CHAMELEON

FOR OUR SMARTER WORLD

Denial of Service – Attack Report

Hamish Burnett – (222282244)

Contents

Executive Summary

The attack that will be tested in this operation, will be a Denial of Service attack. A Denial of Service (DoS) attack, is an attack that prevents legitimate users from being able to interact with a resource, due to the resource being overloaded (OWASP Foundation, n.d.). A DoS attack is executed by an individual, or a small group of actors that coordinate their attacks (Cloudflare, n.d.-b). DoS attacks can also be used to launch a Distributed Denial of Service (DDoS) attack, which is where many actors, such as through a botnet, are coordinated to simultaneously launch DoS attacks, in order to prevent the resource from being accessed.

The tools that were utilised during this test, included Kali Linux as the Operating System, Wine, to simulate the Windows environment on the Linux machine, LOIC (Low Orbit Ion Cannon) for performing TCP, UDP, and HTTP DoS attacks, and HOIC (High Orbit Ion Cannon), which is the successor to LOIC.

It was found that the Chameleon website is vulnerable to HTTP DoS attacks, as the HTTP DoS attack executed through LOIC was successful in flooding the website with information, which prevented legitimate users from being able to access the resource. The Chameleon website may also be vulnerable to TCP, and UDP flooding attacks, as the website experienced short delays when loading the information during these attacks.

Recommendations to prevent DoS attacks in the future, include installation of a Web Application Firewall, implementation of CAPTCHA verifications, and blacklisting of malicious IP addresses.

Introduction

A Denial of Service (DoS) attack, is an attack that prevents legitimate users from being able to interact with a resource, due to the resource being overloaded (OWASP Foundation, n.d.). In this security test, the resource that will be tested is the Chameleon website. If the attack is successful, the attack will temporarily prevent legitimate users from being able to interact with the Chameleon website. A DoS attack is executed by an individual, or a small group of actors, that coordinate their attacks (Cloudflare, n.d.-b). DoS attacks can also be used to launch a Distributed Denial of Service (DDoS) attack, which is where many actors, such as through a botnet, are coordinated to simultaneously launch DoS attacks, in order to prevent the resource from being accessed. The difference between the two, is that DoS attacks consist of small amounts of attackers, whereas DDoS attacks consist of large groups of attackers.

The operation of this security test will determine whether the Chameleon website is vulnerable to DoS attacks, and if so, present recommendations to prevent DoS attacks in the future.

Tools Used

Three main tools were that were used, are listed below:

- Kali Linux
- Wine
- LOIC – Low Orbit Ion Cannon
- HOIC – High Orbit Ion Cannon

The attacks were performed using Kali Linux as the Operating System (OS), through a virtual machine. The application Wine was used on Linux to run the tools, as they are Windows based tools. Wine simulates the Windows OS (WINE HQ, n.d.). The LOIC (Low Orbit Ion Cannon) was used to launch DoS attacks against the Chameleon website. LOIC is an open source network stress tool, and can launch DoS attacks through a range of mechanisms, including through the use of TCP, UDP, and HTTP protocols (NewEraCracker, 2022; imperva, n.d.-b). HOIC is the successor to the LOIC tool, and has enhanced features, which include being able to launch DDoS attacks, can target multiple websites simultaneously, and has added measures to avoid detection, when launching DoS attacks (imperva, n.d.-a).

Scope of Testing

The scope of this security test, was limited to the Chameleon website. LOIC and HOIC were the only tools used to launch the DoS attacks, for the tests.

Methodology

Wine was first installed onto the Kali Linux machine, as LOIC/HOIC are run as windows executables (.exe format). Wine allows Linux to run .exe files. Wine was downloaded from the website

https://wiki.winehq.org/Debian . This program was installed, and to check that it was installed correctly, the command "*wine*" was entered into the command line (See Figure 1). This confirmed that Wine was installed successfully.

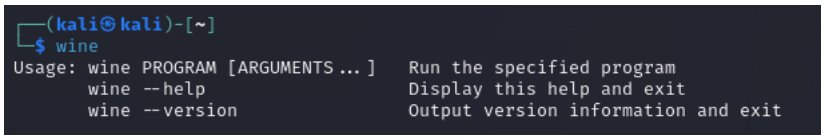

Figure 1: Screenshot showing that Wine was installed successfully.

LOIC

LOIC was then installed from the SourceForge website (https://sourceforge.net/projects/loic/). The *Download* button was pressed, which downloaded the LOIC.exe file. LOIC was then run using the command: *wine LOIC.exe*, as shown in Figure 2.

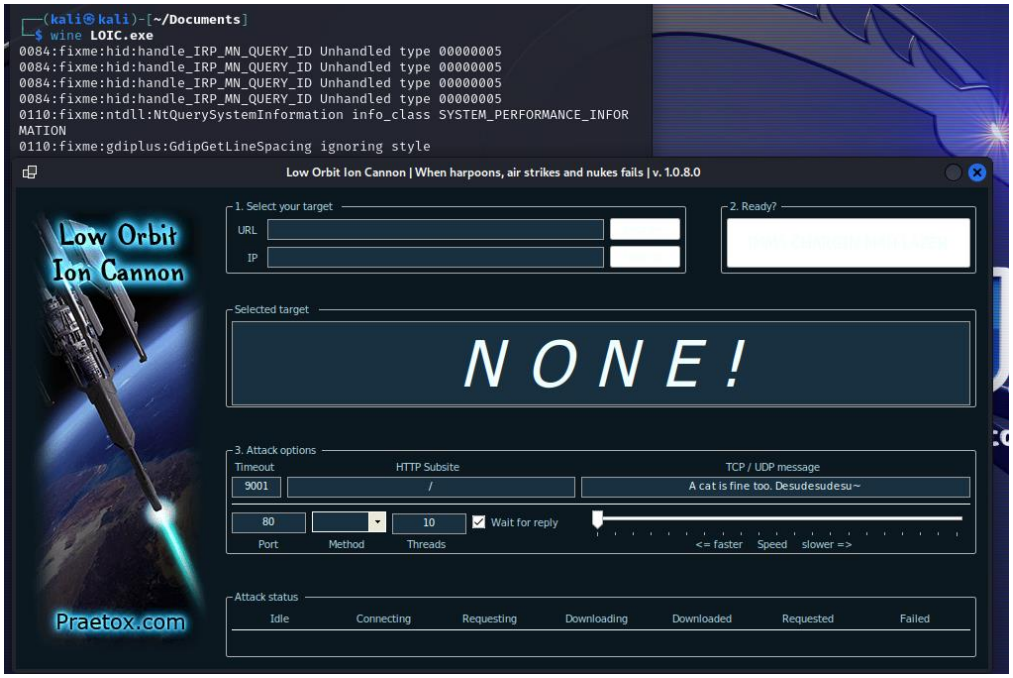

Figure 2: Screenshot showing the command: wine LOIC.exe, and the LOIC tool being opened.

In LOIC, the parameters were set, including the target URL, the timeout duration for packets, the port, method (TCP, UDP or HTTP), the number of threads (number of users LOIC should emulate), and a checkbox to choose whether LOIC will wait for responses. An example of the parameters are shown below, in Figure 3.

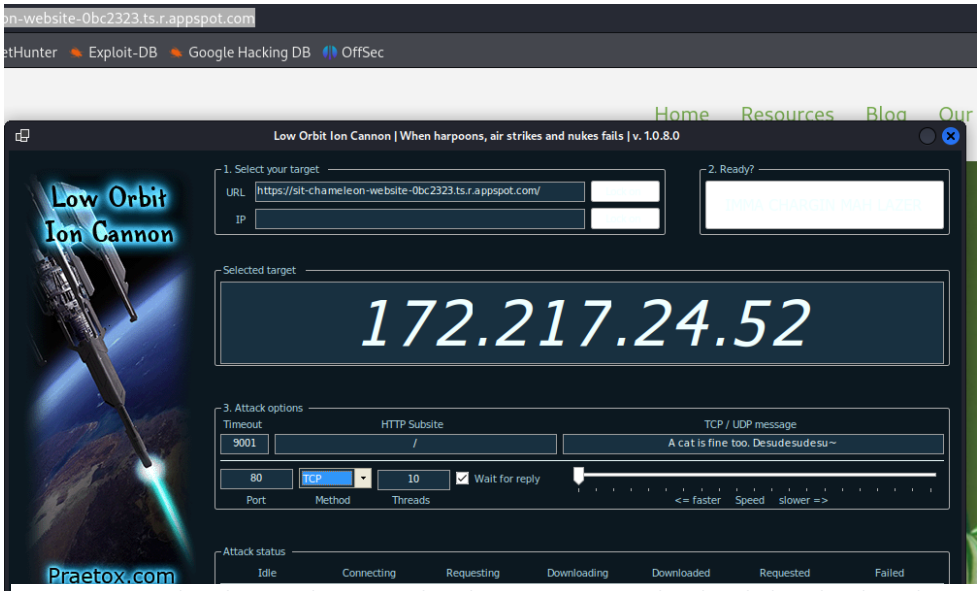

Figure 3: Screenshot showing the LOIC tool, with parameters completed, including the Chameleon Website URL, the timeout being set to 9001, the port to 80, the method to TCP, number of threads to 10, and the checkbox for program will wait for a response.

LOIC was then run, using the white box in the top right corner (See Figure 3). A screenshot of LOIC running is shown in Figure 4. Note the Attack Status, which identifies that there were 10,051,791 requests sent to the Chameleon website (See Red Box in Figure 4).

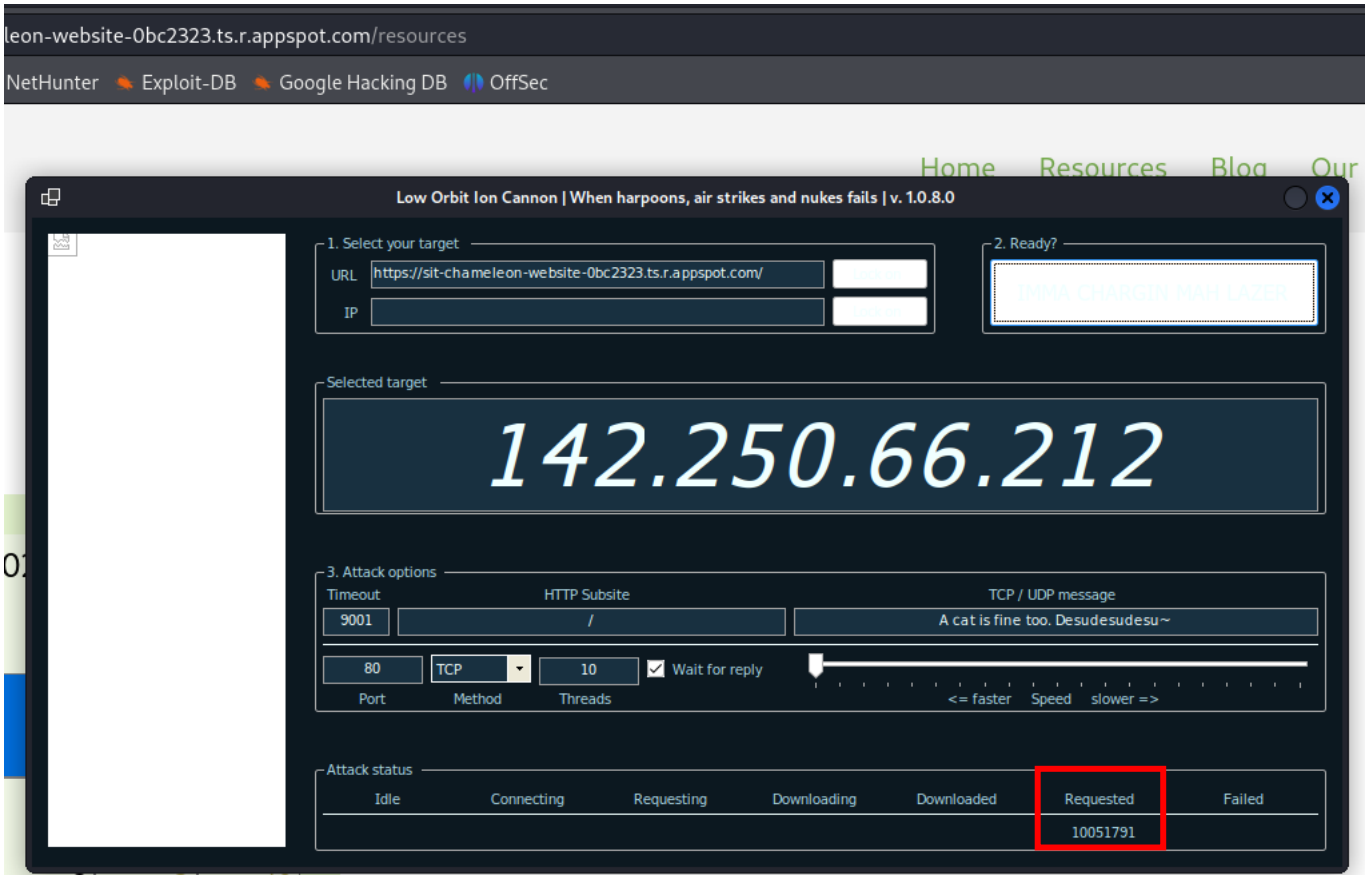

Figure 4: Screenshot of LOIC running. Note the amount of requests that have been sent, in the red box.

LOIC was then run a number of times, each with differing parameters, to determine whether the Chameleon website is vulnerable to DoS attacks.

HOIC

After the testing was completed using LOIC, HOIC was used to test the website. HOIC was installed from SourceForge (https://sourceforge.net/projects/high-orbit-ion-cannon/). The download included a series of files, which included a .exe file. Once HOIC was installed, it was run, using the command: *wine hoic2.1.exe*. The command, and the HOIC tool which was opened, are shown below, in Figure 5.

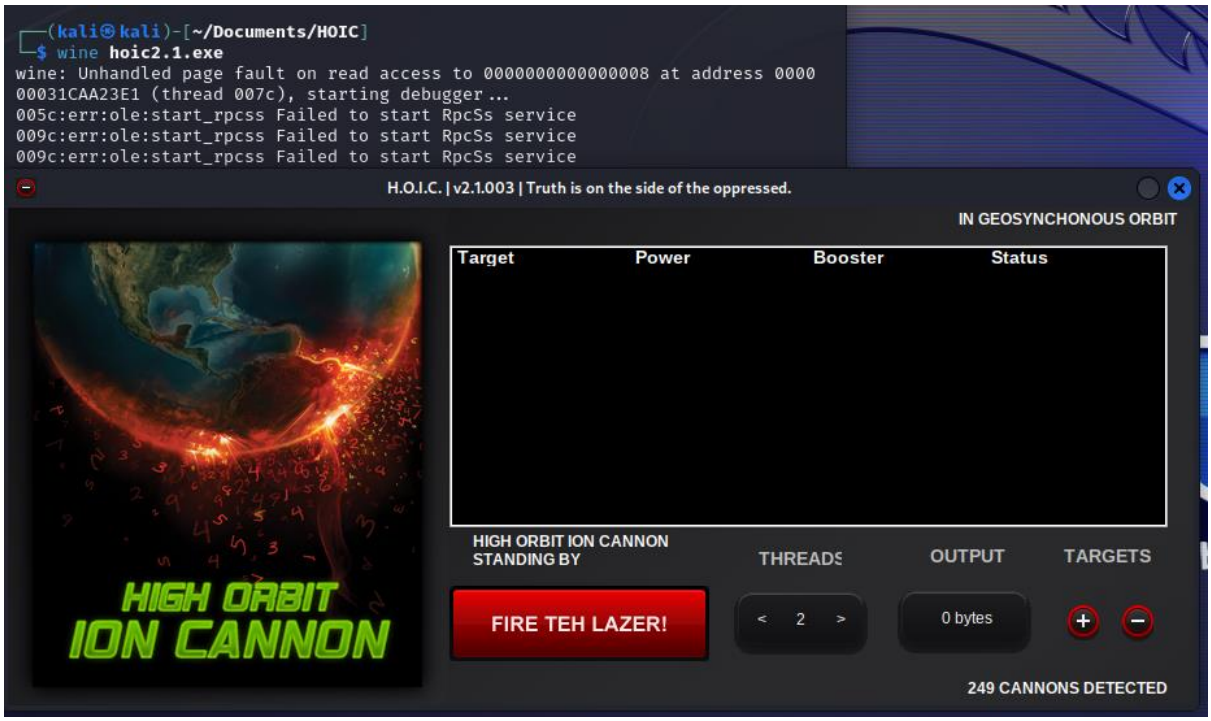

Figure 5: The command to open HOIC, and the HOIC Home Screen, once opened.

Once HOIC has been opened, the user clicks on the "+" button, located on the bottom right of the screen (See Figure 5). This will allow the user to specify the target that they wish to perform a DoS attack against, and provide them with option to specify the power of the attack. Multiple targets can be selected at once, to perform wide-spread DoS attacks simultaneously.

Once the target has been chosen, the user can specify the number of threads, to perform the attack, which can modify the power of the attack. Figure 6 shows a target being selected, and the attack has been launched.

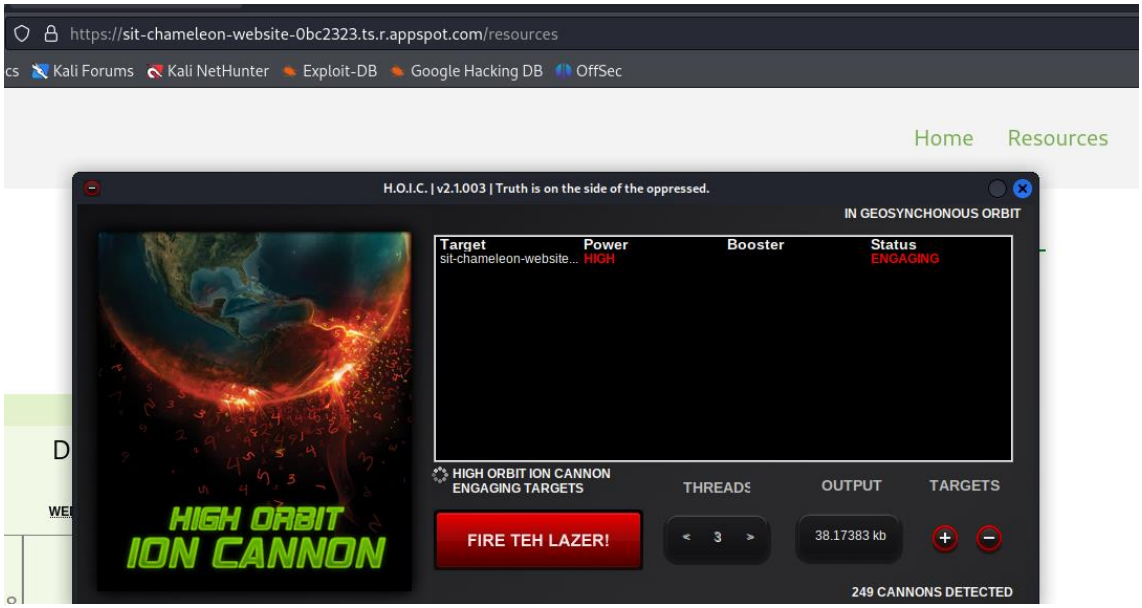

Figure 6: Screenshot of HOIC performing a DoS attack. Note the output displays 38.17 kB of data has been sent to the Chameleon Website.

Results

A series of tests were performed using LOIC and HOIC, with varying parameters. The results of these tests are listed below.

LOIC

TCP

LOIC was used to test the Chameleon website against a DoS attack through the TCP protocol, with the tool simulating 100 users (through the Threads parameter being set to 100), and a packet timeout being set to 9999001 milliseconds. In total, the attack lasted 22 minutes, and 10,556,857 packets were sent to the Chameleon website. The website remained up, and there was a minimal delay in accessing the website. Therefore, the TCP DoS attack was unsuccessful. To determine whether the DoS attack was successful, the site was opened in the VM browser, a phone browser connected to the same WiFi network that was being used for the attack, and a separate phone, which was connected to a different WiFi network than the network that facilitated the DoS attack. See Figure 7, for a screenshot of the attack, after being completed.

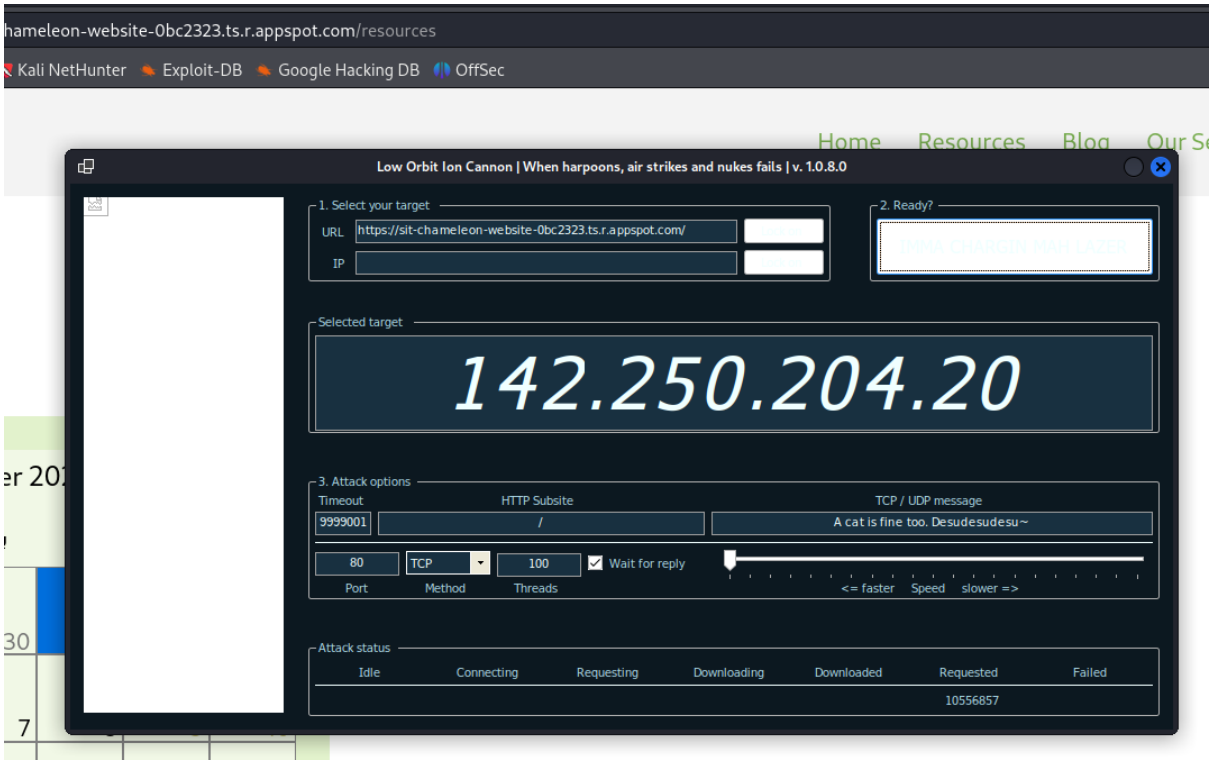

Figure 7: Screenshot showing TCP DoS attack launched through LOIC.

UDP

LOIC was then used to test the Chameleon website against a DoS attack through the UDP protocol, with similar parameters to the TCP DoS attack. In total, the attack lasted 15 minutes, and 8,828,146 packets were sent to the Chameleon website. Although the UDP test was shorter than the TCP test, several tests were completed using UDP, with different parameters, and increased time periods. Apart from minor delays, the website remained accessible. Therefore, the UDP DoS attack was unsuccessful. To determine whether the DoS attack was successful, the same tests that were used for the TCP attack were used, including accessing the site through multiple devices, and using networks that the DoS attack was executed on, and networks that were not used for the attack. See Figure 8, for a screenshot of the attack, after being completed.

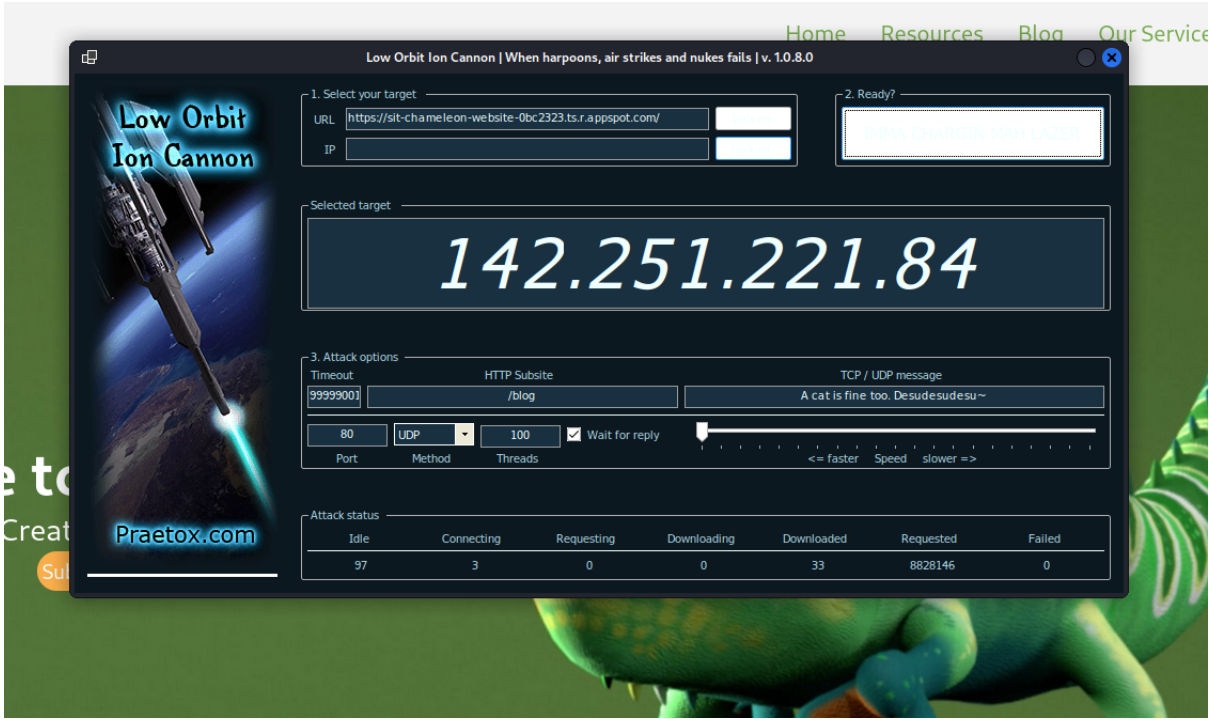

Figure 8: Screenshot showing the UDP DoS attack launched through LOIC.

HTTP

LOIC was then used to test the Chameleon website against a DoS attack through the HTTP protocol, with similar parameters to the TCP, and UDP DoS attacks. In total, the attack lasted 9 minutes, and 157 requests were made to the Chameleon website (See Figure 9). The HTTP attack was found to be successful, as the Chameleon website became temporarily unavailable. When the page was requested in the VM, the website loaded for 30 seconds, before failing, and displaying an error stating that the connection had timed out (See Figure 10). The Chameleon website was then requested on the phone that was using the same network as the VM, and encountered a similar error (See Figure 11). As a final check, the website was loaded on another phone, that was not using the WiFi network that transmitted the DoS attack, which resulted in the same error (See Figure 12). For all three pages, the website was unable to be accessed. It can be concluded that the DoS attack via HTTP was successful. Therefore, as the legitimate user was unable to access the website (via the phone on a different network to the network that enabled the attack), the attack was successful.

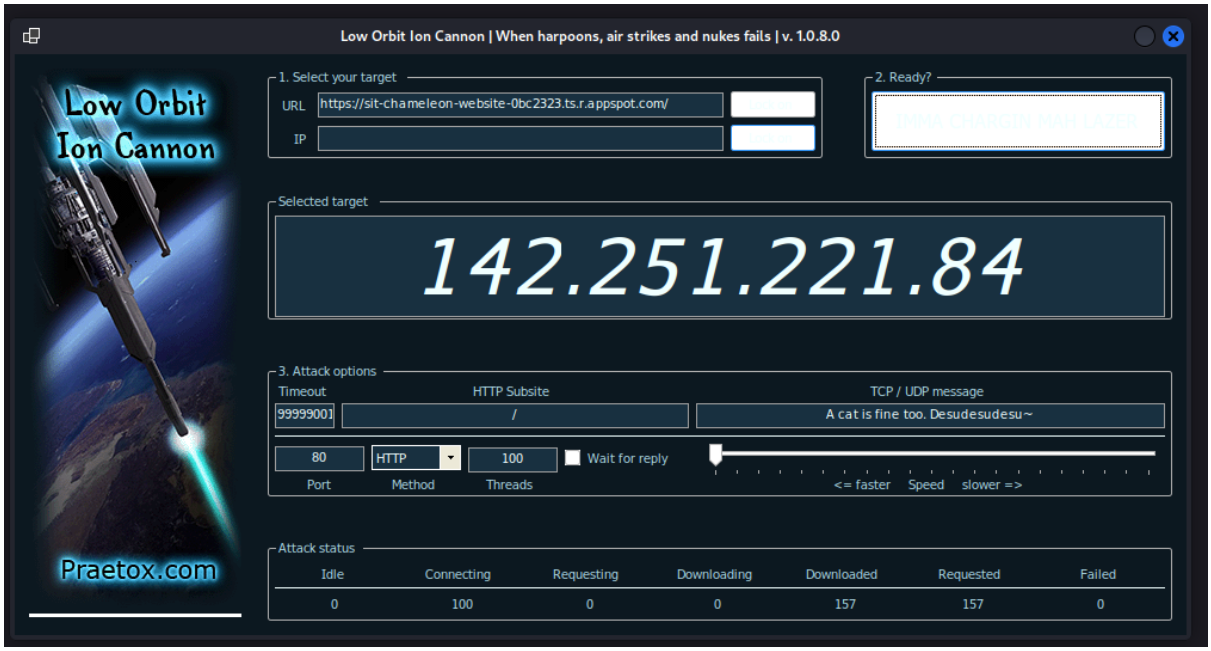

Figure 9: Screenshot showing the HTTP DoS attack launched through LOIC.

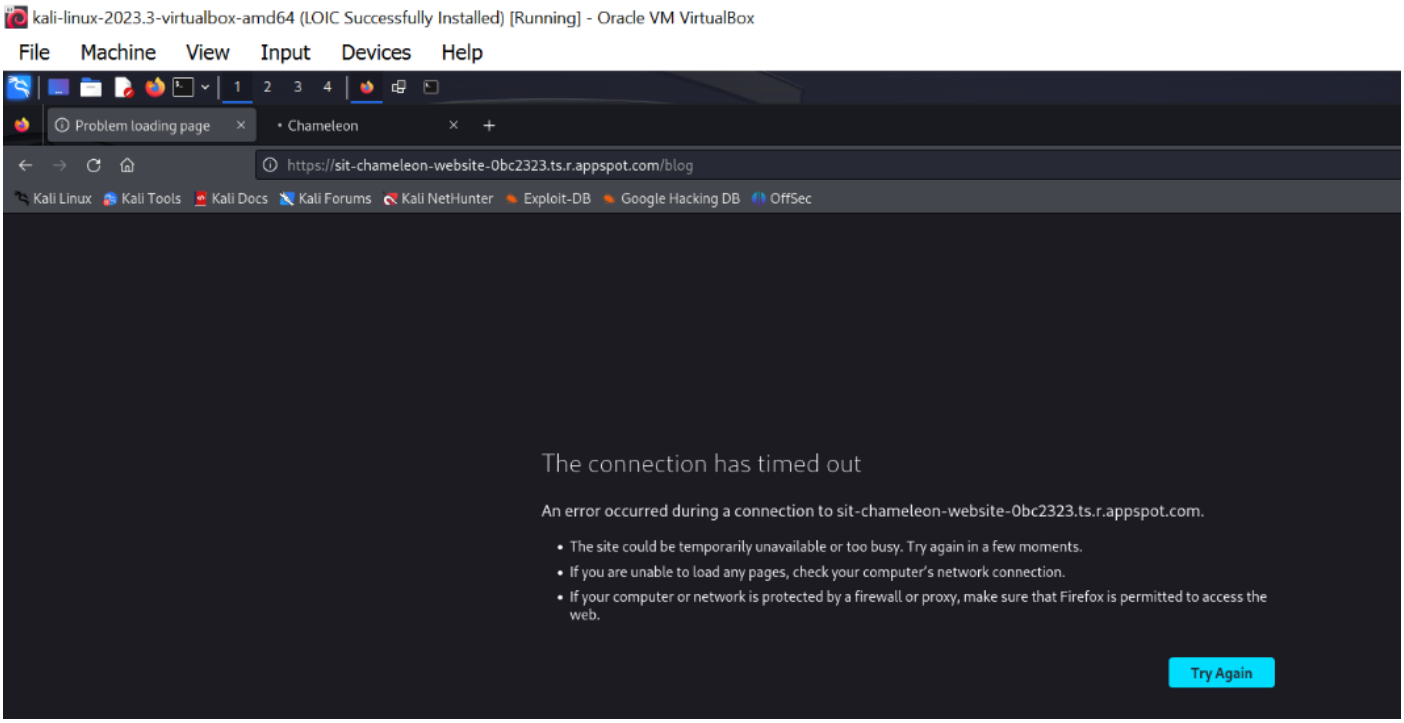

Figure 10: Screenshot showing the Chameleon website encountering an error, resulting in the page being inaccessible. The page needed to load for 30 seconds, before the connection timed out, which resulted in the error shown above.

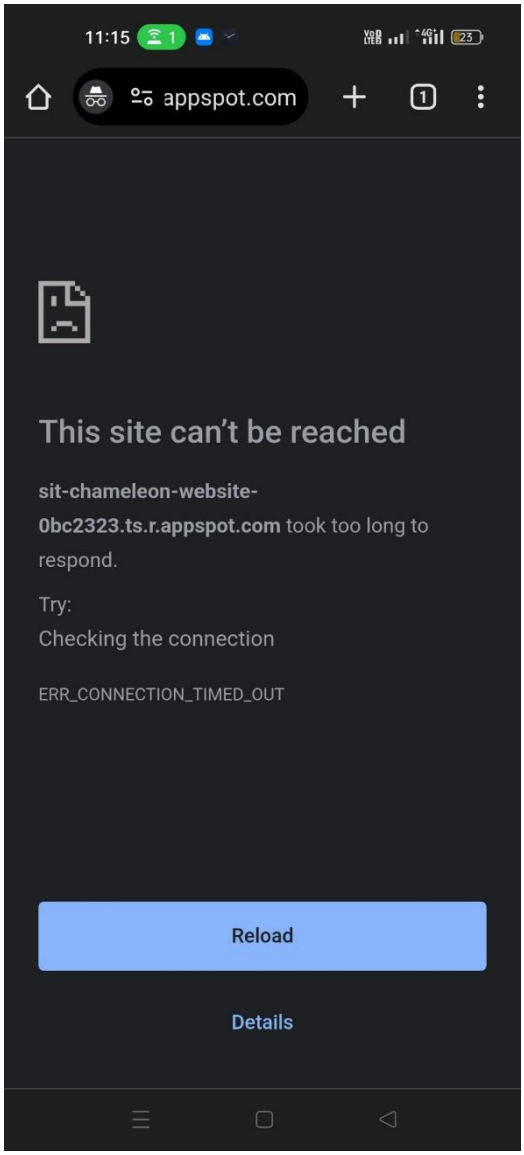

Figure 11: Screenshot (Left) showing the Chameleon website encountering an error, resulting in the page being inaccessible. This screenshot shows a phone that requested the website. This phone was connected on the same network as the VM that was performing the test. The page needed to load for 30 seconds, before the connection timed out, which resulted in the error shown to the left.

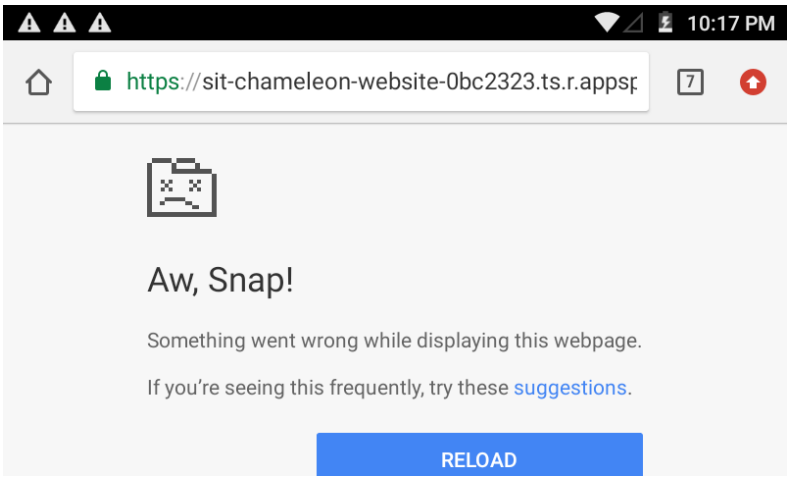

Figure 12: Screenshot (Left) showing the Chameleon website encountering an error, resulting in the page being inaccessible. This screenshot shows a phone that requested the website. This phone was connected to a different network, than the network that was used to commit the DoS attack. The page needed to load for 30 seconds, before the connection timed out, which resulted in the error shown to the left.

A HTTP DoS attack is when an attacker floods a resource (Chameleon server), with many requests for a website (Chameleon website) (Cloudflare, n.d.-a). The attack will overwhelm the server, and the result will be that the legitimate users will be unable to request the website (Chameleon website). HTTP is an application layer protocol of the TCP/IP stack, and is used to load websites.

HOIC

HOIC was used to test the Chameleon website against a DoS attack. A single target was selected (Chameleon website), and the power of the attack was set to high. Two threads (out of 6) were used for the attack. Apart from this attack, another attack was run using HOIC, which was set to six threads, but this resulted in the HOIC application crashing. This attack resulted in hundreds of kilobytes being sent to the Chameleon website within a matter of seconds. The attack (with two threads) was launched, and lasted 15 minutes. After this time, the Chameleon website was still available and responsive, and the DoS attack was terminated. The test sent 38 kB of data to the Chameleon website (See Figure 13). The Chameleon website remained accessible, throughout the HOIC attack. Therefore, the HOIC DoS attack was unsuccessful.

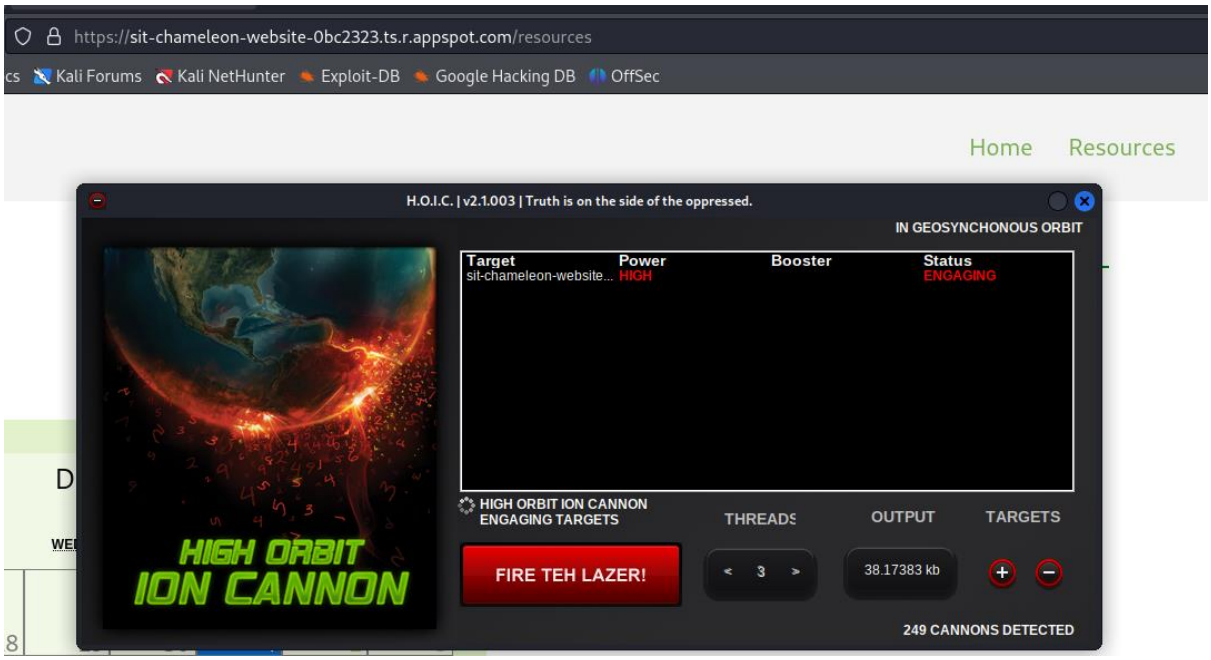

Figure 13: Screenshot showing DoS attack launched through HOIC.

Recommendations

As it has been found that the Chameleon website is vulnerable to DoS attacks, it is recommended that actions be taken to prevent DoS attacks in the future. The HTTP DoS attack occurs on the application layer of the TCP/IP networking stack, which can make it difficult to distinguish from legitimate traffic. However, there are several measures that can be taken, to reduce the vulnerability of the Chameleon website to HTTP DoS attacks. These include:

- Implement a Web Application firewall, to detect and prevent HTTP DoS attacks (Cloudflare, n.d.-c). Web Application firewalls can prevent Cross Site Forgery, Cross Site Scripting, File Inclusion, SQL Injection attacks, and DoS attacks. One provider of a Web Application Firewall,

is Cloudflare, which is available from https://www.cloudflare.com/application-services/products/waf/ .

- Implement a CAPTCHA based system, to identify whether requests are legitimate, or if they are being sent as part of a DoS attack. The CAPTCHA would require the user to perform a challenge, which an attacker performing a DoS attack would not be able to complete.
- Implement an IP address database, which flags IP addresses that are being used for malicious activity, and then blacklists those IP addresses. This would be particularly relevant for preventing LOIC DoS attacks, as LOIC does not obfuscate the IP address of the requests it sends, and all attacks committed through LOIC utilise the same IP address, making IP blacklisting less of a challenge.

Conclusion

Through the security testing, it was identified that the Chameleon website is vulnerable to HTTP DoS attacks. The website may also be vulnerable to TCP and UDP attacks, however, these attacks were unsuccessful.

Denial of Service attacks prevent legitimate users from being able to interact with a resource, due to the resource being overloaded. A DoS attack is executed by an individual, or a small group of actors that coordinate their attacks (Cloudflare, n.d.-b). DoS attacks can also be used to launch a Distributed Denial of Service (DDoS) attack, which is where many actors, such as through a botnet, are coordinated to simultaneously launch DoS attacks, in order to prevent the resource from being accessed. The difference between the two, is that DoS attacks consist of small amounts of attackers, whereas DDoS attacks consist of large groups of attackers.

It was found that the Chameleon website is resistant to TCP, and UDP attacks, but was found to be susceptible to HTTP Denial of Service attacks. A HTTP Denial of Service attack is when an attacker floods a resource (Chameleon server), with many requests for the website (Chameleon website) (Cloudflare, n.d.-a). The attack will overwhelm the server, and the result will be that legitimate users will be unable to request the website (Chameleon website). HTTP is an application layer protocol within the TCP/IP stack, and is used to load websites.

Mitigations to prevent HTTP Denial of Service attacks include a Web Application Firewall (Cloudflare, n.d.-a). These firewalls also protect against SQL Injection, and Cross Site Scripting. One provider of these firewalls, is Cloudflare, and is available at: https://www.cloudflare.com/application-services/products/waf/ . Another measure to take, is to implement a CAPTCHA system, to verify that requests are legitimate. A final countermeasure, would be to implement an IP address database, which blacklists requests from certain IP addresses. These addresses can be identified during a Denial of Service attack.

References

Cloudflare (n.d.-a) *HTTP flood attack*, accessed 2023. https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/

Cloudflare (n.d.-b) *What is a denial-of-service (DoS) attack?* accessed 2023. https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/

Cloudflare (n.d.-c) *What is a WAF? | Web Application Firewall explained*, accessed 2023. https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/

imperva (n.d.-a) *High Orbit Ion Cannon (HOIC)*, accessed 2023. https://www.imperva.com/learn/ddos/high-orbit-ion-cannon/

imperva (n.d.-b) *Low Orbit Ion Cannon (LOIC)*, accessed 2023. https://www.imperva.com/learn/ddos/low-orbit-ion-cannon/

NewEraCracker (2022) *LOIC*: GitHub, accessed 2023. https://github.com/NewEraCracker/LOIC

OWASP Foundation (n.d.) *Denial of Service*, accessed 2023. https://owasp.org/www-community/attacks/Denial_of_Service

WINE HQ (n.d.) *What is Wine?,* accessed 2023. https://www.winehq.org/