Secure Web Development Teaching Modules[1]

Security Testing

Contents

1 Concepts

1.1 Security Practices for Software Verification

Security practices in OpenSAMM[2] are incorporated in four stages of software development: governance, construction, verification, and deployment. This document will introduce the security practices to be performed in the verification stage and provide laboratory exercises illustrating some of the practices. Based on OpenSAMM, verification is focused on

"*the processes and activities related to how an organization checks and tests artifacts produced throughout software development. This typically includes quality assurance work such as testing, but it can also include other review and evaluation activities.*"

[1] Copyright© 2009-2011 Li-Chiou Chen (lchen@pace.edu) & Lixin Tao (ltao@pace.edu), Pace University. This document is based upon work supported by the National Science Foundation's Course Curriculum, and Laboratory Improvement (CCLI) program under Grant No. 0837549. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation. A copy of the license is available at http://www.gnu.org/copyleft/fdl.html.

[2] The Open Web Application Project (OWASP), Software Assurance Maturity Model, Version 1.0, released March 25, 2009, , available at http://www.opensamm.org/.

The purpose of verification is to inspect the software for security vulnerabilities before it is deployed. The security practices for this stage include design review, code review and security testing. All three practices are used to identify specific security vulnerabilities embedded in the software but each examines the software at a different level respectively. Design review assesses the design and architecture of the software while code review examines the source codes and security testing inspects the software at the runtime environment. Design review includes reviewing the software design for potential security risks and attacks, examining security mechanisms embedded in the design, and reviewing data-flow diagrams to protect sensitive information assets. Code review includes reviewing source codes against a list of security requirements, applying automatic code review tools and customizing code review for application specific concerns. Security testing include testing software security using test cases derived from known security requirements, conducting penetration testing and utilizing automatic tools for testing application specific concerns. We will focus on the activities and tools that can be used to conduct these security testing.

1.2 Software Security Testing

Security testing assesses a software system from attackers' perspective. It evaluates if your software performs as it is not supposed to do and if its security mechanism works as it is supposed to be. For example, when testing the security of a web server, the tester needs to evaluate the security mechanisms utilized by the server, such as HTTPS for server authentication and login method for client authentication, and to assess if the server is subject to vulnerabilities such SQL injection or cross site scripting.

The most commonly used security testing method is penetration testing, which discovers software vulnerabilities by breaking into it. In this case, testers assume the role of attackers to create testing scenarios based on their knowledge to break into the software system using customized scripts or automatic tools. The effectiveness of penetration testing in terms of uncovering vulnerabilities depends on the experience of the testers since the development of the testing scenarios could be objective.

Fault injection and fuzzing are two common methods for conducting penetration testing. Both fault injection and fuzzing utilize unexpected inputs to observe how a software system would perform. While fault injection specifically crafts malformed inputs based pre-defined threat models or error handling paths, fuzzing enumerates all possible inputs, either valid or invalid. These two terms sometimes are used interchangeably in the industry.

Web application security testing involves examining the security of HTTP and HTTPS protocols, reviewing input validations and error handling, and evaluating authentication methods and the vulnerabilities of web protocols used, such as XML related protocols. OWASP's Application Security Verification Standard[3] provides a guideline for putting web application verification in place. There are various automatic tools to conduct web application security testing, such as SPIKE Proxy[4], WebScarab[5], Paros[6], BurpIntruder[7], etc.

2 Labs Objectives

From this lab, you will learn about

[3] The standard is available at http://www.owasp.org/index.php/OWASP_Application_Security_Verification_Standard_%28ASVS%29.

[4] Available at http://www.immunitysec.com/resources-freesoftware.shtml.

[5] Available at http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project.

[6] Available at http://www.parosproxy.org/.

[7] Available at http://www.portswigger.net/intruder/.

- How to discover vulnerabilities, such as authentication weakness or hidden pages, on a web server
- Investigate fixes to these vulnerabilities

3 Lab Setup

1. You will use the *ubuntu 10* virtual machine for SWEET teaching modules.
2. Extract the virtual machine from *ubuntu10tm.exe*.
3. Under the folder *ubuntu10tm,* double click on *ubuntu10tm.vmx* to start the virtual machine.
4. The username is “user” and the password is “123456”.

4 Lab Guide

4.1 Virtual Machine Startup

Step1: After logging in, you will see the Ubuntu desktop interface. The virtual machine runs Linux as an independent virtual computer separate from the host operating system (i.g. Windows).

Step2: Please spend a little time to familiarize yourself with the Linux interface and try the following.

1) Swap back and forth between your Linux virtual machine and the host machine (i.g. Windows). The Linux is run within its own VMware window.

2) Explore the menu bar of the Linux GUI on top of the VM window. The menu bar includes Applications (similar to Windows Start Panel), Places (all devices and storage), and System (Linux system functions).

3) To copy a file from your host machine to the VM, you can drag and drop the file between the two platforms or vice versa.

Step 3: From the menu bar on the top of the VM, select Applications > Accessories > Terminal. This will open up a Linux shell command terminal, execute the command **ifconfig**

Step 4: You will receive several lines of output. You are going to look for the Ethernet interface **(i.g. eth0)**. Find the **inet addr**: field and write down the IP address in the space below.

4.2 *Setup the Proxy Server*

We will investigate the web traffic between your browser and the BadStore.net using a web proxy called Paros on a same virtual machine. All web communication between the browser and the Web server will be sent to Paros (the proxy server) first before it reaches the appropriate destination. We will be browsing BadStore.net or investigate its vulnerabilities.

Step 1: Make sure Tomcat server is not running, otherwise go to Select Applications > Accessories > Terminal. Run following command in the terminal window to shut down tomcat

```
tomcat-stop
```

Step 2: Open a Firefox browser to browse BadStore.net by typing the IP address of the VM in Exercise I in the URL e.g **http:// IP ADDRESS/badstore** (DO NOT browse www.badstore.net directly since it will redirect you to original website if you have an Internet connection).

Step3: We will then use Paros to investigate the web traffic. To start Paros, run these commands in the terminal window:

```
cd ~/tools/paros
sh startserver.sh
```

The Java-based Paros will execute and you will be greeted with its interface.

Step4: Now, you will need to change the proxy server settings in Firefox to redirect the web traffic to the proxy server. The proxy server is run under localhost (127.0.0.1) and port 8088.

1) Go back to your browser. Select Edit > Preferences > Advanced > Network Tab > Settings.
2) Select the Manual Proxy Configuration radio button.
3) Enter these values into the fields: HTTP: 127.0.0.1 Port: 8088
4) If there are any values in the No Proxy For: text field, delete them. This is important to make the proxy work successfully.

Step5: Refresh your browser. You should see that Paros has intercepted the transactions between the browser and the Badstore site. You have just enabled all HTTP traffic generated by Firefox to be sent to the running Paros proxy server which can analyze HTTP traffic before it is sent off to its final destination.

4.3 Crawling Web Pages and Hidden Web Directories

In order for an attacker to successfully plan and execute an attack, the attacker must know the website's layout and all the pages that might be available for exploitation. While manual web crawling is an option, it is a very time consuming process. An automated web crawler application will speed up the mapping process significantly.

1. You will use Paros to crawl the BadStore website IP address so you can see what pages are available.

2. Switch to the Paros application and click **File > New Session** and click **OK** to have Paros start a new session and purge itself from any logged content.

3. In the Paros menu toolbar, navigate to **Tools > Options** and select the **Spider** option. You will change the **Maximum Depth to Crawl** from its default value to the maximum value of **9**. This will allow Paros to crawl web pages that may be deeply nested in BadStore.net Click **OK** to confirm and return to the main screen.

4. In the **Sites** panel on the left will be all the websites that Paros is logging. It is currently blank, change to the Firefox application and refresh BadStore web page (You may have to clear recent browsing history first to reload the page. **Tools > Clear Recent History**)

5. Switch back to Paros and you will see an arrow next to **Sites** that is point to the right. Click the arrow to un-collapse the logged websites. You will see the IP address of BadStore website.

6. Select the IP address for BadStore under **Sites** in Paros. The IP address will be highlighted in brown and go to **Analyze > Spider** in the Paros toolbar menu.

   A Spider window will open. In the **URL crawling:** field should be the IP address of BadStore.net. If all checks out, click **Start**.

7. Once crawling has begun, the main Paros window will begin to populate with web pages and images that are hosted within BadStore.net.

   In the bottom pane of the main screen, you will see a **URL found during crawl:** panel. Notice that it is located under the **Spider** tab near the bottom.

8. Looking through the entries, you will notice that most the web cgi pages are located in the **/cgi-bin/** directory but some are not. List one other directory that Paros had crawled and one file under this directory

   Directory name:___________________

   File name: ___________________

9. Briefly explain what information one might obtain by crawling a web site.

____

10. What is the potential risk for a web site being crawled?

    ____

4.4 Scanning For Known Vulnerabilities

In the previous exercise, you have mapped BadStore.net; in this exercise you will execute a vulnerability scan on BadStore.net.

1. In the Paros Sites panel, Click on the IP address of BadStore, highlighted in brown.

2. In the Paros toolbar menu, navigate to **Analyze > Scan**. The vulnerability scan will begin. Give it a minute to complete the scan.

3. Once the scan is finished, the results can be viewed on the bottom panel of the Paros application under the **Alerts** tab. If you would like to have an actual report, it is located in the **/user/paros/session** folder and it is called **LatestScannedReport.html.** Open it in a browser. (Click on the menu bar on the top, click on **Places** to access **Home Folder**)

4. The vulnerabilities are called alerts and are classified as High, Low, and Medium. List two vulnerabilities from the report and explain the countermeasures to fix them.

11. Vulnerability 1: ____

    Countermeasure 1: ____

    Vulnerability 2: ____

    Countermeasure 2: ____

5. Not all web crawlers and web vulnerability scanners are as robust. Commercial web crawlers and vulnerability scanners may perform a much more complete crawl and may list more potential vulnerabilities.

4.5 Accessing More Hidden Pages

Looking at the BadStore.net webpage structure you will find nothing that really stands out. This is because there is a hidden webpage which provides the administrative console. Web crawlers will only crawl the pages that are available through links. The process of finding these hidden web pages is called Forced Browsing and there are special tools to perform this task automatically. The process can also be done manually but can be very time consuming. For the sake of time, the hidden administrative webpage is provided to you in this exercise.

1. The hidden webpage reside under the **cgi-bin** directory of BadStore.net. For an attacker to be able to discover this hidden webpage he/she would need to understand URL structure and syntax and all the technologies that comprise a web application.

2. Since most of the links on the left panel of BadStore.net have the same URL prefix, the only part of the URL that is changed is what follows after **action=**

3. What do you think happens when you append **admin** to **action=** ?

   If you are browsing Badstore.com on a different VM, browse
   http:// IP ADDRESS /cgi-bin/badstore.cgi?action=admin

4. What is the result of this URL ?

5. List 3 actions that administrators can take.

   __a.____
   __b.____
   __c.____

6. Select an action and click **Do It**. What happens?

   ____

7. Somehow we must gain administrative privileges to perform these actions. The ability to perform these actions is very lucrative to attackers. The ability to gain administrative privileges will be demonstrated in the next exercise.

4.6 Privilege Elevation via Parameter Tampering

In this exercise you will examine the vulnerability present with parameters that are set and sent from the client to the server. This vulnerability was not present in the Paros vulnerability assessment. You will capture a server request that is sent when you register for an account. You will modify that hidden parameter that is contained in the request to escalate your account from the users privilege group to the administrators privilege group.

1. With Firefox on the BadStore.net webpage, visit the **Login/Register** link in the left pane.

2. You are going to register for a new account, but before you register you are going to switch to Paros and configure the application proxy to capture the register HTTP request as it leaves the browser so that you are able to manipulate the request's parameters before it continues to the BadStore.net server.

3. In Paros, click on the **Trap** tab and select the **Trap request** checkbox. This will trap any request that is sent from the Firefox browser.

4. Go back to the BadStore.net website and start filling out the registration form to **Register for a New Account**. Write down your registration information below for reference.

   Full Name: ____

   Email Address: ____

   Password: ____

5. Once every field is filled, click **Register** and switch back to Paros.

6. Once you switch back to Paros, clicking on the **Trap** tab, you will see the register HTTP request that was intercepted by Paros. There will be a hidden Parameter Name "role". Its current value is "U", which refers to regular user privilege. Replace "U" with "A" and uncheck the **Trap request** checkbox. Hit **Continue to** let the HTTP request reach the server.

   (**Note**: if you put the view in **tabular view**, and change the role to "**A**" it will not register, but if you leave it in **Raw View** and just scroll down to the role section and change the "**U**" to an "**A**" it works.)

7. Switch back to the BadStore.net website. Everything looks the same but the Welcome banner, which welcomes whatever account you had created. So how do you use your newly escalated administrative rights? Go back to the hidden administrative page.

4.7 Compromising Passwords

You have access to all the administrative abilities listed on the secret administration page. In this exercise you will concentrate on the **Show Current Users** ability. You will be compromising user's passwords to create a backdoor.

1. Having your account privileges escalated, visit the hidden administrator's web page.

   http:// IP ADDRESS /cgi-bin/badstore.cgi?action=admin

2. From the drop down menu, select **Show Current Users** and click **Do It.**

3. You have just hit the jackpot. You will see all the registered user's email addresses, hashed passwords, password hints, full names and roles.

4. You want to get the account passwords for all the other administrative users since regular users (role U) and suppliers (role S) are trivial.

5. List the password hash of the administrator.

   Password Hash: ____

6. The passwords are hashed using the MD5 algorithm. MD5 is known as a weak hash function than can be easily cracked using the hash databases. There are many freeware programs that will try MD5 hashes of common words and compare them to the MD5 hash that you provide in order to find a match. Once you find a match you know that you have cracked the password.

7. There is an online MD5 cracker that has the world's largest MD5 hash database. Visit http://www.cmd5.org/default.aspx if you have an Internet connection. Otherwise, use the table below to crack the MD5 hashes that you have recorded above.

Password	**MD5 hash**
welcome	40be4e59b9a2a2b5dffb918c0e86b3d7
admin123	0192023a7bbd73250516f069df18b500
apple	1f3870be274f6c49b3e31a0c6728957f
Welcome	83218ac34c1834c26781fe4bde918ee4
badpass123	909d49a643874753d1a68ea87f379925

8. What is the password? ________________

9. Go back to **Login/Regoster** page on Badstore site. Use "admin" as the email address and the password you discovered before to login. You should be able to login as the "Master System Administrator".

10. Explain briefly how the MD5 cracker works in order to crack the above password.

    ____

11. Explain briefly the vulnerability of the web server which you have just exploited in this exercise.

    ____

12. Explain briefly how you exploit the vulnerability in this exercise.

____

13. Describe a countermeasure to fix the vulnerability.

____

4.8 SQL injection or XSS

1. Describe a method to exploit Badstore.net using SQL injection or XSS vulnerability.

____

2. Describe a method to fix the SQL injection or XSS vulnerability you identified above.

____

4.9 Turn Off Virtual Machines

1. After finishing this exercise, you should reset *Firefox* proxy setting so it stops using the proxy server. Otherwise you would not be able to visit web sites without running the proxy server at port 8088. To do so, Launch your Firefox web browser, and follow its menu item path "Edit|Preferences|Advanced|network Tab|Settings button" to reach the "Connection Settings" window. Check the "Use System Proxy Settings" checkbox.

2. Close Paros (**File|Exit**). Close **Terminal** Windows (type "**exit**" under command line.)

3. Click on the power button on the VM and turn it off.