





Hacking Web Applications

Module 13

Hacking Web Applications

Hacking web applications refers to carrying out unauthorized access of a website or the website details.

ICON KEY

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

Lab Scenario

A web application is an application that is accessed by users over a network such as the Internet or an intranet. The term may also mean a computer software application that is 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 due to the ubiquity of web browsers, and the convenience of using a web browser 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 other functions.

Web hacking refers to exploitation of applications via HTTP which can be done by manipulating the application via its graphical web interface, tampering the Uniform Resource Identifier (URI) or tampering HTTP elements not contained in the URI. Methods that can be used to hack web applications are SQL Injection attacks, Cross Site Scripting (XSS), Cross Site Request Forgeries (CSRF), Insecure Communications, etc.

As an expert **Ethical Hacker** and **Security Administrator**, you need to test web applications for cross-site scripting vulnerabilities, cookie hijacking, command injection attacks, and secure web applications from such attacks.

Lab Objectives


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

- Parameter tampering
- Directory traversals
- Cross-Site Scripting (XSS)
- Web Spidering
- Cookie Poisoning and cookie parameter tampering
- Securing web applications from hijacking

Lab Environment

To carry out the lab, you need:

- A computer running **Windows Server 2012**

 **Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv8 Module 13 Hacking Web Applications**

- A web browser with an Internet connection

Lab Duration

Time: 50 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 within the client **Web browser**.



TASK 1

Overview

Lab Tasks

Recommended labs to assist you in web application:

- Parameter tampering attacks
- Cross-site scripting (XSS or CSS)
- Web spidering
- Website vulnerability scanning using Acunetix WVS

Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target's security posture and exposure.

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



Hacking Web Applications

Though web applications enforce certain security policies, they are vulnerable to various 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 the DailyNews, Cyber-crime targeted in new ICT policy; the government is reviewing the current Information and Communication Technology (ICT) policy in quest to incorporate other relevant issues, including addressing cyber-crime, reported to be on the increase.

“Many websites and web applications are vulnerable to security threat including the government's and non-government's websites, we are therefore cautious to ensure that the problem is checked”, Mr. Urasa said. Citing some of the reasons leading to hacking, he said inadequate auditing in website and web applications caused by lack of standard security auditing were among problems that many web developers faced.

As an expert **Ethical Hacker** and **Security Administrator**, you should be aware of all the methods that can be employed by an attacker towards hacking web applications and accordingly you can implement a countermeasure for those attacks. Hence, in this lab you will learn how to hack a website with vulnerabilities.

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)

Lab Environment


To carry out the lab, you need:

- Powergym website is located at **D:\CEH-Tools\CEHv8 Lab Prerequisites\Websites\Powergym**

 **Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv8 Module 13 Hacking Web Applications**

Module 13 – Hacking Web Applications

- Run this lab in Windows Server 2012 host machine
- Microsoft SQL server 2012
- A web browser with an Internet connection

 <http://localhost/powergym>

Lab Duration

Time: 20 Minutes

Overview of Web Applications

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 the client **web browser**.

TASK 1

Parameter Tampering

Lab Tasks

Web **parameter tampering** attacks involve the **manipulation** of parameters exchanged between a client and a server in order to **modify** application data, such as user credentials and permissions, price, and quantity of products.

1. To launch a web browser move your mouse cursor to lower left corner of your desktop, and click **Start**

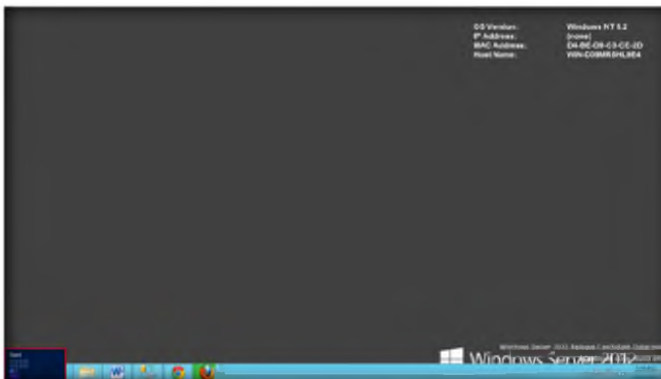




FIGURE 1.1: Windows Server 2012 – Desktop view

2. From start menu apps click on any browser app to launch. In this lab we are using **Firefox** browser

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

Module 13 – Hacking Web Applications

 Parameter tampering can be employed by attackers and identity thieves to obtain personal or business information regarding the user surreptitiously.

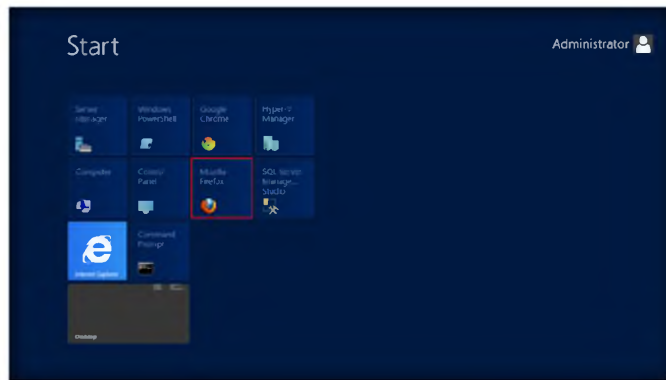


FIGURE 1.2: Windows Server 2012 – Start Menu Apps

3. Type <http://localhost/powergym> in the address bar of the web browser, and press **Enter**
4. The **Home** page of **Powergym** appears

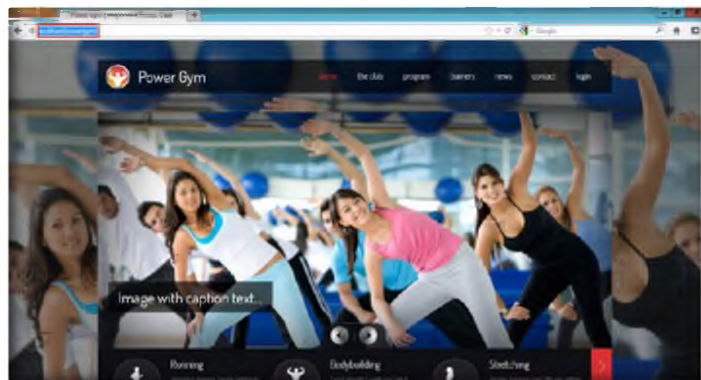



FIGURE 1.3: Powergym home page

 Countermeasures specific to the prevention of parameter tampering involve the validation of all parameters to ensure that they conform to standards concerning minimum and maximum allowable length, allowable numeric range, allowable character sequences and patterns, whether or not the parameter is actually required to conduct the transaction in question, and whether or not null is allowed.

5. Assume that you are **not a member** of this site and you don't have a **Login ID** for this website
6. In the address bar, try to tamper the parameter by entering various keywords. Perform a **Trial and Error** on this website
7. Click on trainers and type '**Sarah Partink**' in the search option. Click **Search**


Module 13 – Hacking Web Applications



FIGURE 1.4: Powergym Trainers page



FIGURE 1.5: Powergym ID page

 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.

8. Now tamper with the parameters **id=Sarah Partink** to **id=Richard Peterson** in the address bar and press **Enter**
9. You get the search results for **Richard Peterson** without actually searching **Sarah Partink** in search field. This process of changing the **id value** and getting the result is known as **parameter tampering**

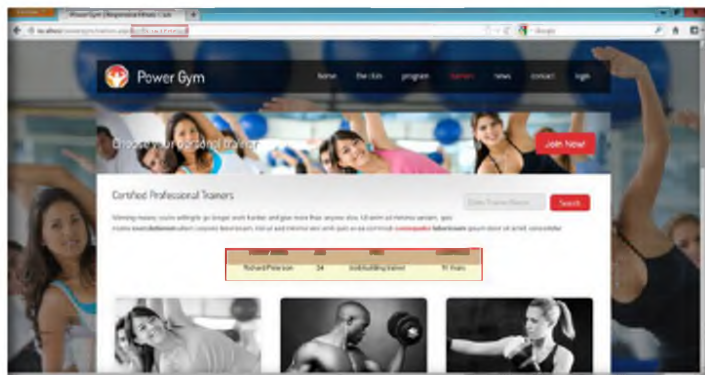


FIGURE 1.6: Powergym with parameter tampering

10. You have browsed a site to which you don't have **login ID** and access to view details of **products**. You have performed this by **parameter tampering**

T A S K 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.

11. Open a web browser, type <http://localhost/powergym>, and press **Enter**
12. The **home page** of Powergym appears

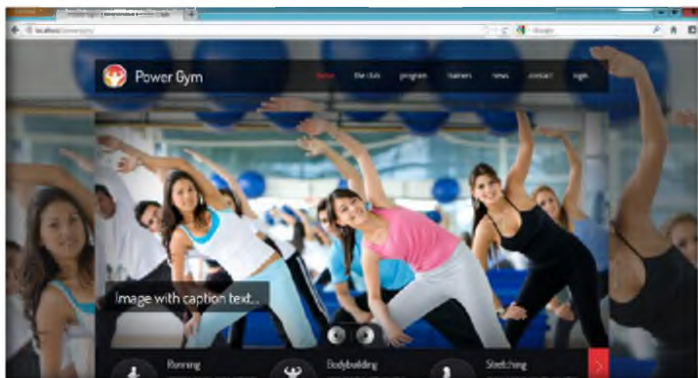


FIGURE 1.7: Classic Cars Collection home page

13. To log in to the site, click on **LOGIN**

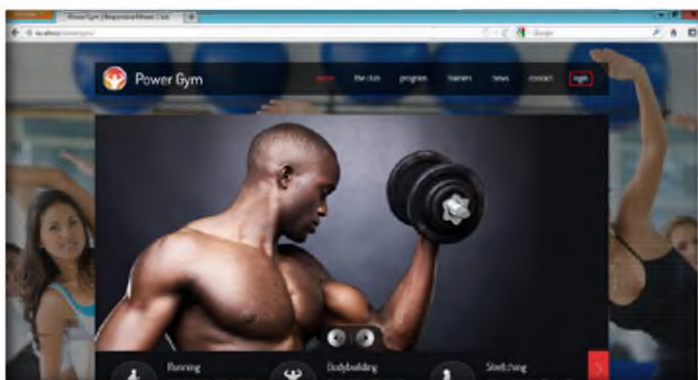



FIGURE 1.8: Powergym home page

14. The **Login page** of the Powergym website appears
15. Enter “**sam**” as **User name** and “**test**” as **Password** in the respective fields and click on **Login** to log into the website

 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.

 <http://localhost/powergym>

Module 13 – Hacking Web Applications

📖 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.

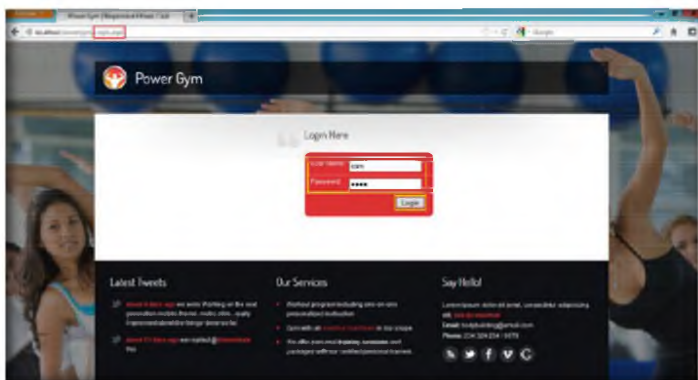


FIGURE 1.9: Powergym Login page

16. After you log in to the website, find an input field page where you can enter **cross-site scripting**. In this lab, the **contact** page contains an input field where you can enter cross-site script
17. After logging in it will automatically open **contact** page

📖 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.

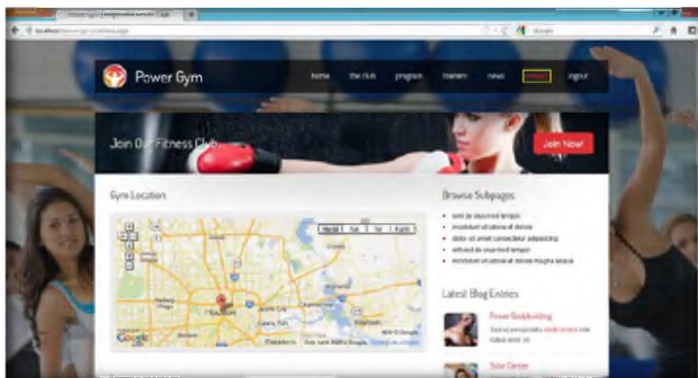



FIGURE 1.10: Powergym Contact page

18. On the contact page, enter your login name (or any name) in **Your name** field
19. Enter any email in email address field. In the **Your message** field, enter this cross-site script, **Chris, I love your GYM! <script>alert("You have been hacked")</script>** and click **Submit**
20. On this page, you are **testing** for cross-site scripting vulnerability

Module 13 – Hacking Web Applications

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

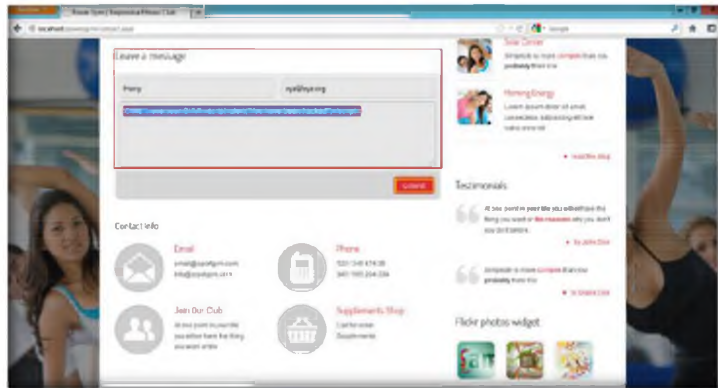


FIGURE 1.11: Powergym contact page with script

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

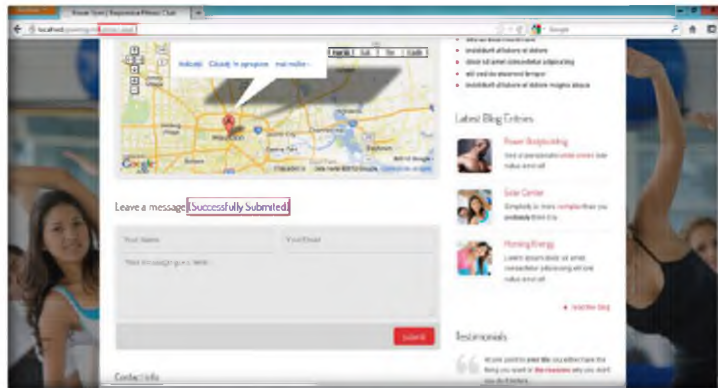



FIGURE 1.12: Powergym contact page script submitted successfully

22. Whenever any **member** comes to the contact page, the **alert pops up** as soon as the web page is loaded.

 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.

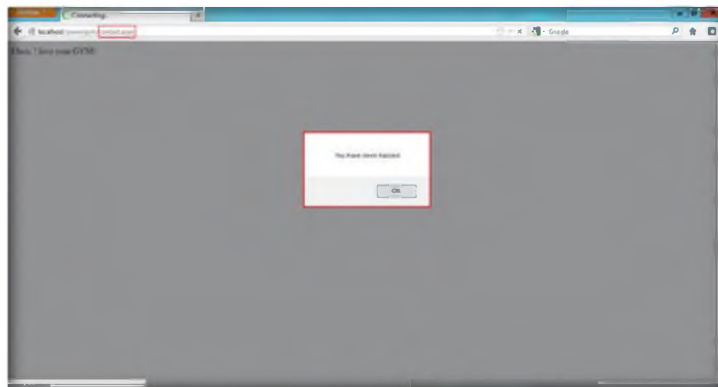


FIGURE 1.13: Powergym Error page

Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target’s security posture and exposure.

Tool/Utility	Information Collected/Objectives Achieved
Powergym Website	<ul style="list-style-type: none"> ▪ Parameter tampering results ▪ Cross-site script attack on website vulnerabilities

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

Questions

1. Analyze how all the malicious scripts are executed in a vulnerable web application.
2. Analyze if encryption protects users from cross-site scripting attacks.
3. Evaluate and list what countermeasures you need to take to defend from cross-site scripting attack.

Internet Connection Required	
<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> No
Platform Supported	
<input checked="" type="checkbox"/> Classroom	<input checked="" type="checkbox"/> iLabs



Website Vulnerability Scanning Using Acunetix WVS

Acunetix web vulnerability scanner (WVS) broadens the scope of vulnerability scanning by introducing highly advanced heuristic and rigorous 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

With the emergence of Web 2.0, increased information sharing through social networking and increasing business adoption of the Web as a means of doing business and delivering service, websites are often attacked directly. Hackers either seek to compromise the corporate network or the end-users accessing the website by subjecting them to 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, etc. 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 right in to the heart of the application and corporate data. Tailor-made web applications are often insufficiently tested, have undiscovered vulnerabilities and are therefore easy prey for hackers.

As an expert **Penetration Tester**, find out if your website is secure before hackers download sensitive data, commit a crime using your website as a launch pad, and endanger your business. You may use **Acunetix Web Vulnerability Scanner (WVS)** that checks the website, analyzes the web applications and finds perilous SQL injection, Cross site scripting and other vulnerabilities that expose 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.

Lab Environment

To perform the lab, you need:

- Acunetix Web vulnerability scanner is located at **D:\CEH-Tools\CEHv8 Module 13 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
- A web browser with an Internet connection
- Microsoft SQL Server / Microsoft Access

Lab Duration

Time: 20 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

1. Follow the wizard-driven installation steps to install **Acunetix Web Vulnerability Scanner**.
2. To launch **Acunetix Web Vulnerability Scanner** move your mouse cursor to lower left corner of your desktop and click **Start**

 **Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv8 Module 13 Hacking Web Applications**

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


 **NOTE: DO NOT SCAN A WEBSITE WITHOUT PROPER AUTHORISATION!**

TASK 1

Scan Website for Vulnerability

Module 13 – Hacking Web Applications

6. Check the type of Scan you want to perform, input the website URL, and click on **Next >** to continue
7. You can type <http://localhost/powergym> or <http://localhost/realhome>
8. In this lab we are scanning for vulnerabilities in for this webpage <http://localhost/powergym>

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

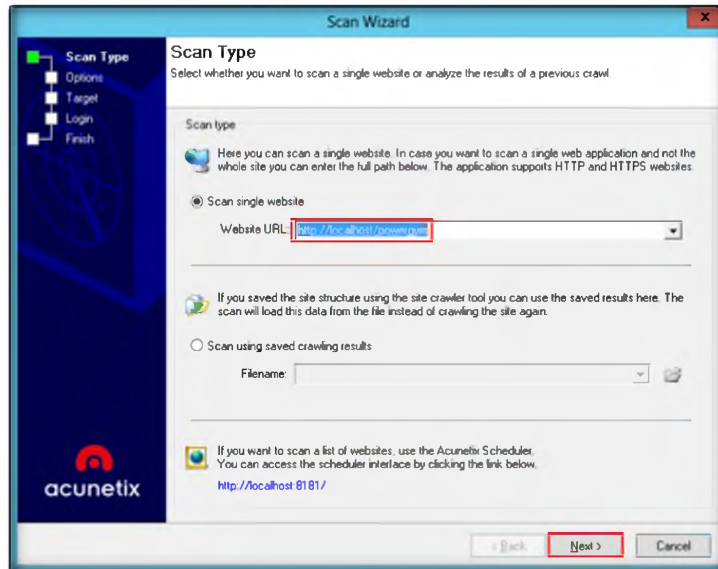


FIGURE 2.4: Acunetix WVS Scan Wizard Window

9. In **Options** live the settings to default click **Next**

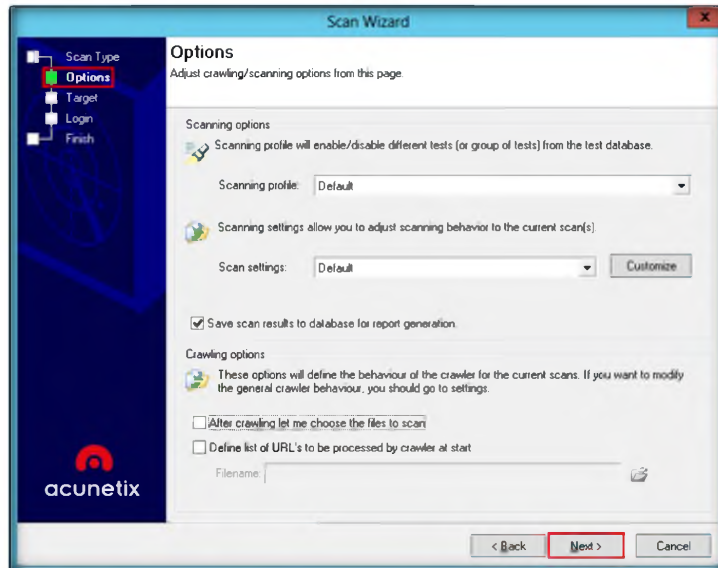




FIGURE 2.5: Acunetix WVS Options Wizard

10. Confirm targets and technologies detected by clicking on **Next**

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

Module 13 – Hacking Web Applications

 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.

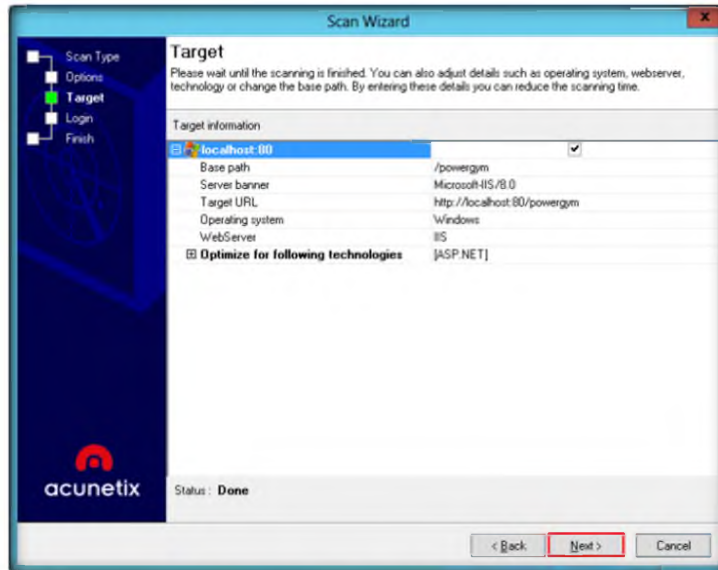



FIGURE 2.6: Acunetix WVS Scan Wizard Target

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

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

11. In **Login** wizard live the default settings and click **Next**

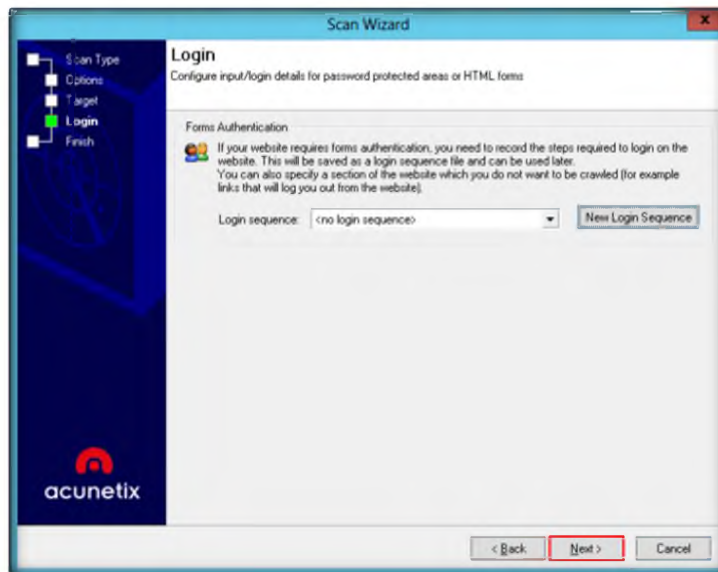




FIGURE 2.7: Acunetix WVS Scan Wizard Login Option

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

12. Click on **Finish** button to check with the vulnerabilities of website

Module 13 – Hacking Web Applications

 In Scan Options, Quick mode, the crawler fetches only a very limited number of variations of each parameter, because they are not considered to be actions parameters.

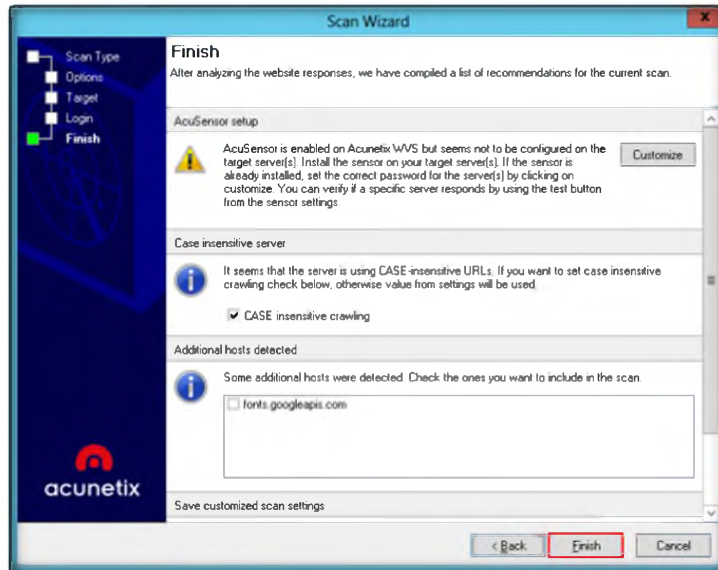



FIGURE 2.8: Acunetix WVS Scan Wizard Finish

- Click on **OK** in Limited XSS Scanning Mode warning

 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.

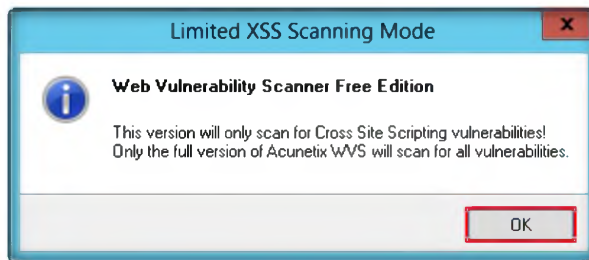



FIGURE 2.9: Acunetix WVS Scan Wizard -Warning

- Acunetix Web Vulnerability Scanner **starts** scanning the input website. During the scan, **security alerts** that are discovered on the website are listed in real time under the Alerts node in the **Scan Results** window. A node Site Structure 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.

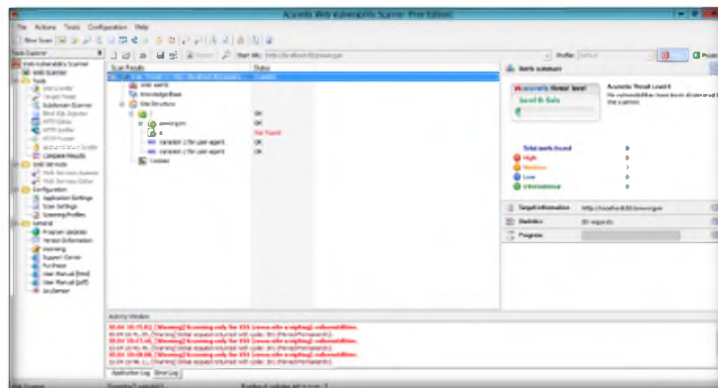



FIGURE 2.10: Acunetix WVS Main Window after Scan

Module 13 – Hacking Web Applications

15. The Web Alerts node displays all vulnerabilities found on the target website.

 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.

16. 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

17. The number of vulnerabilities detected is displayed in brackets () next to the alert categories.

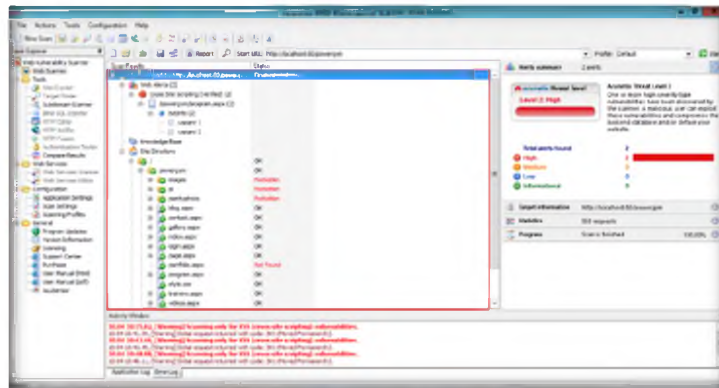


FIGURE 2.11: Acunetix WVS Result

TASK 2


Saving Scan Result

18. When a scan is complete, you can **save the scan results** to an external file for analysis and comparison at a later stage.

19. To **save** the scan results, click **File** → **Save Scan Results**. Select a desired location and save the scan results.

20. **Statistical Reports** allow you to gather vulnerability information from the results database and present periodical vulnerability statistics.

21. This report allows developers and management to track security changes and to compile trend analysis reports.

 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.

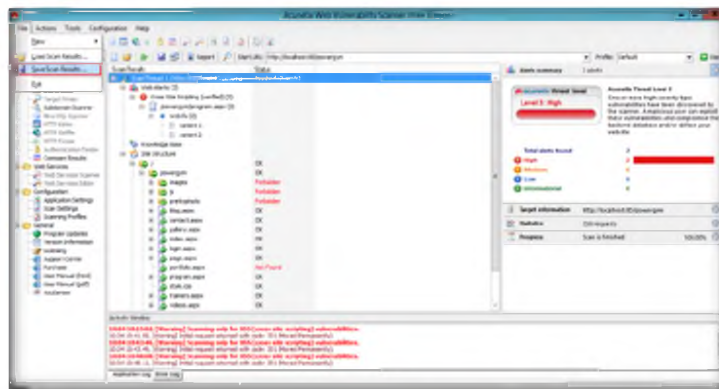


FIGURE 2.12: Acunetix WVS Saving Result

Note: In this lab we have used trial version so we could not able the save the results. To save the result it Acunetix WVS should be licensed version

TASK 3
Generating Report

22. To generate a report, click on the  report button on the toolbar at the top.

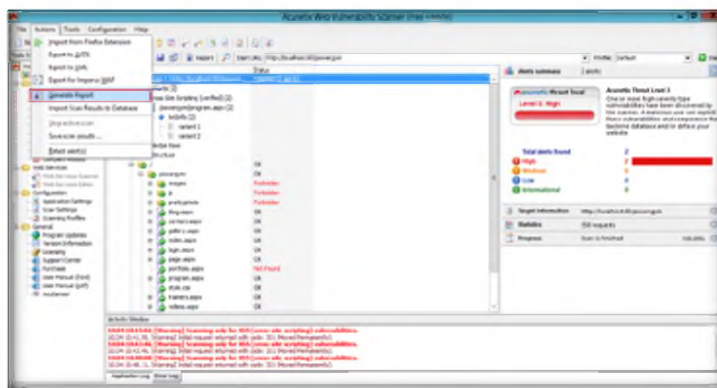



FIGURE 2.13: Acunetix WVS Generate Report option

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.

23. This action starts the **Acunetix WVS Reporter**.
24. The Report Viewer is a standalone application that allows you to **view, save, export, and print generated reports**. The reports can be exported to PDF, HTML, Text, Word Document, or BMP.
25. To generate a report, follow the procedure below. Select the type of report you want to generate and click on **Report Wizard** to launch a wizard to assist you.
26. 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 you would like to report. Click **Next** to proceed to the next step.
27. 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 to generate a report.
28. Select what properties and details the report should include. Click **Generate** to finalize the wizard and generate the report.
29. 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

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

 **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.

- Compliance Standard – PCI DSS, OWASP, WASC

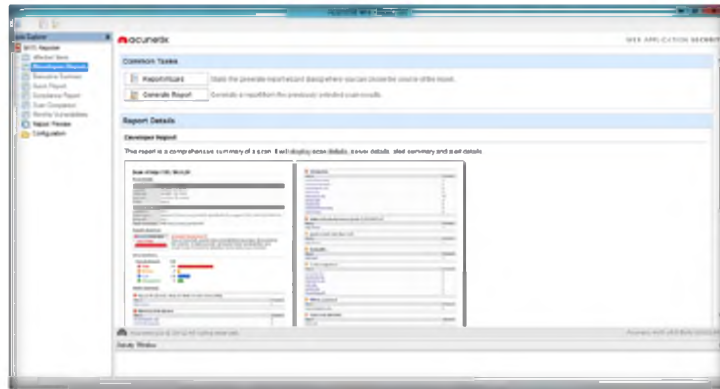


FIGURE 2.14: Acunetix WVS Generate Report window

Note: This is sample report, as trial version doesn't support to generate a report of scanned website

Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target's security posture and exposure.

Tool/Utility	Information Collected/Objectives Achieved
Acunetix Web Vulnerability Scanner	Cross-site scripting vulnerabilities verified

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

Questions

1. Analyze how you can schedule an unattended scan.
2. Evaluate how a web vulnerability scan is performed from an external source. Will it use up all your bandwidth?
3. Determine how Acunetix WVS crawls through password-protected areas.

Internet Connection Required	
<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
Platform Supported	
<input checked="" type="checkbox"/> Classroom	<input type="checkbox"/> iLabs