HTTP Desync Attacks: Smashing into the Cell Next Door

Shashank Kanakapura Srivatsa
University of Paderborn
Paderborn, Germany
srivatsa@mail.uni-paderborn.de

Seminar report – May 15, 2020.
System Security Group.

Supervisors: **Prof. Dr-Ing. Juraj Somorovsky**
University of Paderborn
Paderborn, Germany
juraj.somorovsky@upb.de

Abstract

Internet has become an integral part of the modern world and it is extremely difficult to imagine a world without internet. One of the most significant activity on the internet is exchange of data between computers, generally referred to as Clients and Servers. There are a huge number of formats in which the data can be transferred over internet and many new formats are being introduced. To organize and govern such data transfers, there are several protocols defined by reputed researchers and organisations. One such prominent protocol is Hyper Text Transfer Protocol(HTTP), where a Client submits a request to the Server over the internet and the Server responds to the Client with approriate data or message.
Like several other protocols, HTTP also contains some vulnerabilities which can be exploited by malicious attackers and thus it becomes extremely necessary to identify such vulnerabilities. Security experts or *White hat hackers* help to identify vulnerabilities present in protocols such as HTTP and notify the concerning organisations or individuals about the same. This helps to fix the vulnerabilities present and thus prevent information leakage. There are several vulnerabilities that have been identified in the past and many of these problems are already addressed. In this report, we talk about a relatively new technique known as 'HTTP Desync Attack' which is developed to exploit the vulnerability in HTTP protocol. In this technique, the attacker tricks the Server into believing that the malicious request is actually a part of a normal user‘s request and thus allows the attacker to gain control over the request. This report reviews HTTP Desync Attacks in detail and further evaluates it to understand the pros and cons of the technique.

1 Introduction

As the world around us is developing rapidly, many things are moving online. As organisations and individuals are increasingly preferring revolutionary inventions such as Cloud Storage, IoT devices, Online Banking etc., these technologies require one basic thing - Internet. Internet has become an integral part of our daily lives. In its early years, internet was only used for crucial transactions inside corporate and government organisations and for mail exchanges. However, over the last two decades it has completely taken over the lives of a common individual. Majority of the world's population depends on internet for day-to-day activities.

With such advancements, the need for internet security is more than ever and organisations are investing a substantial part their revenue on security [2]. Even though we try to reduce security related incidents, attackers are inventing clever ways to bypass exsiting security protocols. According to a reputed corporate organisation, there has been a 11% increase in security breaches since 2018 and 67% since 2014 [3]. It is estimated that hackers attack every 39 seconds, on average 2,244 times a day [4]. If an attacker is able to control or snoop on the data being exchanged over internet, it can be used for malicious purposes. It can contain personal information of a person or an organisation, confidential data of goverments and organisations, financial data and so on.

Internet security protocols and standards help in tackling the above stated problems. Even though attackers have found ways to exploit the vulnerabilities in the existing protocols governing the internet, experts and researchers are constantly improving these protocols to better equip the world in tackling this menace.

Some of the popular security protocols are Secure File Transfer Protocol (SFTP), HyperText Transfer Protocol (HTTP), Secure Socket Layer (SSL) and so on. In this seminar we consider HTTP. HTTP is a protocol mainly used for data exchanges between computer systems. It is usually used in a Client-Server environment, where the Client sends a request to the Server and the Server responds with data, resource or message. A request from the Client or a response from the Server can contain sensitive information such as credentials, financial data, confidential files belonging to an organisation or a government, personal information and so on. If an attacker intercepts these information it could put the victim at risk.

HTTP has been constantly improved and has become an important protocol to secure the data that is transmitted over the internet. However, as the growth of internet increases the need for securing these data transfers also increases with it.

Like any other security protocol, even HTTP has vulnerabilities which are being exploited by attackers. There are several techniques for such exploits. Here we consider a relatively new technique known as 'HTTP Desync Attack' with which the attacker can create havoc on the target systems. The basic ideas of this technique was first document by Watchfire in 2005 [5]. This technique was however recently brought to the limelight by the works of by James Kettle [6]. Using this technique, the attacker can introduce carefully crafted malicious request through the front-end. Consider a front-end application which communicates with a back-end server. Whenever a HTTP request is created, it includes HTTP Header and HTTP Body. The same are sent to the servers for synchronisation. In a real setup, there are streams of requests coming from a large set of users. Hence it is important for the server to know where a request starts and ends. These HTTP Headers help the server to identify these crucial points and thus help in synchronisation. The attacker can however utilize the information present in the HTTP Header to cleverly trick the server and desynchronise the system. The corressponding malicious data is thus treated as part of a legitimate user‘s request. This results in attacker gaining control of the victim user‘s request. We will elaborately discuss about the details of this technique in the further parts of thos report and will also look at some case studies.

The organisation of this report is as follows - In Chapter 2 we introduce some background concepts which might be required to understand the technique we discuss. Chapter 3 describes the problem and in Chapter 4 we elaborately discuss the details of this technique. In Chapater 5 we look at some real case studies to better understand the technique and Chapter 6 further evaluates the technique based on all the information discussed till then. We finally give a connclusion.

2 Background

2.1 HTTP

[1] Hyper Text Transfer Protocol(HTTP) is a stateless application layer protocol meant for distributed, collaborative, hypertext information systems. The exchange of data between a client and a server is governed by HTTP. A client sends a HTTP request which can contain standard as well as user's custom data together. The client receives this request and decodes the same. Based on the request the server decides to either perform an action or send data back to the client which can contain requested information or server messages.

2.2 HTTP 1.1

[1] HTTP 1.1 is an improvised version of HTTP which succeeds the version HTTP 1.0. This version includes several improvements, few of which are :

- support for keep-alive feature where a connection can be re-used.
- pipelining
- chunked responses

2.3 HTTP Request

[7]A HTTP Request is a message sent by the client, to the server, to invoke an action at the server side.

2.4 HTTP Response

[7]A HTTP Response is a message sent back by the server, to the client, which contains the requested information, resource and status code.

2.5 HTTP Request Methods

[8]These are verbs which specify the action that has to be performed by the server. The popular ones are :

- GET : To request data from the resource.
- POST : To submit information to a resource.
- PUT : To submit information to a resource but replaces the existing information.
- DELETE : Deletes the specified resource.

2.6 HTTP Headers

Headers contain meta-deta which corressponds to the request, request method and the information being communicated.

2.7 HTTP Body

It is the information being transmitted and the response message received.

3 Problem Statement

4 Approach

5 Case Studies

6 Evaluation

7 Conclusion

Bibliography

[1] https://tools.ietf.org/html/rfc7231

[2] https://www.gartner.com/en/newsroom/press-releases/2018-08-15-gartner-forecasts-worldwide-information-security-spending-to-exceed-124-billion-in-2019

[3] https://www.accenture.com/us-en/insights/security/cost-cybercrime-study

[4] https://eng.umd.edu/news/story/study-hackers-attack-every-39-seconds

[5] https://www.cgisecurity.com/lib/HTTP-Request-Smuggling.pdf

[6] https://i.blackhat.com/USA-19/Wednesday/us-19-Kettle-HTTP-Desync-Attacks-Smashing-Into-The-Cell-Next-Door-wp.pdf

[7] https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages

[8] https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods