Docker-Based SYN Flood Attack Simulation

Kayvon Karimi

Shirley-Macros School of Engineering (SMSE), University of San Diego

CYBR-508: Secure Network Engineering

Professor Templeton

July 21st, 2025

Introduction

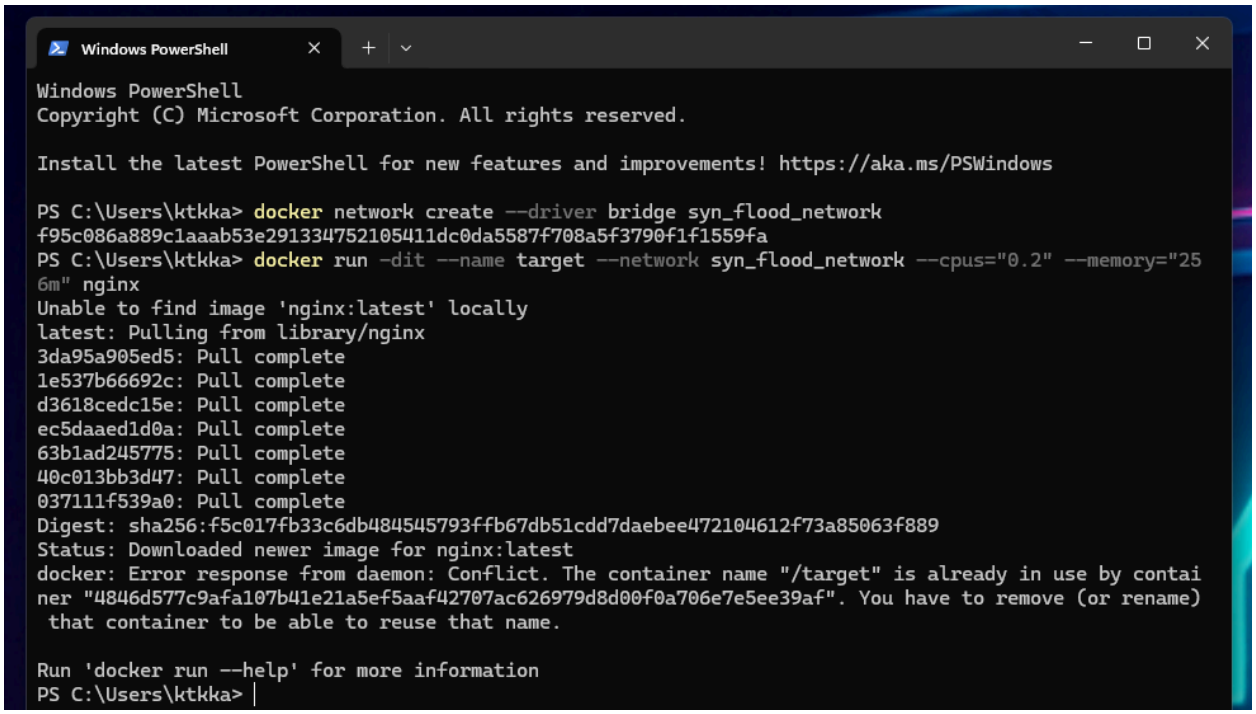
This lab simulated a SYN flood attack using Docker containers to understand Denial of Service (DoS) mechanics. The setup included a target (nginx) and attacker (hping3) on a custom network, monitored with tcpdump and analyzed in Wireshark.

Step 1: Set up Docker Containers

- **Created network:**

Open Docker on home device and open Powershell.

In Powershell, create a network with *'docker network create --driver bridge syn_flood_network'*



- **Configured and configured target:**
    - *'docker run -dit --name target --network syn_flood_network --cpus="0.2" --memory="256m" nginx'*

- apt update && apt install iproute2 -y
- mkdir -p /mnt/data
- apt update && apt install tcpdump -y

```
Run 'docker run --help' for more information
PS C:\Users\ktkka> docker exec -it target bash
root@de08ccb097b2:/# apt update && apt install iproute2 -y
```

```
root@de08ccb097b2:/# mkdir -p /mnt/data
root@de08ccb097b2:/# apt update && apt install tcpdump -y
```

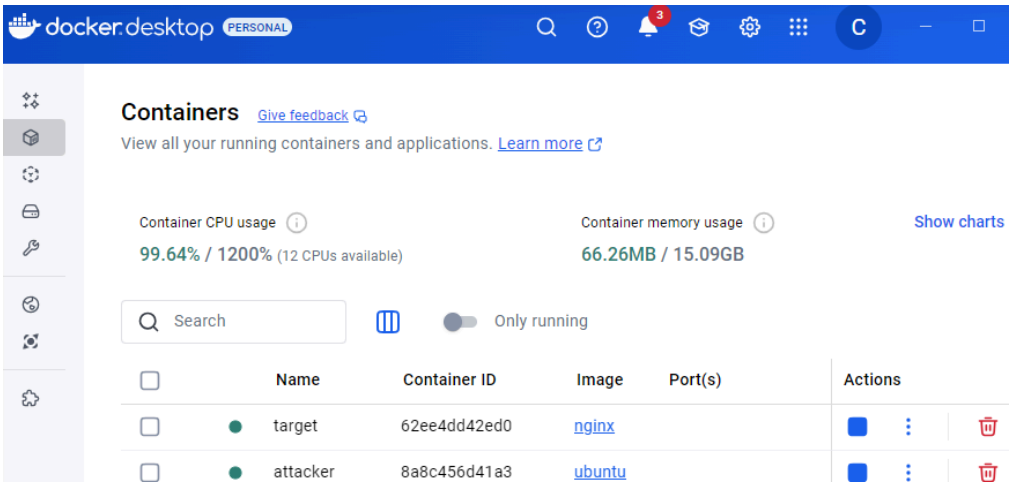
- **Created attacker:**

'docker run -it --name attacker --network syn_flood_network ubuntu /bin/bash'

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ktkka> docker start attacker
attacker
PS C:\Users\ktkka> docker run -it --name attacker --network syn_flood_network ubuntu /bin/bash
root@87fe29ff1e9d:/# apt update && apt install hping3 -y
```



Docker Application on desktop shows target and attacker containers running.

Step 2: Launch the Attack

I ran '*hping3 --flood --syn --destport 80'* target for 30-60 seconds and stopped with ctrl+c.

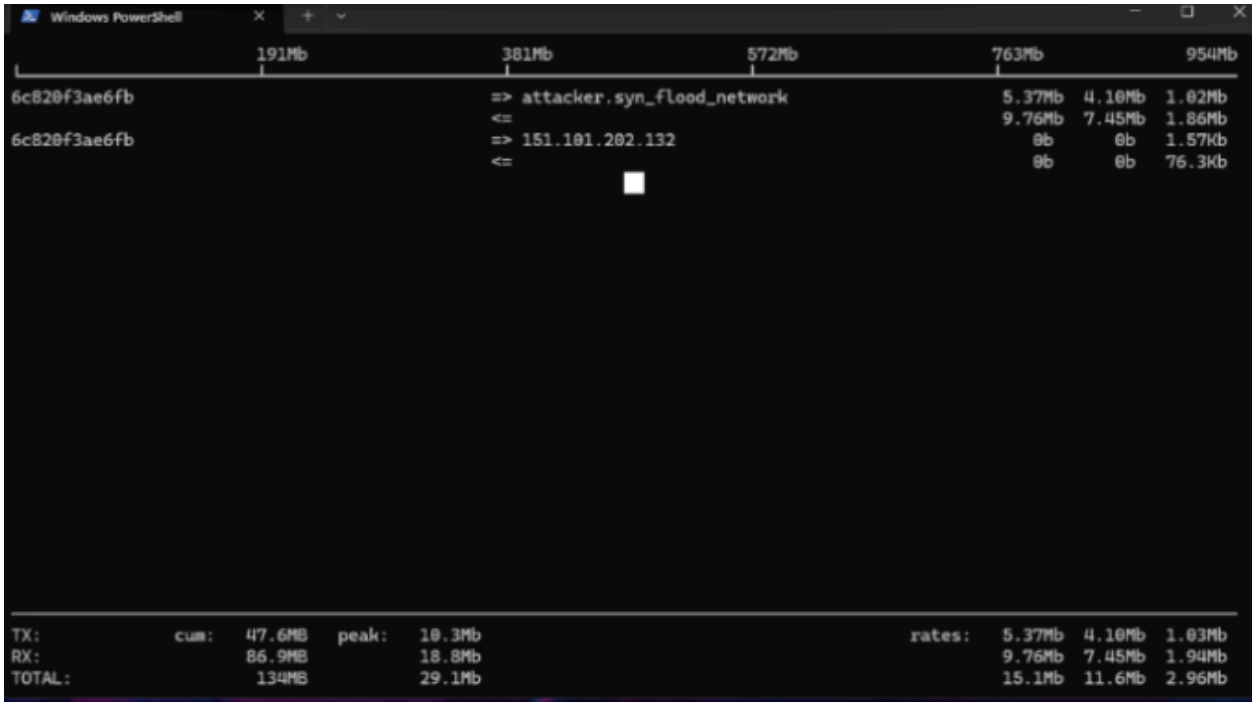
```
root@87fe29ff1e9d:/# hping3 --flood --syn --destport 80 target
HPING target (eth0 172.20.0.2): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Step 3: Monitor with tcpdump

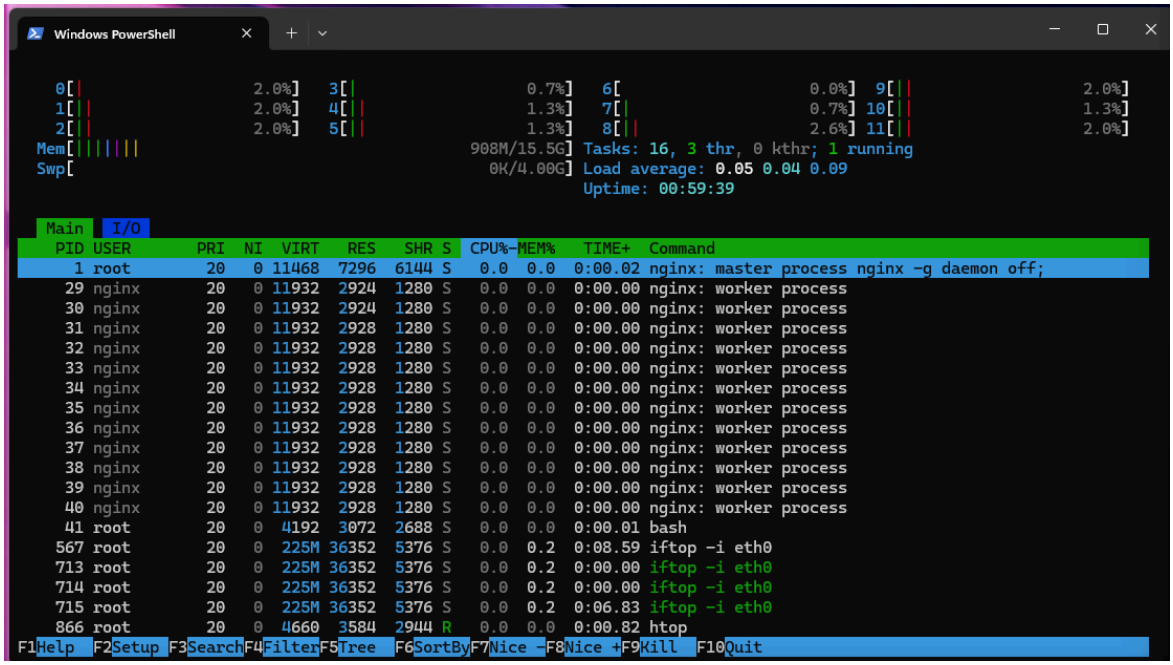
Used the command '*tcpdump -i eth0 -c 980 'tcp[tcpflags] == tcp-syn' and not port 22 -w /mnt/data/syn_flood_capture.pcap*' during attack.

```
PS C:\Users\ktkka> docker exec -it target bash -c "tcpdump -i eth0 'tcp[tcpflags] == tcp-syn' and not port 22 -w /mnt/da
ta/syn_flood_capture.pcap -c 980"
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
980 packets captured
1855 packets received by filter
0 packets dropped by kernel
```

To monitor the SYN flood attack, I re-entered the target container and ran tcpdump to capture SYN packets. The command used was tcpdump -i eth0 'tcp[tcpflags] == tcp-syn' and not port 22 -w /mnt/data/syn_flood_capture.pcap, which listened on the eth0 interface and saved the capture to a PCAP file. During the attack, I also monitored bandwidth usage with iftop in a separate window, which showed increased traffic from the attacker IP (151.101.202.132) to the target.



The tcpdump capture confirmed 980 packets, as limited by the -c 980 option in a 30 sec span.

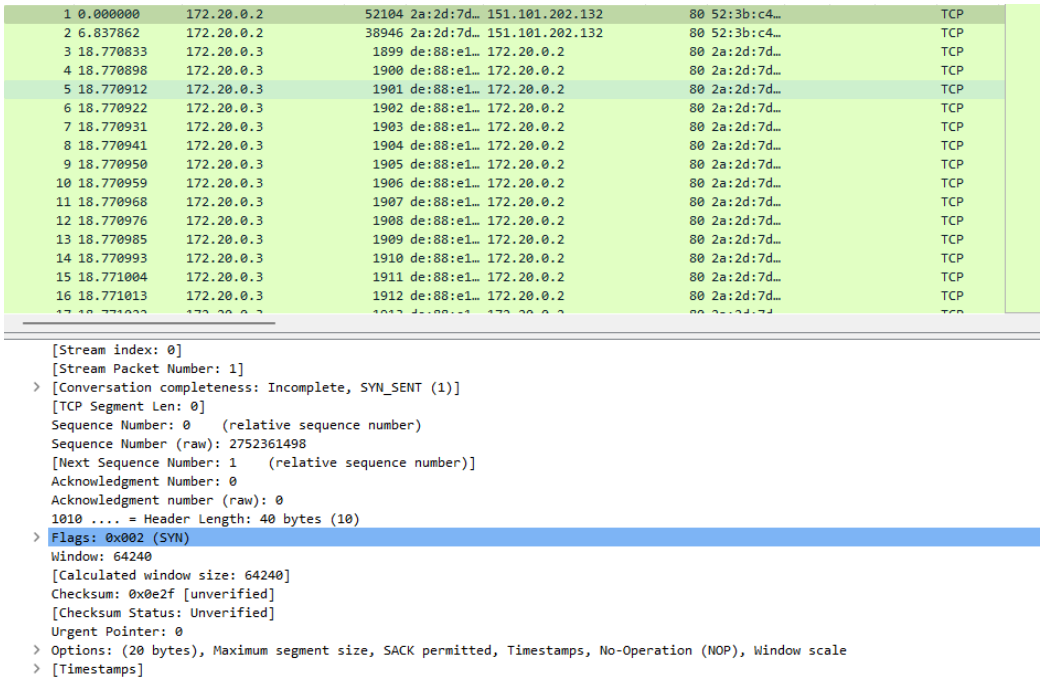


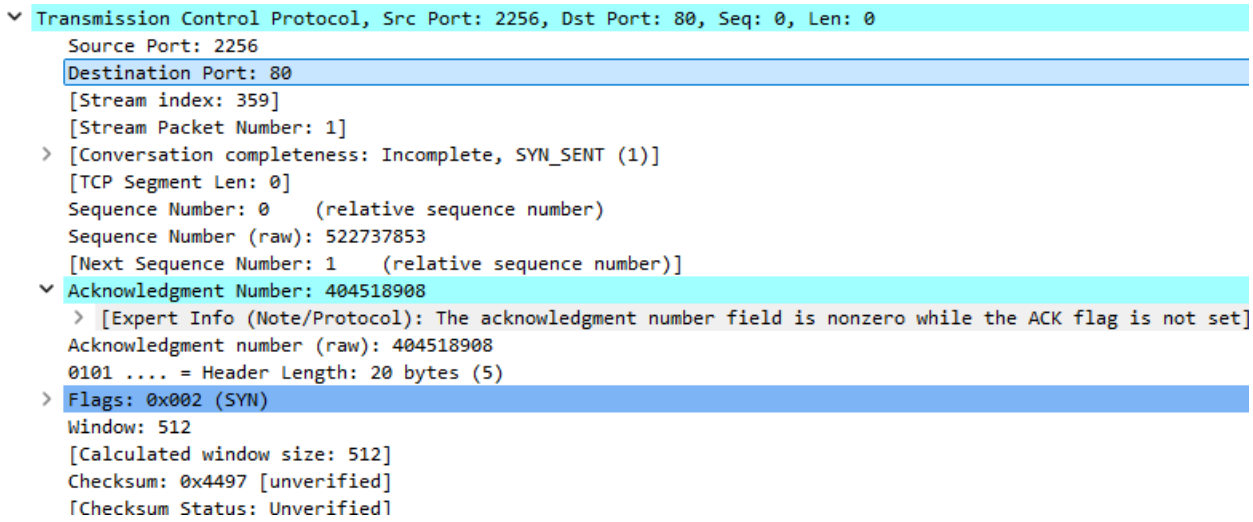
Step 4: Copied file and Opened Wireshark

- Used the '*docker cp target:/mnt/data/syn_flood_capture.pcap .*' command to copy the file.
- Opened Wireshark, filtered for SYN packets

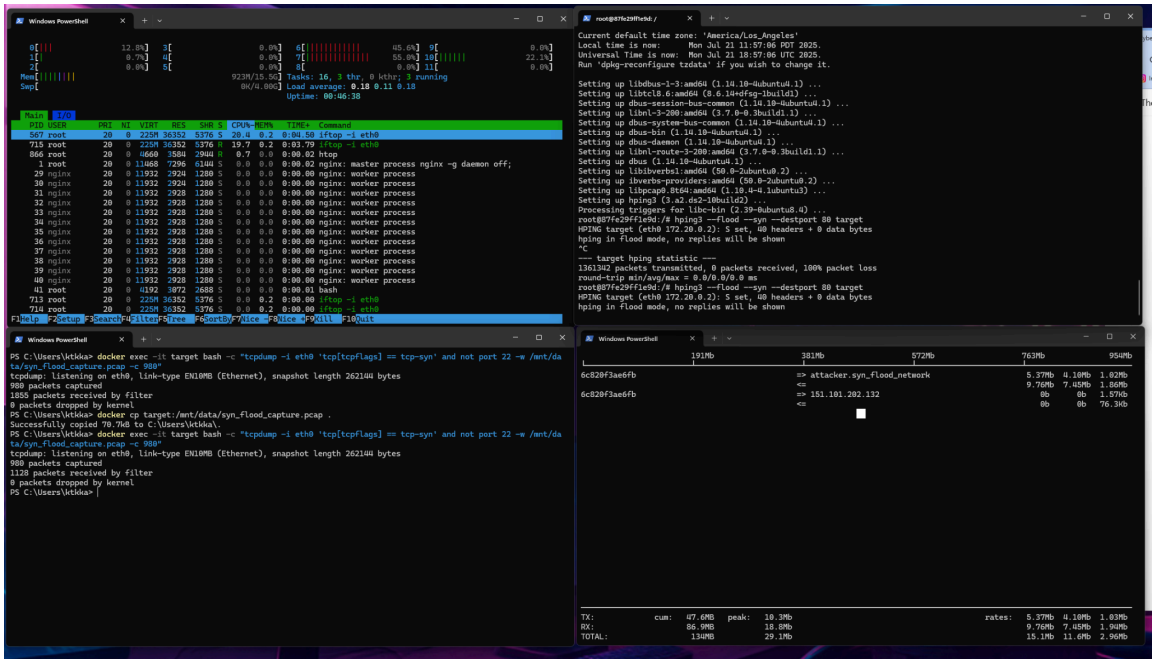
```
PS C:\Users\ktkka> docker cp target:/mnt/data/syn_flood_capture.pcap .
Successfully copied 70.7kB to C:\Users\ktkka\.
```

- Observed a high volume of SYN packets without ACKs, confirming flood





Screenshot overview of 4 windows - target, attacker, tcpdump, and ftop.



Step 5: Clean-Up

Ran '*docker stop target attacker, docker rm target attacker, docker network rm syn_flood_network.*' to clean and remove the containers.

```
PS C:\Users\ktkka> docker container stop target attacker
target
attacker
PS C:\Users\ktkka> docker container rm target attacker
target
attacker
PS C:\Users\ktkka> docker network rm syn_flood_network
syn_flood_network
```

Analysis

This lab walked us through a hands-on experience in simulating a SYN flood attack within a controlled Docker environment. It enhanced my understanding of Denial of Service mechanics

and network security principles. The successful setup of four terminals, target, attacker, tcpdump, and iftop, allowed for real time monitoring of the attack's impact. The wireshark analysis of the syn_flood_capture.pcap file revealed a high volume of SYN packets without corresponding ACKs, which confirms the flood’s effectiveness in overwhelming the target's limited resources. The iftop window highlights bandwidth spikes, reinforcing the attack’s network load. Lastly, the tcpdump window captured the packet flow, limited to 980 packets for manageable analysis. The impact of the attack was visible through increased CPU usage and high bandwidth consumption and rapid growth of captured SYN packets, which are clear indicators of strained systems.

SYN flood attacks are especially dangerous because of their simplicity and low barrier to execution. In practical terms, this type of attack poses serious implications for online services, particularly for websites, e-commerce platforms, or critical infrastructure relying on uninterrupted access. A successful SYN flood could lead to financial losses, reputational damage, and compromised user trust.