-arch1-1
CPU: 11th Gen Intel i5-11400H (12) @ 4.500GHz
Memory: 15732MiB
Network controller: MEDIATEK Corp. MT7921 802.11ax PCI Express Wireless
Network Adapter
```

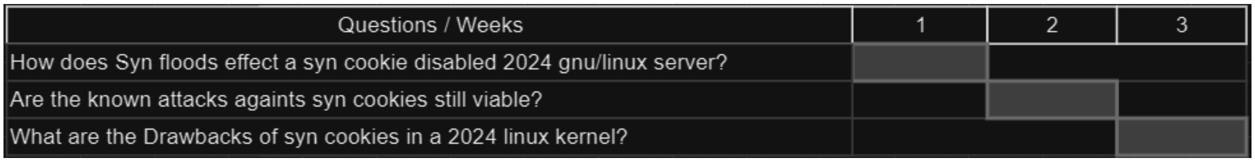
Device 2:

```
OS: Android 13 aarch64
Host: POCO 2201116PG
Kernel: 5.4.259-qgki-gcb8e1baa6b45
CPU: (8) @ 1.804GHz
Memory: 5437MiB
Network controller: Not root on this device, can't check
```

Device 3:

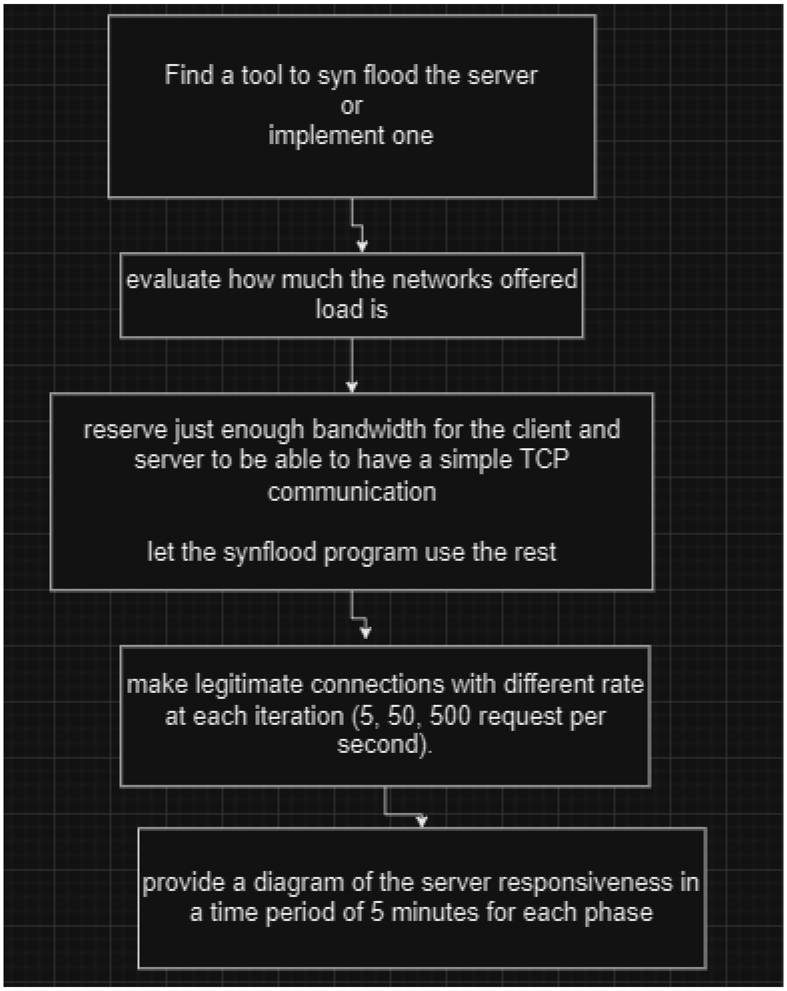
```
OS: Debian GNU/Linux 12 (bookworm) x86_64
Host: Aspire A315-58G V1.26
Kernel: 6.1.0-22-amd64
CPU: 11th Gen Intel i3-1115G4 (4) @ 4.100GHz
Memory: 11739MiB
Network controller: Intel Corporation Wi-Fi 6 AX201
```

Overview of Research Plan



Questions / Weeks	1	2	3
How does Syn floods effect a syn cookie disabled 2024 gnu/linux server?	▨		
Are the known attacks againts syn cookies still viable?		▨	
What are the Drawbacks of syn cookies in a 2024 linux kernel?			▨

Question 1



Find a tool to syn flood the server
or
implement one

evaluate how much the networks offered load is

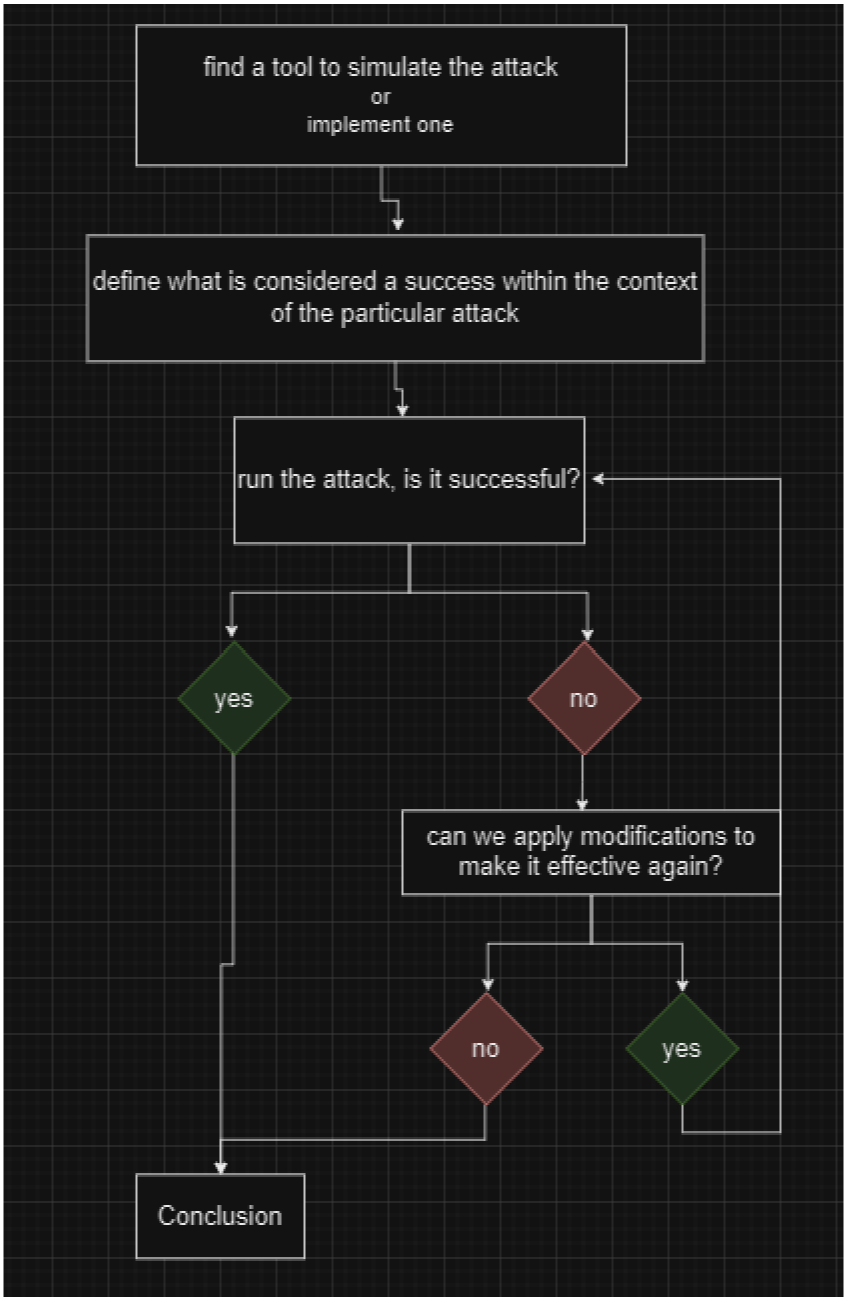
reserve just enough bandwidth for the client and server to be able to have a simple TCP communication

let the synflood program use the rest

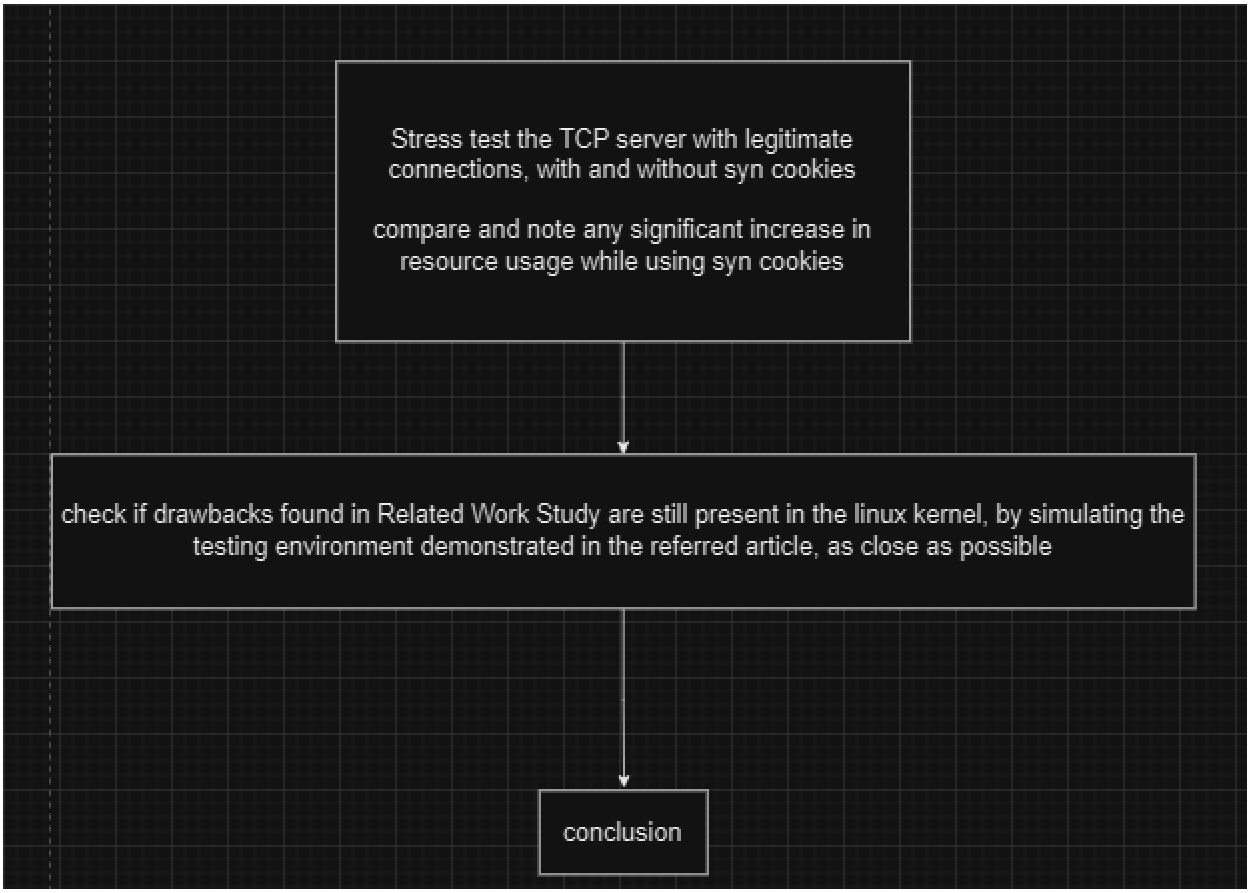
make legitimate connections with different rate at each iteration (5, 50, 500 request per second).

provide a diagram of the server responsiveness in a time period of 5 minutes for each phase

Question 2


find a tool to simulate the attack
or
implement one
define what is considered a success within the context of the particular attack
run the attack, is it successful?
yes
no
can we apply modifications to make it effective again?
no
yes
Conclusion

Question 3



Stress test the TCP server with legitimate connections, with and without syn cookies

compare and note any significant increase in resource usage while using syn cookies

check if drawbacks found in Related Work Study are still present in the linux kernel, by simulating the testing environment demonstrated in the referred article, as close as possible

conclusion

Related Work Study

Linux Kernel

Cookie generation:

The encoded information (client's ISN and MSS) is combined with two cryptographically hashed values (like `HASH(secret , saddr, sport, daddr, dport, secret)`) secret keys and a count value that is incremented every minute to make the cookie and sent to the client [4][5].

Cookie validation is done via extracting the encoded MSS from client's Acknowledge number and checking if it's out of range [5].

Minisocks

Linux stores half opened connections via tcp_request_sock structure which is 96 bytes compared to a regular tcp_sock which is a 1616 structure [4].

Considering how smaller the minisock is, the possibility of running out of memory during a SYN flood is extremely low.

Benefits of SYN Cookies

- no need to track any half-opened connections.

Drawbacks

- Limited support for TCP options.
- SYN cookies are ineffective against SYN flood attacks that target the victim's bandwidth.
- In effective connection spoofing scenario, firewalls focusing on filtering connections by SYN flag will be bypassed.
- Needs more CPU resources to be computed and validated.

Packet Loss??

in 3th march 2022, Kevin Graham stated that because of how MSS is encoded into the server's ISN, a TCP segment with 3 bytes of data can be lost once in a connection that is initiated using SYN cookies [6].

Other operating systems might be suffering from the same problem as well, but due to the nature of close source software, we can’t state anything unless special experiments are done.

Connection Spoofing Attacks

As a Linux TCP server in SYN cookie mode accepts connection after receiving a valid SYN cookie from the client, connections can be made by guessing SYN cookies.

In 13th August 2013, Jakob Lell provided a method to guess a valid SYN cookie in around 8 minutes with a gigabit connection to the target [7].

References

[1] https://miro.medium.com/v2/resize:fit:1400/1*BHjxQmlldZv7Qx6TNCG8TQ.png

[2] https://www.imperva.com/learn/wp-content/uploads/sites/13/2019/01/syn-flood.jpg.webp

[3] https://www.researchgate.net/publication/336872866/figure/fig2/AS:956296866066433@1605010348044/Mechanism-of-SYN-cookies-4.ppm

[4] https://lwn.net/Articles/277146/

[5] https://github.com/torvalds/linux/blob/master/net/ipv4/syncookies.c

[6] https://wpbolt.com/syn-cookies-ate-my-dog-breaking-tcp-on-linux

[7] https://www.jakoblell.com/blog/2013/08/13/quick-blind-tcp-connection-spoofing-with-syn-cookie