Final Project for Richard Givens

Software Vulnerabilities and Defense

SU17-CPSC-66500-002

Synopsis:

This document is a step by step guide to completing The CHALLENGE section of WebGoat, using specific methods and tools discussed below.

For the purposes of this assignment, students are responsible for the following:

1. Complete the challenge successfully (50%)
2. Use Burp Suite to aide in your attacks. (5%)
3. Automate wherever possible. (40%)
4. Clearly explain the steps you took in a video or step by step how-to guide. (5%)

A word about techniques:

Various resources describe how to complete The CHALLENGE with less involved and far quicker methods. However, those methods do not meet the requirements for this project. There are multiple paths to solving any problem, this is merely one of them. No claim is made regarding the superiority or preference of one method over any other.

Requirements:

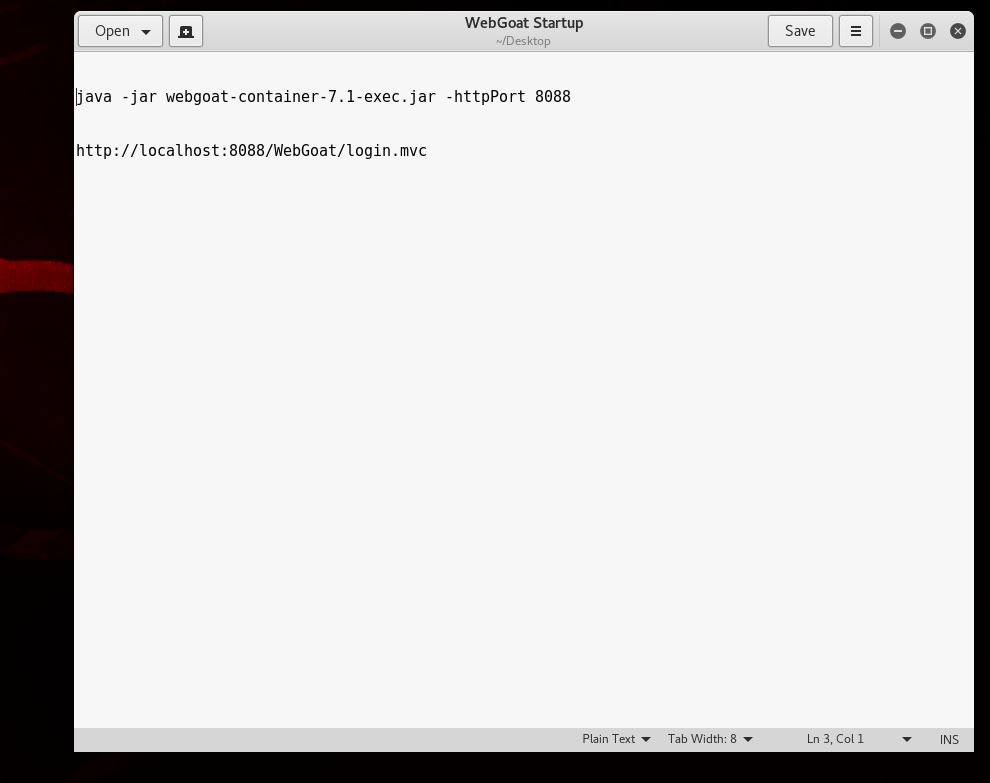
To use this document as a step by step guide for completing The CHALLENGE, users will require the following:

* A copy of “The Web Application Hacker’s Handbook, 2nd Edition,” by Dafydd Studdard and Marcus Pinto. A familiarity with the text is necessary to understand the concepts addressed in the challenge.
* The most recent, and updated rolling distribution of Kali Linux, by Offensive Security, available at <https://www.kali.org/>. Though the exact method for installing and starting Kali Linux is up to the user, the methods described here were completed on a Windows 10 PC running VMWare Workstation Pro with a fresh install of the Kali 64-bit ISO file available at <https://www.kali.org/downloads/>. Instructions for downloading, installing, running, and updating Kali Linux on any medium or format are outside the scope of this document, but readily available online.
* BurpSuite, part of the Kali Linux distribution. The Free Edition is sufficient for completion of The CHALLENGE, and other Lab Sections of WebGoat. All settings should remain at default.
* A recently released, stable version of WebGoat, available at <https://github.com/WebGoat/WebGoat/wiki/Running-WebGoat>. Legacy and Development versions may have different functionality or unresolved bug. Once downloaded, it is recommended to move the file out of your Downloads directory to a permanent location. Open a Text Editor, Copy, then Paste the following lines into the body of the text document:

java -jar webgoat-container-7.1-exec.jar -httpPort 8088

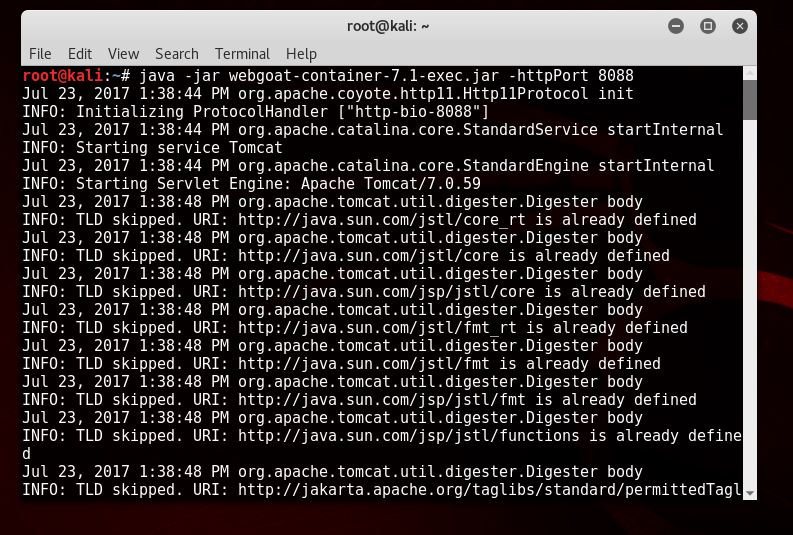
<http://localhost:8088/WebGoat/login.mvc>

Line one of the text document is the Terminal Command you will need to enter when starting WebGoat. Use the specific version information of your application, if different from 7.1, in the body of the command. The appended “-httpPort 8088” specifies the port used by the application, which by necessity must be different than the BurpSuite Proxy. Save this document under a name, and in a location, you will easily remember.



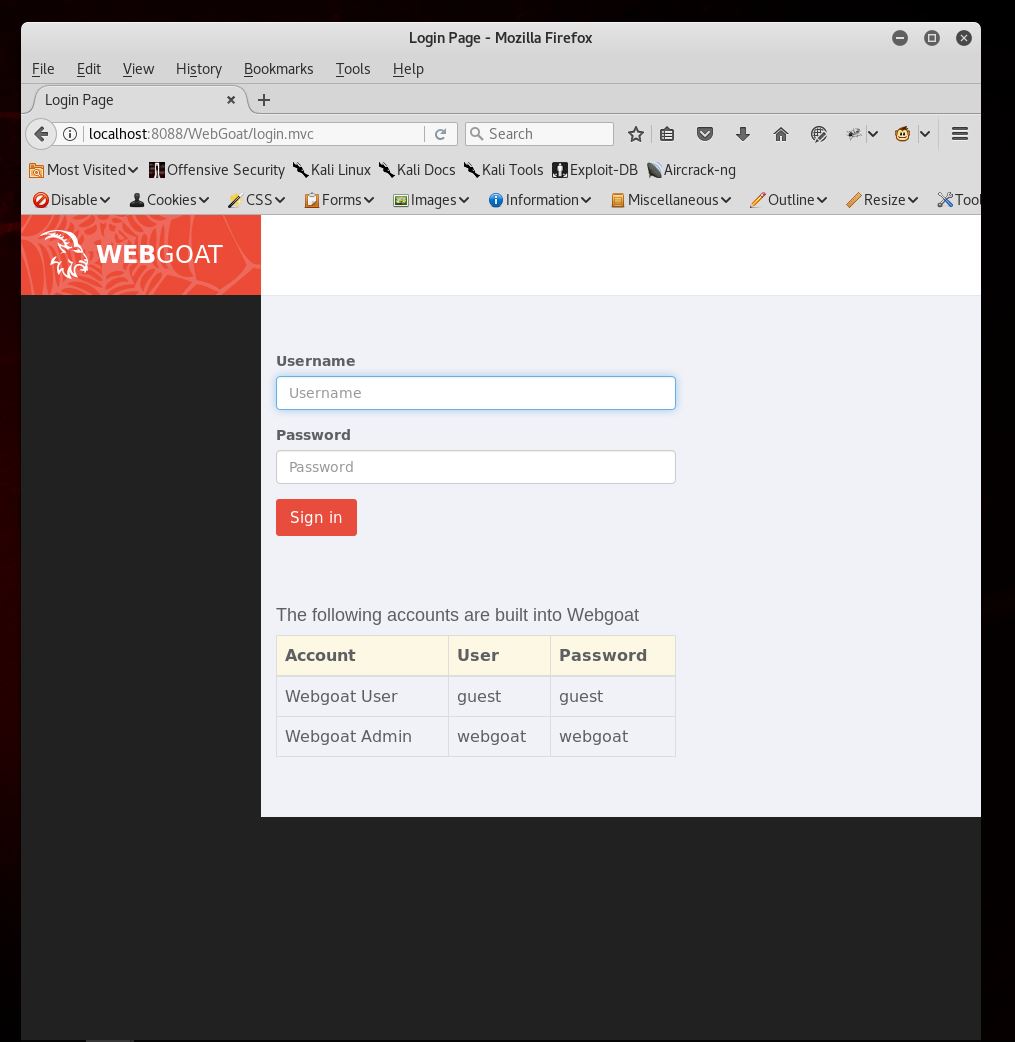
Initial Setup

1. Once you have downloaded the required resources, and are ready to begin the challenge, navigate to the directory you placed WebGoat, right click, and select Open Terminal.
2. Navigate to the previously saved text file containing the WebGoat exe command and localhost address, and open it. Copy the first line and paste it into the Terminal. Then press “Enter.”
3. Many lines of text will scroll across the terminal readout. Wait until the text readout stops, then copy the second line from the text document.



VERY IMPORTANT, KEEP THIS TERMINAL WINDOW OPEN! Closing the terminal will cause WebGoat to stop functioning.

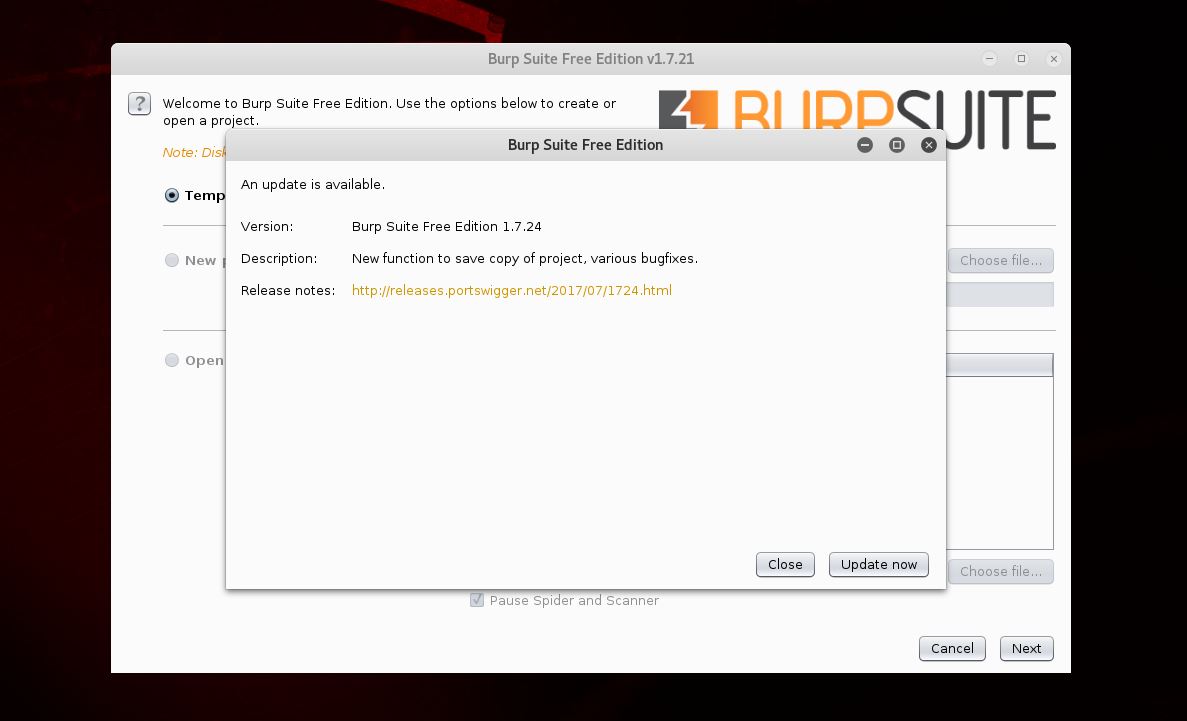
1. Open Firefox ESR (default web browser for the current Kali rolling release), and paste the address into the browser bar. Then, press “Enter.”
2. If you have followed the previous steps correctly, you will be greeted with the WebGoat sign in page. Enter the User Name “Guest” followed by the Password “Guest.”



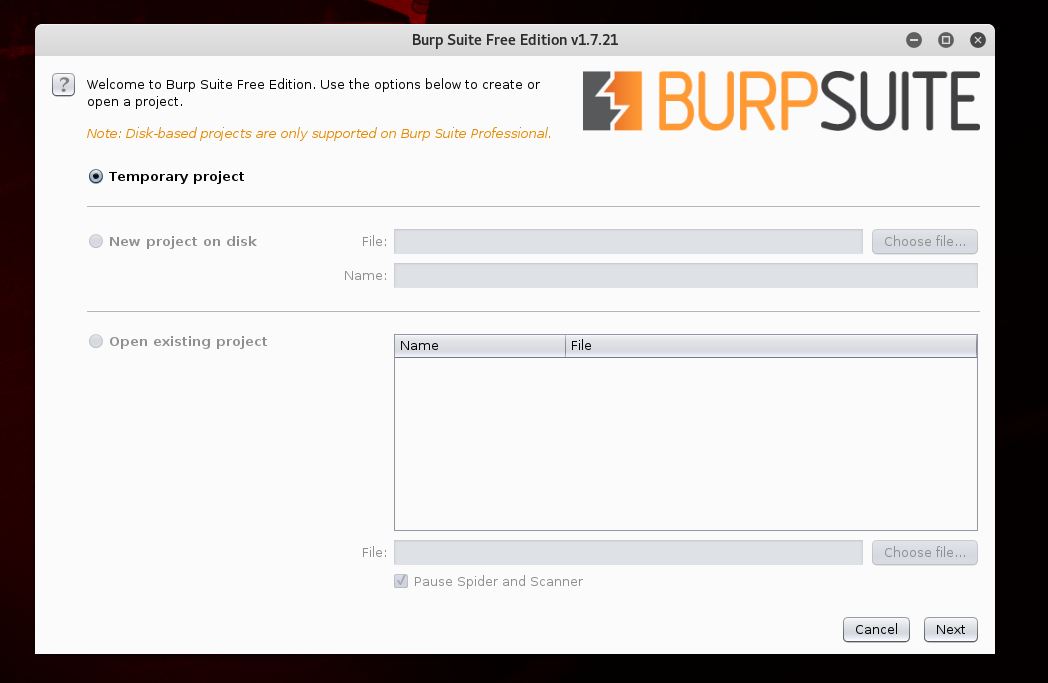
1. From the Kali Favorites Bar, select BurpSuite (a square black and orange icon that resembles a styled profile of a face). Optionally you may navigate to it using Applications > Web Application Analysis, and selecting the first icon.



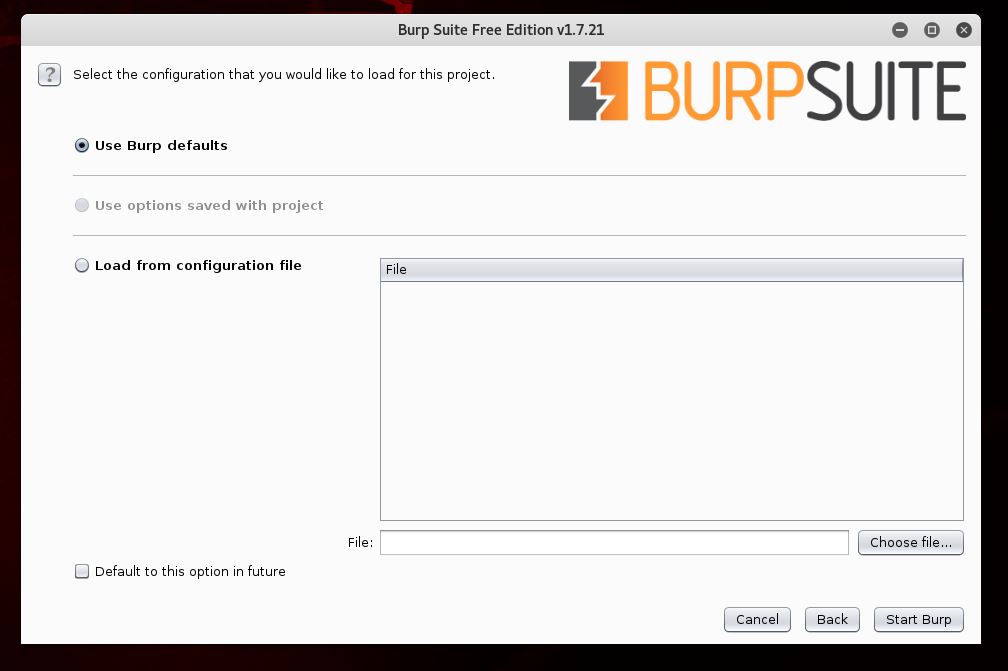
1. Updating the application is not necessary, so you may close any alert box prompting you to do so.



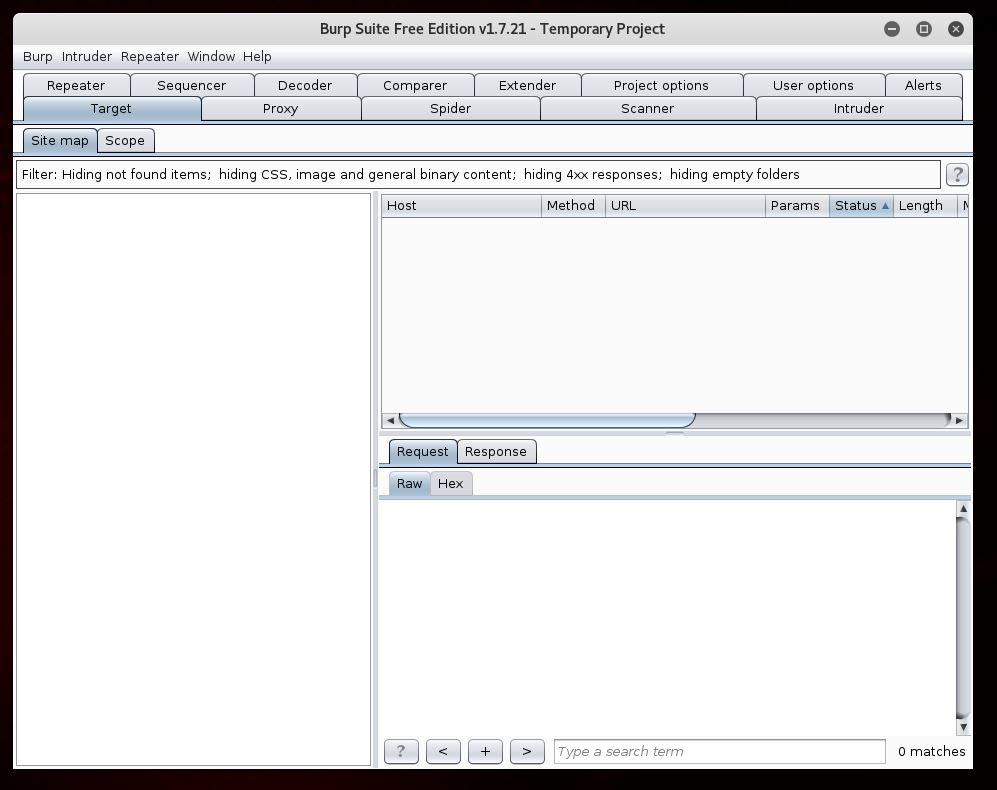
1. Once the application loads, leave the selection on Temporary Project, and select “Next”.



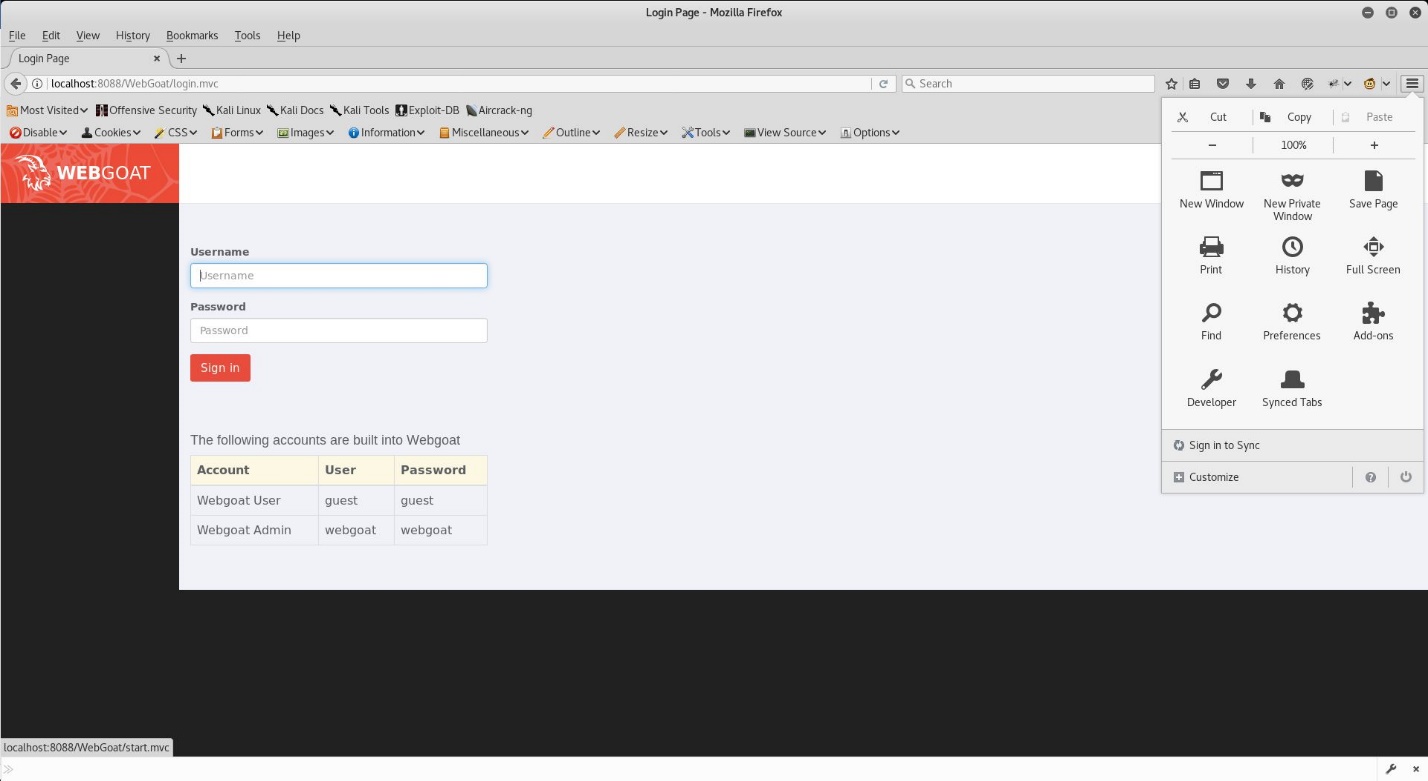
1. Leave the selection on “Use Burp defaults”, and select “Start Burp”.



1. You will be greeted with the project dashboard. Along the top of the application, below the menu bar, will be two rows of tabs. You will navigate to various sections using these tabs throughout The CHALLENGE, so it is necessary to familiarize yourself with their layout.



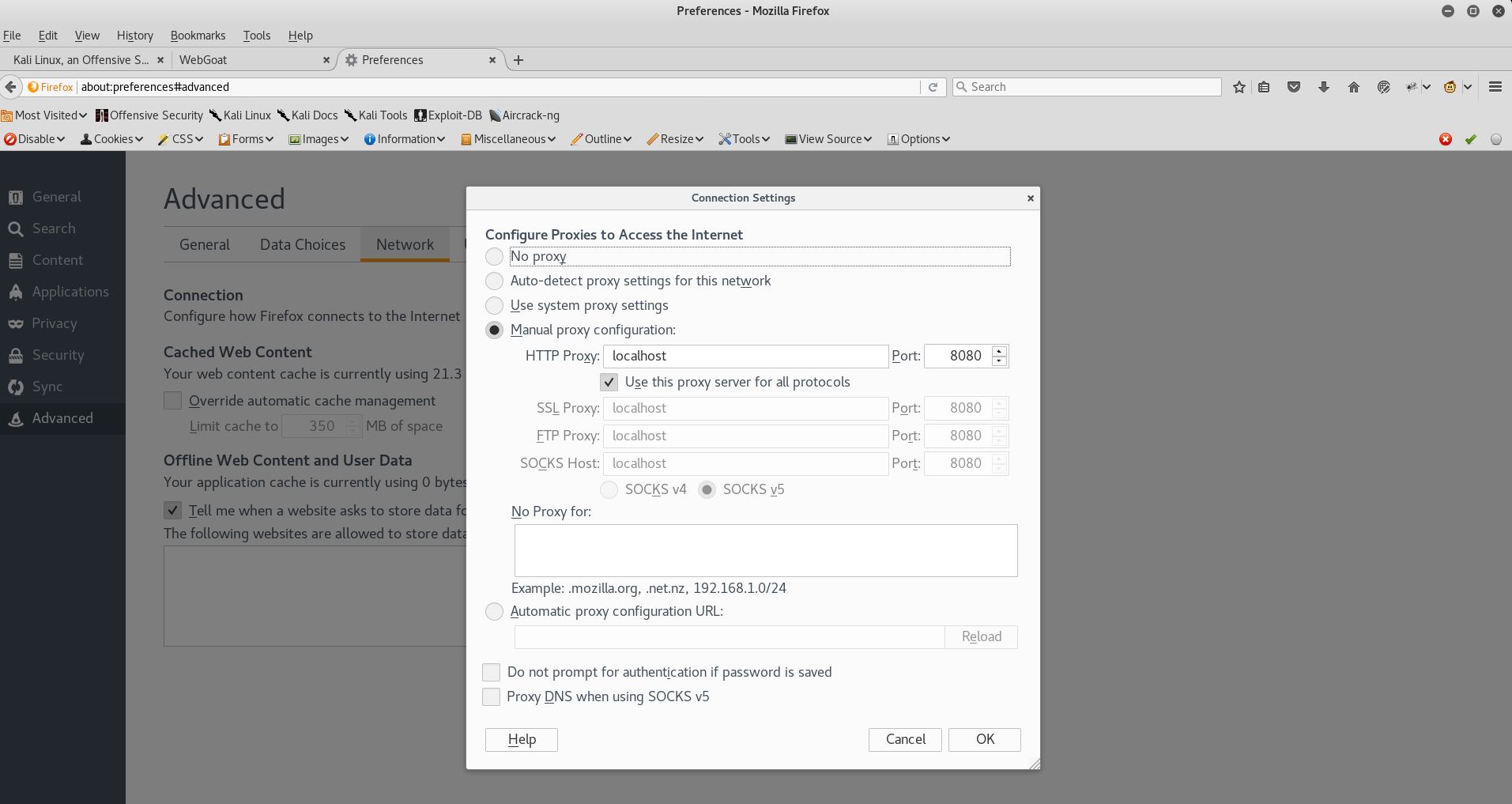
1. The Proxy section of BurpSuite uses port 8080 (also the default port of WebGoat, which is why you must specify a custom port at startup using the exe command). Before going any further, you must configure your browser to work with a proxy connection.
2. On Firefox ESR, navigate to the far right of the browser bar, where you will see three horizontal lines. Clicking on them will open the menu, navigate to “Preferences” and select it.



1. A new tab will open. Navigate to the bottom of the left side, to “Advanced”, and select it.



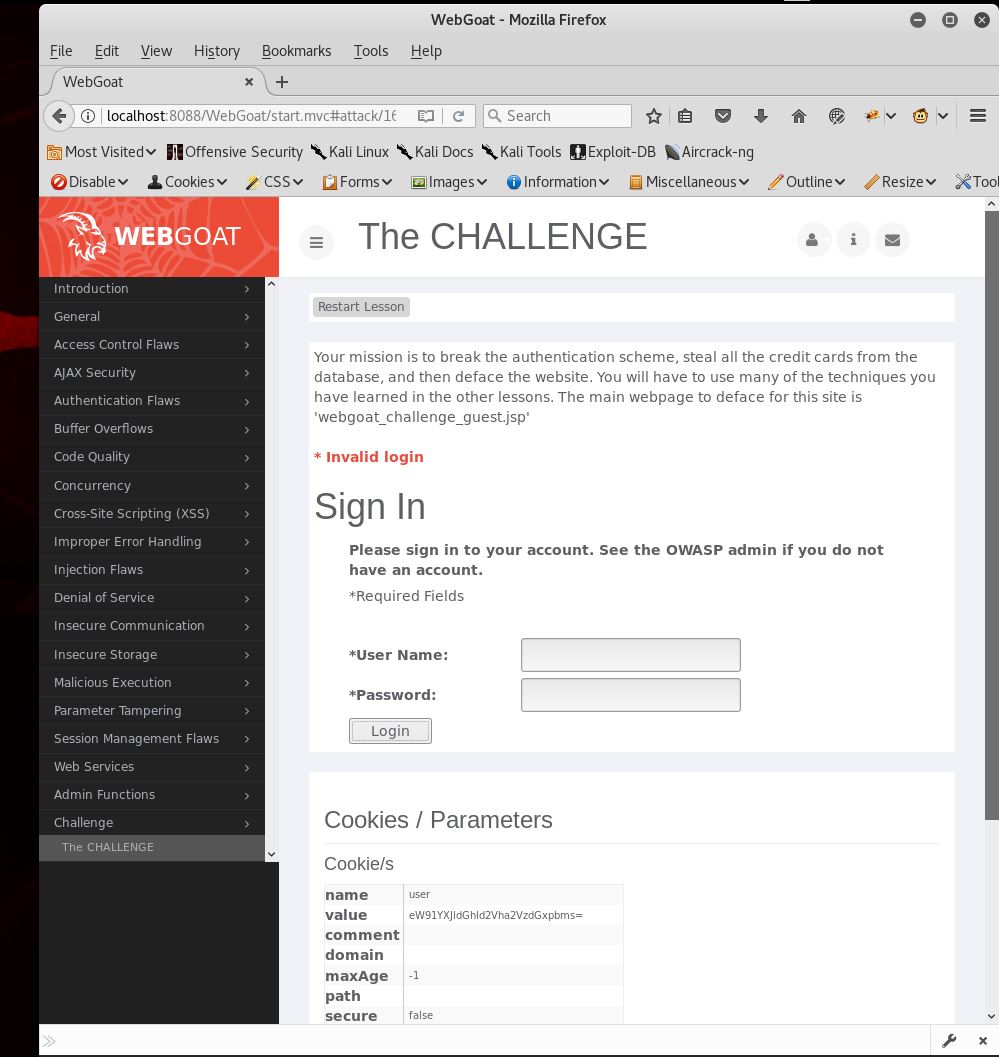
1. On the Advanced tab, select “Network”, then “Settings.”
2. A popup window will appear, change the configuration to match the included screenshot exactly, paying close attention to dialog boxes. Your Connection Settings must be identical to that of the included screenshot for the WebGoat Application to work with BurpSuite’ s built in proxy (or any other proxy).



1. It is recommended that users familiarize themselves with the Introduction, General, and preceding labs before attempting The CHALLENGE, as it will provide users with a well-rounded background experience necessary to understand how to best complete all necessary parts of The CHALLENGE.

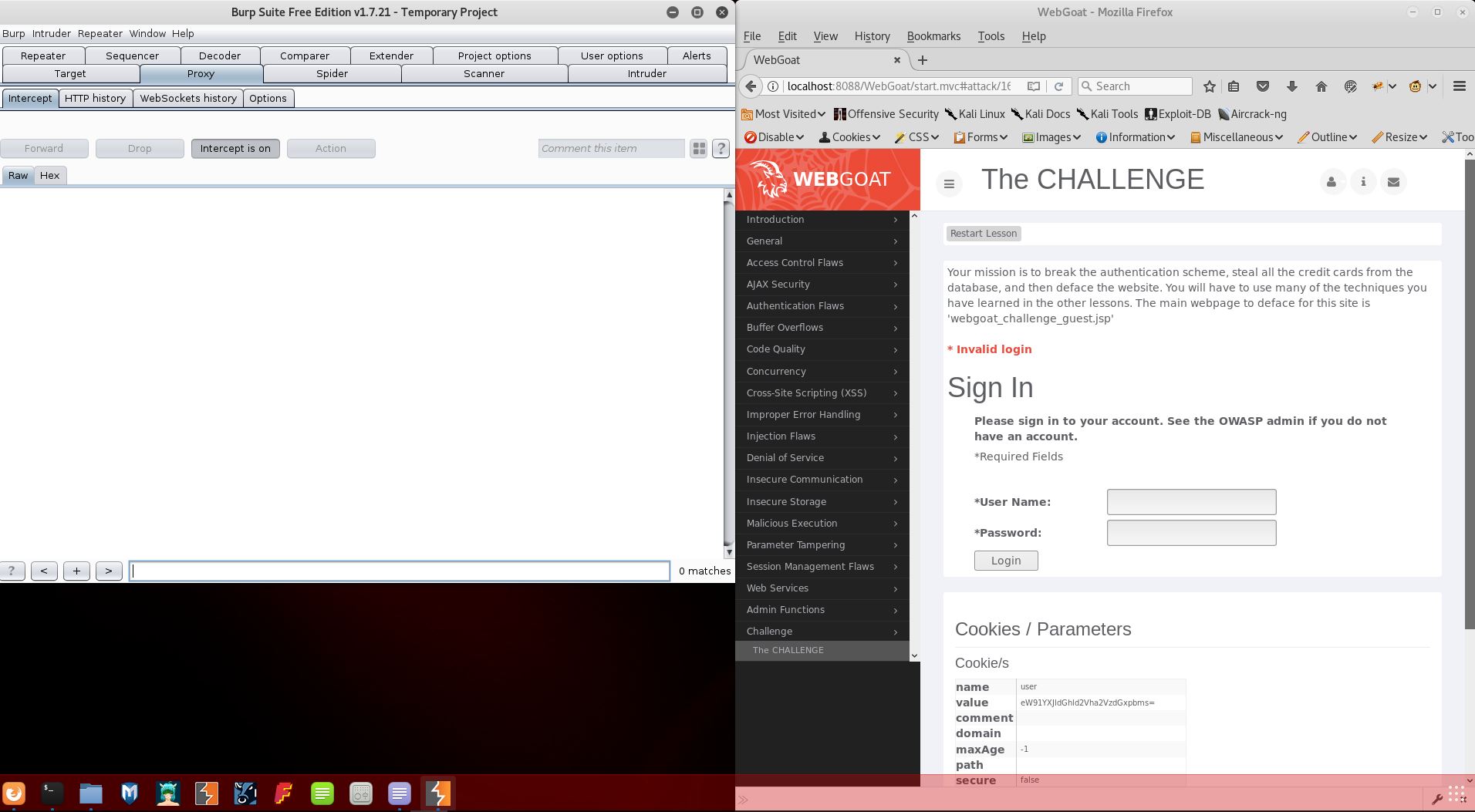
Taking on The Challenge

Once you have sufficiently familiarized yourself with the concepts and exercises of the text, and the previous WebGoat Labs, try The Challenge. Navigate to the very bottom of the WebGoat Application contents.



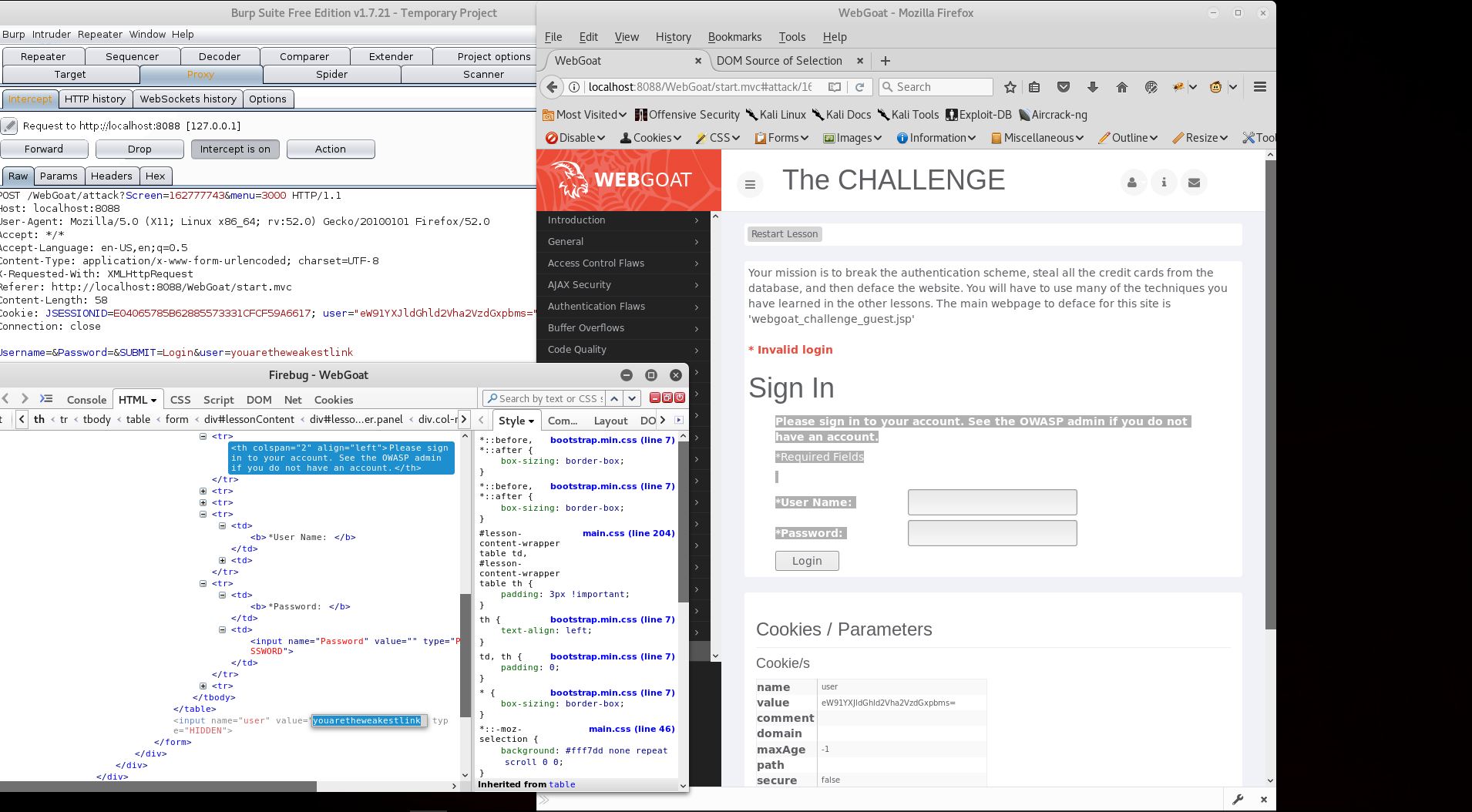
Now, start BurpSuite and set your proxy (if you have not done so already). Remember you must have your connection settings to automatic when first starting WebGoat, then you set your proxy after logging in.

1. We are greeted with a login screen and the directions for the challenge, which are in the screenshot above.
2. Previous exercises have shown that there may be hidden content. So, on BurpSuite, navigate to “Proxy” and make certain Intercept is toggled on. Click “Login,” without adding any text to the input boxes, and observe the intercepted traffic in BurpSuite Proxy.

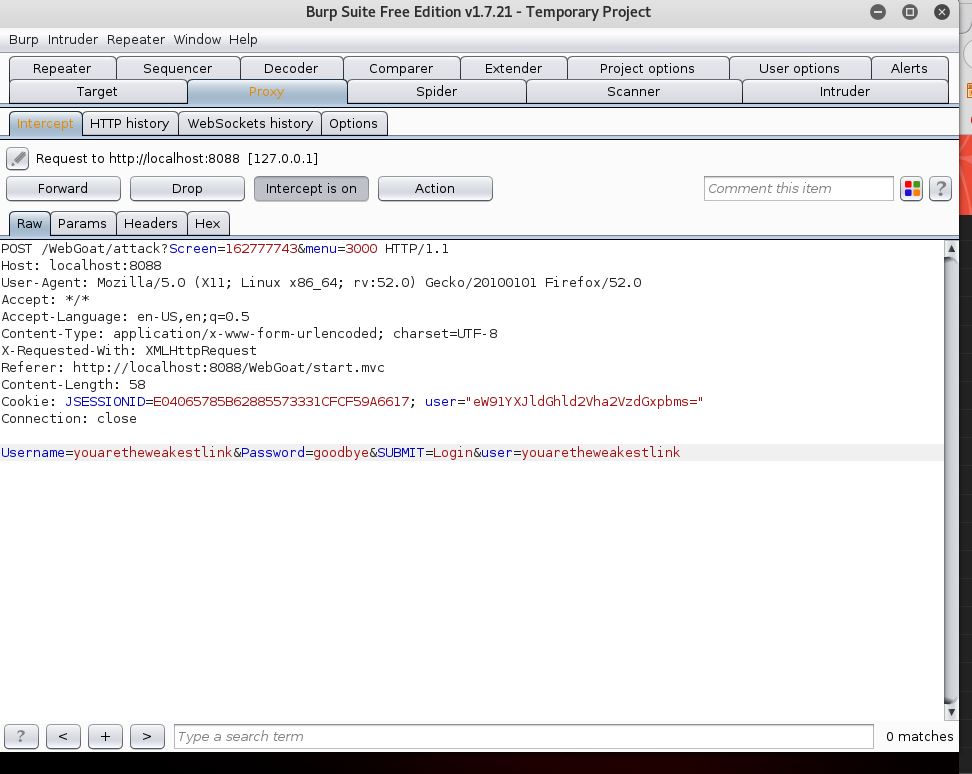


1. By examining the raw data in BurpSuite, it becomes apparent that there is a hidden form field, “user” with the value “youaretheweakestlink”. This can be confirmed by examining the page source using an extension such as Firebug, or another tool sufficient to access the raw page code.

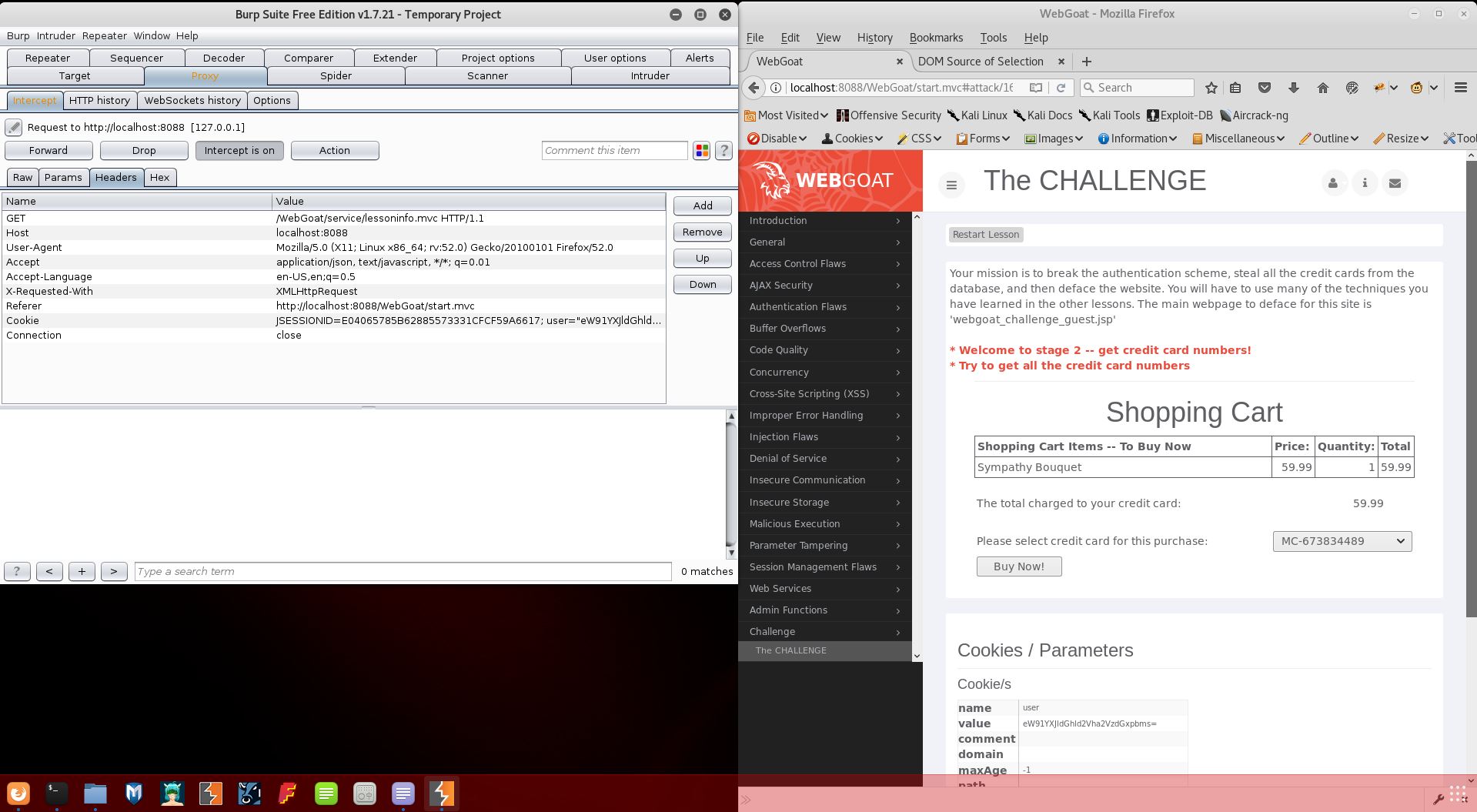
(Note that examining the raw code is not necessary, but provides an alternative means of discovering the same information).



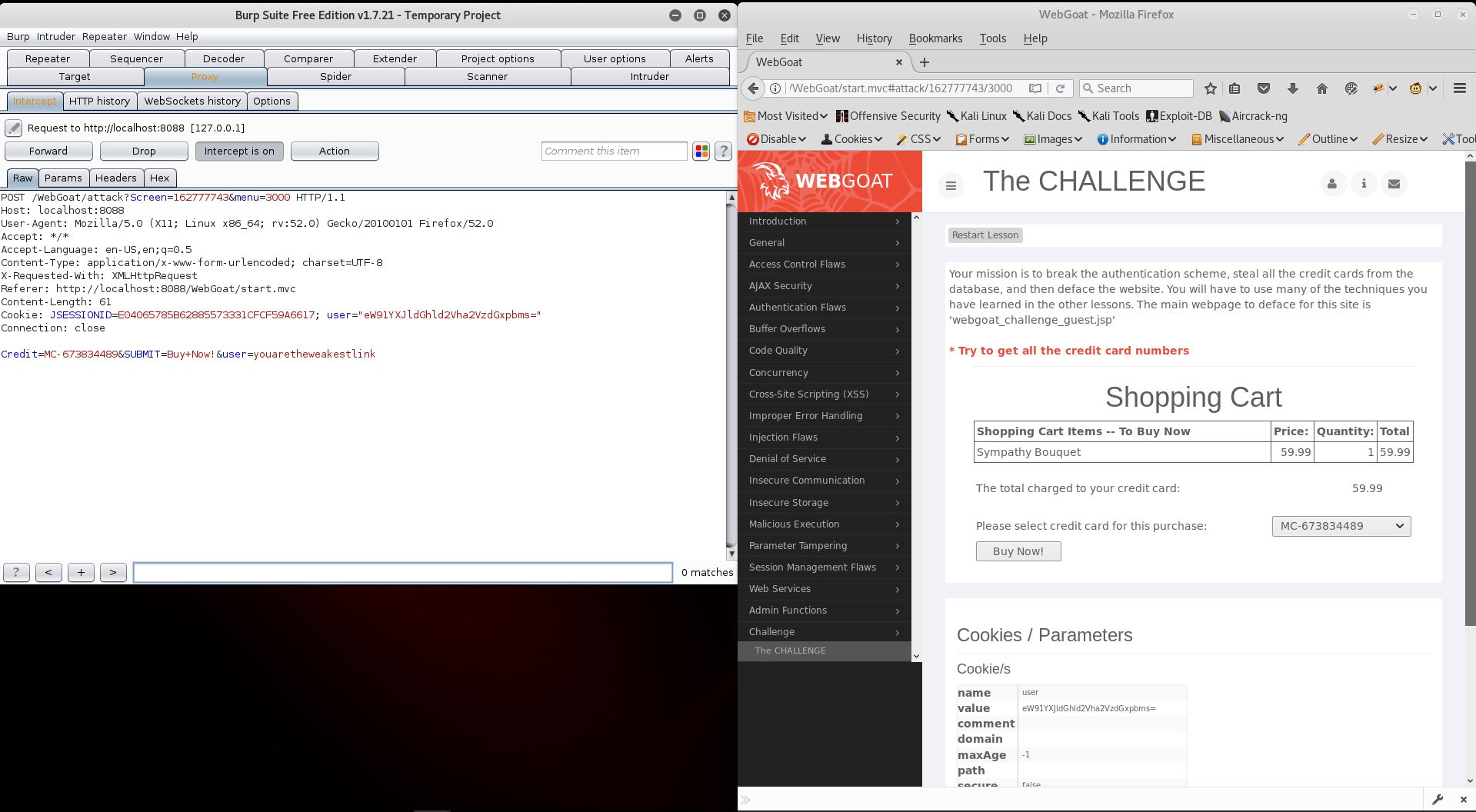
1. The phrase “youaretheweakestlink” is a reference to a once popular television show from the 90s, one that I happened to watch a few times. This gives us a clue as to the possible password. From BurpSuite Proxy, navigate to the “Raw” tab, and insert the following text as shown in the screenshot. Then, select Forward to send the packet to the application.



1. This will complete Stage 1; welcome to Stage 2. At this stage, we are presented with a shopping application, containing a dropdown menu with two credit cards pre-filled; the phrase “Please select credit card for this purchase” is to the left of the credit card box.



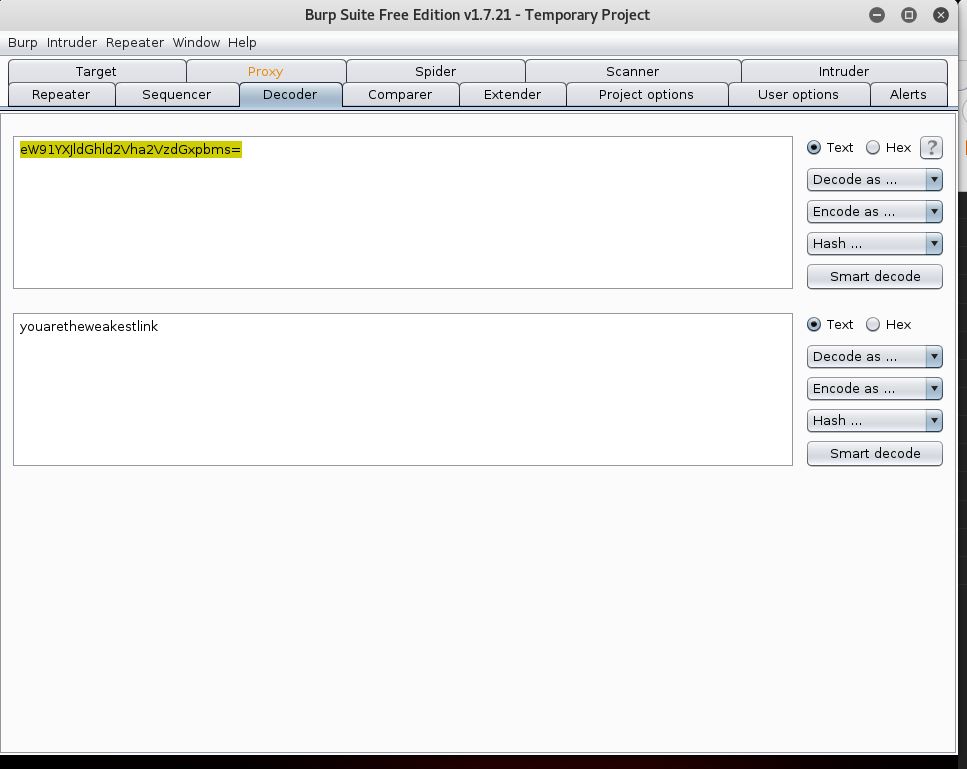
1. Once again using BurpSuite Proxy, with Intercept On, we click the “Buy Now!” button to capture the traffic.



The request reveals two column names for the database, as shown in the Raw tab. “Credit” and “user”. At this point we could attempt, and probably succeed at, SQL injection by replacing the user name and credit card fields with appropriate SQL, but that would not fulfill the automated requirement of this challenge. One could also tamper with the user section of the cookie header, but that also violates the automation requirement.

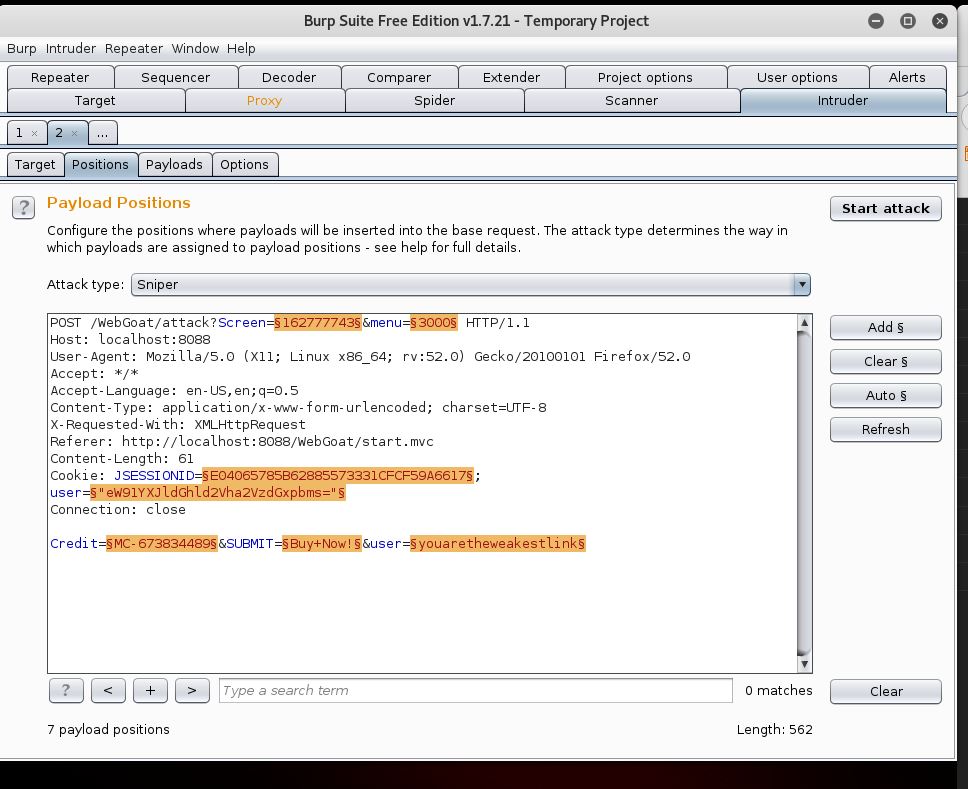
Click the “Action” button, and select “Send to Intruder,” and then navigate to the “Intruder” tab on the far right, bottom row.

1. There are two data points which contain “user=”. One has the value set to “youaretheweakestlink,” and the other to “eW91YXJldGhld2Vha2VzdGxpbms=” located in the Cookie, which seems to control the session. To decode the user info in the cookie, highlight the value, right click and select “Send to Decoder”. The Decoder tab will now be highlighted in orange. Navigate to it, select “Decode as” and choose “Base64”. The value of the user info will now be translated to plain characters, and will be revealed to be identical to the other user info value; “youaretheweakestlink.”

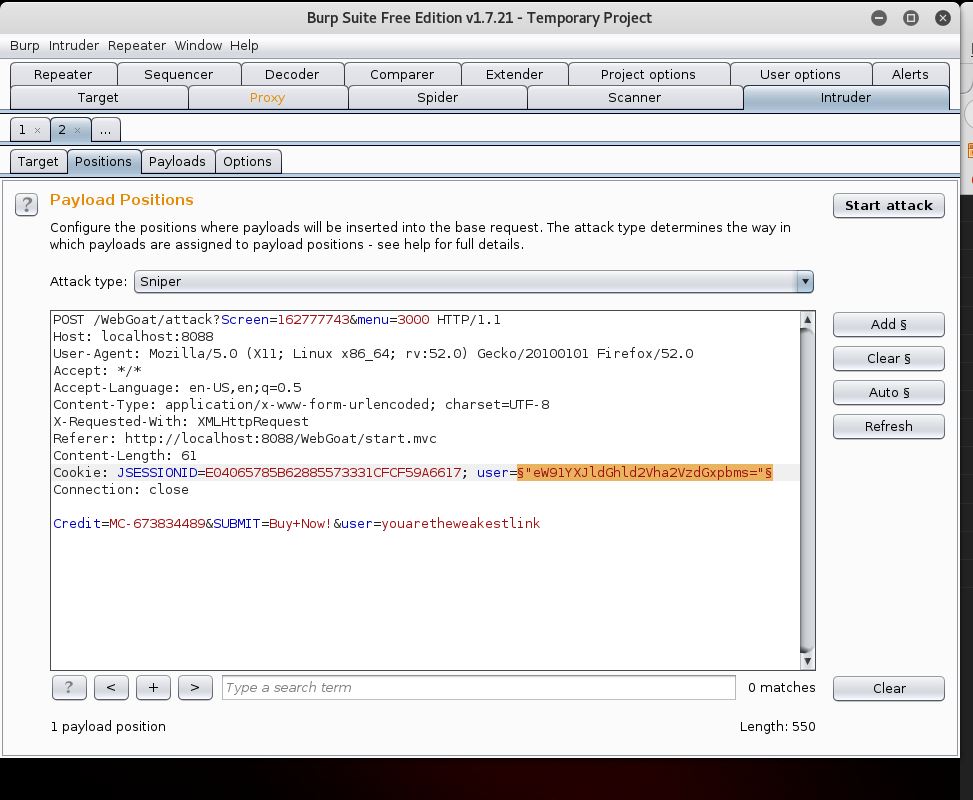


We can surmise that the user info in the cookie is indeed responsible for controlling the display of the credit card numbers. As it is encoded, simply injecting standard SQL into the field probably will not work. We will need to encode any injection before it is passed to the server.

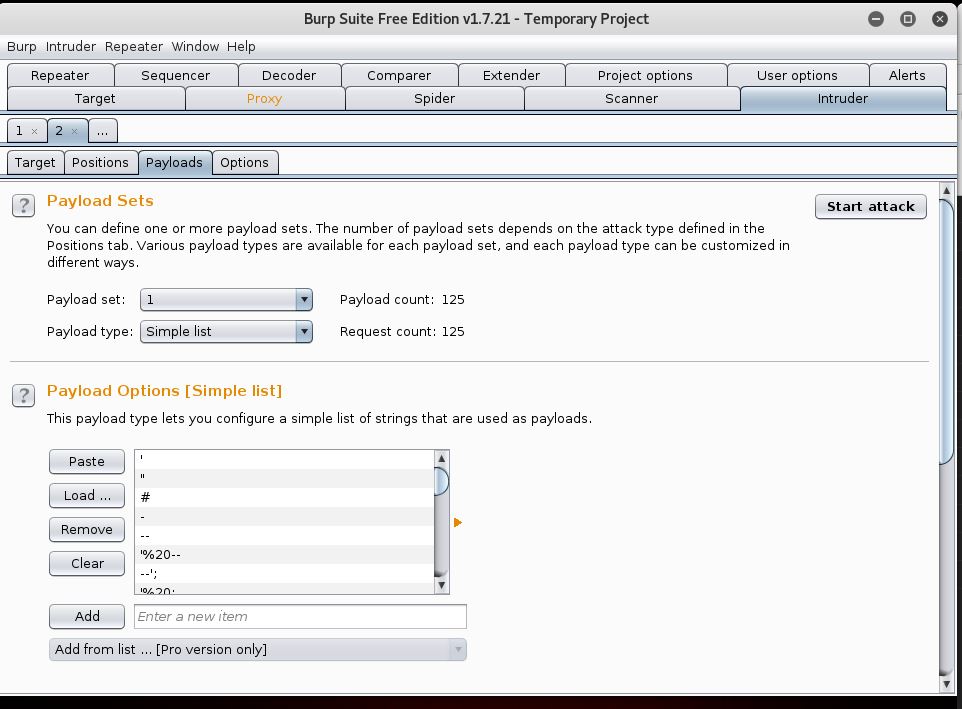
1. Return to the Proxy tab, and select “Action”. Then select “Send to Intruder.” Then, navigate to “Intruder” and select its second tab, titled “Positions.”



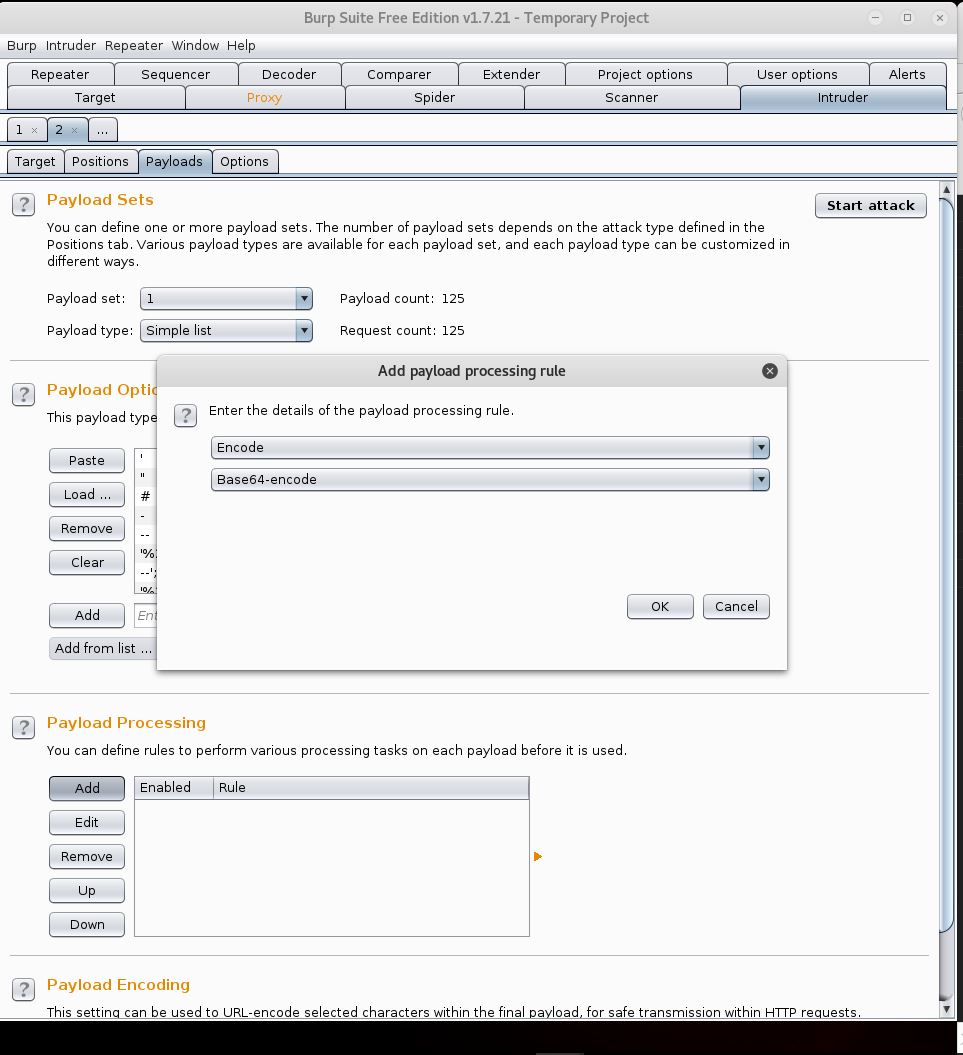
1. Select “Clear” to remove all highlighted areas. Then use your mouse to manually highlight the Base64 encoded user info, and then select “Add”.



1. Navigate to the “Payloads” tab, then scroll down to “Payload Options” and select “Load.” Use the browser to navigate to the /usr/share/wfuzz/wordlist/Injections directory and select “SQL.txt”.



1. Scroll down to “Payload Processing” and then select “Add”. And then use the “Select rule type” dropdown menu to choose “encode.” A second dropdown will appear, select “Base64” about halfway down the menu. Click “OK.”



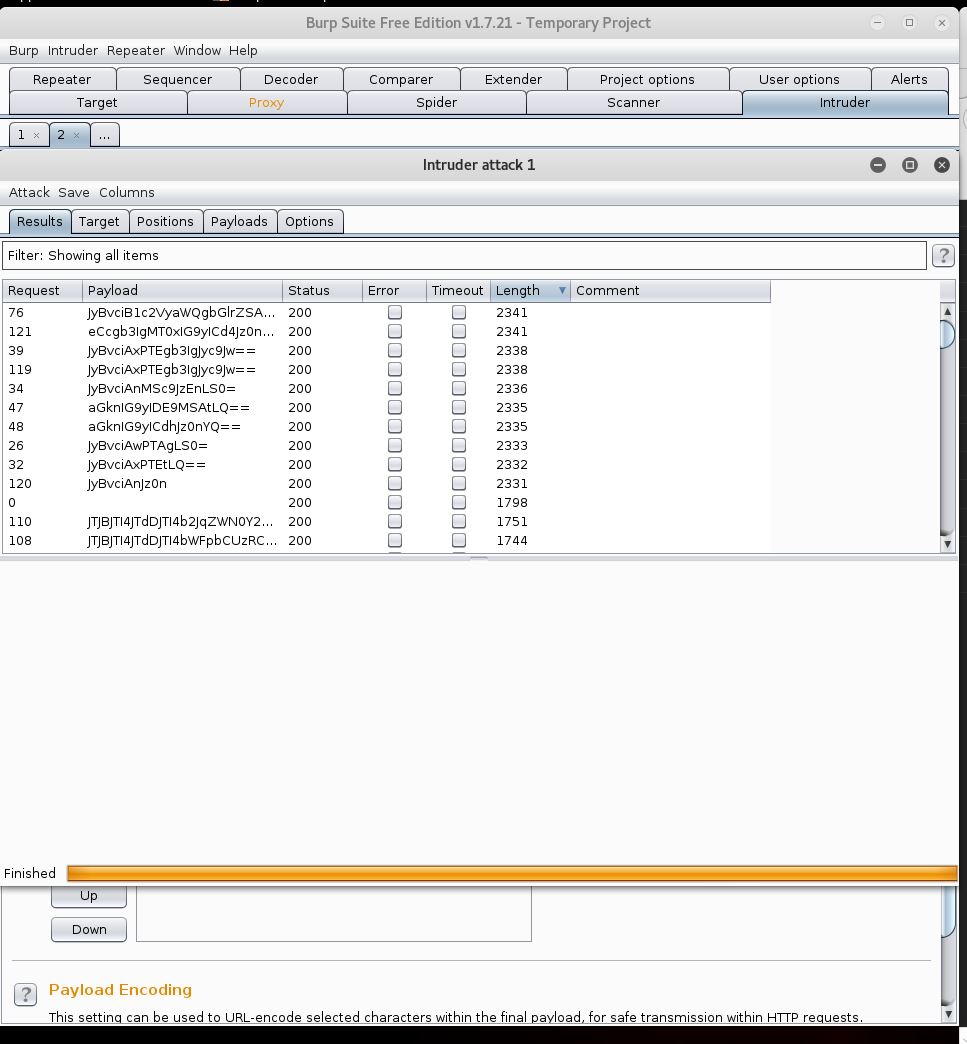
Selecting Base64 encode will guarantee that the SQL injection is encoded to the correct value before it is sent via the payload. Select “Start Attack at the upper right area of the “Payloads” tab.

1. Because we are using the free edition of BurpSuite, the attack may take a few minutes to complete. Allow it to continue uninterrupted. You will notice that as the Request number advances, Payload and Length will vary. This will be important after the attack concludes.

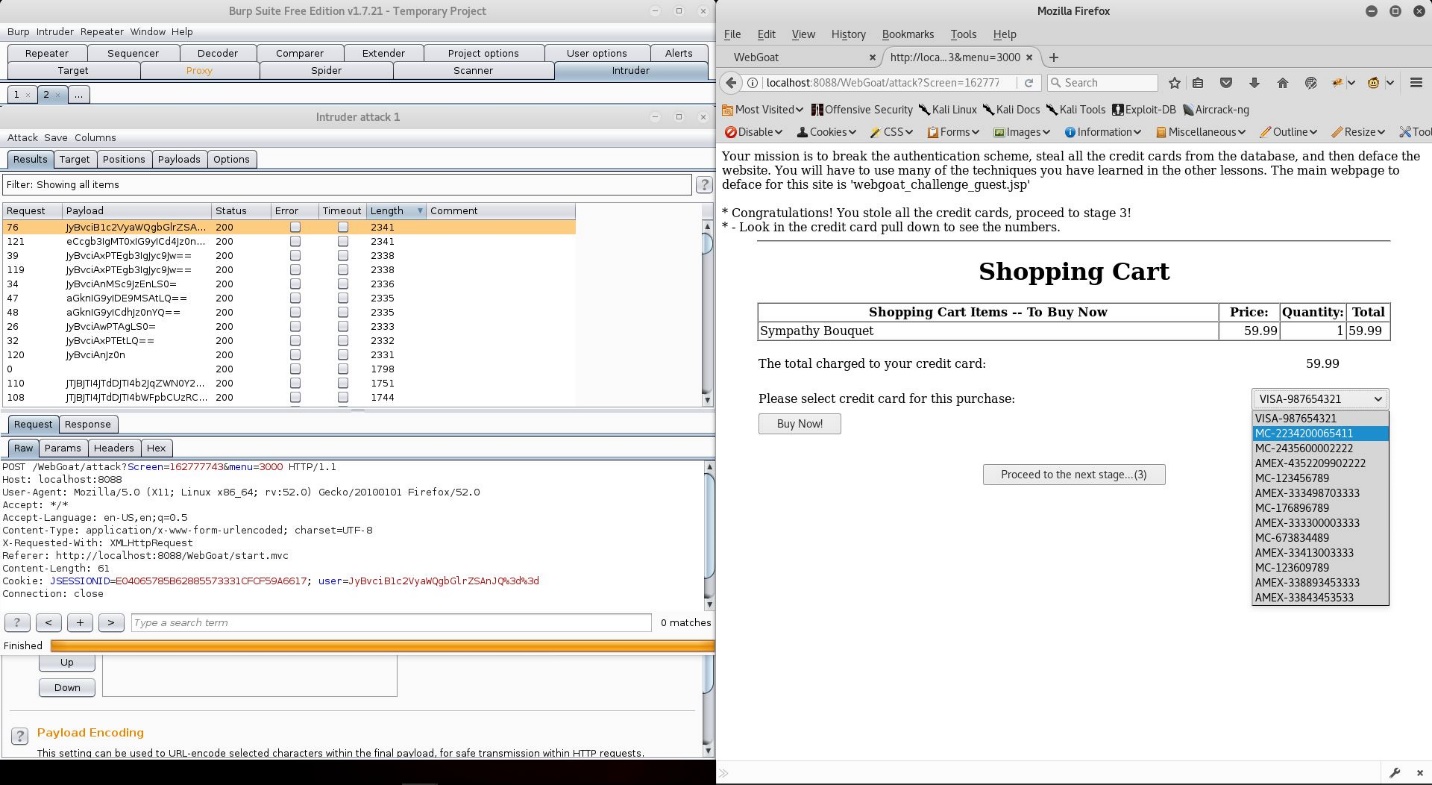


The attack will replace the information in the Base64 encoded “user=” cookie field with SQL statements, that have been Base64 encoded. Since we do not know the exact version of SQL used in the database, automation will run through all values included in the SQL.txt file and return the server response.

1. Toggle the “Length” field twice to order the returned results by descending values. Since we are attempting to pull extra data from the server than would normally be allowed, it is safe to assume that one of the request results with larger fields will contain the desired information. Start with the top.

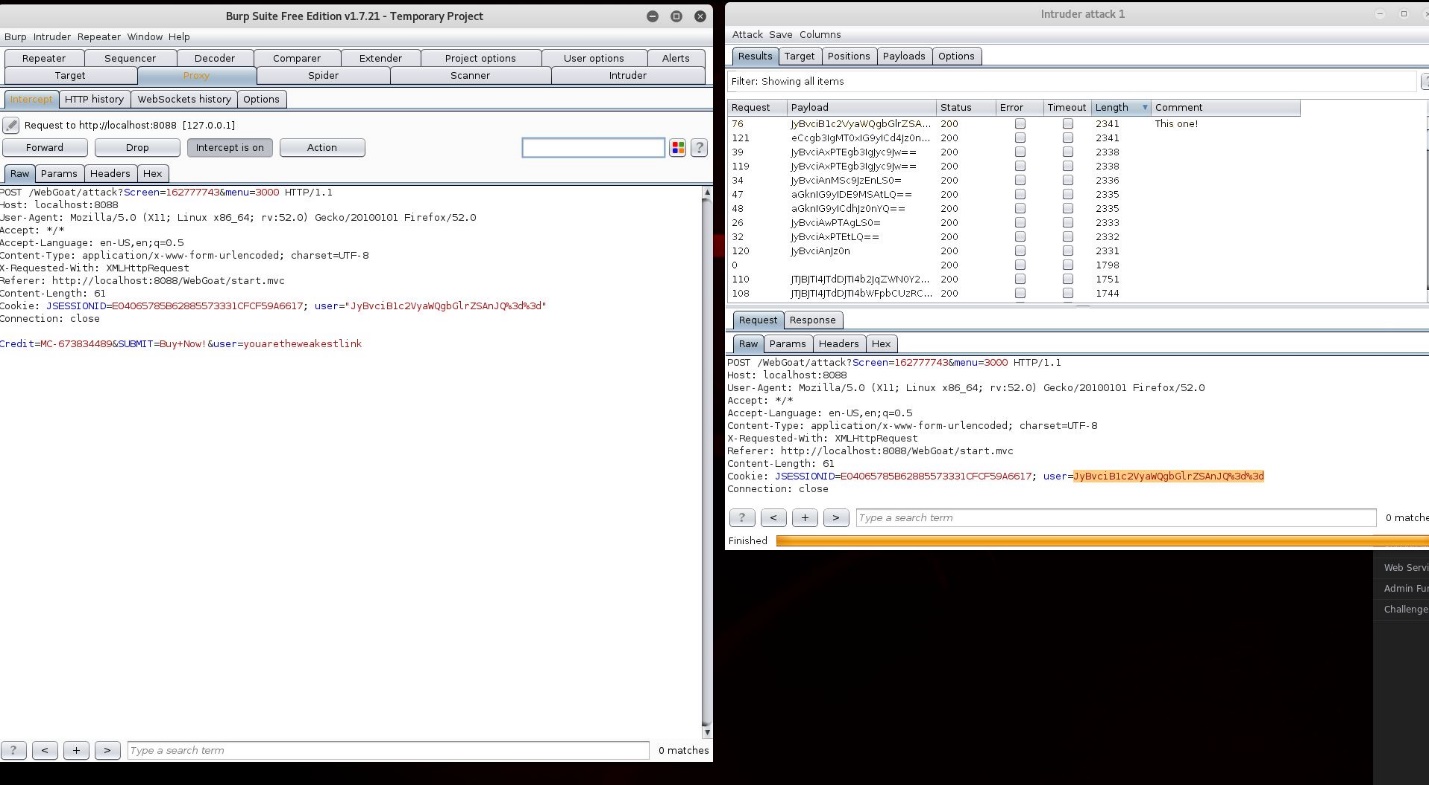


1. Highlight the top result with your mouse, the Request/Response area will populate with data. Right click and select “Show Response in browser”, a popup will appear prompting you to copy a link. Select “Copy”, and then Paste the data into a new tab on your Firefox ESR browser. Selecting the dropdown menu will reveal 13 separate credit card numbers (as shown in the screenshot below).



But, we are still not done. This is only a preview of the results as they may appear in the WebGoat application. We must now forward the correct data back to our proxy. Note that the bottom “user=” field is unaffected by the tampered cookie. Also, Request 121 has an identical length to request 76 (our successful request), indicating that more than one injection command will work.

1. The simplest way to forward this tampered cookie to application is to simply copy and paste the data from the “Raw” tab for the successful attack to the same field in the “Raw” tab on your proxy. But, don’t close your attack until you are certain the effort is successful!

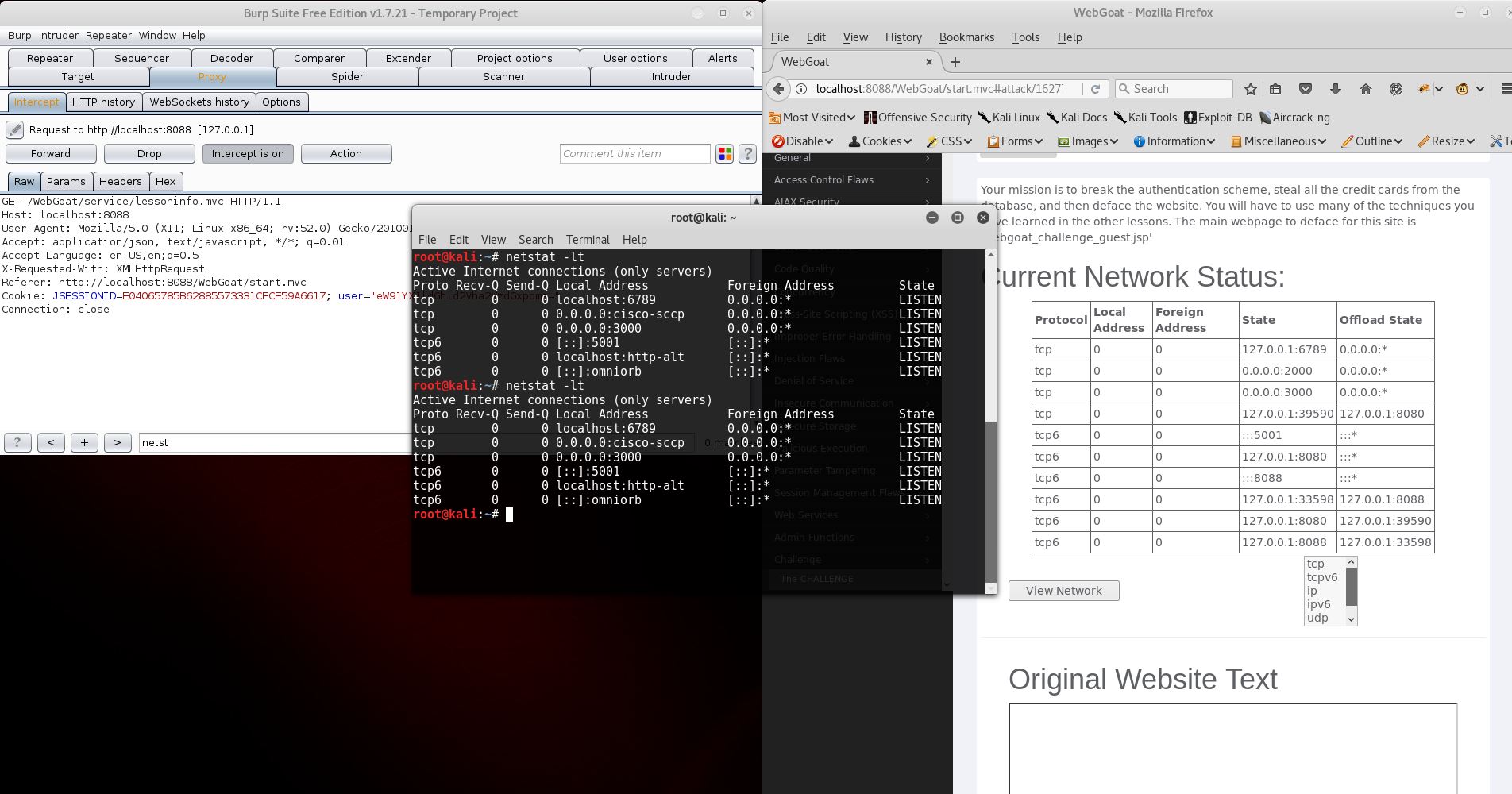


Once you have completed copying the data over, select “Forward” to send the data to the server.

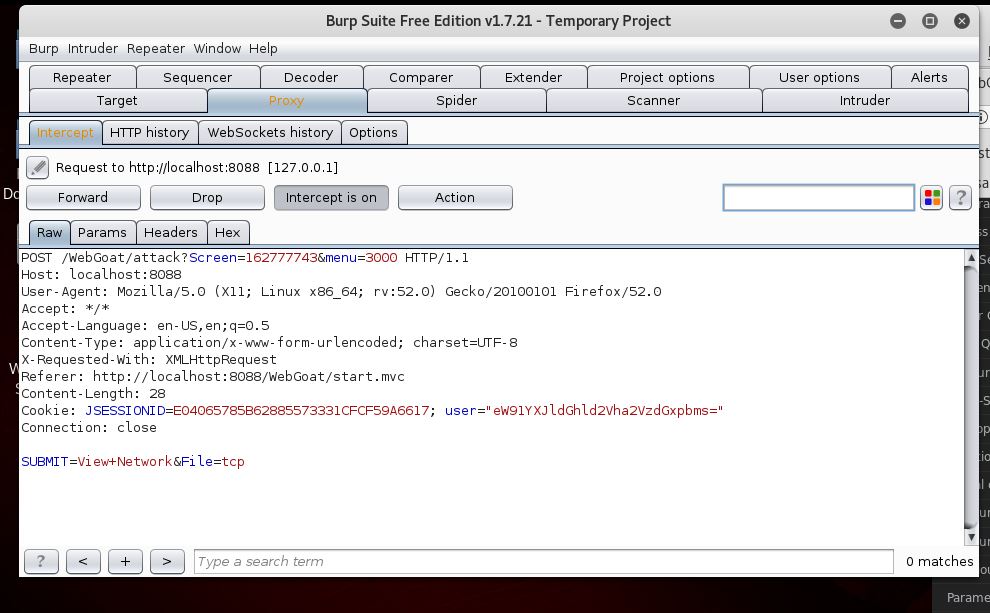
1. Success with Stage 2, ready to move on to Stage 3.



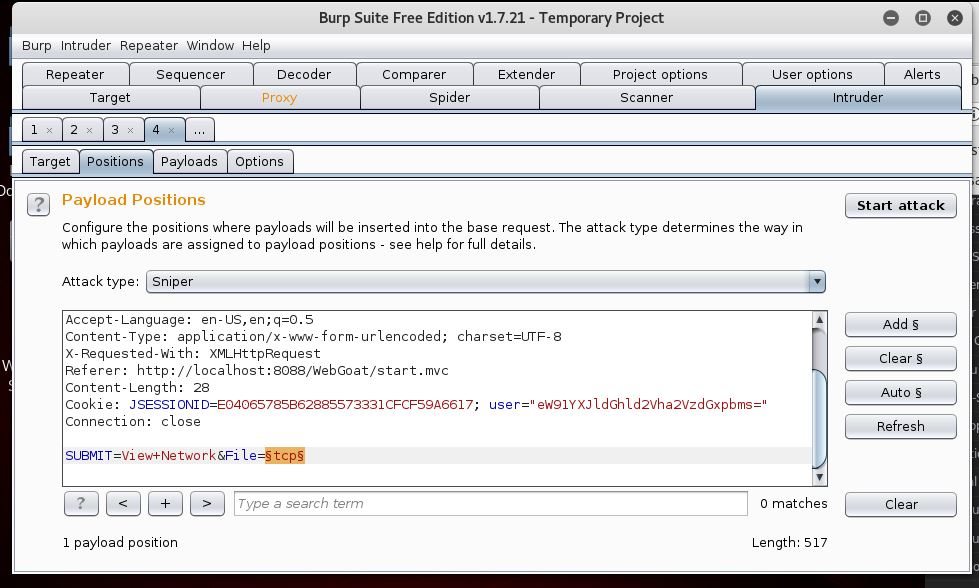
1. We are greeted with a table showing ports and connections on our localhost. This resembles the output of a netstat command. To verify the format, right click on your desktop and select “Open Terminal.” Type netstat -lt and examine the output. You may close the terminal when you are done.

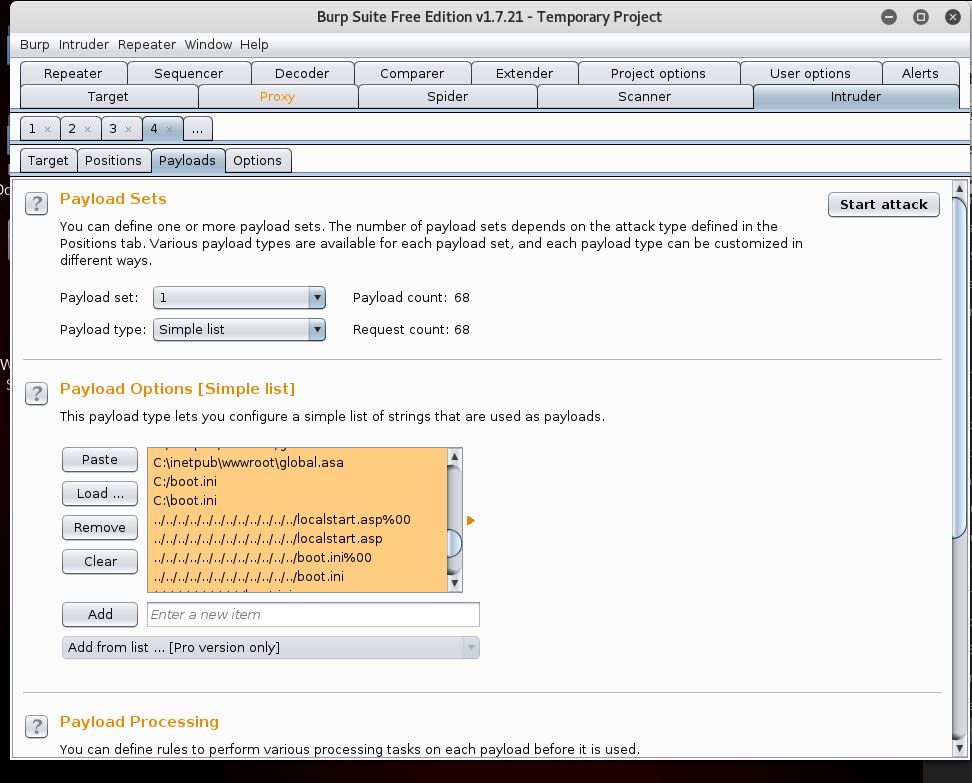


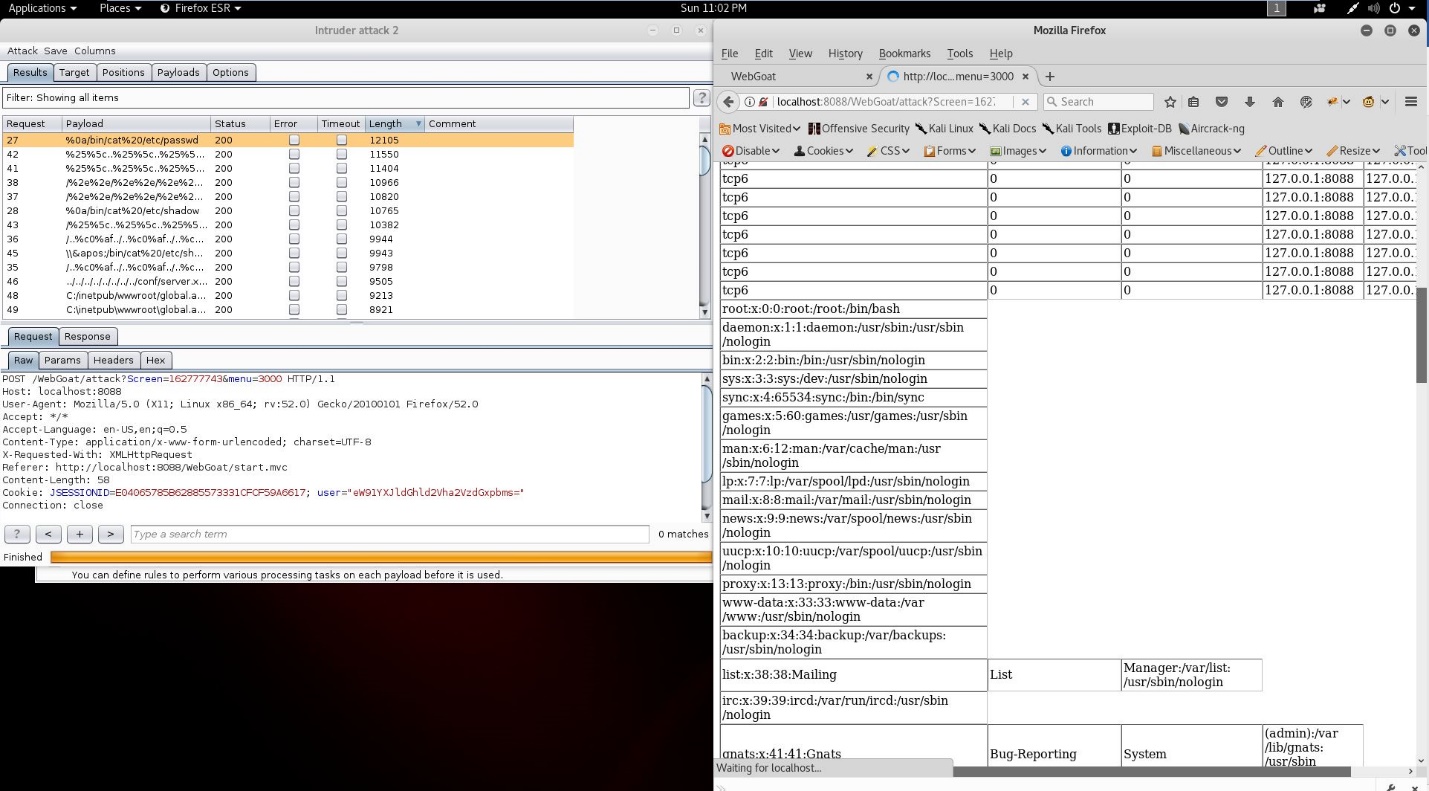
1. As per previous exercises, ensure the “Intercept is on” button in BurpSuite Proxy is toggled, select any of the protocols, and the click “View Network.” BurpSuite Proxy will show the intercepted request. Note the bottom line “SUBMIT =View+Network&File=tcp”; tcp will be the insertion point for our next commands. Click the “Action” button and select “Send to Intruder”. Clear the highlighted areas as previous, highlight “tcp” and click “Add.”



(18 cont.)

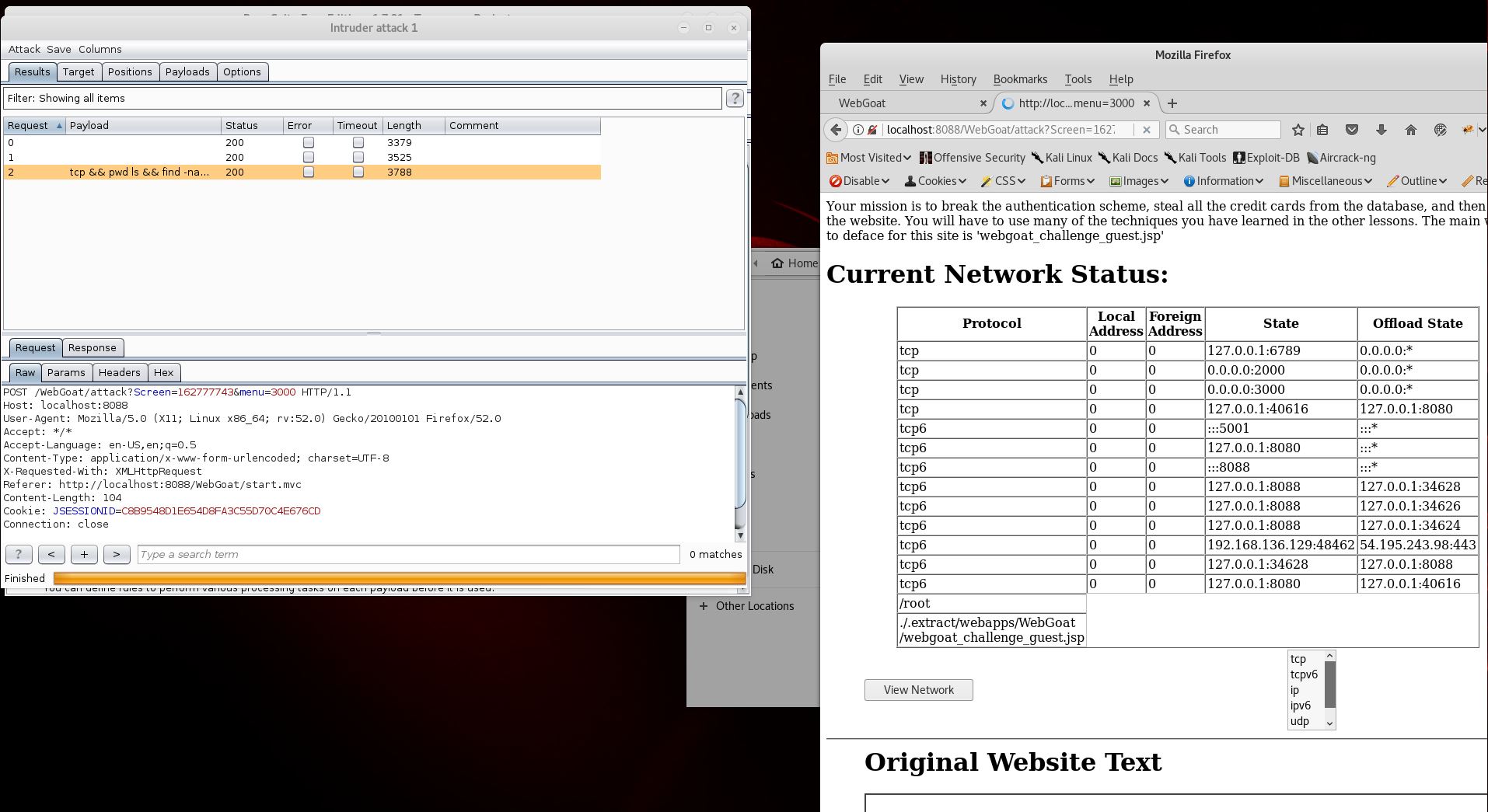


1. Now we want to confirm the suspected vulnerability. Go to the “Payloads” tab, load “Traversal.txt” and click “Start Attack”.
2. As with our previous Intruder attack, the length of the responses will vary. For brevity, select the request with the longest length, right click and show the response in the browser. You will see that the injected commands confirm the suspected vulnerability.



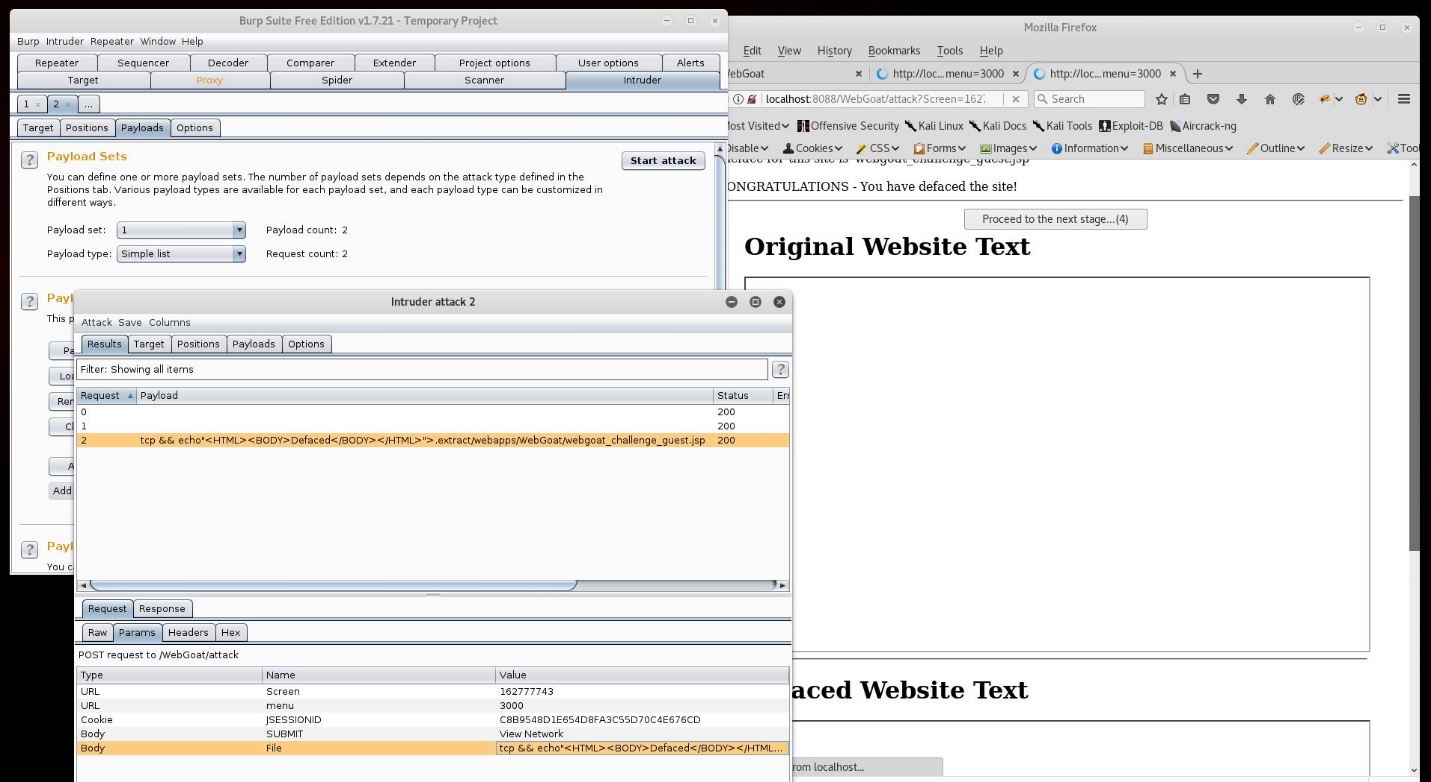
So, we have confirmed our suspicion that the output responds to Terminal commands, now it is time to find the exact location of the .jsp file we need to deface.

1. Injecting some “tcp && pwd ls && find -name webgoat\_challenge\_guest.jsp” into Intruder reveals the location of the webgoat\_challenge.jsp file.



Next, we will use the same method to craft HTML and overwrite the file.

1. First, we’ll test the method, as we did in Stage 2. From Intruder, navigate to “Payloads,” then type the text as shown in “Intruder Attack 2”, modified to include your chosen text, and the unique location of the .jsp file in question.

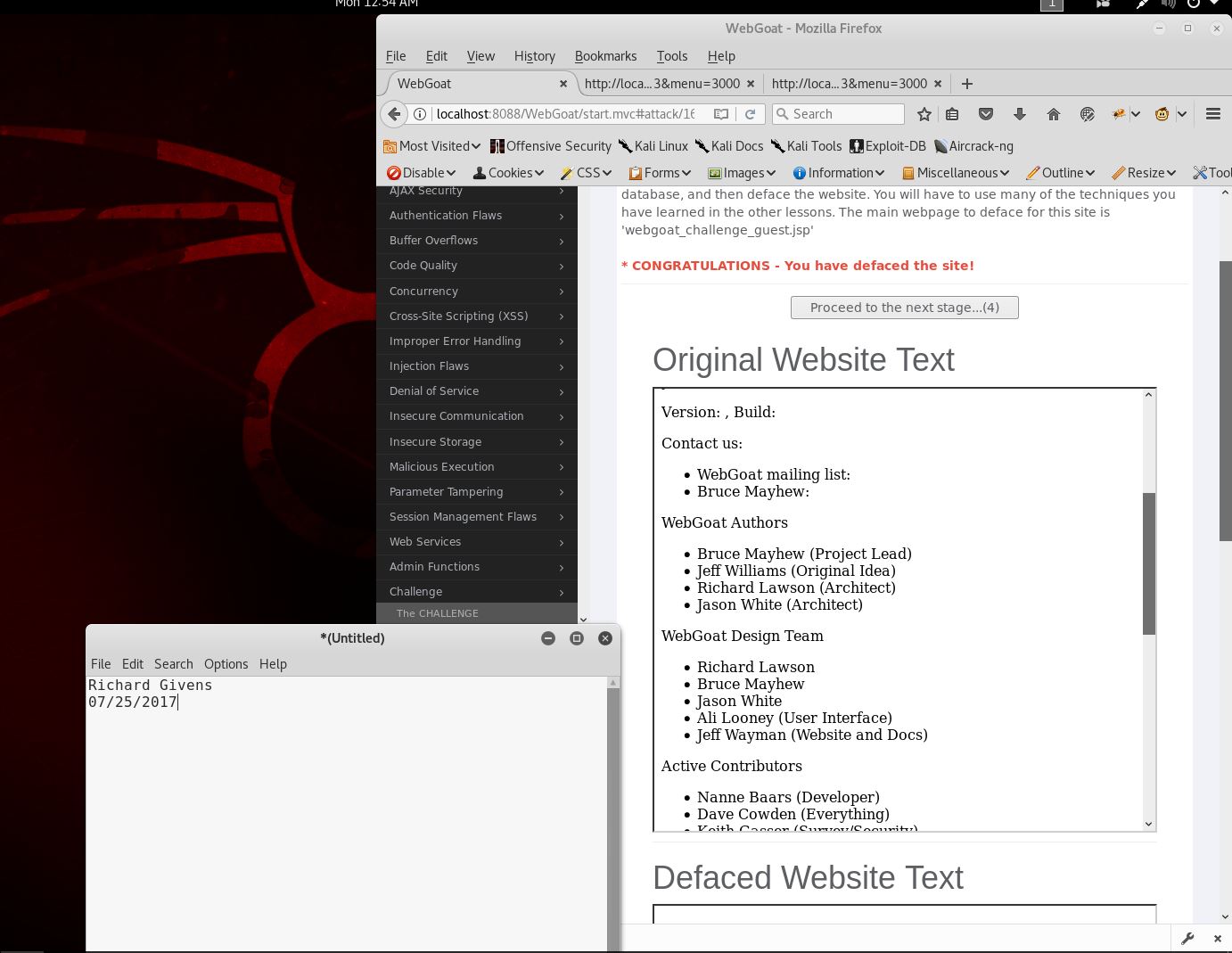


As you can see from the congratulations message, this attack will succeed.

1. While on the Intruder Attack box, highlight the successful request, right click and select “Send to Repeater.” You won’t have to use Repeater in this case, but this is merely creating your custom POST Request. (Note, this method is different from the Copy/Paste method in Stage 2, and is meant to highlight another possible way of accomplishing the same task.)



1. Though you are only injecting into the original “tcp” listed after&File=, go ahead and highlight all the data in the Raw section of Repeater, right click, copy, then paste back into your Proxy. Forward the traffic (you may have to toggle intercept off before it updates). After this, the webpage should update showing the Congratulations message. Proceed to Stage 4.0.



1. Stage 4 is merely the congratulations message, and the obligatory warning to never use the knowledge gained for illegal purposes. The screenshot below is proof of completion, with a text document merely confirming that this is original work.

