Vulnerability Assessment and Systems Assurance Report - Tunestore

Tunestore Security Report

Omar White

ITIS 4221/5221

March 12th, 2024

# Vulnerability Assessment and Systems Assurance Report

Table of Contents

Section Page #

1.0 General Information 3

1.1 Purpose 3

2.0 CSRF Vulnerability 4

2.1 Involuntarily Adding a Friend 4

2.2 Involuntarily Giving a Gift 8

2.3 Changing a Password 12

3.0 Access Control Vulnerability 15

4.0 XSS Vulnerability 18

5.0 Clickjacking Vulnerability 21

**1.0 General Information**

*1.1 Purpose*

The objective of this Tunestore application security assessment is to identify and analyze vulnerabilities that are present within the application. Specifically, the purpose of this security assessment and penetration test is to determine the overall security of the application that is within the scope of engagement. The vulnerabilities being discussed in this report include a Cross-Site Request Forgery (CSRF) vulnerability, an Access Control vulnerability, a Cross-Site Scripting vulnerability (XSS), and a Clickjacking vulnerability.

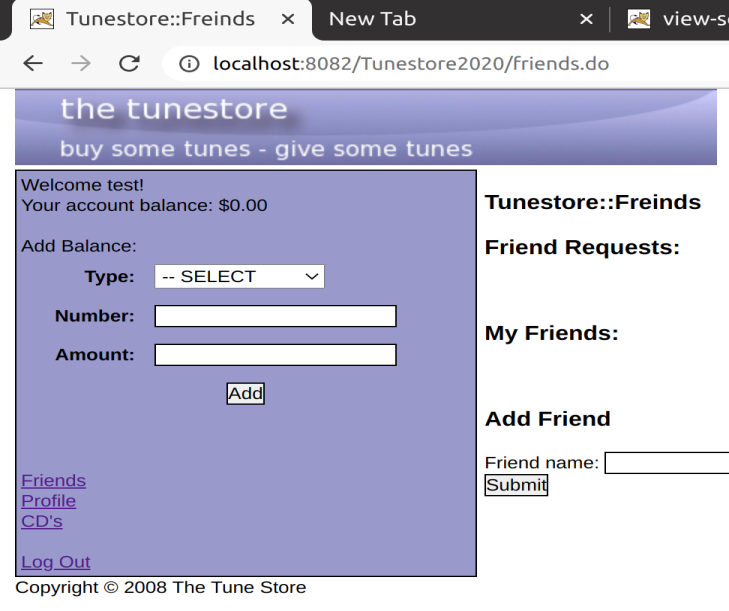
**2.0 CSRF Vulnerability**

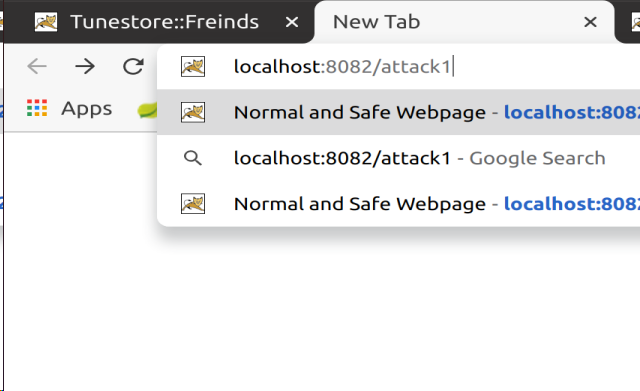
The Tunestore application contains a CSRF vulnerability. This is when a malicious website sends requests to a vulnerable website on the user’s behalf without the user’s knowledge. This vulnerability could allow attackers to impersonate users to send malicious requests. For example, if a user is logged into the Tunestore application and visits a malicious website, the malicious website could give a gift to another user on their behalf. During this assessment, it was found that this vulnerability could be exploited to add users as friends, give gifts, and change passwords, all without input from the actual logged in user.

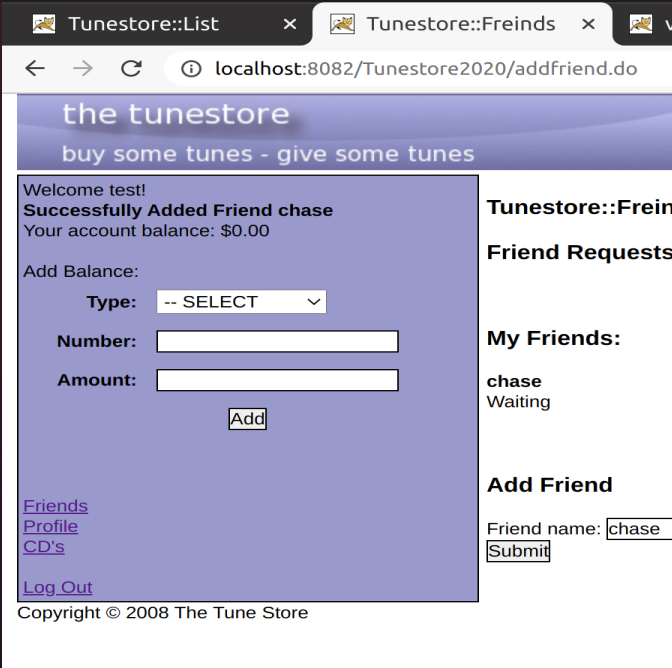
*2.1 Involuntarily Adding a Friend*

One of the specific attacks that a malicious actor could perform on a Tunestore user with the CSRF vulnerability is adding a friend to their account without the knowledge of the user. The vulnerability lies in the “friendForm” form. Because there is no check to make sure that a post request addressed to “/TuneStore2020/addfriend.do”, and attacker can make a form with the precious url segment as the action and the application will process it as a legitimate request. Below are screenshots of the malicious website’s source code and pictures of what the user would do to become victim to such an attack.





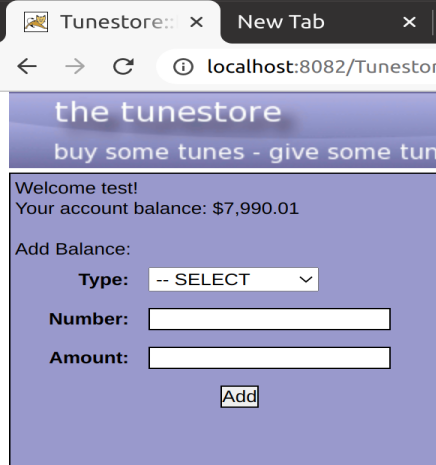


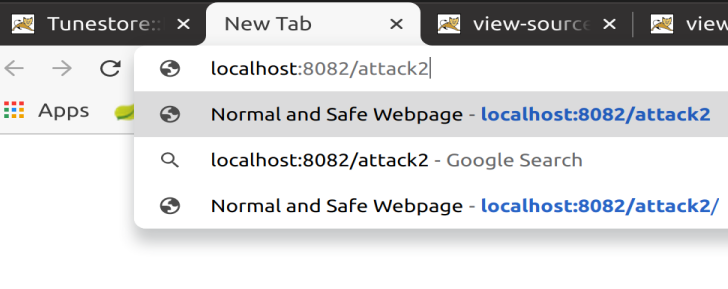


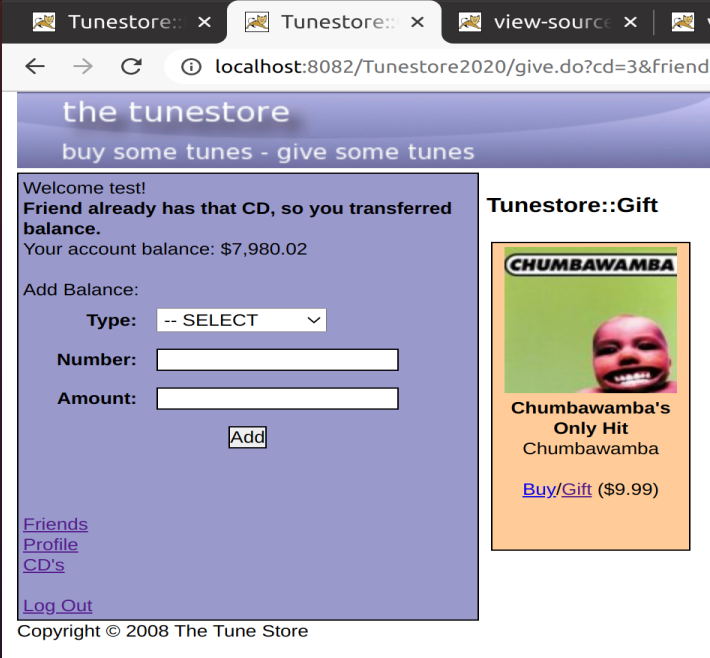
*2.2 Involuntarily Giving a Gift*

This vulnerability also allows an attacker to give a gift on the victim’s behalf. By creating a form on a malicious website that posts to “/Tunestore2020/give.do?cd=<number>&friend=<friend name here>”, a logged in user will give the cd specified by <number> to a user specified by <friend name here>. The following screenshots show the source code of an example malicious website and the order of events a victim would go through.

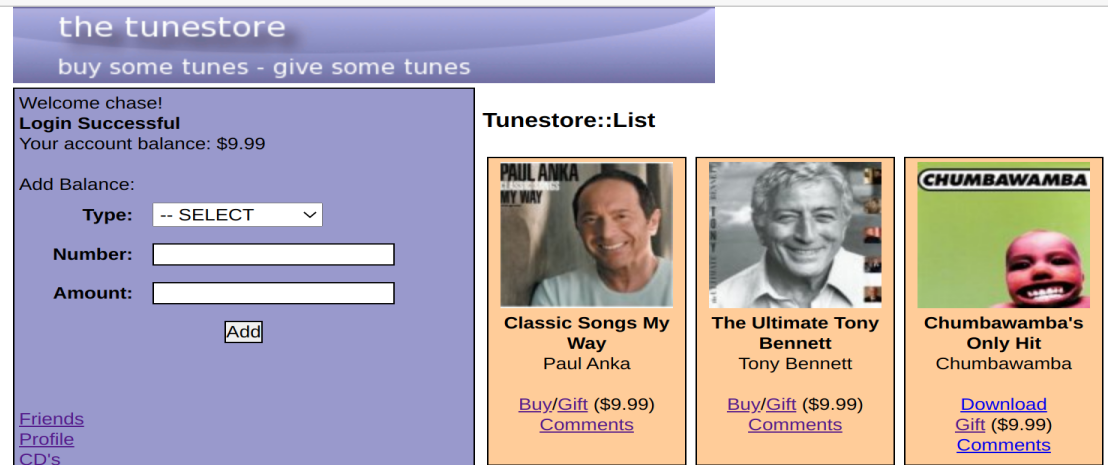








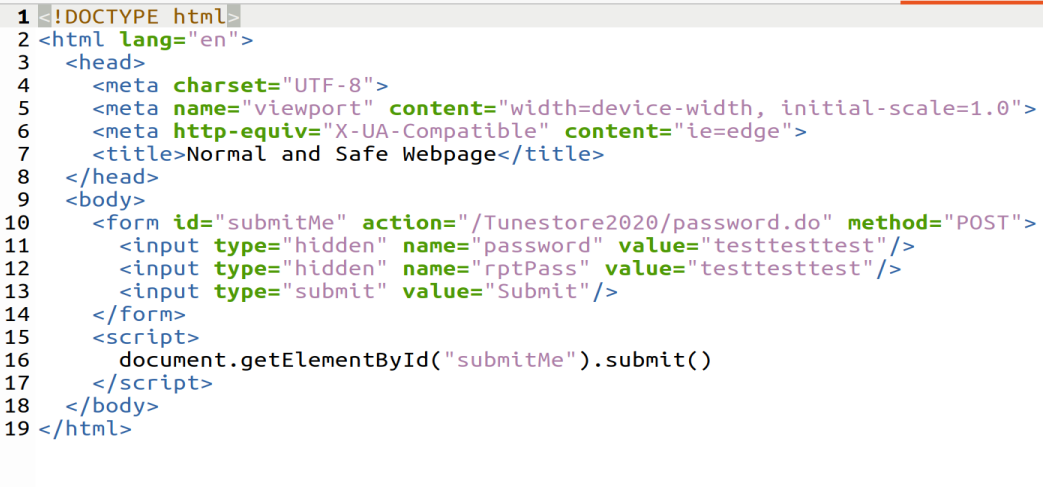
(this wasn’t the first time I ran the test)

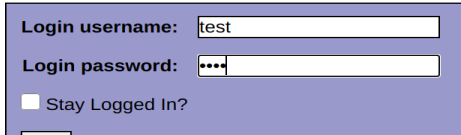


(balance and visible download button show that user chase successfully received the gifts)

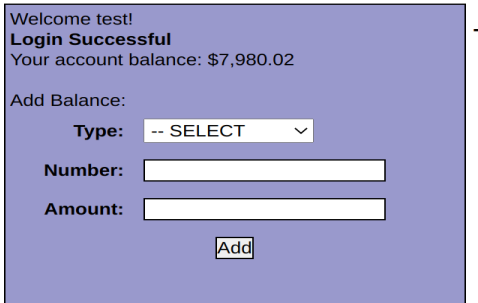
*2.3 Changing Password*

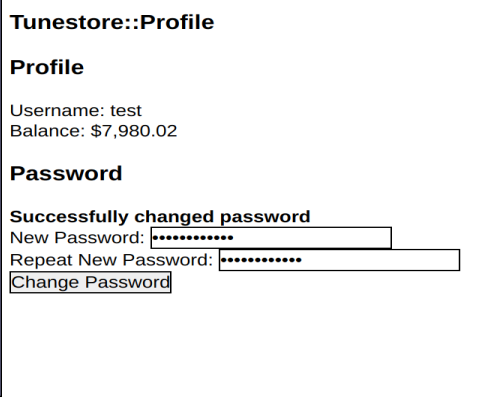
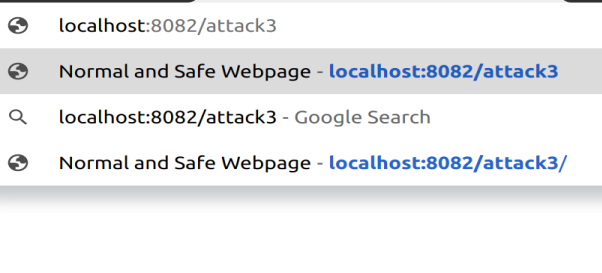
The CSRF vulnerability also allows an attacker to change a victim's password. This can be done by sending a post request with the action “Tunestore2020/password.do” with inputs named “password” and “rptPass”. Below are screenshots of an example malicious website, as well as the order of events that would lead a victim to having their password changed. The screenshots below show an example of a malicious website capable of performing this attack as well as a sequence of events that would result in a victim having their password changed.





(password is test)

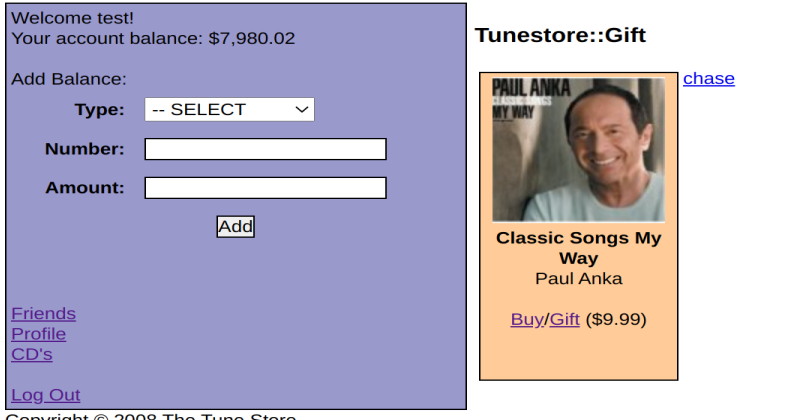


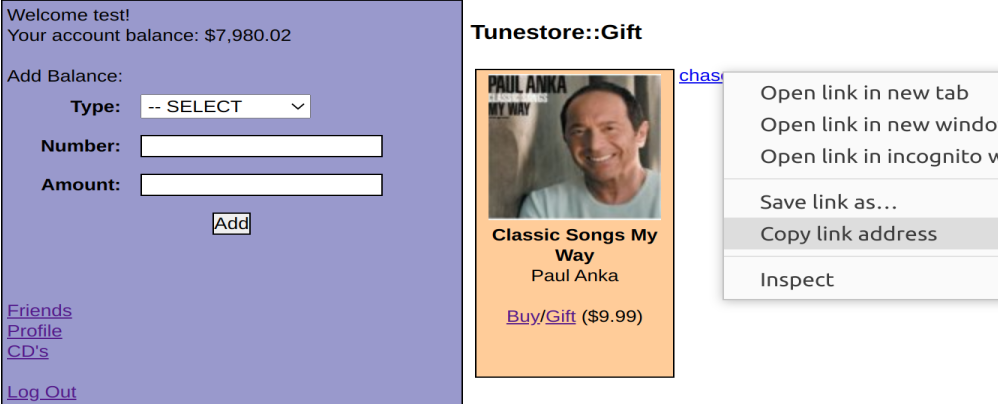


(password is now testtesttest)

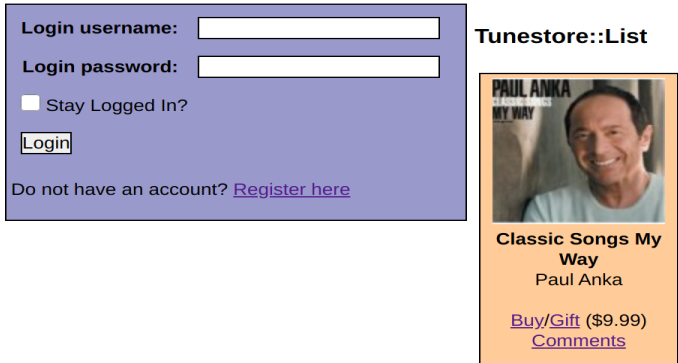
**3.0 Access Control Vulnerability**

The Tunestore application also contains an Access Control Vulnerability. This is when access controls are configured incorrectly. As a result, an attacker could perform a malicious action such as gifting a cd to another user while logged out, circumventing a charge to their account. This vulnerability doesn’t even require malicious code. Attached below are screenshots showing the sequence of events to exploit this vulnerability.

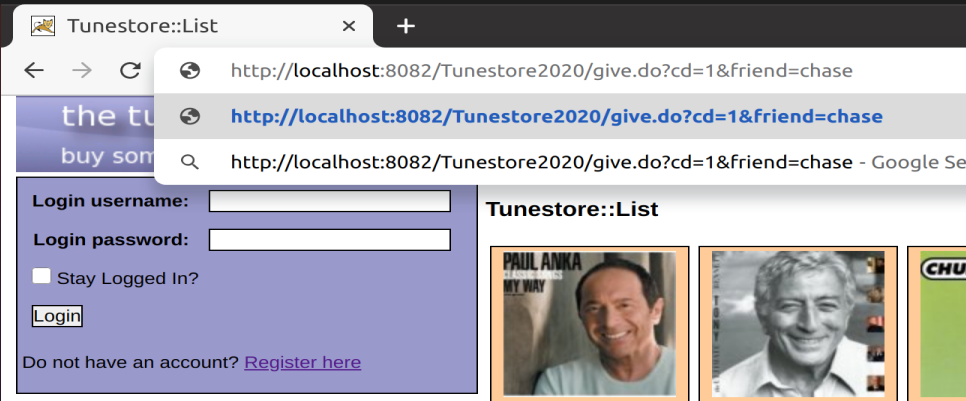




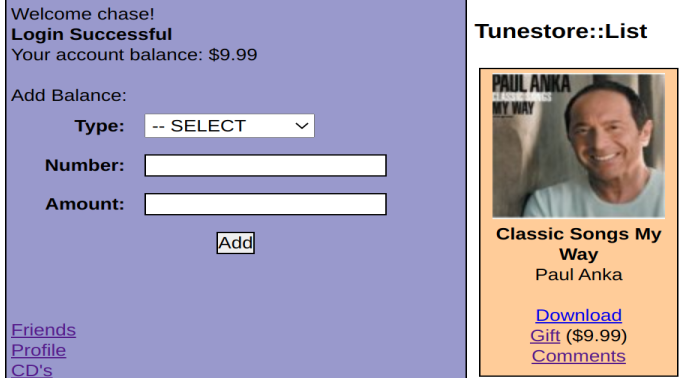
(link address is copied)



(user is now logged out)



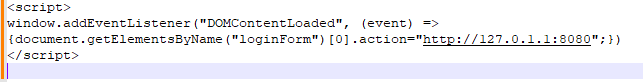
(this will give user chase a copy of cd 1)



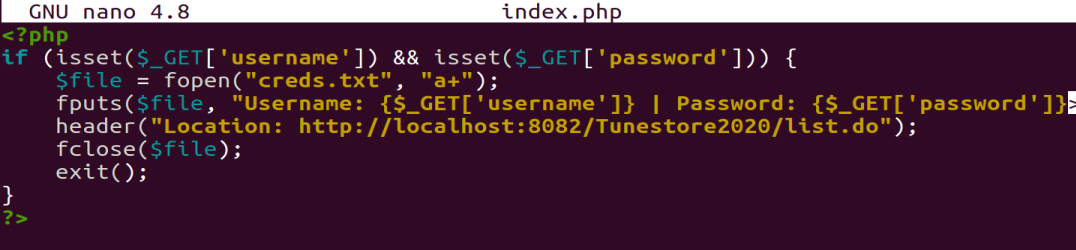
(download button confirms user chase received the cd)

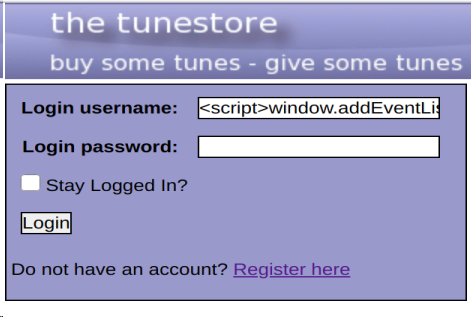
**4.0 XSS Vulnerability**

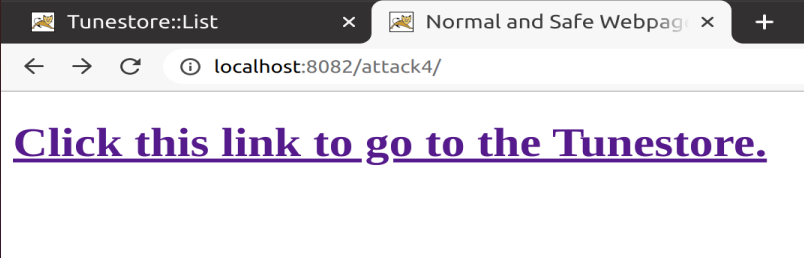
Tunestore suffers from an XSS vulnerability that allows attackers to steal credentials if users try to log in on a page modified using this vulnerability. The vulnerability is the “username” field of the form “loginForm”. The following screenshots show an example of the malicious code used to perform such an attack, malicious code that could be used to make a phishing webpage, the sequence of events that would lead to a victim having their credentials stolen, and an image showing sample credentials stolen by the attack.

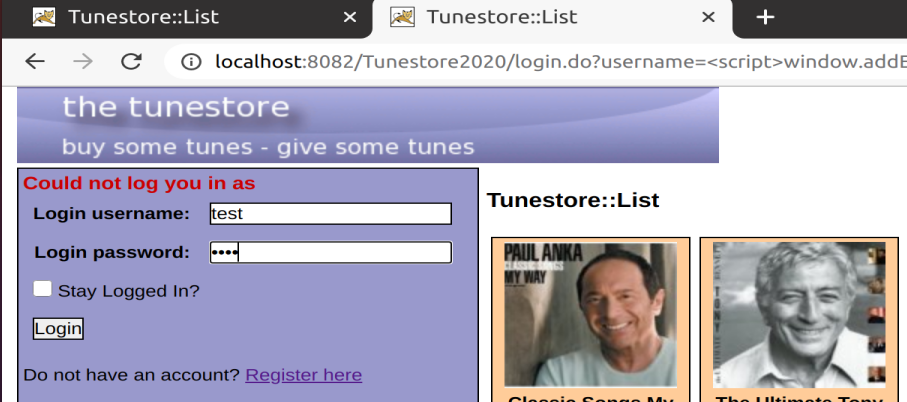


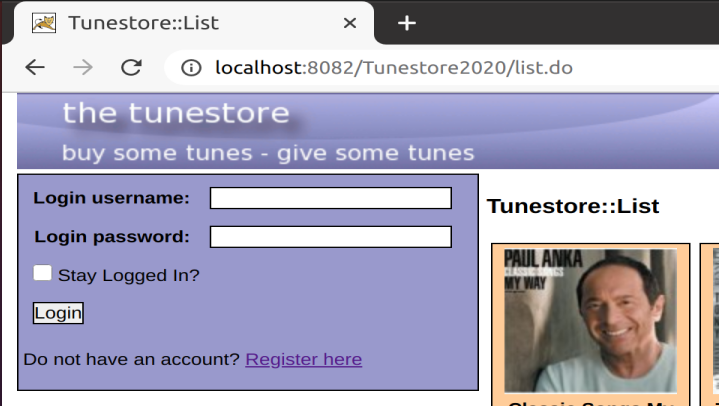












(user gets redirected to Tunestore as if nothing happened)

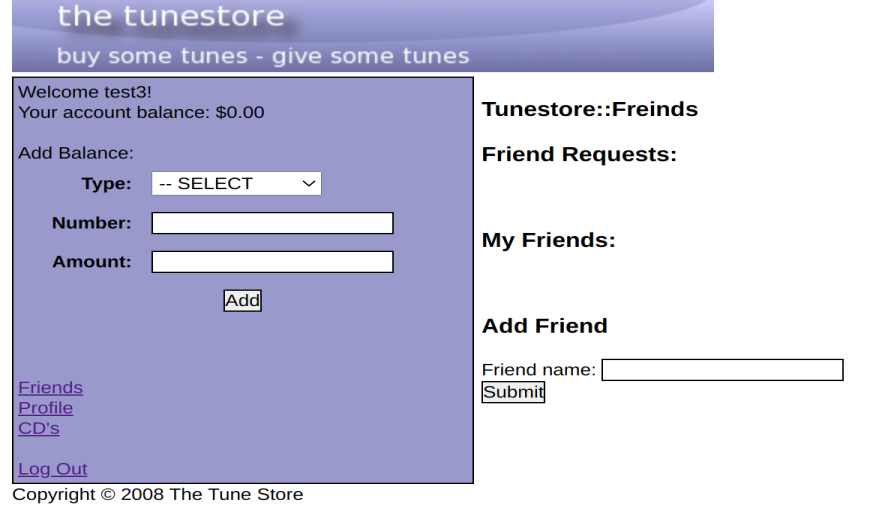


(stolen credentials stored in a text file)

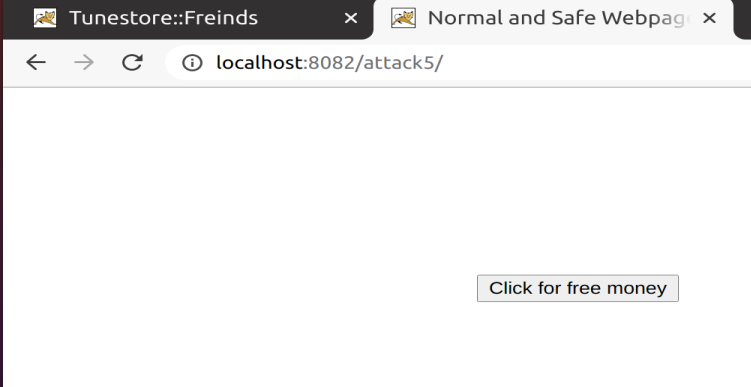
**5.0 Clickjacking Vulnerability**

The Tunestore application is also vulnerable to clickjacking attacks. Said attacks can be used to confirm requests that the user is completely unaware of by clicking on seemingly innocuous buttons. For example, a clickjacking attack could be used to send a fraudulent friend request. The screenshots below show the source code of a malicious website that could be used to perform such an attack, as well as images showing a sequence of events leading to the execution of a successful attack.





(user has no friends before attack)



(user clicks button)

