10/5/2015 **CSE 565: Computer Security** **Due: 11/05/2015**

Fall 2015

Project 2

Web Security and Symmetric Key Crypto Implementation

[Total Points that can be earned: 100]

This project is in two parts, viz. Web security and AES implementation. The objective is to gain a good understanding of web security that was covered only at a high level in class and to gain some hands-on experience on the symmetric key encryption algorithm that was covered in depth. In some sense, the first part involves some exploration whereas the second part is primarily a step-by-step implementation to solidify the AES concepts.

Part I – Web Security [48 points]

Background:

The primary goal of this part 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 which is a platform developed by OWASP to teach web security lessons.

Details:

This part will require considerable reading and answering specific questions, some tool installations and running tutorials and implementing and testing exploits. This part is divided into several sections for clarity. You are required to answer all the questions shown in red (also *italicized for clarity on grayscale print)*. For more details, read Section "**Project Deliverables**" below.

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.

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. Briefly answer the following questions about SQL injection. [4 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.*

Q. Briefly describe and demonstrate the various prevention measures that can be taken against SQL injection attacks (no implementations necessary). [4 points]

WebGoat

We will mainly use WebGoat, an insecure J2EE web application to understand web application security. The downloadable package of WebGoat (5.4) comes with Java and Apache Tomcat installations. If you are using older WebGoat versions, you may need to install them separately. There are packages for both *nix and Windows systems.

WebGoat is a J2EE web application that 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.

Q. Because WebGoat is a vulnerable web application, 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.

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. [8 points]

Q. Do some research on these items and then answer the following questions.

Input validation	Adware	Browser hijacking	Payload
Race condition	Spam	Checksum	Variant
Privilege confusion	Spyware	Sniffer	Firewall
Privilege escalation	Hacker	Password sniffing	Antivirus
Zero-day attack	Backdoor	Proxy	Retrovirus
Phishing	Denial of Service (DoS)	Man-in-middle	Intrusion Detection System
Logic/time bomb	Brute-force attack	Pharming	Sandboxing

1. Pick out 4 terms that belong to attack techniques and explain each of them.

2. Pick out 4 terms that belong to defense techniques and explain each of them.

Task 2: In sections “Injection Attacks” and “WebGoat,” there are questions marked in red (also in *italics*). Under this task, answer all those questions. [16 points]

Task 3a): Run WebGoat and execute all the lessons under ‘Injection flaws’ (leave out the ones that require development version of WebGoat) and write a summary of your experience in one paragraph.

Task 3b): From the lessons you learned (Task 3a), implement and demonstrate 1) string SQL injection and 2) Cross-Site Scripting attack (XSS attack). For each of these attacks, summarize your understanding including: [24 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.*

Project Guidelines for Part I:

You need to submit a project report (Part I) that is no more than 7-10 single line typed pages (use point size 11). Remember, terseness is the key. Grading will be based on the overall writing, preciseness, and presentation.

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). Also, do not stay connected to the Internet when you run this project. You are strongly encouraged to run this part by housing everything inside a virtual machine. If you have already set up a virtual machine on your laptop, you may skip the following step. We will use Ubuntu installed on a virtual machine. (This is necessary for installing OpenSSL that is needed for Project 3 next month.)

To install a virtual machine, follow these steps:

(a) Download and install the free software VirtualBox, a virtualization product from: http://www.virtualbox.org/
(b) Download and install the latest stable version of Ubuntu. Get an ISO disk image from: http://www.ubuntu.com/desktop/get-ubuntu/download
(c) Fire up the virtual box
    a. create a new virtual machine with the default settings
    b. Point the CD image to the Ubuntu ISO image
    c. Start the Virtual machine: the installation of Ubuntu should proceed with minimum or no delays/problems.

Part II - Block Ciphers: Symmetric Key Crypto Systems [52 points]

While Part I covered web security, Part II closely follows the material covered in class and it gives you an opportunity to put the various cryptographic principles into practice. In this part, you will learn more on encryption/decryption, cryptanalysis, and usage of cryptographic toolkits by exploring the operation of the AES encryption algorithm by tracing its execution, computing one round by hand, and then exploring the various block cipher modes of use. This may appear like a routine exercise but is aimed at enhancing the understanding of AES and its use. An online Java applet is used (or can be downloaded) to execute AES. This part of the project is divided into three separate parts:

- **Block cipher internals:** This part involves encrypting plaintext and analyzing the intermediate results after each round. There is an online calculator for both AES and DES that provides the intermediate results and the final ciphertext.
- **Block cipher round:** This part involves calculating one round by hand and comparing the results to those produced by the calculator.
- **Block cipher modes of use:** Enables the student to compare the operation of CBC and CFB modes.

Use the link: http://williamstallings.com/Crypto/AESCalc/AESlab.html to go over the project guidelines, deliverables and *what is expected in the project report*. **Group leaders must send an email to** shambhu@buffalo.edu **to receive the AES Triple that is needed to complete this project.**

Project Guidelines for Part II:

You need to submit a project report (Part II) that is no more than 5 single line typed pages (use point size 11).

Guidelines for the Entire Project:

Form a group of 4-5 and submit only one report (including both parts) per group. If you refer to any particular paper or web document for this project, remember **to mention the source in your report**.

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 for the entire project and should be ready to take oral questions if asked during grading. Random groups may be asked to explain your solution.