2024

IT Security

AISHWARYA KAGGDAS

Introduction :

This project centres on the Denial of Service threat type, delving into the efficacy of tools wielded by both attackers and defenders in addressing such attacks. A Denial-of-Service (DoS) attack is a deliberate attempt to render a machine or network unusable, thereby preventing legitimate users from accessing it. These attacks are executed by inundating the target with an excessive amount of traffic or by sending malicious data that can lead to system crashes. In either case, the goal of a DoS attack is to disrupt the availability of services or resources, causing inconvenience or harm to users, such as employees, members, or account holders, who rely on those services for their intended purposes.

Criteria 1: Justification and planning

A denial-of-service (DoS) attack represents a security threat where an attacker disrupts legitimate users' access to computer systems, networks, services, or other IT resources. Typically, attackers inundate web servers, systems, or networks with an excessive amount of traffic, overwhelming the victim's resources and rendering them inaccessible to others.

While restarting a system may resolve an attack that crashes a server, recovering from flooding attacks is more challenging. Distributed DoS (DDoS) attacks, where attack traffic originates from numerous sources, pose even greater recovery challenges.

DoS and DDoS attacks often exploit vulnerabilities in networking protocols and their handling of network traffic. For instance, attackers might inundate a vulnerable network service with numerous packets from various Internet Protocol (IP) addresses to overwhelm the service.

History of denial-of-service attacks

Denial-of-Service (DoS) attacks targeting internet-connected systems have a lengthy history, with notable incidents such as the Robert Morris worm attack in 1988. Morris, then a graduate student at Massachusetts Institute of Technology (MIT), unleashed a self-propagating malware, known as a worm, onto the internet. This malicious program rapidly proliferated across networks, causing buffer overflows and initiating DoS attacks on vulnerable systems. During this era, the internet predominantly linked research and academic institutions. Despite its relatively narrow reach, Morris's worm managed to impact approximately 10% of the 60,000 systems in the United States. According to reports from the U.S. General Accounting Office (GAO), now known as the Government Accountability Office, the resulting damage was estimated to be as high as $10 million. Morris faced legal repercussions under the 1986 Computer Fraud and Abuse Act (CFAA), receiving a sentence of 400 community service hours, three years of probation, and a $10,000 fine as a consequence.

Since then, DoS and DDoS attacks have become increasingly common. Recent incidents include:

- Imperva reported a significant DDoS attack against one of its clients on April 30, 2019. The attack reached a peak of 580 million packets per second but was successfully mitigated using the company's DDoS protection software.
- In February 2020, Amazon Web Services (AWS) mitigated one of the largest DDoS attacks ever recorded. This attack, detailed in the AWS Shield Threat Landscape Report Q1 2020, reached a volume of 2.3 Tbps, making it 44% larger than any previous attack encountered by AWS.

- Cloudflare faced a massive HTTP DDoS attack in February 2023, with a peak of 71 million requests per second. This attack was claimed to be the largest of its kind at the time, measuring HTTP requests per second instead of packets or bits per second.
- In August 2023, the hacktivist group NoName057 targeted several Italian financial institutions with slow DoS attacks.
- October 2023 saw the exploitation of a new vulnerability in the HTTP/2 protocol, resulting in the largest HTTP DDoS attack recorded to date. Cloudflare observed one attack with 201 million requests per second, followed by another observed by Google with 398 million requests per second.

How exactly does a DoS attack operate?

A DoS attack disrupts or disables a target system by overwhelming one or multiple layers of the Open Systems Interconnection (OSI) model, which comprises seven layers. Typically, these attacks focus on Layer 3 (network), Layer 4 (transport), Layer 6 (presentation), and Layer 7 (application) of the OSI model, exploiting vulnerabilities at these levels to render the system inaccessible or unusable.

In a DoS attack, cyber attackers often utilize a single internet connection and device to send a barrage of rapid and relentless requests to a target server, aiming to overwhelm the server's available bandwidth.

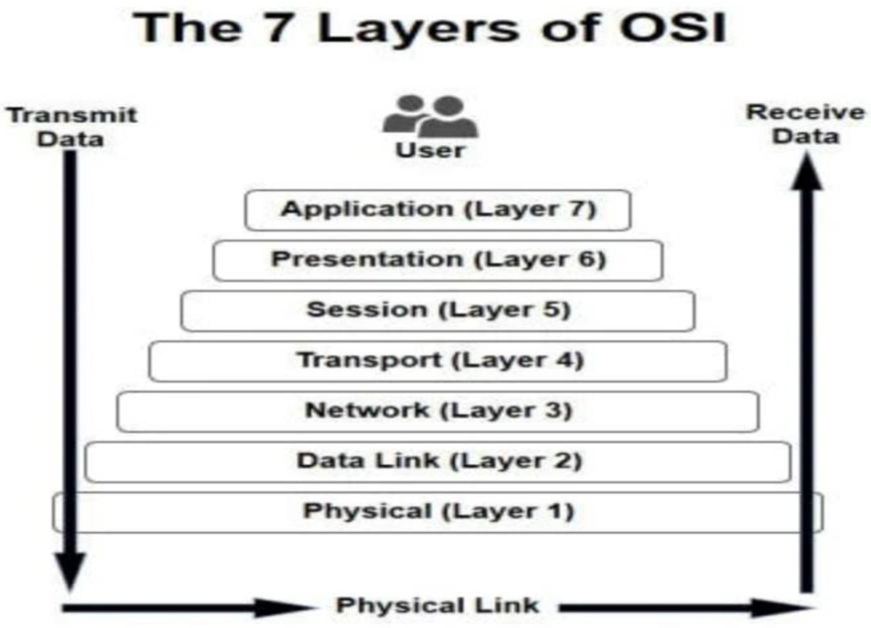


Image 1 : The 7 layers of OSI

DDoS Attack Classification and Types:

DDoS attacks vary significantly in their initiation methods and the extent of impact on the target server. However, they all share the common objective of disrupting legitimate traffic. Given the heterogeneous nature of IoT networks and devices, there exists a diverse range of threats targeting them specifically. As depicted in image 2 , DDoS attacks can thus be categorized into three main groups: volumetric-based, protocol-based, and application-based.

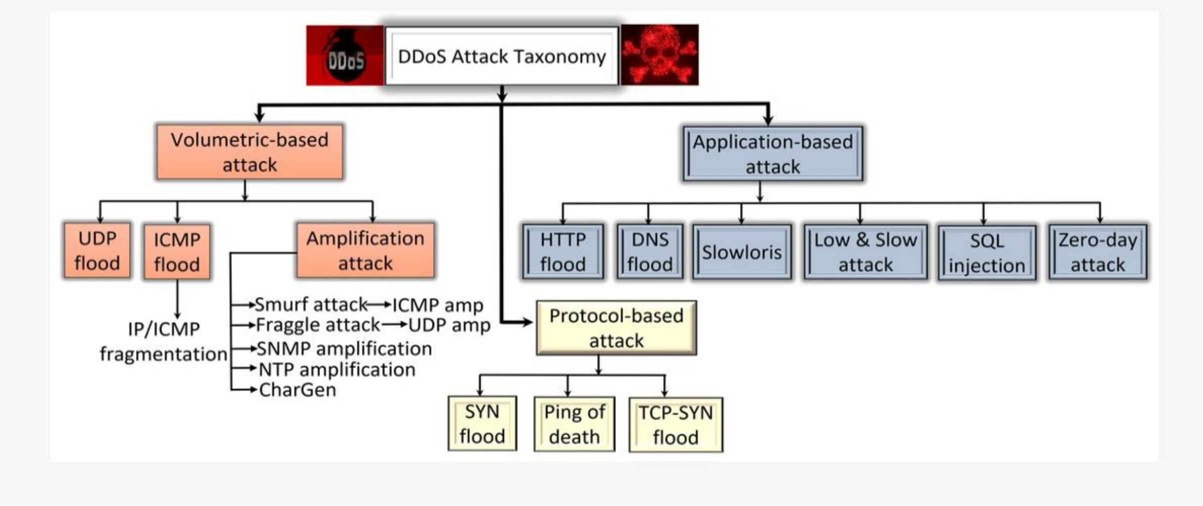


Image 2 : Dos attack and classification

Volumetric-Based DDoS Attacks inundate the target network's available bandwidth with massive data packets, resulting in overwhelming traffic volume. This flood of malicious data saturates the targeted network, effectively denying service to legitimate users. Any server unable to manage this surge in traffic can be swiftly incapacitated by such attacks. The typical structure of a conventional volumetric-based DDoS attack is illustrated in image 3. In this attack strategy, the perpetrators aim to overpower the bandwidth of the victim site by flooding it with a high volume of traffic. Attackers often employ amplification techniques, which entail sending brief legitimate requests to a domain name server (DNS) with a falsified source IP address of the victim, thus intensifying the impact on the target server.

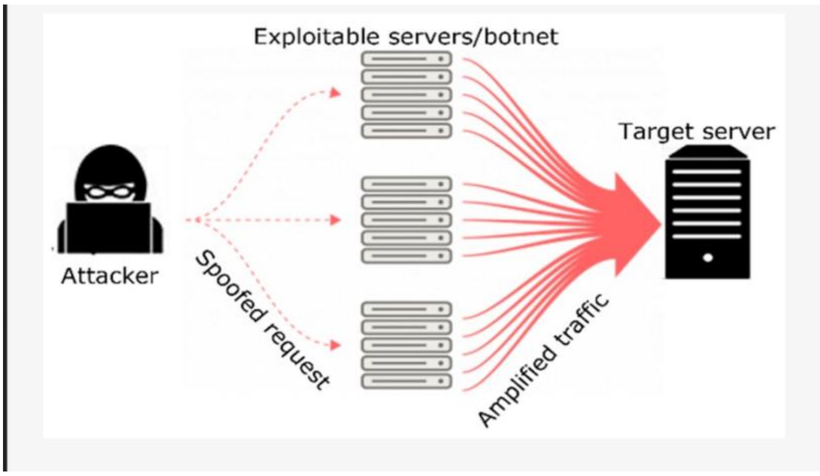


Image 3 : Volumetric Based DDoS attack

The **protocol-based attack**, also known as the "network-layer attack," operates by exploiting vulnerabilities within the protocol stack's Layer 3 and Layer 4. Unlike the volumetric-based attack, which primarily relies on overwhelming traffic volume, this method aims to render the target server inaccessible by exploiting weaknesses in the protocol stack. As a result, existing server resources, as well as other resources like firewalls, are consumed by this type of attack.

Application-Based DDoS Attacks, also recognized as "Layer 7" attacks due to their occurrence at the seventh layer of the OSI model, involve overloading the application layer with an excessive number of login or search requests. These attacks pose challenges in detection and mitigation as the attacker generates attack traffic at a lower rate, making it difficult to distinguish from legitimate traffic. Additionally, the requests sent during these attacks closely resemble regular traffic. The intensity of this attack is measured in requests per second (rps).

DoS attack and defence tools:

DoS attack tools are employed by attackers to target vulnerable networks, systems, and applications, often with the aim of financial gain or furthering political motives. These tools vary widely, ranging from basic scripts that target individual servers to advanced bots and botnets. The primary purpose of DDoS attack tools is to inundate the victim's systems with an overwhelming volume of traffic originating from numerous sources.

DoS attacks disrupt systems by sending malicious traffic from a single machine, typically a computer. These attacks can be quite straightforward; for instance, a basic ping flood attack involves inundating a targeted server with more ICMP (ping) requests than it can efficiently process and respond to.

On the other hand, DDoS attacks utilize multiple machines to send malicious traffic to their target. Frequently, these machines are part of a botnet—a network of compromised computers or devices controlled remotely by an individual attacker. Alternatively, multiple individual attackers may collaborate to launch DDoS attacks by directing traffic from their respective computers.

In today's internet landscape, DDoS attacks are more prevalent and damaging for two main reasons. Firstly, modern security tools have advanced to counter some conventional DoS attacks effectively. Secondly, DDoS attack tools have become relatively inexpensive and userfriendly, facilitating their widespread use.

In instances like MyDoom and Slowloris, the tools operate stealthily within malware, initiating attacks without the system owner's awareness. Stacheldraht exemplifies a typical DDoS tool, employing a hierarchical structure wherein the attacker utilizes a client program to establish connections with handlers. These handlers are compromised systems responsible for issuing commands to zombie agents, which ultimately execute the DDoS attack. Agents are compromised through the handlers, as the attacker employs automated routines to exploit vulnerabilities in programs accepting remote connections on targeted hosts. Each handler has the capability to oversee up to a thousand agents, amplifying the impact of the attack significantly.

Several commonly utilized attacker tools include:

- **Low Orbit Ion Cannon (LOIC):** LOIC is an open-source stress testing application that facilitates TCP and UDP protocol layer attacks via a user-friendly WYSIWYG interface. Derivatives of the original tool have emerged, enabling attacks to be initiated through a web browser.
- **High Orbit Ion Cannon (HOIC):** HOIC was developed to enhance LOIC's capabilities by introducing customizations and expanding its functionalities. Operating through the HTTP protocol, HOIC can execute targeted attacks that are challenging to mitigate. It is designed to be employed by at least 50 individuals collaborating in a coordinated attack effort.
- **Slowloris:** Slowloris is a tool designed to orchestrate low and slow attacks on targeted servers, requiring minimal resources to generate significant impact.
- **R.U.D.Y (R-U-Dead-Yet):** R.U.D.Y. is another tool for conducting low and slow attacks, featuring a simple point-and-click interface for easy initiation. By initiating

multiple HTTP POST requests and maintaining these connections for extended periods, the attack gradually overwhelms the targeted server.

DoS and DDoS attacks manifest in diverse forms, mitigating them necessitates employing a range of strategies. Common techniques to counter DDoS attacks encompass:

- **Rate limiting**: Imposing restrictions on the volume of requests a server will entertain within a specific timeframe.
- **Web application firewalls:** Implementing tools that screen web traffic by applying a set of predefined rules.
- **Anycast network dispersion**: Deploying an extensive, dispersed cloud network positioned between a server and incoming traffic, furnishing supplementary computational resources to handle requests effectively.
- **Firewall:** For a straightforward attack, a firewall could implement a basic measure by incorporating a rule to block any incoming traffic originating from the attackers. This rule could be configured based on specific protocols, ports, or the IP addresses of the source.

Criteria 2 : Application and documentation

I chose the Slowloris tool for its effectiveness in executing low and slow attacks on targeted servers while requiring minimal resources. Slowloris operates by establishing connections with the target server and maintaining them for extended periods, gradually overwhelming the server's capacity to respond to legitimate requests. This tool's ability to create a significant impact with relatively limited resources makes it an ideal choice for simulating a Denial of Service (DoS) attack in Kali Linux.

Slowloris operates as an application layer DDoS assault by employing incomplete HTTP requests to initiate connections between a sole computer and a specific web server. Subsequently, it endeavors to maintain these connections for an extended duration, thereby inundating the target and causing a reduction in its speed and functionality.
This particular DDoS technique demands minimal bandwidth for its execution and solely affects the designated web server, leaving other services and ports unharmed. Slowloris is capable of targeting various types of web server software, although it has demonstrated notable effectiveness particularly against Apache versions 1.x and 2.x.

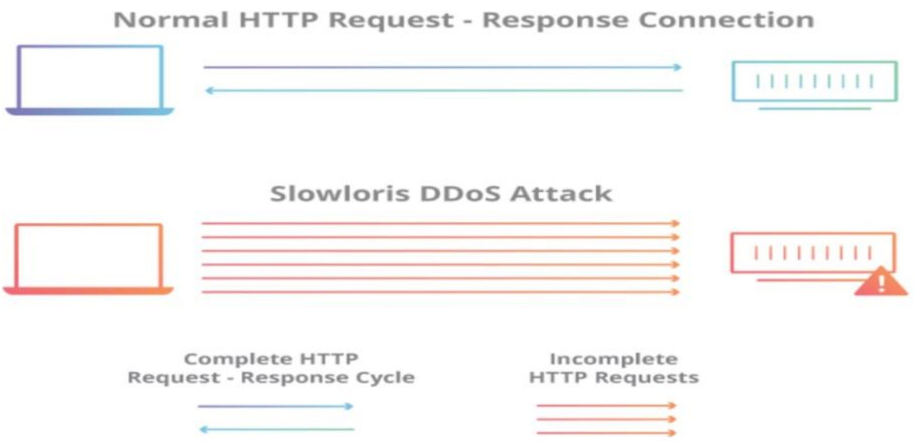
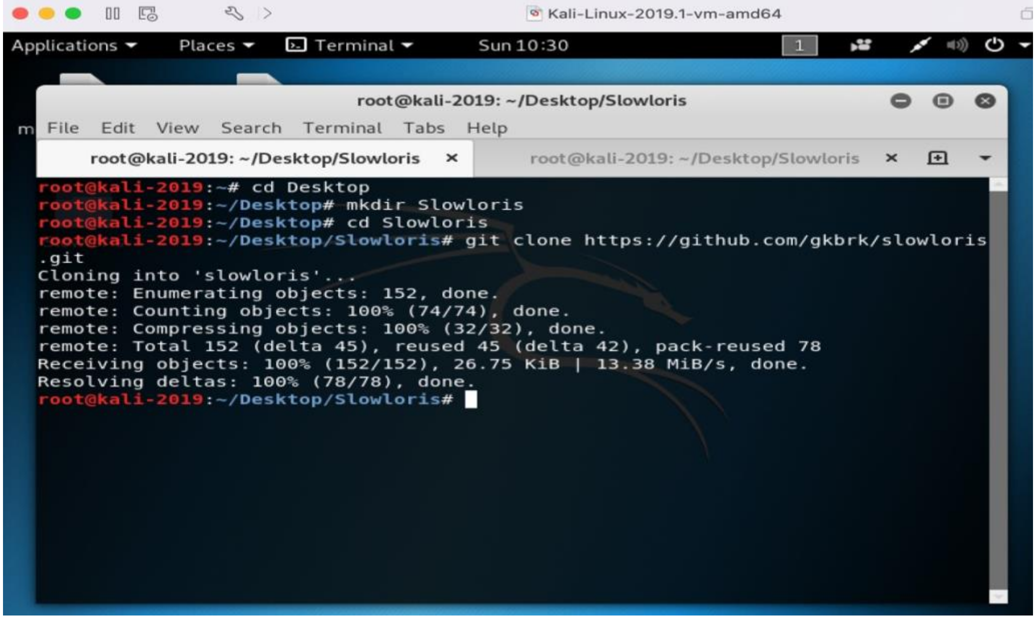

Normal HTTP Request - Response Connection
Slowloris DDoS Attack
Complete HTTP Request - Response Cycle
Incomplete HTTP Requests

Image 4: Slowloris DDoS attack

Install Slowloris :

Note : All screenshots of Kali Linux showcased here belong to me and were captured on my device. Therefore, no references have been made to external sources for these images.

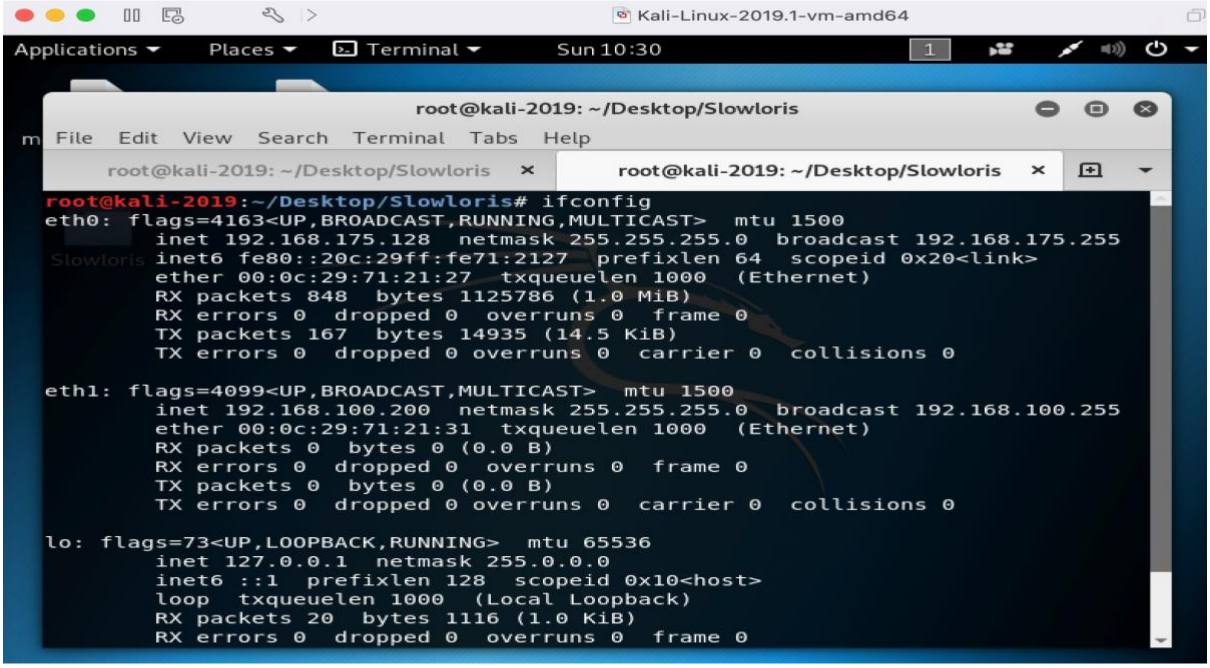
- Launch Kali Linux and access the Terminal application.
- Create a new Directory named Slowloris .
- Clone the Slowloris tool from its GitHub repository to install it on the Kali Linux system. Simply input the provided URL into the terminal while we're in the Slowloris directory we've just created.


Kali-Linux-2019.1-vm-amd64
Applications
Places
Terminal
Sun 10:30
root@kali-2019: ~/Desktop/Slowloris
File Edit View Search Terminal Tabs Help
root@kali-2019: ~/Desktop/Slowloris
root@kali-2019: ~/Desktop/Slowloris

```
root@kali-2019:~# cd Desktop
root@kali-2019:~/Desktop# mkdir Slowloris
root@kali-2019:~/Desktop# cd Slowloris
root@kali-2019:~/Desktop/Slowloris# git clone https://github.com/gkbrk/slowloris
.git
Cloning into 'slowloris'...
remote: Enumerating objects: 152, done.
remote: Counting objects: 100% (74/74), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 152 (delta 45), reused 45 (delta 42), pack-reused 78
Receiving objects: 100% (152/152), 26.75 KiB | 13.38 MiB/s, done.
Resolving deltas: 100% (78/78), done.
root@kali-2019:~/Desktop/Slowloris#
```

Image 5: Clone Slowloris tool from github

Now, it's essential to check the IP address of our machine. Execute ‘ifconfig’ to retrieve the machine's IP address. The IP address is 192.168.175.128 as we can see in the screenshot below.


Kali-Linux-2019.1-vm-amd64
Applications
Places
Terminal
Sun 10:30
root@kali-2019: ~/Desktop/Slowloris
File Edit View Search Terminal Tabs Help

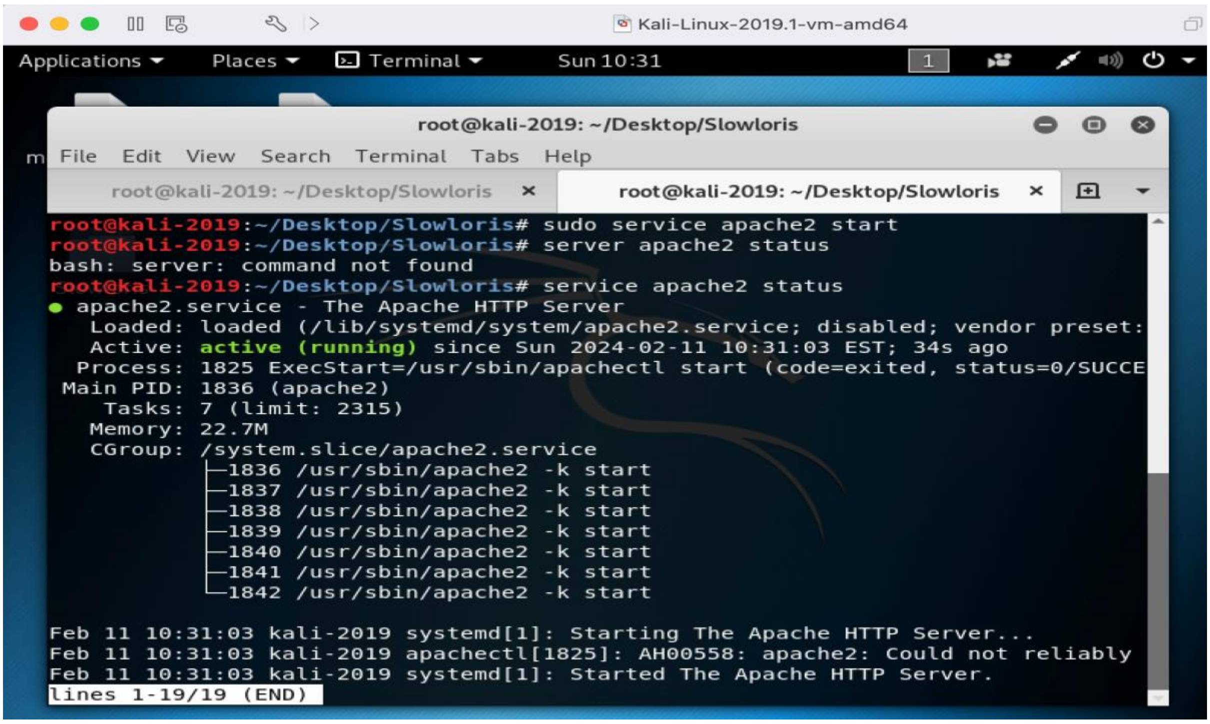
```
root@kali-2019:~/Desktop/Slowloris# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.175.128  netmask 255.255.255.0  broadcast 192.168.175.255
        inet6 fe80::20c:29ff:fe71:2127  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:71:21:27  txqueuelen 1000  (Ethernet)
        RX packets 848  bytes 1125786 (1.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 167  bytes 14935 (14.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.100.200  netmask 255.255.255.0  broadcast 192.168.100.255
        ether 00:0c:29:71:21:31  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 20  bytes 1116 (1.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
```

Image 6: Retrieve IP address

Once we have the IP address, proceed to start the Apache server . After starting the server, it's crucial to verify its status to ensure if it's active. It is active as we can see in the screenshot below.

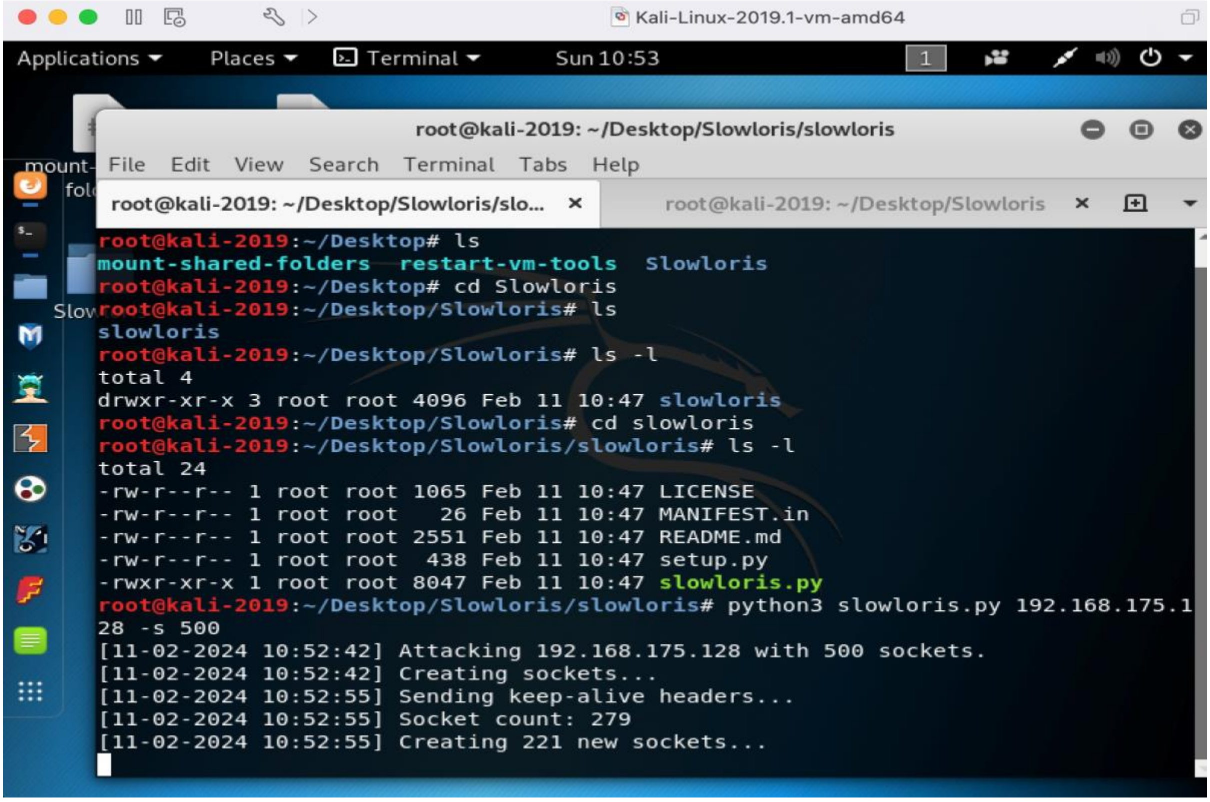

Kali-Linux-2019.1-vm-amd64
Applications
Places
Terminal
Sun 10:31
root@kali-2019: ~/Desktop/Slowloris
File Edit View Search Terminal Tabs Help

```
root@kali-2019:~/Desktop/Slowloris# sudo service apache2 start
root@kali-2019:~/Desktop/Slowloris# server apache2 status
bash: server: command not found
root@kali-2019:~/Desktop/Slowloris# service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset:
   Active: active (running) since Sun 2024-02-11 10:31:03 EST; 34s ago
  Process: 1825 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCE
 Main PID: 1836 (apache2)
    Tasks: 7 (limit: 2315)
   Memory: 22.7M
   CGroup: /system.slice/apache2.service
           ├─1836 /usr/sbin/apache2 -k start
           ├─1837 /usr/sbin/apache2 -k start
           ├─1838 /usr/sbin/apache2 -k start
           ├─1839 /usr/sbin/apache2 -k start
           ├─1840 /usr/sbin/apache2 -k start
           ├─1841 /usr/sbin/apache2 -k start
           └─1842 /usr/sbin/apache2 -k start

Feb 11 10:31:03 kali-2019 systemd[1]: Starting The Apache HTTP Server...
Feb 11 10:31:03 kali-2019 apachectl[1825]: AH00558: apache2: Could not reliably
Feb 11 10:31:03 kali-2019 systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

Image 7: Start the Apache server

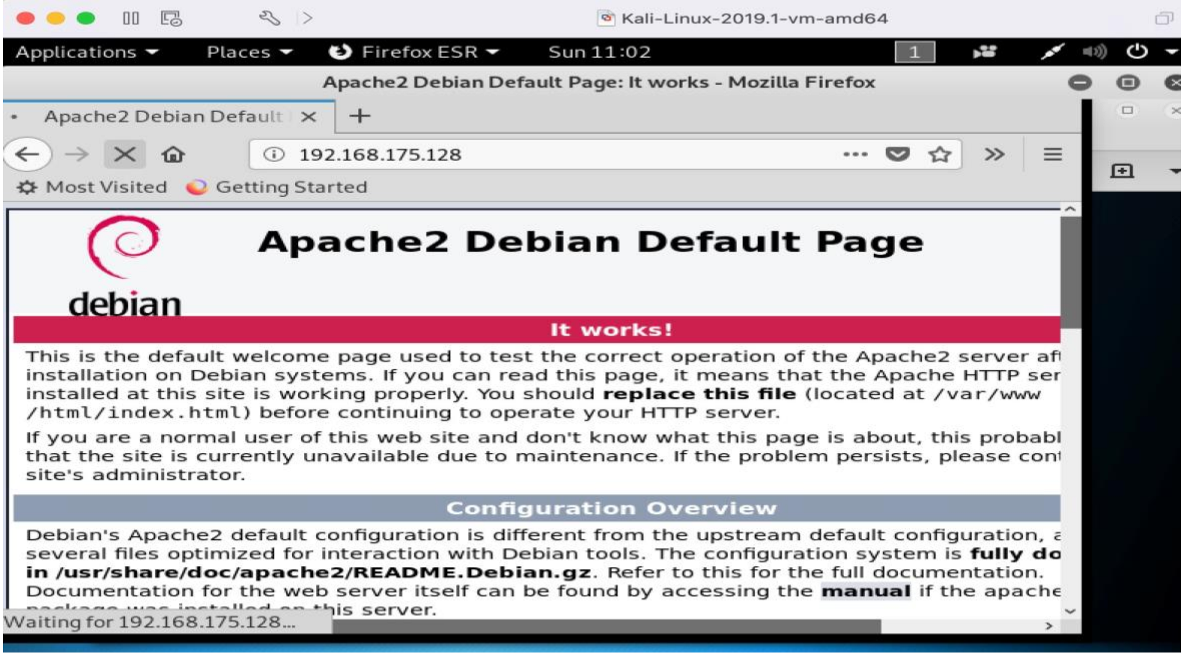
Now we run the tool using this command ‘python3 slowloris.py 192.168.175.128 -s 500’. Now the tool will start attacking the IP address mentioned.


Kali-Linux-2019.1-vm-amd64
Applications
Places
Terminal
Sun 10:53
root@kali-2019: ~/Desktop/Slowloris/slowloris
File Edit View Search Terminal Tabs Help
root@kali-2019: ~/Desktop/Slowloris/slo...
root@kali-2019: ~/Desktop/Slowloris

```
root@kali-2019:~/Desktop# ls
mount-shared-folders  restart-vm-tools  Slowloris
root@kali-2019:~/Desktop# cd Slowloris
root@kali-2019:~/Desktop/Slowloris# ls
slowloris
root@kali-2019:~/Desktop/Slowloris# ls -l
total 4
drwxr-xr-x 3 root root 4096 Feb 11 10:47 slowloris
root@kali-2019:~/Desktop/Slowloris# cd slowloris
root@kali-2019:~/Desktop/Slowloris/slowloris# ls -l
total 24
-rw-r--r-- 1 root root 1065 Feb 11 10:47 LICENSE
-rw-r--r-- 1 root root   26 Feb 11 10:47 MANIFEST.in
-rw-r--r-- 1 root root 2551 Feb 11 10:47 README.md
-rw-r--r-- 1 root root  438 Feb 11 10:47 setup.py
-rwxr-xr-x 1 root root 8047 Feb 11 10:47 slowloris.py
root@kali-2019:~/Desktop/Slowloris/slowloris# python3 slowloris.py 192.168.175.1
28 -s 500
[11-02-2024 10:52:42] Attacking 192.168.175.128 with 500 sockets.
[11-02-2024 10:52:42] Creating sockets...
[11-02-2024 10:52:55] Sending keep-alive headers...
[11-02-2024 10:52:55] Socket count: 279
[11-02-2024 10:52:55] Creating 221 new sockets...
```

Image 8: The tool starts attacking the IP address

We can see that the tool is working because it says wating for 192.168.175.128 and the page is reloading.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server af installation on Debian systems. If you can read this page, it means that the Apache HTTP ser installed at this site is working properly. You should **replace this file** (located at /var/www /html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probabl that the site is currently unavailable due to maintenance. If the problem persists, please cont site's administrator.

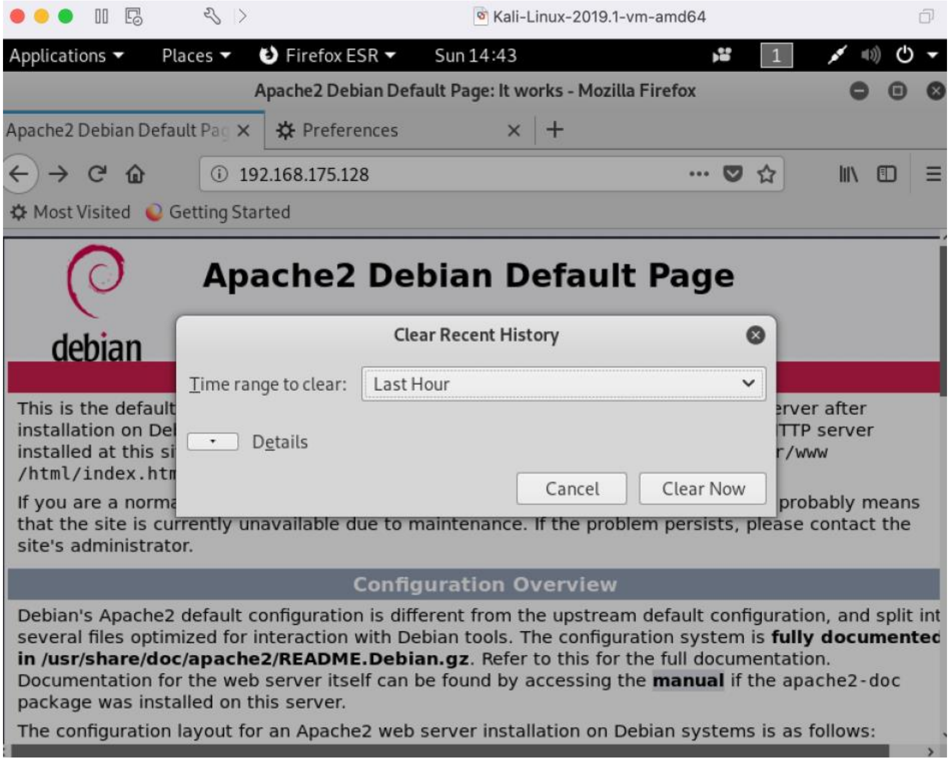
Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, a several files optimized for interaction with Debian tools. The configuration system is **fully do in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache package was installed on this server.

Waiting for 192.168.175.128...

Image 9: The Apache page is reloading

We will clear the cache and try accessing the page in a new tab as seen in the below screenshots.



Apache2 Debian Default Page

Time range to clear: Last Hour

This is the default ... erver after installation on De... TTP server installed at this si... r/www /html/index.htm...

If you are a norma... probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

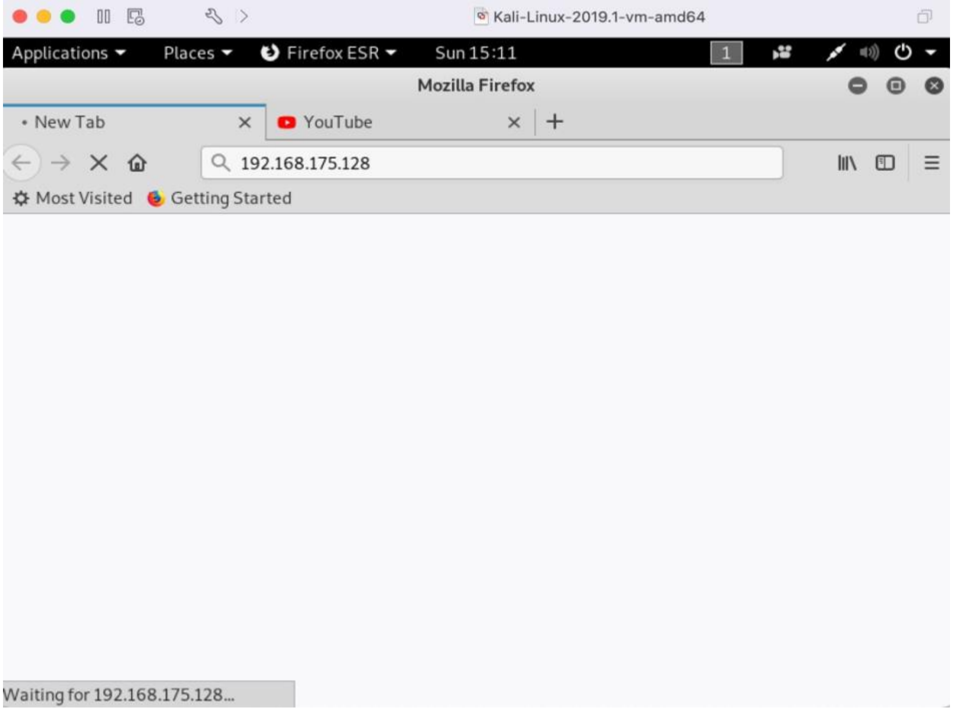
Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split int several files optimized for interaction with Debian tools. The configuration system is **fully documentec in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

Image 10: Clear recent History

Even though the Apache web server appears to be loading and waiting in one-tab, other services like YouTube may still function normally in another tab. This is because Slowloris consumes minimal bandwidth.



192.168.175.128

Waiting for 192.168.175.128...

Image 11: Apache server still loading

Slowloris operates with minimal bandwidth requirements, making it effective in initiating attacks without the need for substantial traffic volume. Its effectiveness is particularly notable against specific versions of Apache web server software, notably Apache 1.x and 2.x, where it can overwhelm and significantly slow down the targeted servers. Due to this capability, Slowloris is favoured by attackers seeking to disrupt servers running these versions of Apache. Other web servers like nginx are not susceptible to Slowloris because they do not allocate a thread per connection. Instead, they utilize a worker thread pool system where each thread has its own task queue.

It's challenging for a firewall or other defence mechanisms to detect Slowloris attacks because the HTTP requests it sends appear valid. Typically, when Apache receives a new connection with an HTTP request, it sets up a new thread to handle that request, which is terminated once the request is processed. However, if a connection remains open longer than expected, it can exhaust Apache's connection limit. Slowloris exploits this by opening numerous connections, seizing control of all available threads. As a result, even though the Apache web server appears to be loading and waiting in one-tab, other services like YouTube may still function normally in another tab. Additionally, Slowloris consumes minimal bandwidth, making it difficult to detect based on traffic volume alone.

How to mitigate slowloris attack :

- **Keep Web Server Software Updated:** Regularly updating your web server software is crucial to defend against Slowloris attacks. Some servers, like Apache 2.2.15 and above, come with built-in protection modules such as 'mod_reqtimeout' specifically designed to mitigate Slowloris attacks.
- **Set Connection Time Limits:** Configure your server to limit the maximum duration a client can keep a connection open without fully transmitting a request. This helps prevent Slowloris from holding connections indefinitely, especially if the server allows flexibility in connection time based on client behaviour.
- **Monitor and Manage Connection Pool:** Maintaining visibility of the connection pool can help detect abnormal behaviour indicative of a Slowloris attack. If an unusually large number of connections remain open for an extended period, it may signal a Slowloris attack. Manually freeing up suspiciously slow connections can halt the attack.
- **Implement IP Address-Based Rate Limiting:** Utilize rate limiting based on IP addresses to prevent a single IP from monopolizing all available connections.
- **Employ Intrusion Detection Systems (IDS):** IDS systems are effective in detecting unusual traffic patterns and can identify potential Slowloris attacks. Some IDS systems can automatically take corrective actions upon detecting an attack.
- Regularly update your software and systems with the latest **security patches** to prevent attackers from taking advantage of known vulnerabilities.
- **Load balancers** play a crucial role in distributing network traffic across multiple servers, thereby mitigating the impact of a Slowloris attack.
- Deploying a DDoS protection service is strongly advised to enhance your security posture. Such a service can effectively intercept and block malicious traffic aimed at your website, ensuring uninterrupted access for legitimate users

It's essential to note that while each method offers protection, employing a combination of techniques provides the most robust defence against Slowloris attacks.

Criteria 3 : Analysis

***Impact, Challenge, and Consideration**:*

Impact on Defenders:

- Slowloris poses a significant challenge to defenders due to its ability to target specific versions of Apache web servers effectively. The attack method consumes minimal bandwidth, making it difficult to detect based solely on traffic volume. This can lead to prolonged downtime for affected servers, impacting the availability of services and potentially causing financial losses or reputational damage for organizations relying on these servers.

Impact on Attackers:

- Slowloris provides attackers with a relatively simple yet potent tool to disrupt targeted web servers. By exploiting vulnerabilities in the way Apache handles HTTP connections, attackers can cause significant slowdowns or outages without the need for substantial resources. However, the effectiveness of Slowloris is limited to certain versions of Apache, restricting its applicability against other server software like nginx.

Challenges for Defenders:

- Detecting Slowloris attacks poses a challenge for defenders due to the stealthy nature of the attack. The legitimate-looking HTTP requests sent by Slowloris make it challenging to distinguish malicious traffic from normal requests. Additionally, defending against Slowloris requires specific countermeasures tailored to mitigate the attack's unique characteristics, such as setting connection time limits or monitoring connection pools for abnormal behavior.

Challenges for Attackers:

- Coordinating and executing Slowloris attacks requires meticulous planning. Despite the tool's simplicity, identifying vulnerable versions of Apache servers and gaining access to target systems can pose challenges.

Considerations for Defenders:

- Defenders must prioritize keeping web server software updated to leverage built-in protection mechanisms against Slowloris attacks. Implementing IP address-based rate limiting and intrusion detection systems (IDS) can help detect and mitigate ongoing attacks.

Considerations for Attackers:

- Attackers must weigh the potential consequences of launching Slowloris attacks, considering both the legal and ethical implications. Unauthorized access to computer systems or disruption of online services can lead to severe legal penalties and damage to the attacker's reputation.

The scenario described in criteria 2 illustrates how attackers can exploit tools such as Slowloris to undermine the accessibility of web services, thereby showcasing the tangible repercussions of DDoS attacks on the online operations of organizations. Through detailed step-by-step instructions on installing and executing the Slowloris tool, the scenario underscores the ease with which such attack techniques can be accessed and utilized by malicious actors.

Based on the information and screenshots provided under criteria two, we can outline metrics that specify a win for the attacker's tool, Slowloris, as follows:

1. **Successful Disruption:** Slowloris achieves a win if it successfully disrupts the availability of the target web service, causing significant slowdowns or outages that impact the organization's online operations.
2. **Extended Downtime:** A win for Slowloris is indicated by prolonging the downtime of the target web service, leading to extended periods of unavailability for legitimate users, thereby causing disruption and potential financial losses for the organization.
3. **Stealthy Operation:** Slowloris achieves a win if it operates stealthily, making it difficult for defenders to detect and mitigate the attack promptly. Its ability to generate legitimate-looking HTTP requests while conducting the attack enhances its effectiveness in evading detection.
4. **Efficiency in Resource Utilization:** Slowloris wins if it can cause significant disruption to the target web server while consuming minimal bandwidth and resources. This efficiency makes it challenging for defenders to identify and mitigate the attack.

All the above points were achieved in criteria 2 using Kali Linux, which signifies a win for the attacker’s tool. All the screenshots mentioned of kali Linux are of my own, taken on my machine.

Criteria 4 : Evaluation

The evolution of DDoS attacks has been propelled by various factors, including the proliferation of IoT devices and advancements in network infrastructure. Initially, DDoS attacks primarily relied on compromised computers and servers to generate fake traffic. However, the rise of the Internet of Things (IoT) introduced a new dimension to these attacks. IoT devices, notorious for their poor security measures, became prime targets for hackers to hijack and use in creating massive botnets. This shift significantly amplified the scale and impact of DDoS attacks, as botnets comprising IoT devices could generate floods of traffic ranging from gigabytes to terabytes.

Furthermore, advancements in network infrastructure, such as the advent of 5G technology, inadvertently bolstered the effectiveness of DDoS attacks. The introduction of 5G networks, characterized by increased bandwidth and extremely low latency, provided hackers with an ideal platform to orchestrate large-scale DDoS floods. The combination of IoT botnets and high-speed networks enabled attackers to pose significant threats even to tech giants like Google and Amazon.

The scenario outlined in the project aligns with this evolution, showcasing instances of massive DDoS attacks and highlighting the use of IoT botnets and advanced network technologies by attackers.

Looking ahead, the future of DDoS attacks appears increasingly ominous. The proliferation of IoT devices and the widespread adoption of 5G technology are poised to fuel further escalation in the frequency and magnitude of these attacks. The COVID-19 pandemic has accelerated the digitization of businesses, leading to a larger attack surface as more assets are moved online. This trend, combined with the expanding arsenal of hackers, paints a grim picture for cybersecurity in the years to come.

In light of these developments, it becomes increasingly crucial for businesses to adopt a proactive approach to cybersecurity, leveraging comprehensive defence strategies that encompass threat intelligence, robust mitigation measures, and continuous monitoring. Additionally, collaboration among stakeholders, including governments, industry players, and cybersecurity experts, will be essential in combating the escalating threat landscape posed by DDoS attacks.

Defending against DDoS attacks remains a formidable challenge. In the ever-evolving landscape of cybersecurity, defenders stand as guardians of digital fortresses, armed with a strategic approach rooted in the Essential Eight framework. They prioritize proactive measures, foremost among them being the vigilant patching of applications and operating systems to preemptively address known vulnerabilities. Complementing this, the implementation of multifactor authentication and the careful restriction of administrative privileges form formidable barriers against unauthorized access attempts, fortifying critical entry points. To further fortify defences, meticulous application control and the prudent limitation of Microsoft Office macros are employed, effectively narrowing the attack surface. Through rigorous user application hardening we can ensure a robust defence posture. Meanwhile, the cornerstone of resilience lies in the adoption of regular backup protocols, safeguarding data integrity and offering a lifeline for swift recovery should the adversary breach defences.

Conclusion:

In conclusion, the examination of Denial of Service (DoS) attacks, particularly through the lens of the Slowloris tool, underscores the persistent and evolving threat landscape faced by organizations in the realm of cybersecurity. It sheds light on the historical context of DoS attacks, showcasing their trajectory from early incidents like the Robert Morris worm to contemporary large-scale assaults targeting major tech entities. Through detailed analysis and demonstration of attack tools like Slowloris, the project highlights the challenges faced by defenders in mitigating sophisticated threats that exploit vulnerabilities in network protocols and infrastructure.

Furthermore, the project underscores the need for proactive cybersecurity measures, emphasizing the importance of regular software updates, implementation of intrusion detection systems, and collaboration among stakeholders to counter the growing menace of DoS attacks. As technology continues to advance and the digital landscape expands, the imperative for robust cybersecurity strategies becomes increasingly evident. By staying vigilant, leveraging cutting-edge defense mechanisms, and fostering collaboration, organizations can better protect their digital assets and mitigate the impact of DoS attacks in an ever-evolving threat landscape.