

Session Hijacking

Module 11

Hijacking Sessions

Session hijacking refers to the exploitation of a valid computer session, wherein an attacker takes over a session between two computers.

ICON KEY

 Valuable information

 Test your knowledge

 Web exercise

 Workbook review

Lab Scenario

Source: <http://krebsonsecurity.com/2012/11/yahoo-email-stealing-exploit-fetches-700>

According to KrebsOnSecurity news and investigation, zero-day vulnerability in yahoo.com that lets attackers hijack Yahoo! email accounts and redirect users to malicious websites offers a fascinating glimpse into the underground market for large-scale exploits.

The exploit, being sold for \$700 by an Egyptian hacker on an exclusive cybercrime forum, targets a “cross-site scripting” (XSS) weakness in yahoo.com that lets attackers steal cookies from Yahoo! webmail users. Such a flaw would let attackers send or read email from the victim’s account. In a typical XSS attack, an attacker sends a malicious link to an unsuspecting user; if the user clicks the link, the script is executed, and can access cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

KrebsOnSecurity.com alerted Yahoo! to the vulnerability, and the company says it is responding to the issue. Ramses Martinez, director of security at Yahoo!, said the challenge now is working out the exact yahoo.com URL that triggers the exploit, which is difficult to discern from watching the video.

These types of vulnerabilities are a good reminder to be especially cautious about clicking links in emails from strangers or in messages that you were not expecting.

Being an administrator you should implement security measures at Application level and Network level to protect your network from session hijacking. Network level hijacks is prevented by packet encryption which can be obtained by using protocols such as IPSEC, SSL, SSH, etc. IPSEC allows encryption of packets on shared key between the two systems involved in communication.


Application-level security is obtained by using strong session ID. SSL and SSH also provides strong encryption using SSL certificates to prevent session hijacking.

Lab Objectives

The objective of this lab is to help students learn session hijacking and take necessary actions to defend against session hijacking.

In this lab, you will:

- Intercept and modify web traffic

 **Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv8 Module 11 Session Hijacking**

- Simulate a Trojan, which modifies a workstation's proxy server settings

Lab Environment

To carry out this, you need:

- A computer running **Windows Server 2012 as host machine**
- This lab will run on **Windows 8** virtual machine
- Web browser with Internet access
- Administrative privileges to configure settings and run tools

Lab Duration

Time: 20 Minutes

Overview of Session Hijacking

Session hijacking refers to the **exploitation** of a valid computer session where an attacker **takes over** a session between two computers. The attacker **steals** a valid session ID, which is used to get into the system and **sniff** the data.

In **TCP session** hijacking, an attacker takes over a TCP session between two machines. Since most **authentications** occur only at the start of a TCP session, this allows the attacker to **gain access** to a machine.

Lab Tasks

Pick an organization that you feel is worthy of your attention. This could be an educational institution, a commercial company, or perhaps a nonprofit charity.

Recommended labs to assist you in session hijacking:

- Session hijacking using **ZAP**

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.

 **TASK 1**
Overview




Session Hijacking Using Zed Attack Proxy (ZAP)

The OWASP Zed Attack Proxy (ZAP) is an easy-to-use integrated penetration testing tool for finding vulnerabilities in web applications.

ICON KEY

 Valuable information

 Test your knowledge

 Web exercise

 Workbook review

Lab Scenario

Attackers are continuously watching for websites to hack and developers must be prepared to counter-attack malicious hackers by writing strong secure codes. A common form of attack is session hijacking, i.e., accessing a website using someone else's session ID. A session ID might contain credit card details, passwords, and other sensitive information that can be misused by a hacker.

Session hijacking attacks are performed either by session ID guessing or by stolen session ID cookies. Session ID guessing involves gathering a sample of session IDs and “guessing” a valid session ID assigned to someone else. It is always recommended not to replace ASP.NET session IDs with IDs of your own, as this will prevent session ID guessing. Stolen session ID cookies session hijacking attack can be prevent by using SSL; however, using cross-site scripting attacks and other methods, attackers can steal the session ID cookies. If an attacker gets ahold of a valid session ID, then ASP.NET connects to the corresponding session with no further authentication.

There are many tools easily available now that attackers use to hack into websites or user details. One of the tools is Firesheep, which is an add-on for Firefox. While you are connected to an unsecure wireless network, this Firefox add-on can sniff the network traffic and capture all your information and provide it to the hacker in the same network. The attacker can now use this information and login as you.

As an **ethical hacker**, penetration tester, or **security administrator**, you should be familiar with network and web authentication mechanisms. In your role of web security administrator, you need to test web server traffic for **weak session IDs**, insecure handling, **identity theft**, and **information loss**. Always ensure that you have an encrypted connection using https which will make the sniffing of network packets difficult for an attacker. Alternatively, VPN


connections too can be used to stay safe and advise users to log off once they are done with their work. In this lab you will learn to use ZAP proxy to intercept proxies, scanning, etc.

Lab Objectives

The objective of this lab is to help students learn session hijacking and how to take necessary actions to defend against session hijacking.

In this lab, you will:

- Intercept and modify web traffic
- Simulate a Trojan, which modifies a workstation's proxy server settings

 **Tools**
demonstrated in
this lab are
available in
D:\CEH-
Tools\CEHv8
Module 11
Session Hijacking

Lab Environment

To carry out the lab, you need:

- **Paros Proxy** located at **D:\CEH-Tools\CEHv8 Module 11 Session Hijacking\Session Hijacking Tools\Zaproxy**
- You can also download the latest version of **ZAP** from the link <http://code.google.com/p/zaproxy/downloads/list>
- If you decide to download the **latest version**, then screenshots shown in the lab might differ
- A system with running Windows Server 2012 Host Machine
- Run this tool in **Windows 8** Virtual Machine
- A web browser with Internet access
- Administrative privileges to configure settings and run tools
- Ensure that **Java Run Time Environment (JRE) 7** (or above) is installed. If not, go to <http://java.sun.com/j2se> to download and install it.

Lab Duration

Time: 20 Minutes

Overview of Zed Attack Proxy (ZAP)

Zed Attack Proxy (ZAP) is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing as well as being a useful addition to an experienced pen tester's toolbox. Its features include intercepting proxy, automated scanner, passive scanner, and spider.

Lab Tasks

1. Log in to your **Windows 8** Virtual Machine.

 **TASK 1**

Setting-up ZAP

2. In **Windows 8** Virtual Machine, follow the wizard-driven installation steps to install **ZAP**.
3. To launch **ZAP** after installation, move your mouse cursor to the lower-left corner of your desktop and click **Start**.

You can also download ZAP <http://code.google.com/p/zaproxy/downloads/list>



FIGURE 2.1: Paros proxy main window

At its heart ZAPS is an intercepting proxy. You need to configure your browser to connect to the web application you wish to test through ZAP. If required you can also configure ZAP to connect through another proxy - this is often necessary in a corporate environment.

4. Click **ZAP 1.4.1** in the **Start** menu apps.

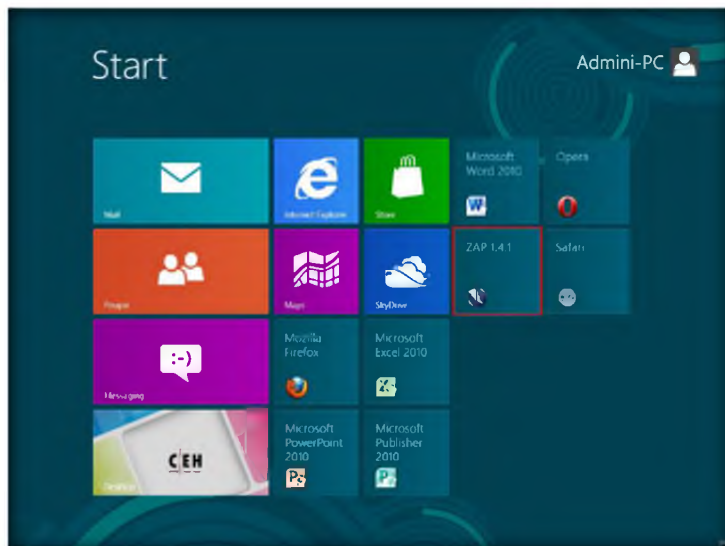


FIGURE 2.2: Paros proxy main window

If you know how to set up proxies in your web browser then go ahead and give it a go!

If you are unsure then have a look at the Configuring proxies section.

5. The main interface of **ZAP** appears, as shown in the following screenshot.
6. It will prompt you with **SSL Root CA certificate**. Click **Generate** to continue.

Once you have configured ZAP as your browser's proxy then try to connect to the web application you will be testing. If you can not connect to it then check your proxy settings again. You will need to check your browser's proxy settings, and also ZAP's proxy settings.

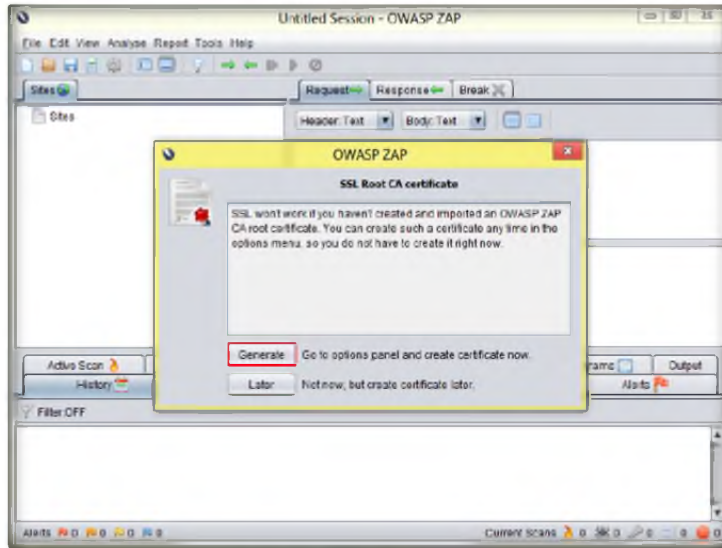


FIGURE 2.3: Paros proxy main window

- In the **Options** window, select **Dynamic SSL certificates** then click **Generate** to generate a certificate. Then click **Save**.

Active scanning attempts to find potential vulnerabilities by using known attacks against the selected targets.

Active scanning is an attack on those targets. You should NOT use it on web applications that you do not own.

It should be noted that active scanning can only find certain types of vulnerabilities. Logical vulnerabilities, such as broken access control, will not be found by any active or automated vulnerability scanning. Manual penetration testing should always be performed in addition to active scanning to find all types of vulnerabilities.

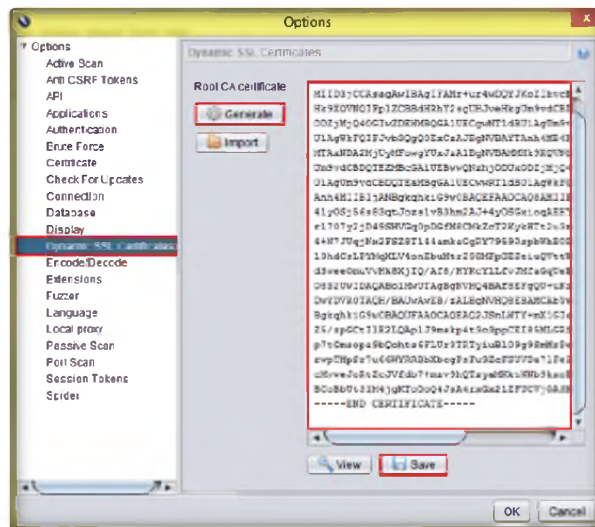



FIGURE 2.4: Paros proxy main window

- Save** the certificate in the default location of **ZAP**. If the certificate already exists, replace it with the new one.

Module 11 – Session Hijacking

 An alert is a potential vulnerability and is associated with a specific request. A request can have more than one alert.

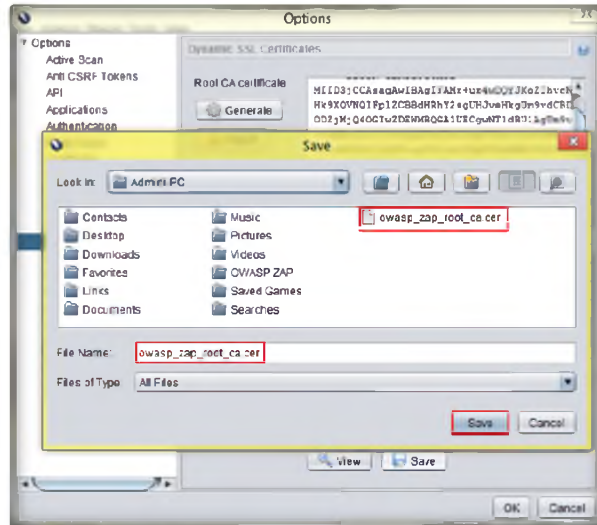



FIGURE 2.5: Paros proxy main window

9. Click **OK** in the **Options** window.

 Anti CSRF tokens are (pseudo) random parameters used to protect against Cross Site Request Forgery (CSRF) attacks.

However they also make a penetration testers job harder, especially if the tokens are regenerated every time a form is requested.

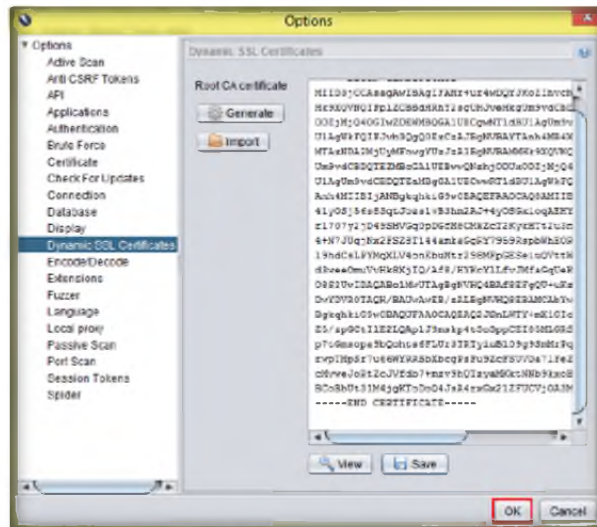


FIGURE 2.6: Paros proxy main window

10. Your Paros proxy server is now ready to intercept requests.

Module 11 – Session Hijacking

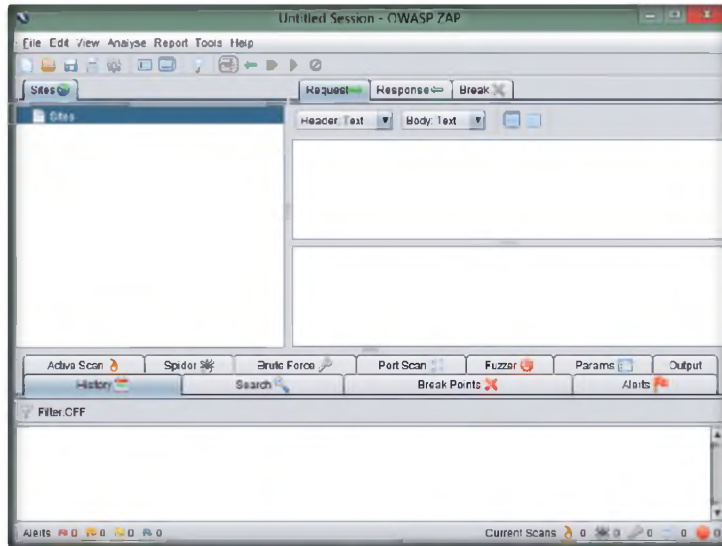



FIGURE 2.7: Paros proxy main window

 ZAP detects anti CSRF tokens purely by attribute names - the list of attribute names considered to be anti CSRF tokens is configured using the Options Anti CSRF screen. When ZAP detects these tokens it records the token value and which URL generated the token.

11. Launch any web browser, in this lab we are using the **Chrome** browser.
12. Your VM workstation should have **Chrome version 22.0** or later installed.
13. Change the **Proxy Server settings** in Chrome, by clicking the **Customize and control Google Chrome** button, and then click **Settings**.

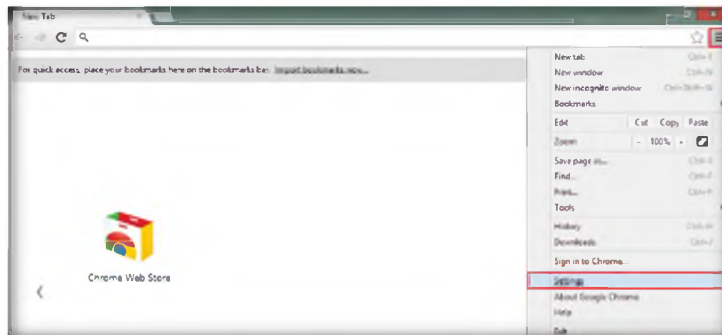



FIGURE 2.8: IE Internet Options window

 ZAP provides an Application Programming Interface (API) which allows you to interact with ZAP programmatically.

The API is available in JSON, HTML and XML formats. The API documentation is available via the URL <http://zap/> when you are proxying via ZAP.

14. On the Google Chrome Settings page, click the **Show advanced settings...** link bottom of the page, and then click the **Change proxy settings...** button.

Module 11 – Session Hijacking

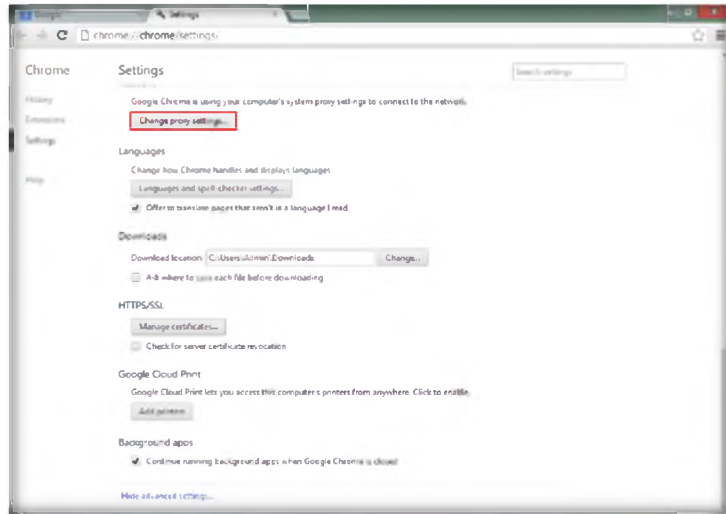


FIGURE 2.9: Paros proxy main window

15. In **Internet Properties** wizard, click **Connections** and click **LAN Settings**.

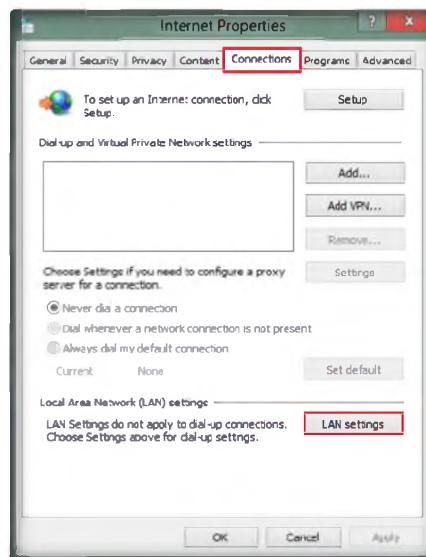



FIGURE 2.10: IE Internet Options window with Connections tab

16. Check **Use a proxy server for your LAN**, type **127.0.0.1** in the **Address**, enter **8080** in the **Port** field, and click **OK**.

 Click OK several times until all configuration dialog boxes are closed.

It should be noted that there is minimal security built into the API, which is why it is disabled by default. If enabled then the API is available to all machines that are able to use ZAP as a proxy. By default ZAP listens only on 'localhost' and so can only be used from the host machine.

The API provides access to the core ZAP features such as the active scanner and spider. Future versions of ZAP will increase the functionality available via the API.

Module 11 – Session Hijacking

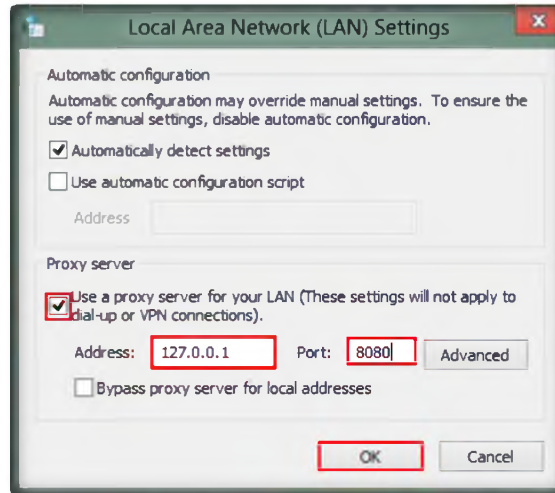


FIGURE 2.11: IE Internet Options Window with Proxy Settings Window

TASK 2

Hijacking Victim's Session

ZAP allows you to try to brute force directories and files.

A set of files are provided which contain a large number of file and directory names.

A break point allows you to intercept a request from your browser and to change it before it is submitted to the web application you are testing. You can also change the responses received from the application. The request or response will be displayed in the Break tab which allows you to change disabled or hidden fields, and will allow you to bypass client side validation (often enforced using javascript). It is an essential penetration testing technique.

17. Click **Set break on all requests** and **Set break on all responses** to trap all the requests and responses from the browser.

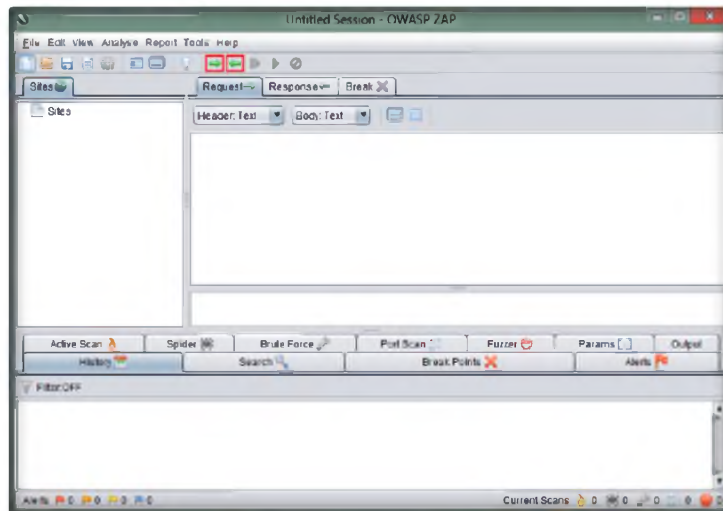



FIGURE 2.12: Paros proxy main window

18. Now navigate to a chrome browser, and open www.bing.com.
19. Start a search for “Cars.”
20. Open ZAP, which shows first trapped incoming web traffic.
21. Observe the first few lines of the trapped traffic in the **trap** windows, and keep clicking **Submit and step to next request or response** until you see cars in the **GET** request in the **Break** tab, as shown in the following screenshot.

Module 11 – Session Hijacking

 Filters add extra features that can be applied to every request and response. By default no filters are initially enabled. Enabling all of the filters may slow down the proxy. Future versions of the ZAP User Guide will document the default filters in detail.

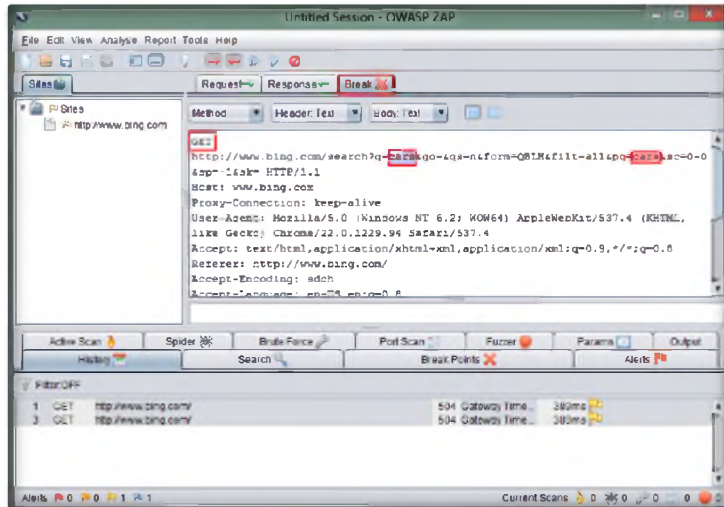
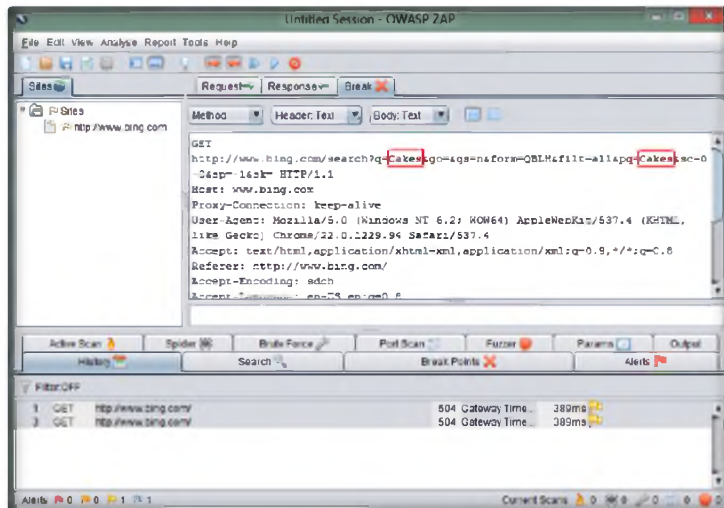



FIGURE 2.6: Paros Proxy with Trap option content


22. Now change the query text from **Cars** to **Cakes** in the GET request.



 Fuzzing is configured using the Options Fuzzing screen. Additional fuzzing files can be added via this screen or can be put manually into the "fuzzers" directory where ZAP was installed - they will then become available after restarting ZAP.

23. Click **Submit** and step to next request or response.

24. Search for a title in the **Response** pane and replace **Cakes** with **Cars** as shown in following figure.

 The request or response will be displayed in the Break tab which allows you to change disabled or hidden fields, and will allow you to bypass client side validation (often enforced using javascript). It is an essential penetration testing technique.

Module 11 – Session Hijacking

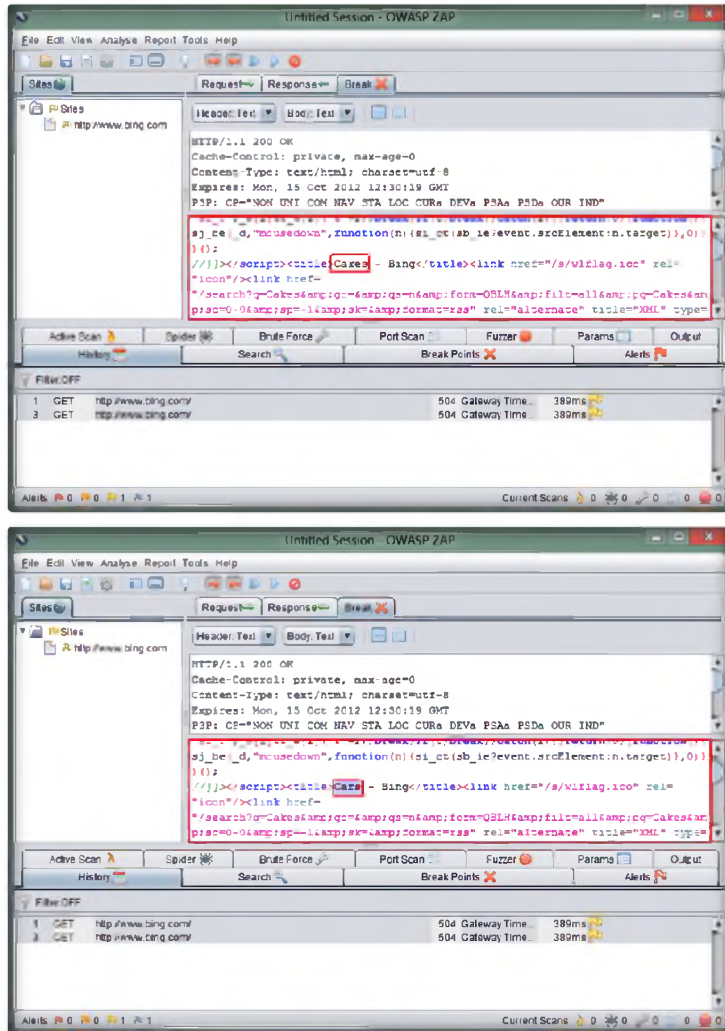

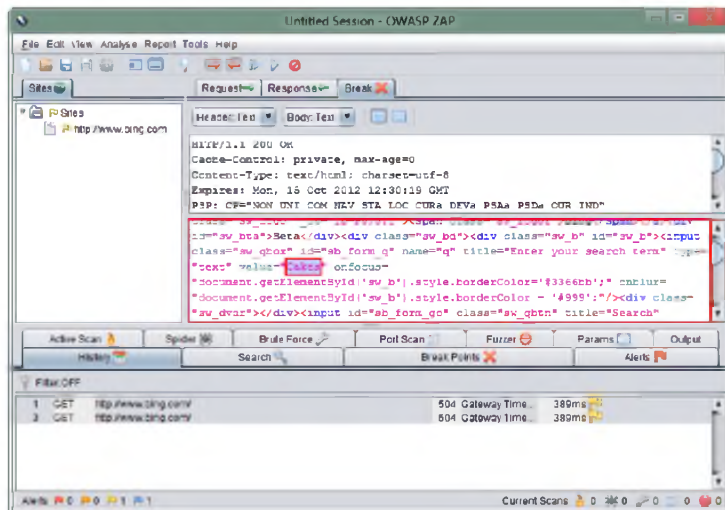



FIGURE 2.7: Paros Proxy search string content

25. In the same **Response** pane, replace **Cakes** with **Cars** as shown in the following figure at the value shown.

 This functionality is based on code from the OWASP JBroFuzz project and includes files from the fuzzdb project. Note that some fuzzdb files have been left out as they cause common anti virus scanners to flag them as containing viruses. You can replace them (and upgrade fuzzdb) by downloading the latest version of fuzzdb and expanding it in the 'fuzzers' library.



Module 11 – Session Hijacking

 This tool keeps track of the existing Http Sessions on a particular Site and allows the Zaproxy user to force all requests to be on a particular session.

Basically, it allows the user to easily switch between user sessions on a Site and to create a new Session without "destroying" the existing ones.

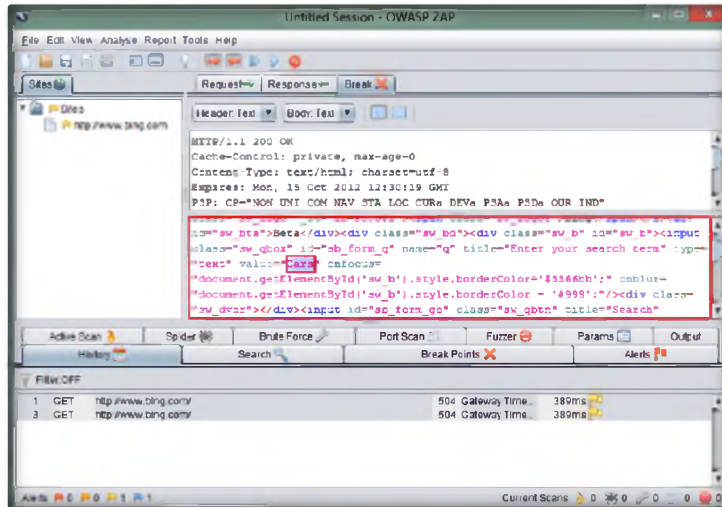



FIGURE 2.8: Paros with modified trap option content

Note: Here we are changing the text Cakes to Cars; the bing search shows Cars, whereas the results displayed are for Cakes.

26. Observe the **Bing search** web page displayed in the browser with search query as “**Cakes.**”

 It is based on the concept of Session Tokens, which are HTTP message parameters (for now only Cookies) which allow an HTTP server to connect a request message with any previous requests or data stored. In the case of Zaproxy, conceptually, session tokens have been classified into 2 categories: default session tokens and site session tokens. The default session tokens are the ones that the user can set in the Options Screen and are tokens that are, by default, automatically considered session tokens for any site (eg. phpsessid, jsessionid, etc). The site session tokens are a set of tokens for a particular site and are usually set up using the popup menus available in the Params Tab.

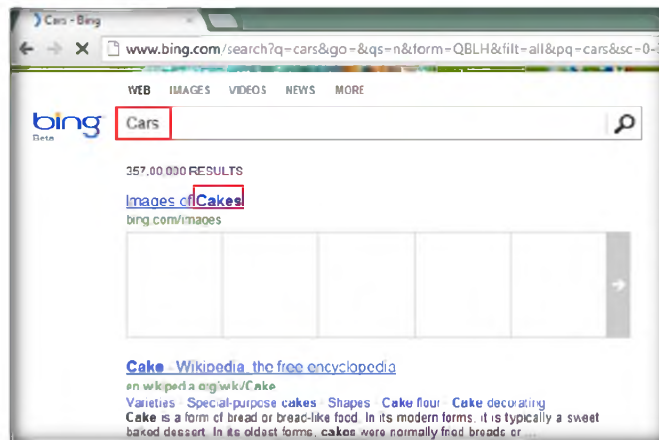


FIGURE 2.6: Search results window after modifying the content

27. That's it. You just forced an unsuspecting web browser to go to any page of your choosing.

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
Zed Attack Proxy	<ul style="list-style-type: none"> ▪ SSL certificate to hack into a website ▪ Redirecting the request made in Bing

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

Questions

1. Evaluate each of the following Paros proxy options:
 - a. Trap Request
 - b. Trap Response
 - c. Continue Button
 - d. Drop Button

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