CS 5460: Computer Security I
Fall 2020

Assignment 3: Security Scanners
Total Marks: 190

Assume the following scenario:
You have decided to build a startup company, which would design security scanners for general users and software developers to protect against various types of cyberattacks. To achieve the goal, you first need to build an initial prototype that you can present to the investors, to convince them to provide seed funding for your company. As a part of designing your initial prototype, you need to complete the following tasks. Your prototype may not be ready yet for real-life use, but should be able to assure the investors about your knowledge, expertise, and creativity in the field of cybersecurity.

You can use any programming language of your choice for this assignment. You can use either a Graphical User Interface (GUI) or Console/Command Line to take user inputs and show the outputs. You should carefully read through the assignment description to identify which programming language would work best for you.

Task I

In this task, you will need to build a security scanner that can identify a phishing email based on the sender's email ID and the text in the email body that may include the URL to a website.

Inputs:

i) Sender’s Email ID
ii) Text in Email body (no longer than five sentences [excluding URL], for this assignment)

Outputs:

i) Likelihood of the email to be a phishing email. You can use percentage value, or a likert scale of your choice (e.g., 1 to 10; 1: not likely at all, 10: very likely) to express the likelihood of an email to be a phishing email. You can call it a ‘phishing vulnerability scale’, where a higher score represents a higher vulnerability.

ii) Brief Explanation of why the email is a likely phishing email (Such explanations could help the regular users understand the vulnerabilities associated with an email they have received). Keep the explanation simple and focused, pointing to the sender’s email ID and/or the part of email contents, because of which the email is identified as a likely phishing email. For a legitimate email, no explanation is needed.

Report 1 (Format: Word/PDF):

a. Explanation of how you designed your ‘phishing vulnerability scale’.

b. The scopes of improving your security scanner in identifying phishing emails.

c. Examples of inputs you used to test your system and the outputs you received. You can either write down your inputs/outputs or attach the screenshots from your program’s input/output screen. For each input/output pair, include a heading for what you tested. Note: We will evaluate your program through additional inputs, too.

d. Pick ‘emailID’ of any five recognized organizations of your choice. You can call it a ‘white list’. Include this ‘white list’ with your report (see the phishing identification techniques in Grading Rubric below for further understanding of how you could use this ‘white list’).

Grading Rubric (Task 1):

Task 1: Computer Programming (Total Marks: 72.5)	
Phishing Identification Techniques	**Marks**
Sender’s Email ID	
Common letter substitution in the email ID from your ‘white list’, e.g., ‘...paypa1.com’ instead of ‘...paypal.com’	10
Addition of keyword (e.g., real, admin, etc.) to the email ID from your white list (e.g., ‘gapfactory_real@email.gapfactory.com’ instead of gapfactory@email.gapfactory.com)	10
Contents in Email Body	
Urgent Action Required	10
Promise of Reward	10
Suspicious URL	
Start with the number	7.5
Does have a keyword (e.g., update, login, verify, etc.) in domain-name	7.5
Measuring and presenting the likelihood of an email to be a phishing email based on the above techniques	12.5

Presented explanation of why the email is a likely phishing email	5
Report 1 (Total Marks: 17.5)	
a	5
b	5
c	5
d	2.5

Notes: You do not need to consider ‘mismatched URL’ in this assignment, as a technique of identifying phishing emails. Also, this list of phishing identification techniques is not conclusive, although sufficient for this assignment. You can go through your lecture materials to learn about different techniques to identify the phishing attempts.

Task II

Pseudocode of **Query 1**:

```
SELECT ItemDescription, ItemPrice
FROM Items
WHERE ItemName = <input from user>
```

Pseudocode of **Query 2**:

```
SELECT Accounts
FROM Users
WHERE Username=<input from user> AND Password=<input from user>
```

Assume that attackers could exploit <input from user> fields in the above two queries to conduct an SQLI attack. In this task, you will need to build a security scanner that can identify the following types of SQLI attacks by scanning an sql query: Tautologies, Illegal/Logically Incorrect Queries, Union Queries, Piggy-backed Queries, Inference, Alternate Encodings.

Inputs:

i) An SQL Query

Outputs:

i) Likelihood of the query to be malicious. You can use percentage value, or a likert-scale of your choice (e.g., 1 to 10; 1: not likely at all, 10: very likely) to express the likelihood of an SQL query to be malicious. You can call it a ‘SQLI vulnerability scale’, where a higher score represents a higher vulnerability.

ii) Brief Explanation of why the query is likely to be malicious. Keep the explanation simple and focused, pointing to what makes the query likely to be malicious. For a legitimate query, no explanation is needed.

Report 2 (Format: Word/PDF):

a. Explanation of how you designed your ‘SQLI vulnerability scale’.

b. The scopes of improving your security scanner in identifying SQLI attacks.

c. Examples of inputs you used to test your system and the outputs you received. You can either write down your inputs/outputs or attach the screenshots from your program’s input/output screen. For each input/output pair, include a heading for what you tested. Note: We will evaluate your program through additional inputs, too.

Task 2: Computer Programming (Total Marks: 85)		
Attack Types	**Marks**	
	Query 1	**Query 2**
Tautologies	5	5
Illegal/Logically Incorrect Queries	5	5
Union Queries	5	5
Piggy-backed Queries	5	5
Inference		
Blind Injection	5	5
Timing Attack	5	5
Alternate Encodings	5	5
Measuring and presenting the likelihood of a query to be malicious based on the above techniques	5	5
Presented explanation of why the query is likely to be malicious	2.5	2.5

Report 2 (Total Marks: 15)	
a	5
b	5
c	5

Submission Instructions

- The program without required input fields and visible output is not acceptable. A program that does not run (e.g., due to errors/bugs in code) is not acceptable.

- You are required to do your own work. Individual submission is needed from each student.

- You will need to submit the reports (e.g., Report 1 and Report 2) and the working version of your codes via Canvas before **11:59 PM on Wednesday, December 09**. Add necessary instructions for running your code in a ‘Read Me’ file. For multiple files, you can zip them before submission.

Grading Rubric (Overall)

Tasks	**Marks**
Task 1	
Computer Programming	72.5
Report 1	17.5
Task 2	
Computer Programming	85
Report 2	15
Total Marks	**190**

See below for further information on the grading criteria for Computer Programming in each of the above tasks.

Criteria	**What does *excellent* performance represent for each criterion?**	**Percentage of Total Marks (for a Task)**
Requirements, Delivery, and Efficiency	● Completed all of the tasks and requirements. ● Delivered in correct format. ● Thorough and organized testing or input validation has been completed. ● Solution is efficient, and easy to understand.	70%
Coding Standards and Presentation	● Excellent use of white space. ● Creatively organized work. ● Excellent use of variables. ● Excellent user prompts, good use of symbols, spacing in output	15%
Documentation	● Clearly and effectively documented including descriptions of all variables. ● Specific purpose noted for each function, control structure, input requirements, and output results.	15%

Possible Questions and Answers

Q. *How can we design a generalized framework to identify phishing attacks?*

Ans. There is no surefire way to protect against such attacks, but the lack of an effective security scanner in the wild is certainly making the tasks easier for attackers! This assignment motivates you to address these issues, where you need to recall your learning from class lectures and use your intuition, critical thinking, and creativity to reflect that learning in practice.

For example, one of the tricks used by attackers in Phishing attacks is 'immediacy (i.e., ‘urgent action required’)', which could be presented in an email in different ways, like the keywords: 'immediately', 'as soon as possible', 'right now' may relate to such intent of attackers. Remember, the presence of such keywords may not be sufficient indication of a phishing attack, like, 'We will get back to you as soon as possible' might seem less alarming than 'Your offer will expire unless you redeem it right now'. It could also be written as: 'Your offer will expire in the next one hour', which might be more alarming than 'Your offer will expire in next three days'. Well, is it more/less alarming according to your scanner? A large fraction of phishing sites remain live for less than a day. So, at which point will your scanner raise a red flag based on the time given to the user to act? Do you want to set a fixed threshold, like 1 hour? -OR- Do you want to make this threshold 'dynamic' depending upon other contexts of an email?

Sometimes, the urgency might be expressed through the context of a statement without using any keywords of 'urgency' noted in the paragraph above, like 'We will close your account if your information is not verified' - it does not contain any keyword from above (well, it might contain such keywords too!), but the perceived consequence (e.g., losing access to an account; it might be a bank account or any other accounts - does the email ID of the sender tell us about that?) may puzzle a user and make him/her taking immediate action (and fall for phishing!). On the other hand, this could be a legitimate statement, too, right? However, if there is a suspicious URL (e.g., https://www.cap1talone.com) along with the above statement where the user is asked to click on the link to provide information for verification, then the context might present a risk of phishing!

Remember, the above discussion is neither complete nor sufficient to detect phishing attacks based on identifying 'immediacy' related to the malicious intents of attackers. Rather, it provides you with some 'food for thought' that you could leverage as you write your program for security scanner.

Consider other tricks for phishing, and possible variations in exploiting those tricks. There might be a hundred variations that a single person could not even imagine, but definitely, there is not just one! See assignment description above for the basic phishing identification techniques that you need to consider for this assignment.

Q. *How can we design a generalized framework to identify SQLI attacks?*

Ans. Consider the intents of each SQLI attack you have learnt, and find examples related to those intents. For this assignment, you only need to consider the intents discussed in class lectures, like bypassing authentication, extracting data could present the intents of Tautologies Attacks. With carefully going through your list of identified examples, find a general framework that would detect malicious SQL queries in your list and the ones similar to those.

Consider possible variations in SQL queries that could be crafted by the attacker. For example, in Tautologies SQLI attacks, the conditional statement evaluates to True, so 3=3, 5>2, ASCII ('a')=97: any of these statements could be exploited by the attackers. Remember, attackers will try to outsmart your security scanner, so you need to be creative, too! If your scanner can identify ASCII ('a')=97, but does not raise a red flag for ASCII (Substring ('SQLI', 1, 1)) > 10, then it's a pretty good news for the attacker!

False identification of suspicious queries is a notorious issue for the security scanner, it could reject legitimate queries from being executed. Like, the presence of a special character does not necessarily make a query malicious (the instructor of this course has a special character in his last name!). Similarly, the presence of an SQL comment operator may not be a sole indication of a query to be malicious (e.g., a password may contain such characters; can I create a user name: 'al--ameen.usu'?). Also, 'Select' is the name of a real-world product, 'Update' is the part of a movie name: "Status Update". So, you may not want to mark a query to be malicious just based on the presence of a keyword, or special characters.

Leverage your knowledge of SQL statements, keywords, functions, comment characters – that you have learnt from class lectures to effectively identify different possible variations of SQLI attacks.