Aoyan Sarkar

10/20/2023

Test 1

1.1 Describe in your own words how the web works! In as much detail as you can, describe **all** the sequences of events that take place from the time a user presses Enter on the keyboard after typing in [www.rpi.edu](http://www.rpi.edu) into the address bar to when the webpage is finished rendering in the browser. Specifically, tell me in great detail the **two protocols** we discussed in class in action. (8 points)

The World Wide Web relies on the TCP/IP protocol and the HTTP protocol to deliver a request from the client to the server and back to the client. Computers can only connect to IP addresses, but the requesting address is a domain, so the client has to first find the IP address that maps to the domain. A service called DNS is used, which has a record of domains and the IP addresses that are associated with the domain. The browser will first check if it has the domain in its cache, and if it does not, the browser will ask the operating system. If neither has the domain in cache, it will ask a DNS server, which can either have it cached or it will recursively resolve the domain. It will ask the root server for the IP for the .edu domain server, and then ask the .edu domain server for the IP of the rpi.edu domain server, which will finally ask for the [www.rpi.edu](http://www.rpi.edu) IP address. Once the IP address is found, the browser will open a TCP connection to the IP address for [www.rpi.edu](http://www.rpi.edu). A 3-way handshake is performed in order to establish the connection, and then the data is sent in the TCP packet. As a TCP packet has a maximum size, a request can be broken up into multiple packets, which are reassembled on the server. Once the server gets all the TCP packets and finishes reassembly, the HTTP/HTTPS protocol is now used to find out what resource the client wanted. If the HTTPS protocol was used, the client would have encrypted the request body, so the server would decrypt it, and then process it as HTTP. In this case, a browser would have sent a GET request to “/”. The server will send back headers and the response text, which in this case would tell the browser that the response is HTML text. The browser would wait for the entire body to arrive, and then parse and render the text and display the content to the user.

1.2 Explain what is meant by a Universal Interface in a REST API. (5 points)

A universal interface in a REST API means 4 things. The first is that every resource is identified in some way, such as by a URI in a web resource. This means that the resource that is returned to the client should be functionally similar but not the same as how the server has stored the resource. An example is reading a database to send back a JSON response for a resource, since the client is not getting the raw data from the database, but the JSON response is represented by a URI and contains the same information. The second is that the client can modify the resource using only the metadata and the representation that it receives. Following the earlier example, if a client got a JSON representation and the HTTP headers, it would be able to edit or delete the resource without having to fetch any additional information in a separate request. The third is that a client can decode the representation that it receives without needing any additional requests to learn how to parse it. For a JSON response, if the client is told that the response is JSON, it can decode the JSON into the client’s internal data structures and use the data, without having to make additional calls to the API. Finally, the server uses hypermedia as the engine of application state, which means that the initial response plus the initial metadata should contain URIs that lead to any other actions that can be done for the resource, as well as any related resources the client might want.

1.3 Explain how your browser chooses which CSS rule to apply to a tag in the case where there are multiple rules that could apply. (3 points)

CSS has three ways to determine how to apply conflicting rules – cascading, specificity, and inheritance. Inheritance means that rules that apply to parent elements will also apply to child elements. For example, making a paragraph have the color red will also make any emphasis children elements also colored red. The second method is specificity – meaning that a more specific selector will take precedence. In order of highest specificity is ID selectors, then class selectors, attribute selectors & pseudo-class selectors, and finally tag selectors. Nesting multiple selectors, such as “#id .class” or “#id.class” will make it more specific than one without the nested selector, such as just “#id”. Specificity is also used for the sources of styles, as styles inside of style attributes are more specific, then styles in style tags, and finally external stylesheets. The “!important” flag will raise the specificity of a rule but the same specificity algorithm is used for conflicting “!important” rules. The final method is cascading, which means that for two rules with the same specificity, the one defined last takes precedence. This is also used for external stylesheets, so whichever stylesheet is loaded last will take precedence if their rules have the same specificity.

1.4 What command would you use to change the ownership of a file or directory on a Unix machine (such as your Azure VM)? Show me a complete command invocation to make a directory named /var/www/html be owned by a user named callab5 and a group also named callab5. (4 points)

I would use the chown command. The full command is “sudo chown -R callab5:callab5 /var/www/html”.

2: <https://sarkaa3rpi169285.eastus.cloudapp.azure.com/itws2110-sarkaa3/quiz1/>

3:

An attack I learned of is the XSS vulnerability when displaying user input. This happens when not properly sanitizing user input, so a user can put in HTML tags like script and run JavaScript, which can send cookies off to the attacker when the server outputs the user’s input word for word. One way to combat this is to sanitize any input to get rid of any unwanted HTML tags. This way, an attacker cannot abuse HTML to insert script tags or any other method of executing JavaScript, which means that they cannot run malicious code. This is an easy fix as all user input should be sanitized, and there are many libraries for many programming languages that can do this for you.