Web Application Penetration Testing With Burp Suite

CS 498 - Capstone Project

John Happel

Table of Contents

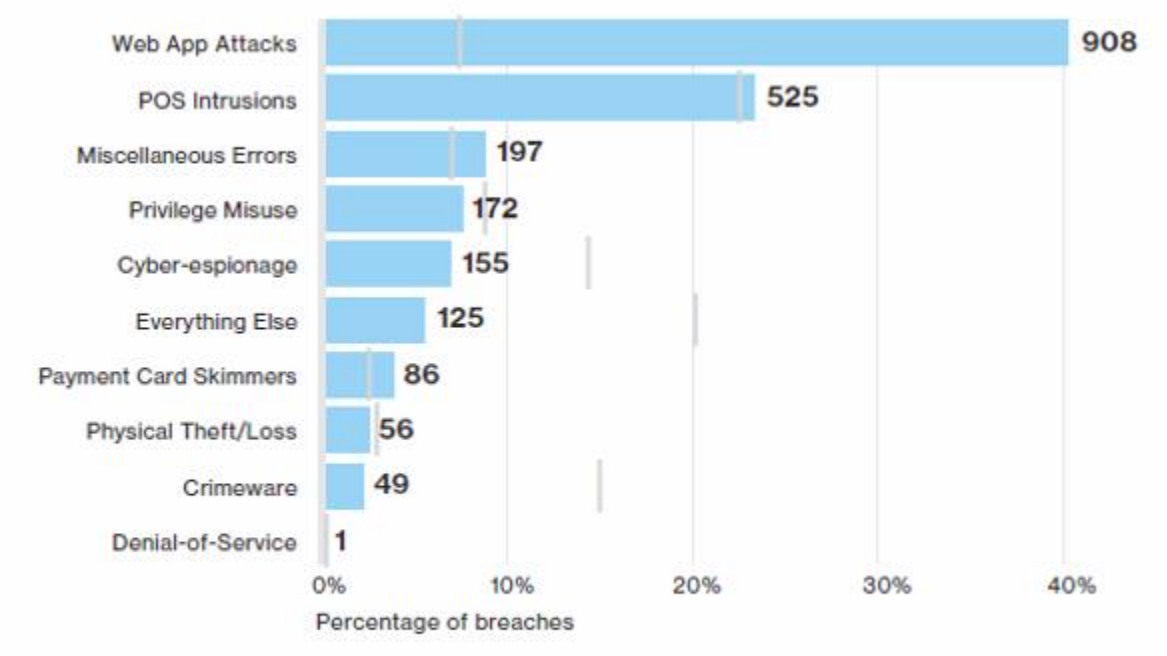
Introduction

Capstone Purpose

This capstone is intended to educate both students who have an interest in creating web applications and students interested in penetration testing. For the students interested in web application development, the information contained in this capstone will open their eyes to the risks and vulnerabilities that may arise if safe coding practices are not being adhered to. The labs contained in this capstone will introduce methodologies, techniques, and the usage of Burp Suite to students who may show an interest in penetration testing. Being that there are many risks and vulnerabilities to web applications, this capstone is only meant to be an introduction and not an exhaustive study. This area of computer security is extremely important as web applications are the primary interface between online businesses and their customers. If web applications are riddled with vulnerabilities, both businesses and customers are at risk of a data breach.

Consequences of Vulnerable Web Apps

The Verizon 2016 Data Breach Investigations Report lists web application attacks as a little less than 10 percent of all 2016 incidents. However, web application attacks made up 40 percent of all reported 2016 data breaches [1].



Verizon, 2016

The Internet has been growing at an exponential rate since the 1990s. As it grows, its complexity also evolves. In the early years of the Internet, web pages were static. As web technologies became more advanced, businesses and organizations coded websites to be more dynamic. Today, web applications can be defined as a client-server software application that runs the client in the web browser. The application may store data in a database located server side, manipulate data through scripting, and even transfer data to other clients over the network [2].

Vulnerability of Web Apps

There are several reasons why there are a lot of vulnerable web applications today. One reason is that web applications can be the driving force for earning a company profit. Because of this, there may be pressure on development teams to ensure that the web application is delivered by a deadline. This may cause careless

coding techniques, which may render the application vulnerable to a world full of hackers. Focus on delivering a functioning product instead of a secure functioning product has led to a lot of data leakage across many web applications. Another reason for the existence of web app vulnerabilities is training [3]. Inexperienced developers may rely on code reuse. If the code that they are borrowing is insecure, their web app will also be insecure. All developers should be trained in secure coding techniques to eradicate vulnerabilities from their code. Continual education in secure coding techniques is also paramount to preventing data breaches via their web applications. Institutions of higher learning should require developers to take courses in secure coding techniques and web application testing so that they can churn out competent developers who are able to identify and resolve security vulnerabilities in existing code, as well as develop secure future code.

OWASP

In 2001, the Open Web Application Security Project (OWASP) was founded by Mark Curphey to make trustworthy computing the highest priority for developers. The OWASP Foundation is a not-for-profit international organization and open community which exists to enable organizations to design, develop, operate, and maintain trustworthy applications. All documents, forums and tools created by OWASP are free to all who are interested in securing their applications [4].

One of the projects that OWASP is famous for is the OWASP Top Ten Project. It was first published in 2003. The OWASP Foundation collects information from organizations concerning critical risks and vulnerabilities which those organizations have encountered. OWASP then compiles that information into an in-depth report on

the top ten risks faced for that particular year. Two major reports that can be found on the OWASP web site are the OWASP Top Ten for 2010 and the OWASP Top Ten for 2013. The project is currently calling on organizations for data for 2017. Within the reports, you can find a ranking of the top ten risks, a description of the risks, information of determining if you are vulnerable to the risk, examples of attack scenarios for the particular risk, information on prevention, and references to additional information provided by OWASP and third parties. In addition to these top ten risks, OWASP encourages developers to continue beyond these top ten risks as there are hundreds of issues plaguing the Internet and web application development [5]. Other top publications provided by OWASP are the OWASP Developer's Guide, the OWASP Testing Guide, the OWASP Code Review Guide, and the OWASP Cheat Sheet Series [6].

Two other projects that OWASP is known for is WebGoat and OWASP ZAP Project: The Zed Attack Proxy (ZAP) [7]. WebGoat is a deliberately insecure web application created to teach secure coding practices. Webgoat comes with a tutorial and a collection of different lessons to teach students how to exploit web application vulnerabilities. Students can then apply the lessons that they learned toward making more secure web applications. ZAP is an easy to use penetration testing tool for web applications. It was designed to be used by beginners and developers alike.

OWASP also created the Mutillidae and Mutillidae 2 projects [8]. They are similar to WebGoat. The main difference between these applications is the environment. WebGoat contains the Java Runtime Environment and a preconfigured Tomcat instance so that all that you have to do is unzip the compressed file and execute the script to

launch WebGoat. Mutillidae and Mutillidae 2 have to be installed on an existing web server with XAMPP. It uses Apache, MySQL, and PHP. Another popular open source vulnerable web application is Damn Vulnerable Web Application that was started by ethickalhack3r on GitHub. Like Mutillidae, it has to be installed on an existing web server [9].

bWAPP

One web application that I found would be perfect for students just getting into web application security is the buggy web application or bWAPP. bWAPP is unique as it contains over 100 web vulnerabilities. It contains all risks from OWASP Top 10 project as well as all major web bugs that have been seen over the years. It is a PHP application that uses a MySQL database, and it must be installed on a web server. It can be installed with XAMPP or WAMP. An even easier installation option is bee-box, which is a custom Linux VM that already has bWAPP installed on it [10]. I found that this was the easiest option to set up a lab environment for students to practice penetration testing on a web application.

Burp Suite

A company called PortSwigger Web Security created Burp Suite, which is an invaluable tool for the penetration testing of web applications. It is a Java based platform of tools, so it can be installed on different operating systems. Burp Suite comes bundled with Kali Linux and Kali Linux 2.0. It is free to use, but some functionality such as the scanner only exists in the Professional version, which costs about $350 at the time of this writing [11]. Yes, a web developer or web security professional can perform pentesting on an application by making a change to a parameter in a URL, sending the

request, checking the response, changing it again, sending the request, checking the response, etc. However, wouldn't it be much easier to just have a tool enumerate test parameters and check the response automatically for you? That is just one of the functionalities of Burp Suite. You will experience the other functionalities that Burp Suite has to offer as you go through the labs.

Lab Creation Process

My process for developing the labs included watching tutorials performed on other vulnerable web applications such as WebGoat and Damn Vulnerable Web Application, and then trying to apply those techniques to bWAPP. Some techniques I was not successful in duplicating on bWAPP. It appears that bWAPP simply displays the cookie of the session. Burp Proxy captured the cookie easily enough, but I had problems hijacking the session after stealing the cookie. This was due in part to the way that I set up the environment. I ensured that the victim VM and attacker VM were not connected to the Internet for security purposes by changing the network setting of each to internal network. Opening up another instance of the browser and connecting to bWAPP simply continued the same session that was already established. I then installed Chrome on the attack machine and tried hijacking the session with the stolen cookie but for some reason, I was unable to establish the session. So, in that lab, I avoided talking about session hijacking. However, if you would like to practice session hijacking, this can be done on the Damn Vulnerable Web Application. Overall, I feel that these labs will give students exposure to performing penetration testing on a vulnerable web application, and help them to think about developing applications with security in mind.

Conclusion

A series of labs have been created to introduce students to using Burp Suite tools to detect the most prevalent vulnerabilities found in web applications as per the 2013 OWASP Top Ten Project. The first three labs walk students through setting up the virtual test environment and configuring Burp Suite. Labs four through seven guide students through detecting and attacking the top four categories of vulnerabilities that were reported in 2013 according to the OWASP Top Ten Project. Lastly, lab eight demonstrates further functionality of Burp Suite that may be useful in detecting and attacking vulnerable web applications. As stated by the creator of bWAPP, there are over 100 vulnerabilities in bWAPP alone, so these labs only touch the tip of the ice berg when it comes to web application security. With that being said, continue to educate yourself so that you may have a positive impact on the industry of web application development and security.

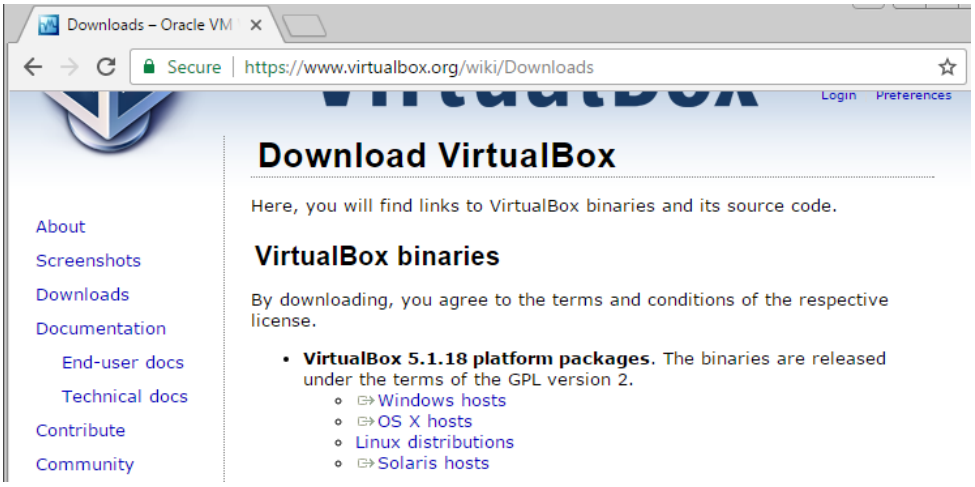
Lab 1: Setting up the attack VM

1. Introduction

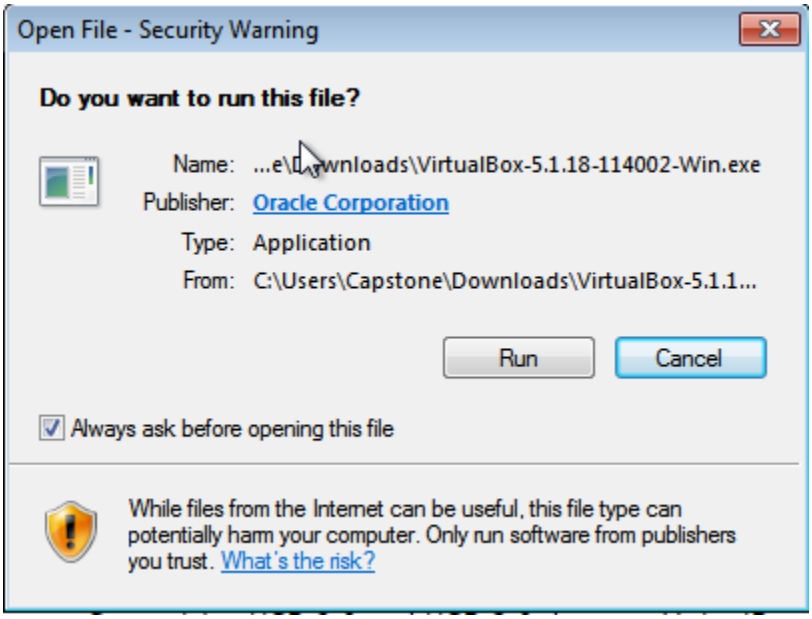
The main objective of this lab is to install VirtualBox, install a Kali 2.0 virtual machine (VM) on VirtualBox, and configure Kali 2.0 to prepare for the attack. Kali 2.0 contains many penetration testing tools, including Burp Suite which will be used in subsequent labs for the purpose of finding vulnerabilities in web applications. In the next lab, we will install our victim virtual machine that will contain the vulnerable web application that we will be scanning for vulnerabilities.

2. Installing VirtualBox

Navigate to https://www.virtualbox.org/wiki/Downloads and click on Windows hosts under VirtualBox 5.1.18 platform packages.

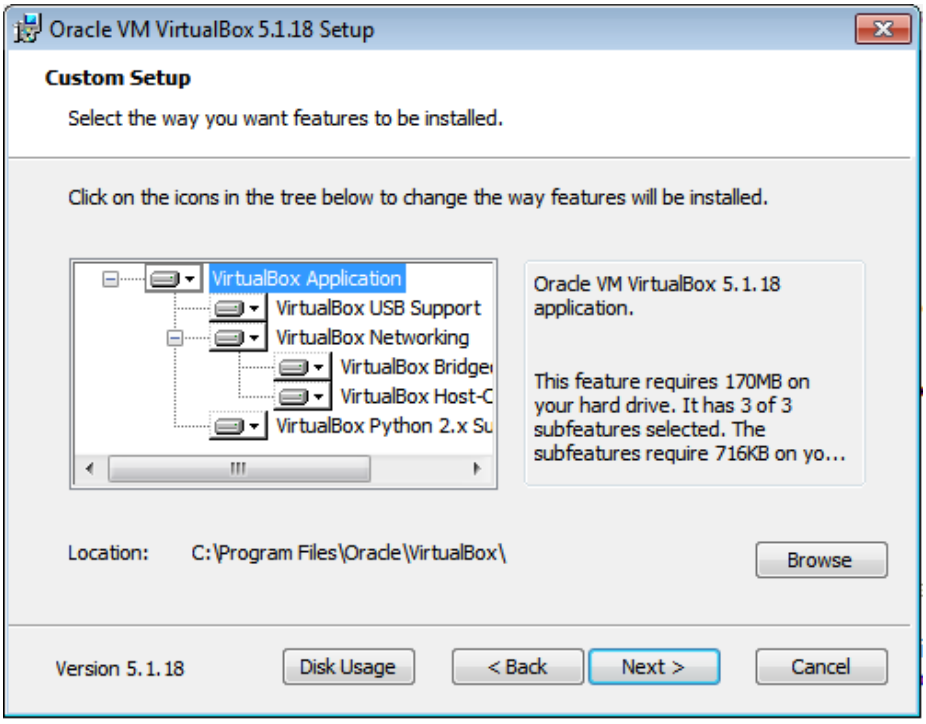


Under VirtualBox 5.1.18 Oracle VM VirtualBox Extension Pack, Click All supported platforms. After it downloads, double click on it, click Install, and agree to the license.

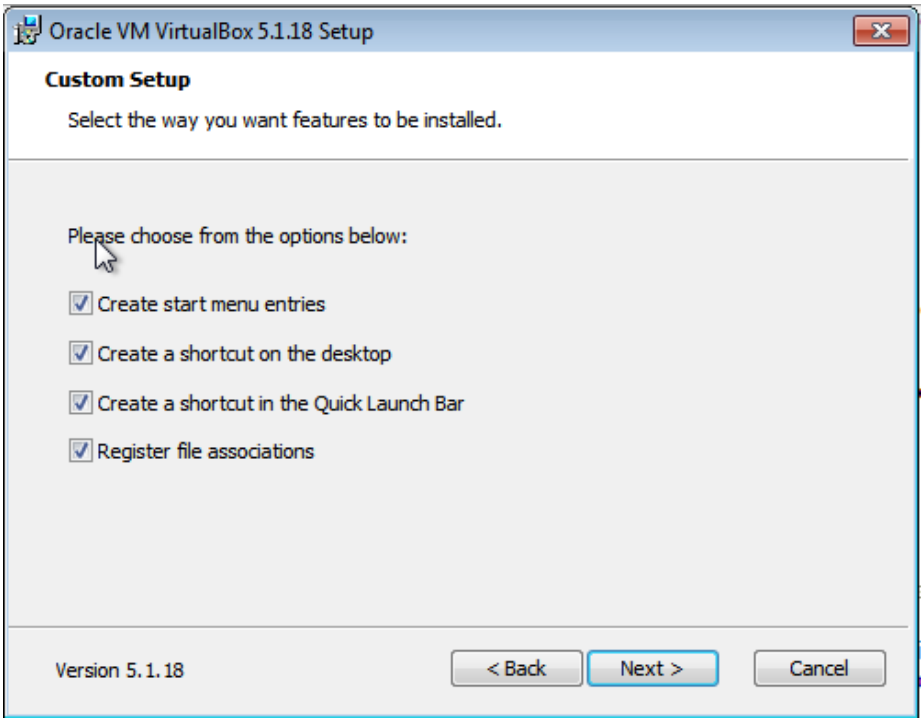


After it finishes downloading, double click on the executable and then Click Run.

Click Next.

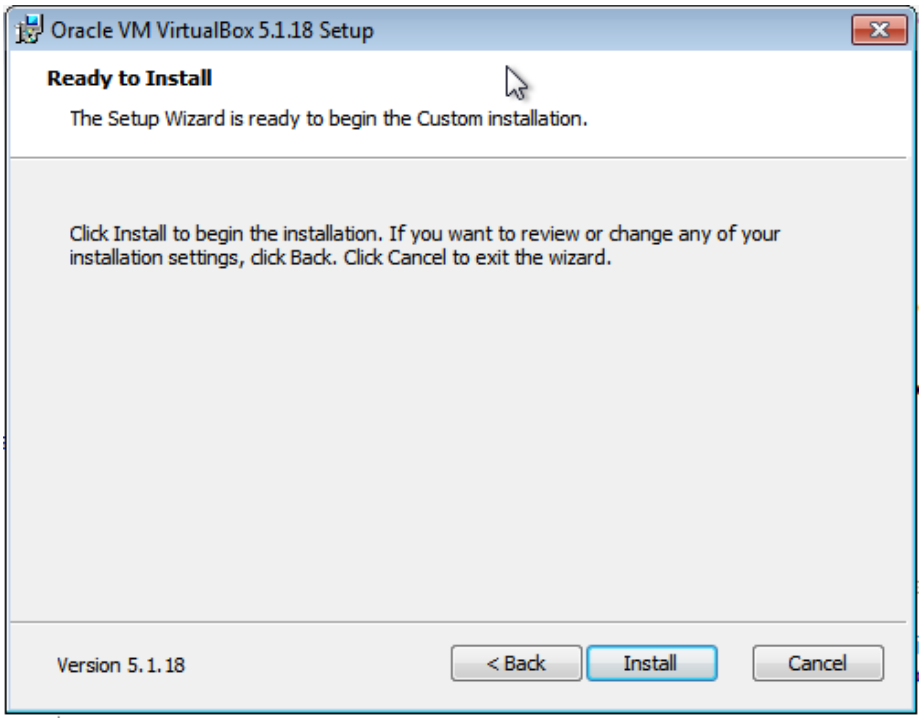


Click Next.

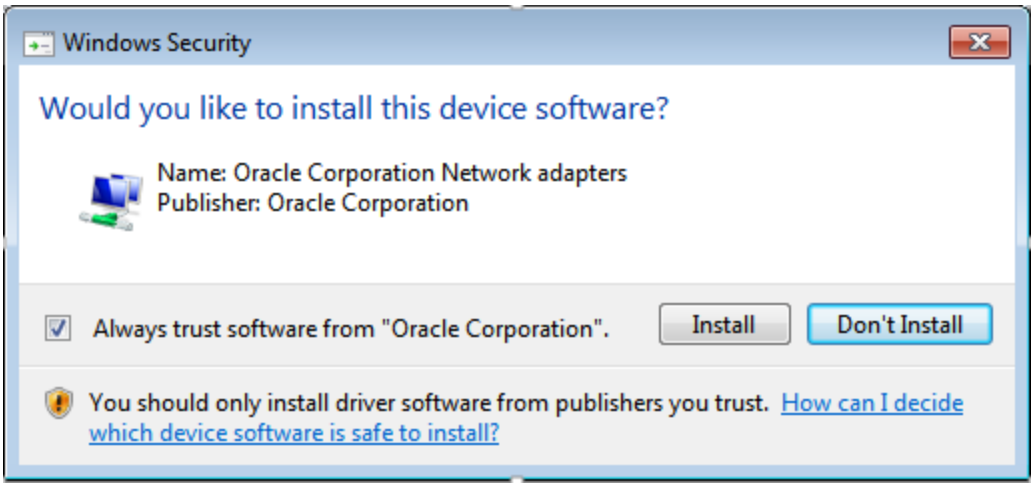


Click Next.

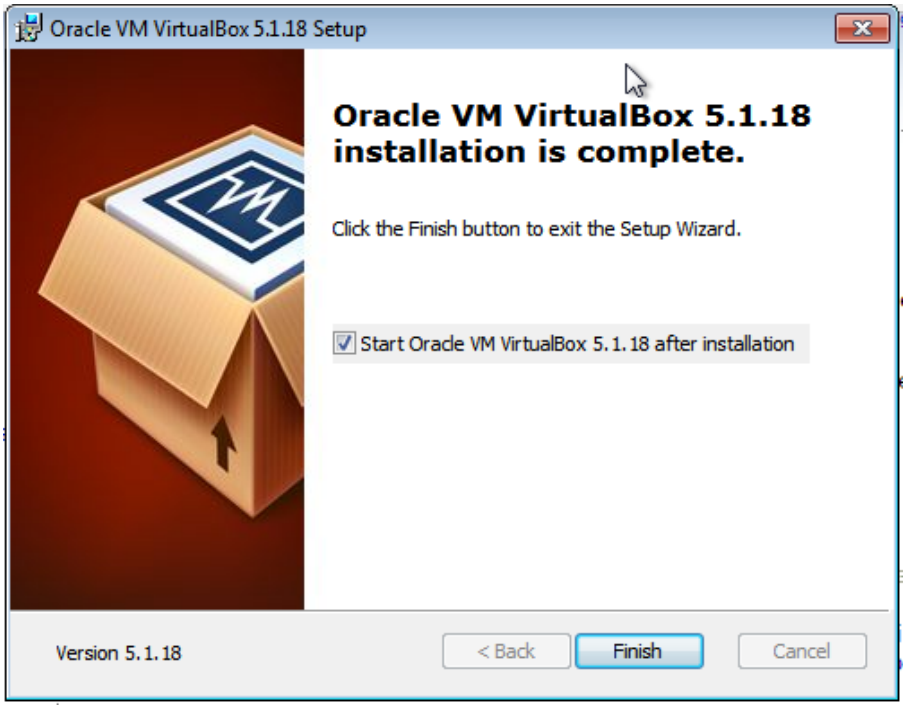
Click Yes.



Click Install.



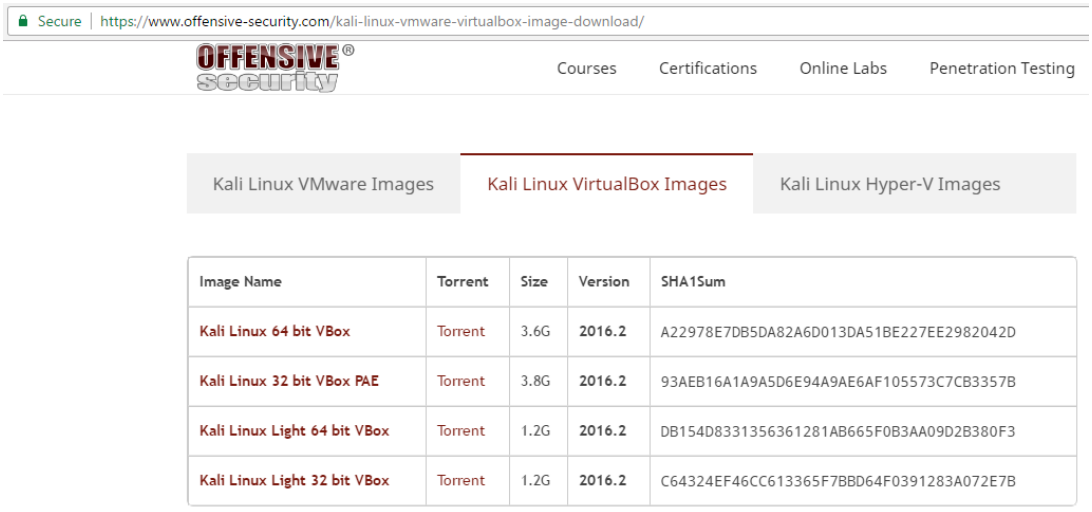
Check Always trust software from “Oracle Corporation” and then click Install.



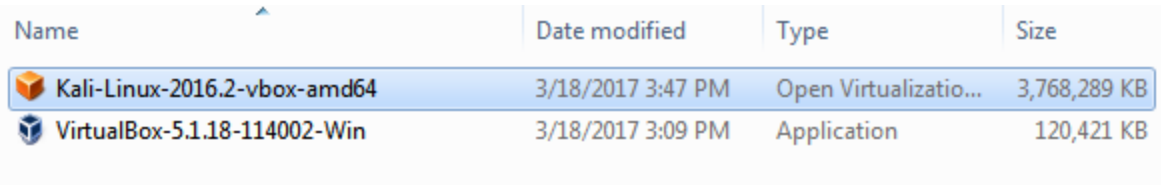
Click Finish.

3. Installing Kali Linux 2.0

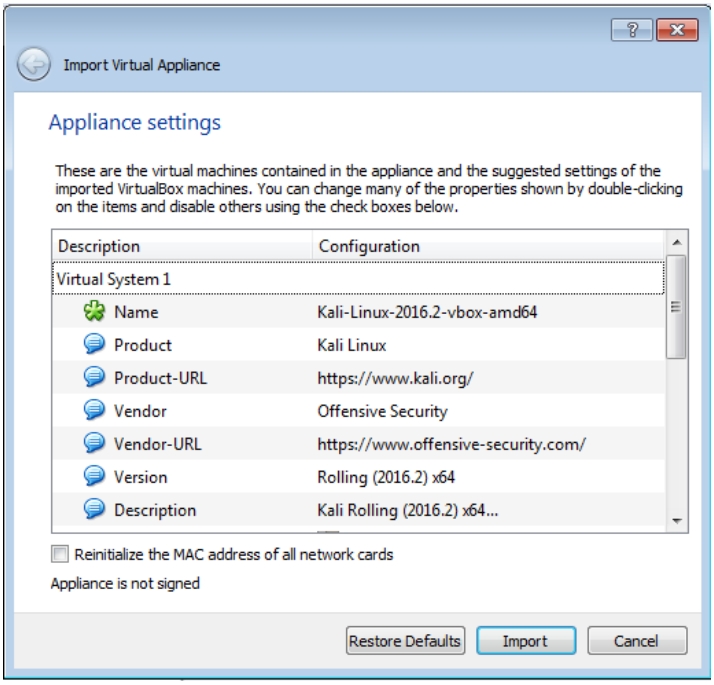
Navigate to https://www.offensive-security.com/kali-linux-vmware-virtualbox-image-download/.



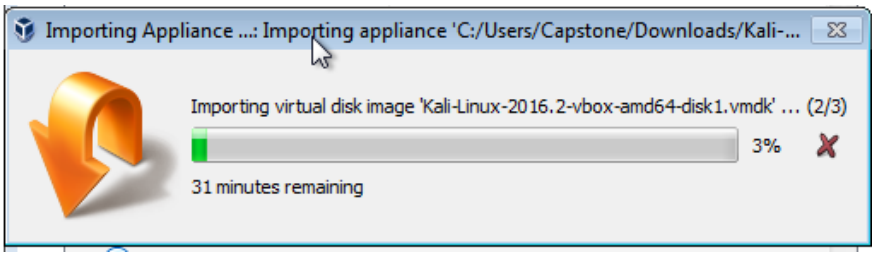
Choose the image that is compatible with your version of VirtualBox.



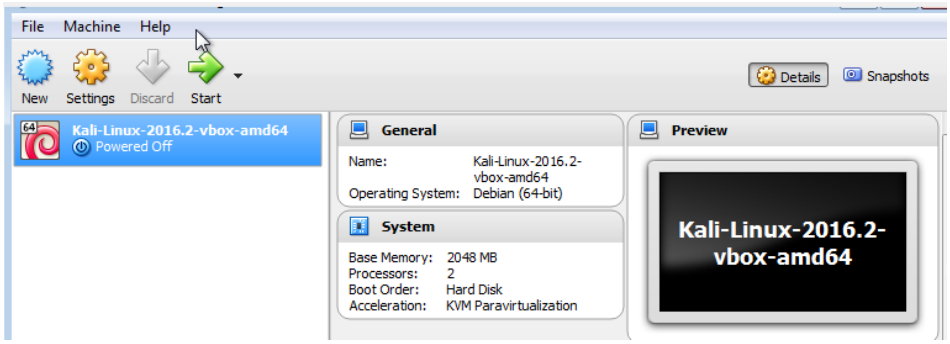
Double click on Kali-Linux-2016.2-vbox-amd64.



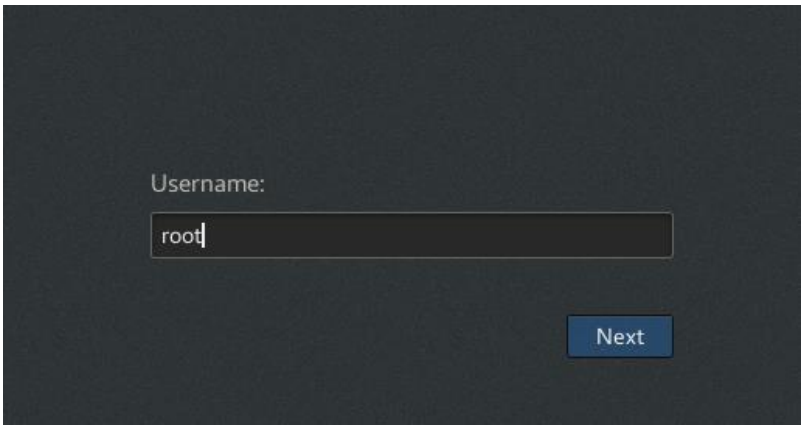
Click on Import.



It may take a while for the image to be imported into VirtualBox.

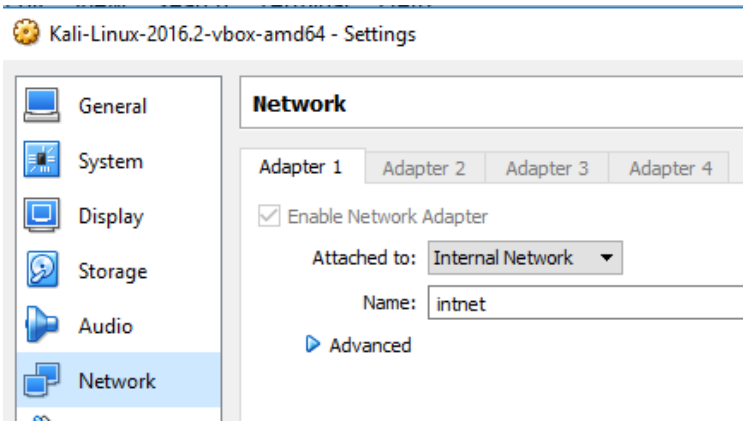


After it imports, select the Kali Linux VM and click Start.

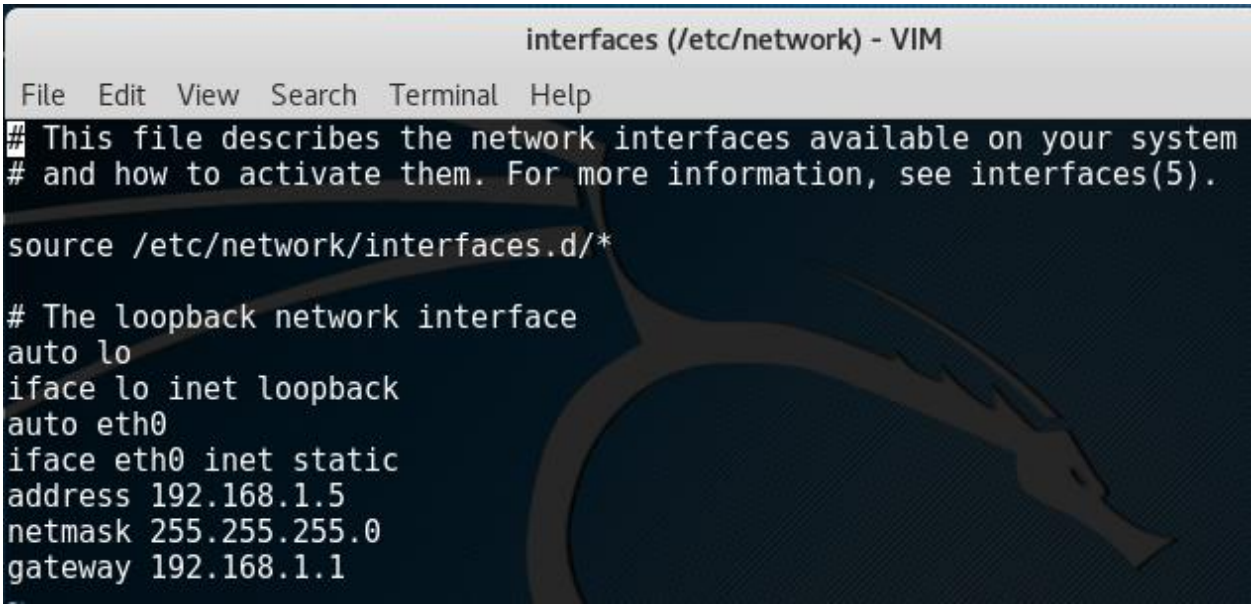


Username: root

Password: toor



Go into Settings and change Network settings to attach to Internal Network.



Run **vim etc/network/interfaces** and add the following to the end of the file:

auto eth0
iface eth0 inet static
address 192.168.1.5
netmask 255.255.255.0
gateway 192.168.1.1

Save the file and then restart the networking services by running:
services networking restart

You are now ready to configure the victim VM.

Lab 2: Setting up the victim VM

1. Introduction

Buggy web application, here on out referred to as bWAPP, is an open source web application that is intentionally insecure with over 100 web vulnerabilities. bWAPP contains all of the major web vulnerabilities as well as those listed in the OWASP Top 10 Project. bWAPP is a very valuable tool for the use of web application security-testing and is intended for educational purposes only. Do not install this on a production server.

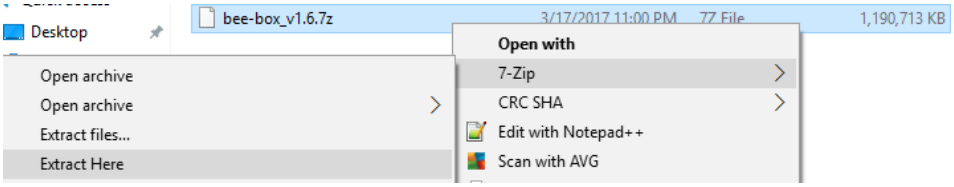
bWAPP can be installed with WAMP or XAMPP. It also can be hosted on Windows or Linux.

An even better option for this lab is to install bee-box which is a preconfigured VM that you can manipulate and gain root access to. The web platform used on the bee-box is LAMP based, meaning that it uses Linux for the operating system, Apache for the web server, MySQL for the relational database management system, and PHP for the object-oriented scripting language [10].

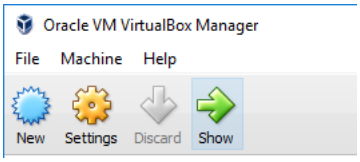
2. Installing Bee-Box

Navigate to https://sourceforge.net/projects/bwapp/files/bee-box/. Click on bee-box_v1.6.7z to download the bee-box.

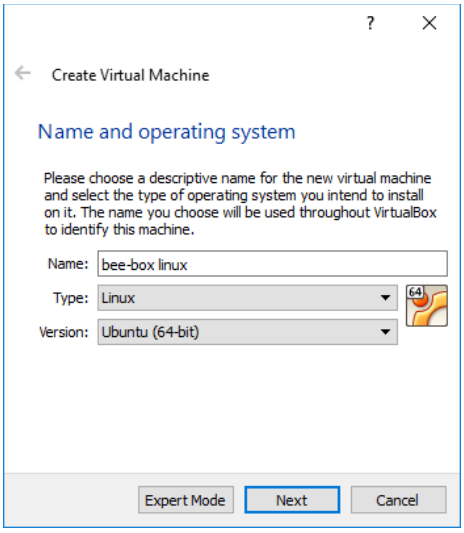
If you don’t already have 7-Zip, download the version of 7-Zip compatible with your system from http://www.7-zip.org/download.html.



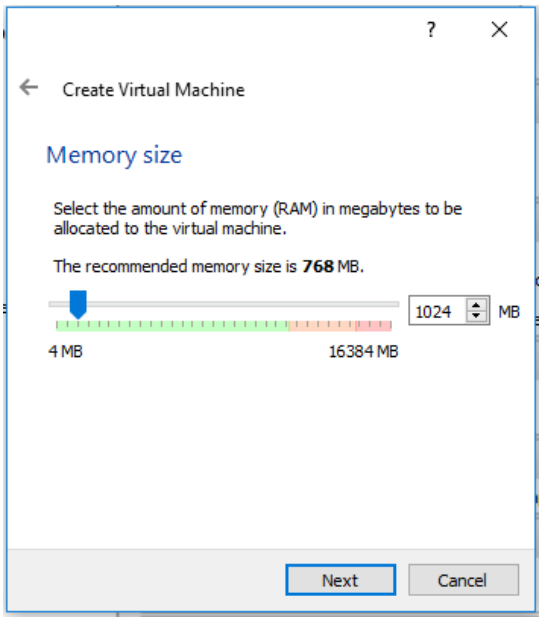
Extract bee-box_v1.6.7z with 7-Zip.



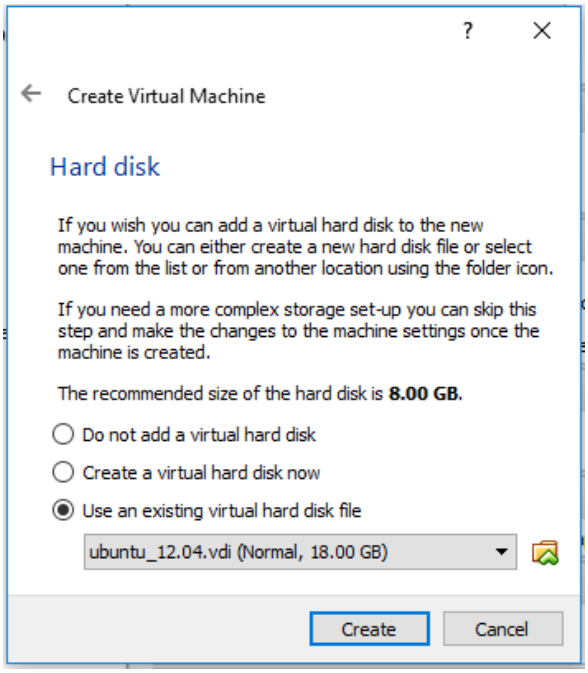
Click on New.



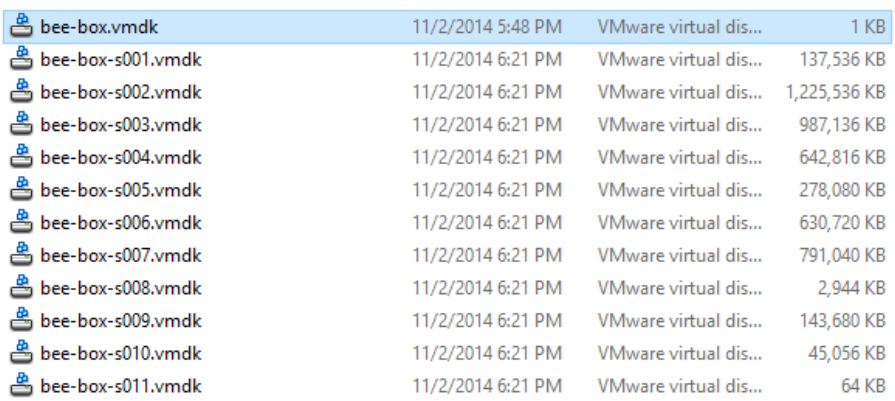
Enter a name for the VM, select Ubuntu (64-bit) for version and click Next.



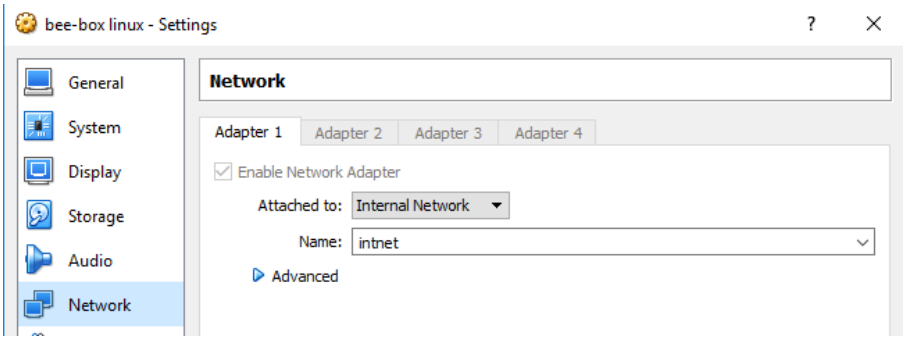
Select 1024 MB of RAM and click Next.



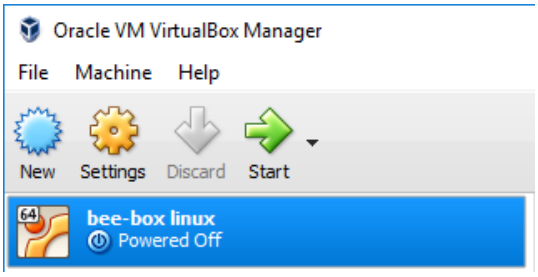
Click Use an existing virtual hard disk file then click on the folder icon to browse to the bee-box images.



Select bee-box.vmdk, click Open, and then click Create.

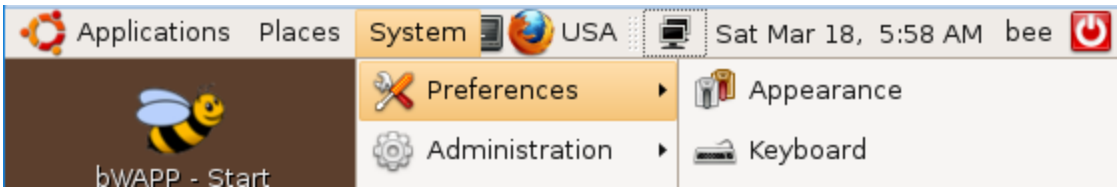


Change the network settings to Internal Network on your Kali box and bee box.

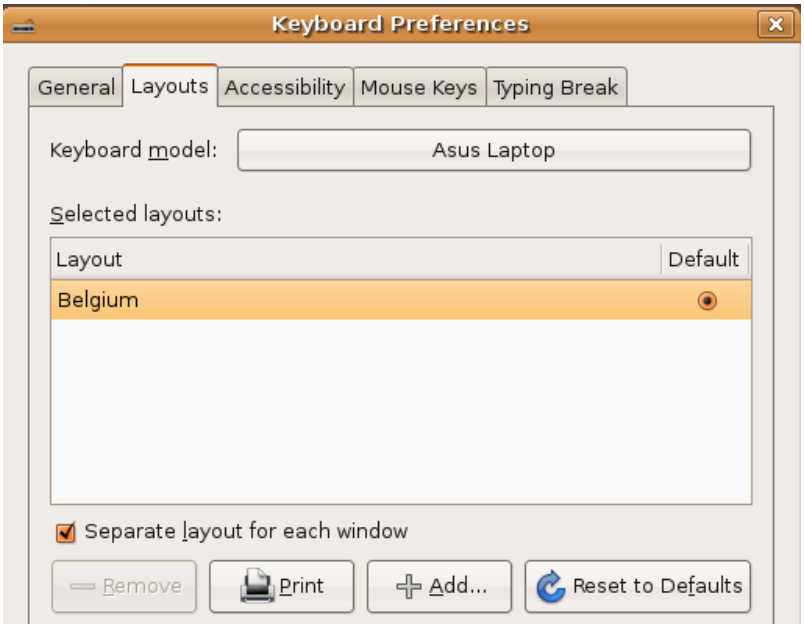


Click on the bee-box VM and then click Start.

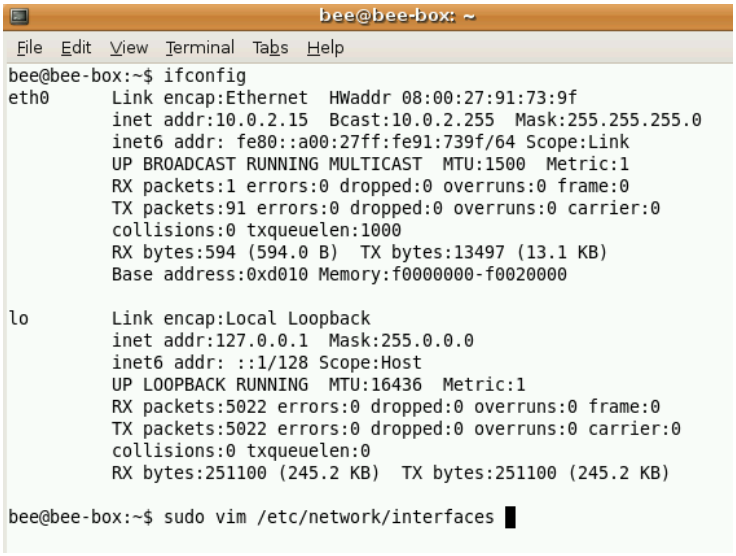
Once bee-box is loaded up, you are going to want to change the keyboard layout as it is set to Belgium by default.



Click System > Preferences > Keyboard



Click on the Layouts tab, click on the Add button below, select USA for Layouts, then click Add. When that window closes, set USA to default and close out of the window. Sometimes the keyboard layout reverts back. If this happens, just delete the USA Layout and re-add it.



Open a command prompt and check the IP address with the ifconfig command.

If it is not on the same network as your Kali box, configure it by first typing:

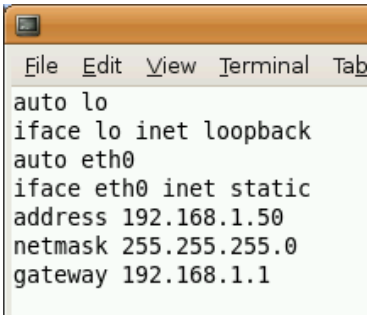
sudo vim /etc/network/interfaces

The password is bug.

In the interfaces file add:

auto eth0
iface eth0 inet static
addresss 192.168.1.50
netmask 255.255.255.0
gateway 192.168.1.1

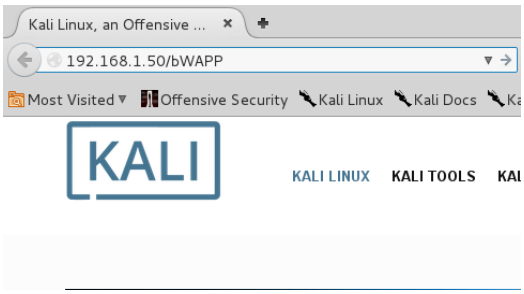
Example:



Then reboot the machine for the changes to take effect because this is an older version of Ubuntu that will not let you restart networking with the **services networking restart** command.

Verify that you are able to ping between your Kali box and your bee-box.

In the browser on your Kali box, type the IP address of your bee-box followed by '/bWAPP' without the quotes.



Your buggy web application is now up and running.

Lab 3: Configuring Burp to attack bWAPP

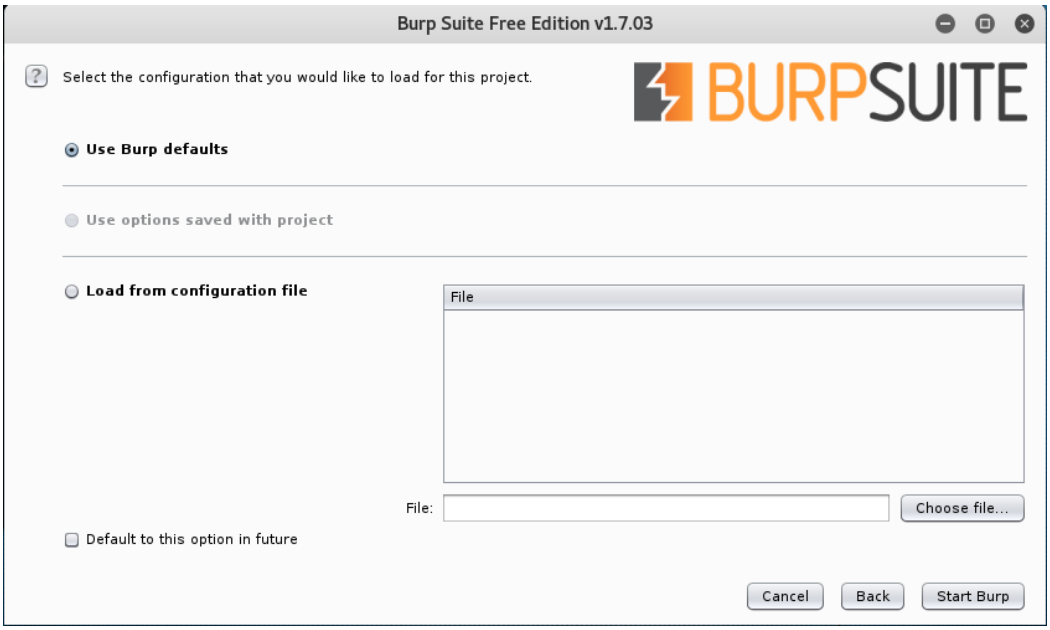
1. Introduction

Burp Suite is one of the many tools that comes bundled with Kali Linux. It is a platform used for testing the security of web applications. Burp's many tools operate together to map a web application, analyze the attack surface of the web application, discover, and exploit security vulnerabilities. Burp Suite Free Edition is more of a demo version to entice users to purchase a professional version of this competitively priced security testing software.

The free version allows you to inspect and modify web traffic between the browser and the target application by utilizing the Burp Proxy. Burp Spider allows you to crawl the content and functionality of the target application. Burp Repeater allows you to make changes to requests and resend those changed requests. More automation and time saving tools are available for use with the Professional version [12].

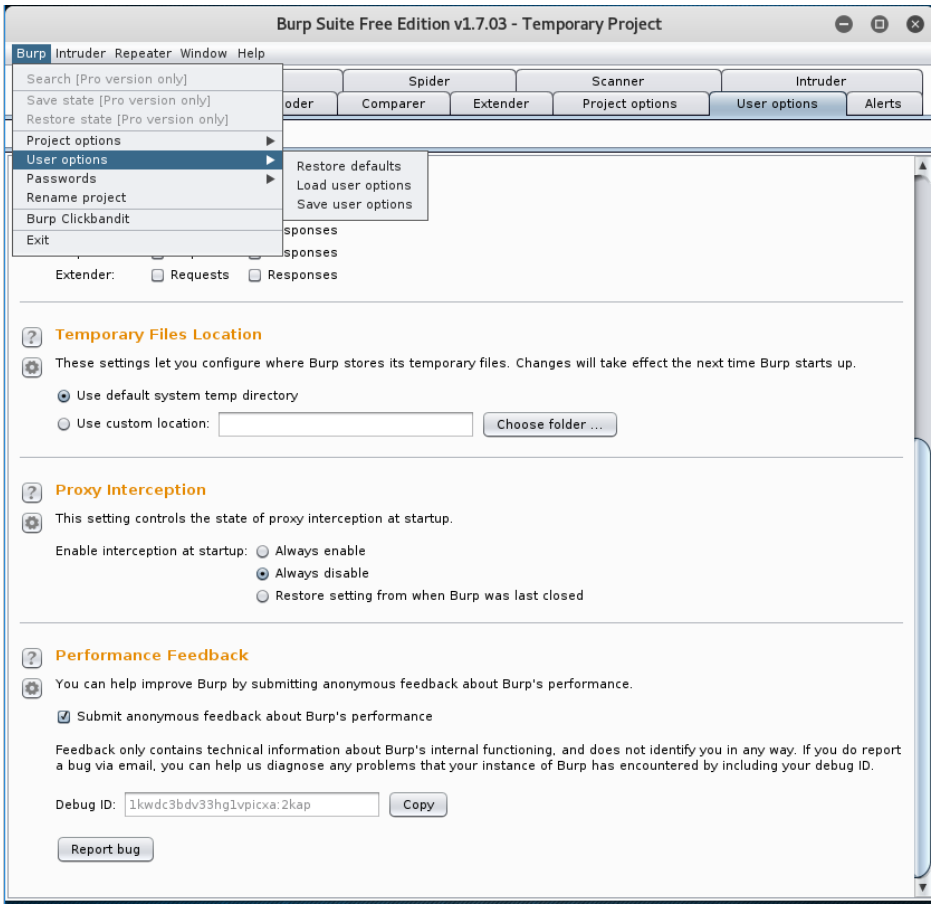
2. Configuring Burp

Open up Burp and click Next.

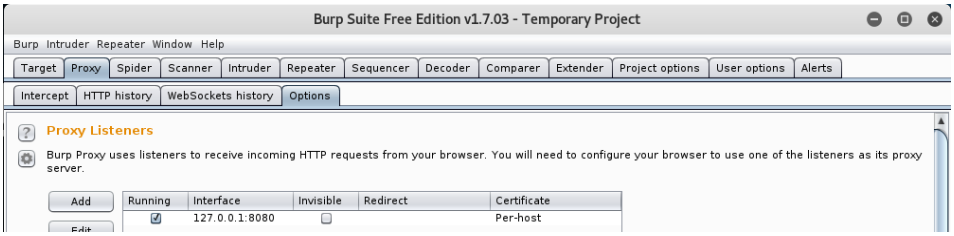


Keep the defaults and click Start Burp.

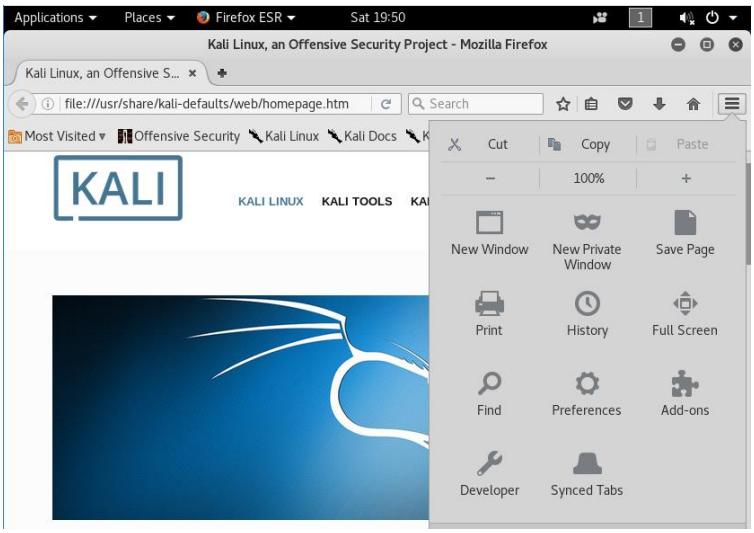
Click the Proxy tab and turn Intercept off.



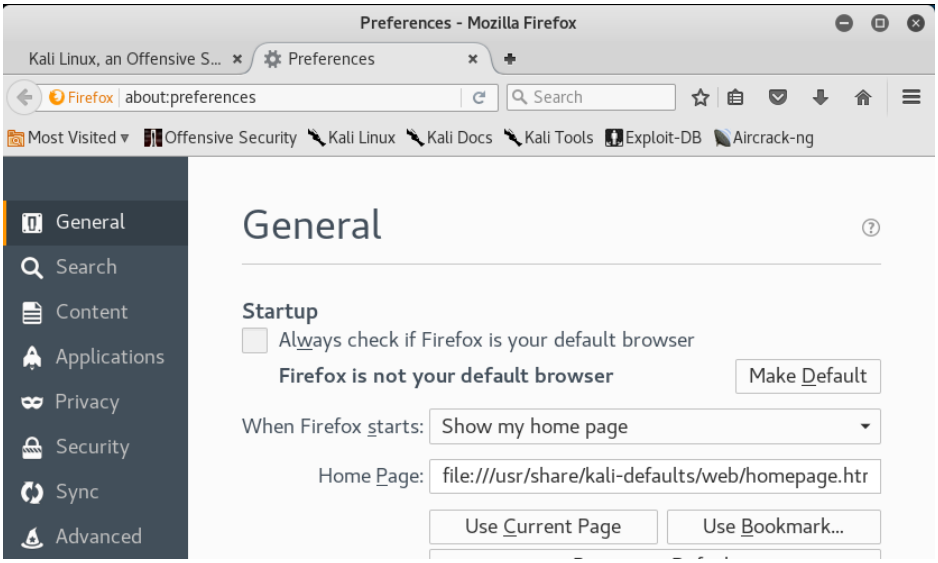
Intercept is on by default. To change this, go to the User options tab, scroll to Proxy Interception and click radio button Always disable. Then click on Burp menu > User options > Save user options. Then give the file a name and save it to a location of your choosing. You can then load that configuration when Burp starts up.



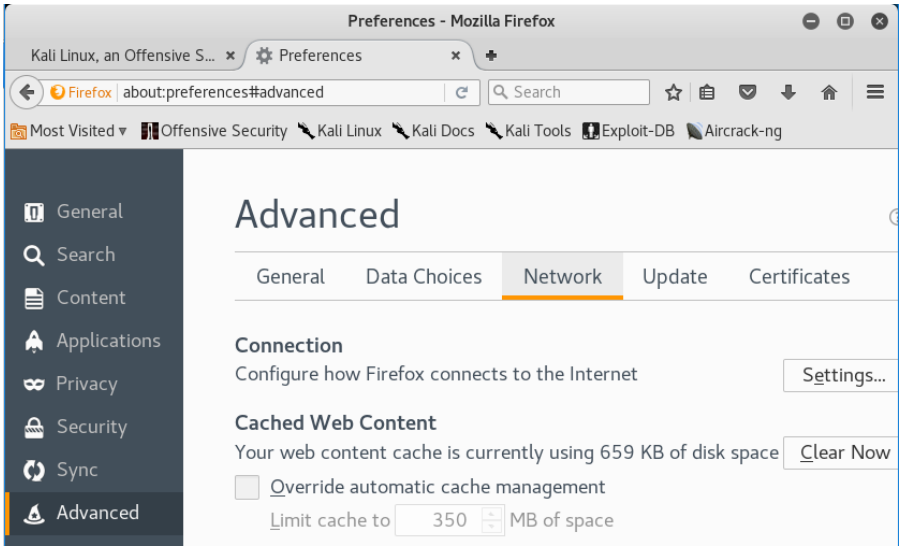
Click the Options sub-tab and ensure that the checkbox under Running is checked and that Interface is set to 127.0.0.1:8080.



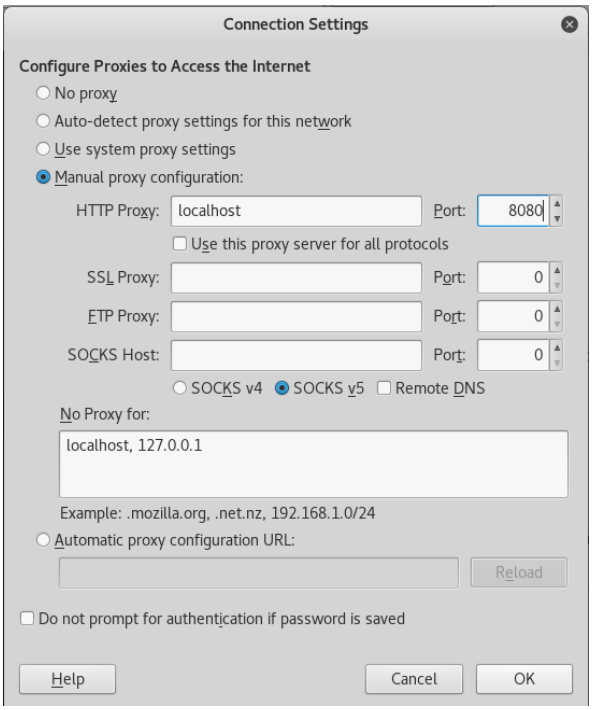
Open Firefox and click on Preferences.



Click the Advanced tab on the left.



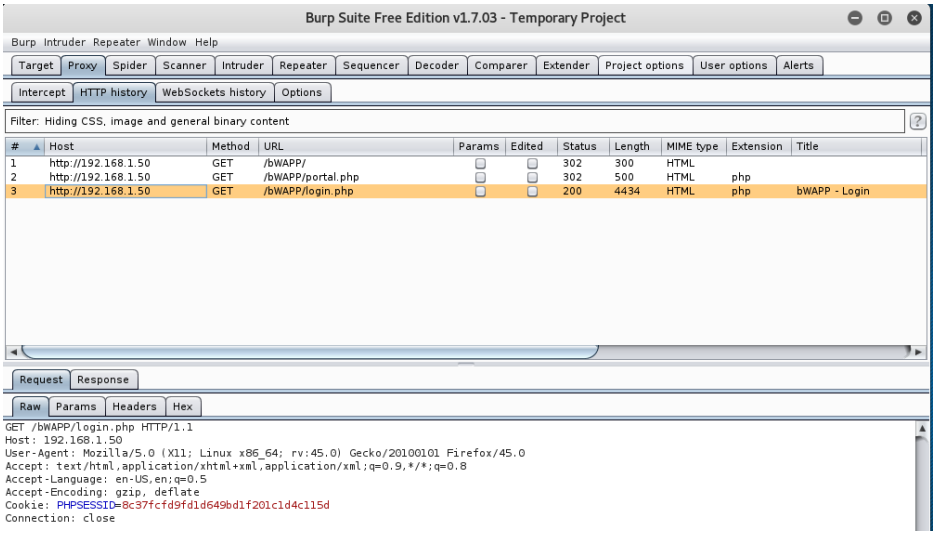
Select Network then click on Connection Settings.



Select Manual proxy configuration. Set the HTTP Proxy to localhost, set the Port to 8080, and click OK.

Navigate to 192.168.1.50/bWAPP. Switch back to Burp, select Proxy > HTTP history.

You should now see the request being analyzed by Burp.



You can also click on the Response tab to see the response of the web application.

Questions:

What is the IP address of local host?

What is a proxy?

What are the top 10 vulnerabilities in 2013 according to OWASP?

In your own words, explain the purpose of the Forward and Drop buttons under Proxy > Intercept.

Lab 4: SQL Injection

1. Introduction

Databases contain a lot of valuable information. This information could be usernames, passwords, credit card numbers, or other information that a black hat hacker would invest time in retrieving. If a database is not configured properly, a hacker would be able to send SQL commands to the web application to retrieve valuable information.

A SQL injection vulnerability can be simply detected by putting a single quotation mark in a text box that gets submitted to the database. If it throws some error stating that the SQL syntax is incorrect, then the web application has a SQL injection vulnerability.

Example: A web form has a user submit a username and password, and the user types Joe for username and Cats123 for password. When the user hits the login button, his input gets sent to the database where the following query is performed:

SELECT * FROM Users WHERE Name = 'Joe' AND Pass = 'Cats123';

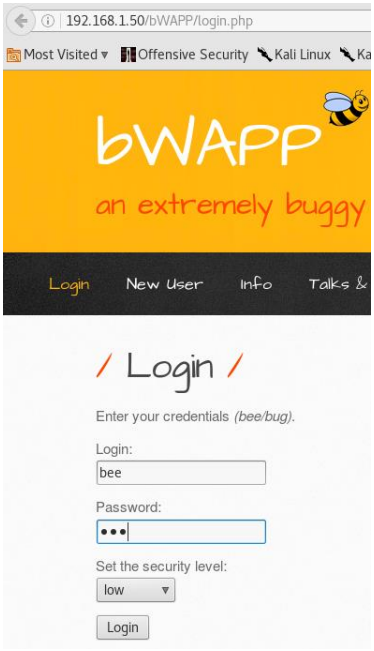
An error gets thrown if the user provided username or password are not present. However, if the web application is not programmed securely, a hacker could input username as (' or 1 = 1 --) without the parentheses. For password, the hacker would also enter (' or 1 = 1 --) without the parentheses. The database would then perform the following:

SELECT * FROM Users WHERE Name = '' or 1 = 1 AND Pass = '' or 1 = 1;

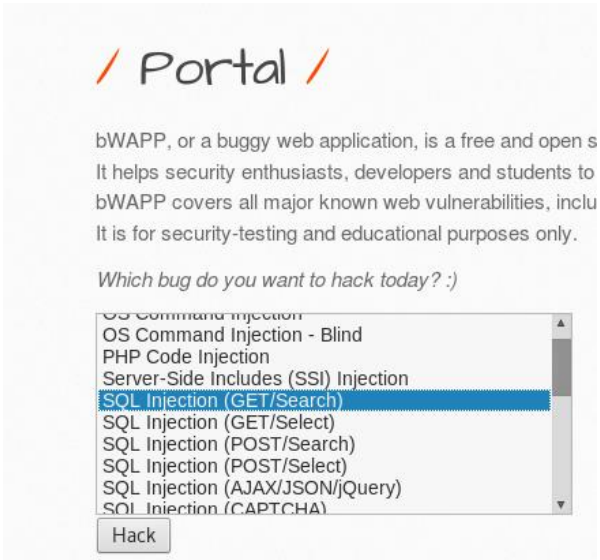
This is a valid query that will return all rows from the Users table because **WHERE 1 = 1**

is always true.

2. Prepare bee-box for SQL Injection Attack



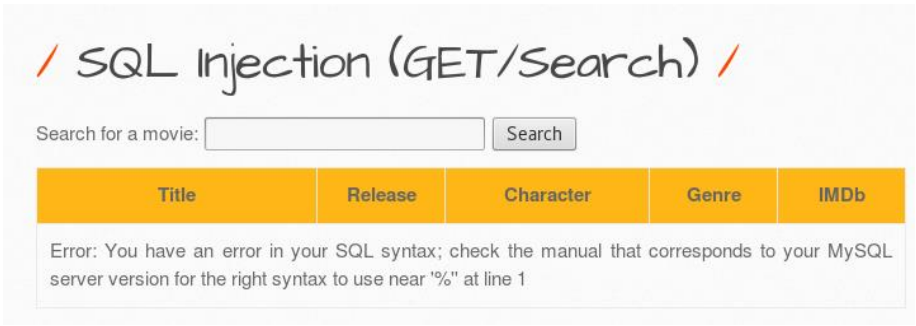
Log into bWAPP with Login of bee and Password of bug.



Scroll down a little, select SQL Injection (GET/Search), and click Hack.

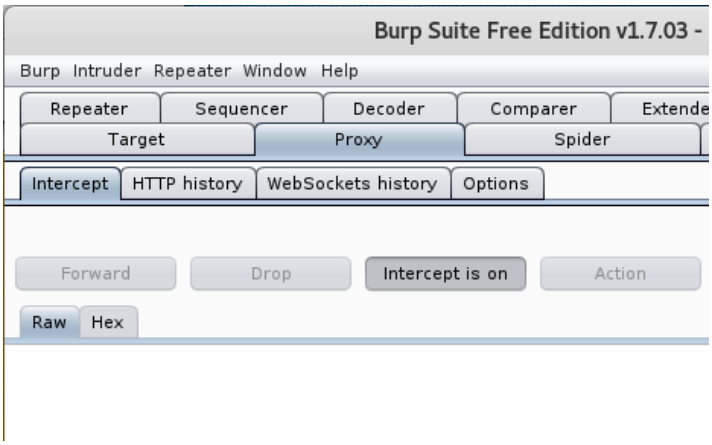
3. Check for SQL injection

Enter a single quote (‘) in the search box and click Search.



This error indicates the presence of a SQL injection vulnerability.

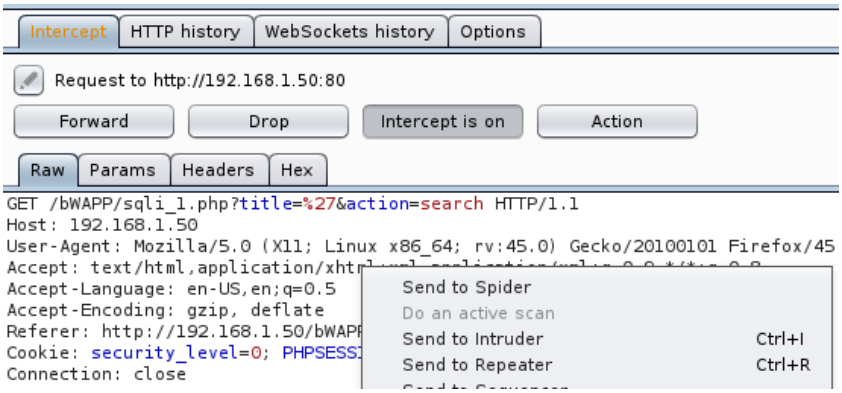
4. Prepare Burp for SQL Injection Attack



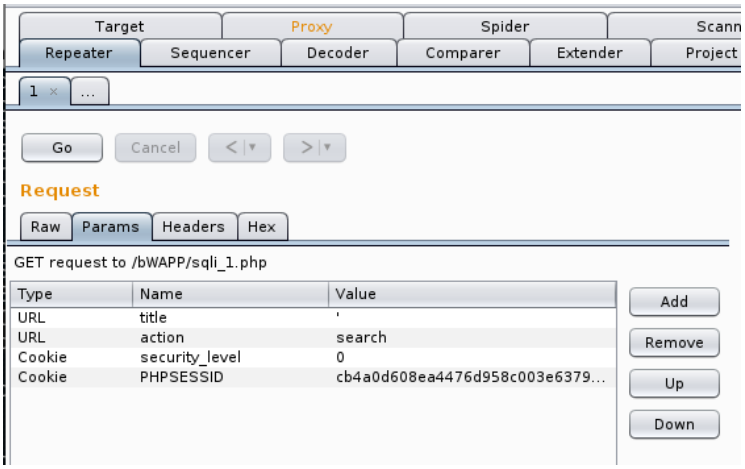
In Burp, click on the Proxy > Intercept tabs, then turn Intercept on.

5. Perform attack and capture request with Burp

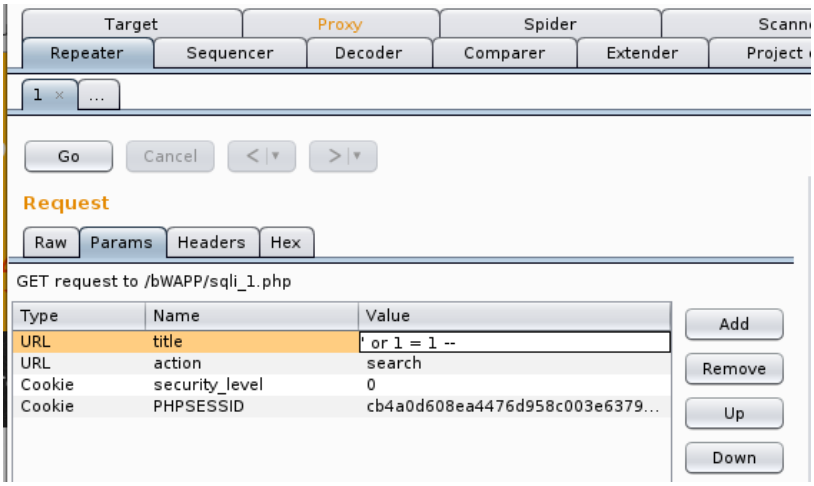
Enter a single quote (‘) in the search box and click Search.



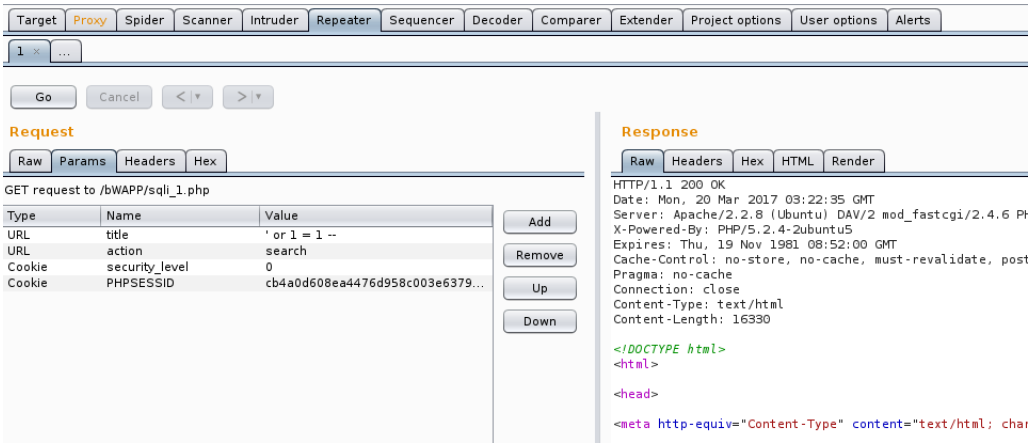
Right click on the raw text that was captured and then click on Send to Repeater.



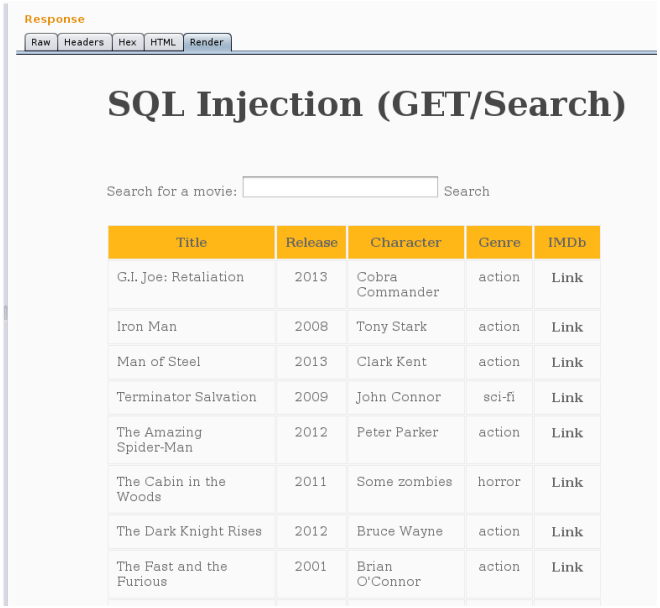
Click on the Repeater > Params tabs. Notice the first row has a name of title and a value of ‘.



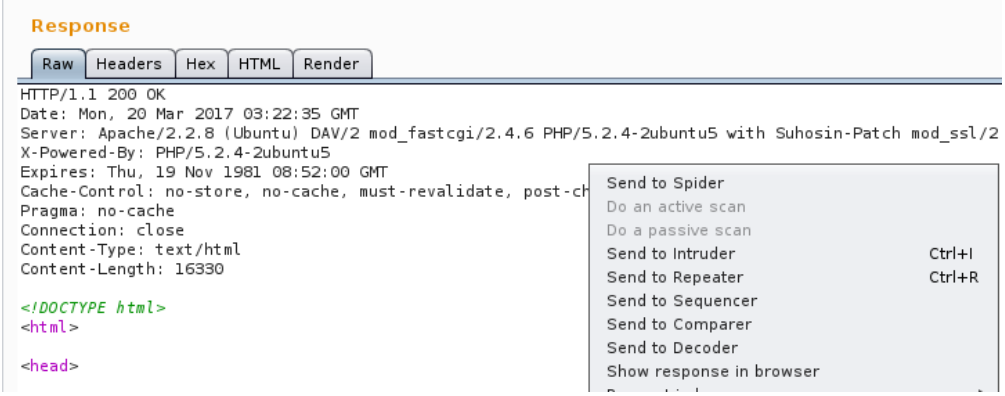
Click in the box that has the single quote and change the value to (‘ or 1 = 1 --) without the parentheses. Be sure to put a space after the double dash. Then click on Go.



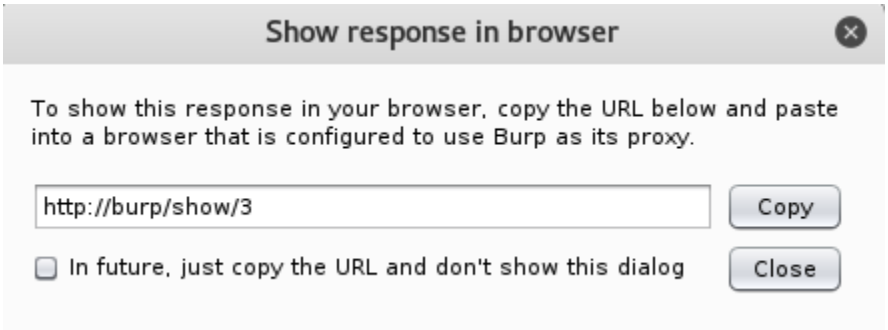
After clicking go, you get a response in the pane to the right. It is in Raw by default. Click on Render.



The results of clicking Render.

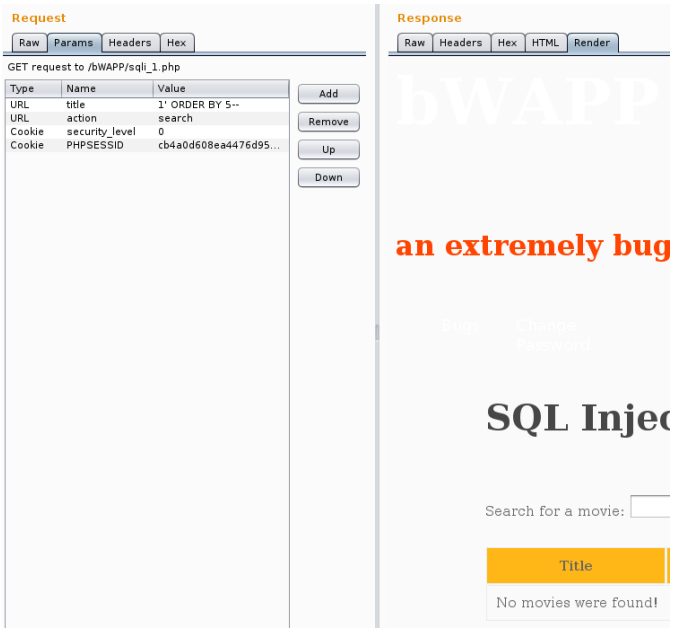


Also, if you wanted to view the response in the browser outside of Burp, right click anywhere in the Raw response pane, and click on Show response in browser.

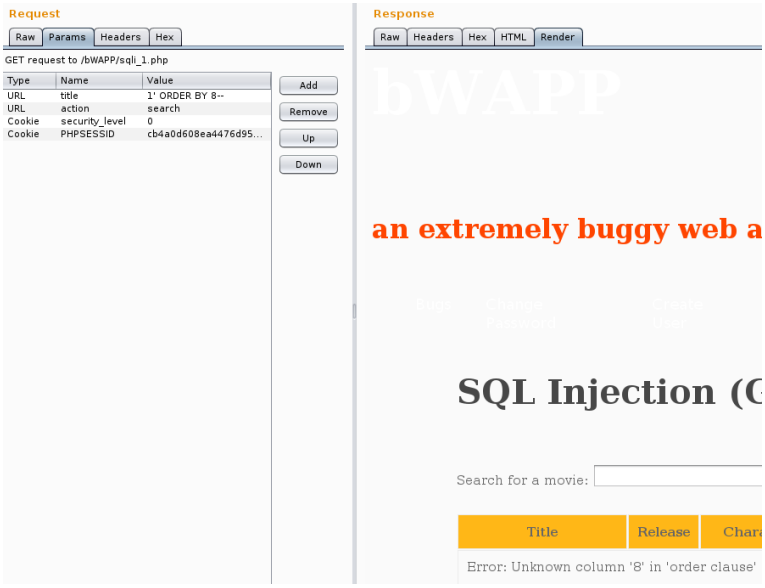


Result of clicking on Show response in browser.

Let’s now see if we can get some valuable information from the database.

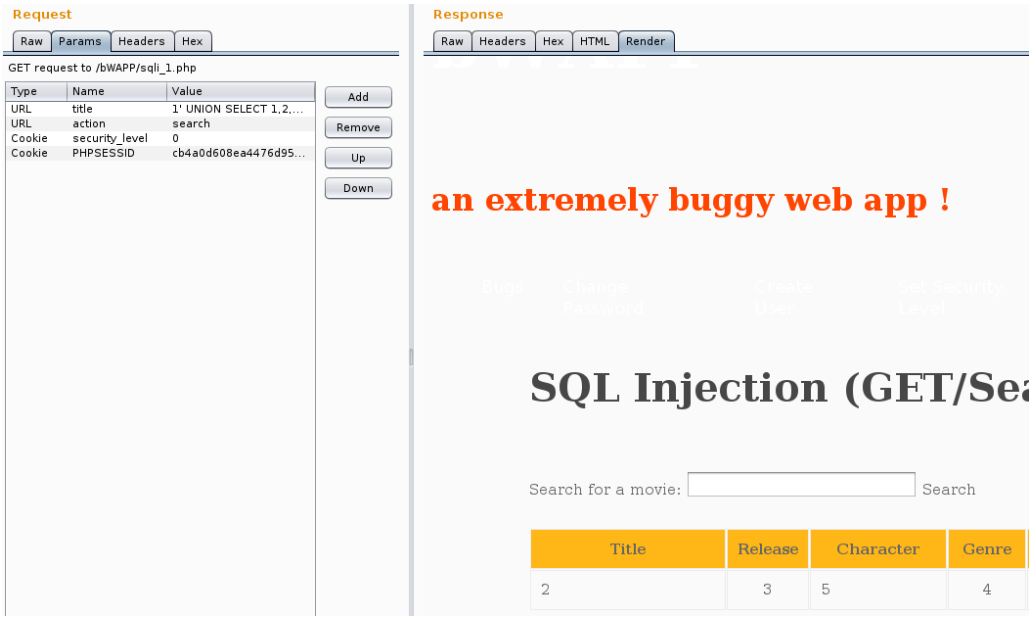


In the Repeater Request Params, change the value to (1’ ORDER BY 5--+) without parentheses. Notice that the response says No movies were found! Keep increasing the number in the injection command until you get an error [13].

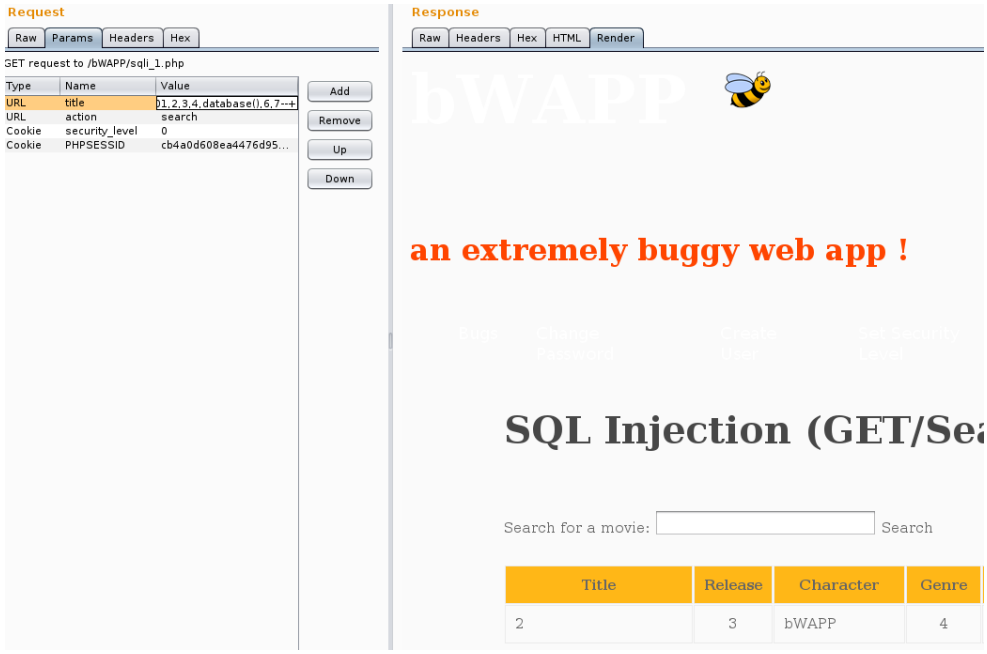


1’ ORDER BY 8--+ throws an error of Error: Unknown column ‘8’ in ‘order clause’. 7 just gives the message that no moves were found.

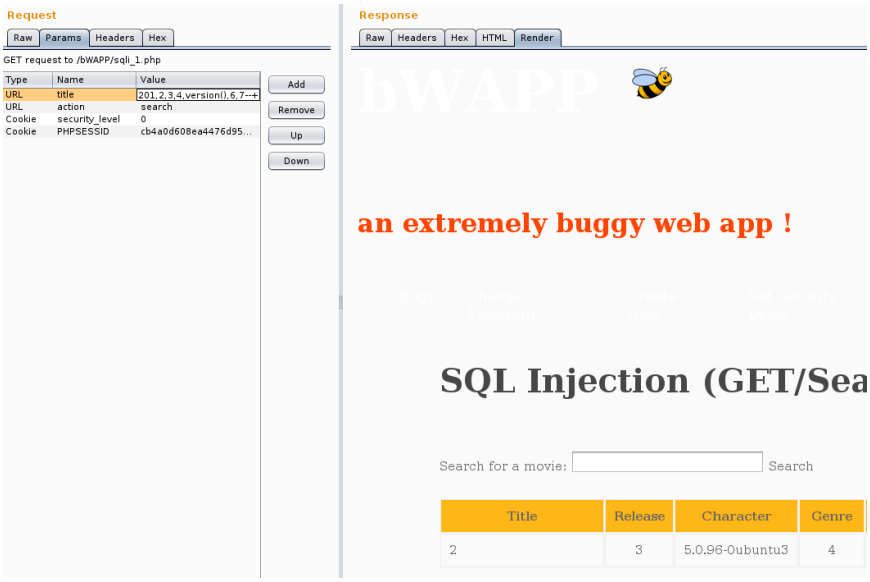
This indicates that there are only 7 columns in the table.



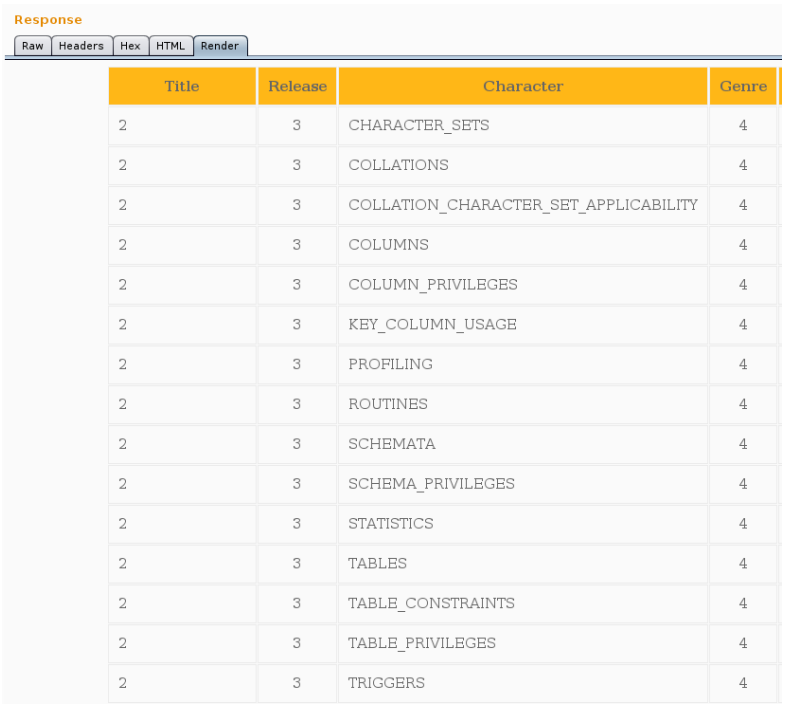
In the Repeater Request title Value, type (1’ UNION SELECT 1,2,3,4,5,6,7--+) without parentheses. This shows the exploitable column numbers on the web page.



Replace the 5 in the injection command with database(). This shows the name of the application.



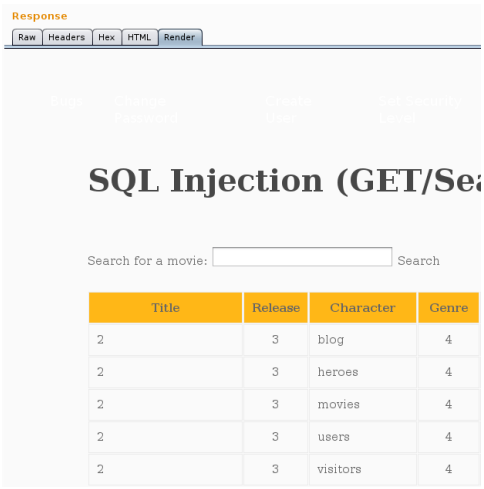
Replace the 5 in the injection command with version() to get the operating system version [14].



Title	Release	Character	Genre
2	3	CHARACTER_SETS	4
2	3	COLLATIONS	4
2	3	COLLATION_CHARACTER_SET_APPLICABILITY	4
2	3	COLUMNS	4
2	3	COLUMN_PRIVILEGES	4
2	3	KEY_COLUMN_USAGE	4
2	3	PROFILING	4
2	3	ROUTINES	4
2	3	SCHEMATA	4
2	3	SCHEMA_PRIVILEGES	4
2	3	STATISTICS	4
2	3	TABLES	4
2	3	TABLE_CONSTRAINTS	4
2	3	TABLE_PRIVILEGES	4
2	3	TRIGGERS	4

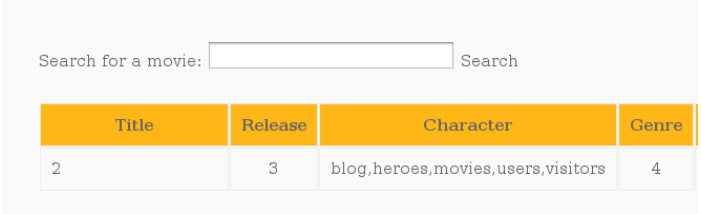
Results from entering the following injection command in the title Value field and clicking Go:

1’ UNION SELECT 1,2,3,4,table_name,6,7 FROM information_schema.tables--+



Results from entering the following injection command in the title Value field and clicking Go:

1’ UNION SELECT 1,2,3,4,table_name,6,7 FROM information_schema.tables WHERE table_schema = database()--+

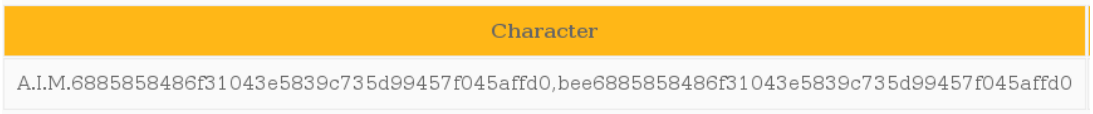


Results from entering the following injection command in the title Value field and clicking Go:

1’ UNION SELECT 1,2,3,4,GROUP_CONCAT(table_name),6,7 FROM information_schema.tables WHERE table_schema = DATABASE()--+

Results from entering the following injection command in the title Value field and clicking Go:

1’ UNION SELECT 1,2,3,4,GROUP_CONCAT(column_name),6,7 FROM information_schema.columns WHERE table_name=”users”--+



Retrieved usernames and hashed passwords by entering the following injection command in the title Value field and clicking Go:

1’ UNION SELECT 1,2,3,4,GROUP_CONCAT(login,password),6,7 FROM users--+

Questions:

What injection command would you use to get the user email?

What injection command would you use to get all columns in the movies table?

What injection command would you use to get just the title and release year of all movies in the movies table?

Go to the SQL injection (Login Form/Hero) in bWAPP and write a command in Burp Repeater to return the hero name and password.

In your own words, explain how to mitigate SQL injection attacks.

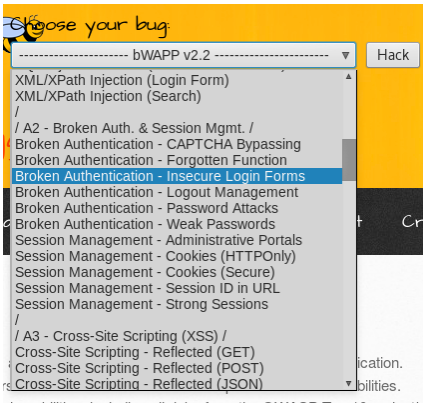
Lab 5: Broken Authentication and Session Management

1. Introduction

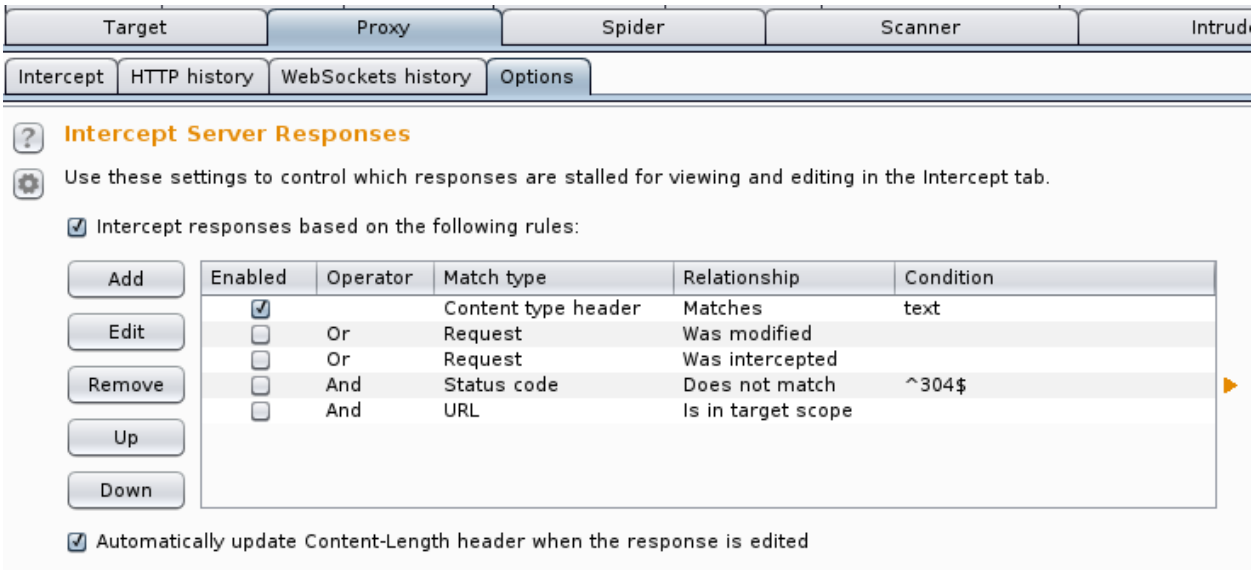
When developing a website, you want to make sure that only authorized users have access to information tied directly to their accounts. Insecure authentication can arise from users choosing weak passwords, developers utilizing insecure authentication strategies, or even implementing poor session management techniques [15].

In this lab, we will exploit insecure login forms, overcome weak passwords, and automate URL manipulation to unlock an administrative portal. Insecure login forms can be exploited by searching the raw data of the response in Burp Proxy. Weak passwords can be cracked by making use of Burp Intruder [16]. We will also use the Intruder Sniper attack with the Numbers payload to iterate through different URLs to defeat session management and unlock the administrative portal.

2. Attacking insecure login forms

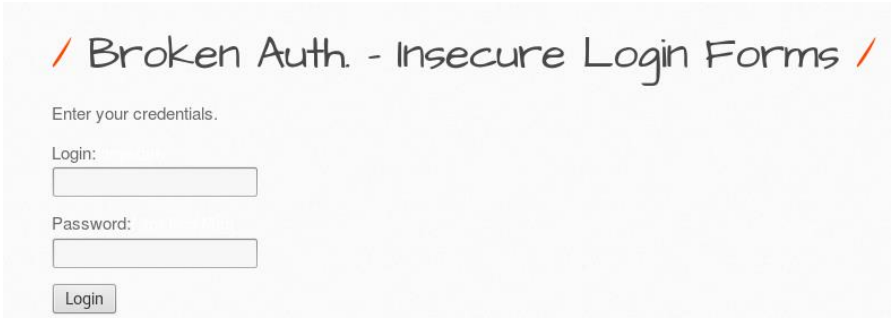


Login to bWAPP and select Broken Authentication - Insecure Login Forms from the bug choice menu. Then click the Hack button.

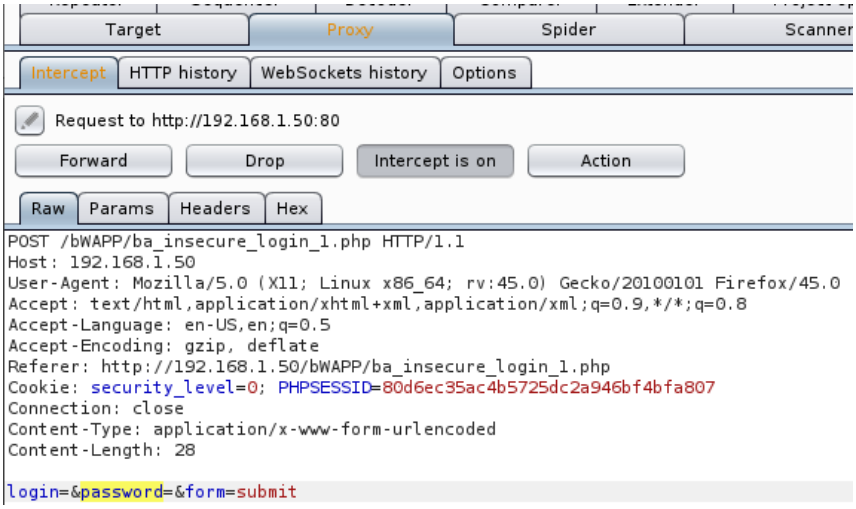


Go to Burp Proxy > Options. Scroll down to Intercept Server Responses and check Intercept responses based on the following rules. Then ensure that Content type header Matches text is enabled.

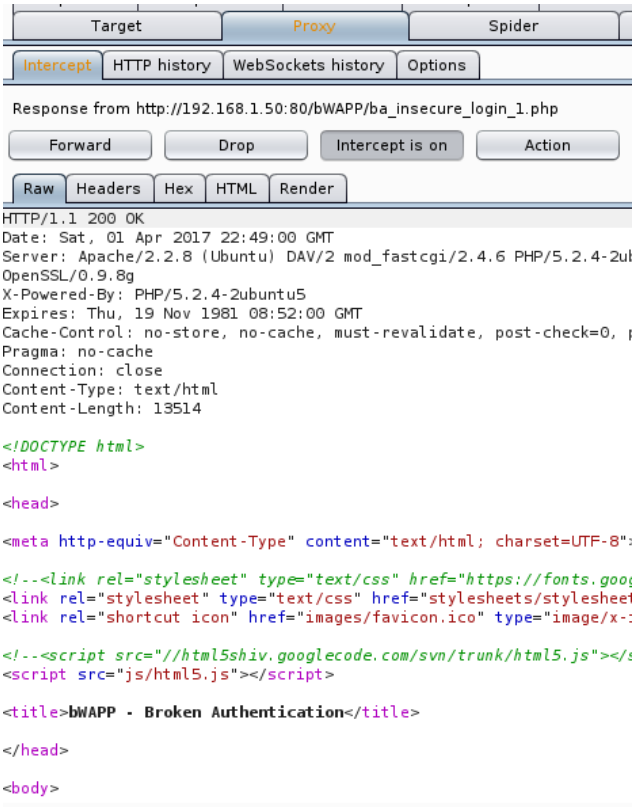
Ensure that Intercept is on in Burp Proxy > Intercept.



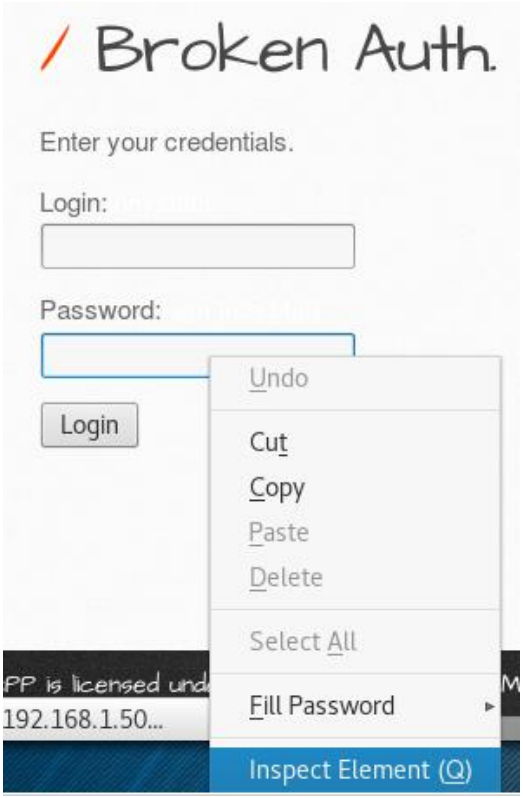
Then in the browser, click the login button without entering any credentials.



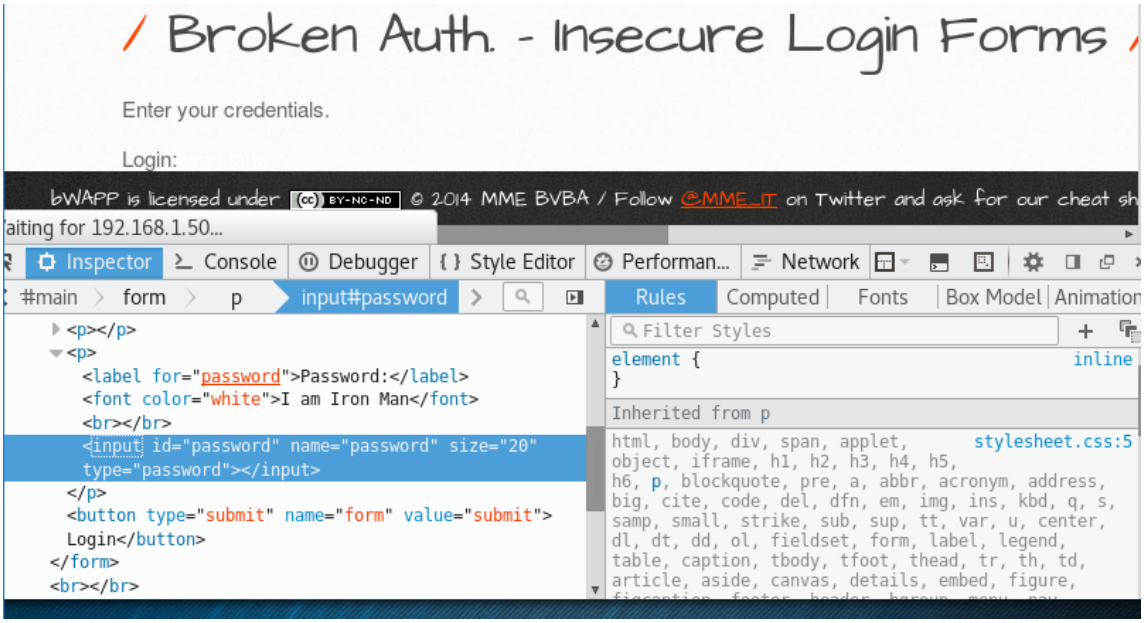
Verify that the request is intercepted in Burp Proxy > Intercept.



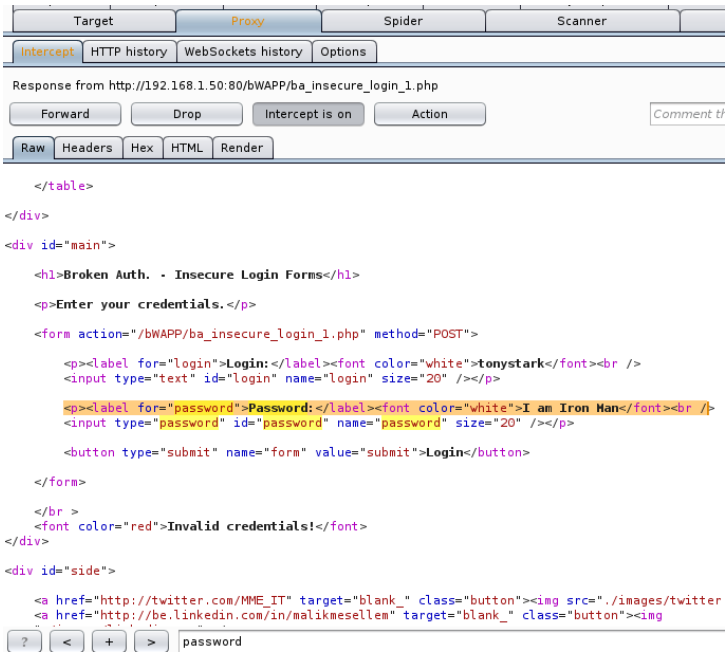
Click the Forward button that is two buttons to the left of the “Intercept is on” button, and verify that the response is displayed in Burp Proxy > Intercept.



Right click on the password textbox and click Inspect Element.



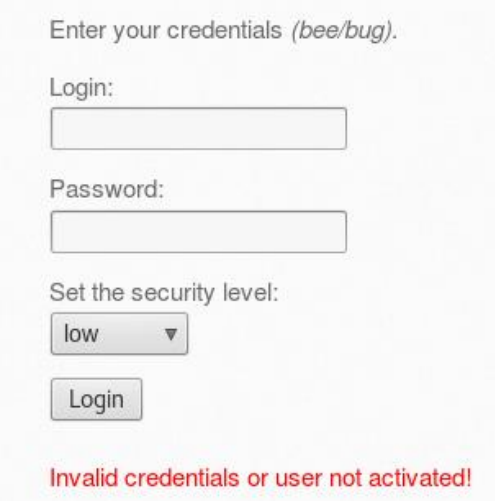
The name of that textbox is password.



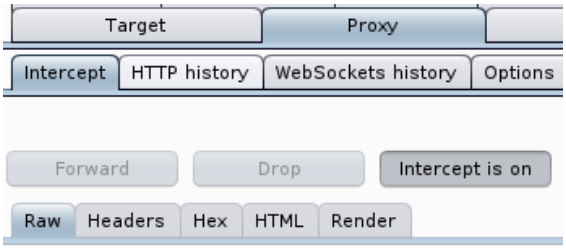
At the bottom of the Intercept window, type password in the search box and hit enter. Click on the right arrow next to the search box to iterate through all of the instances of password. You will see that the login for tonystark and password of “I am Iron Man” are embedded in the response.

3. Password Attacks

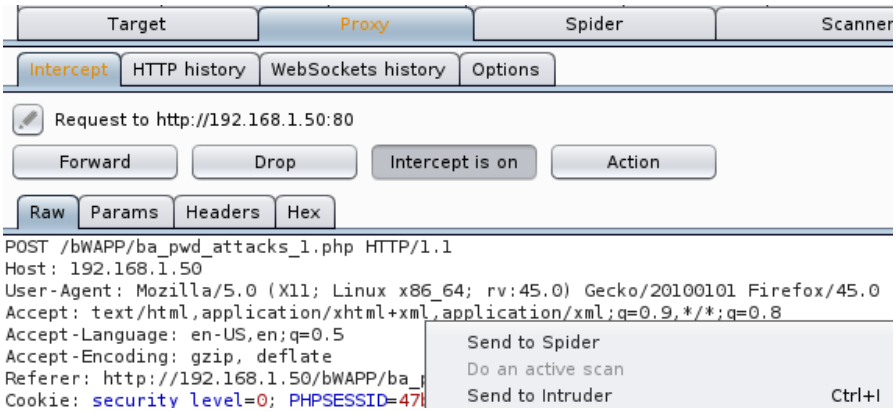
Select Broken Auth. - Password Attacks from the bug selection menu. Enter user and password for the credentials and click Login.



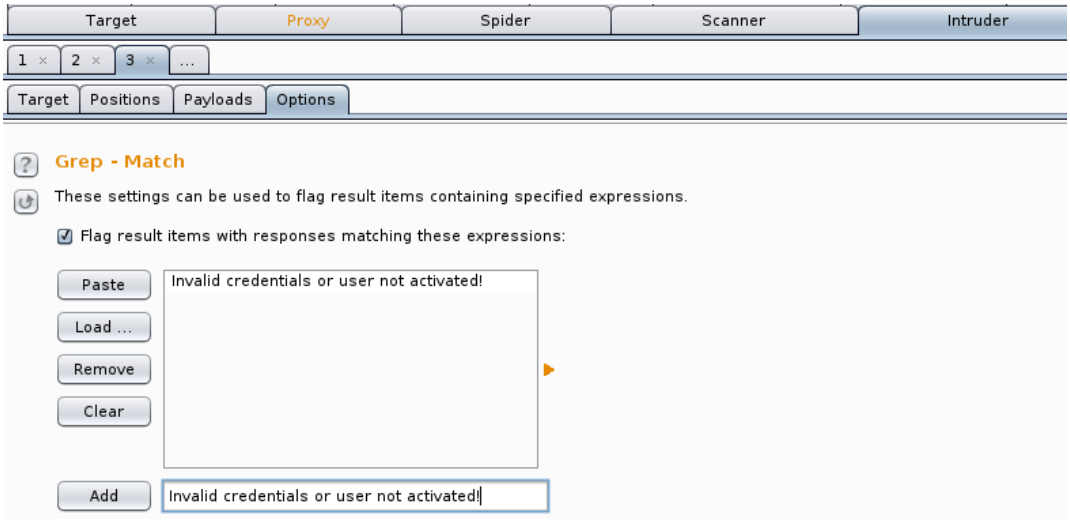
Verify that you get the error message displayed.



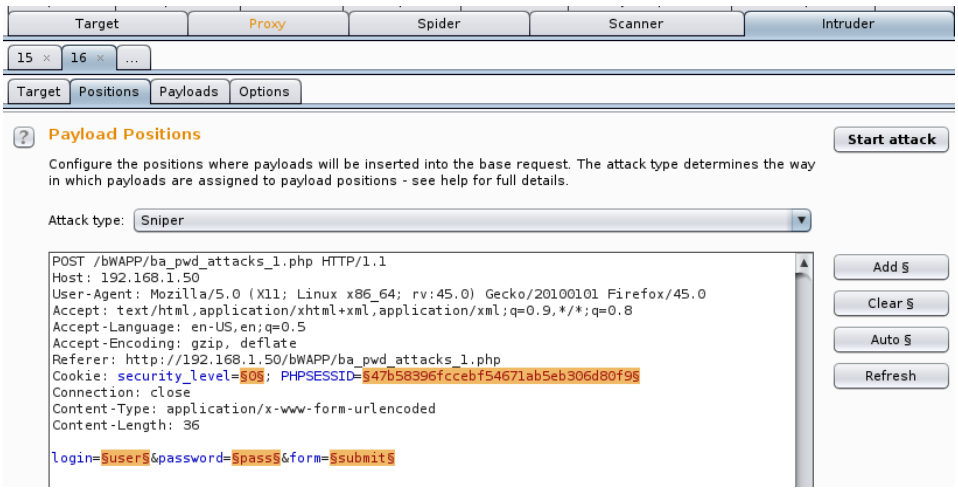
Turn intercept on in Burp Proxy then go back to the browser, type user/pass for the credentials and then click Login.



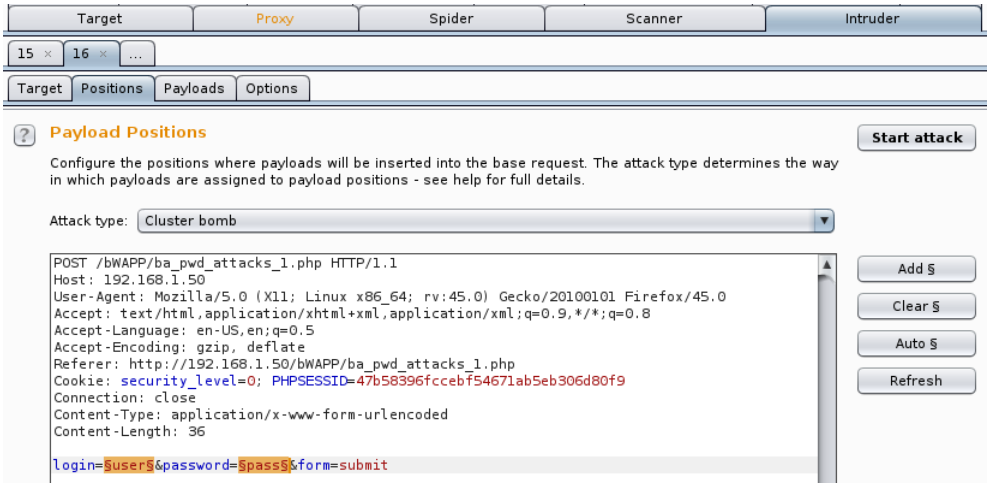
In Burp Proxy, right click on the Raw Request window and click on Send to Intruder.



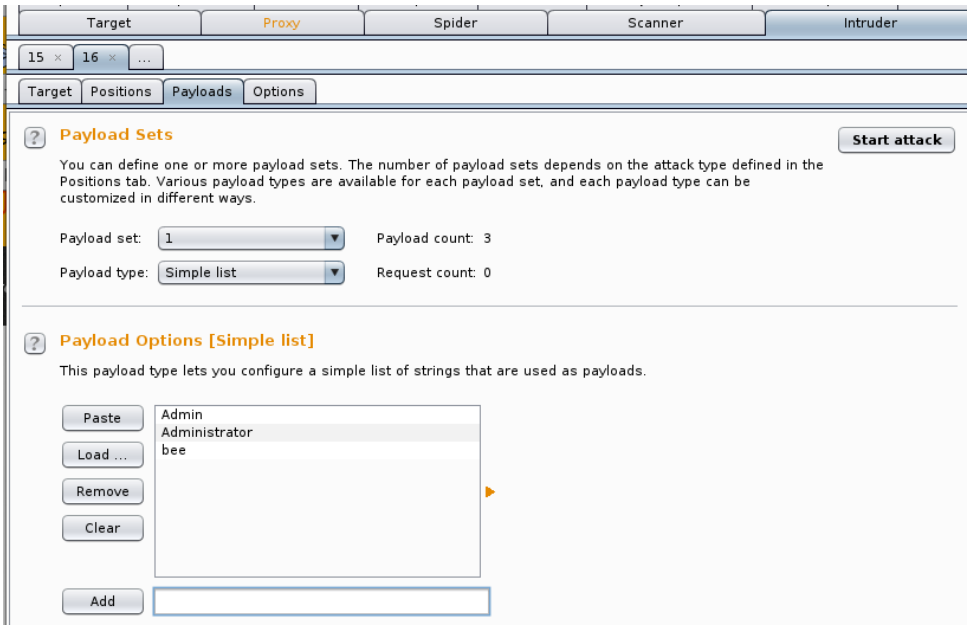
Navigate to Burp Intruder > Options. Scroll down to Grep - Match. Click the check box for “Flag result items with responses matching these expressions.” Remove the contents of the expression box by clicking the Clear button, copy the login error message from the browser, paste the error into the add text box, and then click the Add button.



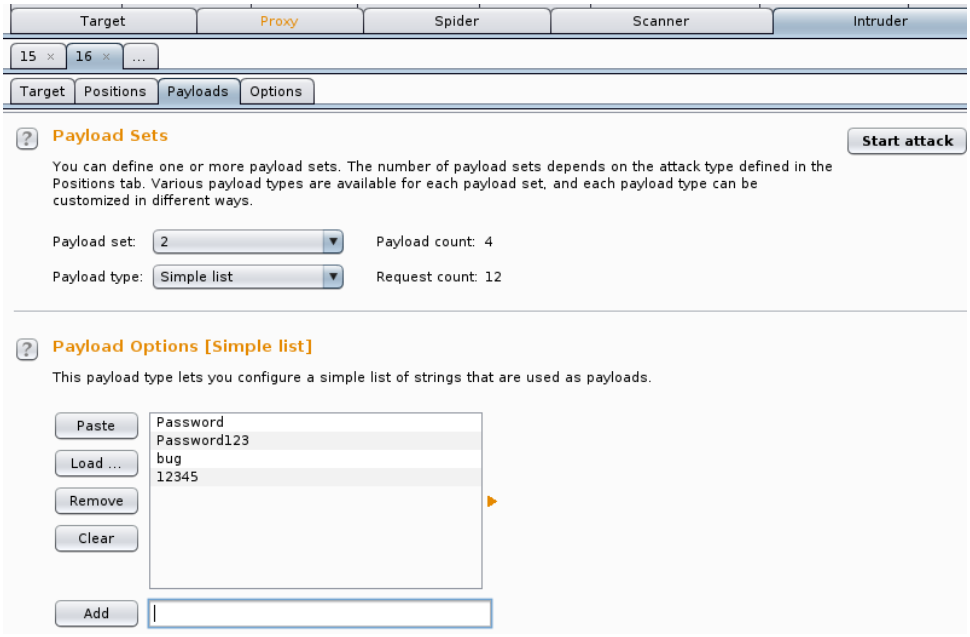
Click on the Positions tab in Intruder then click the clear button to the right.



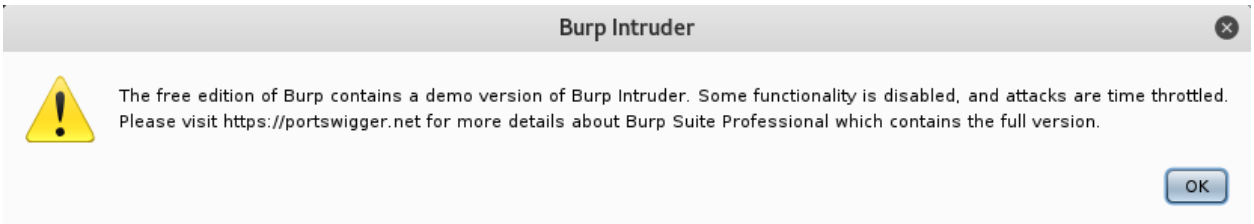
Change the attack type to Cluster bomb. Highlight user and click Add. Highlight pass and click Add.



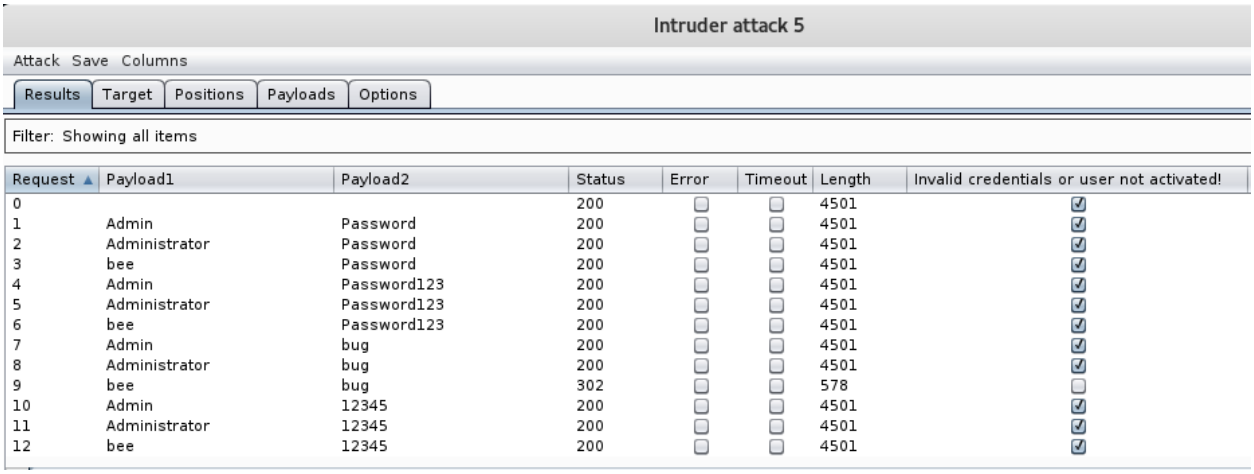
Go to Intruder > Payloads. Under Payload Options, add the users listed above by typing them in the add textbox and clicking the Add button.



Select 2 from the Payload set selection box. Add the passwords listed above by typing them in the add textbox and clicking the Add button. Alternatively, you can click the Load button and load a password list.



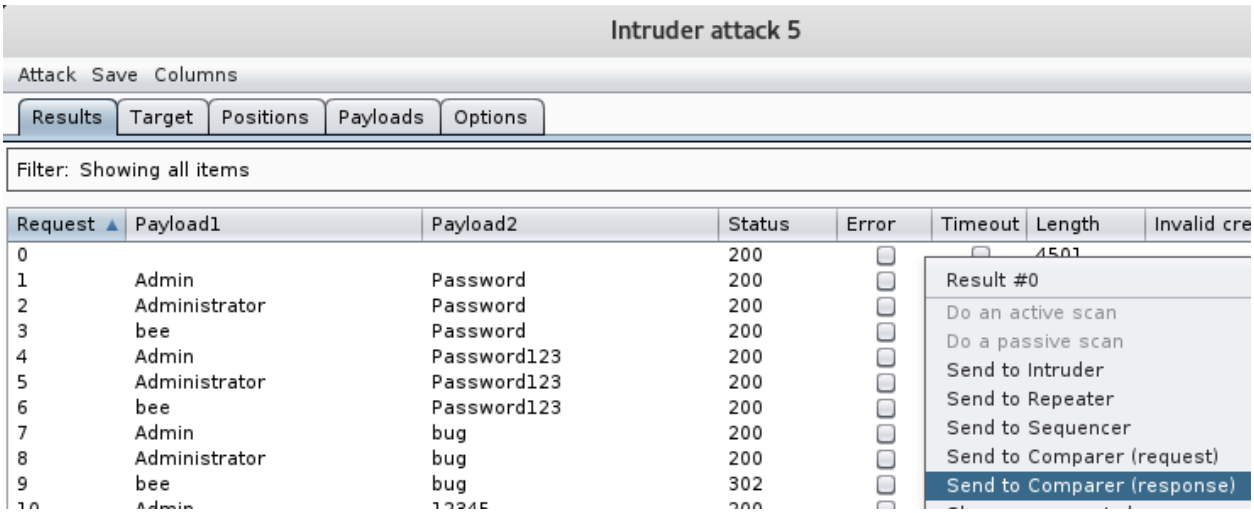
Click Start attack and then click OK on the warning that pops up which notifies the user of the attack being time throttled in the free version of Burp Suite.



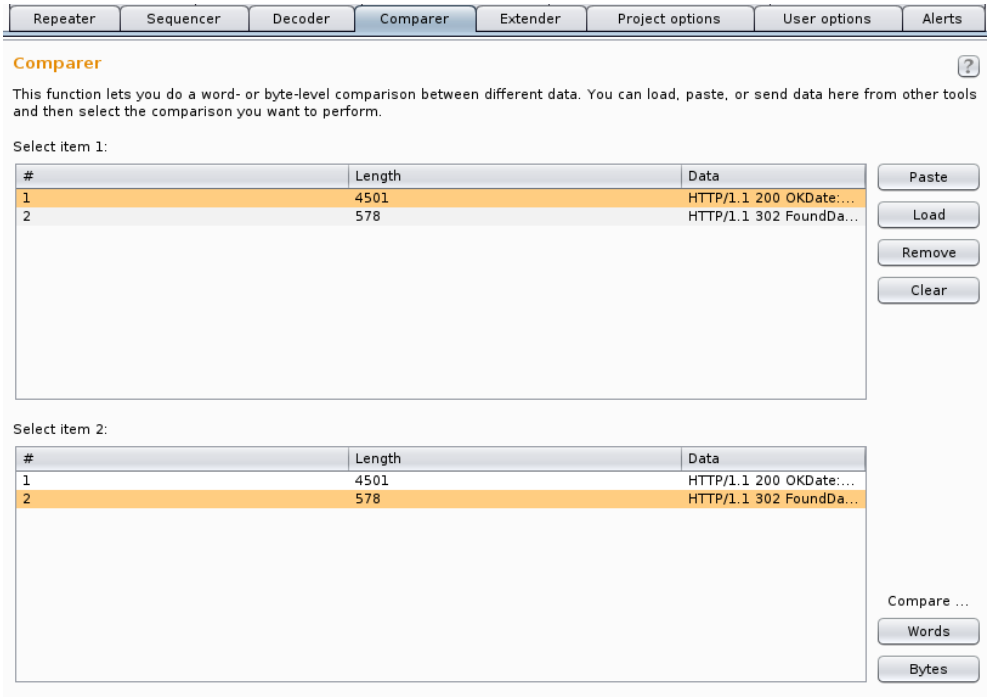
Request	Payload1	Payload2	Status	Error	Timeout	Length	Invalid credentials or user not activated!
0			200	☐	☐	4501	☑
1	Admin	Password	200	☐	☐	4501	☑
2	Administrator	Password	200	☐	☐	4501	☑
3	bee	Password	200	☐	☐	4501	☑
4	Admin	Password123	200	☐	☐	4501	☑
5	Administrator	Password123	200	☐	☐	4501	☑
6	bee	Password123	200	☐	☐	4501	☑
7	Admin	bug	200	☐	☐	4501	☑
8	Administrator	bug	200	☐	☐	4501	☑
9	bee	bug	302	☐	☐	578	☐
10	Admin	12345	200	☐	☐	4501	☑
11	Administrator	12345	200	☐	☐	4501	☑
12	bee	12345	200	☐	☐	4501	☑

Burp will then show all user/password combinations, the status of each attack, any errors, timeouts, length of the request, and the grep expression for flagging purposes. Notice that the only user/password combination that is not flagged is bee/bug, which are valid credentials.

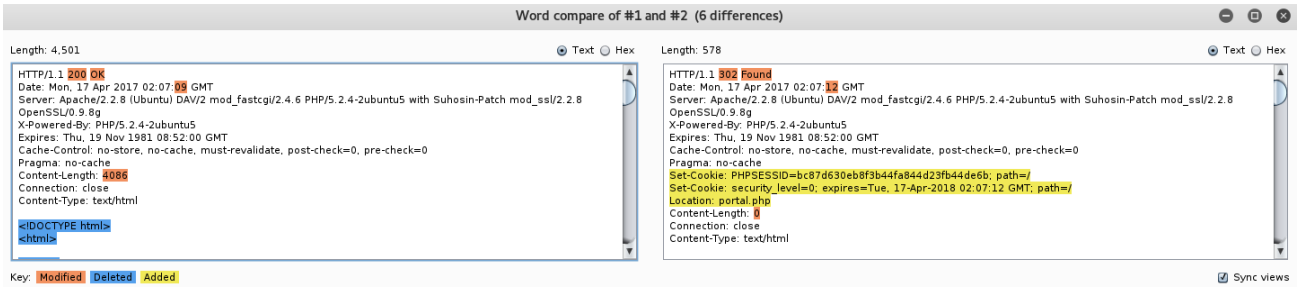
Alternatively you could send requests to the comparer to see the difference between one two requests of different lengths [17].



Right click the base request (Request 0) and then click on Send to Comparer (response). Then do the same for Request 9.

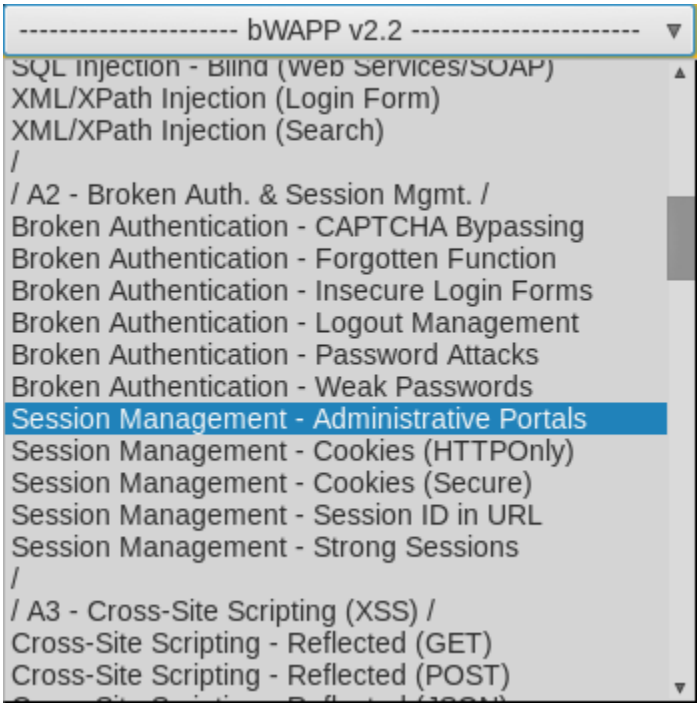


Navigate to Comparer and click on Words in the lower right.

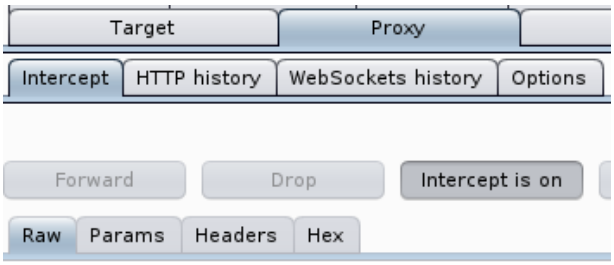


If both responses have a lot of text, it is wise to click Sync views in the lower right so that you can when you scroll down with one scroll bar, they both scroll down in sync. In this instance, it does not matter if you click Sync views as the response to the right is much shorter. The text highlighted in yellow is text that the other response does not have. The response on the right has two cookies. The response on the left does not.

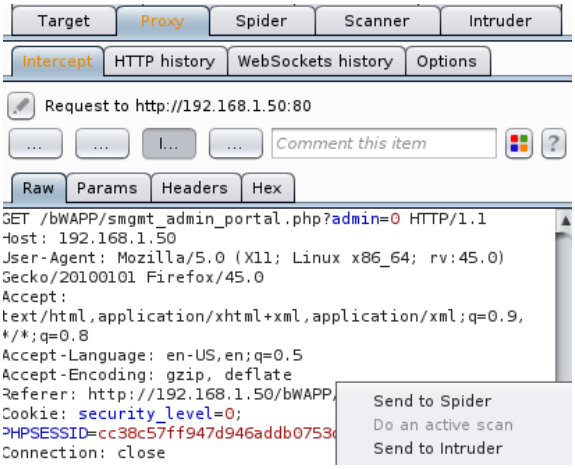
4. Attacking locked administrative portal



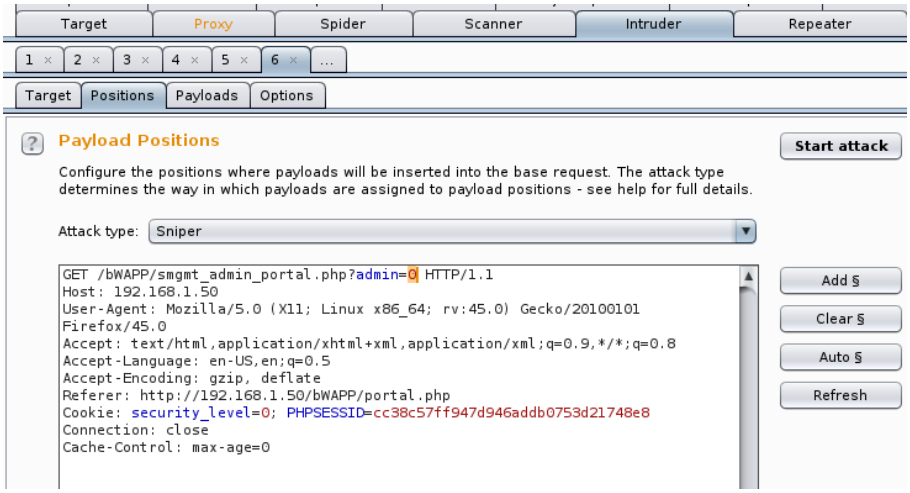
Log into bWAPP with bee/bug, select Session Management - Administrative Portals from the bug selection menu, and then click Hack [16].



Turn Intercept to on in Burp Proxy > Intercept.

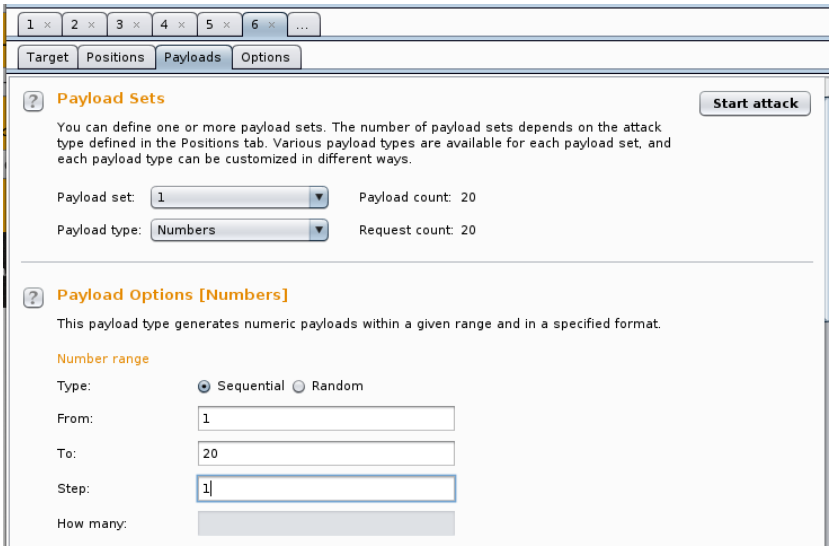


Refresh the browser so that the request is intercepted by Burp Proxy. Then right click in the Raw Request window, and then click Send to Intruder.

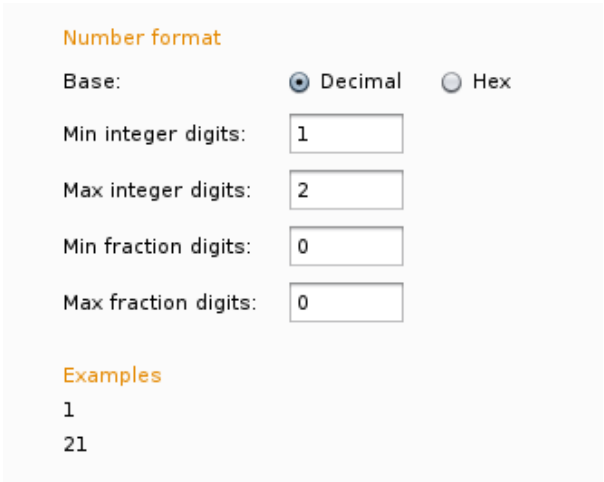


Navigate to Burp Intruder > Positions, click the Clear button, then highlight the zero in “admin=0”.

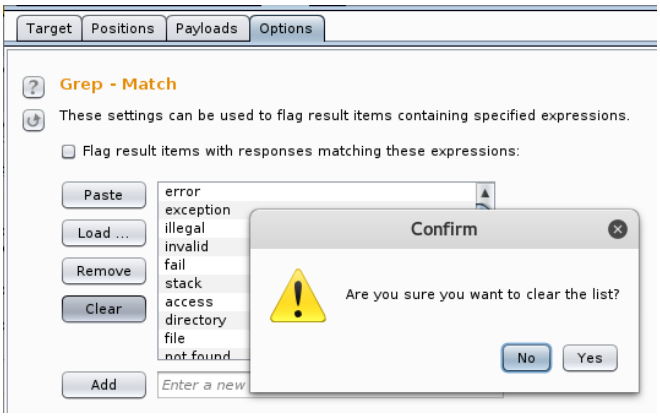
Click Add.



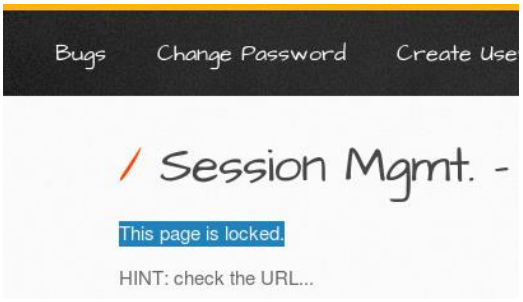
Click on the Payloads tab, change the Payloads type to Numbers. Under Payload Options, type 1 in the From field, type 20 in the To field, and type 1 in the Step field.



Still under Payload Options, set the Min integers digits to 1, set the Max integer digits to 2, and set both fraction digits to zero.

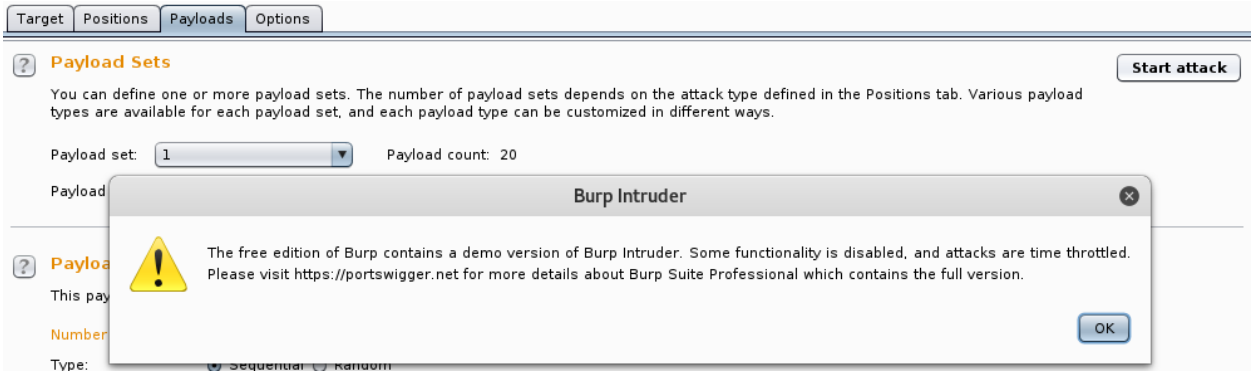


Under Burp Intruder > Options > Grep - Match, click the Clear button, and then click Yes.

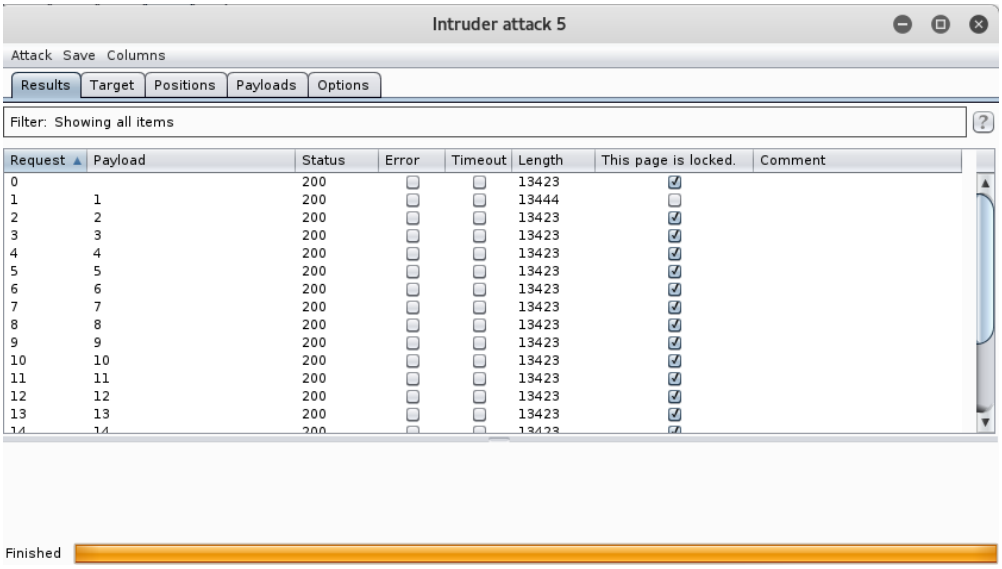


Highlight and copy the error message from the browser.

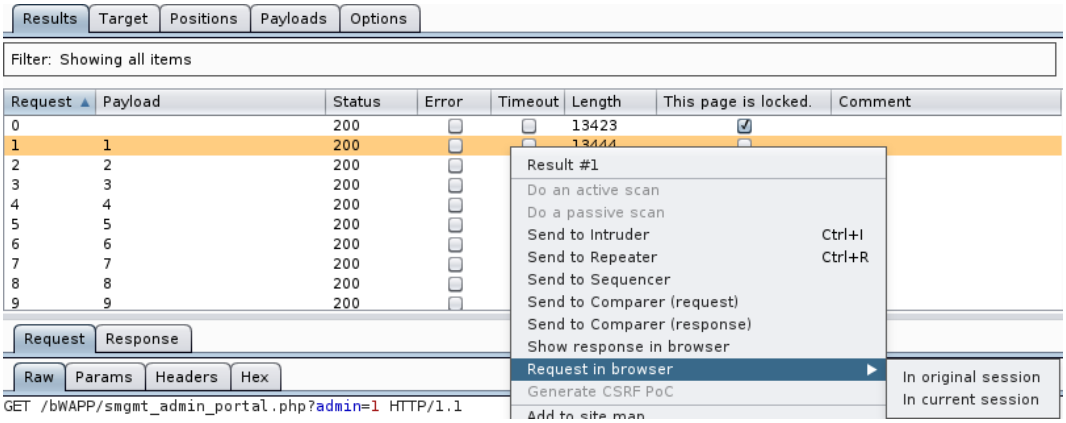
Copy it into the text field to the right of the Add button, click the Add button, and verify that check box gets checked and “This page is locked.” gets added to the match list.



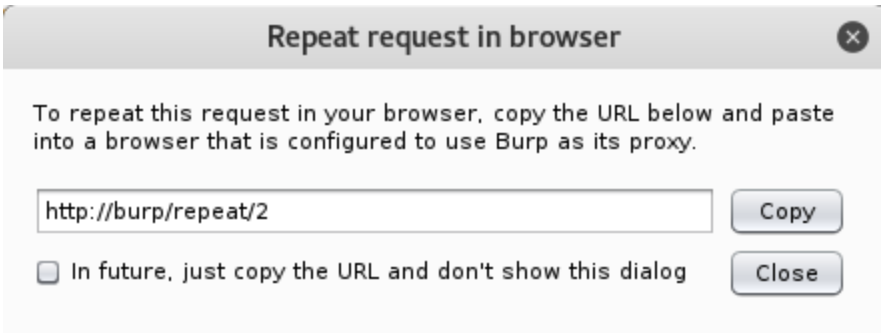
Click back on the Payloads tab, click Start attack, and then click OK.



Noticed that request 1 is not flagged for “This page is locked.”



Right click Request 1, click Request in browser, and then click In original session.



Click copy and paste it into a new tab in the browser. Make sure that Intercept is off in Burp Proxy.

Congrats on defeating session management.

Questions:

What are all of the attack types in Intruder? Describe what each attack type does.

What is a popular password list that comes bundled with Kali?

What is one thing that a web developer can do to mitigate wordlist attacks?

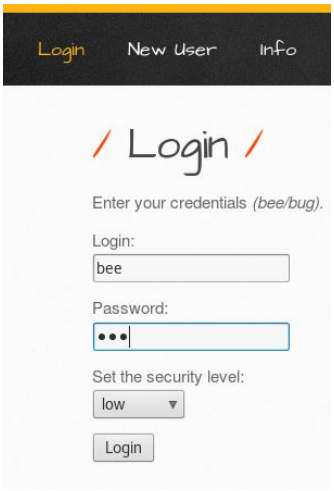
Name a few steps that a developer can take to strengthen session management.

Lab 6: Cross-Site Scripting (XSS)

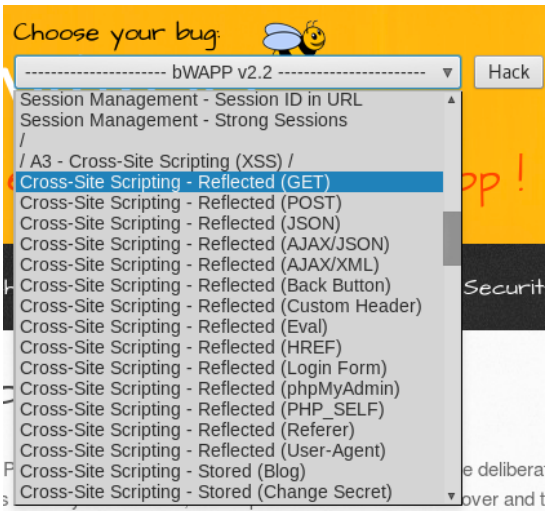
1. Introduction

Cross-site scripting attacks are attacks where an attacker injects malicious scripts into trusted web sites. There are two forms of XSS, reflected and persistent. Reflected XSS occurs when the attacker injects malicious code into a single HTTP response. The code only affects users who open the maliciously crafted link or third-party web page, which does not get stored within the application itself [18]. Persistent or stored XSS allows users to store malicious code into the web application to attack other users who visit the location of the malicious code. This is considered the most dangerous form of XSS as it allows for browser-based attacks such as capturing sensitive information of users who visit the part of the web application where the malicious code resides, hijacking another user's browser, delivering browser-based exploits, and port scanning of hosts internal to users of the web application [19].

2. XSS Reflected Attack

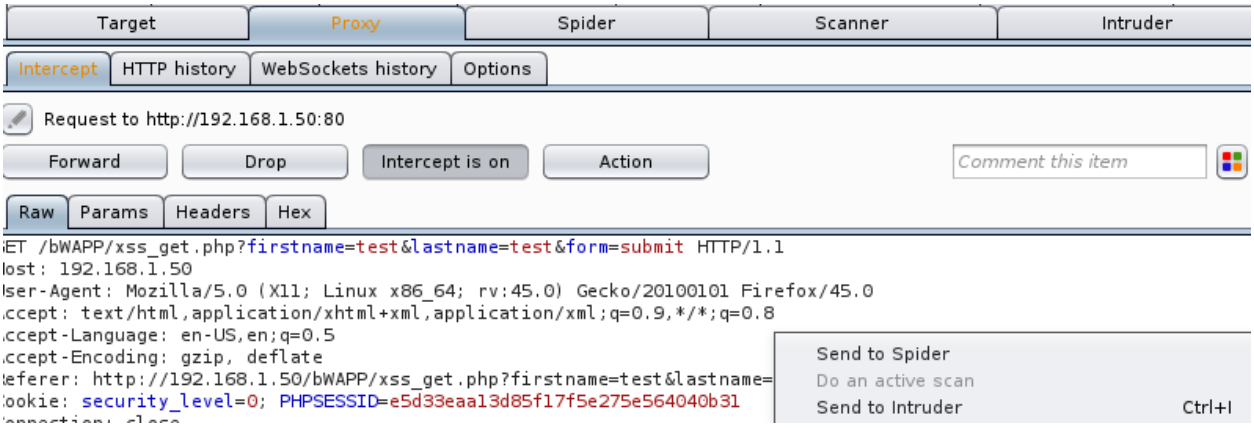


Log into bWAPP with bee/bug credentials.

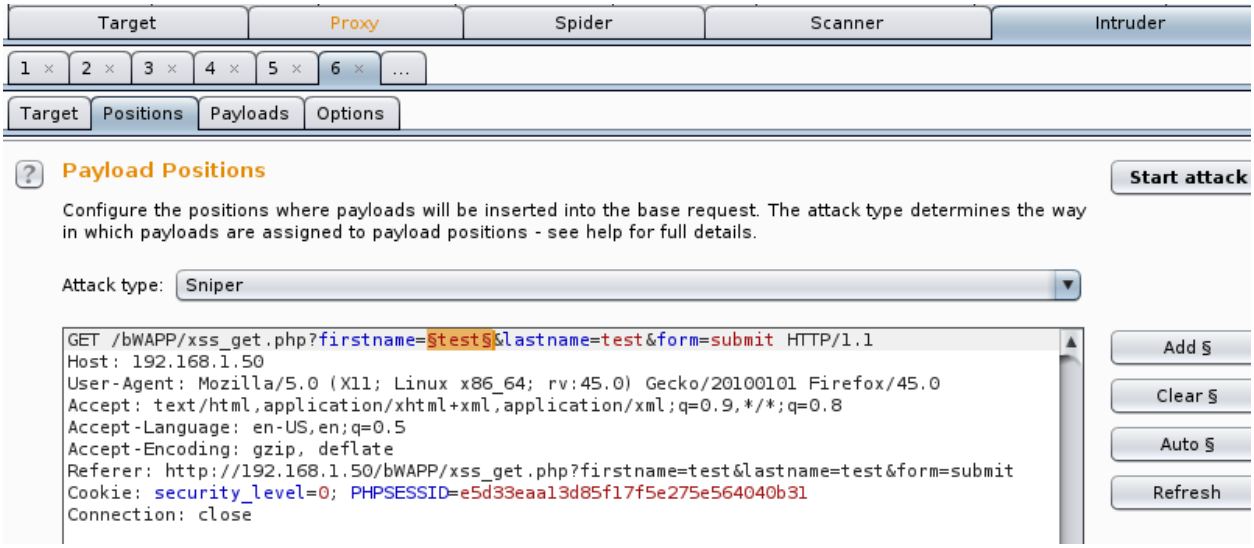


Select Cross-Site Scripting - Reflected (GET) from the bug selection menu and then click Hack.

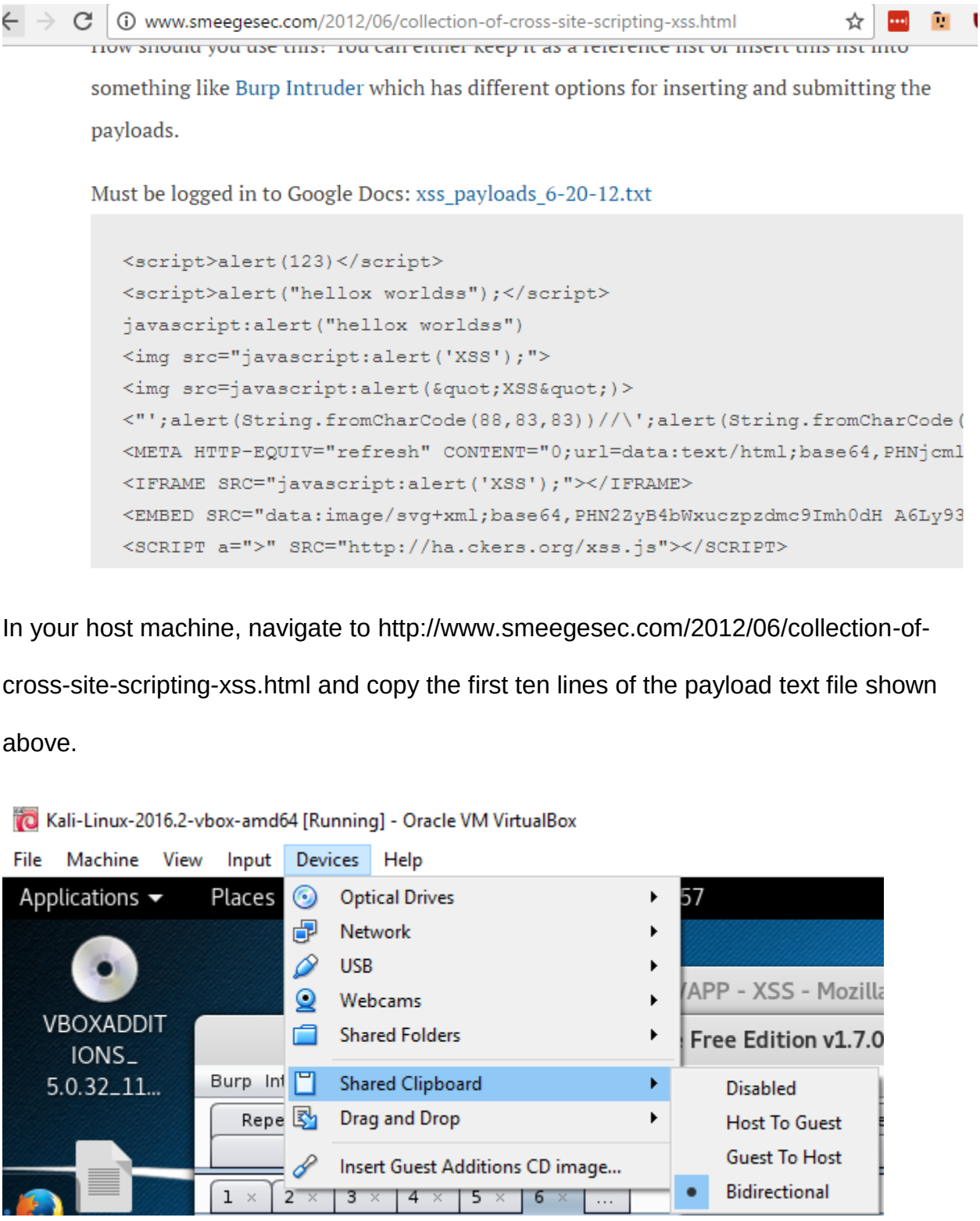
Type test/test for credentials, turn Intercept on in Burp Proxy and then click the Go button in the browser.



Right click in the Raw Request window and click Send to Intruder.

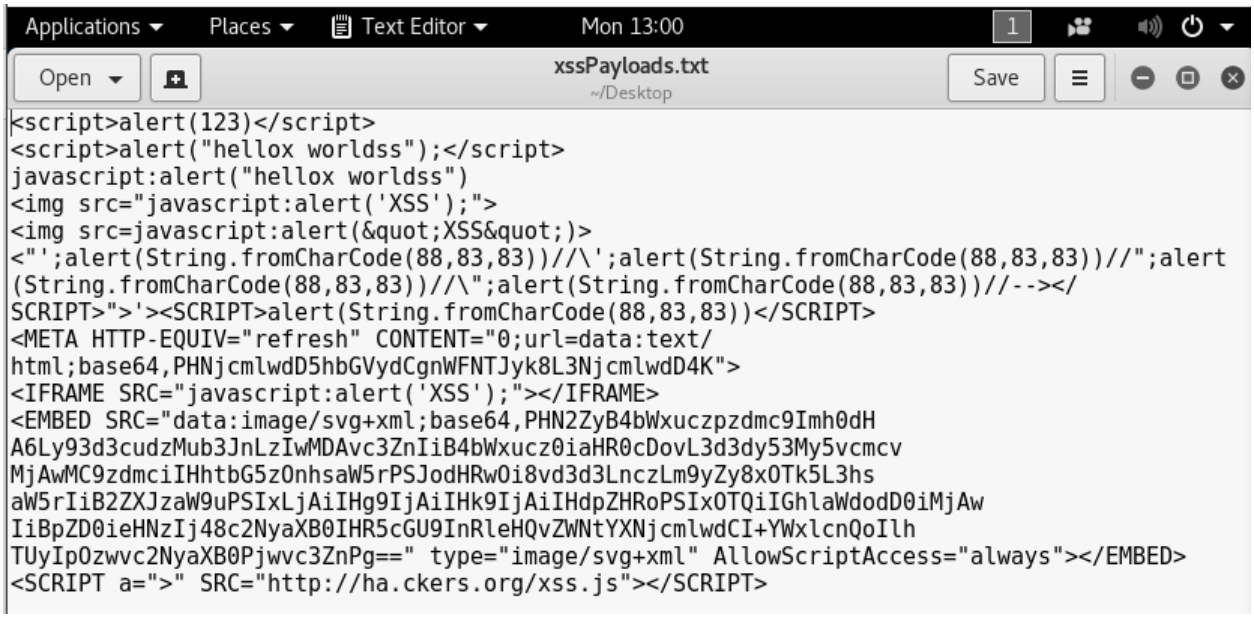


Navigate to Burp Intruder > Positions. Click the Clear button, highlight test after firstname, and then click the Add button.

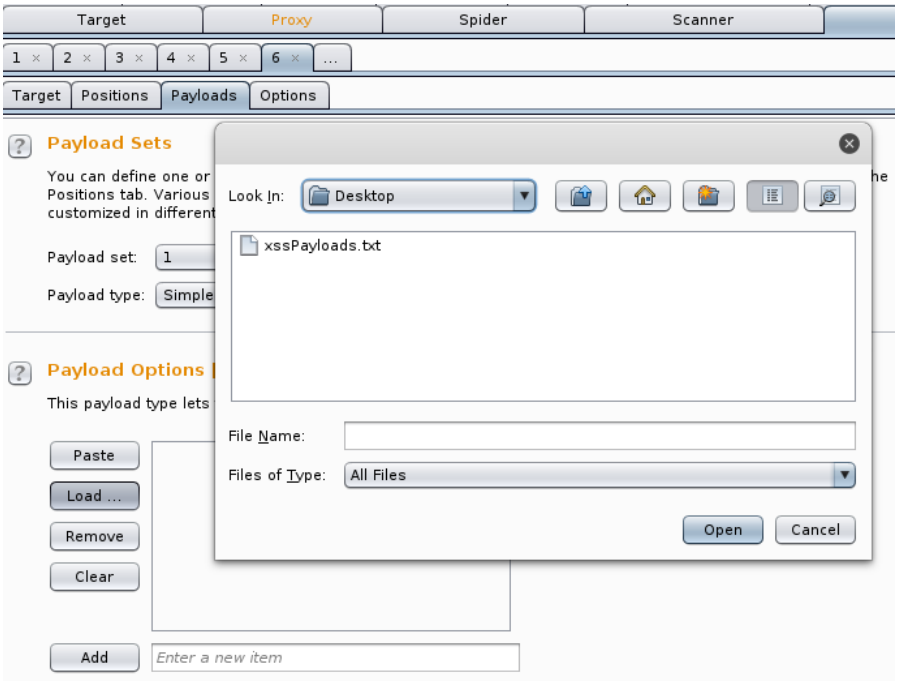


In your host machine, navigate to http://www.smeegesec.com/2012/06/collection-of-cross-site-scripting-xss.html and copy the first ten lines of the payload text file shown above.

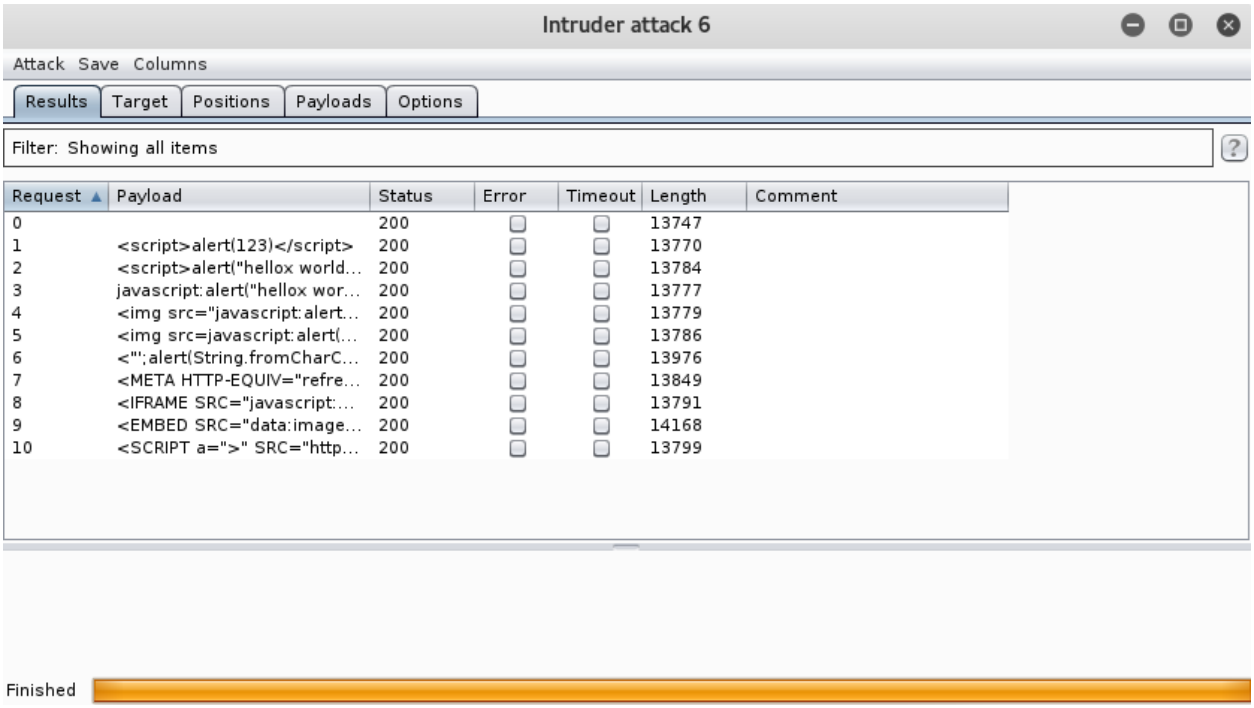
Ensure that clipboard functionality is turned on in your VM. Use either Bidirectional or Host to Guest.



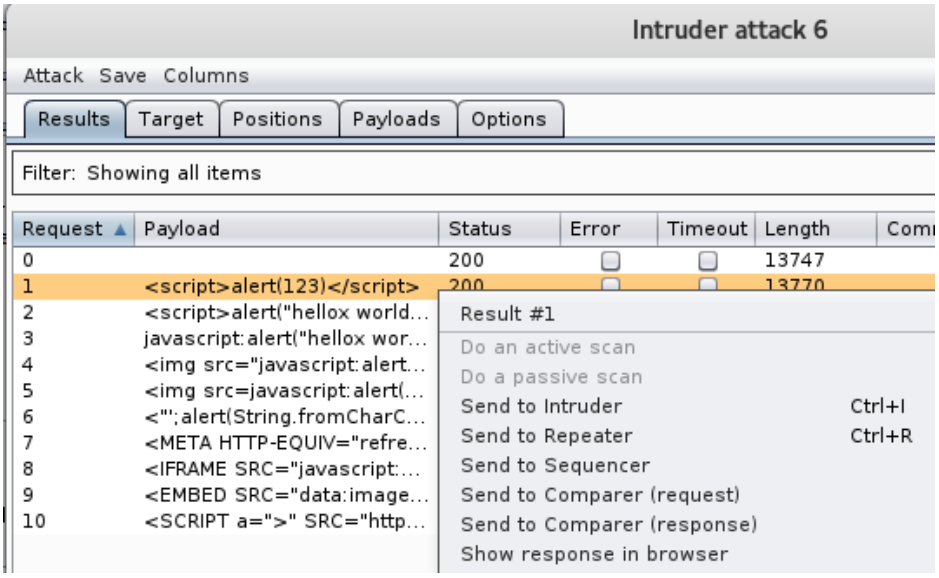
Open Leafpad in your Kali VM, paste the payloads into Leafpad and save the file as xssPayloads.txt.



Navigate to Burp Intruder > Payloads, click Load, select the file that you just created, and click Open.



Click Start Attack and then click OK on the prompt that follows. You should see the window shown above. Take note of the different lengths for each payload compared to the base (or 0) Request.



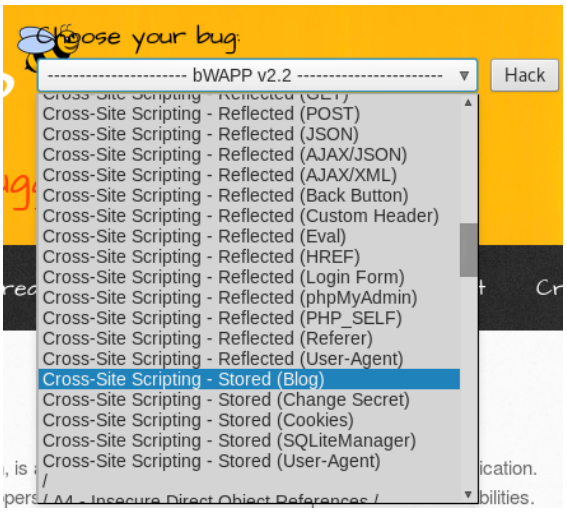
Right click on a row and then click Show response in browser. Click copy, then open a new tab in your VM's browser and Paste & Go.

Here is the result of the payload <script>alert(123)</script>.

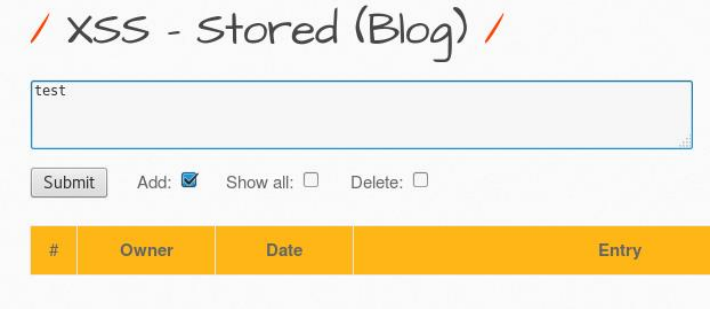
This confirms that the application is vulnerable to XSS.

An attacker can have a victim click on a link that contains maliciously crafted code to steal the session cookie and send it to the attacker, who has a listener running.

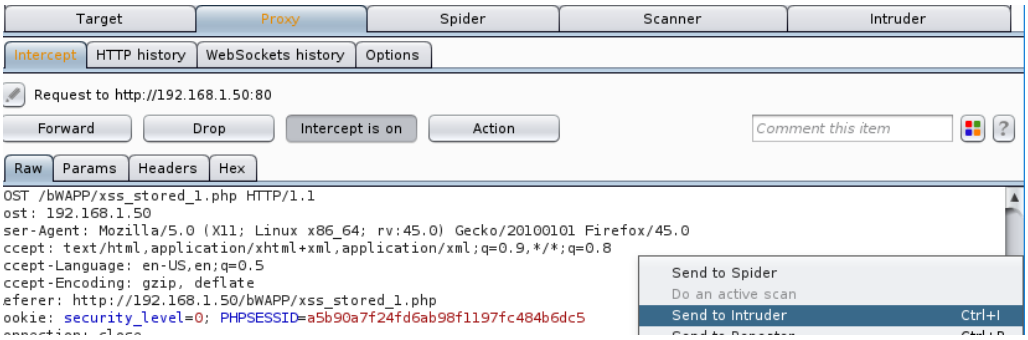
3. Stored XSS Attack



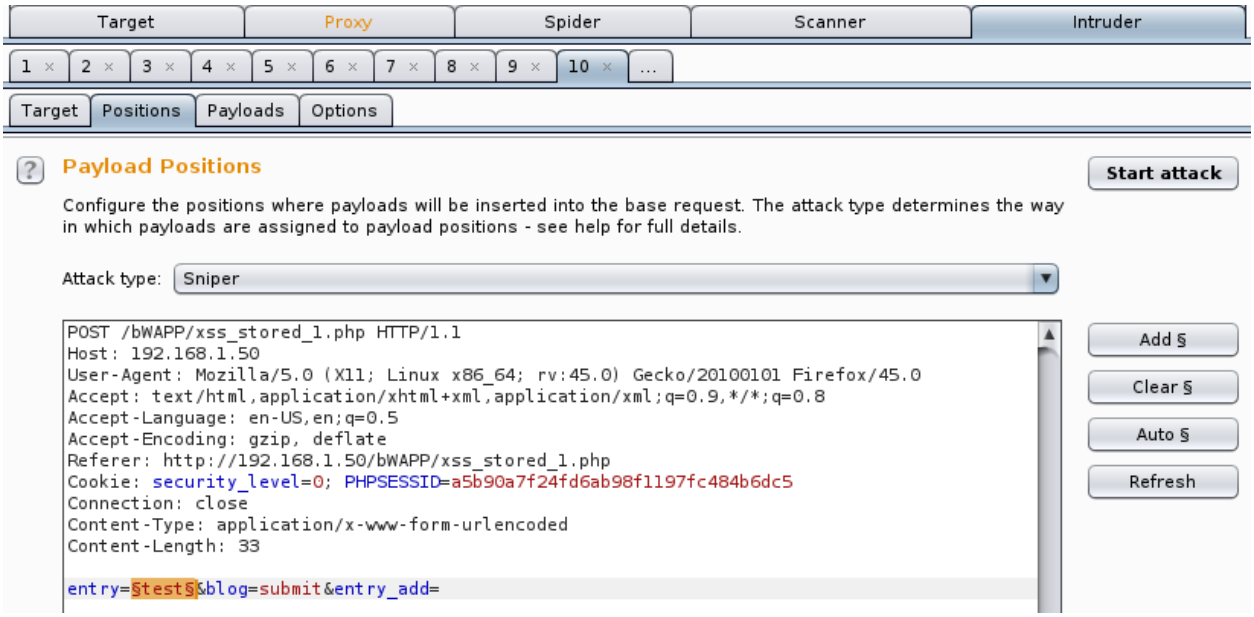
Log in with bee/bug and select Cross-Site Scripting - Stored (Blog) from the bug selection menu and click Hack.



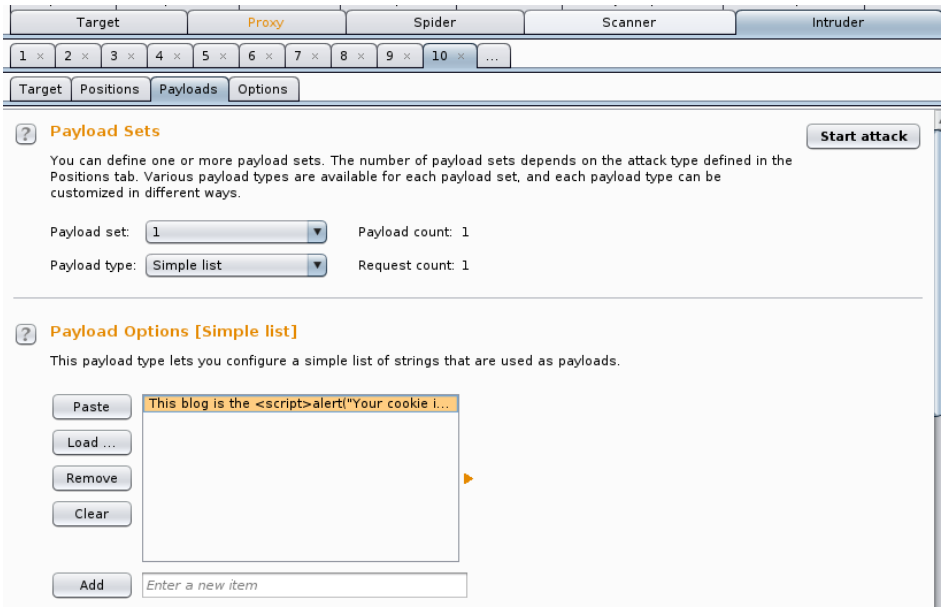
Turn intercept on in Burp Proxy. Type “test” in the form and click Submit.



Right click in the Raw Request window and click on Send to Intruder.



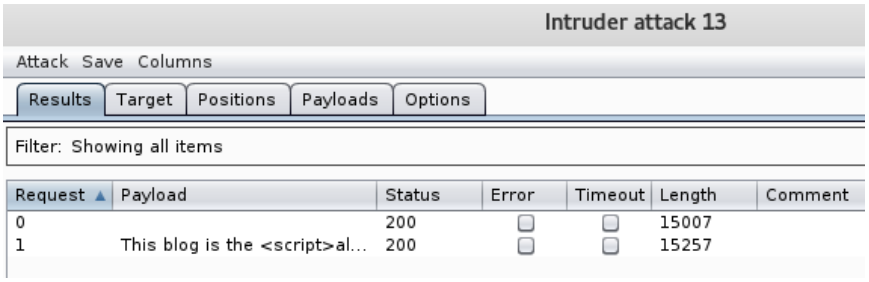
In Intruder > Positions, click on the Clear button, highlight test, and click on the Add button.



In the text field to the right of the Add button type the following:

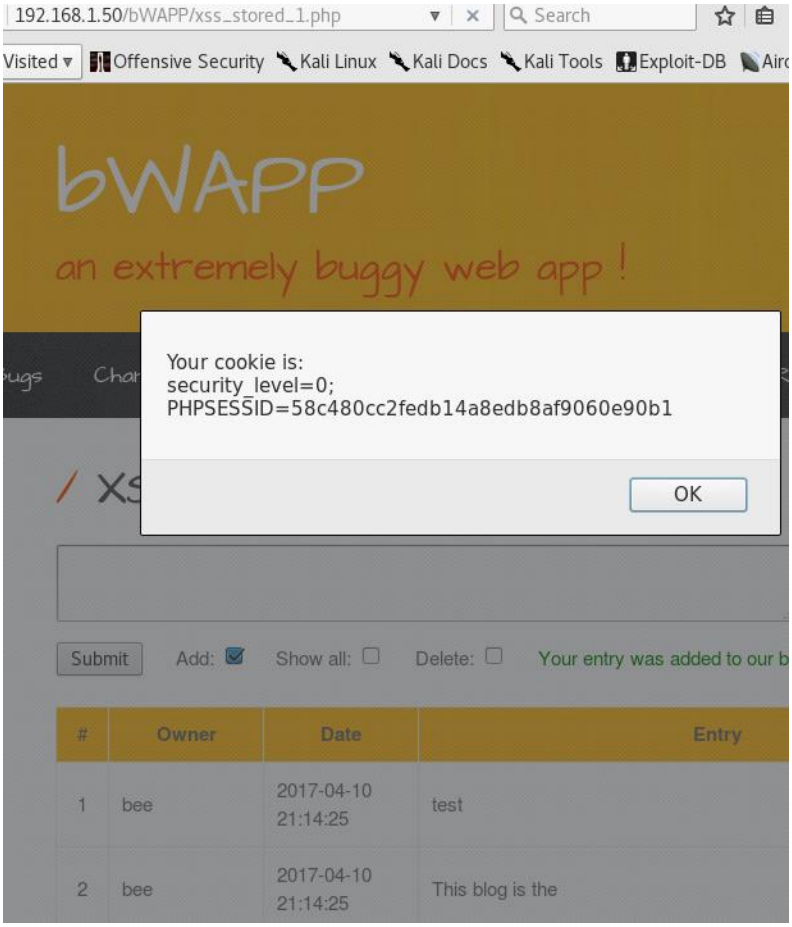
This blog is the <script>alert("Your cookie is:\n"+document.cookie)</script> best!

Click the Add button and then click Start Attack.

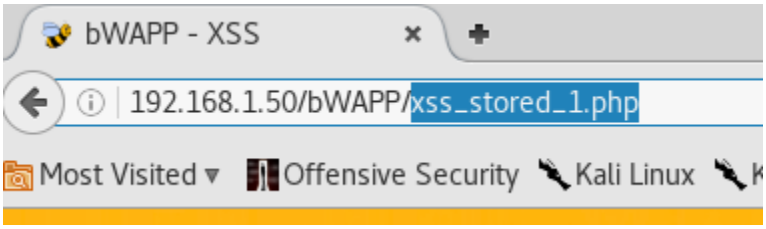


Notice that the payload has a 200 status meaning that it was accepted. It is 250 bytes longer than the base length.

Right click Request 1, click on Show response in browser, click copy, and then paste the link into the browser.



Above is a screenshot of the response in the browser.



Navigate to the portal page by highlighting xss_stored_1.php (copy this as we will return to this page), press the Delete key, and then press the Enter key.

Then add xxs_stored_1.php to the end of the URL and press Enter. The cookie message will display again because this script is stored in the database. Of course, the attacker will not alert you to notify you that you are being attacked. The attacker will

simply send your cookie to his listener so that he can hijack your session.

Stored XSS is very dangerous because the payload is not visible in the browser, and users might activate the payload by simply visiting an affected page.

Questions:

How might an attacker use XSS reflected?

What can attack do to disguise a malicious link?

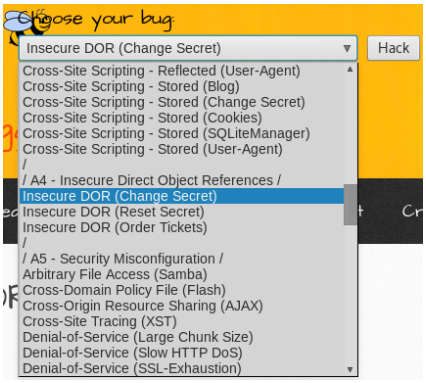
How do you prevent XSS?

Lab 7: Insecure Direct Object References

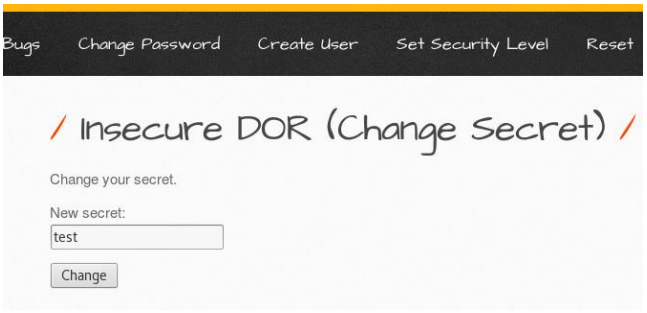
1. Introduction

When a developer exposes a reference to an internal implementation object, such as a directory, database key, or file, an insecure direct object reference occurs. In the case of insecure direct object references, an attacker is a user who is authorized on the system, but is able to change a parameter value somewhere else on the system where the attacker has no authorization. This is a common security weakness because applications often generate web pages utilizing the actual key or name of an object [20]. The vulnerability is easily detected by changing parameter values and analyzing the response generated by the web server. You can quickly verify proper authorization by analyzing the code. If the developer does not make object references unpredictable, an attacker can easily access all data available on the system referenced by the object [21]. Some examples of attacks of this nature include making changes to another user's information, and changing the price of an item on a web store.

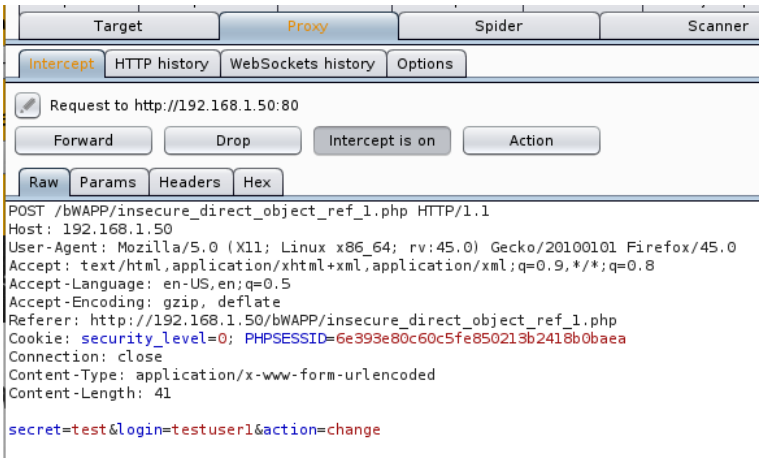
2. Attacking Insecure DOR (Change Secret)



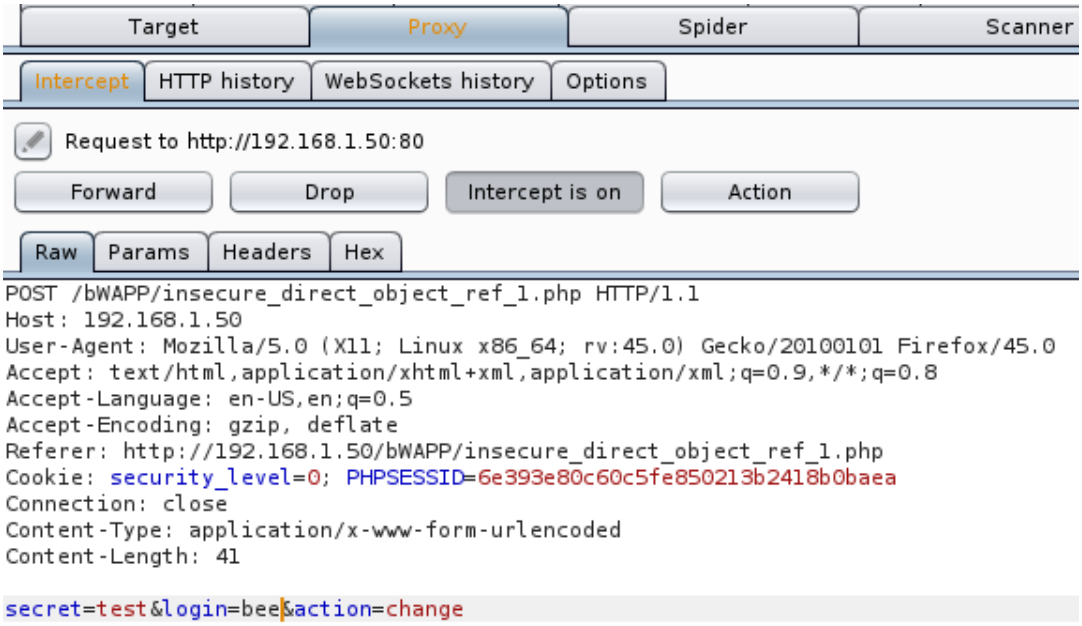
Create a test user (testuser1 is used in this lab), log in with that user, and then select the Insecure DOR (Change Secret) from the bug selection menu and then click Hack.



Make sure that Intercept is turned on in Burp Proxy. Then type test in the browser text field and click the Change button.



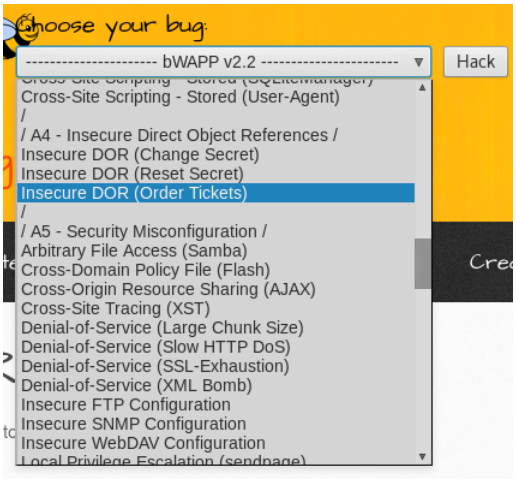
Notice the secret, login, and action at the bottom of the Raw Request window.



Change the login from testuser1 to bee, and then click on the forward button.

Congratulations! You just changed the secret of another user. Such an attack would allow an attacker to take control of another user's account.

3. Attacking Insecure DOR (Order Tickets)

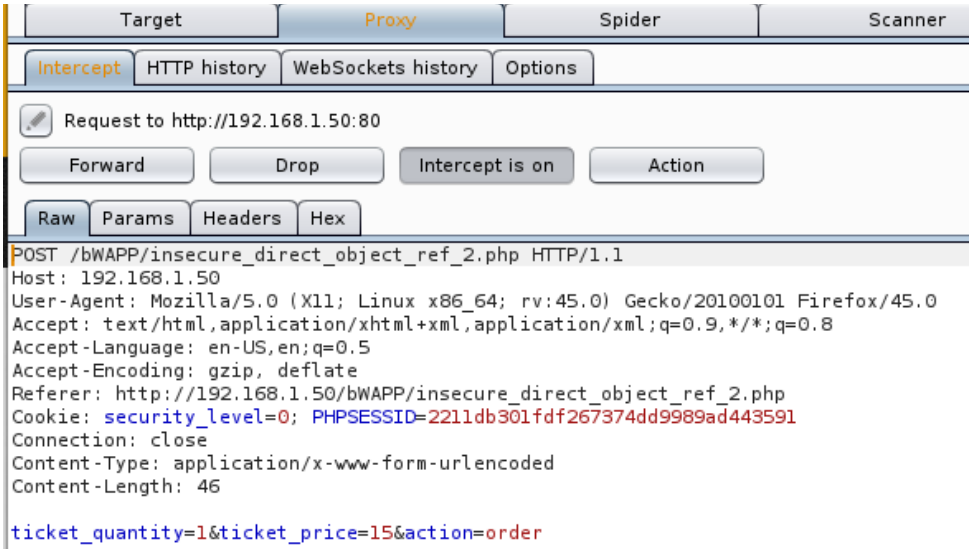


Select Insecure DOR (Order Tickets) from the bug selection menu and then click Hack.

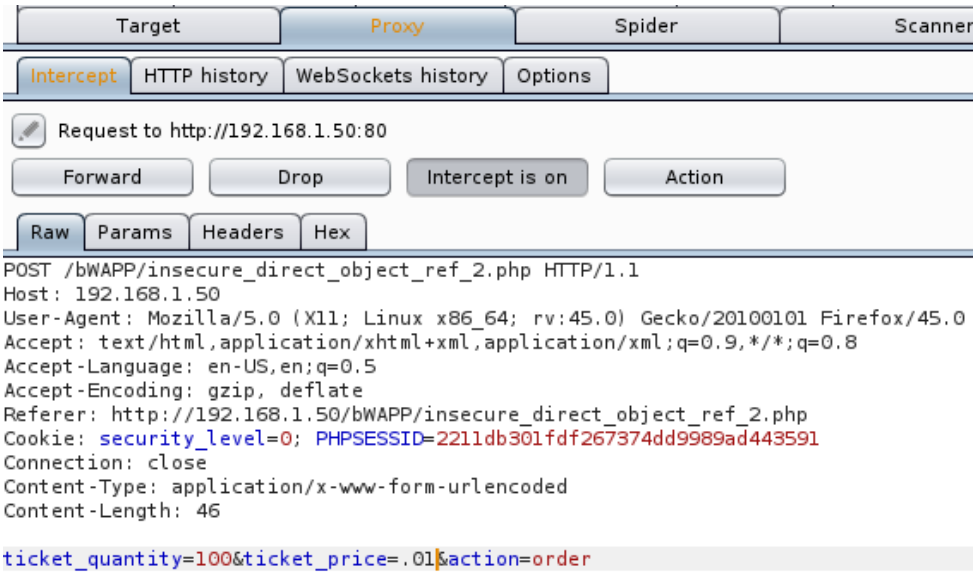
Enter 100 and then click Confirm.

Verify that the total about is 1500 EUR.

Turn Intercept on in Burp Proxy and then click Confirm in the browser.



Either right click and send this to the repeater to change the parameters, or change the parameters right in the Raw Request window.



After making the changes click on the Forward button.

We get 100 movie tickets for 1 EUR.

Questions:

What are the two methods of detecting insecure direct object references?

What is a directory traversal attack? Give an example.

How can a developer prevent insecure direct object references?

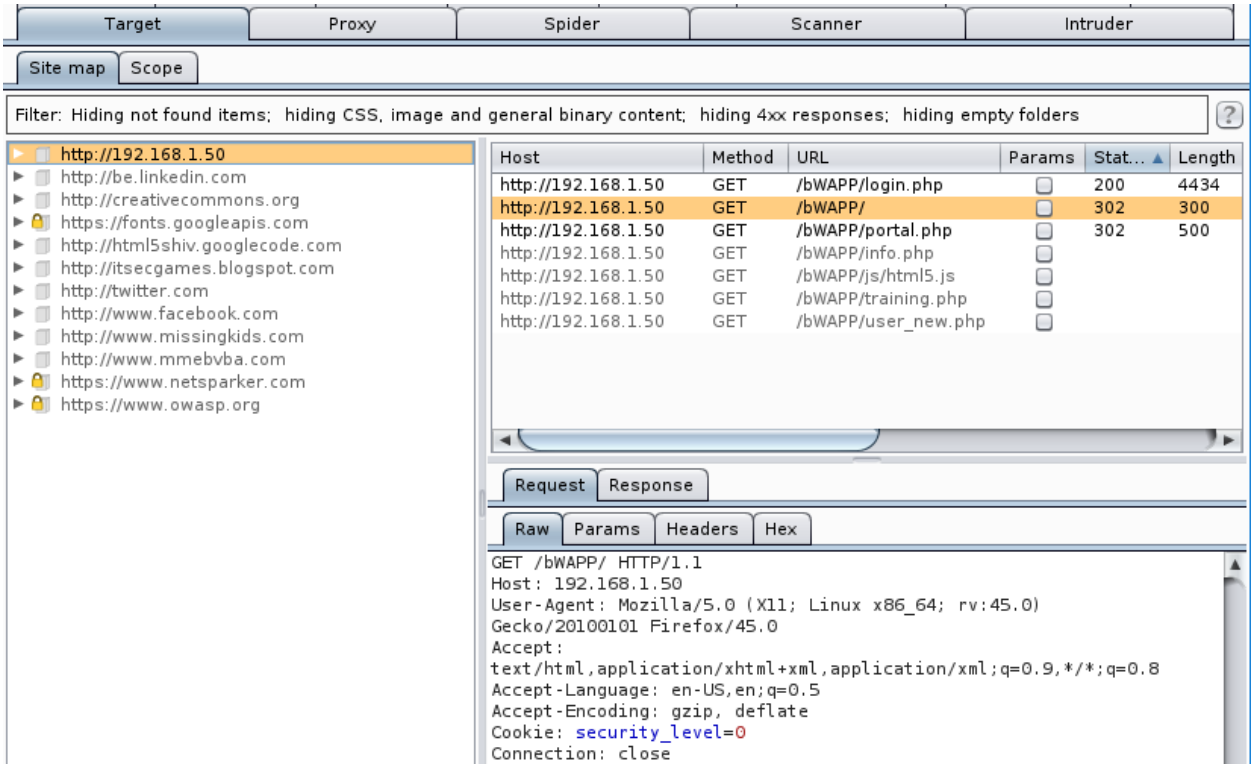
Lab 8: The Rest of the Burp Tools

1. Introduction

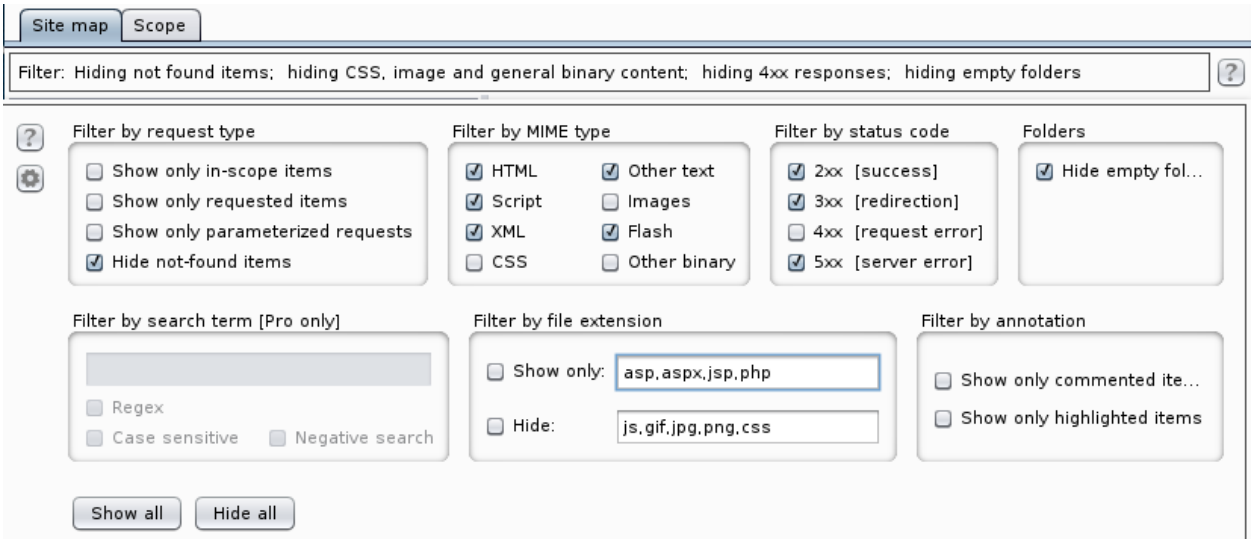
The Burp Tools that have not been used in the previous labs are Target, Spider, Scanner, Sequencer, Decoder, and Extender. The Target tool gives you an overview of your target web application's functionality and content. It contains the site map that provides you with detailed information about your target. It allows you to establish the scope of your penetration testing [22]. The Spider tool automatically crawls web applications to give you a list of all files that comprise the web application that you are targeting [23]. The Sequencer tool allows you to analyze the randomness of objects such as session tokens and password reset tokens [24]. The Decoder tools allows you to encode and decode data from and to URL, HTML, Base64, ASCII hex, Hex, Octal, Binary, and GZIP [25]. The Extender tool gives you the capability to use your own code or some third-party code to extend the functionality of Burp. You can access the BApp Store to download BApp files [26].

2. Target

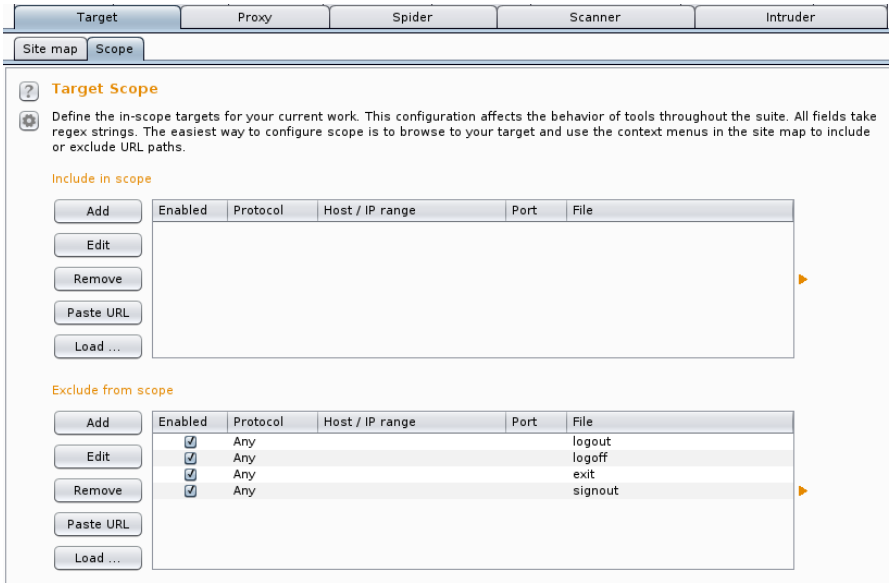
Open up Burp, open the browser, and navigate to the IP address that the bWAPP server resides on (192.168.1.50/bWAPP for this example). Click on the Target tab.



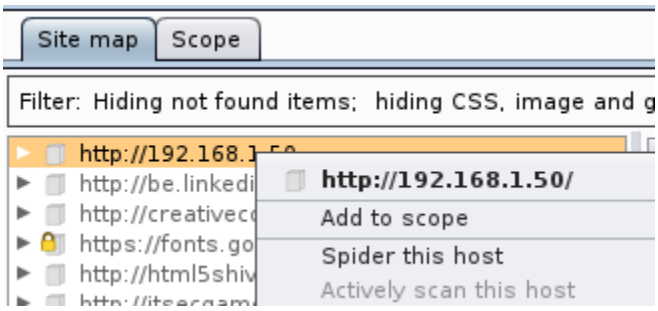
This is the results when you navigate to Burp Target. It is currently hiding not found items; CSS, image and general binary content; 4xx responses; and empty folders.



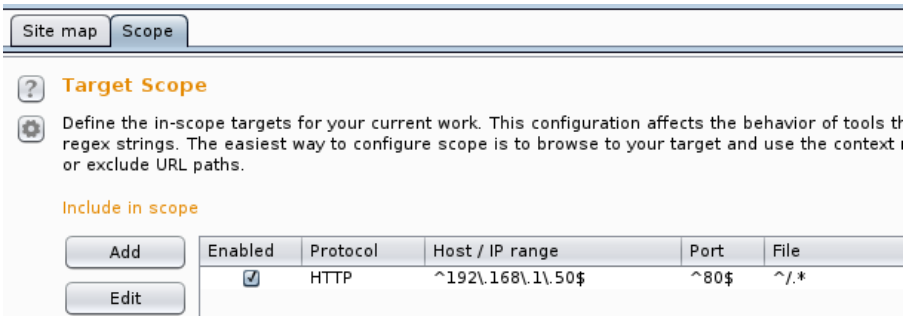
You can filter what is hidden and shown by clicking on the filter menu (bar below Site map and Scope tabs).



Click on the Scope tab to include and exclude objects from the scope of what you intend to spider from the web application.

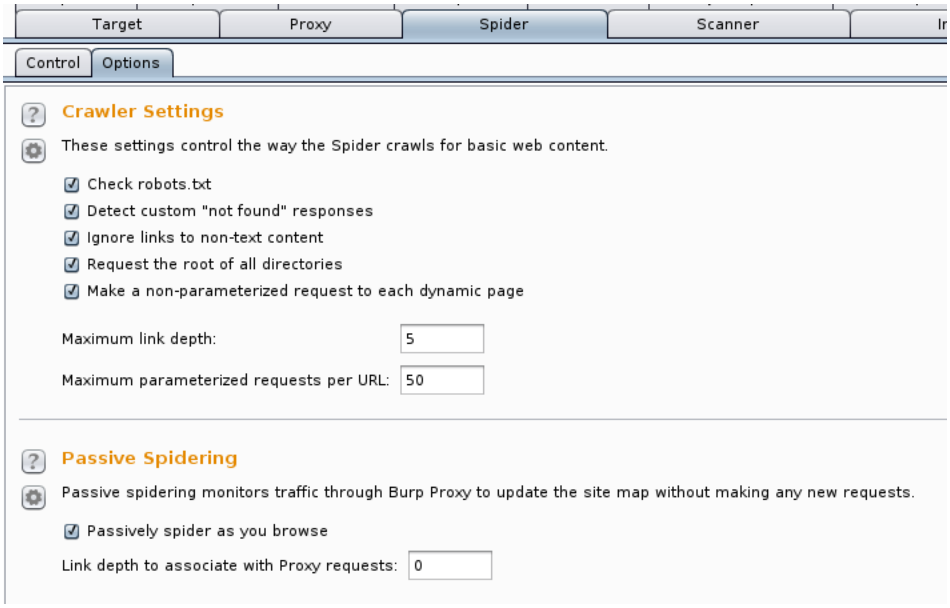


Navigate back to Site map, right click on the host address and click on Add to scope.

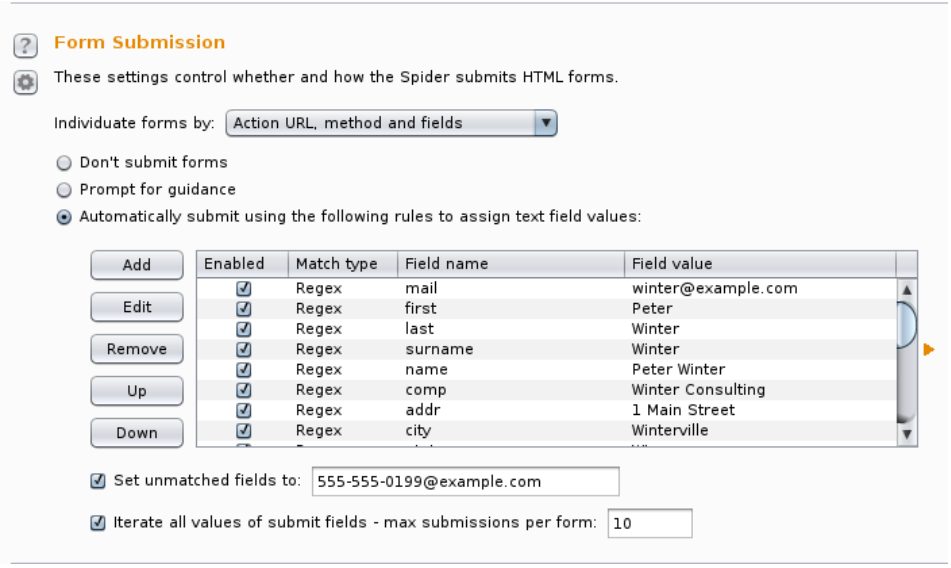


Navigate back to Scope to verify that it the host is included in scope.

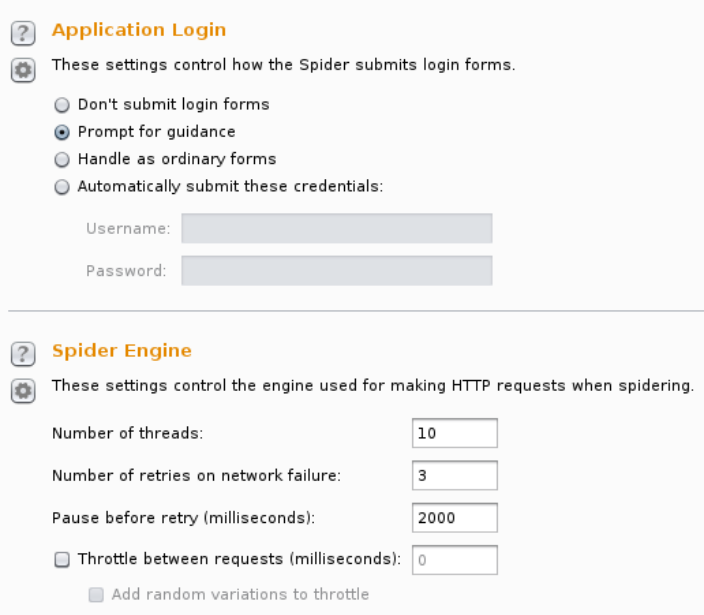
3. Spider



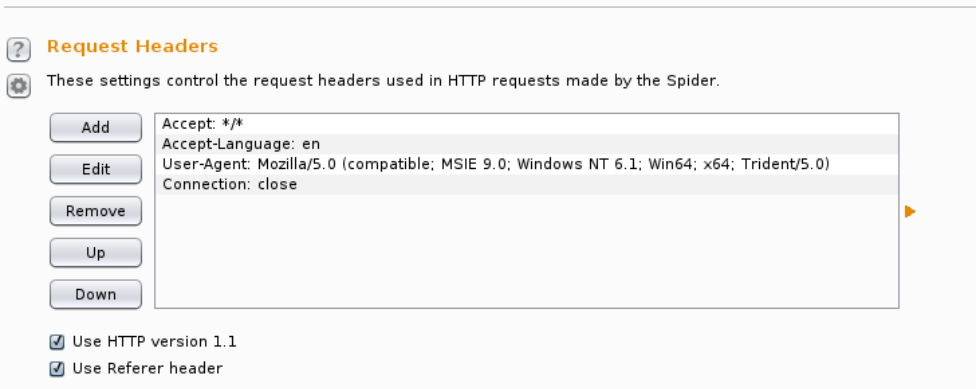
Navigate to Burp Spider > Options. Adjust Crawler Settings to determine what is displayed in the Target Site map. The Target Site map already had content in the previous step because “Passively spider as you browse” was checked.



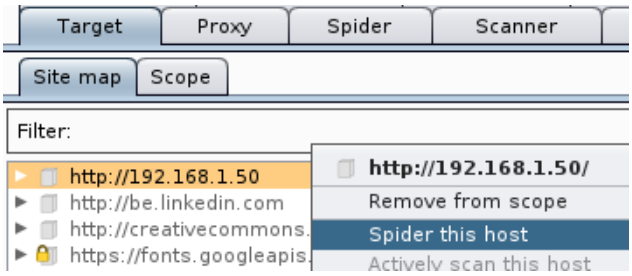
In order to crawl to certain pages, you may need to submit forms at times. Form Submission settings determine whether you access these other pages.



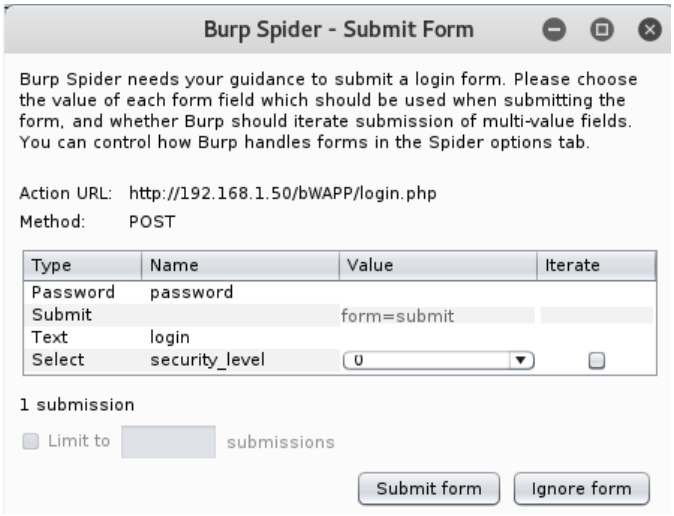
Application Login settings determine how forms are submitted when credentials are needed. Spider Engine settings are used for making HTTP requests when spidering. It is recommended that you adjust the number of threads to three to five as it can consume resources and slow your computer.



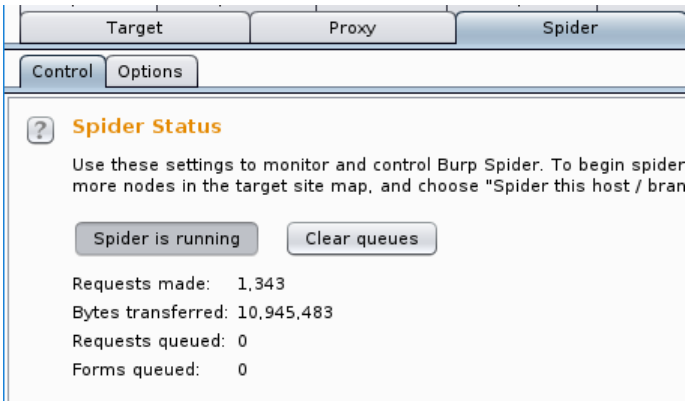
Request Headers settings can be adjusted for how you want the request headers to look.



To start spidering, navigate back to Burp Target > Site map, right click the host address and click Spider this host.



You will be prompted to fill in values for form submissions. Either provide values and then click Submit form, or click Ignore form.



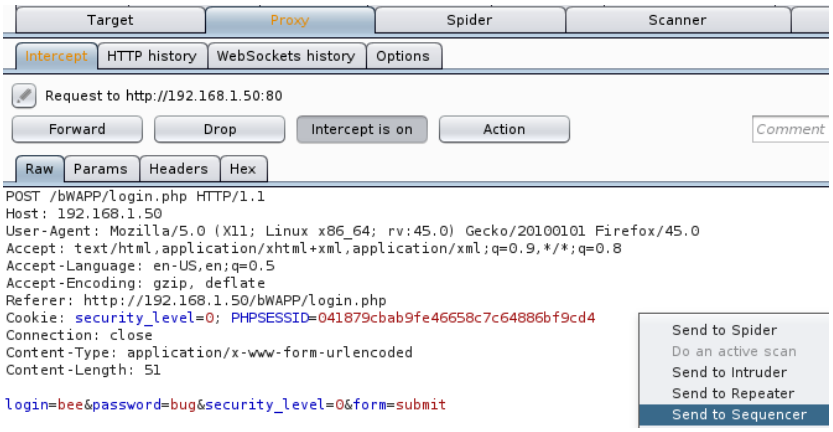
At any time, navigate back to Burp Spider to see the progress of the spidering.

4. Scanner

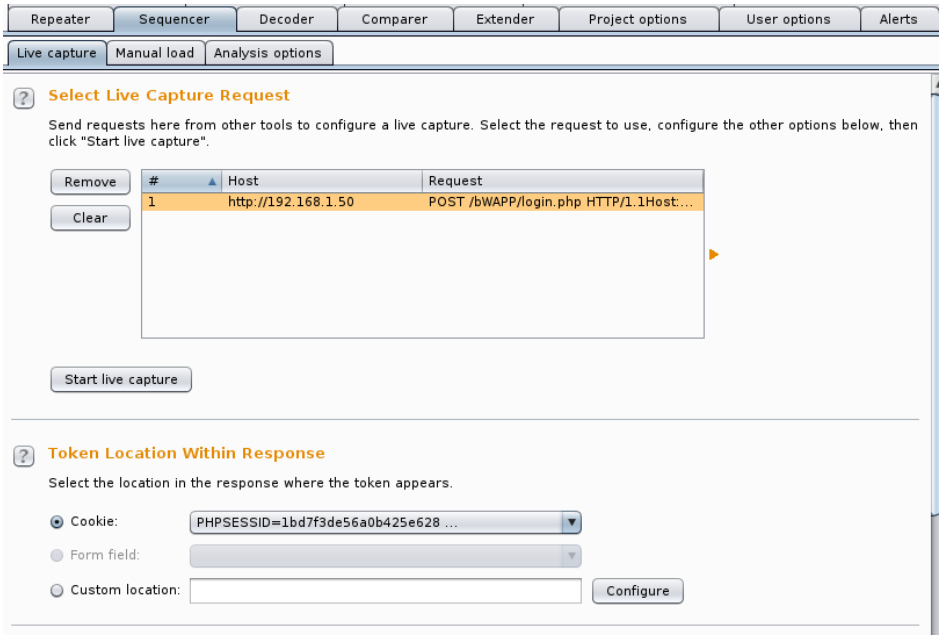
Burp Scanner is outside the scope of this lab as you need to purchase the professional version in order to use it. Some things of note in regards to Burp Scanner is that it can passively scan a web application to automatically determine vulnerabilities. You can also actively scan a particular web page for vulnerabilities [27]. Burp Scanner saves application testers a lot of time, but it is known to throw false positives, so manual testing is needed for verification.

5. Sequencer

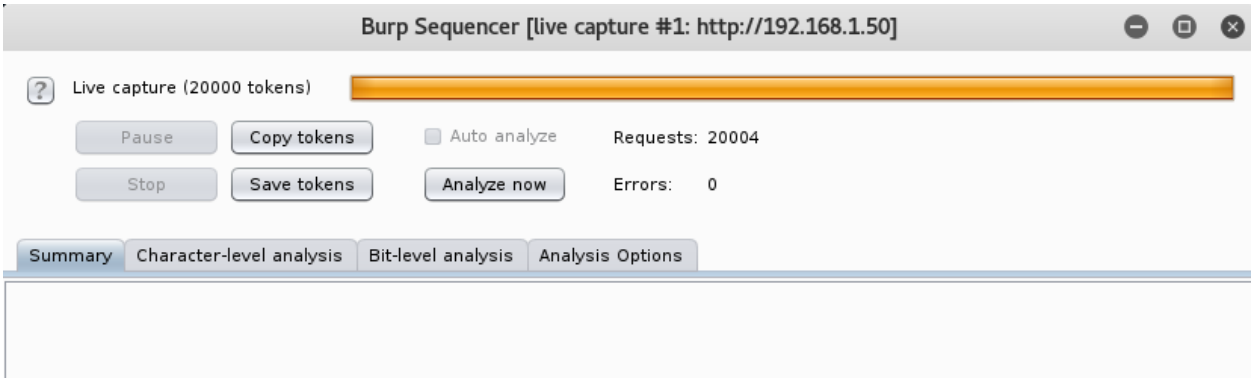
If you are logged into bWAPP, log out, enter the bee/bug credentials, turn Intercept on in Burp and then log in.



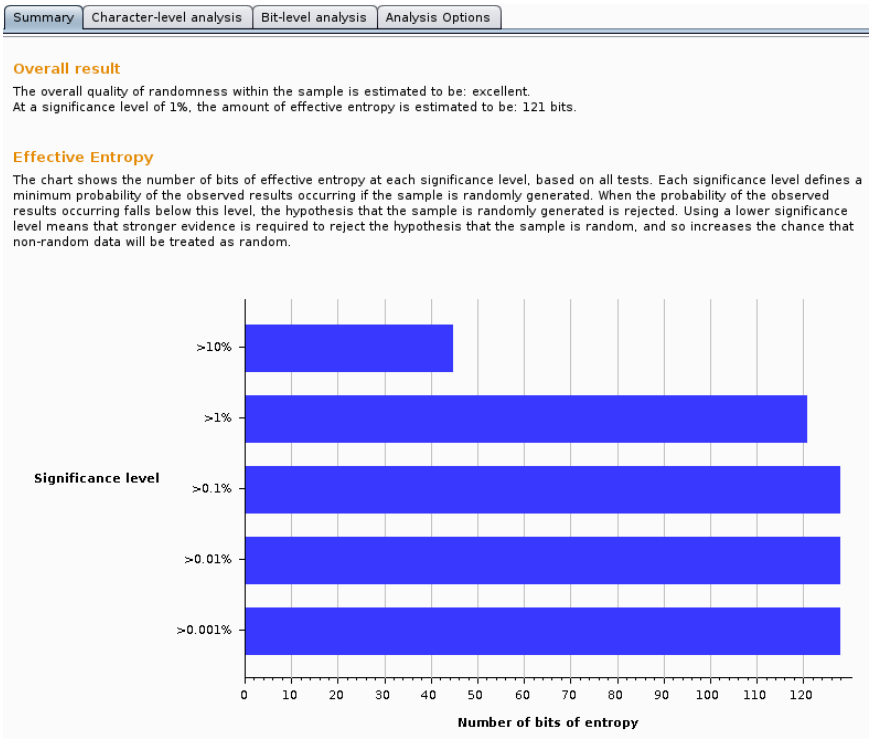
Right click in the Raw Request window and click Send to Sequencer.



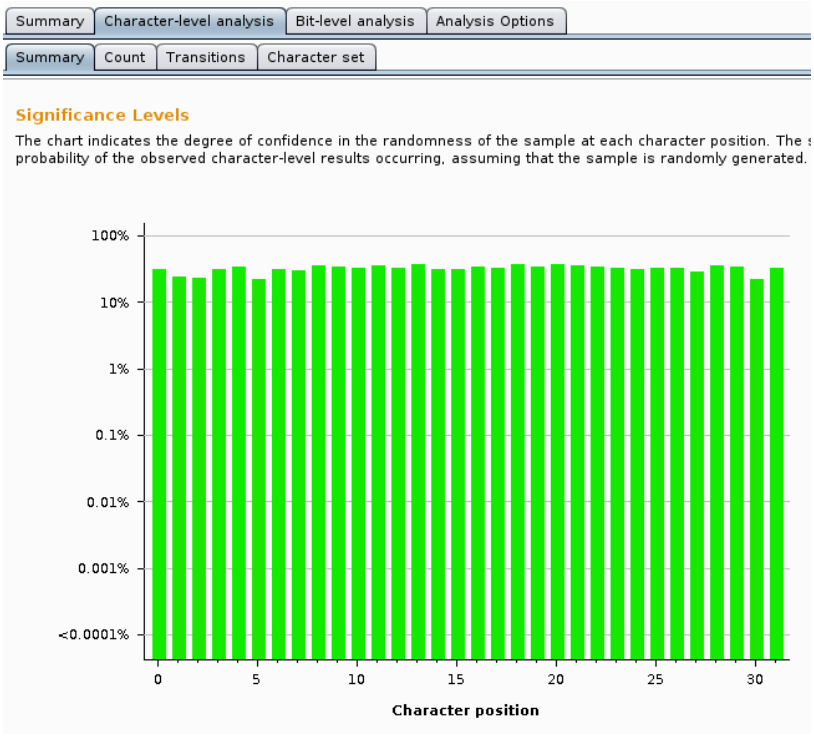
Navigate to Burp Sequencer > Live capture and ensure that the host is selected in the Select Live Capture Request. Also ensure that the session cookie is selected in the Token Location Within Response section. Click on Start live capture.



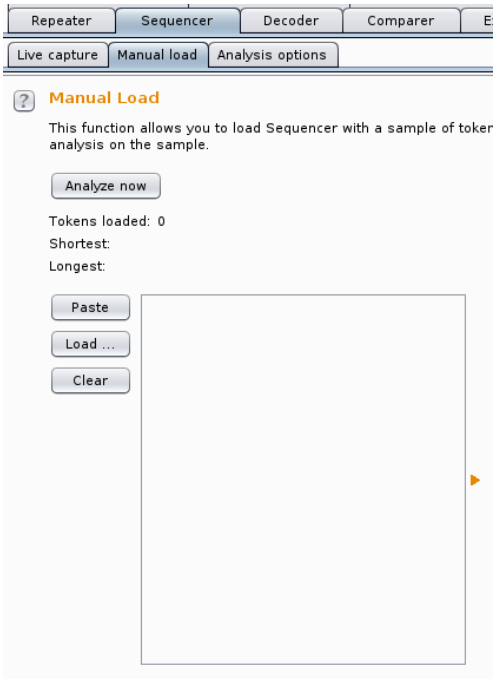
The live capture will stop after it has captured 20,000 tokens. You can also stop it at any time and then analyze what it has captured. After it stops, click Analyze now.



The summary shows that the overall randomization of session cookies is excellent.



Character-level analysis shows that each character is equally likely to be generated next.

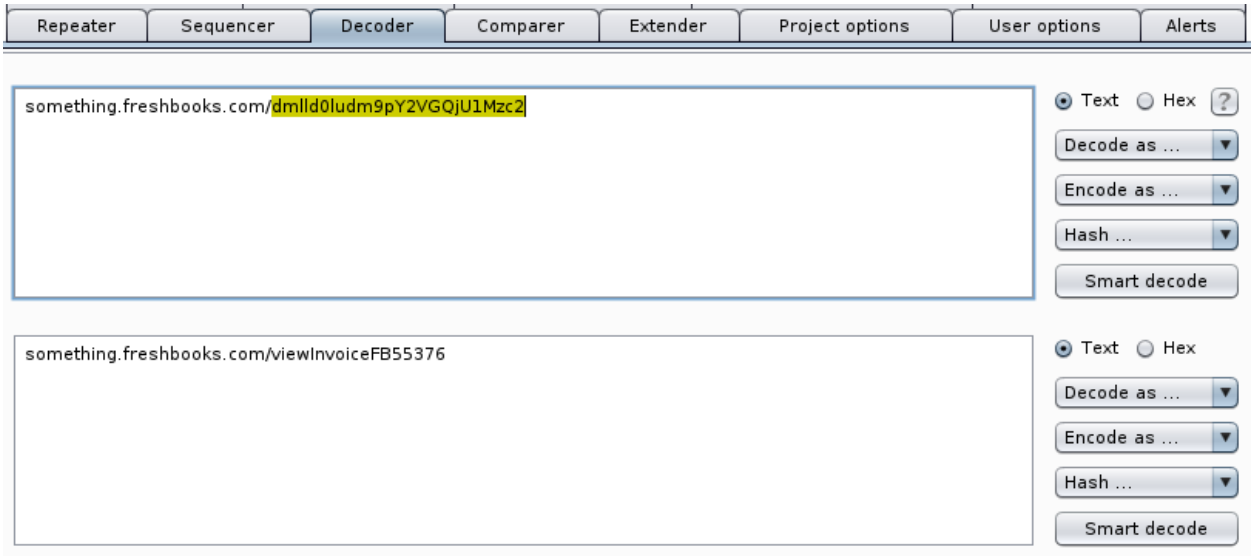


You can also manually load session cookies that were captured previously to analyze their randomness.

6. Decoder

There may be times when you need to decode a string that is in some other format. Burp Decoder allows you to decode and encode as plain, URL, HTML, Base64, ASCII hex, Hex, Octal, Binary, and Gzip. If you intercept a request and notice that the raw request contains data in a different format such as base64, you can right click on the raw request and send it on over to Burp Decoder.

Some websites may even use base64 encoding in their URLs. In December 2013, stackexchange user Jor-el posted that freshbooks.com does this for viewing resources and even POSTing sensitive data [28]. Let’s use his example to demonstrate Burp Decoder’s functionality.



Copy the URL into the Decoder text area. Then double click on the string after the forward slash to highlight it, and then click Decode as base64.

In some situations, clicking Smart decode is faster for decoding data. However, it does not work for base64 and short strings. Burp Decoder also has the functionality of encoding a string with SHA-384, SHA-224, SHA-256, MD2, SHA, SHA-512, and MD5.

7. Extender

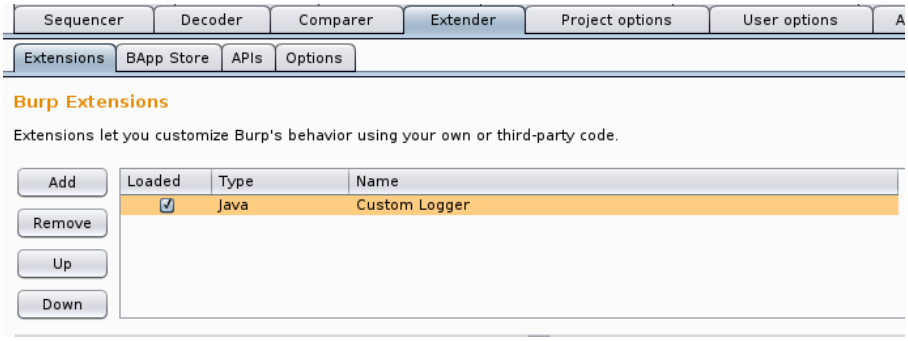
Installing 3rd party Burp extensions:

Burp Extender allows you to import third party extensions or use your own created extensions.

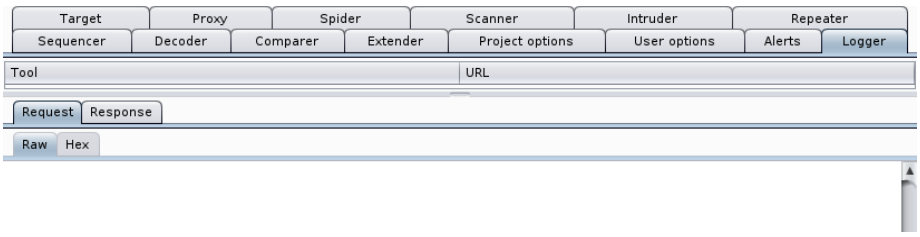
If you have been following along with all of the labs, your network is currently set up for using the internal network. From the VirtualBox menu select Devices > Network > Network Settings. Then change Attached to from Internal Network to NAT. Verify that you are able to access the Internet. In Burp, navigate to Extender > BApp Store and click Refresh list in the lower left.

This is what you should see after refreshing the list.

Scroll down to Custom Logger in the BApp Store and click Install.



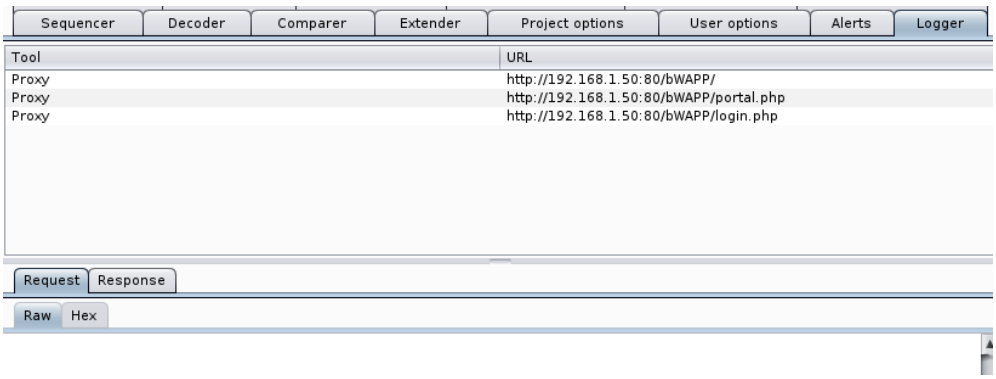
If you navigate to Burp Extender > Extensions, you will see that Custom Logger is loaded to the list of Burp Extensions.



You will also notice that the new tab, Logger, has been added to Burp Suite.

Turn off Intercept if it is on in Burp Proxy, close out of Firefox, switch the VM network back to Internal Network from NAT, turn Intercept back on, open Firefox and navigate to

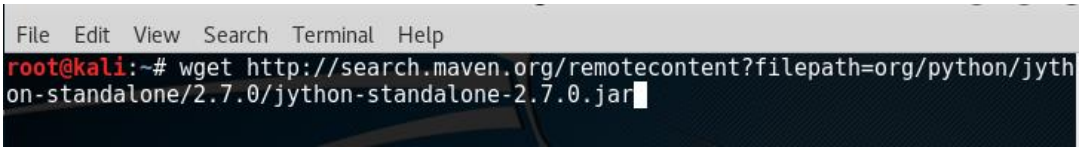
the bWAPP URL.



You now see that the Proxy tool interfaced with three URLs.

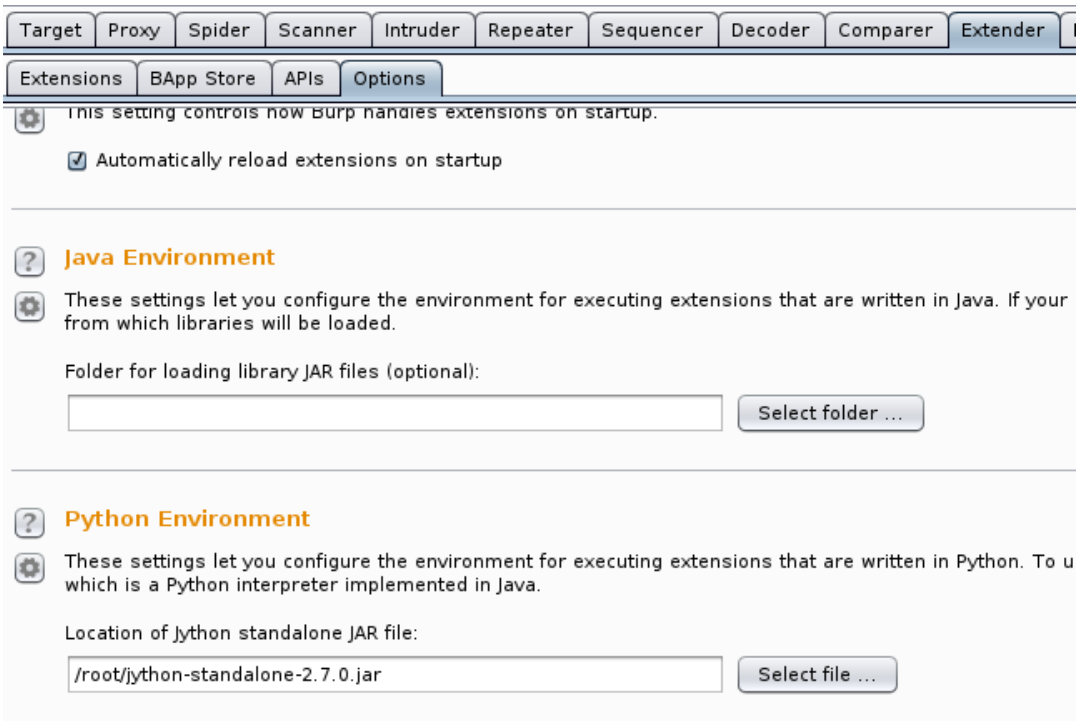
Creating and installing your own Burp Extension:

First you need to install the Jython standalone jar file. Switch your VM's network from Internal Network to NAT.



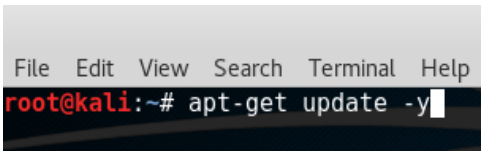
Navigate to www.jython.org/downloads.html, copy the link address of the latest version of the Jython Standalone Jar, type wget in the Kali command line, paste the link address, and press enter.

When it completes, rename it to jython-standalone-2.7.0.jar

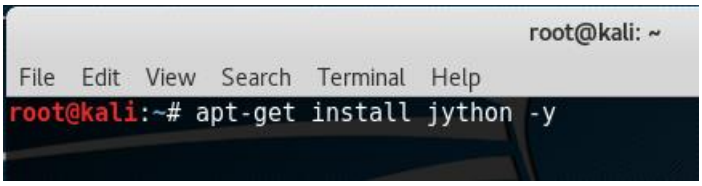


Navigate to Burp Extender > Options and in the Python Environment, click Select file and select the Jython Standalone Jar file that you just installed.

Next, we must install jython.



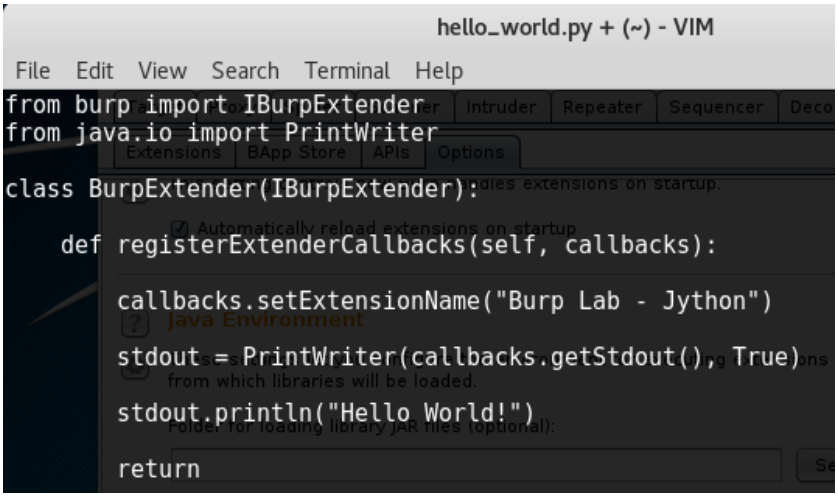
From the command line run apt-get update -y



From the command line run apt-get install jython -y

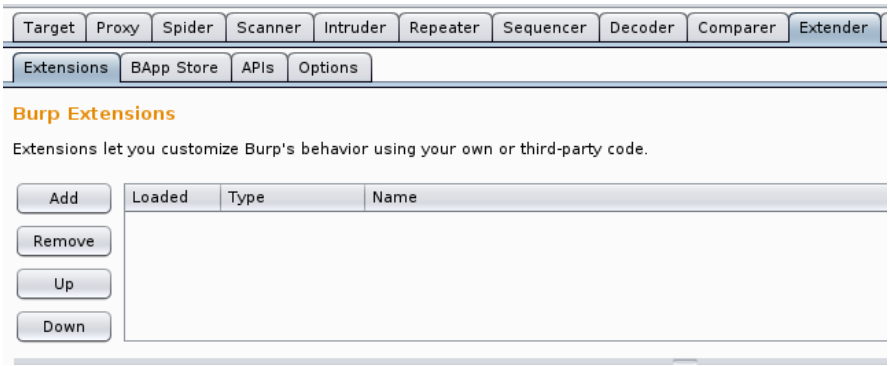
Jython is a Java/Python hybrid type of library that allows developers to create extensions in python. This is needed because Burp is programmed in Java.

Now we are going to create a simple hello world program with to import into Burp Suite [29].

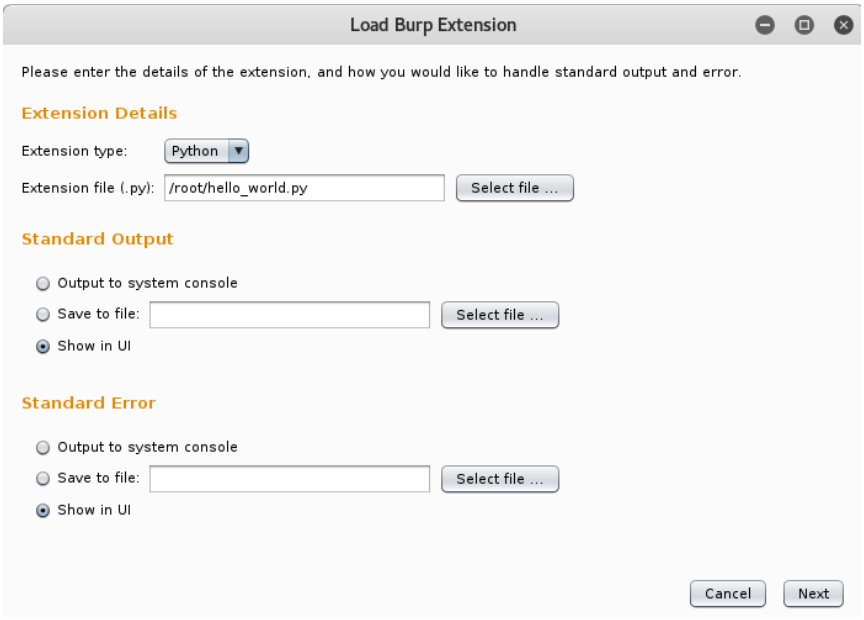


Create a file called hello_world.py with the same content as above.

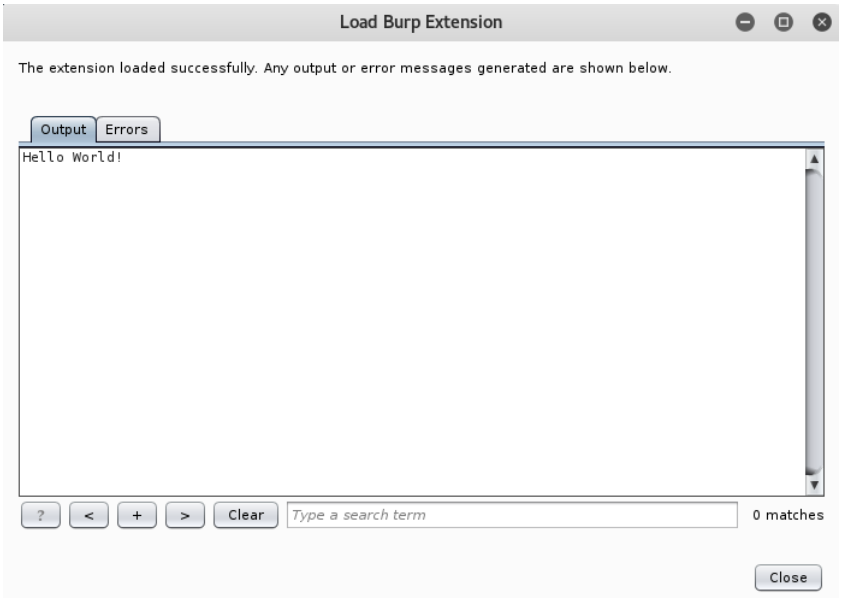
Whenever creating a Burp Extension, you have to import IBurpExtender. We also hve to import PrintWriter so that we can print “Hello World!”.



Navigate to Burp Extender > Extensions and click Add.



Change the Extension type to Python, click Select file, browse to the extension that you just created, select it, and then click Next in the lower right.



Congratulations! You just created and imported your first Burp extension.

Creating anything more than this is outside the scope of this lab. However, there are a bunch of tutorials for learning to create Burp extensions on the Internet.

References

[1] Verizon, “2016 Data Breach Investigations Report,” 2016. [Online]. Available: http://www.verizonenterprise.com/resources/reports/rp_dbir-2016-executive-summary_xg_en.pdf. [Accessed: Mar. 15, 2017].

[2] M. Rouse, “What is Web application (Web app)? - Definition from WhatIs.com,” SearchSoftwareQuality, 2017. [Online]. Available: http://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app. [Accessed: 15-Mar-2017].

[3] R. Sethi, “Why Developers Build Insecure Apps,” The Huffington Post, 02-Jul-2014. [Online]. Available: http://www.huffingtonpost.com/rohit-sethi/why-developers-build-inse_b_5549482.html. [Accessed: 15-Mar-2017].

[4] M. Curphey, “The Start of OWASP – A True Story,” SourceClear, 27-May-2014. [Online]. Available: https://www.sourceclear.com/blog/The-Start-of-OWASP--A-True-Story/. [Accessed: 01-Apr-2017].

[5] OWASP, “Category:OWASP Top Ten Project,” Category:OWASP Top Ten Project - OWASP, 2017. [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. [Accessed: 01-Apr-2017].

[6] OWASP, “Category:OWASP Project,” Category:OWASP Project - OWASP, 2017. [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_Project. [Accessed: 01-Apr-2017].

[7] OWASP, “OWASP Zed Attack Proxy Project,” OWASP Zed Attack Proxy Project - OWASP, 2017. [Online]. Available: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project. [Accessed: 01-Apr-2017].

[8] OWASP, “Category:OWASP Mutillidae,” Category:OWASP Mutillidae - OWASP, 2017. [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_Mutillidae. [Accessed: 01-Apr-2017].

[9] ethicalhack3r, “ethicalhack3r/DVWA,” GitHub, 2017. [Online]. Available: https://github.com/ethicalhack3r/DVWA. [Accessed: 01-Apr-2017].

[10] M. Mesellem, “itsecgames,” itsecgames, 2014. [Online]. Available: http://www.itsecgames.com/. [Accessed: 01-Apr-2017].

[11] PortSwigger, “Burp Suite Editions_,” PortSwigger Web Security, 2017. [Online]. Available: https://portswigger.net/burp/. [Accessed: 01-Apr-2017].

[12] E-SPIN International, “E-SPIN International,” Difference between Burp Suite Free and Paid (Pro) version? : E-SPIN International, 06-May-2016. [Online]. Available: https://espincorp.freshdesk.com/support/solutions/articles/1000230595-difference-between-burp-suite-free-and-paid-pro-version-. [Accessed: 01-Apr-2017].

[13] Polynomial, “SQL Injection - UNION SELECT and returning a 'keyword' to find exploitable columns,” web application - SQL Injection - UNION SELECT and returning a 'keyword' to find exploitable columns - Information Security Stack Exchange, 04-Apr-2013. [Online]. Available: https://security.stackexchange.com/questions/33741/sql-injection-union-select-and-returning-a-keyword-to-find-exploitable-colum. [Accessed: 04-Mar-2017].

[14] K. Tank, “bWAPP SQL Injection GETSearch,” YouTube, 17-Sep-2015. [Online]. Available: https://www.youtube.com/watch?v=2K1fGBfHrw0. [Accessed: 04-Mar-2017].

[15] PortSwigger, “Using Burp to Attack Session Management,” PortSwigger Web Security, 2017. [Online]. Available: https://support.portswigger.net/customer/portal/articles/1964053-using-burp-to-attack-session-management. [Accessed: 04-Mar-2017].

[16] G. Fogerlie, “Hack a Website Login Form Using Burp Suite - Cyber Security - 4k Video,” YouTube, 11-Mar-2016. [Online]. Available: https://www.youtube.com/watch?v=OXIjNfX7BW8#t=300. [Accessed: 04-Mar-2017].

[17] webpwnized, “Introduction to Burp-Suite Comparer Tool,” YouTube, 08-Feb-2012. [Online]. Available: https://www.youtube.com/watch?v=KxqY_bp13gc. [Accessed: 04-Mar-2017].

[18] OWASP, “Testing for Reflected Cross site scripting (OTG-INPVAL-001),” Testing for Reflected Cross site scripting (OTG-INPVAL-001) - OWASP, 2017. [Online]. Available: https://www.owasp.org/index.php/Testing_for_Reflected_Cross_site_scripting_(OTG-INPVAL-001). [Accessed: 04-Mar-2017].

[19] S. Morgenroth, “What is the Cross-site Scripting (XSS) Vulnerability & How to Prevent it?,” Cross-site Scripting | What is XSS Attack? | Netsparker, 22-Dec-2016. [Online]. Available: https://www.netsparker.com/blog/web-security/cross-site-scripting-xss/. [Accessed: 04-Mar-2017].

[20] tutorialspoint.com, “Security Testing - Insecure Direct Object References,” www.tutorialspoint.com, 2017. [Online]. Available: https://www.tutorialspoint.com/security_testing/insecure_direct_object_reference.htm. [Accessed: 04-Mar-2017].

[21] H. Wang, “Preventing Insecure Direct Object References In App Development,” 2014. [Online]. Available: http://www.cs.tufts.edu/comp/116/archive/fall2014/hwang.pdf. [Accessed: 04-Mar-2017].

[22] PortSwigger, “Using the Target Tool,” PortSwigger Web Security, 2017. [Online]. Available: https://portswigger.net/burp/help/target_using.html. [Accessed: 15-Apr-2017].

[23] PortSwigger, “Using Burp Spider,” PortSwigger Web Security, 2017. [Online]. Available: https://portswigger.net/burp/help/spider_using.html. [Accessed: 15-Apr-2017].

[24] PortSwigger, “Getting Started With Burp Sequencer,” PortSwigger Web Security, 2017. [Online]. Available: https://portswigger.net/burp/help/sequencer_gettingstarted.html. [Accessed: 15-Apr-2017].

[25] PortSwigger, “Burp Decoder Documentation,” PortSwigger Web Security, 2017. [Online]. Available: https://portswigger.net/burp/help/decoder.html. [Accessed: 15-Apr-2017].

[26] PortSwigger, “Burp Extender Documentation,” PortSwigger Web Security, 2017. [Online]. Available: https://portswigger.net/burp/help/extender.html. [Accessed: 15-Apr-2017].

[27] PortSwigger, “Using Burp Scanner,” PortSwigger Web Security, 2017. [Online]. Available: https://portswigger.net/burp/help/scanner_using.html. [Accessed: 15-Apr-2017].

[28] J.-el, “Purpose of using base64 encoded urls,” web application - Purpose of using base64 encoded urls - Information Security Stack Exchange, 03-Dec-2013. [Online]. Available: https://security.stackexchange.com/questions/46362/purpose-of-using-base64-encoded-urls. [Accessed: 15-Apr-2017].

[29] nVisium, “intro to burp extender jython,” YouTube, 08-Jun-2016. [Online]. Available: https://www.youtube.com/watch?v=4f05lNULX1I. [Accessed: 15-Apr-2017].