SQL Injection

Module 14

SQL Injection

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

ICON KEY

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

Lab Scenario

A SQL injection attack is done 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 security vulnerability in a website's software. The 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, and 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 b**lind 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 D:\CEH-Tools\CEHv8 Module 14 SQL Injection

Lab Environment

To carry out the lab, you need:

- A computer running **Windows Server 2012**
- **Window 7** running in virtual machine
- A web browser with an Internet connection
- Administrative privileges to configure settings and run 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

Lab Tasks

Overview

Recommended labs to assist you in SQL Injection:

- Performing b**lind SQL injection**
- Logging on without **valid credentials**
- Testing for **SQL injection**
- Creating your **own user account**
- Creating your **own database**
- **Directory** listing
- **Denial-of-service** attacks
- Testing for SQL injection using the **IBM Security AppScan** tool

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.

Lab 1

SQL Injection Attacks on MS SQL Database

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

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 can 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
- Execute **denial-of-service** attacks

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv8 Module 14 SQL Injection

Lab Environment

To carry out the lab, you need:

- A computer running **Window Server 2012** (Victim Machine)

- A computer running **Window 8** (Attacker Machine)
- **MS SQL Server** must be running under local system privileges
- A web browser with an Internet connection

Lab Duration

Time: 30 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

Log on without Valid Credentials

Blind SQL injection is used when a web application is **vulnerable** to 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.

TASK1

1. Run this lab in **Firefox**. It will not work in Internet Explorer.
2. Open a web browser, type **http://localhost/realhome** in the address bar, and press **Enter**.

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

3. The **Home page** of Real Home appears.

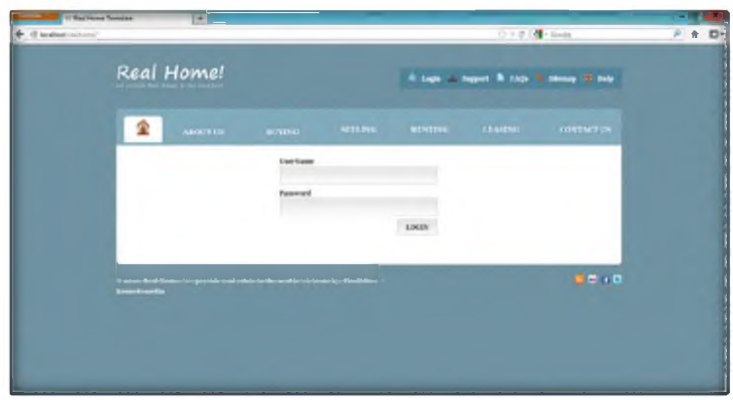


FIGURE 1.1: Old House Restaurant home page

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

4. Assume that you are new to this site and have never **registered** with this website previously.
5. Now log in with code:

```
blah' or 1=1 --
```

6. Enter any password in the **Password** field or leave the password field empty.
7. Click **Login** or press **Enter**.

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=''.

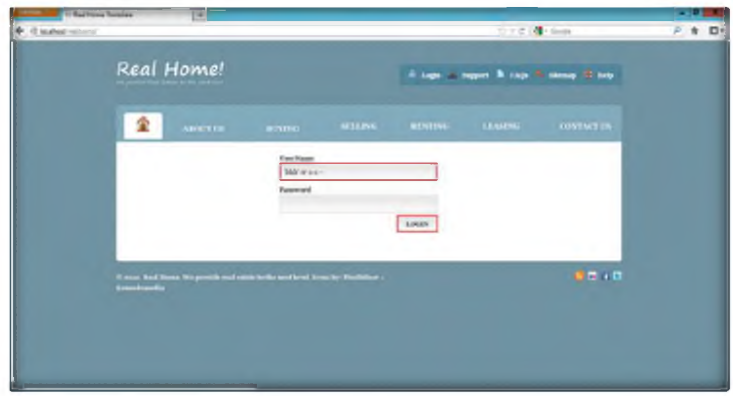


FIGURE 1.2: Old House Restaurant login page

8. You are logged in to the website with a fake login. Your credentials are not valid, but you are logged in. Now you can browse all the web pages of the website as a registered member. You will get a **Logout** link at the upper-corner of the screen.

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

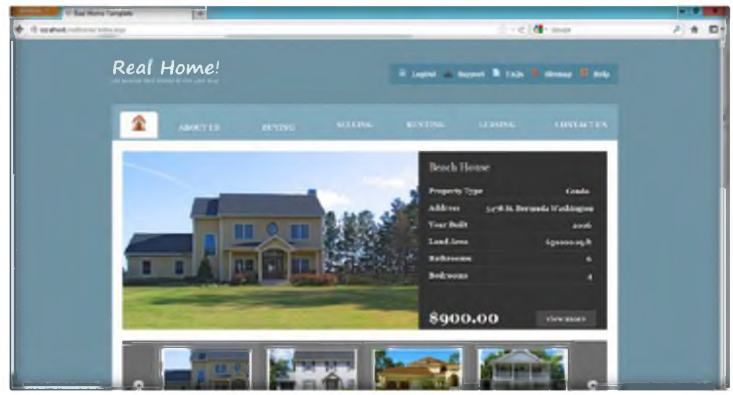


FIGURE 1.3: Old House Restaurant web page

You have successfully logged on to the vulnerable site and created your own database.

TASK2

TASK 2

Creating Your Own User Account

Create a user account using an SQL injection query.

9. Open a web browser, type **http://localhost/realhome** and press **Enter**.
10. The home page of Real Home appears.

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

FIGURE 1.4: Old House home page

11. Enter the query

    ```
    blah';insert into login values ('juggyboy','juggy123'); --
    ```

    in the Login name field and enter any password in the **Password** field or leave the **Password** field empty. In this query, **juggyboy** is the username, and **juggy123** is the password.

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

12. After executing the query you will be redirected to the login page; this is normal.
13. Try **juggyboy** as the username, and **juggy123** as the password to log in.
14. Click **Login** or press **Enter**.

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

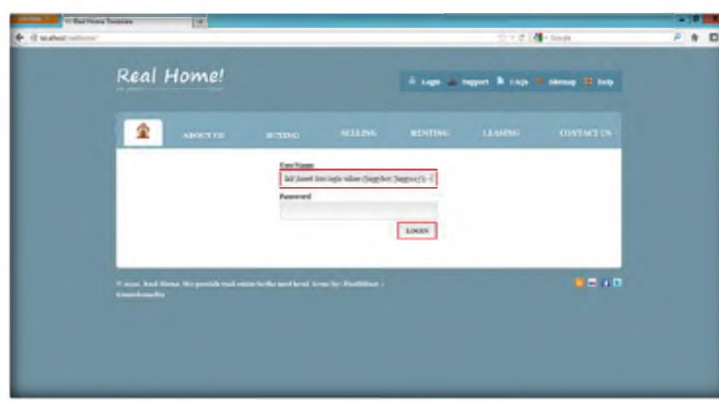


FIGURE 1.5: Old House Login page

15. If no error message is displayed on the web page, it means that you have successfully created your login using SQL injection query.
16. To **verify** whether your login has been created successfully, go to the login page, enter **juggyboy** in the **Login Name** field and **juggy123** in the **Password** field, and click **Login**.

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

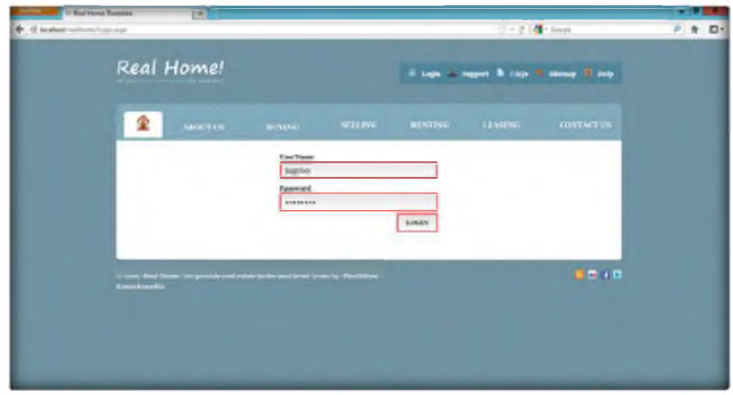


FIGURE 1.6: Old House Login page

17. You will login successfully with the created login. Now you can access all the features of the website.

    Go to **Start** menu apps and launch **SQL Server Management Studio** and login with the credentials.

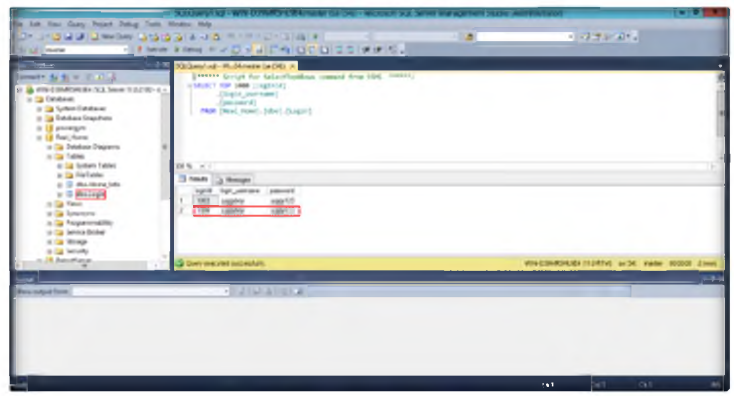


FIGURE 1.7: Old House Login page

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

TASK 3

Create Your Own Database

TASK 3

18. Open a web browser, type **http://localhost/realhome** in the address bar, and press **Enter**.
19. The **Home Page** of Real Home appears.

Most injections land in the middle of a SELECT statement. In a SELECT clause, we almost always end up in the WHERE section.

FIGURE 1.8: Old House Home page

20. In the **Login Name** field, type

    ```
    blah';create database juggyboy; --
    ```

    and leave the **Password** field empty. Click **Login**.

21. In this query, **juggyboy** is the name of the database.

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.

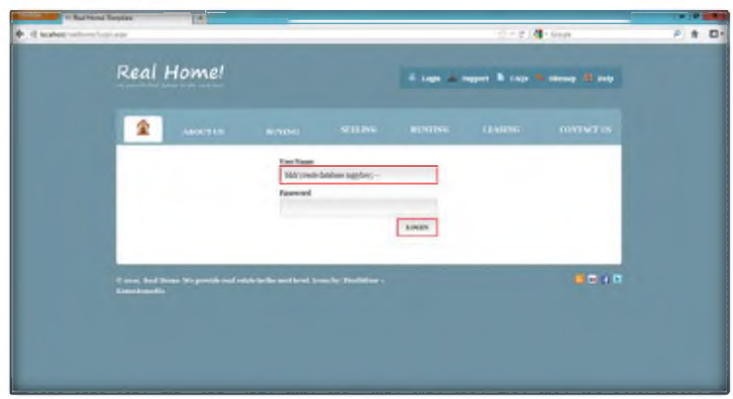


FIGURE 1.9: Old House Login page

22. No error message or any message displays on the web page. It means that the site is vulnerable to SQL injection and a database with the name juggyboy has been created at the database server.

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

23. When you open **Microsoft SQL Server Management Studio**, under **Database** you can see the created database, **juggyboy**.

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.

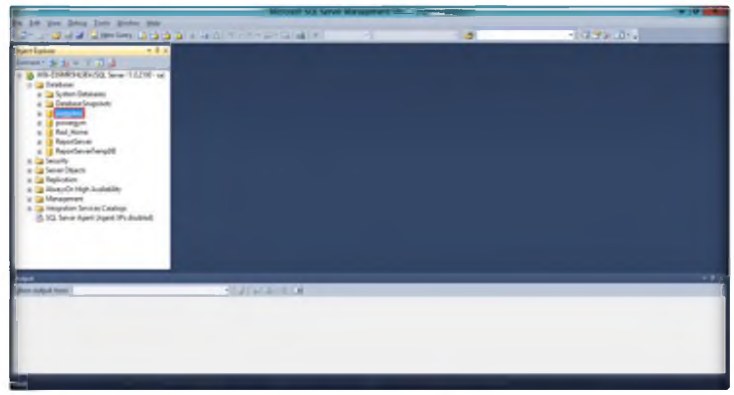


FIGURE 1.10: Microsoft SQL Server Management Studio

TASK 5

Denial-of-Service Attack

24. Open a web browser, type **http://localhost/realhome** in the address bar, and press **Enter**.
25. The **Home Page** of Real Home is displayed.

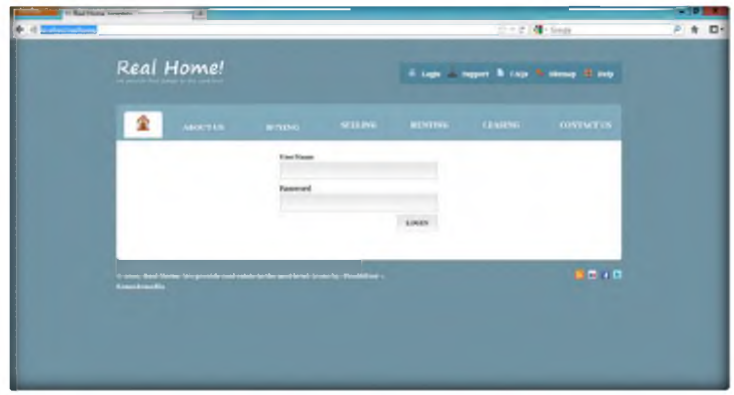


FIGURE 1.11: Old House Home page

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

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

26. In the **Login name** field, type

```
blah';exec master..xp_cmdshell 'ping
www.certifiedhacker.com -l 65000 -t'; --,
```

and leave the **Password** field empty, and click **Login**.

27. In the above query, you are performing a ping for the www.certifiedhacker.com website using an SQL injection query: **-l** is the send buffer size, and **-t** means to ping the specified host until stopped.

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'.

FIGURE 1.12: Old House Login page

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

28. The SQL injection query starts pinging the host, and the login page shows a **Waiting for localhost...** message at the bottom left side of the window.
29. To see whether the query has successfully executed or not and ping is running, open your **Task Manager** window.
30. In **Task Manager**, under the **Details** tab, you see a process called **PING.EXE** running in the background.
31. This process is the result of the SQL injection query that you entered in the login field of the website.

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

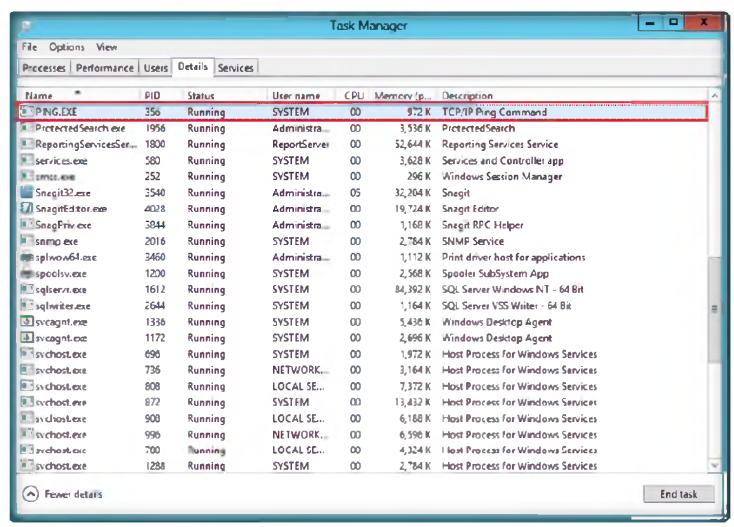


FIGURE 1.13: Task Manager

32. To manually kill this process, right-click the PING.EXE process and select **End Process.** This stops pinging of the host.

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
SQL Injection Attacks on MS SQL Database	▪ **Login id:** 1003, 1004 ▪ **Login Username:** juggyboy ▪ **Password:** juggy123

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED.

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

Lab 2

Testing for SQL Injection Using IBM Security AppScan Tool

ICON KEY

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

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

Lab Scenario

By now, you are familiar with the types of SQL injection attacks an attacker can perform and the impact caused due to 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 denial of service of the application, etc.

In the previous lab you learned to test SQL injection attacks on MS SQL database for website vulnerabilities.

As an expert **security professional** and **penetration tester** of an organization, 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 tests 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 tool.

Tools demonstrated in this lab are available D:\CEH-Tools\CEHv8 Module 14 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 website scans for vulnerabilities
- Analyze scanned results
- Fix vulnerabilities in web applications

- Generate reports for scanned web applications

Lab Environment

To carry out the lab, you need:

- **Security AppScan** located at **D:\CEH-Tools\CEHv8 Module 14 SQL Injection\SQL Injection Detection Tools\IBM Security AppScan**
- A computer running Window Server 2012
- Double-click on **SEC_APPS_STD_V8.7_EVAL_WIN.exe** to install
- 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

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

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: 20 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 a multiple of testing techniques.

TASK 1

Testing Web Application

Lab Tasks

1. Follow the wizard-driven installation steps and install the IBM Security AppScan tool.
2. To launch **IBM Security AppScan** move your mouse cursor to the lower-left corner of your desktop and click **Start**.

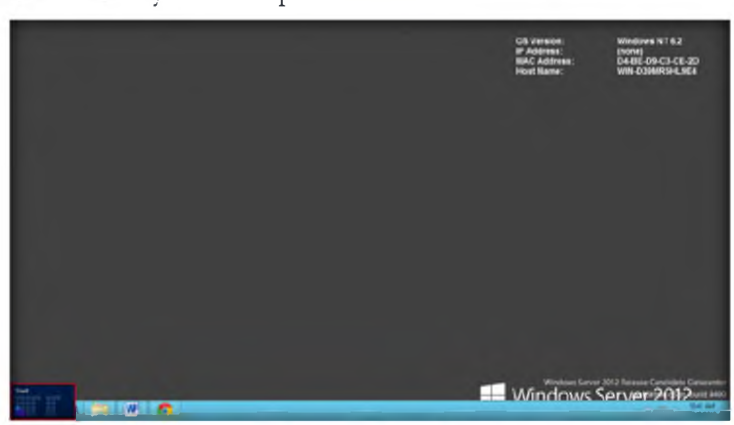


FIGURE 2.1: Windows Server 2012 Desktop view

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.

3. Click the **IBM Security AppScan Standard** app from **Start** menu apps.

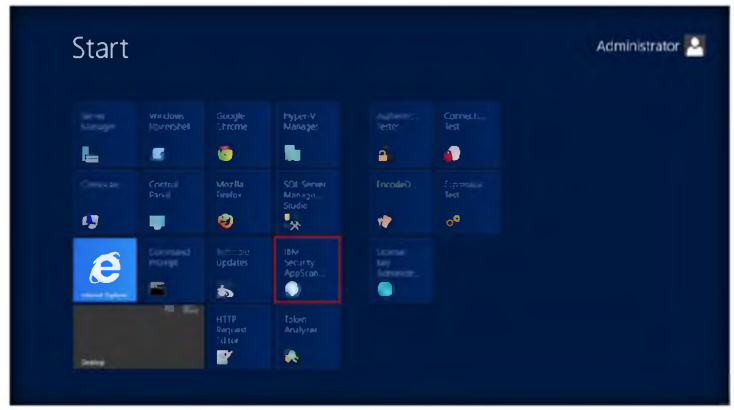


FIGURE 2.2: Windows Server 2012 Desktop view

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

4. The main window of **IBM Security AppScan** – appears; click **Create New Scan...** to start the scanning.

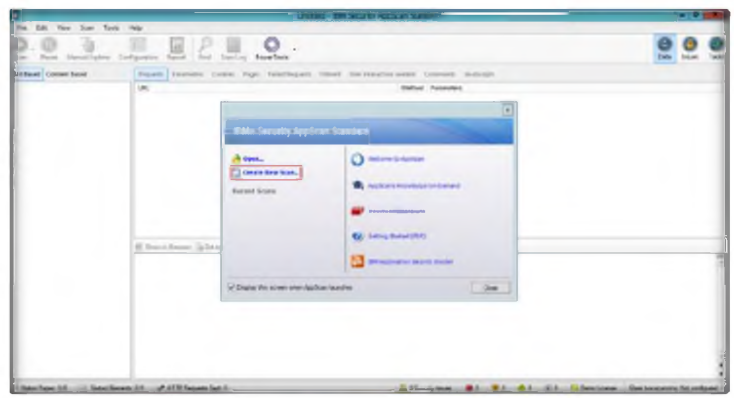


FIGURE 2.3: IBM Rational AppScan main window

AppScan can scan both web applications and web services.

5. In the **New Scan** wizard, click the **demo.testfire.net** hyperlink.

Note: In the evaluation version we cannot scan other websites.

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

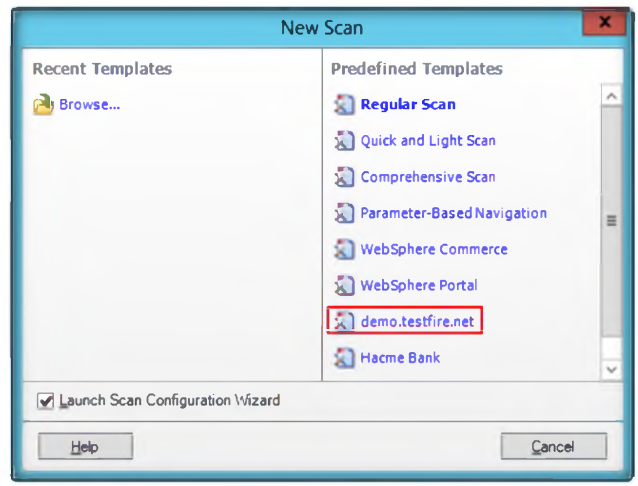


FIGURE 2.4: IBM Rational AppScan – New window

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.

6. In the **Scan Configuration Wizard**, select **Web Application Scan**, and click **Next**.

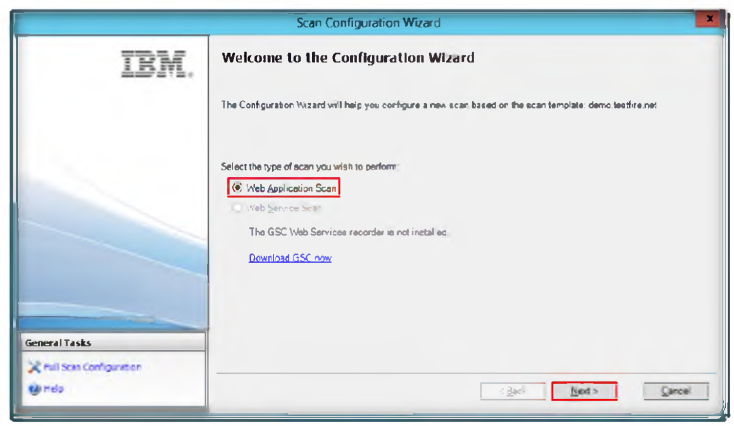


FIGURE 2.5: IBM Rational AppScan – Scan Configuration Wizard

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.

7. In **URL and Servers** options, leave the settings as their defaults and click **Next**.

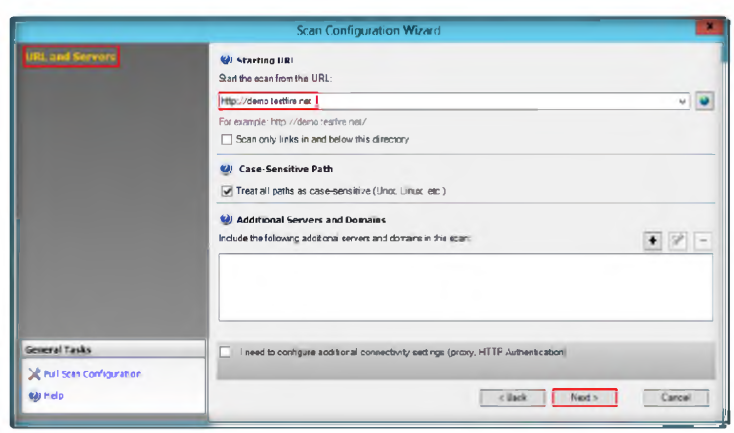


FIGURE 2.6: IBM Rational AppScan – Scan Configuration Wizard

8. In **Login Management**, select option **Automatic** and enter the user name details as Username: **jsmith** and Password: **Demo1234** and click **Next**.

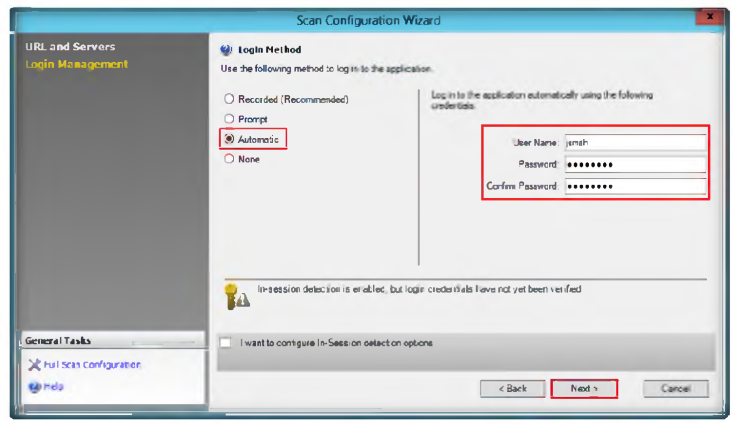


FIGURE 2.7: IBM Rational AppScan Scan Configuration window

9. In **Test Policy** options, click **Next** to continue.

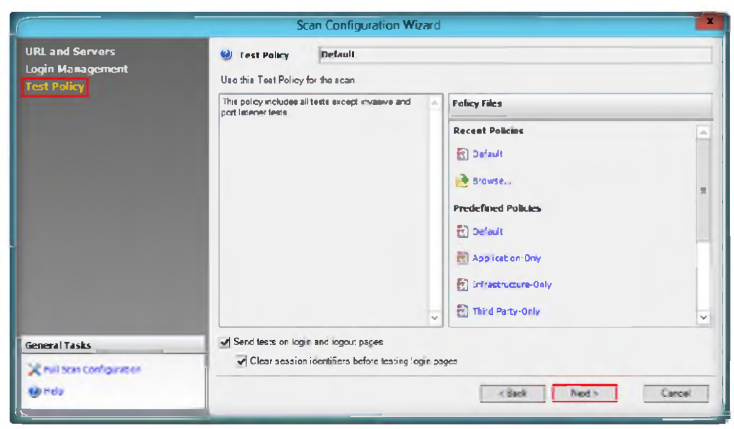
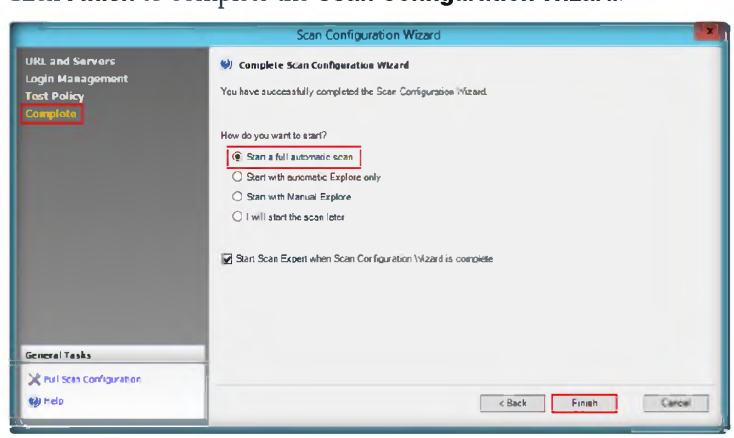


FIGURE 2.8: IBM Rational AppScan Full Scan window

10. Click **Finish** to complete the **Scan Configuration Wizard**.



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

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

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.

FIGURE 2.9: IBM Rational AppScan Full Scan window

11. When the **Auto Save** window prompts you to save **automatically during scan**, click **Yes** to save the file and proceed to scan.

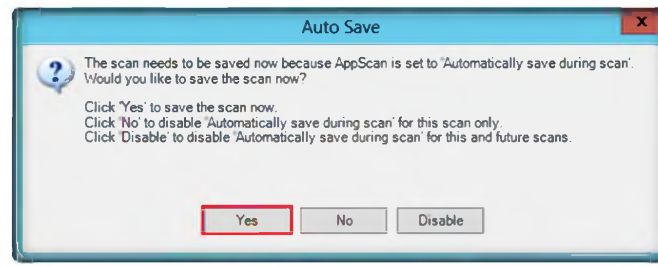


FIGURE 2.10: Auto Save window

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

12. Security AppScan starts scanning the provided URL for vulnerabilities.

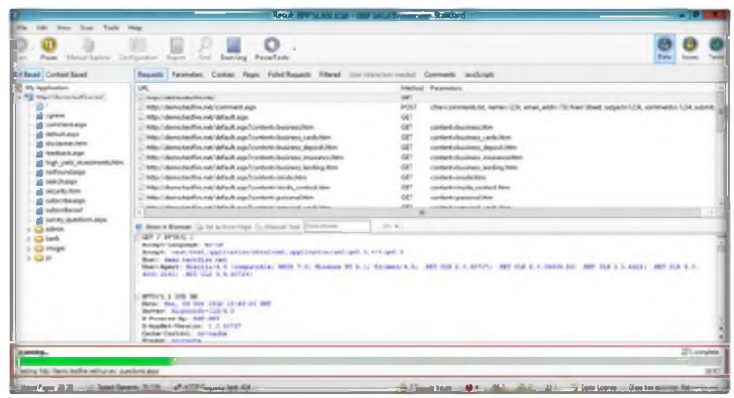


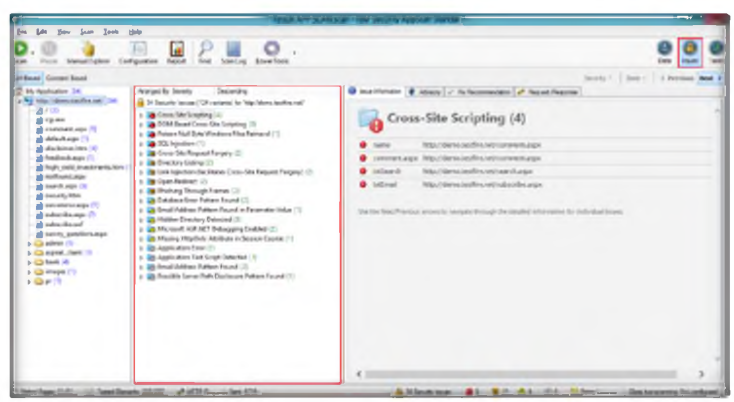
FIGURE 2.11: IBM Rational AppScan Scanning Web Application window

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

Note: It will take a lot of time to scan the complete site; in this lab we have stopped before scanning is complete.

13. After the scan is complete, the application lists all the security issues and vulnerabilities in the website.
14. Results can be displayed in three views: Data, Issues, and Tasks.
15. To view the vulnerabilities and security issues in particular website click the **Issues** tab.



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 2.12: IBM Rational AppScan Scanning Web Application Result window

TASK 2

Analyze Result

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

16. To analyze the scan results, click any of the results, such as **SQL Injection**, to list all the links that are vulnerable to SQL injection.

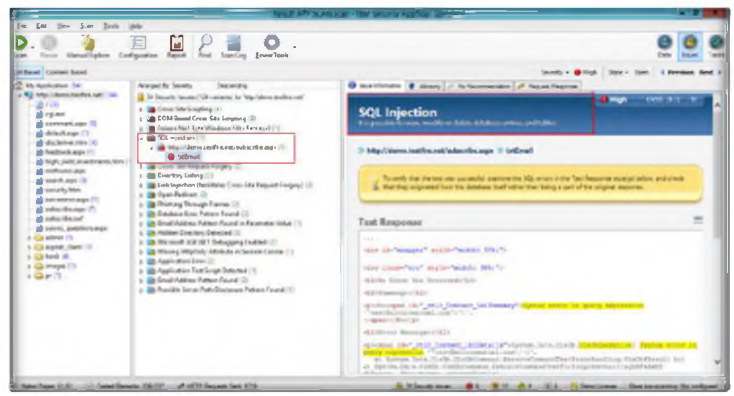


FIGURE 2.13: IBM Rational AppScan Scanning Web Application Result window

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.

17. Click the **Advisory tab** in the bottom pane of the window to see the severity of that particular link.

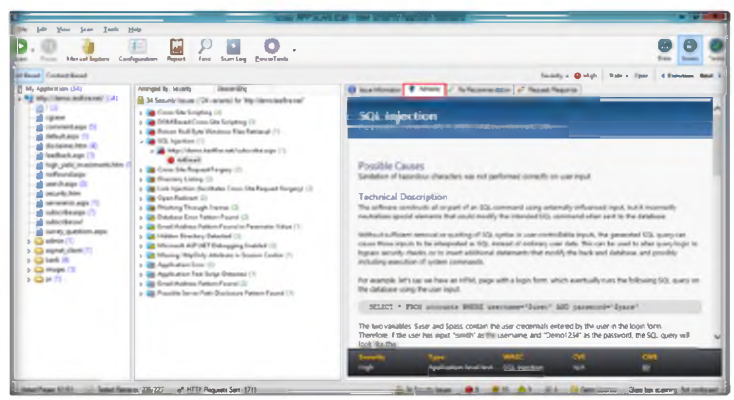


FIGURE 2.14: IBM Rational AppScan Scanning Web Application Result window

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.

18. To fix these threats and vulnerabilities, click **Fix Recommendation** to view a list of advice for fixing these vulnerabilities.

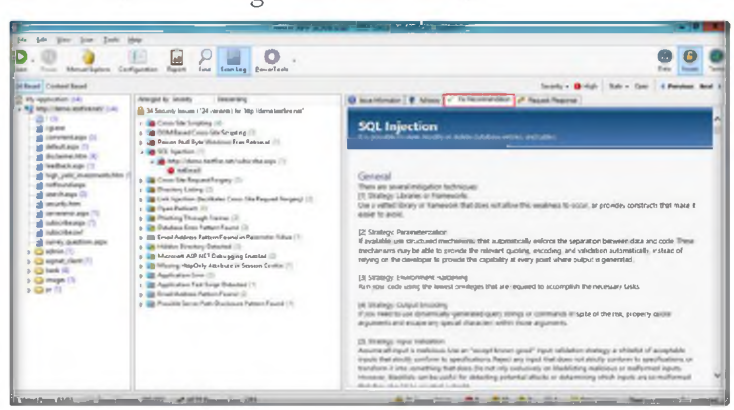


FIGURE 2.15: IBM Rational AppScan Scanning Web Application Result window

TASK 3

Generate Report

19. After Rational AppScan assesses your site's vulnerability, you can generate customized reports configured for the various personnel in your organization.
20. You can open and view the reports from within Security AppScan, and you can **save a report** as a file to be opened with a third-party application.
21. To generate a report, select **Tools → 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.

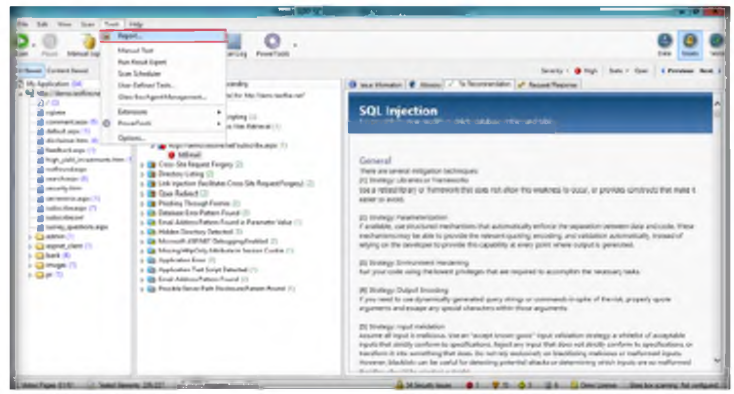
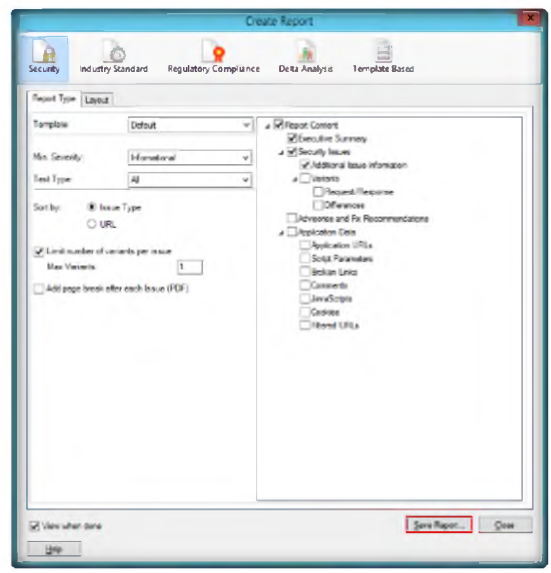


FIGURE 2.16: IBM Rational AppScan Report Option window

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

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

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



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).

FIGURE 2.17: IBM Rational AppScan Create Report window

23. Save the report to the desired location. The saved report will be helpful for future guidance.

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
IBM Security AppScan	▪ SQL Injection attack detected

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

Questions

1. Analyze how to speed up the scanning process and reduce the number of pages that IBM Rational AppScan finds.
2. Evaluate whether it is possible to perform scans against live production environments with IBM Rational AppScan. Will that cause damage or hurt the site?
3. Analyze how variables can be implemented in a multi-step sequence with IBM Rational AppScan.

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

Lab 3

Testing for SQL Injection Using WebCruiser Tool

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

WebCruiser - Web Vulnerability Scanner is an effective and powerful web penetration testing tool that will aid you in auditing your website. It has a Vulnerability Scanner and a series of security tools.

Lab Scenario

A deeper understanding of detecting SQL injection attacks using the IBM Security AppScan too was examined in the previous lab. In this lab we will have a look at a real case scenario where SQL injection attacks were implemented to steal confidential information from banks.

Albert Gonzalez, an indicted hacker, stole 130 million credit and debit cards, the biggest identity theft case ever prosecuted in the United States. He used SQL injection attacks to install sniffer software on the companies' servers to intercept credit card data as it was being processed.

He was charged for many different cases in which the methods of hacking utilized were:

- Structured Query Language ("SQL") was a computer programming language designed to retrieve and manage data on computer databases.
- "SQL Injection Attacks" were methods of hacking into and gaining unauthorized access to computers connected to the Internet.
- "SQL Injection Strings" were a series of instructions to computers used by hackers in furtherance of SQL Injection Attacks.
- "Malware" was malicious computer software programmed to, among other things, identify, store, and export information on computers that were hacked, including information such as credit and debit card numbers and corresponding personal identification information of cardholders ("Card Data"), as well as to evade detection by anti-virus programs running on those computers.

As an expert **security professional** and **penetration tester** you should have a complete understanding of SQL injection attack scenarios and list high=risk

components and note entry points to start testing and exploring. Hence, as another aspect in SQL Injection testing, in this lab you will be guided to test for SQL injection using the WebCruiser Tool.

Lab Objectives

Tools demonstrated in this lab are available D:\CEH-Tools\CEHv8 Module 14 SQL Injection

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 website scans for vulnerabilities
- Analyze scanned results
- Fix vulnerabilities in web applications
- Generate reports for scanned web applications

Lab Environment

You can download WebCruiser from http://sec4app.com/download

To carry out the lab, you need:

- **WebCruiser** located at **D:\CEH-Tools\CEHv8 Module 14 SQL Injection\SQL Injection Detection Tools\WebCruiser**
- Run this tool in Window Server 2012
- You can also download the latest version of **WebCruiser** from the link http://sec4app.com/download.htm
- A web browser with Internet access
- Microsoft .NET Framework Version 4.0 or later

To produce time-consuming SQL sentence and get information from the response time

Lab Duration

Time: 20 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

Testing Web Application

Lab Tasks

1. To launch WebCruiser in your Windows Server 2012 host machine, navigate to **D:\CEH-Tools\CEHv8 Module 14 SQL Injection\SQL Injection Detection Tools\WebCruiser**.
2. Double-click **WebCruiserWVS.exe** to launch it.

Scanning is not necessary for SQL Injection POC, you can launch POC by input the URL directly, or launch from the Scanner.
WebCruiser support:
* GET/Post/Cookie Injection;
* SQL Server: PlainText/FieldEcho(Union)/Blind Injection;
* MySQL/DB2/Access: FieldEcho(Union)/Blind Injection;
* Oracle: FieldEcho(Union)/Blind/CrossSite Injection;

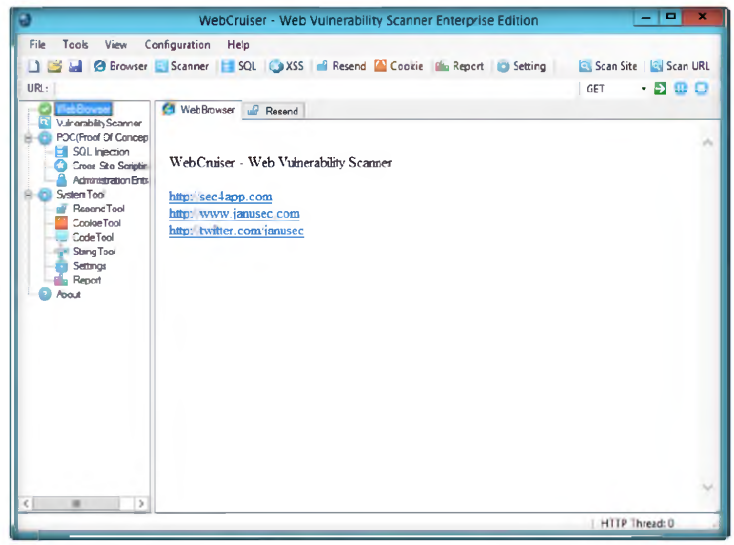


FIGURE 3.1: WebCruiser main window

3. Enter the URL that you want to scan; in this lab we are scanning **http://10.0.0.2/realhome/** (this IP address is where the realhome website is hosted).

WebCruiser Web Vulnerability Scanner for iOS, an effective and convenient web penetration testing tool that will aid you in auditing your website! WebCruiser can find the following web vulnerabilities currently:
* GET SQL Injection(Int, String, Search)
* POST SQL Injection(Int, String, Search)
* Cross Site Scripting(XSS)

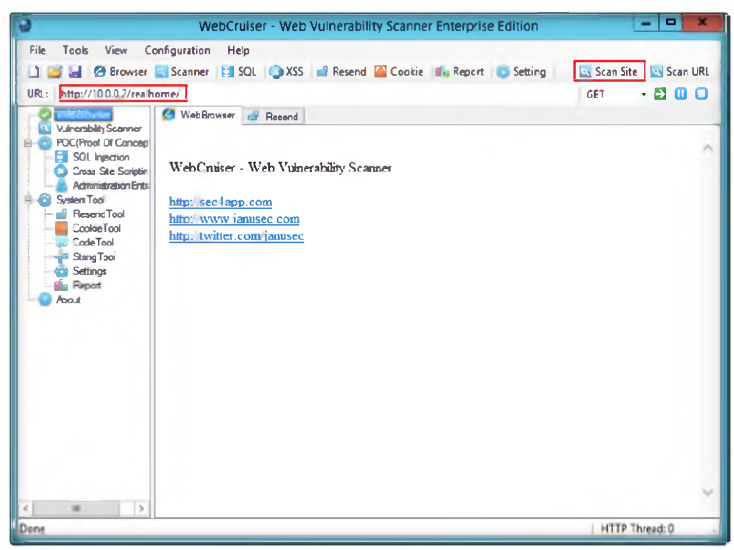


FIGURE 3.2: WebCruiser Scanning a site

It can support scanning website as well as POC (Proof of concept) for web vulnerabilities: SQL Injection, Cross Site Scripting, XPath Injection etc. So, WebCruiser is also an automatic SQL injection tool, an XPath injection tool, and a Cross Site Scripting tool!

4. A software disclaimer pop-up will appear; click .**OK** to continue.

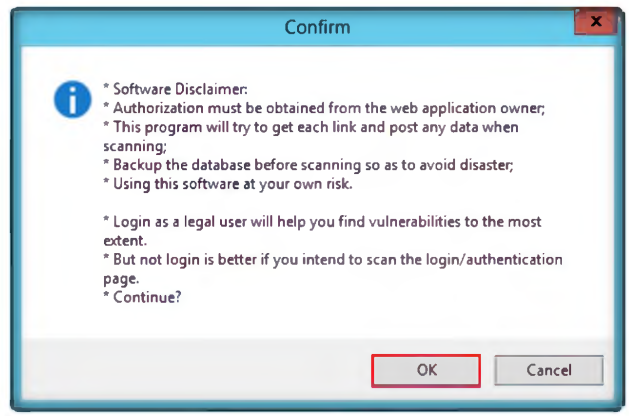


FIGURE 3.3: WebCruiser Software Disclaimer pop-up

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

5. WebCruiser starts with the URL scan as shown in the following screenshot. It shows Site Structure, and the following table is vulnerabilities.

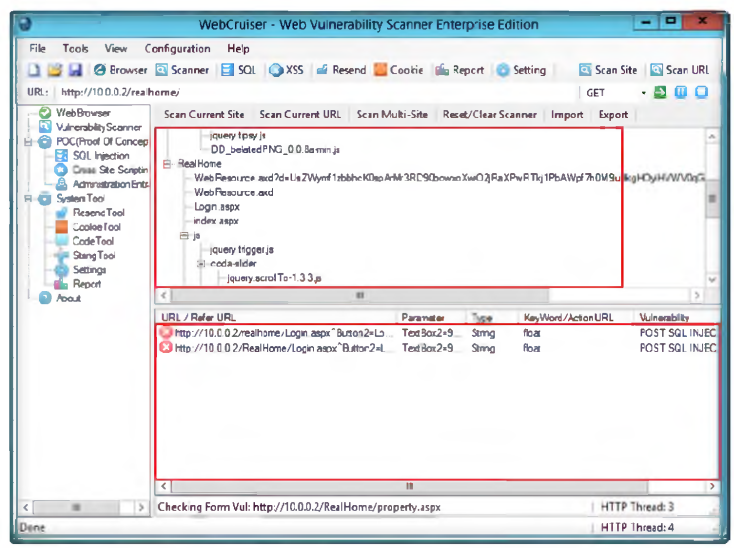


FIGURE 3.4: WebCruiser Scanning Vulnerabilities

SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application.

The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed.

6. Right-click each of the vulnerabilities displayed in the scan result, and then you can launch SQL Injection POC (Proof of Concept).

It is an instance of a more general class of vulnerabilities that can occur whenever one programming or scripting language is embedded inside another. SQL Injection is one of the most common application layer attack techniques used today.

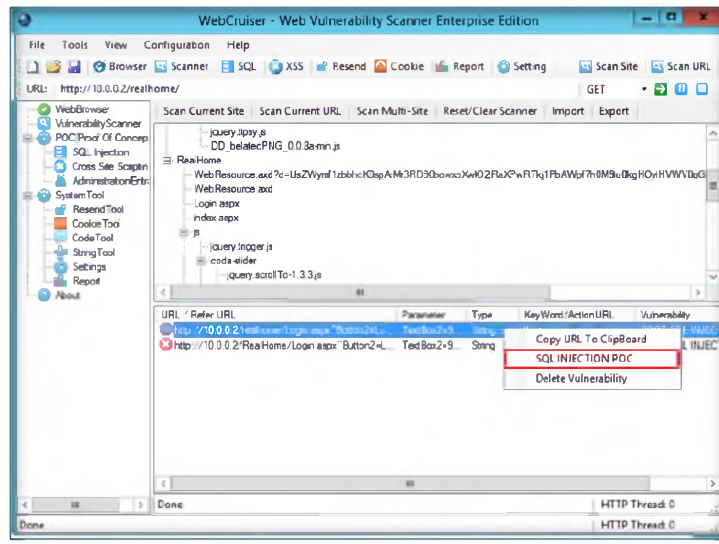


FIGURE 3.5: WebCruiser SQL Injection POC (Proof of Concept)

7. This will launch the SQL injection and fill the relevant fields. Click **Get Environment Information**.

There are many methods to getting data in SQL Injection, but not all these methods are supported in an actual penetration test.

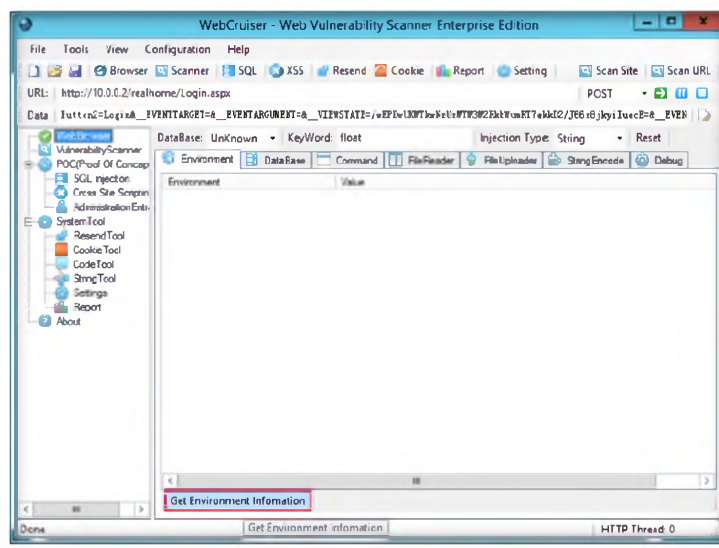


FIGURE 3.6: WebCruiser SQL Injection POC Tool

8. It will display the environment information where the site is hosted.

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
WebCruiser	▪ SQL Injection Detected

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

Questions

1. Analyze how to speed up the scanning process and reduce the number of pages the IBM Rational AppScan finds.
2. Evaluate whether it is possible to perform scans against live production environments with IBM Rational AppScan. Will that cause damage or hurt the site?
3. Analyze how variables can be implemented in a multi-step sequence with IBM Rational AppScan.

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

Lab 4

Testing for SQL Injection Using N-Stalker Tool

ICON KEY

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

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

Lab Scenario

In the previous lab you examined how to use the Webcruiser tool to scan a website as well as POC (Proof Of Concept) for web vulnerabilities: SQL injection.

Few attackers perform SQL injection attacks based on an "error message" received from the server. If an error is responded from the application, the attacker can determine the entire structure of the database, 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 complaining that the SQL Query's syntax is incorrect, an attacker tries all possible True and False questions through 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 use the tool N-Stalker to detect SQL injection attacks in websites.

Tools demonstrated in this lab are available D:\CEH-Tools\CEHv8 Module 14 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 website scans for vulnerabilities

- Analyze scanned results
- Fix vulnerabilities in web applications
- Generate reports for scanned web applications

Lab Environment

To carry out the lab, you need:

- **N-Stalker** located at **D:\CEH-Tools\CEHv8 Module 14 SQL Injection\SQL Injection Detection Tools\N-Stalker Web Application Security Scanner**
- Run this tool in Window Server 2012
- You can also download the latest version of **N-Stalker** from the link http://www.nstalker.com/products/editions/free/download
- A web browser with Internet access
- Microsoft .NET Framework Version 4.0 or later

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

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: 20 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.

Lab Tasks

TASK 1

Testing Web Application

1. To launch N-Stalker move your mouse cursor to the lower-left corner of your desktop and click **Start**.

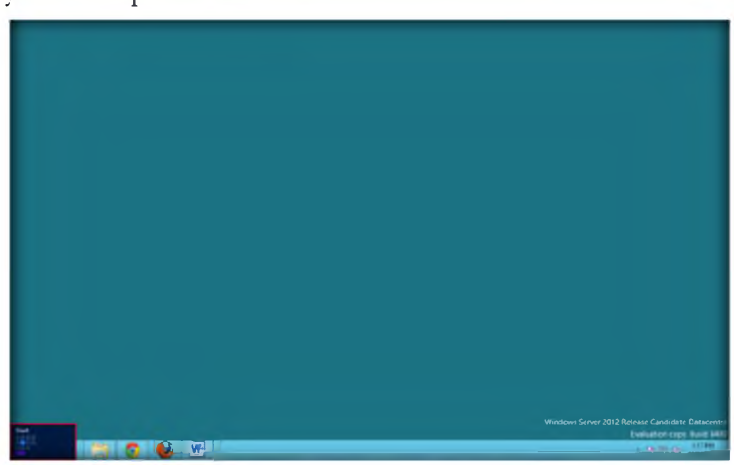


FIGURE 4.1: Windows Server 2012 Desktop view

2. Click the **N-Stalker Free 2012** app to launch it.

N-Stalker Web Application Security Scanner 2012 Enterprise Edition provides the most complete and effective suite of Web Security assessment checks to enhance the overall security of your Web Applications against a wide range of vulnerabilities and sophisticated hacker attacks.

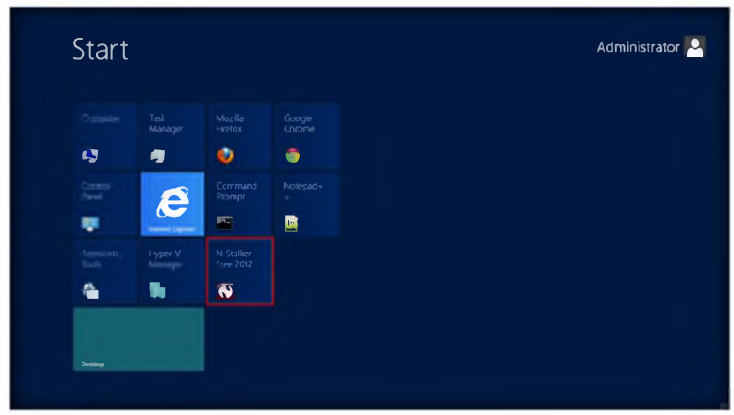


FIGURE 4.2: Windows Server 2012 Start menu Apps

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.

3. Click the **Update** button to update the N-Stalker database in the main window of N-Stalker as shown in the following screenshot.

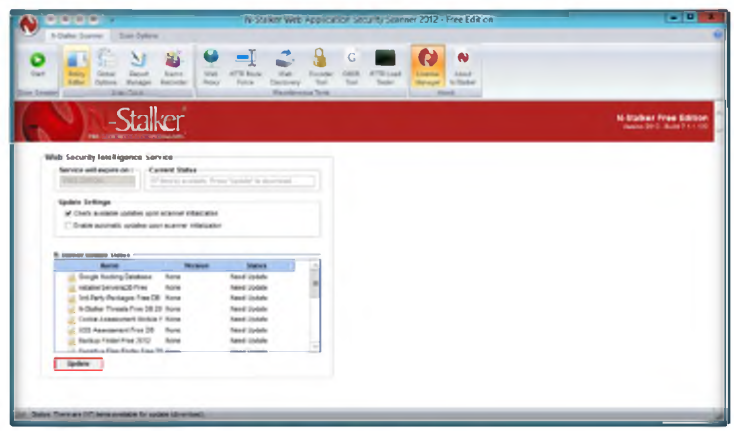


FIGURE 4.3: N-Stalker Main window

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.

4. A software disclaimer pop-up will appear. Click **OK** to continue.

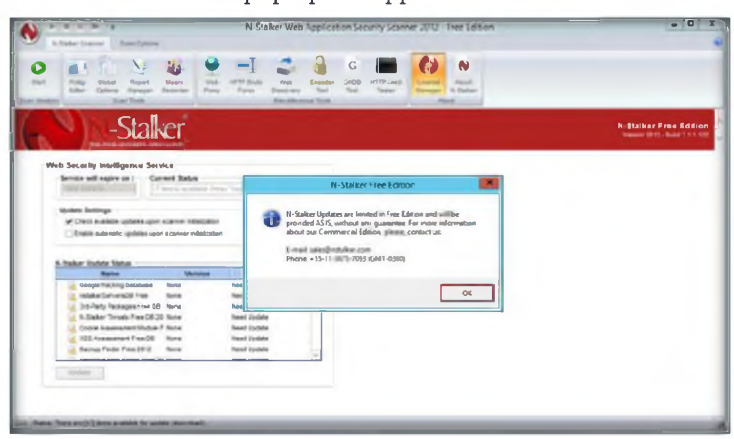


FIGURE 4.4: N-Stalker Free Edition pop-up

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

5. **N-Stalker** will start updating the database; it will take some time to update.

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

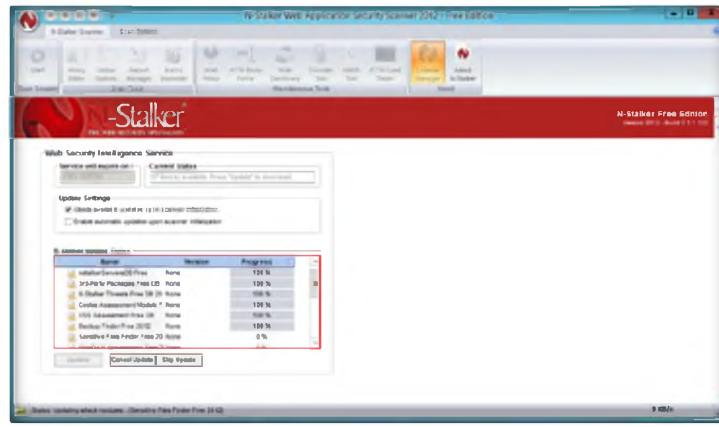


FIGURE 4.5: N-Stalker database updating status

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

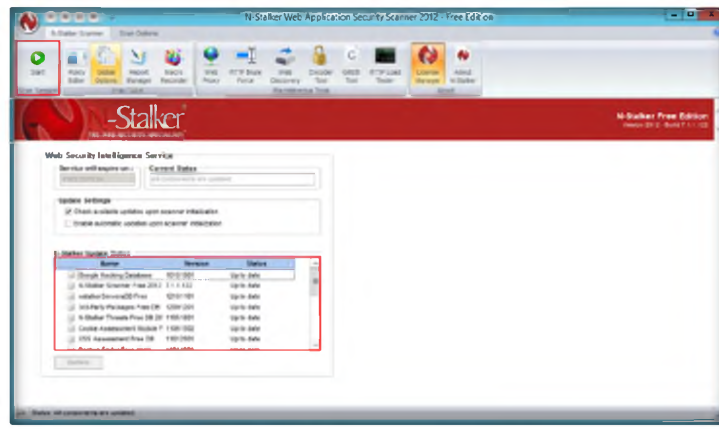


FIGURE 4.6: N-Stalker database updated

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

7. In **N-Stalker Scan Wizard**, enter the URL as **http://10.0.0.2/realhome/** (this IP address is where the realhome website is hosted).
8. Set the **Scan Policy** as **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.

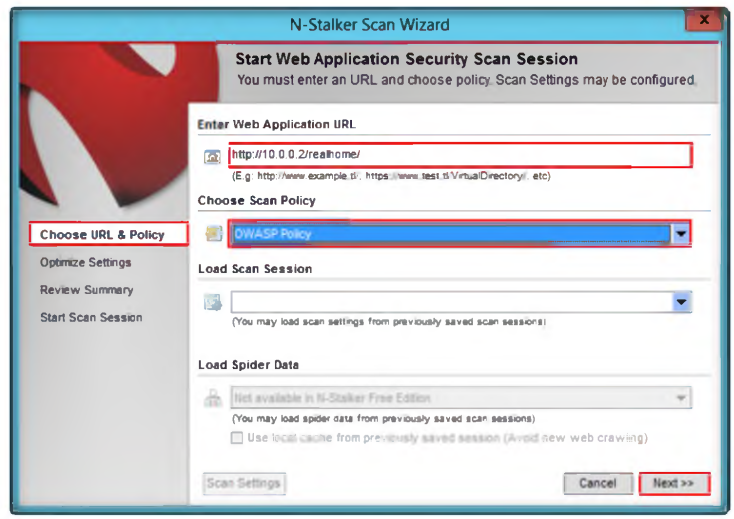


FIGURE 4.7: N-Stalker Choosing URL and Policy

9. Click .**Yes** in the **URI Restriction Found** pop-up to continue.

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).

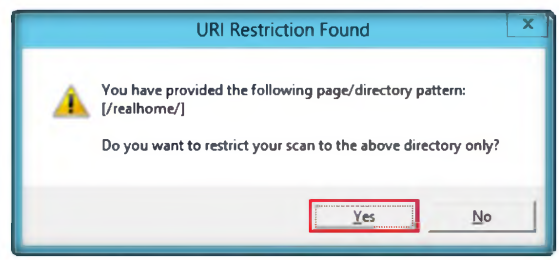


FIGURE 4.8: N-Stalker URI Restriction Found pop-up

10. In Optimize Settings, click **Next** to continue.

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.

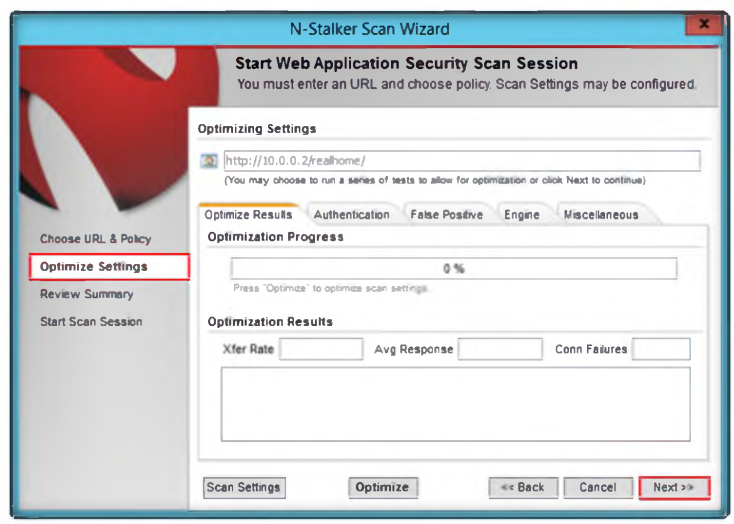


FIGURE 4.9: N-Stalker Optimize Settings

11. Click **Yes** in the **Optimize Settings** 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).

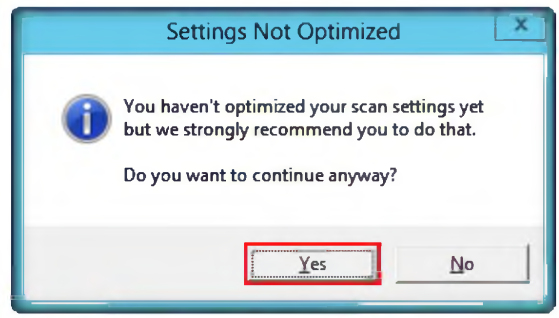


FIGURE 4.10: N-Stalker pop-up

12. On the **Review Summary** tab, click **Start Session** to continue.

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

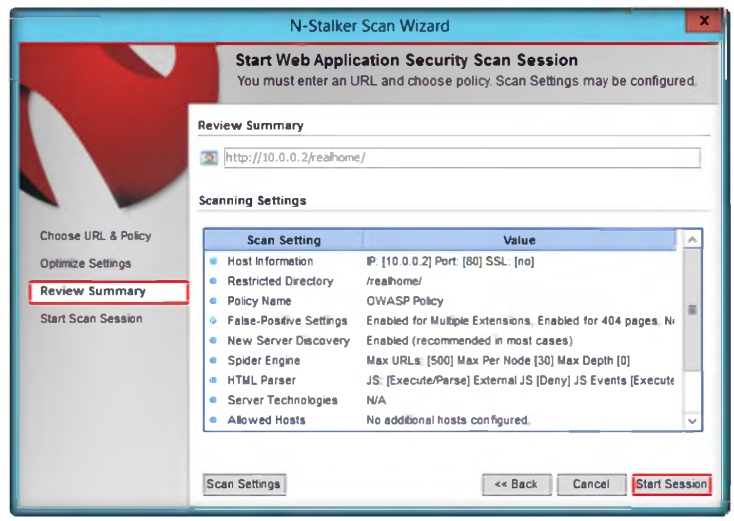


FIGURE 4.11: N-Stalker Review Summary

13. The **N-Stalker Free Edition** pop-up displays a message. Click **OK** to continue.

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.

FIGURE 4.12: N-Stalker Free Edition pop-up

14. Click **Start Scan** after completing the configuration of N-Stalker.

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":

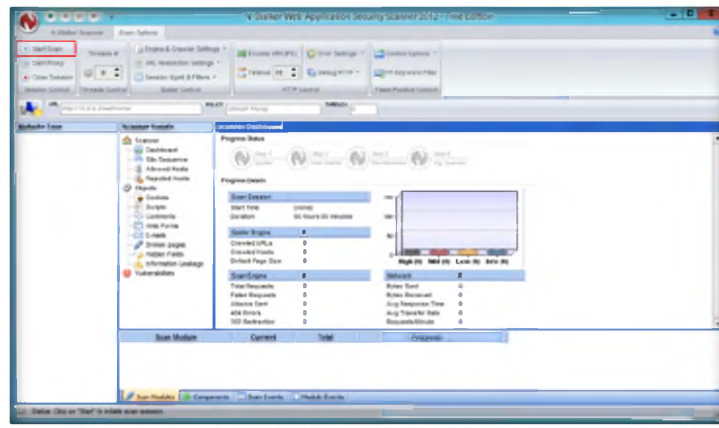


FIGURE 4.13: N-Stalker Start Scan wizard

15. You can view scanning details as shown in the following screenshot.

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.

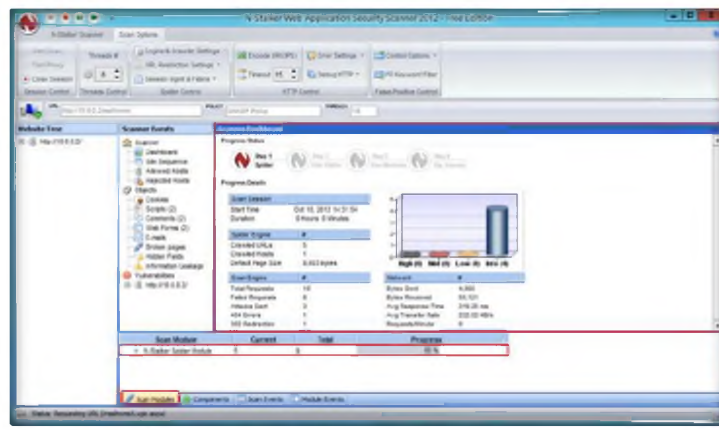


FIGURE 4.14: N-Stalker Start Scan Status

16. N-Stalker will scan the site with four different methods.

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

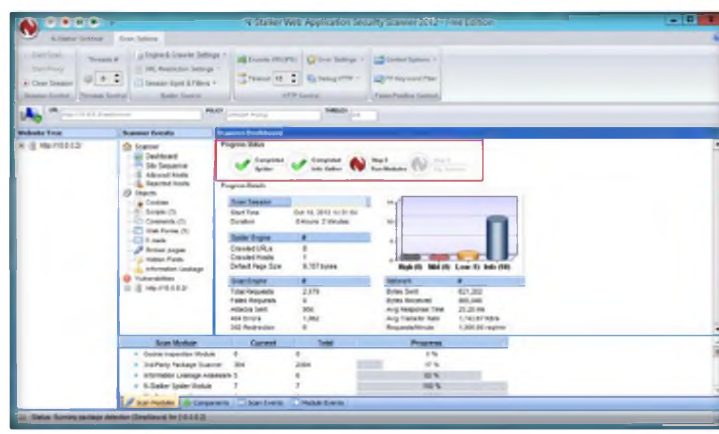


FIGURE 4.15: N-Stalker Scanning methods

17. In the left pane, the **Website** tree displays the pages of the website.

"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.

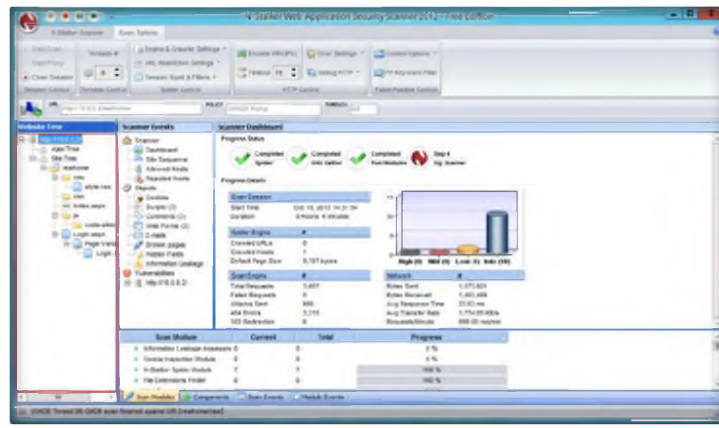


FIGURE 4.16: N-Stalker Website Tree

18. In **Results Wizard**, select the relevant options as shown in the following screenshot and click **Next**.

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

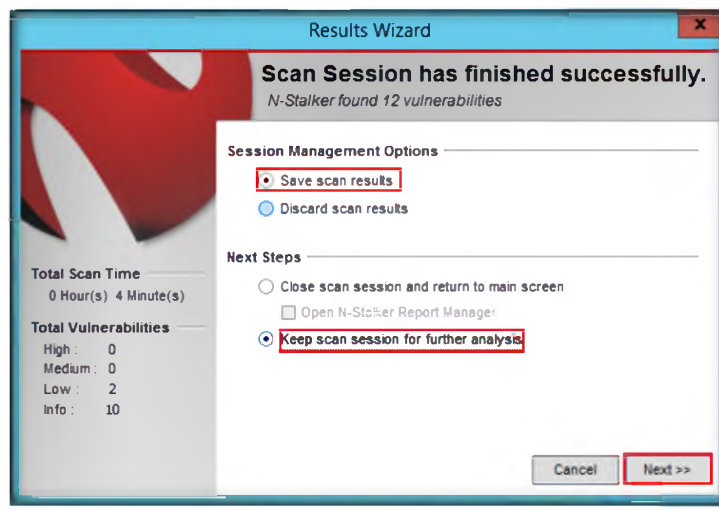


FIGURE 4.17: N-Stalker Results Wizard

19. N-Stalker displays the summary of vulnerabilities. Click **Done**.

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.

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

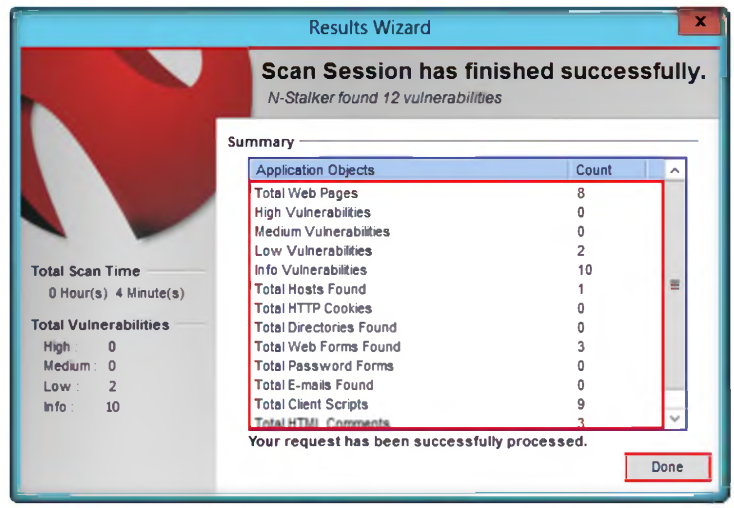


FIGURE 4.18: N-Stalker Summary

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.

20. You can view the complete scan results of the URL in the main dashboard of the **N-Stalker**.

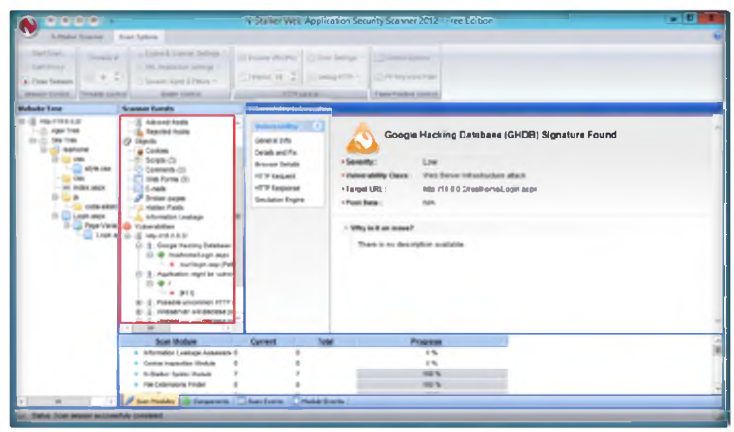


FIGURE 4.19: N-Stalker Dashboard

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 the lab exercise. Give your opinion on your target's security posture and exposure.

Tool/Utility	Information Collected/Objectives Achieved
N-Stalker	Scan session successfully processed with 12 vulnerabilities detected

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

Questions

1. Analyze how to speed up the scanning process and reduce the number of pages the IBM Rational AppScan finds.
2. Evaluate whether it is possible to perform scans against live production environments with IBM Rational AppScan. Will that cause damage or hurt the site?
3. Analyze how variables can be implemented in a multi-step sequence with IBM Rational AppScan.

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