Security Project Design Report

Student ID: **1605006** (Avijit Biswas)
Attack Tool: **Dictionary Attack and Known Password Attack**

Dictionary Attack and Known Password Attack

Introduction:

In crypt-analysis and computer security, a **dictionary attack** is a form of brute force attack technique for defeating a cipher or authentication mechanism by trying to determine its decryption key or password by trying hundreds of or sometimes millions of likely possibilities, such as words in a dictionary.Moreover, A **known password attack** refers to any of the various methods used to maliciously authenticate into password-protected accounts by trying to predict all passwords from popular known password list. These attacks are typically facilitated through the use of software that expedites cracking or guessing passwords.

In contrast to a brute force attack, where a large proportion of the key space is searched systematically, a dictionary and known password attack tries only those possibilities which are deemed most likely to succeed.These attacks often succeed because many people have a tendency to choose short passwords that are ordinary words or common passwords.

In an online dictionary and known password attack, an attacker stores all possible passwords in a file which is known as a dictionary and then targets any website to login to the victim's account. Attacker then sends http post request by taking all possible words from the dictionary. If the attack is successful, the target website will give a positive response.

Topology Diagram:

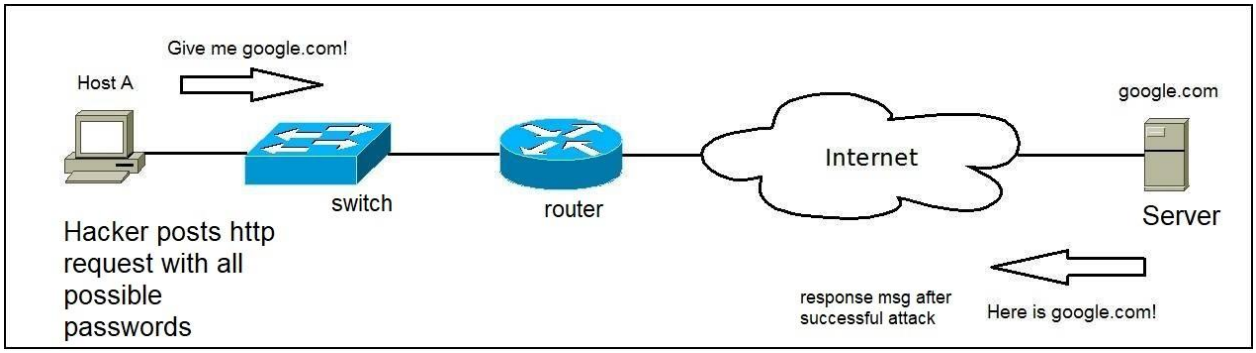

Figure: Topology Diagram of HTTP

Timing Diagram of HTTP Protocol:

HTTP(**HyperText Transfer Protocol**) is an application layer protocol designed within the framework of the Internet protocol suite. It works on top of a reliable transport protocol such as TCP and may sometimes use unreliable protocols such as UDP as well. HTTP functions as a request–response protocol in the client–server computing model. A web browser, for example, may be the client and an application running on a computer hosting a website may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

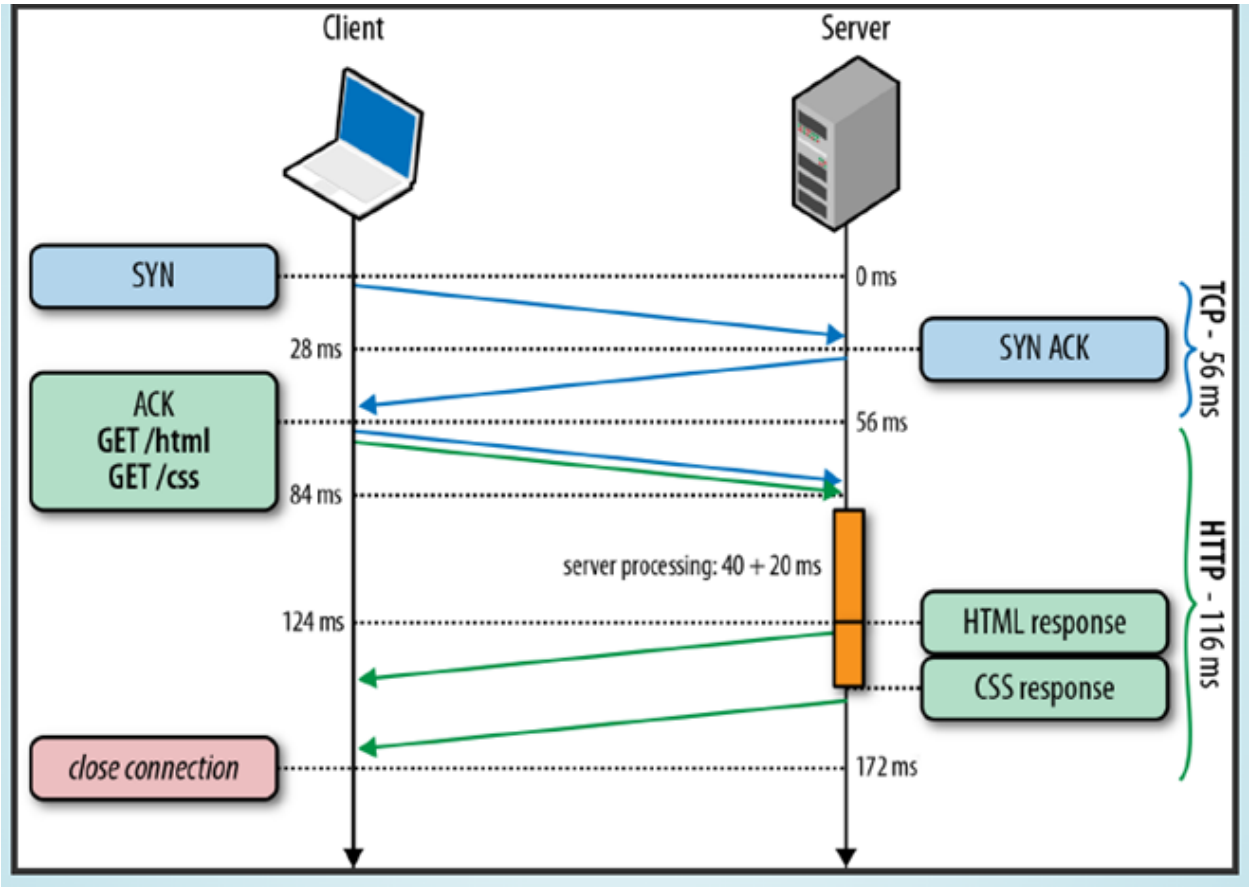

Figure: Timing Diagram of http protocol

The above diagram represents a **persistent** HTTP timing diagram where the connection is established once, then the connection is kept alive as long as the communication between the server and the client continues and then when the session is finally over, the connection is finally closed.

Attack Timing Diagram with Attack strategies:

At the time of attacking, the attacker plays the role of client in this protocol. Here we will not use any builtin http library to post requests. We will use low level language(cpp) by using sockets. At first a **TCP connection** is established by using the TCP socket in cpp in the linux environment and then we will continuously send **HTTP POST requests** from the attacker side using a script written by ourselves. Then we will read the dictionary and the known password list files and place all the passwords inside the request body of the POST request. So the server is fooled into believing that the request is coming from a client side website running on the browser, whereas the requests and the required messages along with the password in the body will be constructed by the attacker side script themselves. We see that upon receiving the wrong password in the first attempts leads the server to respond with an error status code and an error message in it's response message. Then when the correct password is attempted in the POST request we see that the server responds with a success message and a success status code (**201 OK**) to the attacker. After receiving that success status code and message, we will conclude that the attack is successful and then we will finally close the connection.

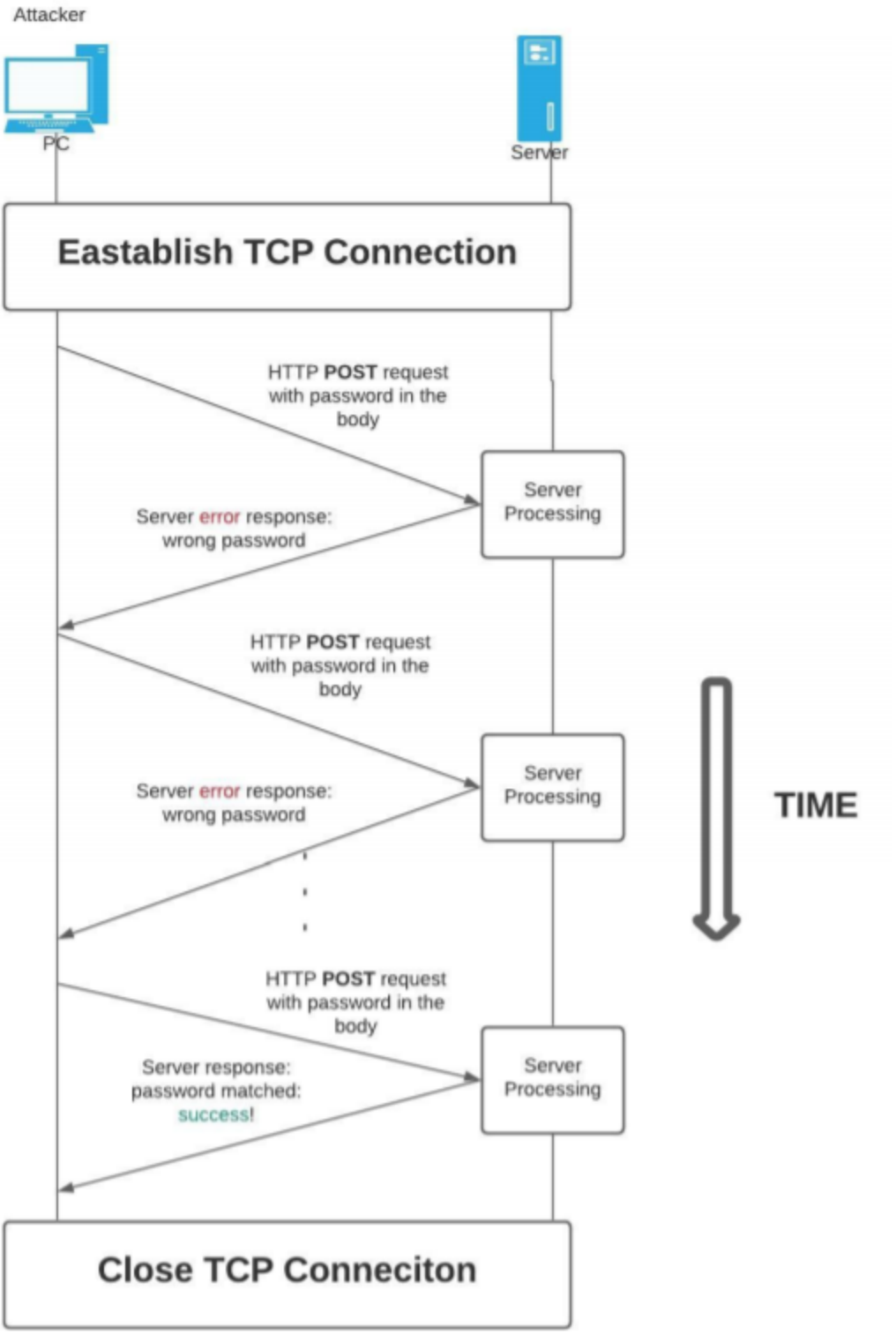

Figure:Timing Diagram of http protocol with respect of attacking

HTTP Packet Details:

HTTP requests are messages sent by the client to initiate an action on the server. HTTP request packet consists of three components known as starting **request line**, **request headers** and **request body**. Starting request line has 3 elements which are the http method(GET,POST,PATCH etc.),request target url and http version.
Request header comprises HostName,Content-Type,Content-Length etc.And finally the request body contains field names of the database and its corresponding values separated by & operator.

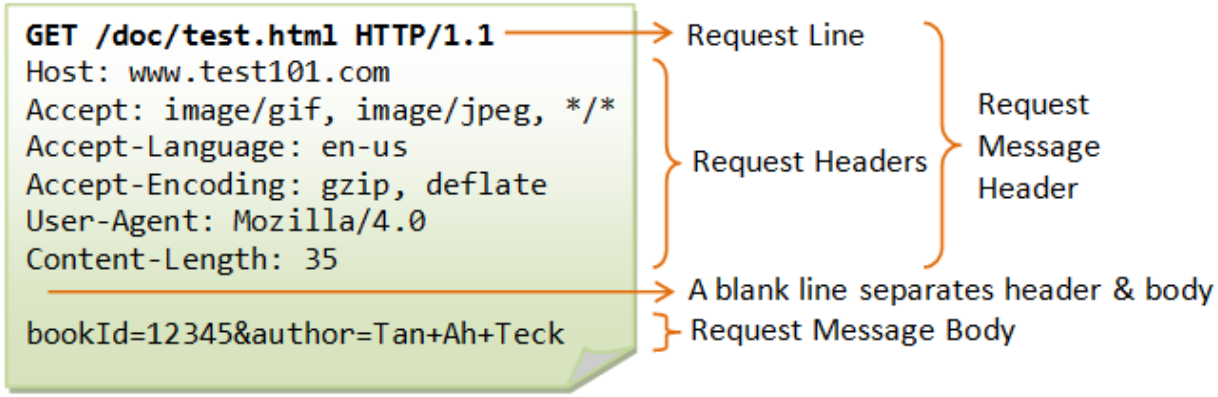

Figure: HTTP request

We use a POST request packet to demonstrate the attack.For our attack, we will use target url **http://localhost:3001/user/signIn** . In the header section, we will use localhost as hostName and Content-Length as a variable which is dynamic as length of various passwords is different from one to another. And in the request body section, we will give two fields which are username and password.Here, value of username attribute is the victim's username and value of password field is read from the dictionary and known password list file.

Justification:

Since I(attacker) post valid requests as a client's valid username using script, the website can not detect whether I am an authenticated user or not. So, if I know the username of the victim then I continuously try to login to the website by guessing all possible passwords from the dictionary.However, people generally give short popular passwords and that's why I think that my design of attack should work.