HTTP Slow Header Attack

Cryptography and Network Security

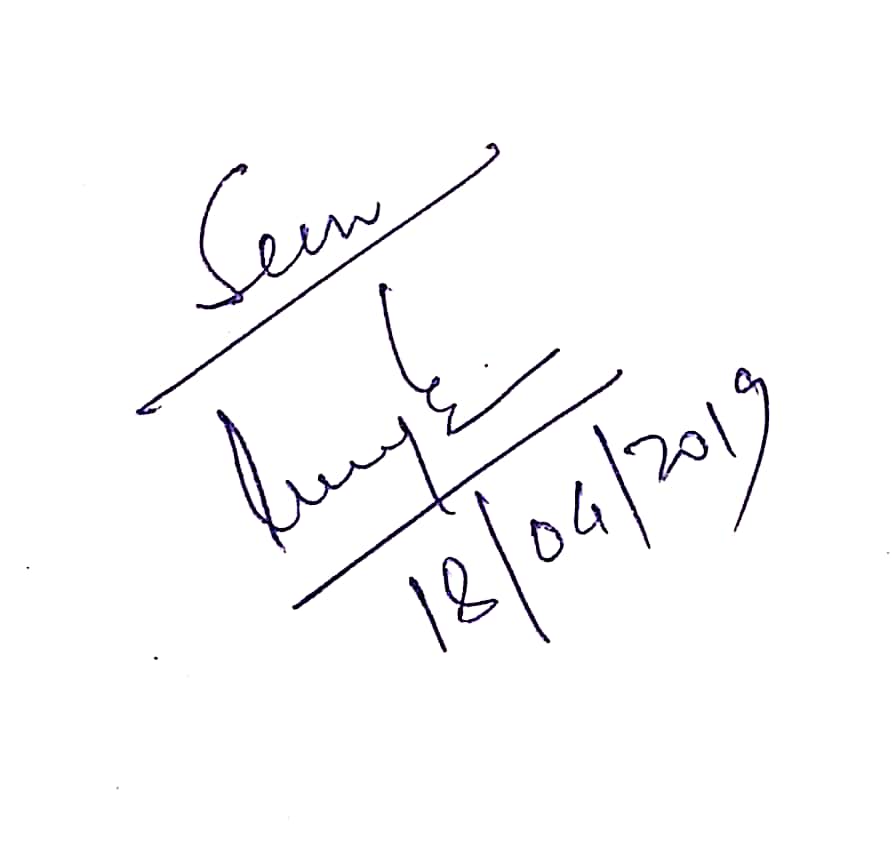
ECSE352L

End Semester Project

Team Members

Arun K L S – E16CSE016

Kamal Sai Raj K – E16CSE028

****Hidayatullah Pula – E16CSE032

**Abstract**

Denial of Service (DoS) attacks evolved and consolidated as severe security threats to network servers, not only for Internet Service Providers but also for governments. Earlier DoS attacks involved high-bandwidth flood-based approaches exploiting vulnerabilities of networking and transport protocol layers. Distributed DoS attacks have been introduced amplifying not only the overall attack bandwidth but also the attack source, thus eluding simple counter measures based on source filtering. Current low bit-rate approaches, instead, exploit vulnerabilities of application layer protocols to accomplish DoS or DDoS attacks. Slow DoS Attacks like, e.g., slowloris are particularly dangerous because they can bring down a well equipped server using small attacker’s bandwidth, hence they can effectively run on low performance hosts, such as routers, game consoles, or mobile phones.

**Keywords**

**Slow Loris**

**HTTP**

**IP**

**DoS - Denial of Service**

**Scapy**

**Apache Web Server**

**Introduction**

Slowloris is an application layer attack which operates by utilizing partial HTTP requests. The attack functions by opening connections to a targeted Web server and then keeping those connections open as long as it can.

Slowloris is not a category of attack but is instead a specific attack tool designed to allow a single machine to take down a server without using a lot of bandwidth. Unlike bandwidth-consuming reflection-based [DDoS attacks](https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/) such as [NTP amplification](https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/), this type of attack uses a low amount of bandwidth, and instead aims to use up server resources with requests that seem slower than normal but otherwise mimic regular traffic. It falls in the category of attacks known as [“low and slow” attacks](https://www.cloudflare.com/learning/ddos/ddos-low-and-slow-attack/). The targeted server will only has so many threads available to handle concurrent connections. Each server thread will attempt to stay alive while waiting for the slow request to complete, which never occurs. When the server’s maximum possible connections has been exceeded, each additional connection will not be answered and denial-of-service will occur.

#### A Slowloris attack occurs in 4 steps:

1. The attacker first opens multiple connections to the targeted server by sending multiple partial HTTP request headers.
2. The target opens a thread for each incoming request, with the intent of closing the thread once the connection is completed. In order to be efficient, if a connection takes too long, the server will timeout the exceedingly long connection, freeing the thread up for the next request.
3. To prevent the target from timing out the connections, the attacker periodically sends partial request headers to the target in order to keep the request alive.
4. The targeted server is never able to release any of the open partial connections while waiting for the termination of the request. Once all available threads are in use, the server will be unable to respond to additional requests made from regular traffic, resulting in denial-of-service.

Traditional DDOS attack tools and methods target to consume the system resources by opening too much TCP connections to the server. However SLOWLORIS is not a TCP DOS attack tool, but a http DOS attack tool.

Slowloris works by making partial http connections to the host(but the TCP connections made by slowloris during the attack is a full connection which is a legitimate tcp connection.)

If undetected or unmitigated, Slowloris attacks can also last for long periods of time. When attacked sockets time out, Slowloris simply reinitiates the connections, continuing to max out the web server until mitigated.

Designed for stealth as well as efficacy, Slowloris can be modified to send different host headers in the event that a virtual host is targeted, and logs are stored separately for each virtual host.

More importantly, in the course of an attack, Slowloris can be set to suppress log file creation. This means the attack can catch unmonitored servers off-guard, without any red flags appearing in log file entries.

A *Slow HTTP DoS Attack* takes advantage of a vulnerability in thread-based web servers which wait for entire HTTP headers to be received before releasing the connection. While some thread-based servers such as Apache make use of a *timeout* to wait for incomplete HTTP requests, the *timeout*, which is set to 300 seconds by default, is re-set as soon as the client sends additional data.

To make matters worse, a Slow HTTP DoS attack is not commonly detected by *Intrusion Detection Systems* (IDS) since the attack does not contain any malformed requests. The HTTP request will seem legitimate to the IDS and will pass it onto the web server.

**Motivation**

When planning to launch an attack, the attackers look for potential gaps in the security protocol through which it can access the service. When it comes to thread-based web servers, there is an inherent fault, i.e. a connection is only released once it receives entire HTTP headers. This creates a vulnerability which attackers have been exploiting through slow DOS attacks. Apache servers have a timeout in place but even that is reset once additional data is sent over the network. This invalidates the purpose of the timeout, which is to allow time for the HTTP headers to be completed.

In this scenario, an attacker can penetrate the system by launching an HTTP request. However, the request will not be closed. This gives the attacker the chance to create multiple connections on the same server, as the server continues receiving bogus data from the attacker during the timeout period. Also, the HTTP connection stays open, allowing the attacker to occupy each and every connection available on that particular web server. This means that if any genuine user wants to access the server, they won’t be able to.

This is why this type is called a ‘Denial of Service’ attack, as the service is denied to real users. The attacker can work this attack using a low bandwidth, but still be able to limit access to the server for anyone else. This method of denying access is what sets Slow DOS attacks from other types of DOS attacks.

A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) of the service or resource they expected.

As mentioned, genuine users of a web server are denied service once an attack has been launched and all the HTTP connections occupied. But this is just one of the ways Slow DOS attacks impact web servers. The more significant aspect of this is the loss of revenue and negative impact on the bottom-line. Clients are unlikely to trust you if they are unable to access the server and retrieve the data they need. This means the profitability of your business will take a turn for the worse.

Victims of DoS attacks often target web servers of high-profile organizations such as banking, commerce, and media companies, or government and trade organizations. Though DoS attacks do not typically result in the theft or loss of significant information or other assets, they can cost the victim a great deal of time and money to handle. There are two general methods of DoS attacks: flooding services or crashing services. Flood attacks occur when the system receives too much traffic for the server to buffer, causing them to slow down and eventually stop.

This is precisely what happened in 2009 when people protesting the Iranian presidential election used Slowloris to launch DoS attacks against websites run by that country's government. Rather than launch a DDoS attack that would consume bandwidth that would affect Internet access for a broad group of people, the protestors used Slowloris to narrowly attack specific government websites. The attacks had a high impact but a relatively low bandwidth rate; i.e., no collateral damage on unrelated services.

Not to mention, these types of attacks are typically hard to detect, particularly over DDoS and DoS. It is quite possible that the attack is discovered some time after the damage is done. Thwarting the attack and reclaiming your connections will mean significant downtime for your website, which results in further loss of revenue, and again, the client experience will be affected. This means loss of business and loss of customers in the future. So, as you can see, it is important for you to keep an eye out for slow DOS attacks.

**Project Details**

There are 8 types of HTTP requests. In the slow loris attack, the malicious client sends a GET request to the server and periodically sends keep alive headers to keep the connection open.

A basic GET request looks as follows

GET /hello.htm HTTP/1.1

User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)

Host: www.tutorialspoint.com

Accept-Language: en-us

Accept-Encoding: gzip, deflate

Connection: Keep-Alive

Since slow loris is a slow header attack, it sends the header part of the GET request.

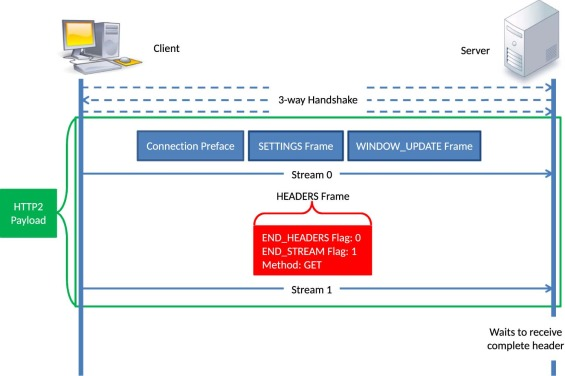


Fig 1. Depicting Slow Loris Attack

Attack was launched using 150 sockets, each opening a connection and keeping it alive using keep alive headers. Server was unable to service new connection request; i.e website hosted in the server was inaccessible in the duration of the attack

**Experimental Details**

**OPERATING SYSTEMS USED:**

● Windows 10 (SERVER)

● Ubuntu 16 (Malicious Client)

● Windows 10 (Regular Client)

**Server:**

An Apache 2.4 server was setup on Windows 10 system. The interface used for IPv4 addresses is NIC card of the system. http.conf file is configured according to suitable conditions for the project.

**Client:**

An ubuntu16 system was setup as Malicious client and Windows 10 system was setup as Regular client which was used to check server status (whether the attack was working).

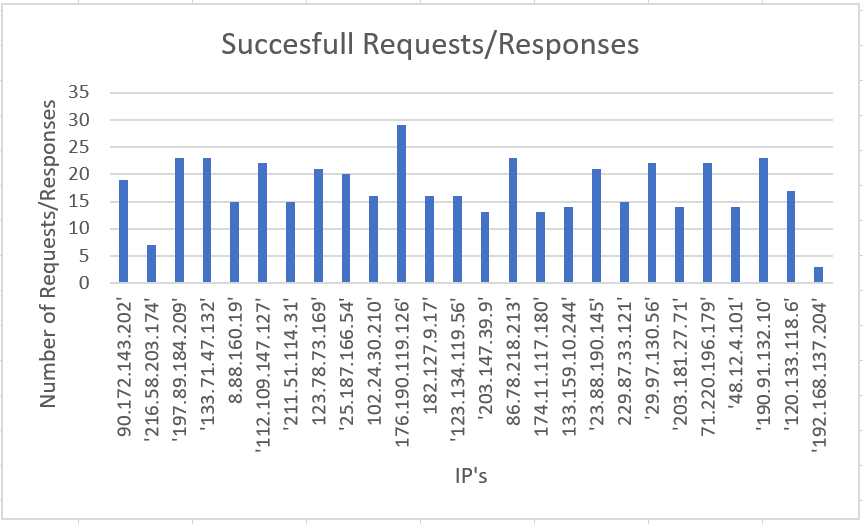
An ad hoc network was setup on the server system so as to provide a direct connection to which both the clients were connected.

**STEPS FOR SLOW LORIS ATTACK:**

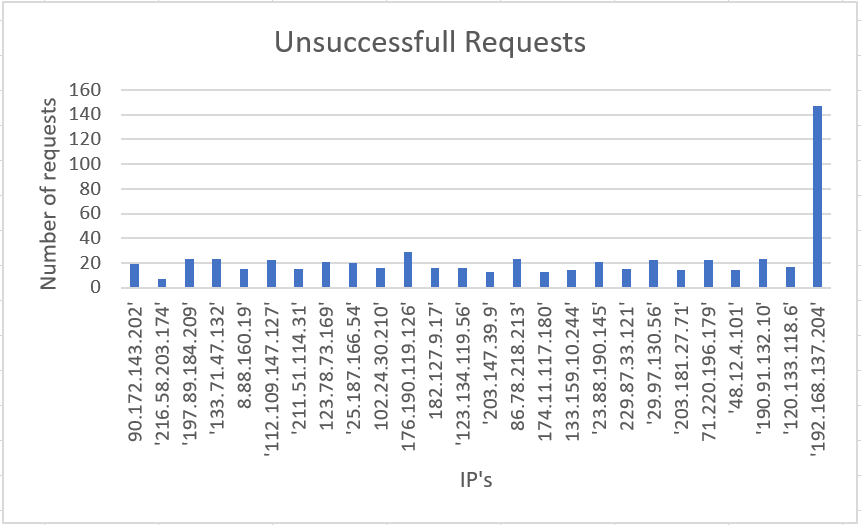
1. Initializing Server: Enable Apache2.4 on Server system.
2. Setting up of ad hoc network on server system and connecting by wifi on client systems
3. Run Python slow loris script on Malicious client, which opens multiple sockets and sends GET connection headers, periodically sending keep alive headers to keep the connection open
4. Attack will slowly consume server’s resources (DoS attack). Eventually server will be unable to service any more requests. Regular client connected to the server will not be able to access the website hosted on the Apache server
5. To check whether the attack is working reload the website on regular client in regular intervals until the page does not load any more indicating attack has been successful.

**STEPS FOR MITIGATION OF SLOW LORIS ATTACK**

1. Packets received by the server can be accessed using tcpdump, which stores these packets into pcap files
2. Pcap files are analyzed using scapy. Malicious clients generally have low success rate (successful GET requests/total requests) and they have high number of unsuccessful packets as compared to regular clients over a period of time.
3. For regular packets, complete requests were sent from regular client with ip spoofing showing 25 clients.
4. Plotting graphs:



Graph 1: Successful requests



Graph 2: Unsuccessful Requests

1. From the above graphs, it is evident that the client with IP 192.168.137.204 is the malicious client and it can be blacklisted

**Conclusion**

We have successfully implemented a Http slow header attack on server and when a genuine user comes into network for webpage from the server the user cannot access the webpage. This protocol is vulnerable to a Denial of Service (DoS) attack popularly known as classic Slow Loris attack, the attacks were designed and generated in a real network setup. We have learnt a lot of things from this project like server implementation and deployment, network establishment and how to use Linux to successfully launch attacks.

**Limitation**

We couldn’t successfully implement this on multiple systems as we had lack of computers and used IP spoofing for achieving results. We used socket programming for attack which isn’t much flexible. Also since this is a slow header attack, this attack does not bring down the server but rather keeps the connections open and the server busy. When the attack is happening other users who come for connection into the network they aren’t able to load the page but when the attack is stopped the client is redirected to the webpage.

**Future Work**

We would like to implement this attack in a much larger scale which has more computers (i.e clients and attackers) and a robust server. We would like to try this out on many other servers which are available in public for free. Also we would like to implement a prevention measure that prevents the attack from being happening and this can be achieved by using programs like SNORT.

**References**

# SLOWLORIS: HTTP DOS(Denial Of Service)attack and prevention

<https://www.slashroot.in/slowloris-http-dosdenial-serviceattack-and-prevention>

# [How To Mitigate Slow HTTP DoS Attacks in Apache HTTP Server](https://www.acunetix.com/blog/articles/slow-http-dos-attacks-mitigate-apache-http-server/)

<https://www.acunetix.com/blog/articles/slow-http-dos-attacks-mitigate-apache-http-server/>

# Low and Slow Attack

<https://www.cloudflare.com/learning/ddos/ddos-low-and-slow-attack/>

SLOW LORIS HTTP DDoS

<https://web.archive.org/web/20150426090206/http://ha.ckers.org/slowloris>