SQL Injection

Module 15

SQL Injection

SQL injection is a technique often used to attack a website and it is also the most common website vulnerability on the Internet.

ICON KEY

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

Lab Scenario

The SQL Injection attack is performed by including portions of SQL statements in a web form entry field in an attempt to get the website to pass a newly formed rogue SQL command to the database (e.g., dump the database contents to the attacker). SQL injection is a code injection technique that exploits a security vulnerability in a website's software. This vulnerability happens when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL commands are thus injected from the web form into the database of an application (like queries) to change the database content or dump the database information like credit card or passwords to the attacker. SQL injection is mostly known as an attack vector for websites, but can be used to attack any type of SQL database.

As an Expert Ethical Hacker, you must use diverse solutions, prepare statements with bind variables and whitelisting input validation and escaping. Input validation can be used to detect unauthorized input before it is passed to the SQL query.

Lab Objectives

The objective of this lab is to provide expert knowledge on SQL Injection attacks and other responsibilities that include:

- Understanding when and how web application connects to a database server in order to access data
- Extracting basic SQL Injection flaws and vulnerabilities
- Testing web applications for Blind SQL Injection vulnerabilities
- Scanning web servers and analyzing the reports
- Securing information in web applications and web servers

Tools demonstrated in this lab are available in Z:\CEH-Tools\CEHv10 Module 15 SQL Injection

Lab Environment

To complete this lab, you will need:

- A computer running Windows Server 2016
- Windows Server 2012 running on virtual machine
- Windows 10 running on a virtual machine
- Window 8 running on a virtual machine
- A web browser with an Internet connection
- Administrative privileges to configure settings and run the tools

Lab Duration

Time: 50 Minutes

Overview of SQL Injection

SQL Injection is a technique used to take advantage of non-validated input vulnerabilities to pass SQL commands through a web application for execution by a backend database.

TASK 1

Overview

Lab Tasks

Recommended labs to assist you in SQL Injection are:

- SQL Injection Attacks on **MSSQL Database**
- Performing SQL Injection Attack against MSSQL to Extract Databases and WebShell using **SQLMAP**
- Testing for SQL Injection using **IBM Security AppScan** Tool
- Scanning Web Applications using **N-Stalker** Tool

Lab Analysis

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

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

Lab 1

SQL Injection Attacks on MSSQL Database

SQL Injection is a basic attack used either to gain unauthorized access to a database or to retrieve information directly from it.

ICON KEY

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

Lab Scenario

Today, SQL Injection is one of the most common and perilous attacks that website's software experience. This attack is performed on SQL databases that have weak codes and this vulnerability can be used by an attacker to execute database queries to collect sensitive information, modify the database entries or attach a malicious code resulting in total compromise of the most sensitive data.

As an Expert Penetration Tester and Security Administrator, you need to test web applications running on the MS SQL Server database for vulnerabilities and flaws.

Lab Objectives

The objective of this lab is to provide students with expert knowledge on SQL Injection attacks and to analyze web applications for vulnerabilities.

In this lab, you will learn how to:

- Log on without valid credentials
- Test for SQL Injection
- Create your own user account
- Create your own database
- Directory listing
- Enforce Denial-of-Service attacks

Tools demonstrated in this lab are available in Z:\CEH-Tools\CEHv10 Module 15 SQL Injection

Lab Environment

To complete this lab, you will need:

- A computer running Window Server 2016 (Victim Machine)
- A computer running Window Server 2012 (Attacker Machine)
- The MS SQL Server must be running under local system privileges
- A web browser with an Internet connection

Lab Duration

Time: 15 Minutes

Overview of SQL Injection Attacks

SQL Injection is a basic attack used either to gain unauthorized access to a database or to retrieve information directly from the database. It is a flaw in web applications and not a database or web-server issue. Most programmers are still not aware of this threat.

Lab Tasks

TASK 1

Logon without Valid Credential

Blind SQL Injection is used when a web application is vulnerable to an SQL injection but the results of the injection are not visible to the attacker.

Blind SQL Injection is identical to normal SQL Injection, except that, when an attacker attempts to exploit an application, rather than seeing a useful error message, a generic custom page displays.

In this lab, the machine hosting the website is the victim machine (i.e., **Windows Server 2016**); and the machine used to perform SQL Injection attack is **Windows Server 2012** machine.

Try logging on using code **' or 1=1 --** as login name.

1. Before starting this lab make sure that you have logged into **Windows Server 2016** and **Windows Server 2012**.
2. In Windows Server 2012 machine Launch a web browser, type **http://www.goodshopping.com** in the address bar, and press **Enter**. In this lab we are using chrome web browser. If you are using any other browser then screenshots will vary in your lab environment.
3. The goodshopping home page appears, as shown in the screenshot:

4. Assume that you are new to this site and have never **registered** with it. Now click **LOGIN**.

When the attacker enters blah' or 1=1, then the SQL query look like this:

SELECT Count(*) FROM Users WHERE UserName='blah' Or 1=1 --' AND Password=''.

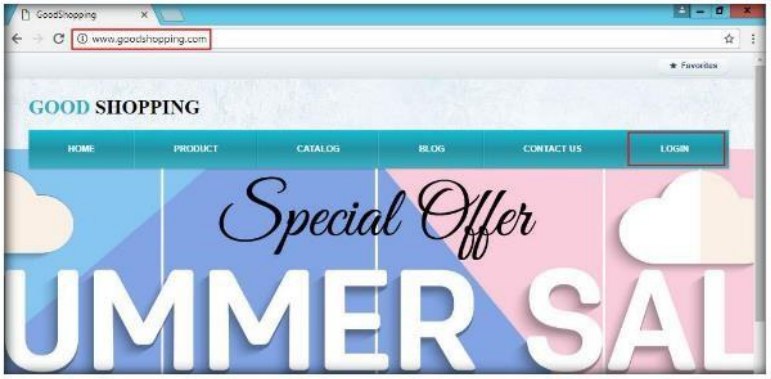


FIGURE 1.1: GOOD SHOPPING login page

5. Type the query **blah' or 1=1 --** in the **Username** field (as your login name), and leave the password field **empty**.
6. Click **Log in**.

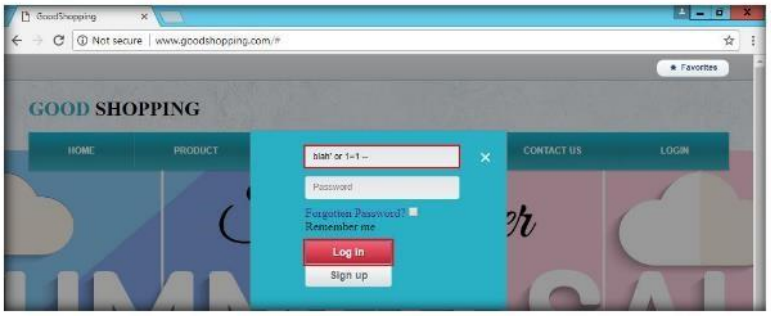


FIGURE 1.2: Performing Blind SQL

7. You are **logged into** the website with a **fake login**. Though your credentials are **not valid**. Now you can browse all the site's pages as a registered member.
8. After browsing the site, click **Logout**.

A user enters a user name and password that matches a record in the Users table.

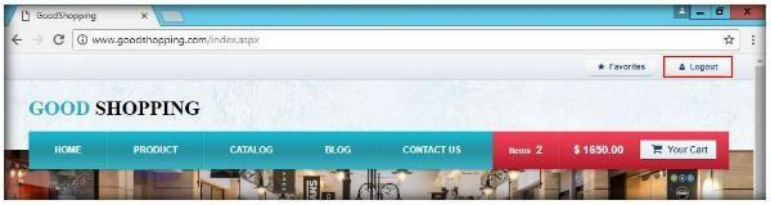


FIGURE 1.3: Website login successful

9. You have successfully logged out of the **vulnerable site**, and close the web browser.

Different databases require different SQL syntax. Identify the database engine used by the server.

10. Before performing the next task i.e., Creating a User Account with the SQL Injection query, first let us confirm with the Login **database** of GoodShopping.
11. Switch to Windows Server 2016 machine and navigate to **Start → Microsoft SQL Server Tools 17** and click **Microsoft SQL Server Management Studio 17**.
12. Microsoft SQL Server Management Studio window appears with Connect to Server pop-up, choose **Windows Authentication** in the Authentication field and click **Connect**.

Try to insert a string value where a number is expected in the input field.

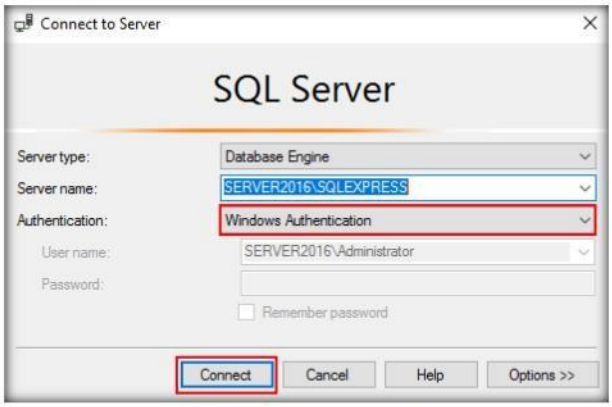
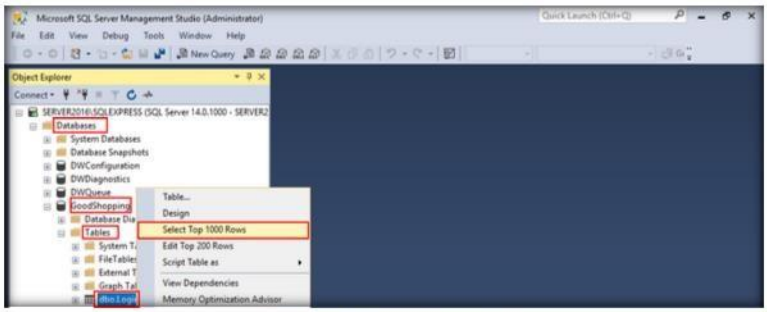


FIGURE 1.4: Connect to SQL server

13. **Microsoft SQL Server Management Studio** window appears as shown in the screenshot. In the left pane of **Object Explorer** expand **Databases → GoodShopping → Tables**. In **Tables** right-click **dbo.Login** and click **Select Top 1000 Rows** from the context menu to view the available credentials.



A dynamically generated SQL query is used to retrieve the number of matching rows.

FIGURE 1.5: Website login successful

14. As you can see in the database we have only one entry i.e., **smith** and **smith123**.

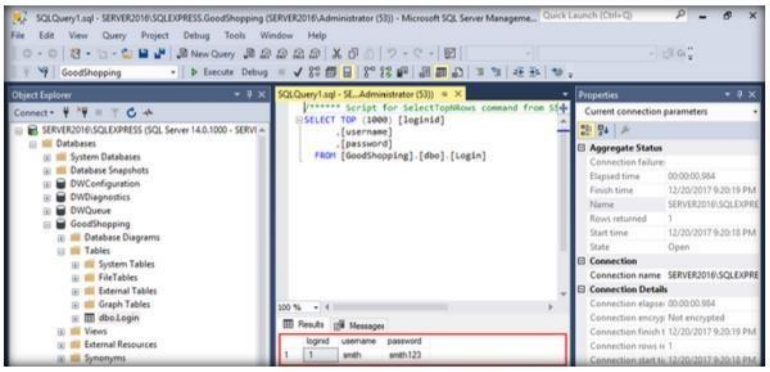


FIGURE 1.6: SQL database entries

TASK 2

Create Your Own User Account

15. Switch back to Windows Server 2012 machine, and launch a browser and type **http://www.goodshopping.com** in the address bar of the browser and press **Enter**. The GOOD SHOPPING home page appears, as shown in the screenshot:
16. Click **LOGIN**, and type the query **blah';insert into login values ('john','apple123'); --** in the **Username** field (as your login name), and leave the password field **empty** as shown in the screenshot.
17. Click **Log in**.

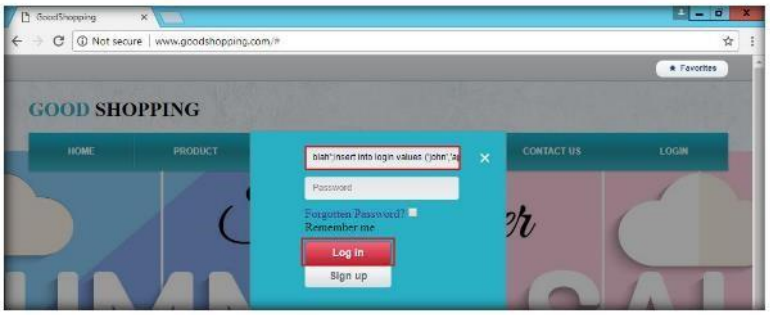


FIGURE 1.7: Creating a user account

To detect SQL Injection, check if the web application connects to a database server in order to access some data.

18. If no error message is displayed, it means that you have successfully created your login using an SQL injection query.

19. After executing the query, to **verify** whether your login has been created successfully, click **LOGIN** tab, enter **john** in the **Username** field and **apple123** in the **Password** field, and click **Log in**.

Error messages are essential for extracting information from the database. Depending on the type of errors found, you can vary the attack techniques.

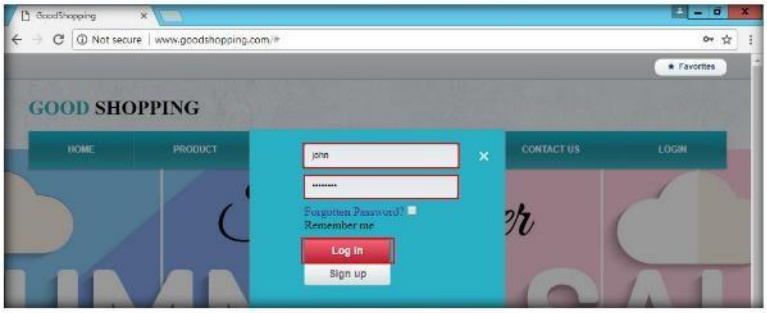


FIGURE 1.8: Logging in to the website

20. You will **login** successfully with the created login. Now you can access all the features of the website.
21. Click **Logout** after browsing the required pages, and close the browser window

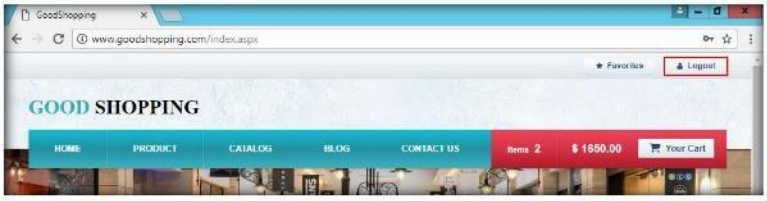


FIGURE 1.9: Log in successful

22. Switch back to the **Windows Server 2016** virtual machine.
23. Right-click **dbo.Login,** and click **Select Top 1000 Rows** from the context menu as shown in the screenshot.

Understanding the underlying SQL query allows the attacker to craft correct SQL Injection statements.

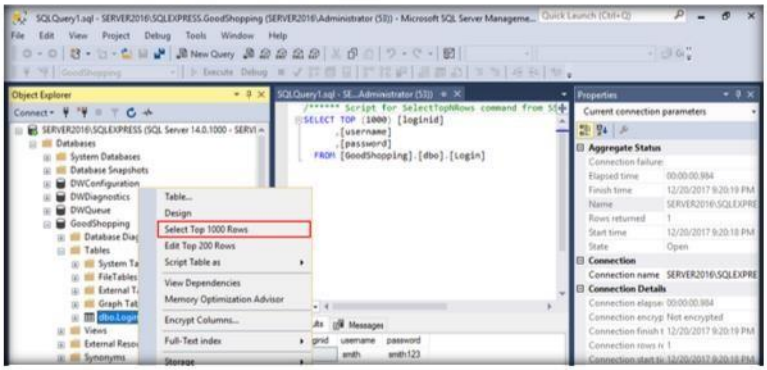


FIGURE 1.10: Selecting Top 1000 Rows

24. Observe that the **username** and **password** have been successfully added to the goodshopping database.
25. Note down the available **databases** and then close the SQL Server Management Studio window.

Mostly the error messages show you what DB engine you are working on with ODBC errors. It displays database type as part of the driver information.

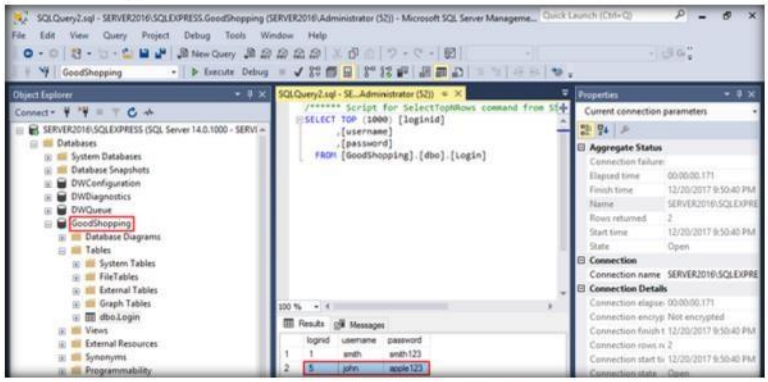


FIGURE 1.11: Table containing the created usernames and passwords

26. Switch back to the **Windows Server 2012** virtual machine.

TASK 3

Create Your Own Database

27. Launch the browser, type **http://www.goodshopping.com** in the address bar, and press **Enter**.
28. The **Home** Page of GOOD SHOPPING appears.
29. Click **LOGIN**, type **blah';create database mydatabase; –** in the **Username** field, leave the **Password** field empty, and click **Login**.
30. In the above query, **mydatabase** is the name of the database.

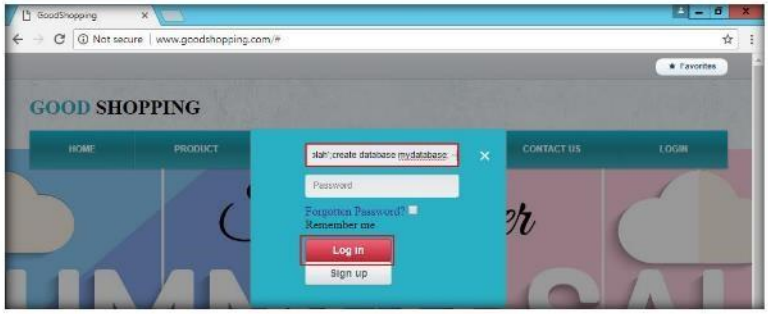


FIGURE 1.12: Creating a database

Try to replicate an error-free navigation, which could be as simple as ' and '1' = '1 Or ' and '1' = '2.

31. If no error message (or any message) displays on the web page, it means that the site is vulnerable to SQL injection; a database with the name **mydatabase** has been created at the database server. Close the browser.
32. Switch back to Windows Server 2016 victim machine, and launch the SQL Server Management Studio and log in.

Time delays are a type of blind SQL Injection that causes the SQL engine to execute a long-running query or a time delay statement, depending on the logic injected.

33. The **Microsoft SQL Server Management Studio** main window appears, as shown in the screenshot:
34. Expand the **Databases** node. A new database has been created with the name **mydatabase**.

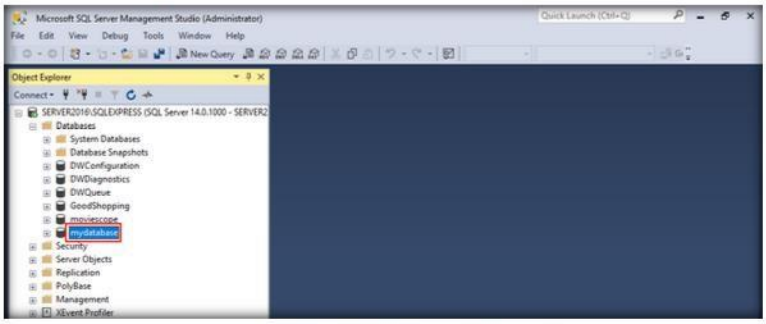


FIGURE 1.13: juggyboy database successfully created

35. Close the **Microsoft SQL Server Management Studio** window.
36. Switch back to **Windows Server 2012** virtual machine.

TASK 4

Denial-of-Service Attack

37. Launch the web browser, type **http://www.goodshopping.com** in the address bar, and press **Enter**.
38. The **Home** page of GOOD SHOPPING appears.
39. Click **LOGIN**, type **blah';exec master..xp_cmdshell 'ping www.certifiedhacker.com -l 65000 -t';** -- in the **Username** field, leave the **Password** field empty, and click **Log in**.

Use the bulk insert statement to read any file on the server, and use bcp to create arbitrary text files on the server.

FIGURE 1.14: Performing Denial of Service Attack

40. In the above query, you are performing a **ping** for the **www.certifiedhacker.com** website using an SQL Injection query: **-l** is the sent buffer size, and **-t** refers to pinging the specified host.

41. The SQL injection query starts pinging the host, and the login page shows a **Waiting for www.goodshopping.com...** message at the bottom of the window.

Once you determine the usernames, you can start gathering passwords:

Username: ' union select password,1,1,1 from users where username = 'admin'--

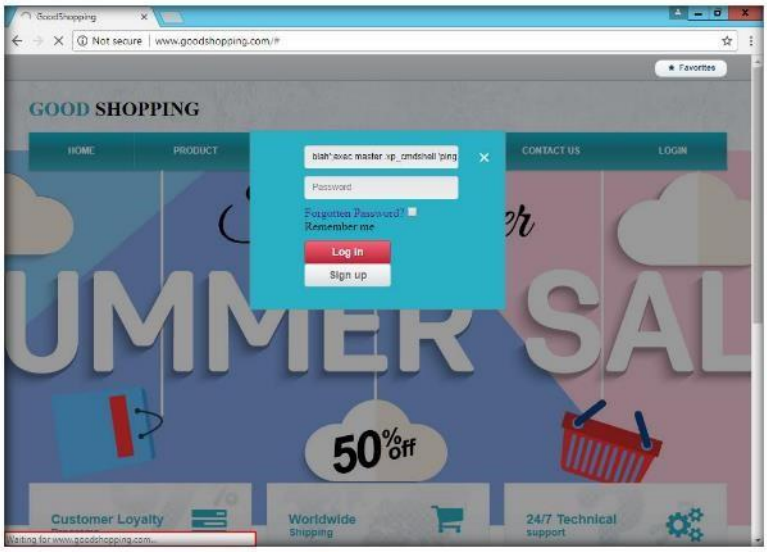


FIGURE 1.15: SQL injection query starts pinging the host

42. To see whether the query has successfully executed, switch back to **Windows Server 2016**.
43. Launch **Task Manager**.
44. In Task Manager, under the **Details** tab, you see a process called **PING.EXE** running in the background.

Using the sp_OACreate, sp_OAMethod and sp_OAGetProperty system stored procedures to create Ole Automation (ActiveX) applications that can do everything an ASP script can do.

45. This process is the **result** of the SQL Injection query that you entered in the login field of the web site.

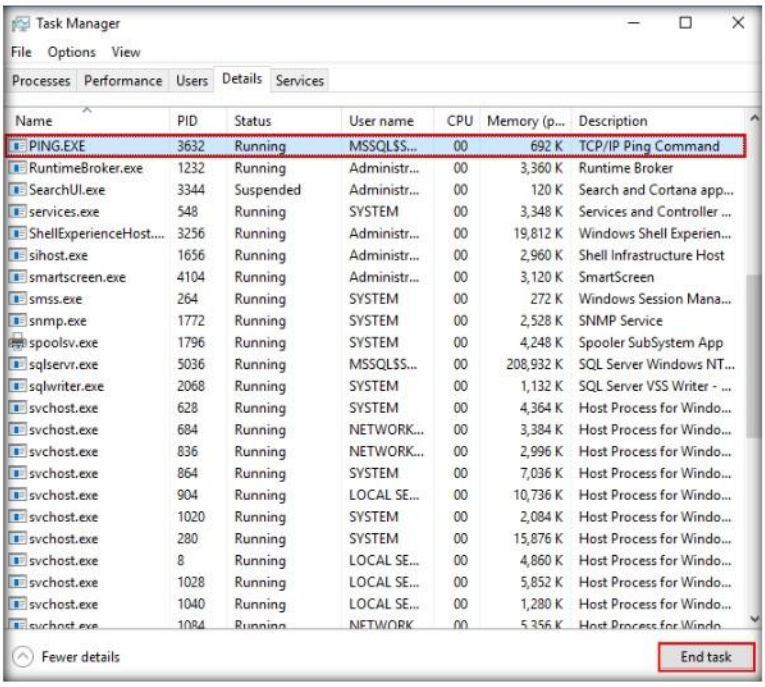


FIGURE 1.16: Task Manager displaying the ping process

The attacker then selects the string from the table, as before:

Username: ' union select ret,1,1,1 from foo--

Microsoft OLE DB Provider for ODBC Drivers error '80040e07.'

46. To manually kill this process, right-click **PING.EXE,** and click **End Process.** This stops/prevents the website from pinging the host.

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.

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

Lab 2

Performing SQL Injection Attack against MSSQL to Extract Databases and WebShell using SQLMAP

ICON KEY

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

SQLMAP is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers.

Lab Scenario

SQL injection is a technique used to take advantage of un-sanitized input vulnerabilities to pass SQL commands through a web application for execution by a backend database. SQL injection is a basic attack used to either gain unauthorized access to a database or to retrieve information directly from the database. It is a flaw in web applications and not a database or web server issue.

Lab Objectives

Tools demonstrated in this lab are available at Z:\CEH-Tools\CEHv10 Module 15 SQL Injection

The objective of this lab is to help students learn how to perform a SQL injection attack and extract databases

Lab Environment

To complete this lab, you will need:

- Windows Server 2016 (Victim Machine)
- Kali Linux machine (Attacker Machine)
- Run this lab on Kali Linux machine
- Make sure that Windows Server 2016 machine is running
- A web browser with Internet access
- Microsoft .NET Framework Version 4.0 or later

Lab Duration

Time: 10 Minutes

Overview of Testing Web Applications

Web applications are tested for implementing security and automating vulnerability assessments. Doing so prevents SQL injection attacks on web servers and web applications. Websites are tested for embedded malware and by employing multiple techniques.

Lab Tasks

TASK 1

Login to MovieScope

1. Logon to Kali Linux machine with Username: root and Password: toor
2. Before starting this lab assume that you are registered a user on the **http://www.moviescope.com** website. And you want to crack the passwords of the other users from the database of the moviescope.
3. Open a web browser and login into the **http://www.moviescope.com** as Username: **sam** and Password: **test@123**
4. Once you are logged into the website click **View Profile** tab, and make a note of the URL in the address bar of the browser.
5. **Right-click** anywhere on the webpage and click **Inspect Element (Q)** from the context menu as shown in the screenshot.

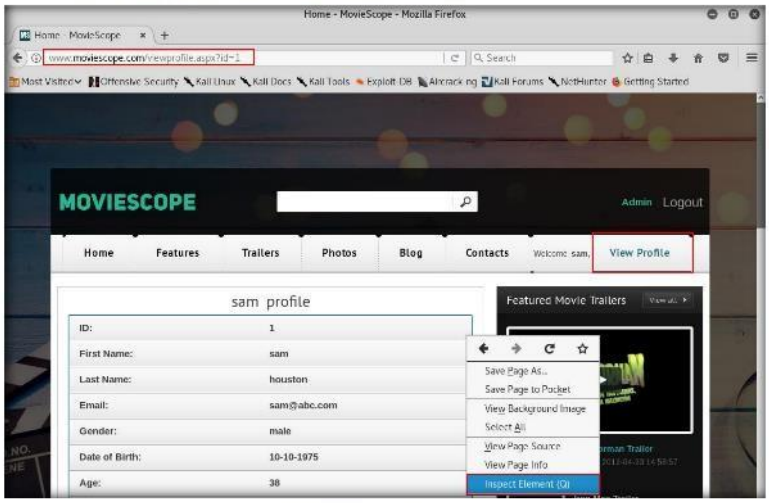


FIGURE 2.1: Inspect Element option

6. Developer Tools section appears as shown in the screenshot, click **Console** tab and type **document.cookie** in the lower left corner of the browser and press **Enter**.

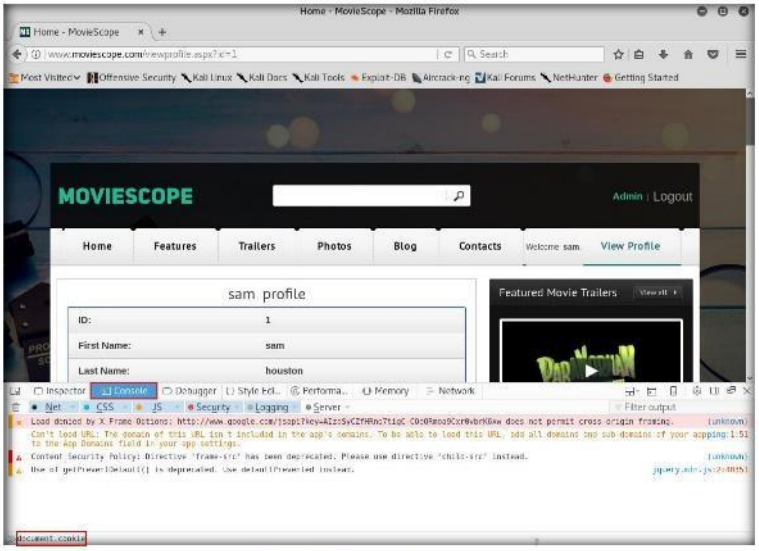


FIGURE 2.2: Requesting the cookie value

7. Select the **cookie value** and right-click and **Copy** the value as shown in the screenshot. Minimize the web browser.

Note: Cookie value may differ in your lab environment.

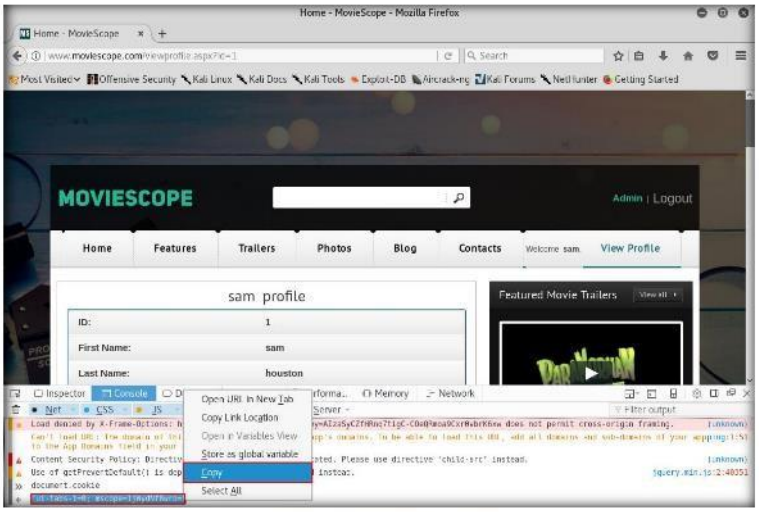


FIGURE 2.3: Copying the cookie value

8. Click **Terminal** icon from the taskbar to launch as shown in the screenshot.

FIGURE 2.4: Kali Linux – Desktop view

9. Type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" –cookie=<"cookie value which you have copied in step #7"> –dbs** and press **Enter**.
10. By issuing the above query, sqlmap enforces various injection techniques on the name parameter of the URL in an attempt to extract the database information of **moviescope** website.

FIGURE 2.5: Getting databases in the SQL server

11. SQLmap retrieves the databases present in **MS SQL Server**. It also displays information about the web server operating system, web application technology and the back-end DBMS as shown in the screenshot.

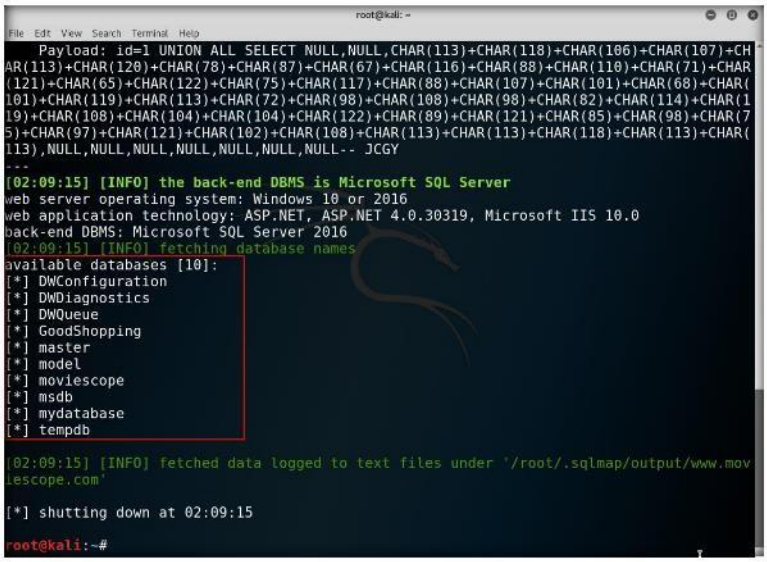


FIGURE 2.6: Databases present in the SQL server

12. Now, you need to choose a database and use sqlmap to retrieve the tables in the database. In this lab, we are going to determine the tables associated with moviescope database. Now type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" –cookie=<"cookie value which you have copied in step #7"> -D moviescope --tables** and press **Enter**. By issuing the above query, sqlmap starts scanning the **moviescope** database in search of tables located in the database.

FIGURE 2.7: SQLmap command to retrieve the tables in moviescope database

13. sqlmap retrieves the table contents of the **moviescope database** and displays them as shown in the screenshot.

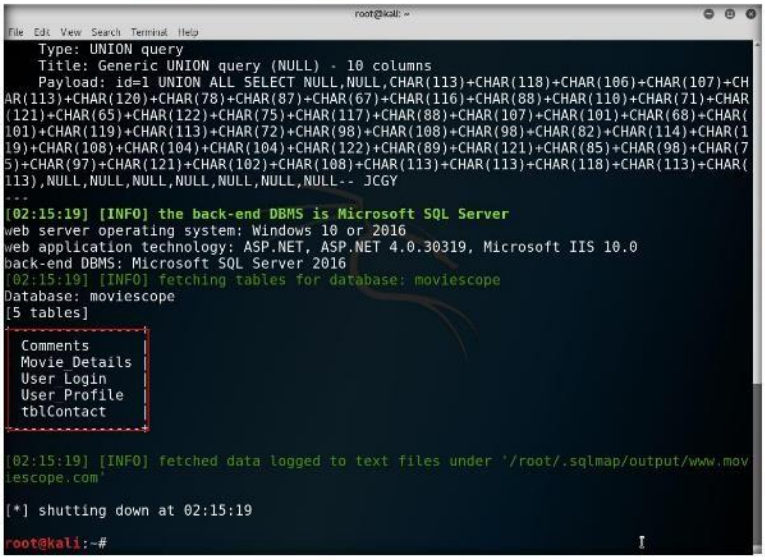


FIGURE 2.8: Tables present in the moviescope database

14. Now, you need to retrieve the columns associated with the tables. In this lab, you will use sqlmap to retrieve the columns of the table named "**User_Login**". For extracting columns information, you need to issue the following sqlmap query: **sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" –cookie=<"cookie value which you have copied in step #7"> -D moviescope -T User_Login –columns**. By issuing the above query, sqlmap starts scanning the User_Login table inside moviescope database in search of columns.

FIGURE 2.9: Command to retrieve the user login info

15. sqlmap retrieved the available columns in the above mentioned table i.e., **User_Login** as shown in the screenshot.

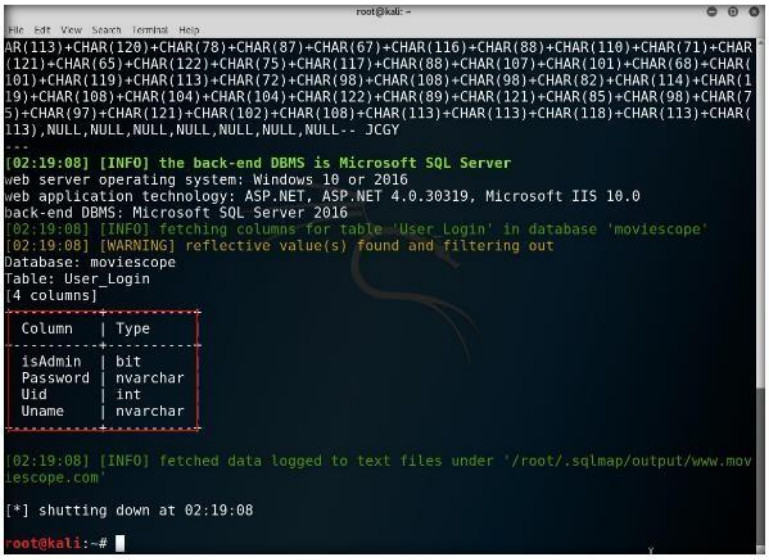


FIGURE 2.10: User login table retrieved by sqlmap

16. Now type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" –cookie=<"cookie value which you have copied in step #7"> -D moviescope -T User_Login –dump** and press **Enter** to dump the all User_Login table content

FIGURE 2.11: Dumping user profiles of moviescope website

17. Now the sqlmap has retrived the complete database of the moviescope which contains the **Username** and **Passwords** of the users as shown in the screenshot.

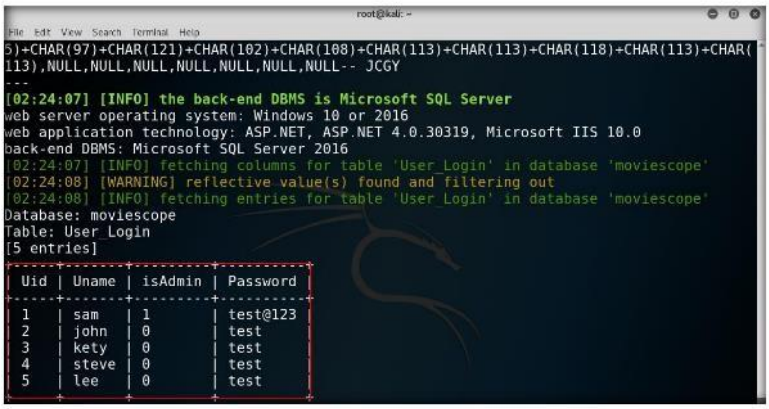


FIGURE 2.12: Retriving the database of moviescope

18. To verify the login details are valid, you can login with the extracted login details of any of the user. Before that close the Developer Tools console and logout from the previous session in the browser and then login. In this lab we are logging in with the user **steve** and password is **test**.

19. As you see in the below screenshot we have successfully logged into the moviescope website with steve's account.

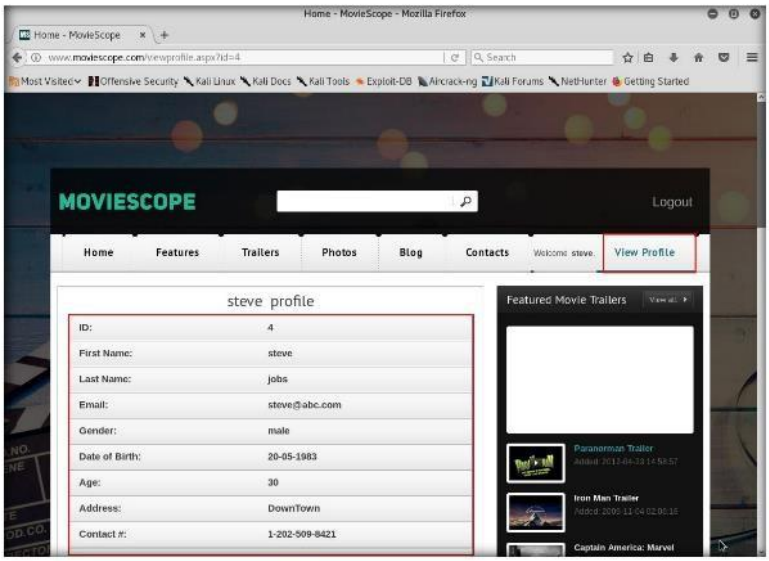


FIGURE 2.13: Steve's account on moviescope

20. Now type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" –cookie=<"cookie value which you have copied in step #7"> --os-shell** and press **Enter**

FIGURE 2.14: SQLmap command with moviescope as target

21. sqlmap tries to optimize value(s) for DBMS delay responses message appears type **Y** and press **Enter** to continue.

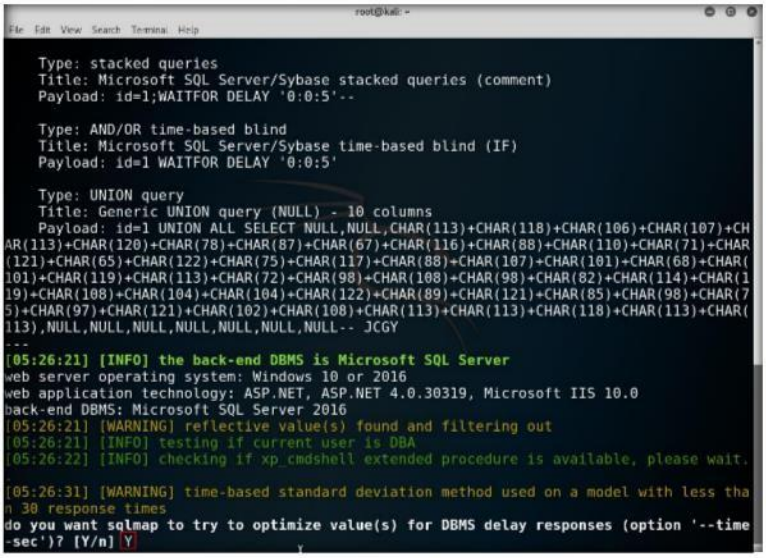


FIGURE 2.15: Optimization of DBMS delay responses option

22. Once sqlmap aquires the permission to optimize the machine, it will gives you with the os-shell. Type **hostname** and press **Enter** to find the machine name where the site is running.

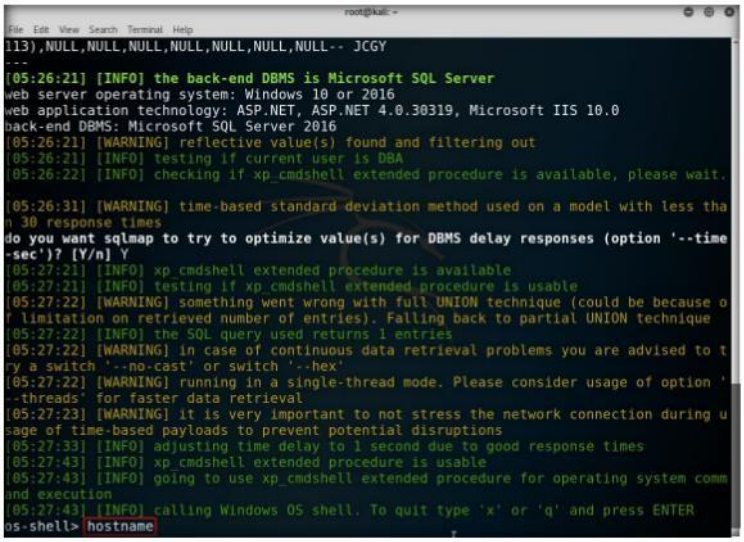


FIGURE 2.16: Hostname command in sqlmap

23. Do you want to retrieve the command standard output? message appears type **Y** and press **Enter**.

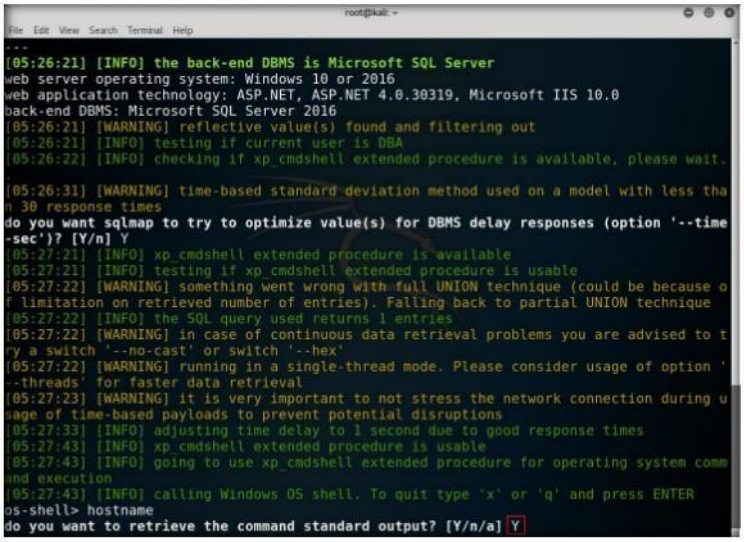


FIGURE 2.17: Retriving the standard output

24. Thus sqlmap will retrieves the hostaname as shown in the screenshot.

Note: If do you want to retrieve the command standard output? message appears type **Y** and press **Enter**.

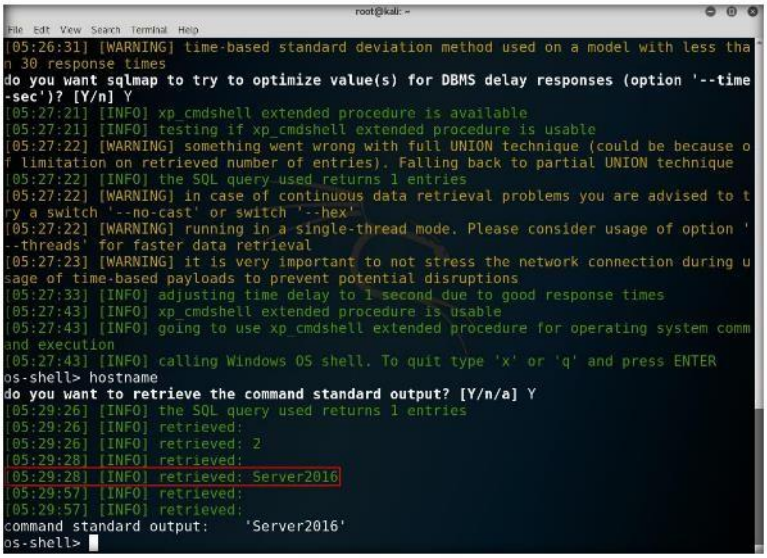


FIGURE 2.18: Retrieving hostnames

25. Type **ipconfig** and press **Enter** to know the IP configuration the machine.

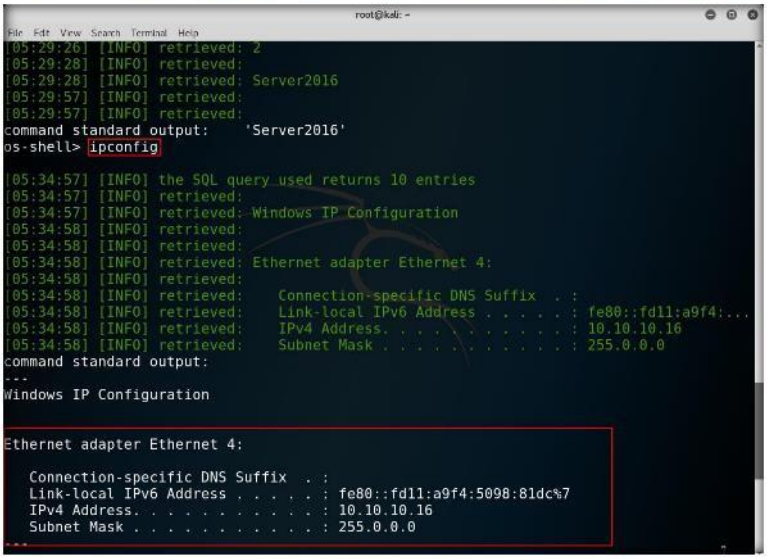


FIGURE 2.19: Running Ipconfig command

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

Testing for SQL Injection using IBM Security AppScan Tool

ICON KEY

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

The IBM Security AppScan is a web application security testing tool that automates vulnerability assessments, prevents SQL injection attacks on websites, and scans web sites for embedded malware.

Lab Scenario

By now, you must be familiar with the types of SQL injection attacks an attacker can perform and the impact caused by these attacks. Attackers can use the following types of SQL Injection attacks: Authentication Bypass, Information Disclosure, Compromised Data Integrity, Compromised Availability of Data and Remote Code Execution which allows them to spoof identity, damage existing data, execute system-level commands to cause a denial of service of the application, and so on.

In the previous lab, you have already learned to test SQL Injection Attacks on MS SQL Database for website vulnerabilities.

As an organization's Expert Security Professional and Penetration Tester, your job responsibility is to test the company's web applications and web services for vulnerabilities. You need to find various ways to extend security test and analyze web applications and employ multiple testing techniques.

Moving further, in this lab, you will learn to test for SQL Injection attacks using IBM Security AppScan.

Tools demonstrated in this lab are available Z:\CEH-Tools\CEHv10 Module 15 SQL Injection

Lab Objectives

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

In this lab, you will learn to:

- Perform web site scans for vulnerabilities
- Analyze scanned results
- Generate reports for scanned web applications

Lab Environment

You can download IBM AppScan from http://www-01.ibm.com.

To complete this lab, you will need:

- IBM Security AppScan located at **Z:\CEH-Tools\CEHv10 Module 15 SQL Injection\SQL Injection Detection Tools\IBM Security AppScan**
- A computer running Window Server 2016
- You can also download the latest version of Security AppScan from the link **http://www-01.ibm.com/software/awdtools/appscan/standard**
- A web browser with Internet access
- Microsoft .NET Framework Version 4.0 or later

Supported operating systems (both 32–bit and 64–bit editions):

- Windows 2003: Standard and Enterprise, SP1 and SP2
- Windows Server 2008: Standard and Enterprise, SP1 and SP2

Lab Duration

Time: 15 Minutes

Overview of Testing Web Applications

Web applications are tested for implementing security and automating vulnerability assessments. Doing so prevents SQL injection attacks on web servers and web applications. Websites are tested for embedded malware and to employ multiple testing techniques.

TASK 1

Install and Configure IBM Security AppScan

Lab Tasks

1. Navigate to **Z:\CEH-Tools\CEHv10 Module 15 SQL Injection\SQL Injection Detection Tools\IBM Security AppScan** and double-click **AppScan_Std_9.0.3.6_Eval_Win.exe**.
2. If an **Open File - Security Warning** pop-up appears, click **Run**.
3. Follow wizard-driven installation steps and **install** the IBM Security AppScan tool.
4. It takes around 5 minutes to complete the installation process.

Note: At the time of installation, a **Web Services Component Download** dialog-box appears asking you to download an additional component. Click **No** to avoid the download.

5. Launch the **IBM Security AppScan** application from the **Apps** list of **Windows Server 2016**.

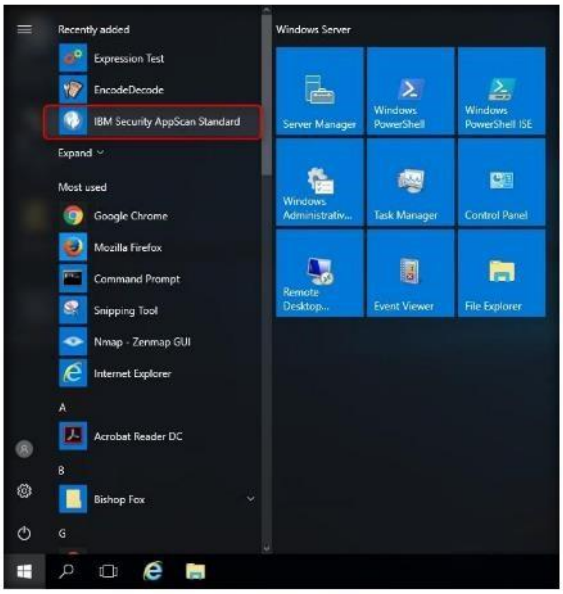


FIGURE 3.1: Launching the application from Apps list

You can configure Scan Expert to perform its analysis and apply some of its recommendations automatically, when you start the scan.

6. The main window of **IBM Security AppScan** appears, click on **Create New Scan...** to begin scanning.

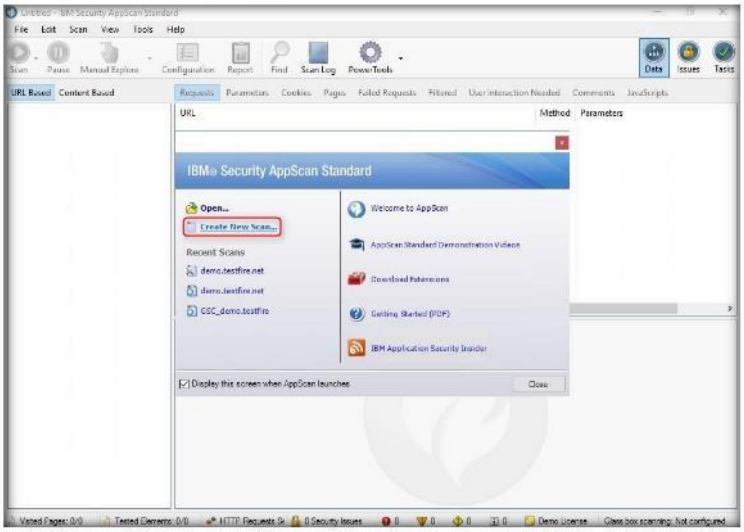


FIGURE 3.2: IBM Security AppScan main window

A personal firewall running on the same computer as Rational AppScan can block communication and result in inaccurate findings and reduced performance. For best results, do not run a personal firewall on the computer that runs Rational AppScan.

7. A **New Scan** pop-up appears; click on **demo.testfire.net** link.

Note: In evaluation version we cannot scan other websites.

AppScan can scan both web applications and web services.

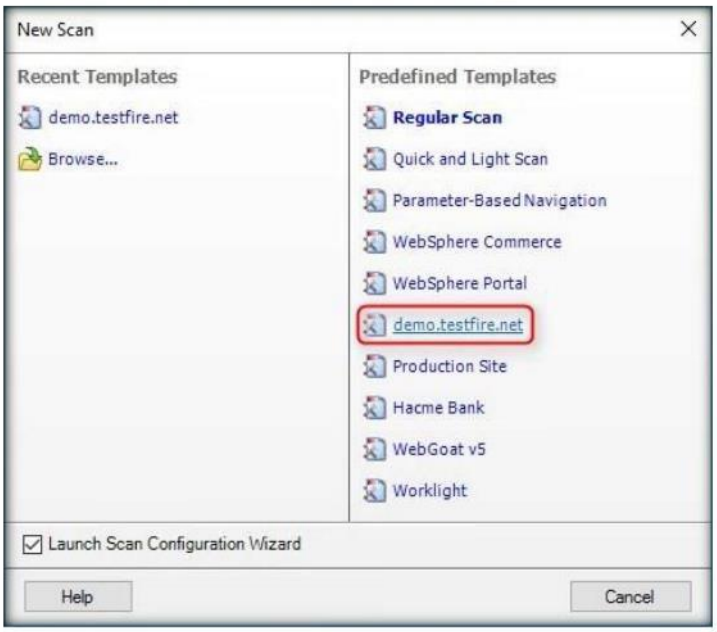


FIGURE 3.3: New Scan pop-up

Malware test uses data gathered during the explore stage of a regular scan, so you must have some explore results for it to function.

TASK 2

Perform Web Application Vulnerability Scan

8. The **Scan Configuration Wizard** appears; select **AppScan (automatically or manually)**, and click **Next**.

One of the options in the scan configuration wizard is for Scan Expert to run a short scan to evaluate the efficiency of the new configuration for your particular site.

FIGURE 3.4: IBM Security AppScan – Scan Configuration Wizard

9. Under **URL and Servers**, leave the default options and click **Next**.

There are some changes that Scan Expert can only apply with human intervention, so when you select the automatic option, some changes may not be applied.

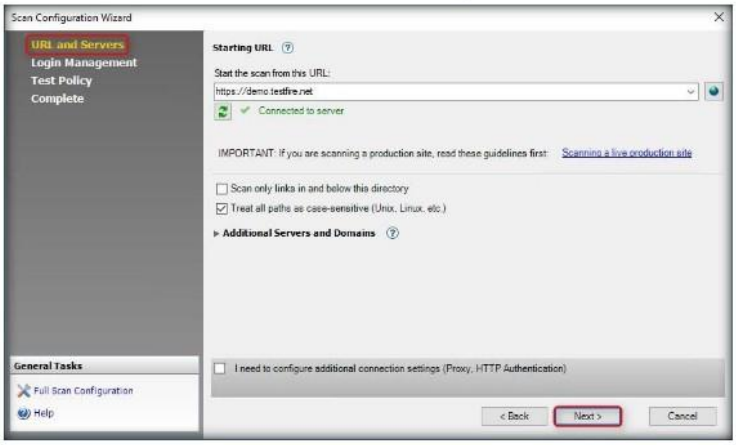


FIGURE 3.5: IBM Security AppScan – Scan Configuration Wizard

10. Under **Login Management**, select **Automatic** and enter the username **jsmith** and password **Demo1234** and click **Next**.

The total number of tests to be sent, or URLs to be visited, may increase during a scan, as new links are discovered.

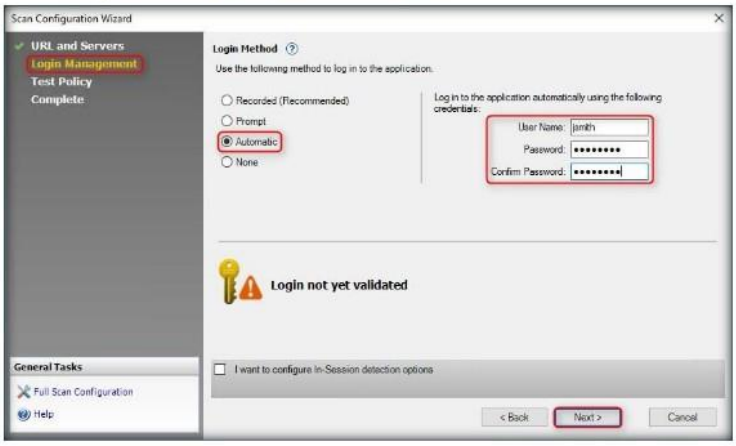


FIGURE 3.6: IBM Security AppScan - Scan Configuration wizard

11. Under **Test Policy**, leave the default options and click **Next**.

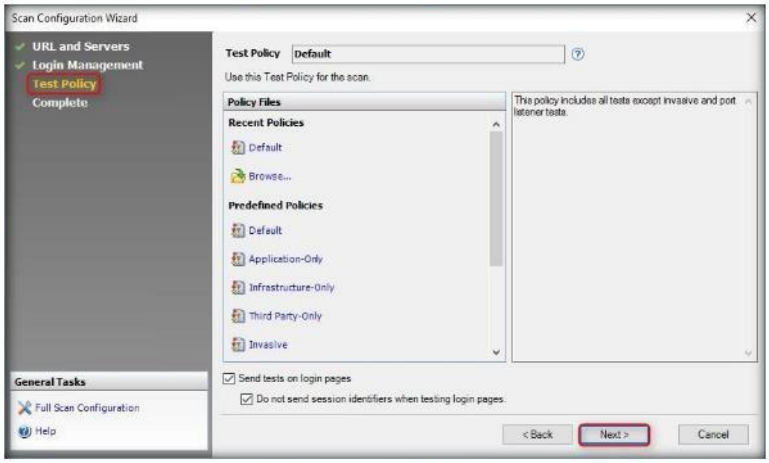


FIGURE 3.7: IBM Rational AppScan: Test Policy section

Security Issues view shows the actual issues discovered, from overview level down to individual requests/responses. This is the default view.

12. Under **Complete**, verify that **Start a full automatic scan** is selected, and click **Finish** to complete the **Scan Configuration**.

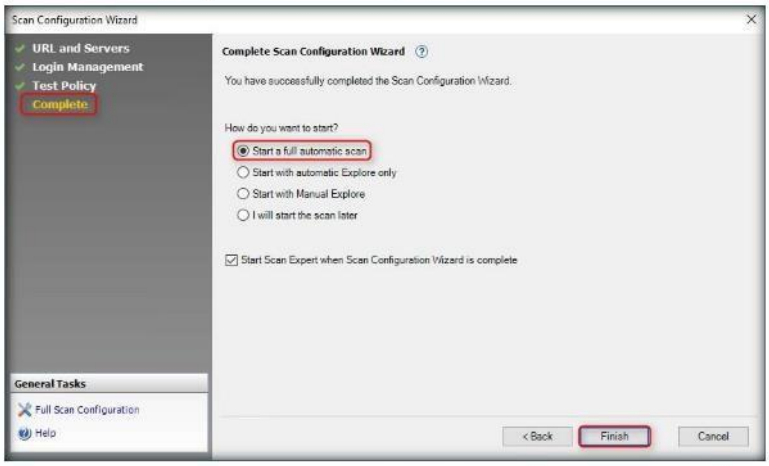


FIGURE 3.8: IBM Rational AppScan: Complete section

Results can display in three views: Security Issues, Remediation Tasks, and Application Data. The view is selected by clicking a button in the view selector. The data displayed in all three panes varies with the view selected.

13. An **Auto Save** dialog-box prompts you to save **automatically during scan**; click **Yes** to save the file and proceed.

Remediation Tasks view provides a To Do list of specific remediation tasks to fix the issues found by the scan.

FIGURE 3.9: Auto Save window

14. The **Save As** window appears; navigate to the location where you would save the scan, specify a name for it, and click **Save**.

The Result List displays the issues for whatever item is selected in the application tree. These can be for:

- Root level: All site issues display
- Page level: All issues for the page
- Parameter level: All issues for a particular request to a particular page

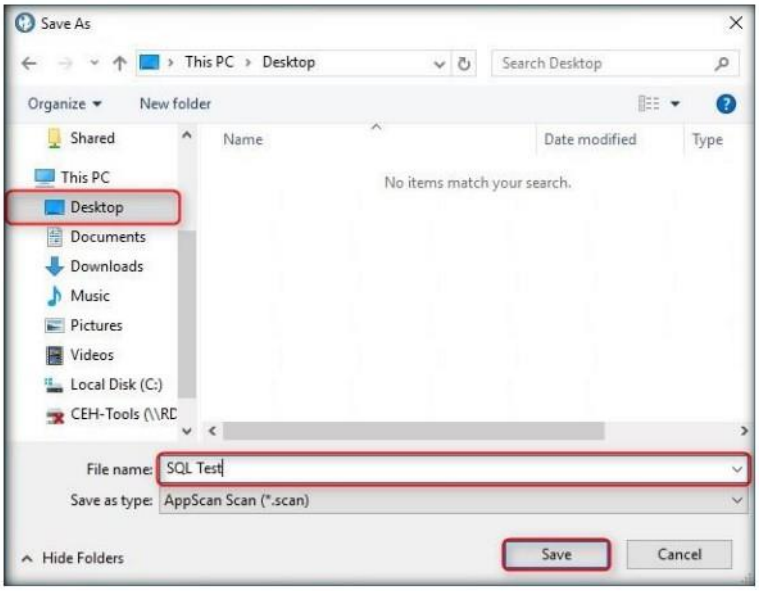


FIGURE 3.10: Save As window

15. The IBM Security AppScan starts **scanning** the provided URL.

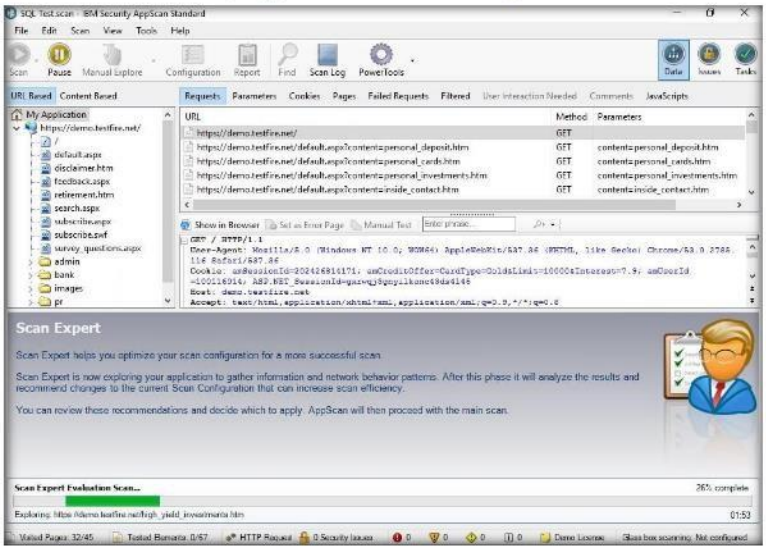


FIGURE 3.11: IBM Rational AppScan Scanning Web Application window

16. The **Scan Expert Recommendations** pane opens; click **Ignore All** in the lower right of the screen.

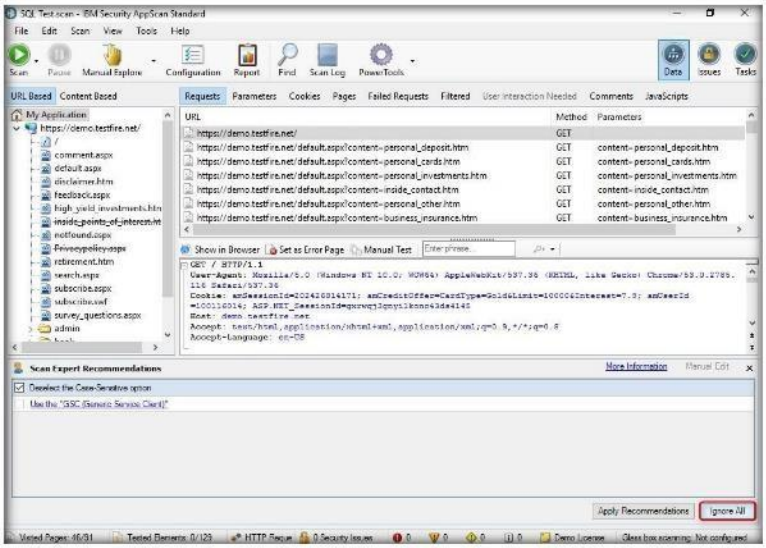


FIGURE 3.12: IBM Rational AppScan: Scan Expert Recommendations section

17. An **AppScan** pop-up appears; click **Yes**.

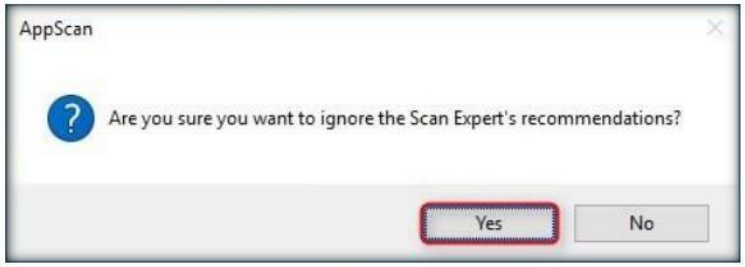


FIGURE 3.13: AppScan pop-up

18. AppScan begins to scan for website vulnerabilities.

You can export the complete scan results as an XML file or as a relational database. (The database option exports the results into a Firebird database structure. This is open source and follows ODBC and JDBC standards.).

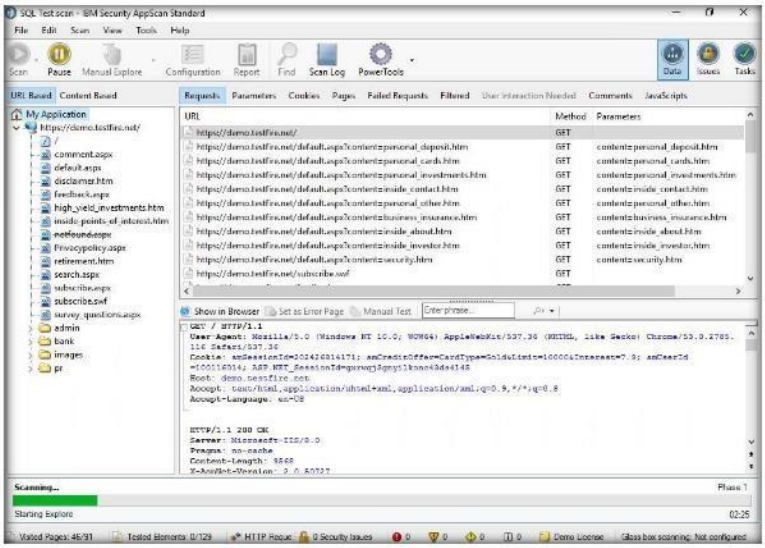


FIGURE 3.14: Vulnerability Scanning

Note: It will take a lot of time to scan the complete site.

19. After **the completion of scanning,** the application lists all the security issues and vulnerabilities it has found.
20. Results can be displayed in three views: **Data, Issues,** and **Tasks.**
21. To view the vulnerabilities and security issues found, click **Issues.**

Result Expert consists of various modules that are used to process scan results. The processed results are added to the Issue Information tab of the Detail pane, making the information displayed there more comprehensive and detailed, including screen shots where relevant.

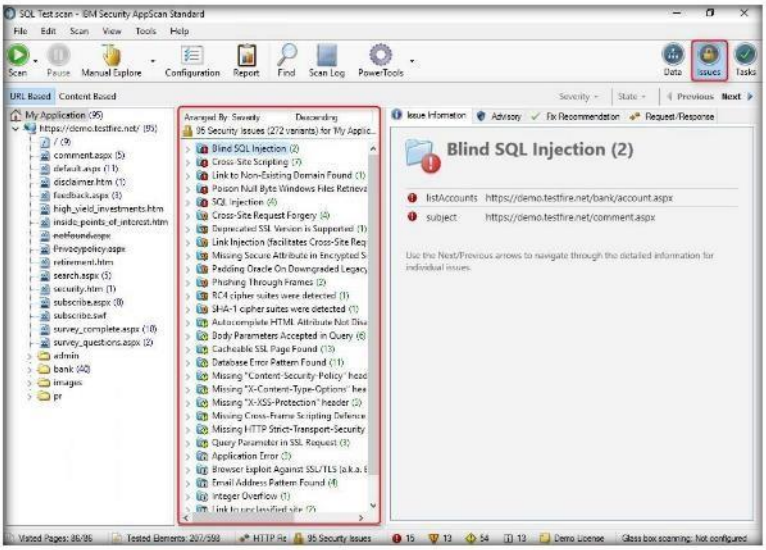


FIGURE 3.15: IBM Rational AppScan Scanning Web Application Result window

TASK 3

Analyze Result

22. To analyze the scan results, click on any of the results, such as **SQL Injection,** and expand the nodes to list all the links that are **vulnerable** to SQL Injection.
23. You can find explanation regarding the selected link in the right pane of the GUI, under **Issue Information.**

The severity level assigned to any issue can be changed manually by right-clicking on the node.

FIGURE 3.16: IBM Rational AppScan Scanning Web Application Result window

24. Click **Advisory** tab in the right pane of the window to see the **severity** of that particular link, as well as the description of the threat.

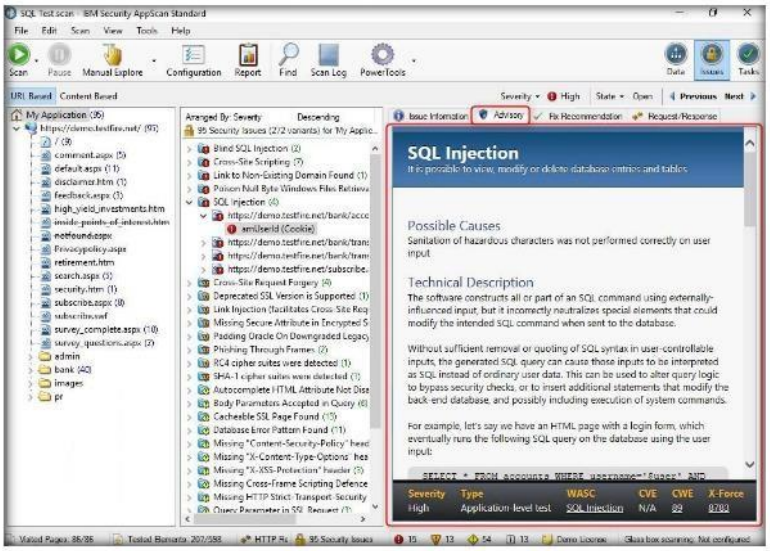


FIGURE 3.17: IBM Rational AppScan Scanning Web Application Result window

The Security Report reports security issues found during the scan. Security information may be very extensive and can be filtered depending on your requirements. Six standard templates are included, but each can easily be tailored to include or exclude categories of information.

25. Click **Fix Recommendation** to seek some **advice** for **fixing** these vulnerabilities.

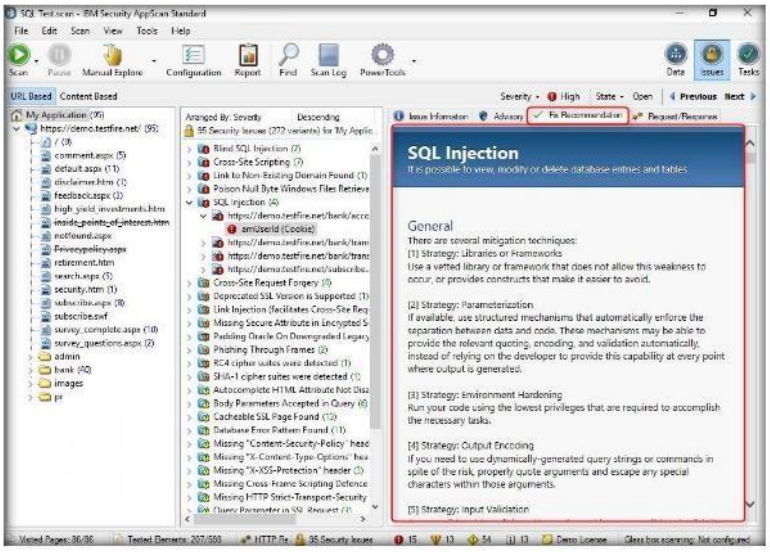


FIGURE 3.18: IBM Rational AppScan Scanning Web Application Result window

The Regulatory Compliance Report: It reports on the compliance (or non-compliance) of your application with a large choice of regulations or legal standards or with your own custom template).

TASK 4

Generate Report

26. After AppScan assesses your site's vulnerability, you can generate **customized reports** configured for the personnel in your organization.
27. You can open and view the reports from within Security AppScan, and you can **save a report** in any other format that can be opened with a third-party application.
28. To generate a report, click on **Tools → Create Report**. The **Create Report** window appears.

The Industry Standard Report reports the compliance (or non-compliance) of your application with a selected industry committee or your own custom standards checklist.

FIGURE 3.19: IBM Rational AppScan Report Option window

29. Select the type of report to generate, check options, and click **Save Report...**.

The Template Based Report is a custom report containing user-defined data and user-defined document formatting in Microsoft Word .doc format.

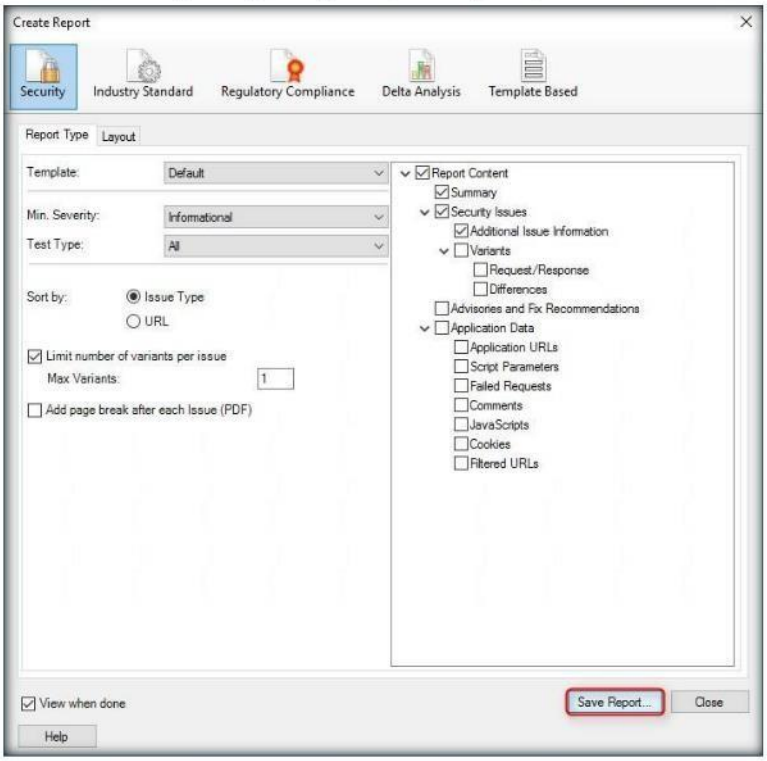


FIGURE 3.20: IBM Rational AppScan Create Report window

30. The **Save As** window appears; select the destination where you would save the scan report, name it, and click **Save**.

The Delta Analysis report compares two sets of scan results and shows the difference in URLs and/or security issues discovered.

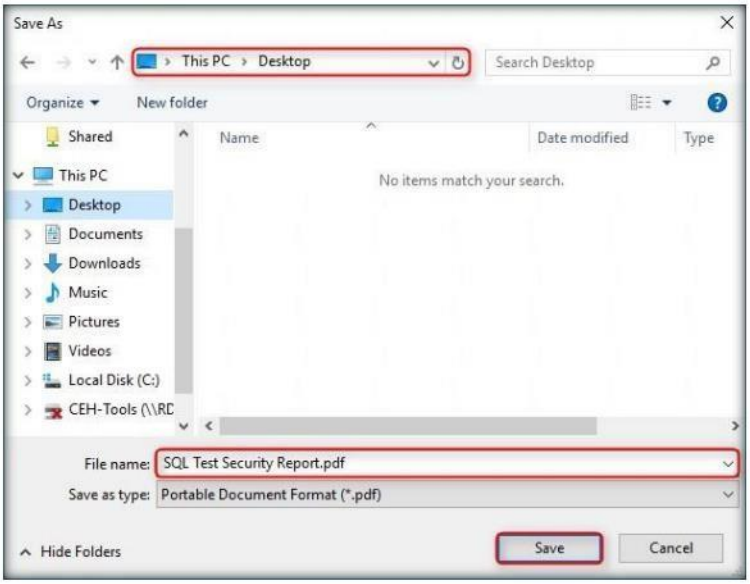


FIGURE 3.21: Save As window

31. The saved **report** will be helpful for future reference.

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 4

Scanning Web Applications using N-Stalker Tool

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

Tools demonstrated in this lab are available at Z:\CEH-Tools\CEHv10 Module 15 SQL Injection

N-Stalker 2012 is a sophisticated web security assessment solution for your web applications. By incorporating its well-known "N-Stealth HTTP Security Scanner" and its 39,000-item Web Attack Signature database, along with its patent-pending component-oriented security assessment technology, N-Stalker is a "must have" security tool for developers, system/security administrators, IT auditors, and others.

Lab Scenario

Few attackers perform SQL injection attacks based on "error messages" received from servers. If an error is responded to by the application, the attacker can determine the database's entire structure, and read any value that can be read by the account the ASP application is using to connect to the SQL server. However, if an error message is returned from the database server stating that the SQL Query's syntax is incorrect, an attacker tries all possible true/false questions via SQL statements to steal data.

As an Expert Security Professional and Penetration Tester, you should be familiar with the tips and tricks used in SQL injection detection. You must also be aware of all the tools that can be used to detect SQL injection flaws. In this lab, you will learn to do so using N-Stalker.

Lab Objectives

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

In this lab, you will learn to:

- Perform web site scans for vulnerabilities
- Analyze scanned results
- Save Scan Results

Lab Environment

You can download N-Stalker from http://www.nstalker.com/products/editions/free/download.

To complete this lab, you will need:

- N-Stalker located at **Z:\CEH-Tools\CEHv10 Module 15 SQL Injection\SQL Injection Detection Tools\N-Stalker Web Application Security Scanner**
- Run this tool in Windows Server 2016
- You can also download the latest version of N-Stalker from the link **https://www.nstalker.com/products/editions/free/download/**
- If you download the latest version of the tool then screenshots will vary
- A web browser with Internet access
- Microsoft .NET Framework Version 4.0 or later

Founded upon the U.S. Patent Registered Technology of Component-oriented Web Application Security Scanning, N-Stalker Enterprise Edition allows for assessment of Web Applications.

Lab Duration

Time: 10 Minutes

Overview of Testing Web Applications

Web applications are tested for implementing security and automating vulnerability assessments. Doing so prevents SQL injection attacks on web servers and web applications. Websites are tested for embedded malware and by employing multiple techniques.

Lab Tasks

TASK 1

Install N-Stalker

1. Navigate to **Z:\CEH-Tools\CEHv10 Module 15 SQL Injection\SQL Injection Detection Tools\N-Stalker Web Application Security Scanner**, double-click **NStalker-WebSecurityScanner-FreeX-b34.exe,** and follow the steps to install the application.
2. Once the installation is completed ensure that Run N-Stalker Web Application Security Scanner option is checked and uncheck Show Readme and then click **Finish**. **N-Stalker** application launched automatically.

3. Alternatively, you can also launch the application from **Start** → **N-Stalker Web Application Security** and click **N-Stalker Free X**, or double-click short-cut icon of the **N-Stalker Free X** from the **Desktop**.

FIGURE 4.1: Windows Server 2012 Apps screen

N-Stalker also allows you to create your own assessment policies and requirements, enabling an effective way to manage your application's SDLC, including the ability to control information exposure, development flaws, infrastructure issues and real security vulnerabilities that can be explored by external agents.

4. The N-Stalker GUI appears; click **Update** to update the application.

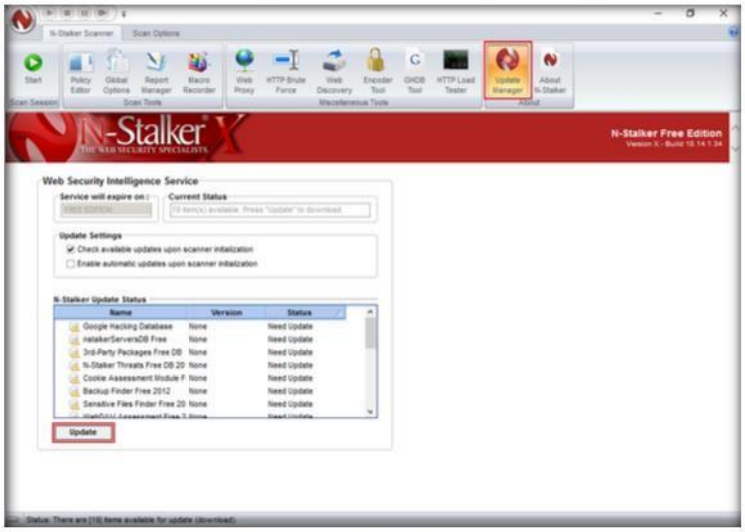


FIGURE 4.2: N-Stalker Main window

Web Security Intelligence Service (WSIS) is provided by WSI Labs and will ensure you always get the latest updates available for N-Stalker Web Application Security Scanner as well as for its attack signature database. New 0-day exploits and common vulnerabilities will be added on daily or weekly basis, giving you the ability to scan you Web Server infrastructure periodically against the latest threats.

System Requirement: .NET Framework V2.0 or higher, you can Download .NET Framework V2.0 From Microsoft.

5. The **N-Stalker Free Edition** pop-up appears; click **OK** to continue.

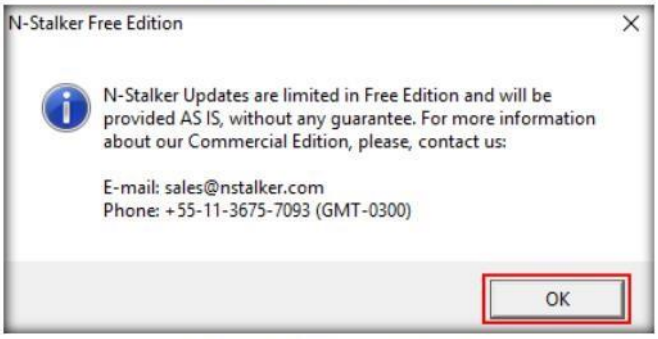


FIGURE 4.3: N-Stalker Free Edition pop-up

6. **N-Stalker** will start updating the database, which takes some time.

To run N-Stalker Web Application Security Scanner appropriately, there are minimum requirements to be met:

· 128MB RAM (available to N-Stalker)

· At least 500MB Hard Disk free space (caching purposes)

· Win32 Platform (Win 2000, XP, 2003 or Vista and later)

· Internet connection to download N-Stalker database/software updates

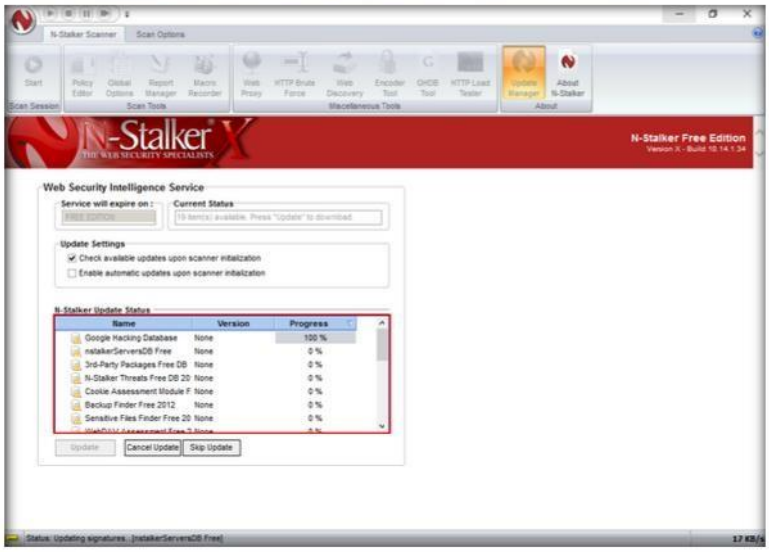


FIGURE 4.4: N-Stalker database updating status

TASK 2

Scan a Web Application

7. After updating is complete, click **Start** to start a new scanning session.

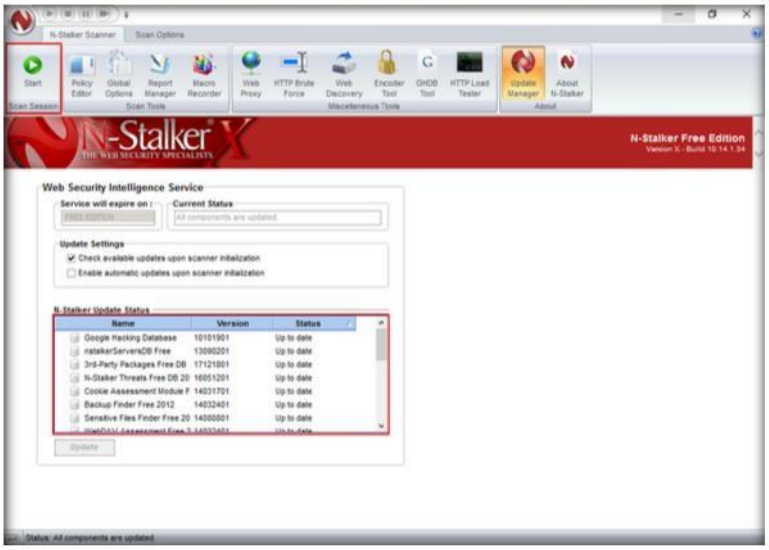


FIGURE 4.5: N-Stalker database updated

You may modify N-Stalker's cache options to avoid web pages from being permanently stored in your hard disk. This might be useful to preserve disk space on large assessments.

8. In the N-Stalker Scan Wizard, enter **http://www.goodshopping.com**.
9. Choose the Scan Policy **OWASP Policy**, and click **Next**.

To run N-Stalker Scanner from command line, you will need a scan session policy that will contain policies, host information and specific configurations needed to run the entire session.

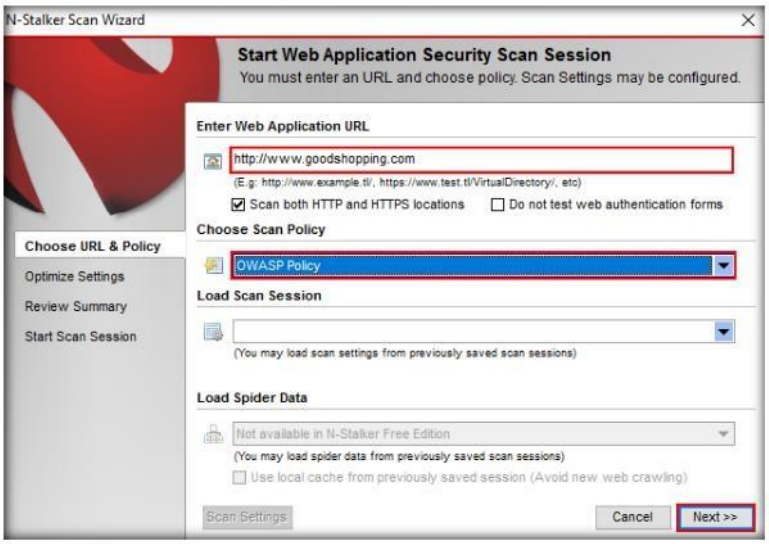


FIGURE 4.6: N-Stalker Choosing URL and Policy

10. Under **Optimize Settings**, leave the default options, and click **Next**.

N-Stalker HTTP Brute Force tool does what the name says. It is an HTTP authentication brute force tool that works by taking a web macro and attempting to run a series of authentication requests to obtain valid credentials (you may provide your own user and password list).

N-Stalker Web Proxy is a combination of web proxy and HTTP inspection tool. It includes a full Web Proxy support (for external browsers) along with an event-driven interception mechanism, that allows you to inspect HTTP communications (even SSL) based on keyword matching.

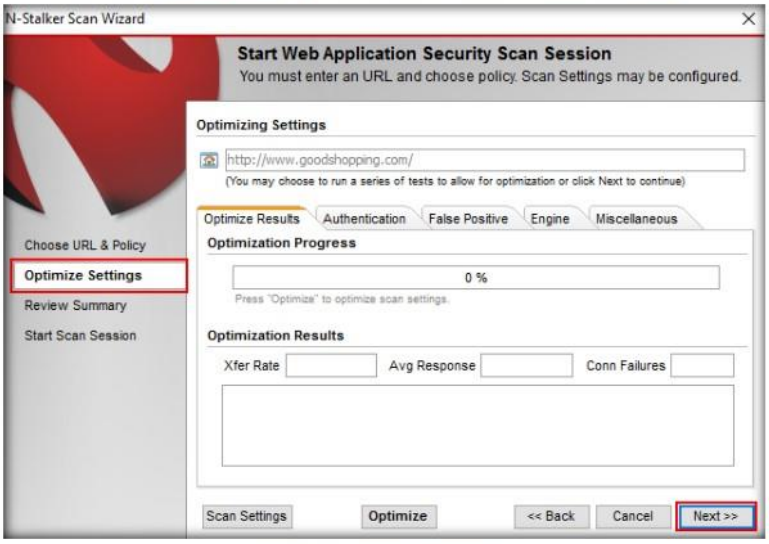


FIGURE 4.7: N-Stalker Optimize Settings

11. Click **Yes** in the **Settings Not Optimized** pop-up.

The term "GHDB" was allegedly coined by Johnny Long, which started to maintain a number of "google-based" queries that would eventually reveal security flaws in websites (without one having to scan the site directly for that vulnerability).

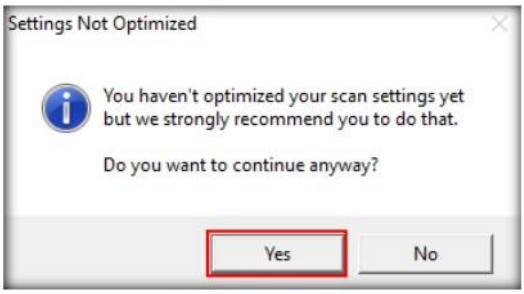


FIGURE 4.8: N-Stalker pop-up

12. Under **Review Summary**, click **Start Session**.

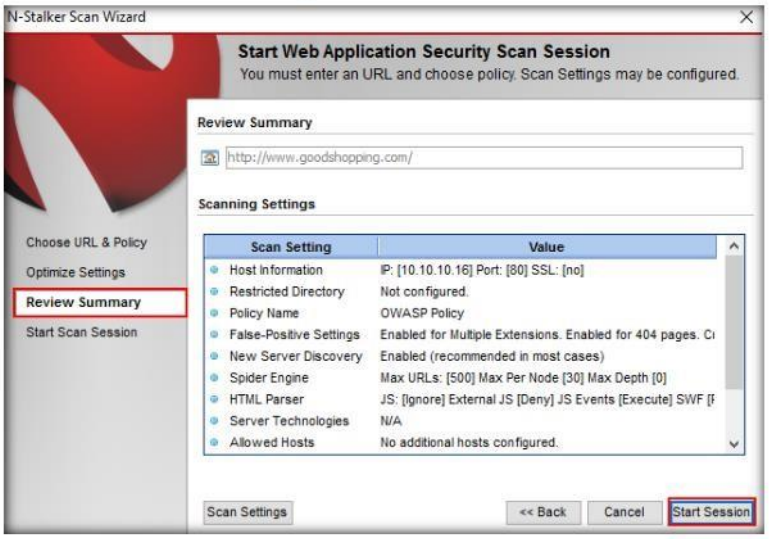


FIGURE 4.9: N-Stalker Review Summary

This is a string encoding tool which is useful to encode/decode data on multiple formats used by Web Applications.

13. The **N-Stalker free edition** pop-up appears; click **OK** to continue.

FIGURE 4.10: N-Stalker Free Edition pop-up

This is a Web Server Discovery tool which will attempt to discover HTTP servers and fingerprint them to obtain their platform version. It might run based on a file list or IP range.

14. After completing the configuration of N-Stalker, click **Start Scan** to begin scanning the Goodshopping website.

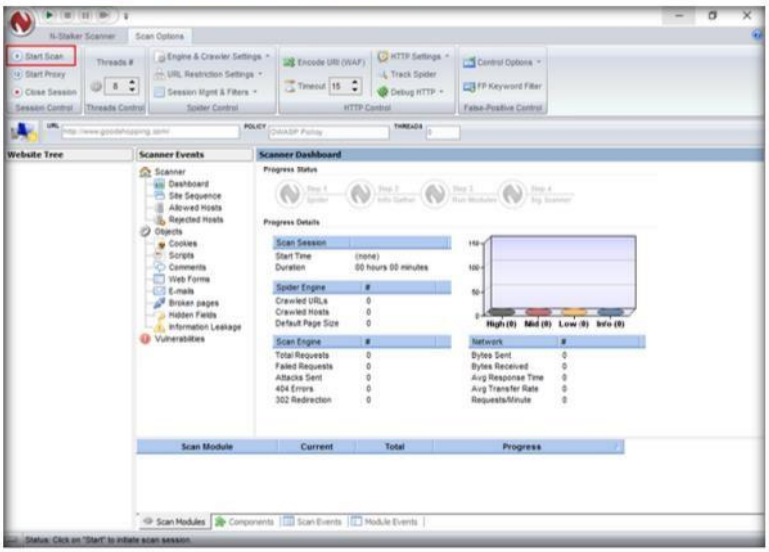


FIGURE 4.11: N-Stalker Start Scan wizard

15. N-Stalker begins to scan the **website**, as shown in the screenshot:

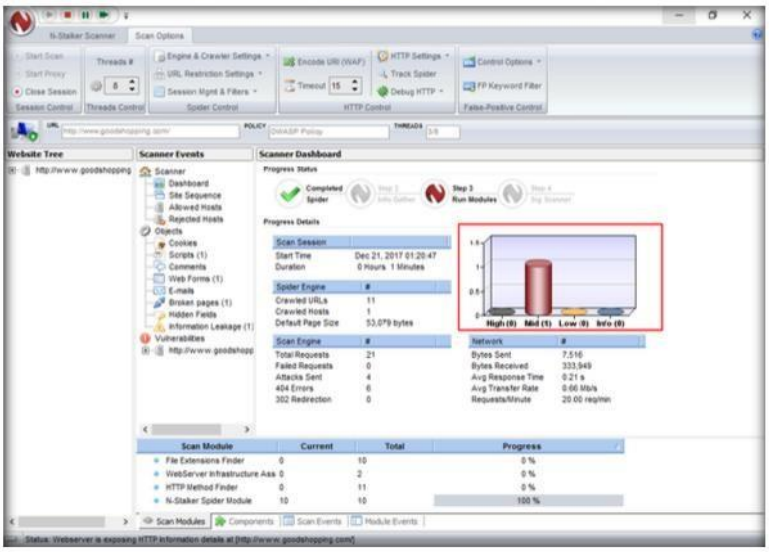


FIGURE 4.12: N-Stalker Start Scan Status

Google Hacking Database (GHDB) Tool is a unique application that will allow you to search for "google-like" queries within a saved spider data. N-Stalker, GHDB Tool can be invoked by clicking on "GHDB Tool" button under "Miscellaneous Tools."

HTTP Load Tester is a performance tester tool. It will run a Web Macro on a concurrent basis (up to you to decide how many instances) and will provide a report on number of connection failures and success.

16. It takes some time for the application to scan the entire website.
17. N-Stalker scans the site in four different steps: **Spider**, **Info Gather**, **Run Modules,** and **Sig Scanner**.

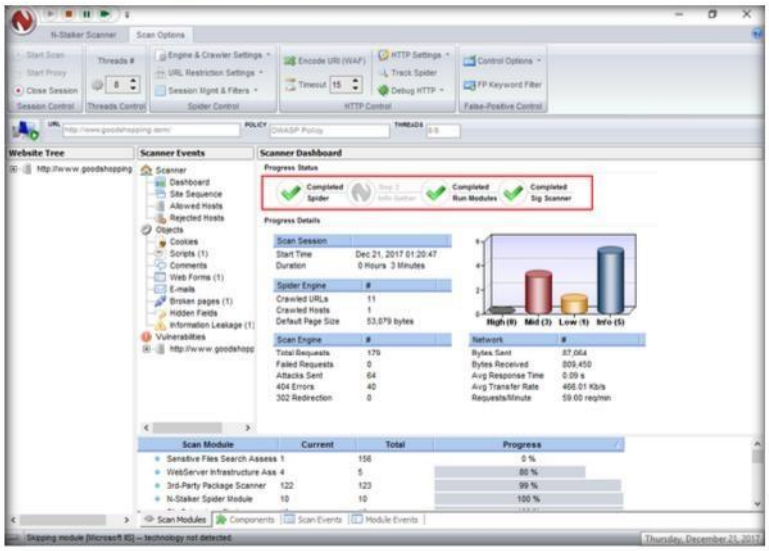


FIGURE 4.13: N-Stalker Scanning methods

Macro Recorder is a tool to manage "Web Macros" within N-Stalker Web Application Security Scanner.

An authentication Web Macro is used to authenticate N-Stalker's against Web Forms or any other of user interaction based authentication.

18. On completion of the scan, the **Results Wizard** appears.

TASK 3

Save the Scan Result

19. Select **Save scan results** (under **Session Management Options**) and **Keep scan session for further analysis** (under **Next Steps**), and click **Next**.

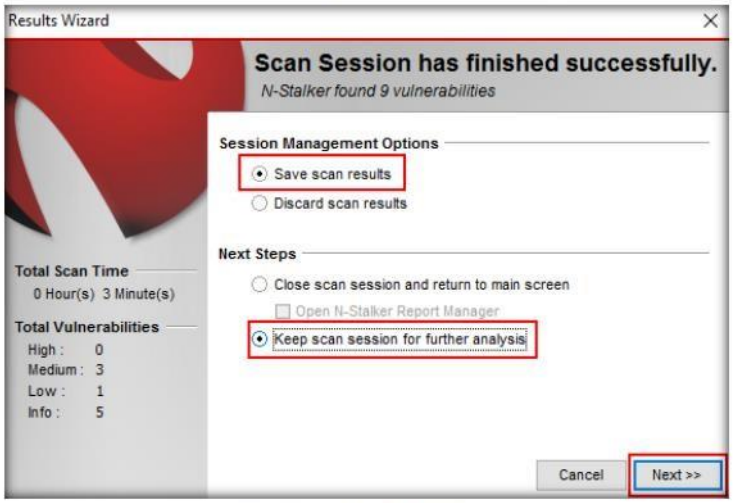


FIGURE 4.14: N-Stalker Results Wizard

As applications provide both a mean to login and logoff, Authentication Macros have a "logout detection" control that can be configured to prevent accidental logoff.

TASK 4

Analyze the Scan Result

20. N-Stalker displays a summary of vulnerabilities found. After examining the summary, click **Done**.

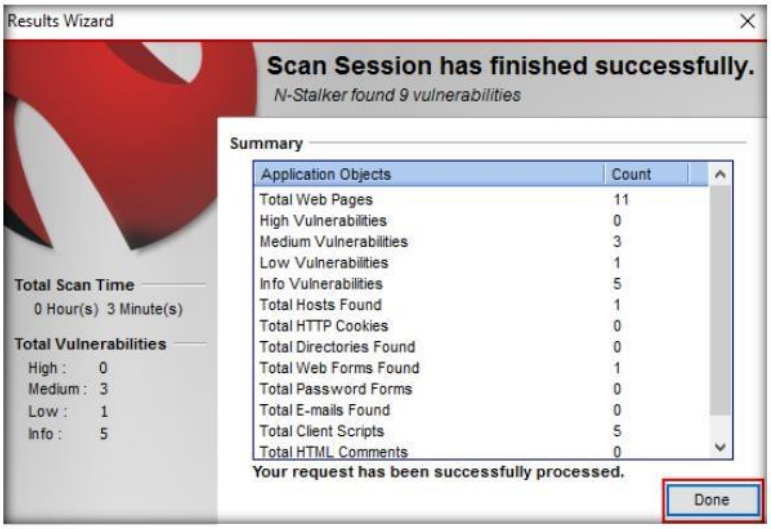


FIGURE 4.15: N-Stalker Summary

21. In the left pane, expand all the nodes and sub-nodes of the URL **http://www.goodshopping.com** (under **Website Tree**). This displays the website's pages.

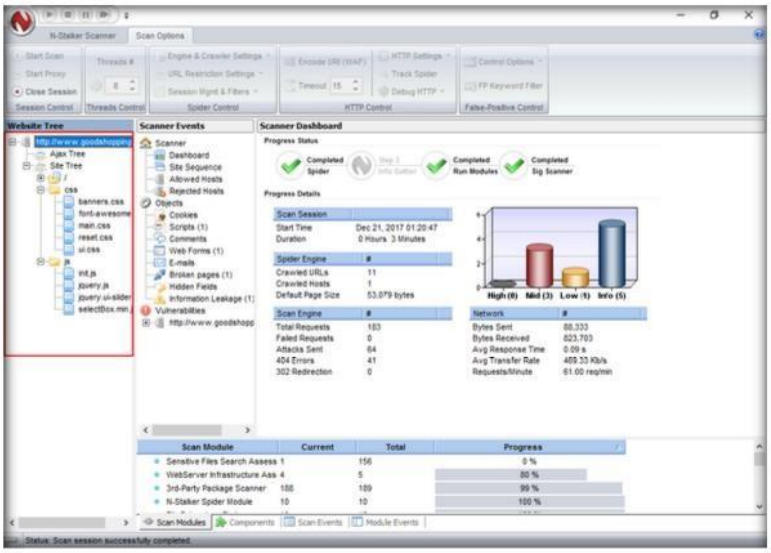


FIGURE 4.16: N-Stalker Website Tree

A navigation Web Macro is used to provide a specific path within the application to be followed by N-Stalker's spider engine.

When you are generating reports, N-Stalker allows you to customize template and data that will be used to generate the final report. Both executive and technical reports allow for that customization.

"Web Macro" is a user-provided navigation script that is usually recorded using a web browser and a web proxy tool. Macro Recorder allows you to insert manual URLs as well and you must choose between an authentication or navigation macro.

22. You can view the complete scan results in **N-Stalker's** main dashboard.
23. You can even expand the URL **http://www.goodshopping.com** (under **Vulnerabilities**) to view all the site's vulnerabilities.

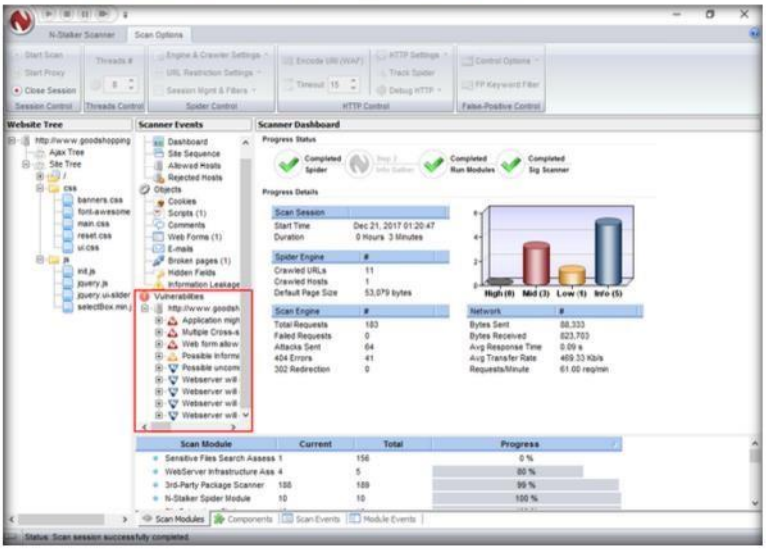


FIGURE 4.17: N-Stalker Dashboard

24. On completion of this lab, close the N-Stalker GUI.

📖 These macros can use any URLs and will not be prevented from calling external services within N-Stalker's spider engine.

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