

CSCE 5580: Computer Network

Bonus Project Report

Title: Implementation and Analysis of HTTP Flood Attack

Author: Kishan Kumar Zalavadia

1. Introduction

This project implements an HTTP Flood attack. This project is implemented in Python programming language, which demonstrates the behaviors and impacts on the target server that we created. The objective is to learn and understand the HTTP attack and gain insights into the methodologies.

2. Methodology

In this project, we have created our own server, which is running on a 127.0.0.1 IP address with 65535 as the port number to ensure that the attack is not impacting the real servers.

2.1 HTTP Flood Attack

In this attack, a client sends a large number of HTTP GET requests with random paths, headers, and numbers. It implements multiple threads to simulate concurrent clients. Each thread generates HTTP GET requests with a 5-character URL path, header, and random port number. This project targets the server with IP address 127.0.0.1 and port number 65535. This server is created by us, so in order to execute the HTTP attack, we first need to run the server and then perform the attack.

We create a number of clients, and each client requests five http requests to the client. So, for example, if we have 100 clients, there will be 500 requests made to the client.

3 Results

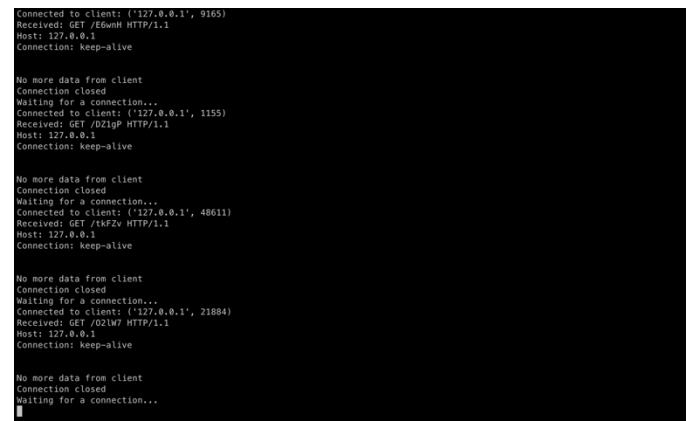
3.1 Experimental Setup

Server: A basic Python socket server that we created to accept connections and log activities.

Client: We use multi-threading to execute clients concurrently.

Wireshark: We use Wireshark to monitor the network traffic during the attacks.

3.2 Console Output



```
Connected to client: ('127.0.0.1', 9165)
Received: GET /6bmk HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive

No more data from client
Connection closed
Waiting for a connection...
Connected to client: ('127.0.0.1', 1155)
Received: GET /DZigP HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive

No more data from client
Connection closed
Waiting for a connection...
Connected to client: ('127.0.0.1', 48611)
Received: GET /thFZv HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive

No more data from client
Connection closed
Waiting for a connection...
Connected to client: ('127.0.0.1', 21884)
Received: GET /02W HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive

No more data from client
Connection closed
Waiting for a connection...
```

Figure 3.2.1

Figure 3.2.1 is a console from the server. To perform the HTTP attack, we run our server on 127.0.0.1:65535. Once the server is running, it waits for the client to establish a connection. Figure 3.2.1 represents the server console after the attack. It shows that the connection to the client has started, and it received a get HTTP request having a random character path, and the connection is kept alive.

```

Connected by '127.0.0.1', 19264 Content is b 'b'GET /vbmw HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 20929 Content is b 'b'GET /d20b HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 63742 Content is b 'b'GET /b1u1 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 45799 Content is b 'b'GET /SE6j HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 24680 Content is b 'b'GET /U39Z HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 2221 Content is b 'b'GET /A4k1 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 18822 Content is b 'b'GET /Lm1u HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 15471 Content is b 'b'GET /7u6d HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 52580 Content is b 'b'GET /5Lsk HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 7913 Content is b 'b'GET /c6d0 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 24853 Content is b 'b'GET /2u1U HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 38236 Content is b 'b'GET /vM4u HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 53243 Content is b 'b'GET /4u12 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 54279 Content is b 'b'GET /9t1u HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 59211 Content is b 'b'GET /8d0F HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 45911 Content is b 'b'GET /8dmc HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 38391 Content is b 'b'GET /1EAU HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 2323 Content is b 'b'GET /23u9 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 13539 Content is b 'b'GET /72A1G HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 33123 Content is b 'b'GET /2X3M HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 53243 Content is b 'b'GET /4u12 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 5783 Content is b 'b'GET /9X0d HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 35840 Content is b 'b'GET /52DQ HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 55165 Content is b 'b'GET /8dmc HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 39453 Content is b 'b'GET /o1b1 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 32148 Content is b 'b'GET /71d0 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 24329 Content is b 'b'GET /A31e HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 54083 Content is b 'b'GET /0V36 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 13867 Content is b 'b'GET /8310 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 35840 Content is b 'b'GET /8dmc HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 59818 Content is b 'b'GET /755L HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 23137 Content is b 'b'GET /XK34 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 59380 Content is b 'b'GET /7223 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 36228 Content is b 'b'GET /7J2D HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 15731 Content is b 'b'GET /6M6E HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 27393 Content is b 'b'GET /71V7 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 23893 Content is b 'b'GET /8dmc HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 59380 Content is b 'b'GET /7223 HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 1155 Content is b 'b'GET /JZ1P HTTP/1.1Connection: keep-alive\r\n\r\n'
Connected by '127.0.0.1', 48611 Content is b 'b'GET /7Kf2V HTTP/1.1Connection: keep-alive\r\n\r\n'

```

Figure 3.2.2

Once the server is up and running, we perform the http attack. Figure 3.2.2 represents the console on the client side after the attack. As you can see, the client is connected to our server, and it sends the HTTP request, which has a random character path.

3.3 Wireshark Output

No.	Time	Source	Destination	Length	Info
8748	0.000000	192.168.1.101	127.0.0.1	60	GET /vbmw HTTP/1.1
8749	0.000000	192.168.1.101	127.0.0.1	60	GET /d20b HTTP/1.1
8750	0.000000	192.168.1.101	127.0.0.1	60	GET /b1u1 HTTP/1.1
8751	0.000000	192.168.1.101	127.0.0.1	60	GET /SE6j HTTP/1.1
8752	0.000000	192.168.1.101	127.0.0.1	60	GET /U39Z HTTP/1.1
8753	0.000000	192.168.1.101	127.0.0.1	60	GET /A4k1 HTTP/1.1
8754	0.000000	192.168.1.101	127.0.0.1	60	GET /Lm1u HTTP/1.1
8755	0.000000	192.168.1.101	127.0.0.1	60	GET /7u6d HTTP/1.1
8756	0.000000	192.168.1.101	127.0.0.1	60	GET /5Lsk HTTP/1.1
8757	0.000000	192.168.1.101	127.0.0.1	60	GET /c6d0 HTTP/1.1
8758	0.000000	192.168.1.101	127.0.0.1	60	GET /2u1U HTTP/1.1
8759	0.000000	192.168.1.101	127.0.0.1	60	GET /vM4u HTTP/1.1
8760	0.000000	192.168.1.101	127.0.0.1	60	GET /4u12 HTTP/1.1
8761	0.000000	192.168.1.101	127.0.0.1	60	GET /9t1u HTTP/1.1
8762	0.000000	192.168.1.101	127.0.0.1	60	GET /8d0F HTTP/1.1
8763	0.000000	192.168.1.101	127.0.0.1	60	GET /8dmc HTTP/1.1
8764	0.000000	192.168.1.101	127.0.0.1	60	GET /1EAU HTTP/1.1
8765	0.000000	192.168.1.101	127.0.0.1	60	GET /23u9 HTTP/1.1
8766	0.000000	192.168.1.101	127.0.0.1	60	GET /72A1G HTTP/1.1
8767	0.000000	192.168.1.101	127.0.0.1	60	GET /2X3M HTTP/1.1
8768	0.000000	192.168.1.101	127.0.0.1	60	GET /4u12 HTTP/1.1
8769	0.000000	192.168.1.101	127.0.0.1	60	GET /9X0d HTTP/1.1
8770	0.000000	192.168.1.101	127.0.0.1	60	GET /52DQ HTTP/1.1
8771	0.000000	192.168.1.101	127.0.0.1	60	GET /8dmc HTTP/1.1
8772	0.000000	192.168.1.101	127.0.0.1	60	GET /o1b1 HTTP/1.1
8773	0.000000	192.168.1.101	127.0.0.1	60	GET /71d0 HTTP/1.1
8774	0.000000	192.168.1.101	127.0.0.1	60	GET /A31e HTTP/1.1
8775	0.000000	192.168.1.101	127.0.0.1	60	GET /0V36 HTTP/1.1
8776	0.000000	192.168.1.101	127.0.0.1	60	GET /8310 HTTP/1.1
8777	0.000000	192.168.1.101	127.0.0.1	60	GET /8dmc HTTP/1.1
8778	0.000000	192.168.1.101	127.0.0.1	60	GET /755L HTTP/1.1
8779	0.000000	192.168.1.101	127.0.0.1	60	GET /XK34 HTTP/1.1
8780	0.000000	192.168.1.101	127.0.0.1	60	GET /7223 HTTP/1.1
8781	0.000000	192.168.1.101	127.0.0.1	60	GET /7J2D HTTP/1.1
8782	0.000000	192.168.1.101	127.0.0.1	60	GET /6M6E HTTP/1.1
8783	0.000000	192.168.1.101	127.0.0.1	60	GET /71V7 HTTP/1.1
8784	0.000000	192.168.1.101	127.0.0.1	60	GET /8dmc HTTP/1.1
8785	0.000000	192.168.1.101	127.0.0.1	60	GET /7223 HTTP/1.1
8786	0.000000	192.168.1.101	127.0.0.1	60	GET /JZ1P HTTP/1.1
8787	0.000000	192.168.1.101	127.0.0.1	60	GET /7Kf2V HTTP/1.1

Figure 3.3.1

No.	Time	Source	Destination	Length	Info
8748	0.000000	192.168.1.101	127.0.0.1	60	GET /vbmw HTTP/1.1
8749	0.000000	192.168.1.101	127.0.0.1	60	GET /d20b HTTP/1.1
8750	0.000000	192.168.1.101	127.0.0.1	60	GET /b1u1 HTTP/1.1
8751	0.000000	192.168.1.101	127.0.0.1	60	GET /SE6j HTTP/1.1
8752	0.000000	192.168.1.101	127.0.0.1	60	GET /U39Z HTTP/1.1
8753	0.000000	192.168.1.101	127.0.0.1	60	GET /A4k1 HTTP/1.1
8754	0.000000	192.168.1.101	127.0.0.1	60	GET /Lm1u HTTP/1.1
8755	0.000000	192.168.1.101	127.0.0.1	60	GET /7u6d HTTP/1.1
8756	0.000000	192.168.1.101	127.0.0.1	60	GET /5Lsk HTTP/1.1
8757	0.000000	192.168.1.101	127.0.0.1	60	GET /c6d0 HTTP/1.1
8758	0.000000	192.168.1.101	127.0.0.1	60	GET /2u1U HTTP/1.1
8759	0.000000	192.168.1.101	127.0.0.1	60	GET /vM4u HTTP/1.1
8760	0.000000	192.168.1.101	127.0.0.1	60	GET /4u12 HTTP/1.1
8761	0.000000	192.168.1.101	127.0.0.1	60	GET /9t1u HTTP/1.1
8762	0.000000	192.168.1.101	127.0.0.1	60	GET /8d0F HTTP/1.1
8763	0.000000	192.168.1.101	127.0.0.1	60	GET /8dmc HTTP/1.1
8764	0.000000	192.168.1.101	127.0.0.1	60	GET /1EAU HTTP/1.1
8765	0.000000	192.168.1.101	127.0.0.1	60	GET /23u9 HTTP/1.1
8766	0.000000	192.168.1.101	127.0.0.1	60	GET /72A1G HTTP/1.1
8767	0.000000	192.168.1.101	127.0.0.1	60	GET /2X3M HTTP/1.1
8768	0.000000	192.168.1.101	127.0.0.1	60	GET /4u12 HTTP/1.1
8769	0.000000	192.168.1.101	127.0.0.1	60	GET /9X0d HTTP/1.1
8770	0.000000	192.168.1.101	127.0.0.1	60	GET /52DQ HTTP/1.1
8771	0.000000	192.168.1.101	127.0.0.1	60	GET /8dmc HTTP/1.1
8772	0.000000	192.168.1.101	127.0.0.1	60	GET /o1b1 HTTP/1.1
8773	0.000000	192.168.1.101	127.0.0.1	60	GET /71d0 HTTP/1.1
8774	0.000000	192.168.1.101	127.0.0.1	60	GET /A31e HTTP/1.1
8775	0.000000	192.168.1.101	127.0.0.1	60	GET /0V36 HTTP/1.1
8776	0.000000	192.168.1.101	127.0.0.1	60	GET /8310 HTTP/1.1
8777	0.000000	192.168.1.101	127.0.0.1	60	GET /8dmc HTTP/1.1
8778	0.000000	192.168.1.101	127.0.0.1	60	GET /755L HTTP/1.1
8779	0.000000	192.168.1.101	127.0.0.1	60	GET /XK34 HTTP/1.1
8780	0.000000	192.168.1.101	127.0.0.1	60	GET /7223 HTTP/1.1
8781	0.000000	192.168.1.101	127.0.0.1	60	GET /7J2D HTTP/1.1
8782	0.000000	192.168.1.101	127.0.0.1	60	GET /6M6E HTTP/1.1
8783	0.000000	192.168.1.101	127.0.0.1	60	GET /71V7 HTTP/1.1
8784	0.000000	192.168.1.101	127.0.0.1	60	GET /8dmc HTTP/1.1
8785	0.000000	192.168.1.101	127.0.0.1	60	GET /7223 HTTP/1.1
8786	0.000000	192.168.1.101	127.0.0.1	60	GET /JZ1P HTTP/1.1
8787	0.000000	192.168.1.101	127.0.0.1	60	GET /7Kf2V HTTP/1.1

Figure 3.3.2

Figure 3.3.1 and figure 3.3.2 shows the Wireshark screenshot after the attack has been triggered.

As seen in the images, there are multiple HTTP requests, and each request goes from a different port number. (Highlighted in yellow)

4 Conclusion

This project gives a hands-on experience with network HTTP flood attack that impacts on server performance. The following are the insights gained.

Understanding of HTTP attack: How the server is impacted by the HTTP attack.

Traffic analysis: Looking at Wireshark, we gain insights on how to use Wireshark to monitor the network.

5 References

- <https://www.python.org/>
- <https://realpython.com/installing-python/>
- <https://scapy.net/>
- <https://www.imperva.com/learn/ddos/http-flood/>