Lab 1 - Experimenting with Application Protocols

CS 330

Khumoyun Abdulpattoev

September 9, 2023

1. Introduction:

In Lab 1, I explored application protocols, where I encountered challenges and insights. I began with an attempt to retrieve web content using `nc`, leading me to experiment with the complexities of HTTP, redirects, and different methods. As I have worked through the lab, I explored the intricacies of SMTP, attempting to send emails via `nc` on iwu's server. Experimenting with it helped me understand how modern email configurations and encryption work (on a high level), highlighting the importance of secure communication. The last part of the lab involved working with Internet Relay Chat (IRC). I explored its history and experimented with various commands, gaining valuable insights into its functionalities. In this report, I am going to dive deeper into the experiments and new things I learnt throughout the lab.

2. Part A: Use nc to Retrieve a Web Page

2.1 Connecting to www.iwu.edu

To establish a connection with www.iwu.edu, I attempted to use the command `nc [host] [port num]`. However, I encountered an HTTP 301 error, indicating that the requested file had been permanently moved. As I found out, this error typically occurs when a URL has been redirected to a different location. This redirection could be due to various reasons, such as a change in the website's structure or URL.

2.2 Connecting to an HTTP server at www.example.com

When I requested the index HTML page from www.example.com using the command:

```
GET / HTTP/1.1
Host: www.example.com
```

I successfully received the HTML layout along with relevant CSS information from the home page. However, when I attempted the same request without including the "Host" header, I received an HTTP 400 Bad Request error. This error signifies that the server did not understand the request due to a missing header. In this case, the "Host" header is essential as it specifies the domain the client is requesting, allowing the server to route the request correctly.

When attempting to access a non-existent web page, the server gracefully redirected the request to the home page. This behavior is common and serves as a way to ensure that users are directed to a valid page even if they mistype a URL or access a non-existent resource.

As the next step, I used the HEAD operation to retrieve metadata about the home page, sending the following command:

```
HEAD / HTTP/1.1
Host: www.example.com
```

The server responded with a bunch of information, including details about the age of the page (222911), cache control settings, and the expiration date of the metadata. The HEAD operation is

a useful way to retrieve information about a resource without downloading the resource's content, making it efficient for obtaining metadata.

2.3 PUT and DELETE Commands

To delete a web page, I initiated an HTTP request using the DELETE method. However, the server responded with an HTTP 405 Method Not Allowed error. This error indicates that the server does not support the DELETE method for the specified resource.

The inclusion of PUT and DELETE commands in the HTTP protocol provides mechanisms for modifying and deleting resources on web servers. However, not all servers support these methods, and their usage is often restricted to specific resources or scenarios for security reasons. Understanding which methods, a server supports and the permissions associated with them is crucial when working with these commands.

3. Part B: Use netcat to Send an Email

3.1 Connecting to smtp.iwu.edu

To initiate an email sending process, I connected to the campus SMTP server, `smtp.iwu.edu`. While the connection process itself was explained, the report lacks details about the associated server and port numbers used. Adding these details would provide a more comprehensive explanation.

3.2 Sending an Email

Despite my attempts, I encountered a failure in sending the email, and I did not receive any response from the server. To diagnose the issue and gain a deeper understanding of the failure, it is essential to investigate each step of the SMTP transaction carefully.

```
nc smtp.iwu.edu 587
HELO test.iwu.edu
MAIL FROM: kabdulpa@iwu.edu
RCPT TO: kabdulpa@iwu.edu
DATA
Subject: Greetings!
From: kabdulpa@iwu.edu
To: kabdulpa@iwu.edu
Content-Type: text/plain

Hello, friend!
.
QUIT
```

3.3 Emails nowadays

Modern email configuration, exemplified by services like Google's Gmail, emphasizes robust security measures. It achieves this by employing encryption protocols such as Transport Layer

Security (TLS) or Secure Sockets Layer (SSL). These encryption methods encode the content of emails during transmission, protecting them from potential interception or eavesdropping. Additionally, specific port numbers, such as 587 for Outgoing Mail Server (SMTP) with TLS encryption and 465 for SSL encryption, are designated to ensure secure connections. In essence, modern email practices prioritize data protection, making email communication more secure in today's digital landscape.

3.4 Open Mail Relays

Open relay mail allows anyone to send an email through it without proper authentication of the user. So, it is extremely vulnerable to manipulation and potential malpractice as it basically invites anyone to use it. Iwu's smtp is not an open relay mail since it does not allow anyone without a proper authentication to access mail services. It is important that it is not an open relay given possible consequences.

4. Part C: Investigate and Experiment with IRC

4.1 Background of IRC

Internet Relay Chat was introduced in August of 1988 by Jarkko Oikarinen. It is a text-based system for instant messaging. It is used for communication between users within groups called "channels." It was one of first chat applications that got popular; in 1989, it had around 40 servers worldwide. An interesting fact I found was that IRC was the platform where Iraq's invasion of Kuwait in 1991 was reported during a media blackout(history-computer.com).

4.2 Cheat Sheet

Below are some of the common commands and their exact structure and purpose:Register a client connection:

Register: `USER <username> * * :<realname>`

`NICK <nickname>`

List the channels on the server: `LIST`

Join a channel: `JOIN #channel_name`

Send a message to all users on the channel:

```
PRIVMSG #channel_name :Greetings, fellow chatters!
```

Quit: `QUIT :<Message about quitting>`

4.3 Experiment

Once I logged in to IRC, I chose a random channel called #zsh. I executed some of the commands above, and they all worked (even though it was not possible to send messages in this channel without NickServ account, it did work fine in other channels). Here is the snapshot of the messages:

```
PRIVMSG #zsh : Hello, everyone!

:platinum.libera.chat 415 khumo #zsh :Cannot send message to channel
(+R) - you need to be logged into your NickServ account

QUIT:khumo!~khumo@198.178.132.247 QUIT :Client Quit

ERROR :Closing Link: 198.178.132.247 (Client Quit)
```

4.4 Understanding PING and PONG

The purpose of the PING and PONG messages in IRC is to keep the connection between the client and server "online" and detect any unresponsive or disconnected clients. The server periodically sends a PING message to the client, and the client must respond with a PONG message that echoes the PING message back to the server. Just like the name suggests, it is like playing a game of ping pong. This exchange ensures that the connection remains active and prevents it from timing out.

5. Conclusion:

Reflecting on Lab 1, I gained valuable insights into application protocols. I have learnt how to interact with the protocols, unexpected roles played by Internet Relay Chat (IRC) in reporting major events and underscored the prevalence of open relay mails in the past, revealing broader implications of internet technologies.

I spent approximately 3-4 hours completing this assignment. While the instructions were generally clear and the tasks comprehensible, I encountered some challenges, particularly in understanding SMTP server responses (or lack of them) during email transmission.

In summary, this assignment deepened my understanding of application protocols, leaving me with a desire to explore networking and communication further. It has provided a foundational understanding for future experiments into these domains and their relevance in today's evolving digital landscape.