**Deliverable #3: Report**

The assignment report must meet the guidelines and section criteria (see below) and be hosted on the web (see Deliverable #4).

**Guidelines:**

* **~~Title Page: Title, Assignment, Team Number, Team member names, Course Code.~~**
* Table of Contents
* Report Body: <10 pages in length

o Introduction

o Section 1:The Exploit – Summary

o Section 2: The Attack

o Section 3: Security Policy

o Conclusion

* Bibliography Page(s): this is a separate page(s) for your references and citations.

The Report Body and Bibliography:

·  ~~Each page should be:~~

~~o~~ **~~Single-spaced line,~~** ~~12 point font (Calibri) – this is non-header text~~

~~o~~ **~~2 cm margins, Header: Team Name & Course Code~~**

~~o~~ **~~Footer: Assignment No. & Page Number~~**

· Each section should start on a new page

Document a particular vulnerability or exploit of a host system. Describe the method used to attack the vulnerability/exploit, and explain the security policy process that would be used to deal with such an attack. You may need to set up a testing lab to demonstrate your exploit. It does not matter whether the exploit was successful; if your exploit failed, you need to document why (you may need to surmise what prevented the attack from working)

**Section 1: The Exploit – Summary**

This section provides a high-level overview/summary of the work you did. Be certain to include each stage of an attack: reconnaissance, scanning, exploiting the system, system remediation, and retesting, ***as appropriate to your topic***.

***Brief Description***

A short description of what the exploit does/how it works (a detailed explanation will be given in Part 2).

***Operating System (OS)***

* Windows 10 with any version

***Protocols/Services/Applications***

List the protocol, service, or application affected, including version numbers/patch levels, if appropriate.

***Variants***

The name(s) of different variants of the exploit, if any, include a brief description of the differences between any variants and the exploits you have selected.

***References***

**Exploit Host:**

* <https://fierce-tundra-99635.herokuapp.com/home>

**How to set-up a DoS attack:**

* <https://www.cloudflare.com/learning/ddos/ddos-attack-tools/how-to-ddos/>
* [**https://heimdalsecurity.com/blog/how-to-ddos/**](https://heimdalsecurity.com/blog/how-to-ddos/)
* [**https://www.imperva.com/learn/application-security/booters-stressers-ddosers/**](https://www.imperva.com/learn/application-security/booters-stressers-ddosers/)

**HTTP Flood attack:**

[**https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/**](https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/)[**https://www.hindawi.com/journals/jcnc/2019/1283472/**](https://www.hindawi.com/journals/jcnc/2019/1283472/)  
 [**https://www.netscout.com/what-is-ddos/http-flood-attacks**](https://www.netscout.com/what-is-ddos/http-flood-attacks)[**https://www.imperva.com/learn/application-security/http-flood/**](https://www.imperva.com/learn/application-security/http-flood/)

**Tool use for the Exploit:**

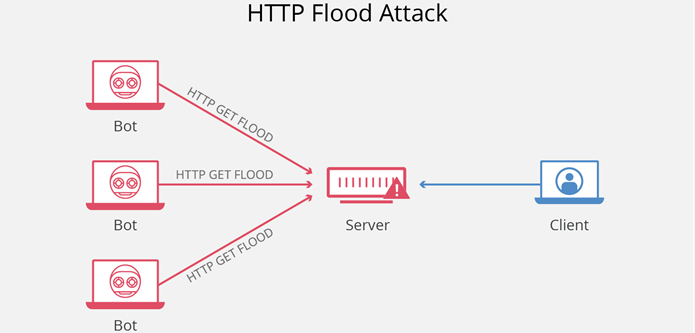
* [**https://www.redlegg.com/blog/3-tools-to-test-denial-of-service-vulnerability**](https://www.redlegg.com/blog/3-tools-to-test-denial-of-service-vulnerability)
* [**http://www.effecthacking.com/2017/11/hulk-web-server-dos-tool.html**](http://www.effecthacking.com/2017/11/hulk-web-server-dos-tool.html)

**Section 2: The Attack**

In section, the team will describe how the exploit was used to attack the host system or web application

· ***Description and diagram of network***

Include a description of the network (or the relevant parts of the network) where the incident occurred. Outline the network configuration that would provide maximum efficacy.



An HTTP flood attack is considered as a “layer 7” DDoS attack where the application layer of the OSI model is affected. To achieve maximum efficacy, attackers usually utilize or create botnets to drastically overwhelm a web server with HTTP requests. Additionally, this type of attack is most effective when it forces the server or application to allocate the most amount of resources to every single request. The attack tool we are using is HULK (HTTP Unbearable Load King is a web server DoS attack tool and is designed to generate large amounts of unique and obscure traffic to a web server (*HTTP Flood DDoS Attack*, n.d.).

· ***Protocol\Service Description***

In most cases, in order to understand the exploit, you need to understand how the protocol or service that is being exploited works and what its weaknesses are. Provide a brief description of the protocol, service, or application that the exploit uses.

Web servers that utilize Hypertext Transfer Protocol are easily exploited to denial-of-service attacks. HTTP is the basis for loading web pages and used to transfer data over the web. When a client initiates a HTTP GET request to a server a TCP connection must be established in order for a client to obtain a web server response. The first process in a HTTP GET request requires a web server to listen to an incoming connection. Next, a socket queue is assigned to serve the request. The socket queue is responsible for holding the HTTP GET request until a dedicated thread is assigned. Then, a request queue processes and responds to individual requests and then the web server sends a HTTP response. Web servers do not have a filtering mechanism to determine authentic HTTP GET requests and therefore during a flood attack, a web server will continuously receive and process these requests. As a result, request queues become flooded and drops incoming requests from authentic users (“What Is an HTTP Flood” n.d.).

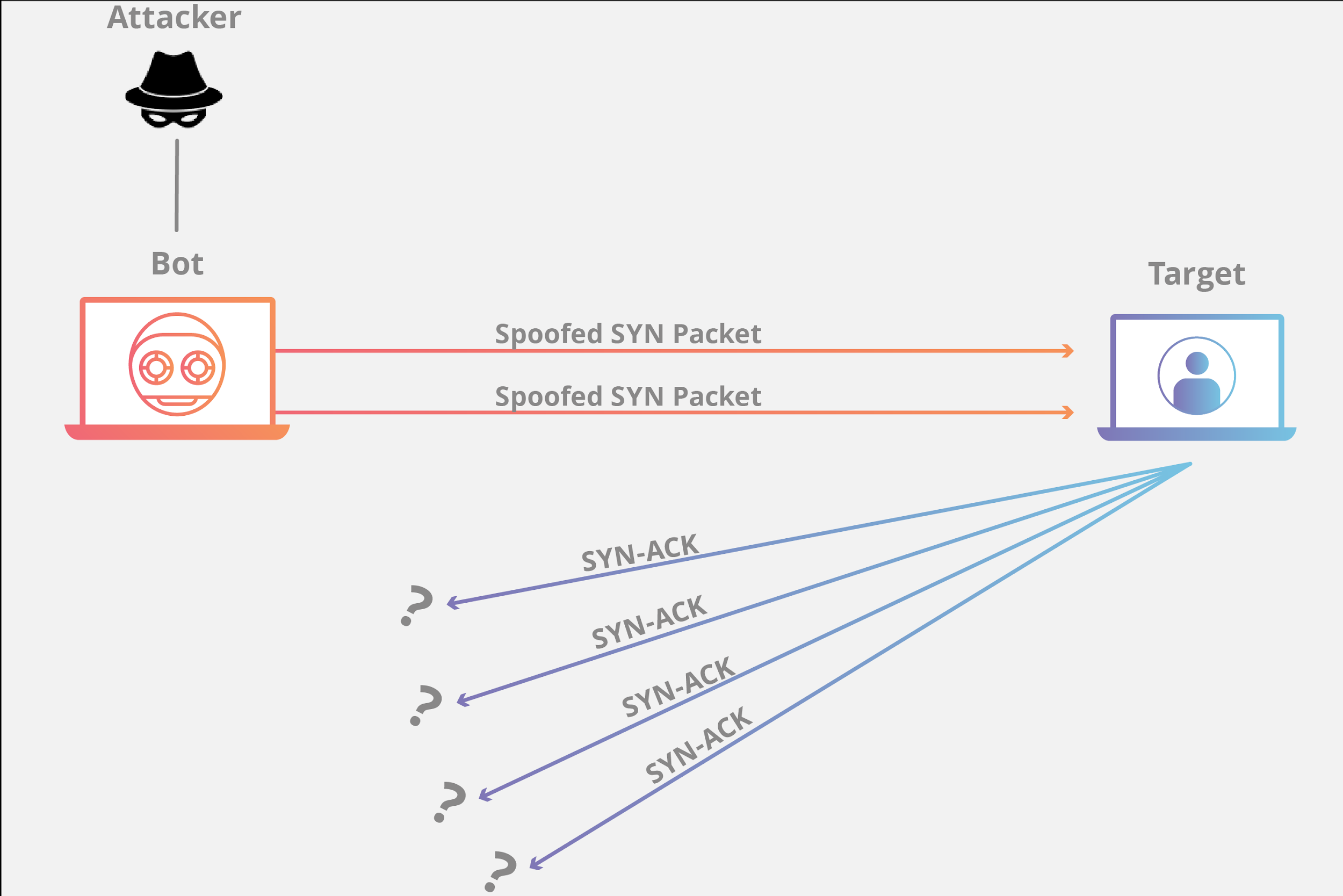
· ***How the exploit works***

Provide a detailed description of how the exploit works and why it is able to exploit the particular vulnerability in the protocol, service, or application. What is it about the protocol/service/etc. that is being exploited that makes it vulnerable? You must provide a step-by-step analysis of the actions the exploit takes in this section. You must include relevant sections of source code to illustrate and explain what the exploit is doing. If any programs exist to exploit this vulnerability, explain how to use the program(s). In addition to the program description, discuss how you could manually run the exploit against a system if an automated program did not exist.

· Also, outline each stage of an attack, including reconnaissance, scanning, exploiting the system (gaining access, elevating access, application-level access and/or denial of service). For each stage, provide details of the steps you performed, the commands that you typed, the tools that you used, and why each step was performed. Use as many screenshots as possible to document the process. Your documentation should be so clear that someone could take your report and reproduce the attack.

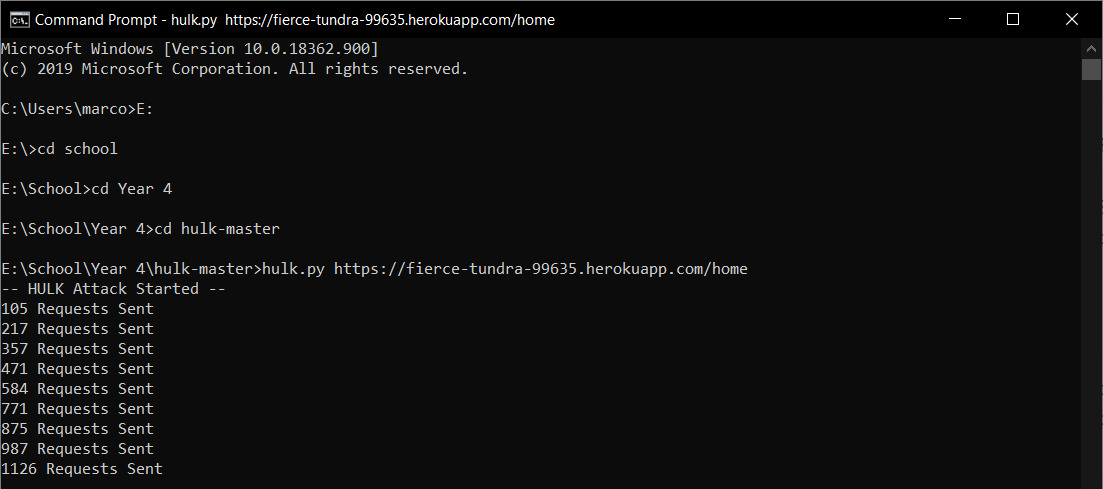
The protocols at the application layer have many vulnerabilities to HTTP flood DDoS attacks. The main reason for this vulnerability is because HTTP flooding attacks are hard to distinguish. The HTTP protocol uses TCP and UDP connections to operate and thus makes it difficult to differentiate from valid traffic. Also, this attack requires less bandwidth than other attacks and can often remain undetected while carrying out attacks (*What Is an HTTP Flooding Attack?*, n.d.).

· ***Description and diagram of the attack***

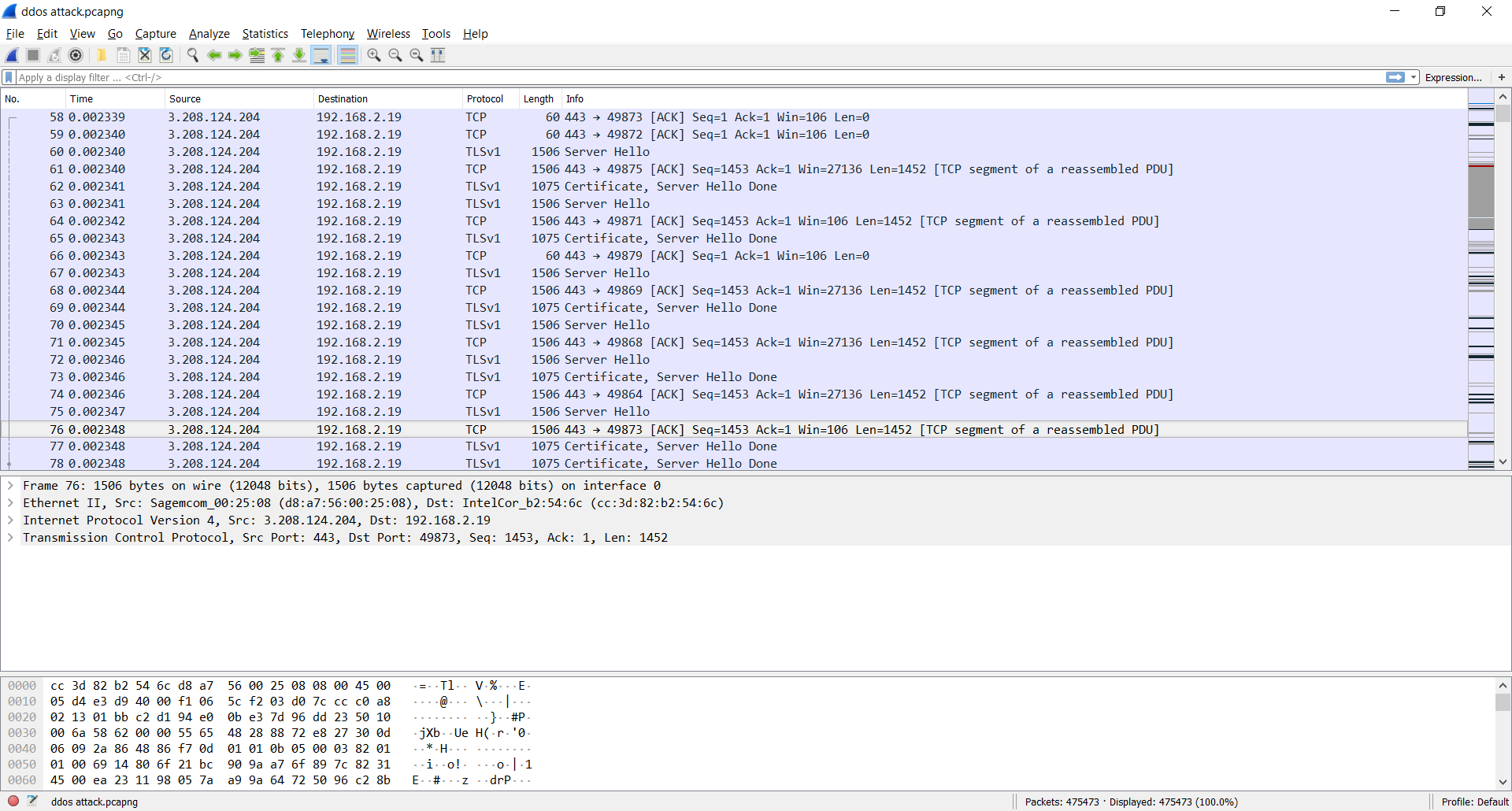
Explain how the attack was carried out. Include a diagram to illustrate how the exploit would typically work. This section should also include sample output (screen captures, packet captures, etc. as appropriate) from running the exploit on a test network.

For our attack, we are planning to locally host our website (we cannot find a host willing to let us DDoS it) and use tools to launch an attack on it inside of the local network. This diagram shows what one of the computers will be doing on the network, and that will be flooding packet requests. We will have multiple machines doing this towards the target.

For our attack, we are planning to use a few tools to do the attack on our website. The main tool we are using is called Hulk. Hulk (Http Unbearable Load King) is an open source DDoS attacking tool that has the ability to bypass the website caching and hit a website's back-end servers with continuous and high volumes of unique and confusing traffic to the system. With this tool, we are planning to use it as the tool to overload our websites server with packets for the attack. The tool will also let us know when the website starts to return 503 errors because it was successful in its attack.



This was a test done on a past heroku website project. Once the specified url link is inserted, the program gets to work and launches the attack, it then displays how many request packets have been sent to the website every few seconds. Now, looking at this example and using wireshark, we can see the packets being sent.



As seen in the capture using wireshark, Hulk creates a unique packet every time for sending to help keep the host system from knowing it is an attack (More on this in the next section).

· ***Signature of the attack***

What traces of the attack are left on the system affected? Does the attack have a "signature" that could be used to detect or block it? If no, explain why no signature exists. If yes, describe the signs and include sniffer output, intrusion detection rules, log files, or other output (such as messages/Syslog etc.) as appropriate and provide a description of what the output means.

With a DDoS attack, there is nothing left on the affected system as with a DDoS attack, we are just overloading the system with packets, these packets do not leave a trace in the system. This attack also does not have a signature that could really be traced. For starters, a system manager is going to have a hard time distinguishing if there is a DDoS attack going on or just an uptick in traffic to the server. Also, with some changes to packets, packets can be completely unique and not have similarities between them causing systems to have trouble creating a pattern to help distinguish real user packets from attacker packets.

· ***How to protect against it***

Describe what can be done to protect against the exploit and/or fix the vulnerability. This section should be covered in two parts. First, what can someone who is running the vulnerable software do so their system can not be compromised? Second, what

Since HTTP flood attacks are hard to distinguish from real incoming traffic, the most effective mitigation techniques involve a combination of traffic profiling methods. One technique involves establishing an IP reputation database to identify and deny suspicious activity. Additionally, implementing security challenges such as JavaScript computation challenges, can be used to test if a bot is being used and thus mitigating the attack (Jaafar et al., 2019).

· ***Remediation System Test***

If appropriate, describe the retesting of your system once remediation has been made to it, to demonstrate the system has been secured

We currently have not done this with our project.

**Section 3 – Security Policy**

Outline what parts of a security policy should be implemented to prevent this attack from occurring and how should the incidence reporting be handled.

The main goal of hackers is to make the attacked resource inaccessible to the user, to do this, a huge number of false requests are sent to the server, which the server is not able to process, and as a result, the website "drops itself" and instead of the usual page, the user sees a static page with an error message. To generate malicious traffic (which is a DDoS attack), most often a large number of devices are infected with a special code and these devices are combined into botnets that send requests to the attacked address.

There is a solution - it is both independent and professional protection against DDoS, including services of specialized companies. But it is important to understand that universal protection measures against DDoS attacks do not exist, hackers are constantly finding new vulnerabilities and ways to trick security systems.

Here are the most popular protection methods against DDoS attacks that should be included in the policy:

1. A firewall is a filter system that helps protect the site from junk traffic before it even gets to the site. The main function of the firewall is the fight against viruses. It also protects against DDoS somehow, but for weak attacks, it is enough. Moreover, it will cost less than a fully equipped Anti-DDoS.

2. Anti-DDoS is more flexible and intelligent than a regular Firewall. The system automatically arranges filters depending on the type and power of the attack and is able to conduct additional manipulations with traffic. This Anti-DDoS ​​service is offered by hosters whose servers are already connected to the protection system. Advantages of Anti-DDoS: one common bill for all services from the hoster and the fact that all settings are performed by the provider’s specialists.

3. Protected IP transit through a virtual tunnel. This service is suitable for projects with large volumes of traffic, to protect against the junk traffic of the entire autonomous system at once. This is an expensive service used by data centers, hosting providers, domain name registrars, telecom operators that do not have their own filtering stations.

There are also preventive some measures that can help you:

1. Do not save money on hardware. Spend a little more money on a more expensive hosting and server. Pay attention to the channel width and the number of CPUs. As a rule, these resources are "destroyed" during DDoS. This will not save you from large and planned attacks, but it will definitely protect you from an inexperienced hacker. You can still be saved using the cloud hosting service, with its help you can add the missing resources at any time.

2. Configure the software on your server. Use traffic distribution between two web servers. (For example Apache and Nginx proxy).

3. Optimize the queries. Try to avoid heavy queries by refactoring the code, adding the missing indexes to the databases, and more. When there are too many of them, the probability of server failure is high, even without a DDoS attack.

4. Track the number of requests per second. Look at the number of requests per second. They grow if a massive botnet arrives, and fall if a botnet leaves that crashes the site, making it completely inaccessible to legitimate users. But we are talking about severe changes in website indicators.

***Deliverable:***

This report should be submitted through Blackboard (see timetable at end of document).

*HTTP Flood DDoS Attack*. (n.d.). Cloudflare. Retrieved July 9, 2020, from<https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/>

Jaafar, G. A., Abdullah, S. M., & Ismail, S. (2019, January 10). *Review of Recent Detection Methods for HTTP DDoS Attack* [Review Article]. Journal of Computer Networks and Communications; Hindawi.<https://doi.org/10.1155/2019/1283472>

What is an HTTP Flood | DDoS Attack Glossary | Imperva. (n.d.). *Learning Center*. Retrieved July 9, 2020, from<https://www.imperva.com/learn/application-security/http-flood/>

*What is an HTTP Flooding Attack?* (n.d.). NETSCOUT. Retrieved July 9, 2020, from<https://www.netscout.com/what-is-ddos/http-flood-attacks>

What is a DDoS Attack? Retrieved July 9, 2020, from

<https://aws.amazon.com/shield/ddos-attack-protection/>

7 Tips for Defending Your Network against DDoS Attacks Retrieved July 9, 2020, from

<https://www.corero.com/blog/7-tips-for-defending-your-network-against-ddos-attacks/>

How to Prevent DDoS Attacks: 6 Tips to Keep Your Website Safe Retrieved July 9, 2020, from

<https://www.esecurityplanet.com/network-security/how-to-prevent-ddos-attacks.html>

Six Steps to Defend Against DDoS Attacks Retrieved July 9, 2020, from

<https://www.a10networks.com/blog/how-defend-against-ddos-attacks-six-steps/>