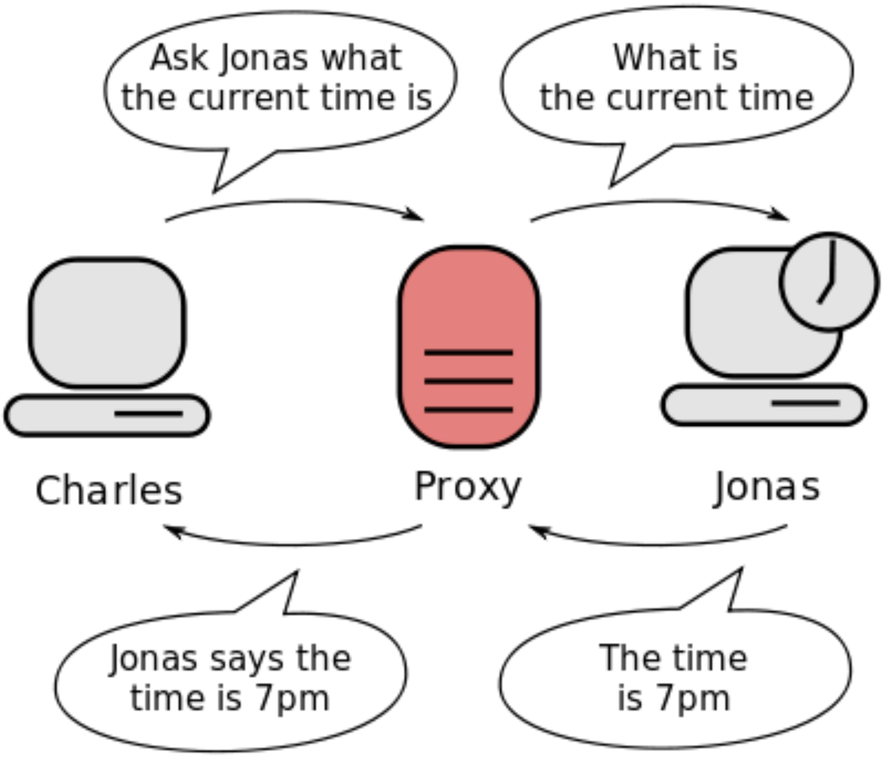
Networking and Cyber security

Gil David

Assignment I: Annoying HTTP Web Proxy

What is a Web proxy?

A Web proxy is a server that acts as an intermediary for requests from the client seeking resources from a Web server. A client connects to the proxy server and asks for web pages or other resources that are available from a remote Web server. The proxy server connects to the remote Web server on behalf of the client and asks for the resource. Then, he returns the response of the Web server to the client.


Ask Jonas what the current time is
What is the current time
Charles
Proxy
Jonas
Jonas says the time is 7pm
The time is 7pm

Assignment

Part 1: build an HTTP Web proxy

How does it work?

1. The proxy accepts TCP connections on port 8080 on the local computer
2. The proxy receives the client HTTP request and directs it to the target web server
3. The proxy receives the reply of the Web server and directs it to the client

How to implement?
Build a program in Python that uses Python’s network libraries

How to test the proxy?

1. Run the proxy listening on port 8080 on the local host
2. Open a browser on the same machine
3. Configure the network settings of the browser - define a manual proxy for http using the localhost and port 8080
4. Your browser will direct its requests to the proxy
5. Browse to couple of websites and verify that you get them correctly

Part 2: extend the HTTP Web proxy to be an annoying proxy

The annoying proxy will manipulate the requests/replies in different ways, before directing them to the target

How does it work?

1. The proxy accepts TCP connections on port 8080 on the local computer

2. The proxy receives the client HTTP request and directs it to the target web server
3. The proxy receives the reply of the Web server and directs it to the client
4. The proxy implements some annoying requests/responses behavior by manipulating the requests from the client prior to transferring them to the web server and the responses from the web server prior to transferring them to the client
    a. Html manipulation – changing the title of the returned HTML page
    b. Images manipulation - randomly replacing the returned images with
        i. The same image but rotated, or
        ii. Another unrelated image
    c. Client's data manipulation - looking for parameters in the client's POST requests and changing them to something else

How to implement?
Override do_GET() and do_POST() in your web proxy

How to test the annoying proxy?

1. Run the proxy listening on port 8080 on the local host
2. Open a browser on the same machine
3. Configure the network settings of the browser - define a manual proxy for http using the localhost and port 8080
4. Your browser will direct its requests to the proxy
5. Browse to couple of websites (http GET) and verify that you experience the annoying behavior (html title and images)
6. Post data to couple of web pages (HTTP POST) and verify that you experience the annoying behavior (the posted parameters are different then what you filled and submitted using the browser)

What to submit?

a. A single python file (WebProxy.py) that gets a single parameter, which is the mode flag: -p for the basic Web proxy mode (part 1 in the assignment) and –a for the annoying version of the Web proxy (part 2 in the assignment)

b. A list of links for 6 web sites that were tested successfully in part 1 and 2 of the assignment (basic proxy and annoying proxy), where at least two of them involve POST requests