PFSENSE SETUP AND DOS DEFENSE DEMO

WRITEUP BY : ALI MALIK

This document outlines the configuration of a pfSense firewall to protect a target system from a TCP SYN flood attack, demonstrating its effectiveness through pre and post-attack analysis.

1. Network Configuration

The network setup involves three virtual machines: Ubuntu, Kali Linux, and pfSense, configured as follows:

Machine	IP Address	Network	Role	Adapter Type
Ubuntu	192.168.18.85	WAN (External)	Attacker	Bridged via VMnet0
Kali	192.168.1.128	LAN (Internal)	Victim	Host-only (VMnet1)
pfSense WAN	192.168.18.87	WAN (bridged)	Firewall	Bridged (VMnet0)
pfSense LAN	192.168.1.1	LAN	Gateway	Host-only (VMnet1)

- **Network Segmentation:** Ubuntu and pfSense WAN are on the same physical network (192.168.18.0/24). Kali and pfSense LAN are on an isolated host-only network (192.168.1.0/24).
- **pfSense Role:** pfSense acts as a bridge and firewall between these two networks , with its WAN interface connected to VMnet0 (Ubuntu and Wi-Fi router) and its LAN interface connected to VMnet1 (Kali).

pfSense Interface Configuration:

Interface	Device	IP Address	Network	Purpose
WAN	em0	192.168.100.101	External (VMnet0)	Ubuntu / Internet side
LAN	em1	192.168.1.1	Internal (VMnet1)	Kali side

2. Baseline Connectivity (Without Block Rules)

Before implementing any firewall rules, a ping test was conducted from the Ubuntu machine (attacker) to the Kali machine (victim) to confirm network connectivity.

- **Ping Command (from Ubuntu):** ping 192.168.1.128

```
ali@ali-VMware-Virtual-Platform:~$ ping 192.168.1.128
PING 192.168.1.128 (192.168.1.128) 56(84) bytes of data.
64 bytes from 192.168.1.128: icmp_seq=1 ttl=63 time=1.56 ms
64 bytes from 192.168.1.128: icmp_seq=2 ttl=63 time=2.70 ms
64 bytes from 192.168.1.128: icmp_seq=3 ttl=63 time=1.46 ms
64 bytes from 192.168.1.128: icmp_seq=4 ttl=63 time=8.23 ms
64 bytes from 192.168.1.128: icmp_seq=5 ttl=63 time=1.27 ms
64 bytes from 192.168.1.128: icmp_seq=6 ttl=63 time=1.53 ms
64 bytes from 192.168.1.128: icmp_seq=7 ttl=63 time=2.42 ms
64 bytes from 192.168.1.128: icmp_seq=8 ttl=63 time=2.68 ms
64 bytes from 192.168.1.128: icmp_seq=9 ttl=63 time=2.94 ms
64 bytes from 192.168.1.128: icmp_seq=10 ttl=63 time=1.72 ms
64 bytes from 192.168.1.128: icmp_seq=11 ttl=63 time=5.46 ms
64 bytes from 192.168.1.128: icmp_seq=12 ttl=63 time=3.07 ms
64 bytes from 192.168.1.128: icmp_seq=13 ttl=63 time=1.66 ms
64 bytes from 192.168.1.128: icmp_seq=14 ttl=63 time=1.08 ms
64 bytes from 192.168.1.128: icmp_seq=15 ttl=63 time=3.11 ms
64 bytes from 192.168.1.128: icmp_seq=16 ttl=63 time=2.70 ms
64 bytes from 192.168.1.128: icmp_seq=17 ttl=63 time=2.97 ms
64 bytes from 192.168.1.128: icmp_seq=18 ttl=63 time=4.07 ms
64 bytes from 192.168.1.128: icmp_seq=19 ttl=63 time=2.56 ms
64 bytes from 192.168.1.128: icmp_seq=20 ttl=63 time=4.08 ms
64 bytes from 192.168.1.128: icmp_seq=21 ttl=63 time=5.85 ms
64 bytes from 192.168.1.128: icmp_seq=22 ttl=63 time=2.83 ms
64 bytes from 192.168.1.128: icmp_seq=23 ttl=63 time=2.58 ms
64 bytes from 192.168.1.128: icmp_seq=24 ttl=63 time=2.92 ms
64 bytes from 192.168.1.128: icmp_seq=25 ttl=63 time=6.98 ms
64 bytes from 192.168.1.128: icmp_seq=26 ttl=63 time=1.59 ms
64 bytes from 192.168.1.128: icmp_seq=27 ttl=63 time=1.96 ms
64 bytes from 192.168.1.128: icmp_seq=28 ttl=63 time=2.62 ms
64 bytes from 192.168.1.128: icmp_seq=29 ttl=63 time=2.24 ms
64 bytes from 192.168.1.128: icmp_seq=30 ttl=63 time=4.39 ms
64 bytes from 192.168.1.128: icmp_seq=31 ttl=63 time=3.79 ms
64 bytes from 192.168.1.128: icmp_seq=32 ttl=63 time=2.74 ms
```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp

No.	Time	Source	Destination	Protocol	Length	Info
7	2.810384420	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=1/256, ttl=63 (reply in 8)
8	2.810431609	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=1/256, ttl=64 (request in 7)
10	3.812188446	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=2/512, ttl=63 (reply in 11)
11	3.812264917	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=2/512, ttl=64 (request in 10)
17	4.815177224	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=3/768, ttl=63 (reply in 18)
18	4.815216407	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=3/768, ttl=64 (request in 17)
19	5.827855960	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=4/1024, ttl=63 (reply in 20)
20	5.827903376	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=4/1024, ttl=64 (request in 19)
22	6.825023442	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=5/1280, ttl=63 (reply in 23)
23	6.825073379	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=5/1280, ttl=64 (request in 22)
24	7.826996385	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=6/1536, ttl=63 (reply in 25)
25	7.827046446	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=6/1536, ttl=64 (request in 24)
27	8.829037938	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=7/1792, ttl=63 (reply in 28)
28	8.829073168	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=7/1792, ttl=64 (request in 27)
31	9.831720352	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=8/2048, ttl=63 (reply in 32)
32	9.831787364	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=8/2048, ttl=64 (request in 31)
40	10.833486741	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=9/2304, ttl=63 (reply in 41)
41	10.833638955	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=9/2304, ttl=64 (request in 40)
43	11.835320749	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=10/2560, ttl=63 (reply in 44)
44	11.835393582	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=10/2560, ttl=64 (request in 43)
46	12.841098446	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=11/2816, ttl=63 (reply in 47)
47	12.841143284	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=11/2816, ttl=64 (request in 46)
49	13.841092440	192.168.18.85	192.168.1.128	ICMP	98	Echo (ping) request id=0x001b, seq=12/3072, ttl=63 (reply in 50)
50	13.841133800	192.168.1.128	192.168.18.85	ICMP	98	Echo (ping) reply id=0x001b, seq=12/3072, ttl=64 (request in 49)

- **Result:** The ping was successful, demonstrating communication between the Ubuntu and Kali machines through the pfSense firewall. This was further verified by observing ICMP echo requests and replies in Wireshark on the Kali Linux machine.

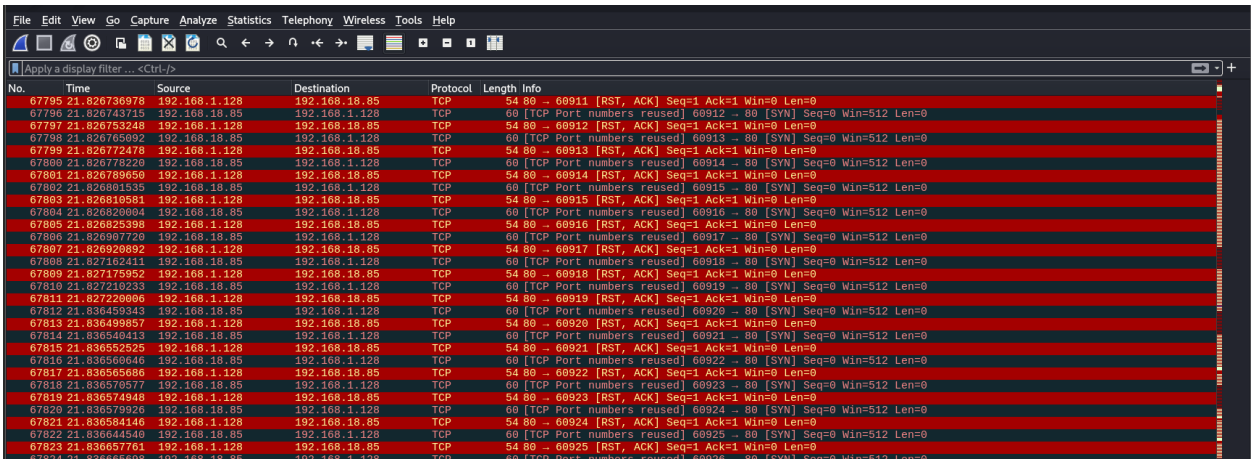
3. TCP SYN Flood Attack (Without Block Rules)

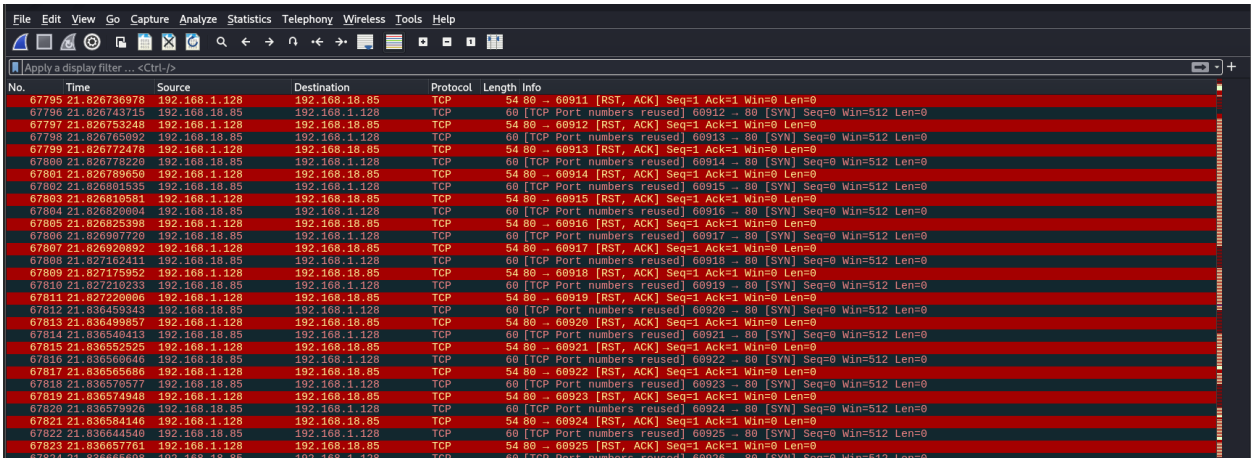
A TCP SYN flood attack was initiated from the Ubuntu machine targeting the Kali machine.

- **Attack Command (from Ubuntu):** sudo hping3 -S -p 80 --flood 192.168.1.128

```
ali@ali-VMware-Virtual-Platform:~$ sudo hping3 -S -p 80 --flood 192.168.1.128
[sudo] password for ali:
HPING 192.168.1.128 (ens33 192.168.1.128): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.128 hping statistic ---
211973 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

- **hping3 Statistics:** The hping3 tool reported a 100% packet loss for transmitted packets, as it was operating in flood mode and not expecting replies.
- **Wireshark Analysis (on Kali):** Wireshark on the Kali machine showed a high volume of TCP packets, indicating the SYN flood attack. These packets were predominantly SYN requests, as expected during a SYN flood.

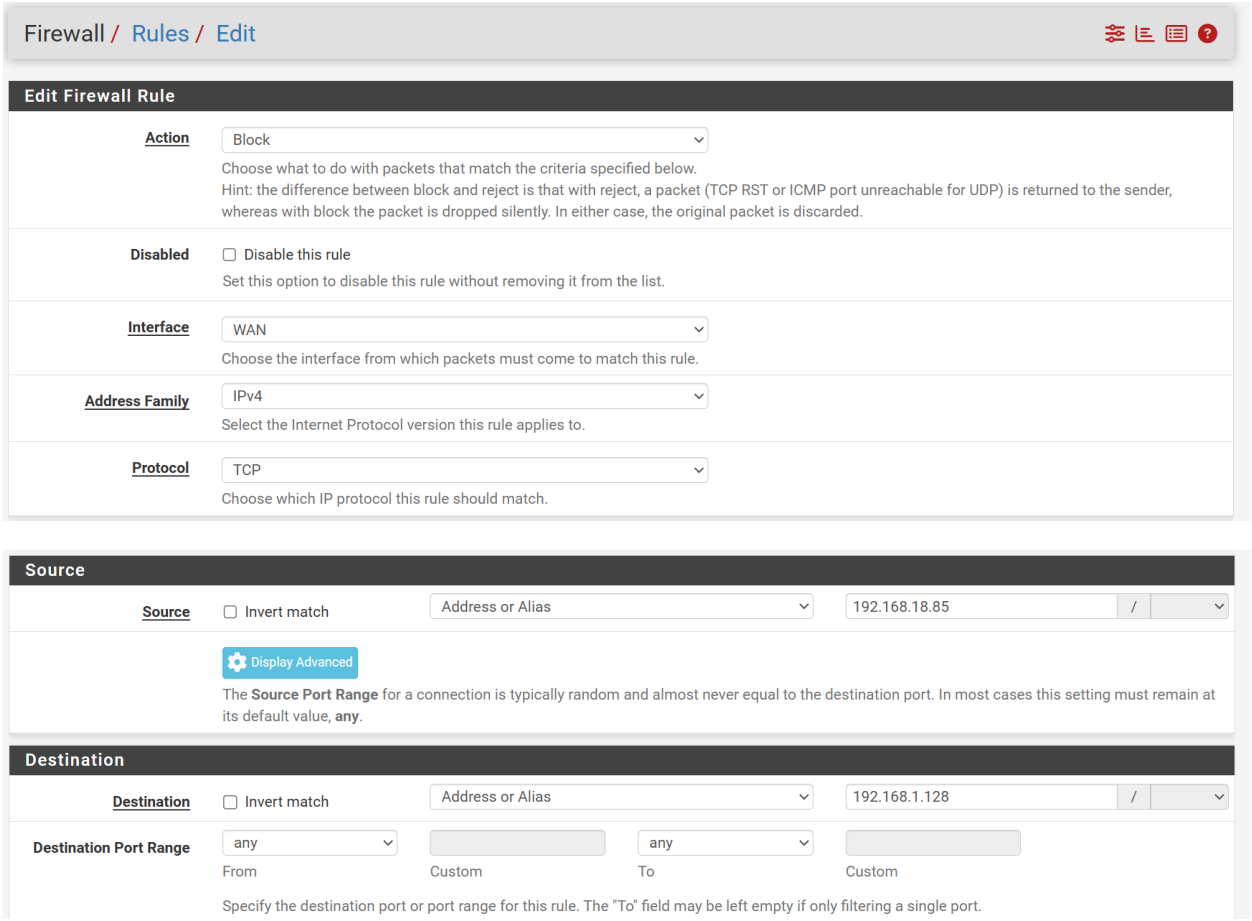




4. Implementing a Block Rule on pfSense

To mitigate the SYN flood attack, a block rule was created in pfSense.

- **Action:** Block (This drops packets silently, unlike "reject" which sends a response back to the sender).
- **Interface:** WAN (Packets matching this rule must originate from the WAN interface).
- **Address Family:** IPv4
- **Protocol:** TCP
- **Source:** 192.168.18.85 (The IP address of the Ubuntu attacker machine).
- **Destination:** 192.168.1.128 (The IP address of the Kali victim machine).
- **Description:** block syn flood
- **Logging:** Enabled (To log packets handled by this rule for verification).



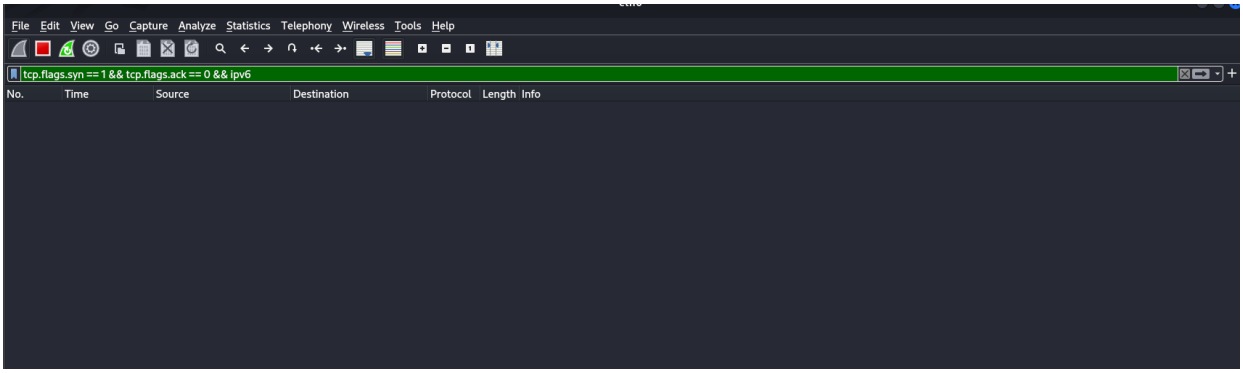
5. Post-Block Rule TCP SYN Flood Attack

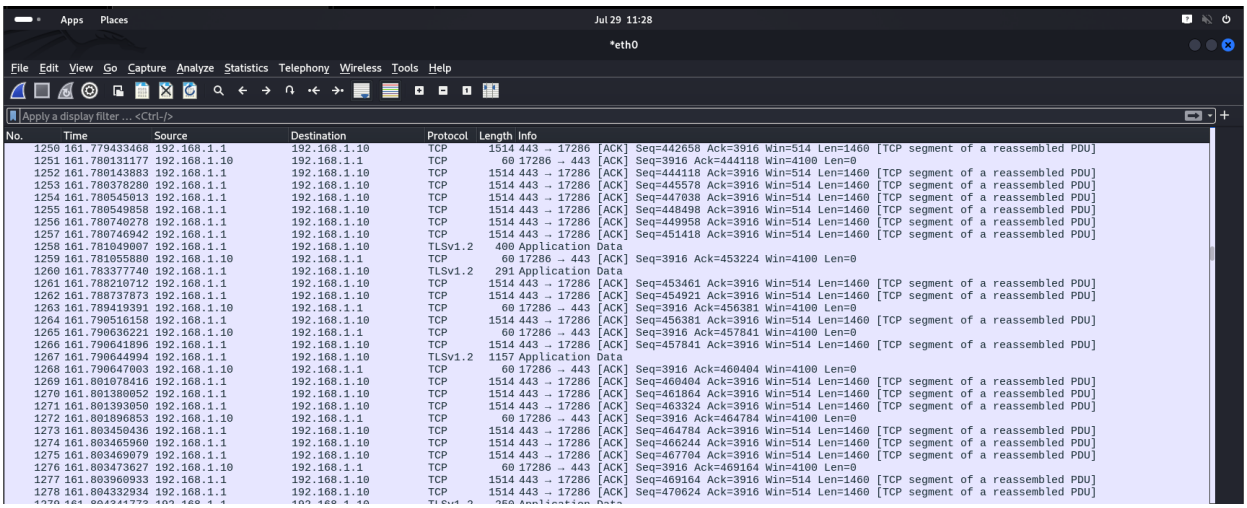
After saving the new block rule, the TCP SYN flood attack was re-initiated from the Ubuntu machine.

- **Attack Command (from Ubuntu):** sudo hping3 -S -p 80 --flood 192.168.1.128

```
ali@ali-VMware-Virtual-Platform:~$ sudo hping3 -S -p 80 --flood 192.168.1.128
HPING 192.168.1.128 (ens33 192.168.1.128): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.128 hping statistic ---
7633068 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

- **hping3 Statistics:** Similar to the previous attack, hping3 reported 100% packet loss.
- **Wireshark Analysis (on Kali):** Wireshark on the Kali machine displayed no incoming packets from the attacker, indicating that the SYN flood was successfully blocked by pfSense.





- **pfSense Logs:** The pfSense firewall logs confirmed that packets from the Ubuntu attacker (192.168.18.85) destined for the Kali machine (192.168.1.128) on TCP port 80 were actively being blocked by the "block syn flood" rule.

Action	Time	Interface	Rule	Source	Destination	Protocol
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61094	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61095	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61096	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61097	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61098	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61099	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61100	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61101	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61102	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61103	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61104	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61105	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61106	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61107	192.168.1.128:80	TCP:S
✕	Jul 29 21:23:50	WAN	block syn flood (1753805059)	192.168.18.85:61108	192.168.1.128:80	TCP:S

Conclusion:

The successful blocking of the TCP SYN flood attack, as verified by both Kali's Wireshark capture and pfSense's firewall logs, demonstrates the effectiveness of the configured pfSense rule in mitigating Denial of Service (DoS) attacks.