9/10/2012 **CSE 565: Computer Security** **Due: 10/01/2012**

Fall 2012
Project 1 – Vulnerability Analysis and Mitigation

Blame it on the WebGoat!

1. Background

The primary goal of this project is to understand web application security. We will use the free and openly available tools under The Open Web Application Security Project (OWASP) to learn about the injection flaws that exist in today's web environment. Additionally, we will learn about the best practices and countermeasures to protect web applications against such exploits.

Students are expected to understand web security concepts and practice injection attacks using WebGoat.

2. Project Details

This project will require a lot of reading and answering specific questions, some tool installations and running tutorials and testing exploits. The project is divided into several sections for clarity. You are required to answer all the questions shown in red and *italics*. For more details, read Section 3 below.

2.1 Web security

Vulnerability: "A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy" [IETF RFC 2828].

Example: Access control refers to defining and restricting user access to the various system resources. Ideally, administrators have complete control over the system whereas guest users have restricted access (lack the ability to make major configuration changes). However, poorly implemented access control can make the system vulnerable to privilege escalations. Like, if guest users can change system configurations, they can declare themselves the administrator and control the system.

Recommended reading: https://www.owasp.org/index.php/Category:Vulnerability

Application security: This topic deals with the process of ensuring the security of an application or its underlying system against possible vulnerabilities. Best application security measures are not the afterthoughts but a part of application development life-cycle (design, development, testing, etc.)

Example: Ensure that the access control for the various application users is well defined and cannot be overwritten.

Recommended reading: https://www.owasp.org/index.php/Top_10_2010-Main

Web application security: It deals with the security of websites, web applications and web servers. Web applications are a particular target because of their easy accessibility.

Example: Ensure that the access control for the web applications/websites/web servers is well defined and cannot be overwritten.

2.2 Injection attacks
Injection flaws allow attackers to relay malicious code through a web application to another system.

Recommended reading: https://www.owasp.org/index.php/Injection_Flaws

SQL injection: This is one of the most widespread forms of injection attacks. It exploits faulty application code between the web application and the database servers to execute malicious database queries. Many web applications use web forms to capture user inputs and then use this input to construct SQL queries. When the user input from these web-forms is not properly validated, (un)intentionally bad input seeps into the SQL queries and corrupts the database or violates the security policies.

Recommended reading: https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet

Q. List any four malicious objectives that an attacker can achieve using SQL injection attack? [2 points]

Q. Can SQL attack enable an attacker to issue commands to a database? Can SQL attack enable an attacker to issue commands to an operating system? [2 point]

Q. Briefly answer the following questions about SQL injection. [8 points]

We have a web application with a form that reads the lastname of a user and searches the database for his firstname. The SQL query used to achieve this is:

SELECT firstname FROM employees where lastname= '$LASTNAME';
$LASTNAME is a parameter that is read from the web-form and is used in this SQL query.

- *Can you see a full list of all the employees in the database? Explain your answer and write down the SQL queries involved, if any.*

- *Can you add a new employee to the database? Explain your answer and write down the SQL queries involved, if any.*

- *Can you update employee information? Explain your answer and write down the SQL queries involved, if any.*

- *Can you delete employees with a certain last name? Explain your answer and write down the SQL queries involved, if any.*

Q. Briefly describe and demonstrate the prevention measures that can be taken against SQL injection attacks. [8 points]

2.3 WebGoat
We will mainly use WebGoat, an insecure J2EE web application to understand web application security. The latest downloadable package of WebGoat (5.4) comes with Java and Apache Tomcat installations. If

you are using an older WebGoat version, you may need to install them separately. There are packages for both *nix and Windows systems.

WebGoat is a J2EE web application which once installed, runs on the Tomcat server on your system.

Following are the three steps involved in installing and running WebGoat:

- Download and unzip the WebGoat package.
- Run WebGoat (it will start the Tomcat server automatically). OS-specific instructions are in the README file that comes with the package.
- Access WebGoat using a browser http://localhost/WebGoat/attack

Easy enough!

Q. Because WebGoat is a vulnerable web application (obviously!), it will make your system insecure while you work on this project. How can an attacker attack your system if you are using WebGoat without taking any precautions? [2 points]

Q. How do you fix the above-mentioned problem? (Hint: There are multiple ways to fix this problem. You get points for mentioning the solution that does not hamper other system functionalities.) [2 points]

Q. What is the principle of least privilege? How can you manage access privileges for using WebGoat? [2 points]

Additionally, you will need to download and install WebScarab. We will use it to intercept and modify HTTP requests and responses. However, before we proceed, familiarize yourself with these terms: proxy, message interception, HTTP GET, HTTP POST.

Running WebScarab involves three steps:

- Download and install WebScarab (Click and run).
- Change your browser/system LAN settings to use WebScarab as a proxy. WebScarab runs at localhost:8008.
- In WebScarab, go to the Proxy tab and intercept request/response messages as required.

3. Project Deliverables

Task 1: The main purpose of this task is to provide you with a strong knowledgebase in web security. Your answers should not exceed a single sentence. Terseness is the key to successfully completing this section. [16 points]

Q. Describe briefly the following concepts in the context of system/application security (any 16):

Input validation	Adware	Browser hijacking	Payload
Race condition	Spam	Checksum	Variant
Privilege confusion	Spyware	Sniffer	Firewall
Privilege escalation	Hacker	Password sniffing	Antivirus

Virus	Backdoor	Proxy	Retrovirus
Worm	Trojan Horse	Man-in-middle	Intrusion Detection System
Logic/time bomb	Brute-force attack	Pharming	Sandboxing
Encryption/Decryption	Denial of Service (DoS)	Phishing	Zero-day attack

Task 2: In sections 2.2 and 2.3, there are questions marked in red and *italics*. Under this task, answer all those questions. [26 points]

Task 3 a): Run WebGoat and execute all the lessons under 'Injection flaws' (leave out the ones that require development version of WebGoat).

Task 3 b): For each lesson, summarize your understanding including (48 points):

- SQL statements used.
- Modified SQL statement used for the SQL injection. How did it work?
- How did WebGoat fix the vulnerability (Hint: See 'Show Java' tab)?
- Screenshots

4. Project guidelines

You need to submit a project report that is no more than 5 single line typed pages (use point size 11). Remember, terseness is the key. Grading will be based on the overall writing, preciseness, and presentation. Form a group of 3 and submit only one report per group.

At the end of the report, each group member's contribution should be clearly specified (e.g., who contributed to which part of the project, what percentage, etc.). We would like to see all the members of the group taking equal responsibility. Also, all the members should be responsible to the entire project and should be ready to take oral questions if asked during grading.

You are expected to run this project on your own laptops or computers. Running WebGoat without understanding the security loopholes that it generates may make your system vulnerable. Do not proceed to Task 3 without completing tasks 1 and 2.