**EXPOSYS DATA LABS**

**Full Stack Development**

**DDoS Attack**

Submitted by :

ASHWIN P M

MADHU MANJU S

MUGILAN.G

S. KRISHNA RAM

M.RISHIKARAN

CONTACT : +919361190319 , +916382410056

**Abstract :**

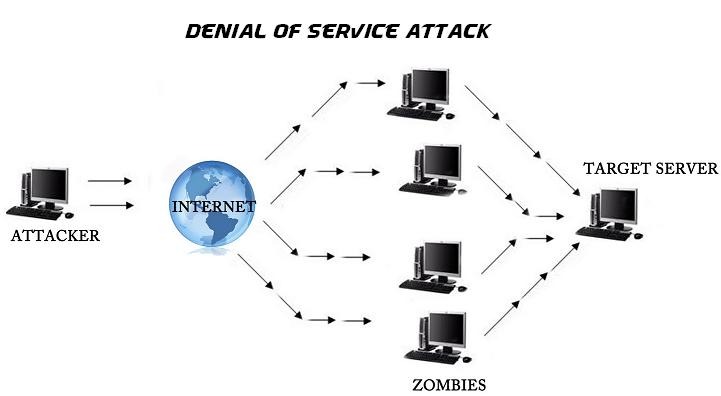
Distributed denial-of-service (DDoS) attack is one of the major threats to the web server.The rapid increase of DDoS attacks on the Internet has clearly pointed out the limitations in current intrusion detection systems or intrusion prevention systems (IDS/IPS), mostly caused by application-layer DDoS attacks. Within this context, the objective of the paper is to detect a DDoS attack using a multilayer perceptron (MLP) classification algorithm with genetic algorithm (GA) as learning algorithm. In this work, we analyzed the standard EPA-HTTP (environmental protection agency-hypertext transfer protocol) dataset and selected the parameters that will be used as input to the classifier model for differentiating the attack from normal profile. The parameters selected are the HTTP GET request count, entropy, and variance for every connection. The proposed model can provide a better accuracy of 98.31%, sensitivity of 0.9962, and specificity of 0.0561 when compared toother traditional classification models.

**INTRODUCTION :**

Different from the traditional architecture of network devices where data and control plane are joined in the same device, a new networking architecture called SDN appeared in the last ten years and separated the control plane from the data plane. In SDN, network devices like switches and routers reside in the data plane and they become simple packet forwarding devices, and network intelligence in the form of software control program, named the controller, reside in the control plane.

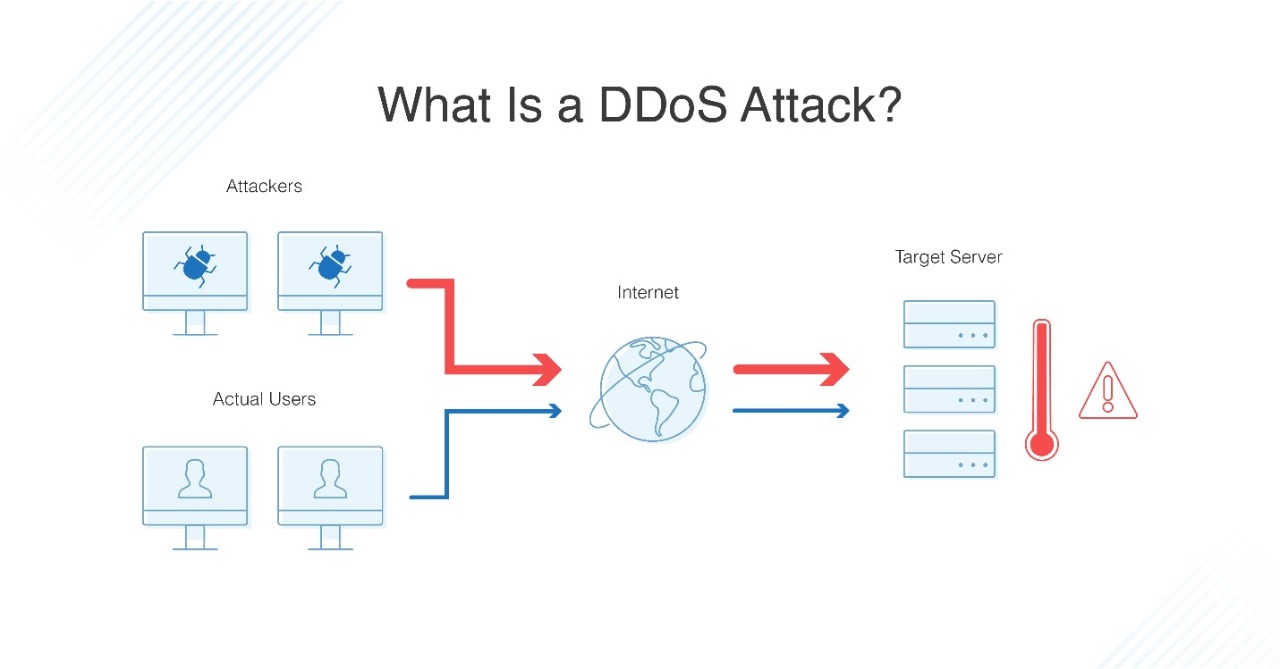
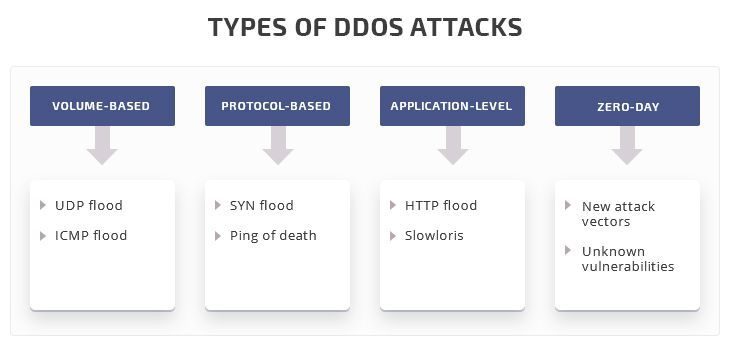
In SDN, DDoS attack is a big security challenge, it mainly occurs due to the flooding of traffic packets from the attacker to the victim to either decrease its performance or stop its service and it will be unavailable for future connection; usually, the source addresses of these incoming packets are spoofed, in this case, the switch is not going to find a match in the flow table, and therefore, it will send the packet to the controller; the resources of the controller will be chained into continuous processing by the combination of DDoS spoofed and legitimate packets until they are completely exhausted; this can bring down the controller causing the loss of the SDN architecture and the newly arrived legitimate packets will not be able to reach the controller.

**DoS Attack**

A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) of the service or resource they expected.

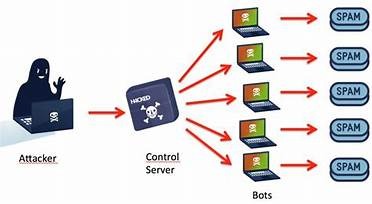
**DDoS Attack**

A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic. DDoS attacks achieve effectiveness by utilizing multiple compromised computer systems as sources of attack traffic. Exploited machines can include computers and other networked resources such as IoT devices. From a high level, a DDoS attack is like an unexpected traffic jam clogging up the highway, preventing regular traffic from arriving at its destination.

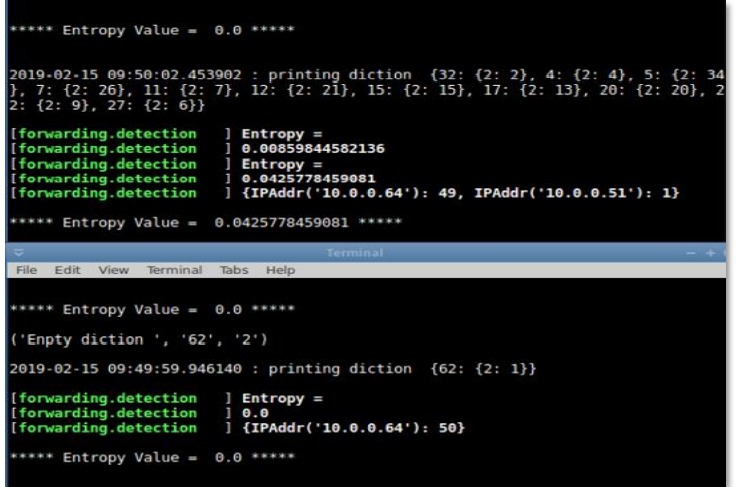


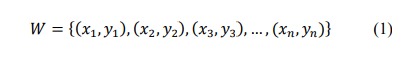
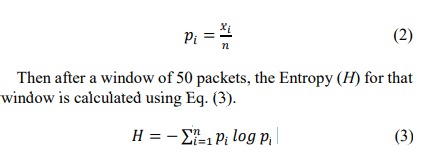
**BOTNET**

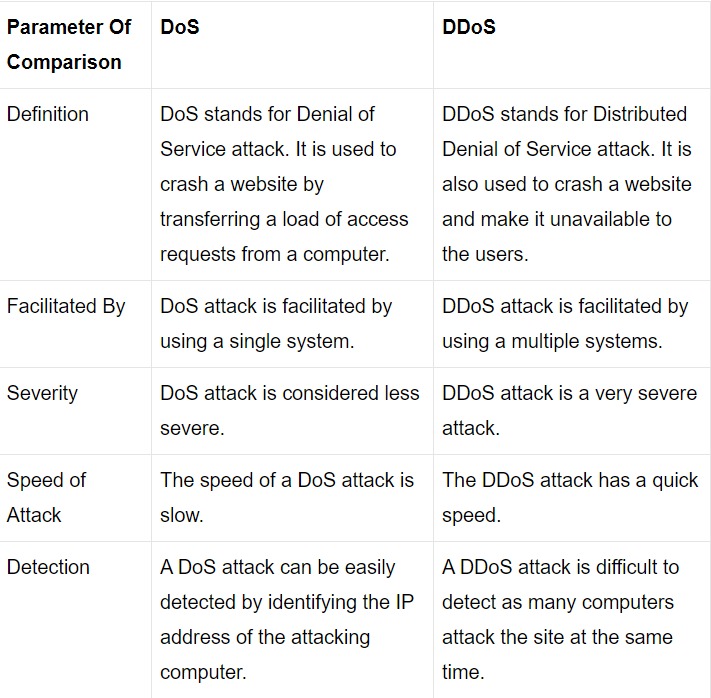
A computer or network device under the control of an intruder is known as a zombie, or bot. The attacker creates what is called a command-and-control server to command the network of bots, also called a botnet. The person in control of a botnet is referred to as the botmaster. That term has also been used to refer to the first system recruited into a botnet because it is used to control the spread and activity of other systems in the botnet.



**ENTROPY COMPUTING**

Entropy-based algorithms can be used for the detection of attacks in communication networks. This statistical approach can detect several types of attacks including DDoS, by identifying the randomness of the incoming traffic. The higher the Entropy is, the higher the randomness of the traffic flow; on the contrary, the lower the Entropy, the higher the determinacy of the traffic flow [7]. This paper uses Entropy-based detection algorithm for DDoS attack proposed by [17] and apply it on different test cases. Entropy in this detection algorithm uses two components to calculate the randomness of the incoming packets; the first component is the window size, which represent how many incoming new packets are used in measuring Entropy and it is set to 50; and the second component is the threshold which is set to 1, and it is compared with the Entropy value to decide whether it is an attack or not. For every new Packet\_In message that arrive in the network its header is parsed for the destination IP address; and a hash table of this address and its number of occurrences is created. The hash table can be represented by Eq. (1), where W represents a window with n elements (n equals 50), x represents the destination IP address, and y represents the number of times it appeared. For each destination IP address, the probability is measured using Eq.

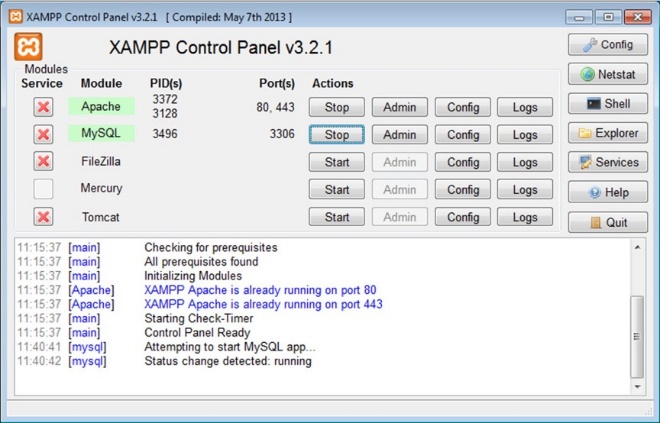
𝑊 = {(𝑥1, 𝑦1), (𝑥2, 𝑦2), (𝑥3, 𝑦3), … , (𝑥𝑛, 𝑦𝑛)} 30𝑝𝑖 =𝑥 𝑖𝑛 Then after a window of 50 packets, the Entropy (H) for that window is calculated using Eq. (3). 𝐻=−∑𝑝𝑖 𝑙𝑜𝑔 𝑝𝑖𝑛 𝑖=1



**SOFTWARE AND TOOLS REQUIRED** :

* XAMPP
* PYTHON
* SLOWLORIS
* METASPLOIT
* NMAP
* WIRESHARK

**XAMPP**

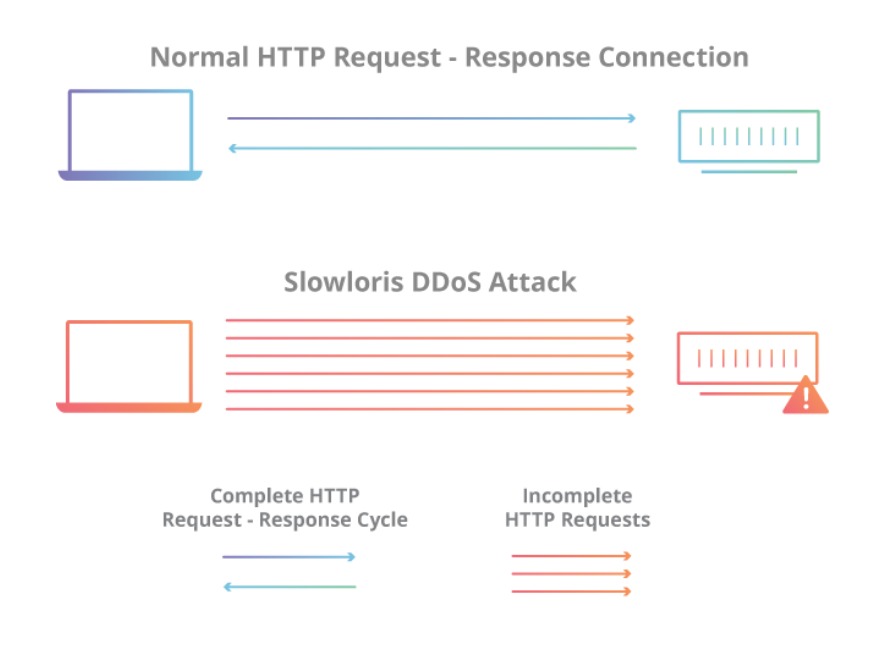
XAMPP is the title used for a compilation of free software. The name is an acronym, with each letter representing one of the five key components. The software packet contains the web server Apache, the relational database management system MySQL (or MariaDB), and the scripting languages Perl and PHP. The initial X stands for the operating systems that it works with: Linux, Windows, and Mac OS X

**PYCHARM**

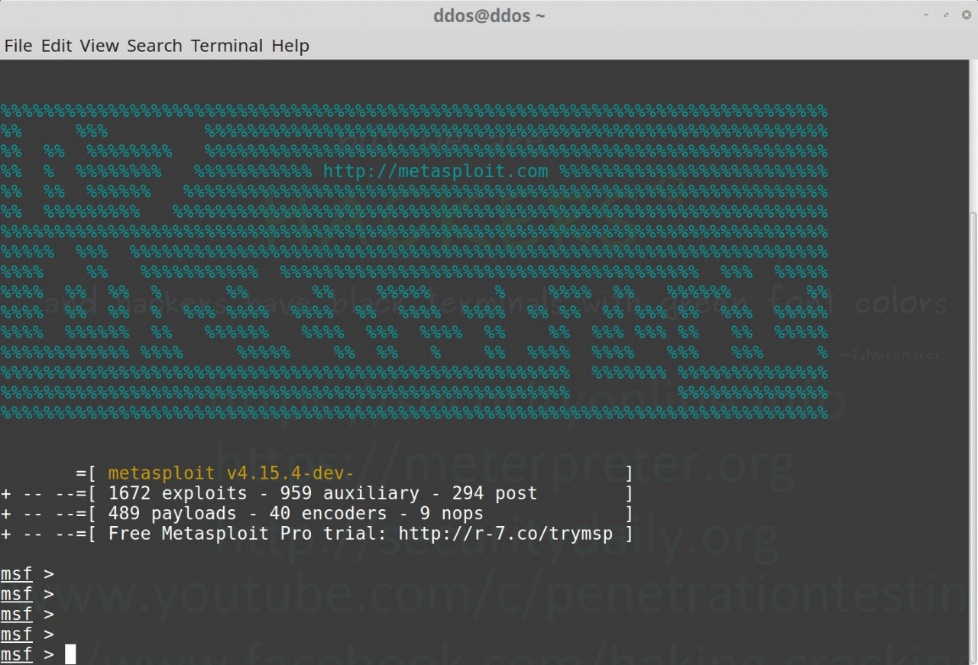
PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.



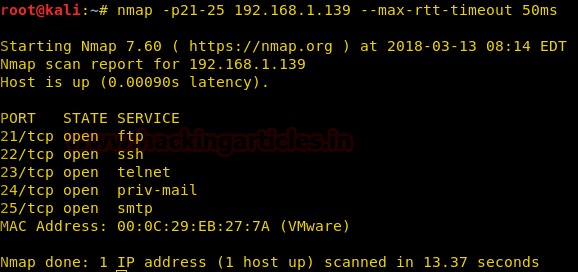
**SLOWLORIS**

Slowloris is a denial-of-service attack program which allows an attacker to overwhelm a targeted server by opening and maintaining many simultaneous HTTP connections between the attacker and the target.

**METASPLOIT**

Metasploit is a penetration testing platform that allows you to find, exploit, and validate vulnerabilities. Also, it provides the infrastructure, content, and tools to conduct penetration tests and comprehensive security auditing. It is a type of DoS attack which use to send a huge amount of Sync to consume all the resources of the target system. Let’s start by launching Metasploit by simply typing msfconsole in your terminal Window. It will take a couple of minutes to launch the console. Then use the select the auxiliary “auxiliary/dos/tcp/synflood” by typing the following command. msf > use auxiliary/dos/tcp/synflood

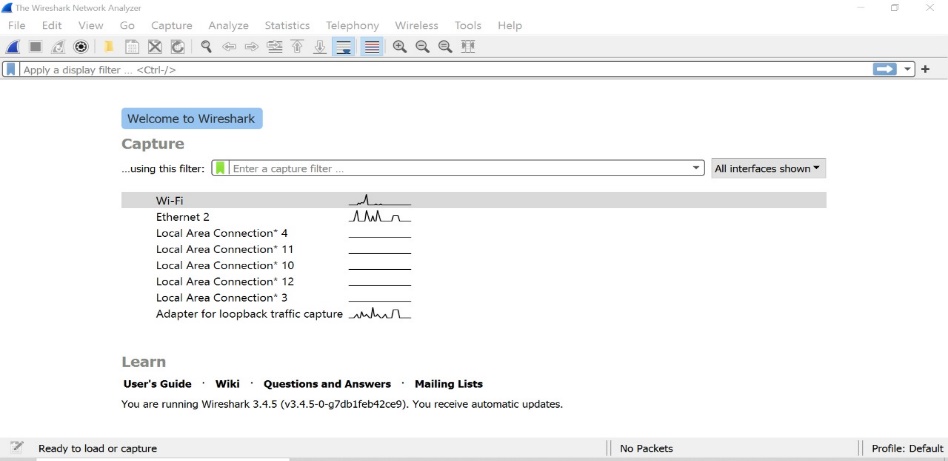
**NMAP**

We can use the db\_nmap command to run Nmap against our targets and our scan results would than be stored automatically in our database. However, if you also wish to import the scan results into another application or framework later on, you will likely want to export the scan results in XML format. It is always nice to have all three Nmap outputs (xml, grepable, and normal). So we can run the Nmap scan using the -oA flag followed by the desired filename to generate the three output files, then issue the db\_import command to populate the Metasploit database.

Run Nmap with the options you would normally use from the command line. If we wished for our scan to be saved to our database, we would omit the output flag and use db\_nmap. The example below would then be db\_nmap -v -sV 192.168.1.0/24.

**WIRESHARK**

Wireshark is an open-source network monitoring tool. We can use Wireshark to capture the packet from the network and also analyze the already saved capture.

This feature is very useful when dealing with what protocols are running on the server. To find this, click on Summary | Protocol Hierarchy in the Wireshark menu. A protocol hierarchy of the captured packets will open its gives a high-level glance at all protocals that are happening over the ethernet system

**Attacking Methodology**

**Attack using Slowloris:**

Slowloris is an application layer attack which operates by utilizing partial HTTP requests. The attack functions by opening connections to a targeted Web server and then keeping those connections open as long as it can.

Slowloris is not a category of attack but is instead a specific attack tool designed to allow a single machine to take down a server without using a lot of bandwidth. Unlike bandwidth-consuming reflection-based DDoS attacks such as NTP amplification, this type of attack uses a low amount of bandwidth, and instead aims to use up server resources with requests that seem slower than normal but otherwise mimic regular traffic. It falls in the category of attacks known as “low and slow” attacks. The targeted server will only have so many threads available to handle concurrent connections. Each server thread will attempt to stay alive while waiting for the slow request to complete, which never occurs. When the server’s maximum possible connections has been exceeded, each additional connection will not be answered and denial-of-service will occur.

A Slowloris attack occurs in 5 steps

1.The attacker first opens multiple connections to the targeted server by sending multiple partial HTTP request headers.

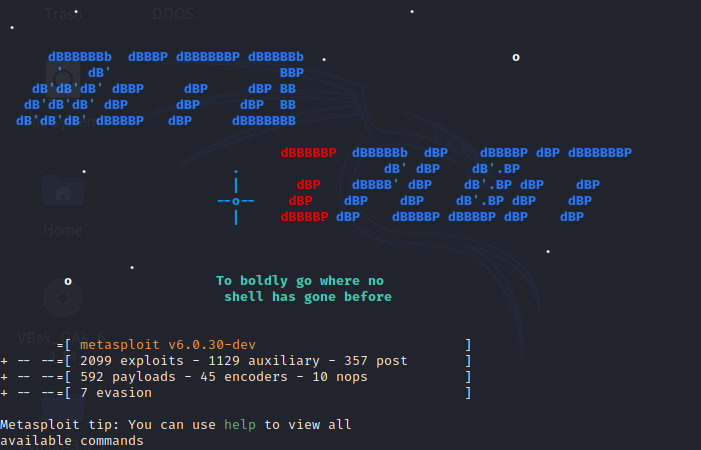
2.The target opens a thread for each incoming request, with the intent of closing the thread once the connection is completed. In order to be efficient, if a connection takes too long, the server will timeout the exceedingly long connection, freeing the thread up for the next request.

3.To prevent the target from timing out the connections, the attacker periodically sends partial request headers to the target in order to keep the request alive. In essence saying, “I’m still here! I’m just slow, please wait for me.”

4.The targeted server is never able to release any of the open partial connections while waiting for the termination of the request. Once all available threads are in use, the server will be unable to respond to additional requests made from regular traffic, resulting in denial-of-service.

5.The key behind a Slowloris is its ability to cause a lot of trouble with very little bandwidth consumption.

**Preparing Metasploit for Port Scanning:**

Scanners and most other auxiliary modules use the ‘RHOSTS’ option instead of ‘RHOST’. RHOSTS can take IP ranges (192.168.1.20-192.168.1.30), CIDR ranges (192.168.1.0/24), multiple ranges separated by commas (192.168.1.0/24, 192.168.3.0/24), and line-separated host list files (file:/tmp/hostlist.txt). This is another use for a grepable Nmap output file. By default, all of the scanner modules will have the ‘THREADS’ value set to ‘1’. The ‘THREADS’ value sets the number of concurrent threads to use while scanning. Set this value to a higher number in order to speed up your scans or keep it lower in order to reduce network traffic but be sure to adhere to the following guidelines:

Keep the THREADS value under 16 on native Win32 systems

Keep THREADS under 200 when running MSF under Cygwin

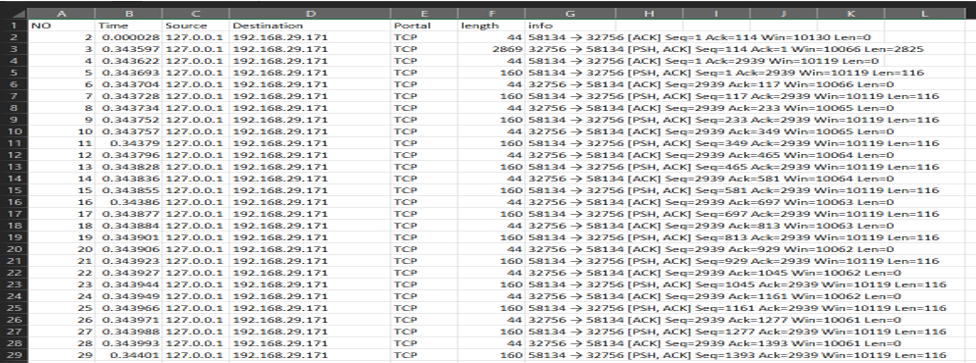
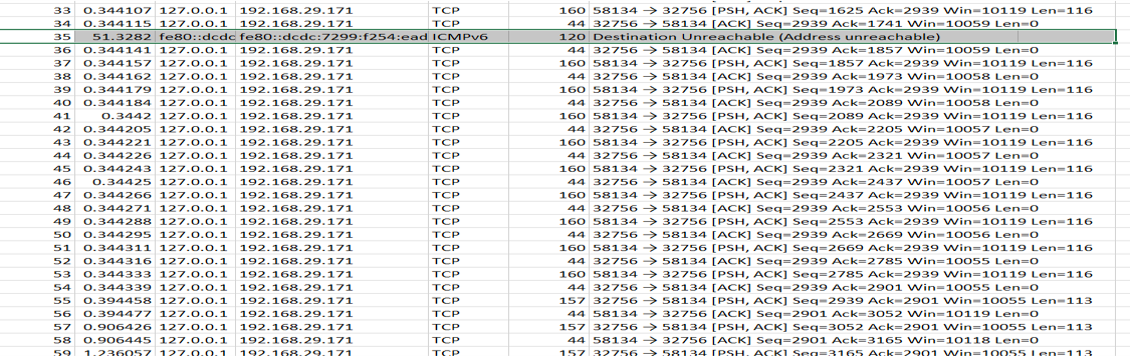
On Unix-like operating systems, THREADS can be set as high as 256

**Nmap & db\_nmap:**

We can use the db\_nmap command to run Nmap against our targets and our scan results would than be stored automatically in our database. However, if you also wish to import the scan results into another application or framework later on, you will likely want to export the scan results in XML format. It is always nice to have all three Nmap outputs (xml, grepable, and normal). So we can run the Nmap scan using the -oA flag followed by the desired filename to generate the three output files, then issue the db\_import command to populate the Metasploit database.

Run Nmap with the options you would normally use from the command line. If we wished for our scan to be saved to our database, we would omit the output flag and use db\_nmap. The example below would then be db\_nmap -v -sV 192.168.1.0/24

**OUTPUT:**



Reference:

1)- <https://youtu.be/7bHQtpcfZ8U>

2.)- <https://doi.org/10.18280/rces.060201>

3.)- S. Azodolmolky, Software Defined Networking with OpenFlow,