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

WRITTEN BY ADMINISTRATOR. POSTED IN NETWORK PROTOCOL ANALYZERS

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.

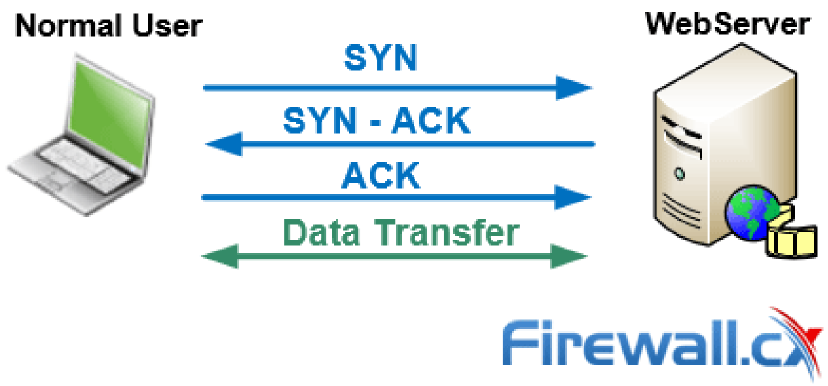
DoS attacks are simple to carry out, can cause serious downtime, and aren't always obvious. In a **SYN flood attack**, a malicious party exploits the **TCP protocol** **3-way handshake** to quickly cause **service and network disruptions**, ultimately leading to an **Denial of Service (DoS) Attack**. 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**.

Here's an overview of what's covered:

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

HOW TCP SYN FLOOD ATTACKS WORK

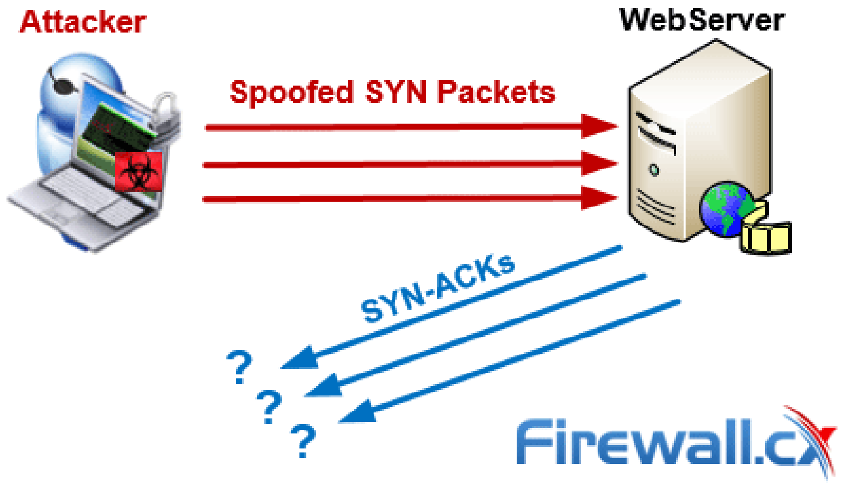
When a client attempts to connect to a server using the **TCP protocol** e.g (HTTP or HTTPS), it is first required to perform a **three-way handshake** before any data is exchanged between the two. Since the **three-way TCP handshake** is always initiated by the client it sends a **SYN packet** to the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



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

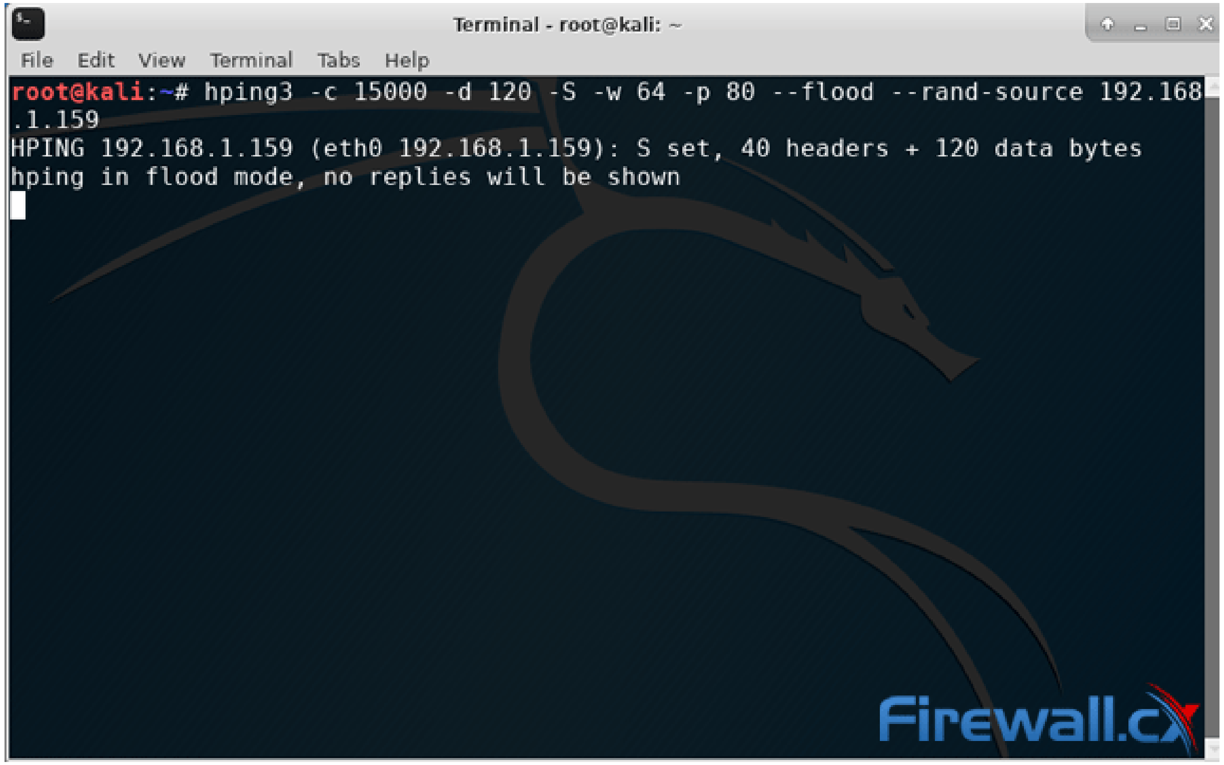
However, to test if you can **detect** this type of a **DoS attack**, you must be able to perform one. The simplest way is via a Kali Linux and more specifically the hping3, a popular **TCP penetration testing tool** included in Kali Linux.

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

```
# 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):

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



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 victum'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.

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

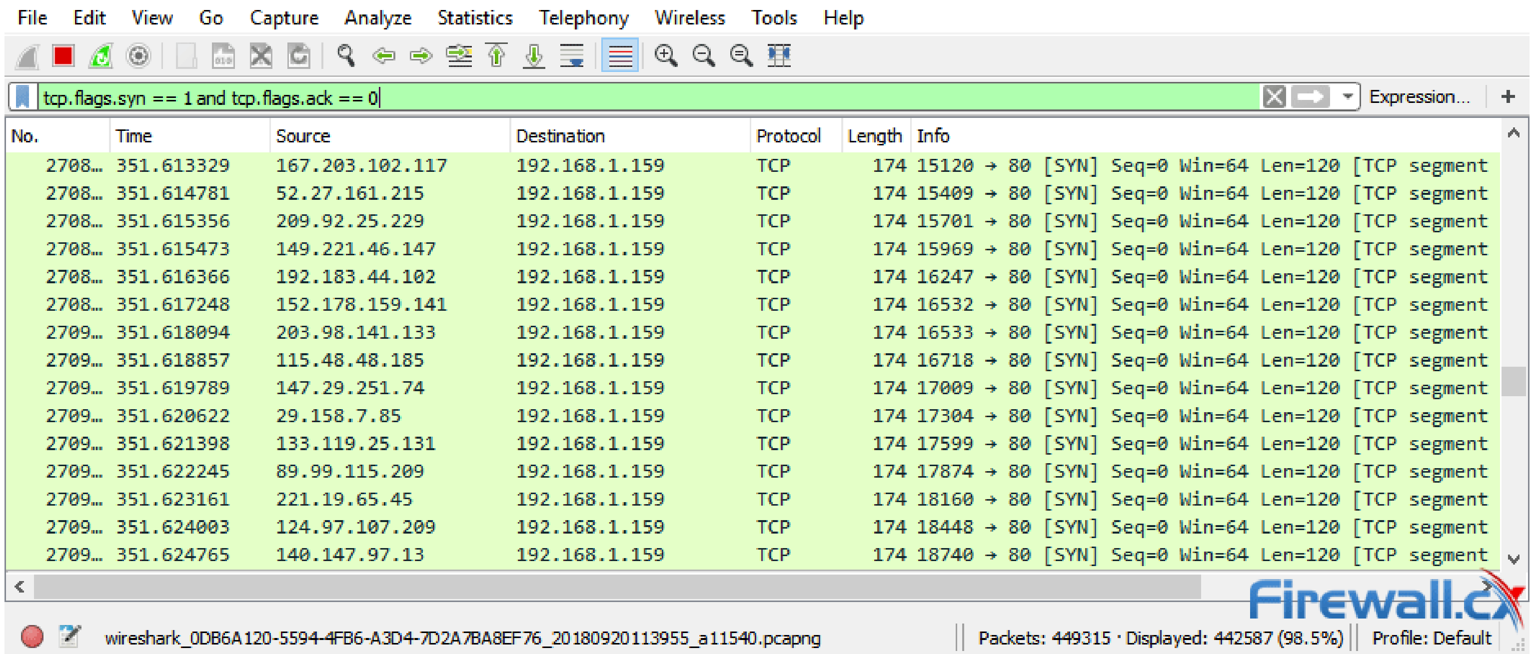
POPULAR LINUX ARTICLES

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**

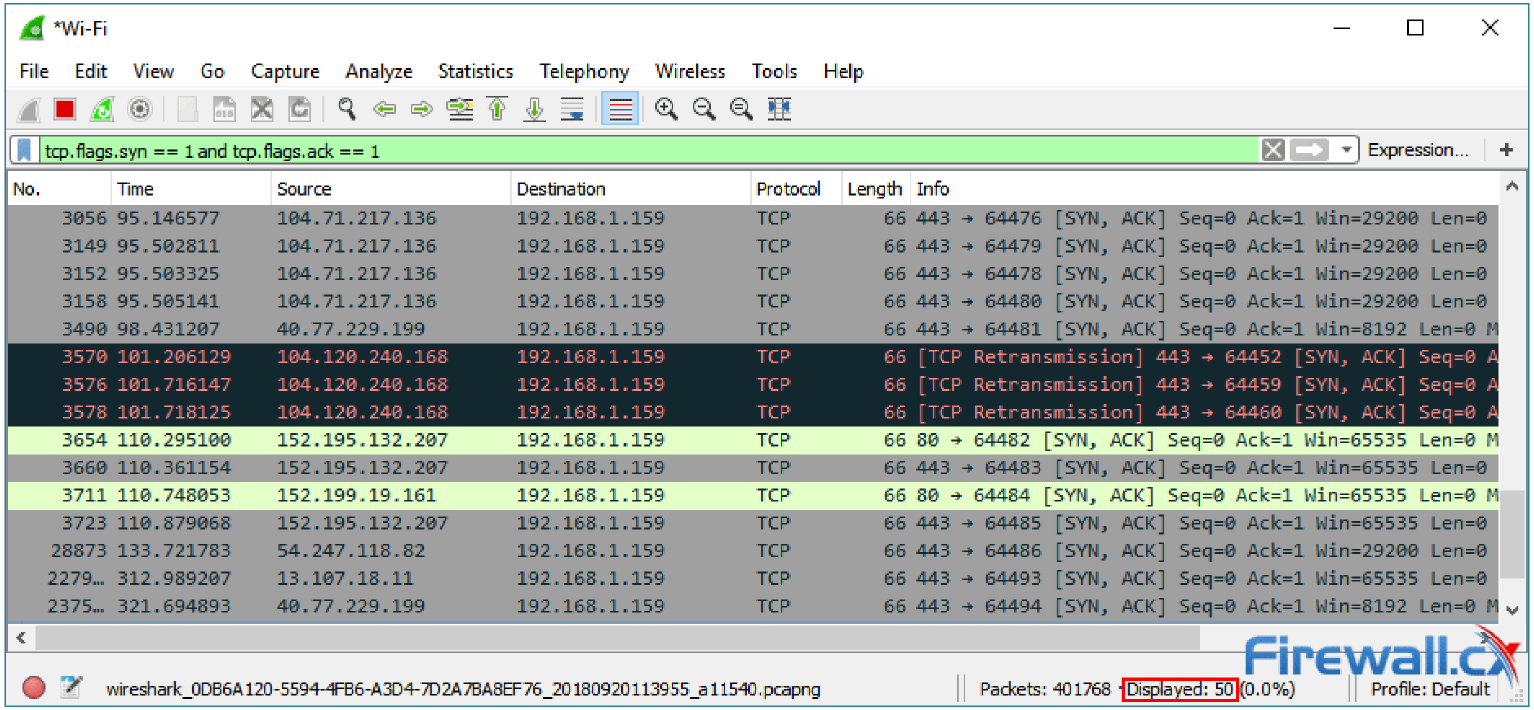


As you can see, there's a **high volume of SYN packets** with very little variance in time. **Each SYN packet** shows it's from a **different source IP address** with a **destination port 80** (HTTP), **identical length of 120** and **window size** (**64**). When we filter with **tcp.flags.syn == 1 and 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.

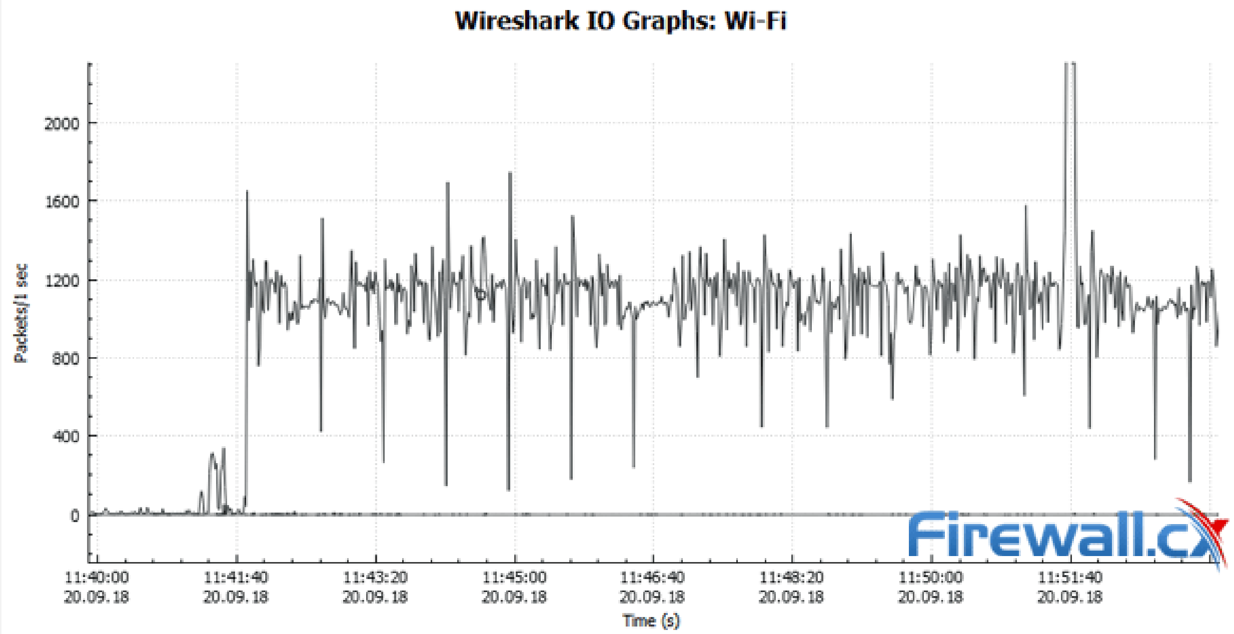
Linux Samba
Linux DHCP Server
Linux Bind DNS
Linux File & Folder Permissions
Linux OpenMosix
Linux Network Config

RSS SUBSCRIPTION

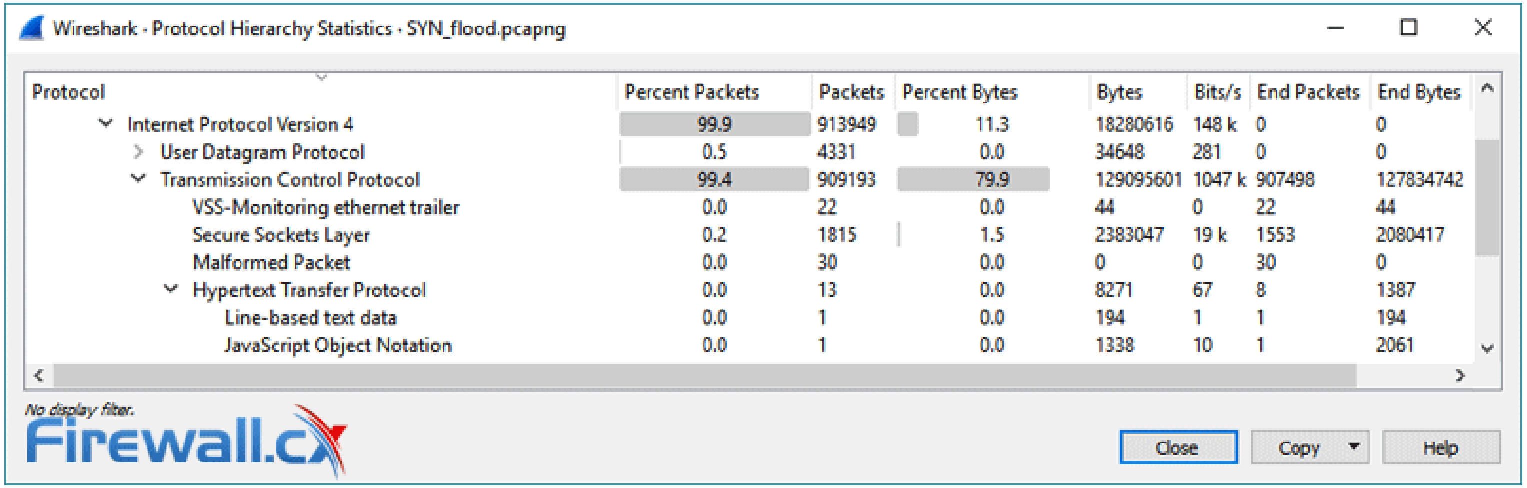
Subscribe to Firewall.cx RSS Feed by Email



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**.



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**:



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

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

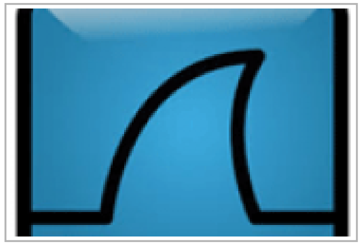
RELATED ARTICLES

- TCP Protocol Analysis
- How to Detect SYN Flood Attacks with Colasoft Capsa

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...