**Vulnerability Assessment and Systems Assurance Report**

***TuneStore***

Shashank Mondrati

ITIS: 4221

February 4, 2022

**VULNERABILITY ASSESSMENT AND SYSTEM ASSURANCE**

**TABLE OF CONTENTS**

Page #

**1.0**  **GENERAL INFORMATION** 3

1.1 Purpose 3

**2.0** **VULNERABILITIES DISCOVERED** 4  
 2.1 Log-in as a random user 4

2.2 Login as a specific user 6

2.3 Register a new user with lots of money in account w/out paying for it. 9

**3.0** **XSS VULNERABILITIES** 12

3.1 Stored XSS 12

3.2 Reflective XSS 13

1. **GENERAL INFORMATION**

***1.1 Purpose***

The objective of this TuneStore application assessment is to identify and analyze the vulnerabilities that are present in this TuneStore application. More specifically the assessment and penetration testing is to determine the security levels that are within the scope of the application. The vulnerabilities being discussed in this report will include use cases, and XSS vulnerabilities.

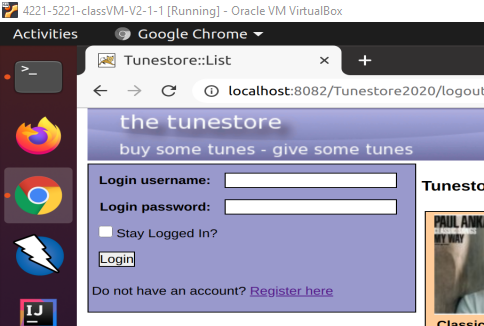
***2.0 VULNERABILITIES DISCOVERED***

***2.1 SQL Injection***

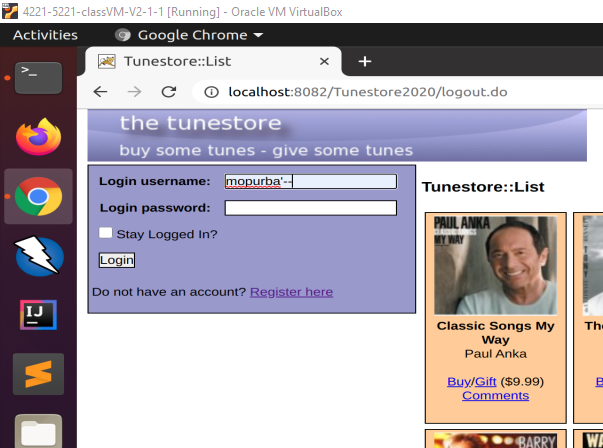
The **TuneStore** application is vulnerable to several SQL injection attacks. An SQL injection attack can occur when a user is asked to enter a normal username or sample credit card number, but instead gives anSQL statement that will unknowingly be run on the database server that is behind the web application. This gives an attacker the opportunity to execute malicious SQL statements on the database. For instance, using SQL injection, an attacker can add the amount they want to their account with the help of a credit card. Additionally, they could access, modify, or delete the data that is contained in the database. In the case of the **TuneStore** application, SQL injection could be used by an attacker to login as a random user without needing to know their credentials.

***2.1 SQL Injection - Login as a random User***

One example would be perfect in an instance where they can log into the application. By logging into the TuneStore application as a random user, this vulnerability exists in the logging in functionality on the login page of the application.The logging in functionality uses an SQL statement to check if the entered username and password inputs exist in the database. The attacker can easily login as a random user or as a specific user just by adding (**‘--**) to the end of the user name, and accessing their information easily. Below is a screenshot of the login page, and where they can login as a random user in the TuneStore application.

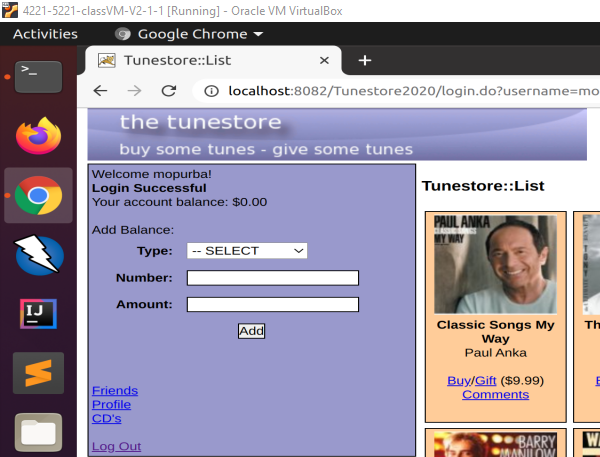


An attacker is able to login with the method that is mentioned above, below is the screenshot of an attacker login using the method.



Using these methods, the attacker can login into almost any user, access their credentials easily making it vulnerable to hack their information.

Below screenshot shows that they can login as a random user of the attacker’s choice.

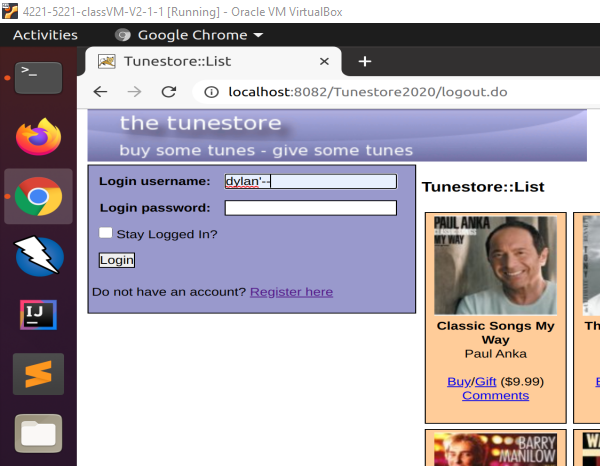


As you can see, the attacker is able to login with the method, and they are able to add amount.

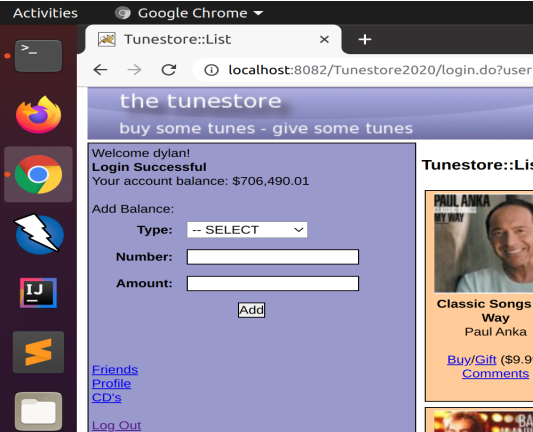
***2.2 SQL Injection- Login as a specific user***

Using the same methods, we can login to almost any application. For this instance, I asked my friend to create account in TuneStore, since I do not know the password, I was able to login to his account with ease, and access stored money in his account.

Below is a screenshot of logging to his account and checking his funds in his account.

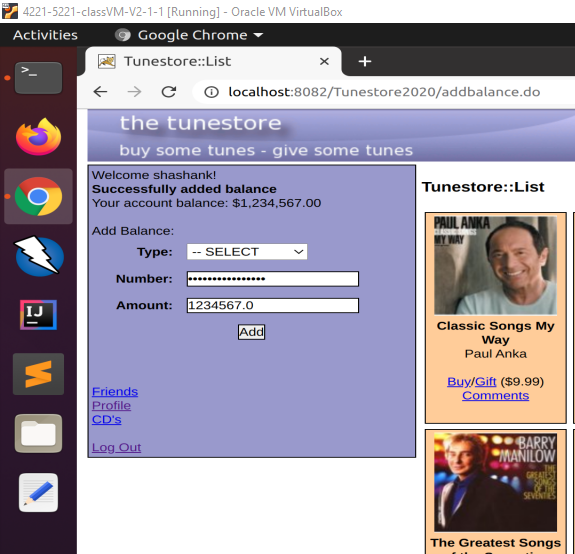


Above screenshot shows, adding (‘--), we can login to the specific user account, and you can check the below screenshot to check his funds.



As you can see above, there are over $700,000 in his account, the vulnerability is anyone can hack into anyone’s account, and access their funds to buy anything in the TuneStore application, not to forget there’s also another vulnerability where the attacker can add money without the help of adding their credit card numbers, the alternate way of doing this is the attacker can add money by putting the type as TYPE when adding balance.

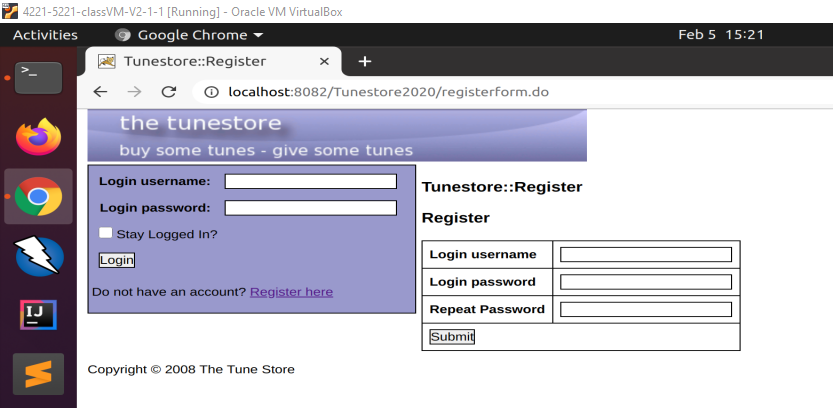
Below is a screenshot where they can add funds without specifying that type of balance to add.



This screenshot shows they can add their money with 16 random numbers, and by putting the type of option to add money as TYPE to add money, once they are done adding money, the attacker is able to add money with ease. You can also see that when they click on ADD, on the top it says “Your new account balance : $1,234,567.00”.

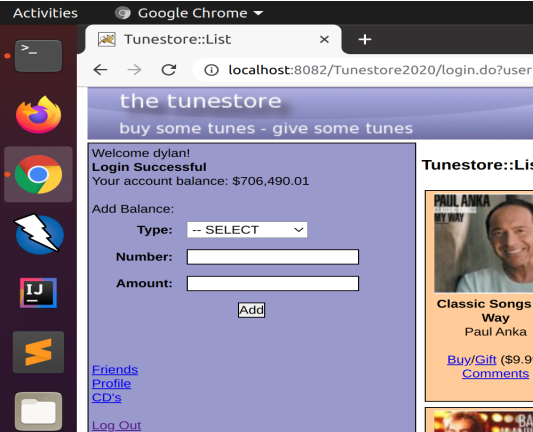
***2.3. SQL Injection- Register a new user with lots of money in account w/out paying for it.***

Users also have the ability to register an account, although there are many use cases in this TuneStore application, where as the users can login, logout, create an application, add their friends, buy something from the application. When trying to register for an account, the TuneStore application registry does not ask for very specific information, asks for basic information. Below is the screen shot, when asked to register for an account.



Above is the screenshot, where you can register for an account. I suggest if you wish to not get hacked use some extra symbols , because that prevents attackers from accessing the user’s information.

I did some SQL injection another account, where I found there were a lot of money in their account. Below is a screenshot where it shows how much money they have.



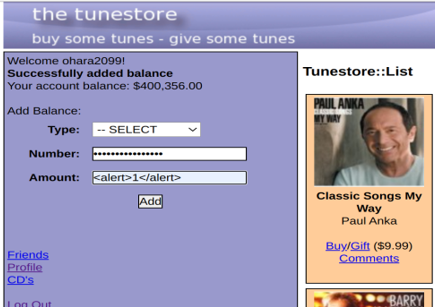
As you can see, the above screenshot shows the user how much money they have in their account.

***3.0. XSS Vulnerabilities***

What is XSS Vulnerability? We can define it as a common attack vector that hackers use by injecting malicious code where they can find a vulnerability in the web application. XSS differs from other web attack vectors (e.g., SQL injections), in that it does not directly target the application itself. Instead, the users of the web application are the ones at risk.

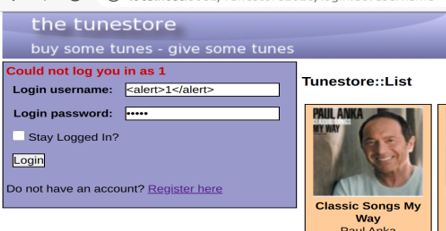
***3.1. Stored XSS Vulnerability***

We can define a Sstored XSS vulnerability as when a perpetrator located a vulnerability in the application, and they inject malicious code into the field to steal valuable information. They can insert malicious code in the comment field or user input field. Below is a screenshot showing Stored XSS vulnerability. That can be used to steal valuable information such as session cookies, and can be activated when ever they visit the site.



***3.2 Reflective XSS Vulnerability***

Reflected XSS attacks, also known as non-persistent attacks, occur when a malicious script is reflected off of a web application to the victim’s browser. As you can see, when executed we can see the parameter javascript which can be seen as a vulnerability in the application.



When hackers type the malicious code in the user name field, you can see that sample javascript field has been executed, when it shouldn’t be executed, this can be seen as a weakness to hack their way through.