CS6035/4235 Project 4: Web Security

Setting Up

Download the virtual machine for this project via one of the following links:

- Mirror #1: http://www.prism.gatech.edu/~nyang38/cs6035/websec.ova
- Mirror #2: https://drive.google.com/open?id=0BxfOFsJpKHKgLUx5TU1BTjdENzA
- Mirror #3: https://gtvault-my.sharepoint.com/personal/nyang38_gatech_edu/_layouts/15/guestaccess.aspx?docid=0607e8736d3ac4893aa2e93ad93aaff21&authkey=ARcNEE74atPSWQG4n0OLj54

You can verify that the VM was properly downloaded by comparing the resulting hash:

Operating System	**Command**
Windows	`CertUtil -hashfile websec.ova MD5`
Mac	`md5 websec.ova`
Linux	`md5sum websec.ova`

md5: d12b32327a5fc5bcd3df6bf8318a652c

You are provided with both root and regular user access to this virtual machine. The credentials are:

Username	**Password**
root	root
user	user

You should only use the **user** account to complete the project. **root** is provided for your convenience in case you need to install extra software or packages.

Interacting with the VM

Xorg and openssh-server are set up on the VM, and the virtual appliance is preconfigured to forward the host's port 2222 to the appliance's port 22. As a result, you can ssh into the VM by running:

```
ssh -p 2222 -X user@127.0.0.1
```

You can use `scp` to transfer files between the host (your machine) and the VM.

The following command transfers **t1.html** from the host's current directory to the home directory of the **user** account on the VM:

```
scp -P2222 t1.html user@127.0.0.1:~/
```

The following command transfers **t1.html** from the home directory of the **user** account on the VM to the host's current directory:

```
scp -P2222 user@127.0.0.1:~/t1.html .
```

You can also launch the appliance in GUI mode. After logging in, you can start a minimal LXDE environment with the command `startx`. VirtualBox guest additions have been pre-installed on the VM. If you wish to install more packages, you may do so by running `apt-get` as root.

Georgia Tech Payroll

The site we will be exploiting in this project is **http://payroll.gatech.edu**, which you can visit on the VM. Please note that this is a made-up site and does not point to a legitimate site in the real world. For testing purposes, you may register accounts at your will. However, **please DO NOT use your actual passwords and banking account information**.

The source code of the site can be found on the VM in **/var/payroll/www**. We will be using **Firefox**, which is provided in the VM, to test your exploits. You may also assume JavaScript is always enabled in Firefox. Do not update your Firefox version since we will grade your scripts using the version provided in the VM.

GOOD LUCK AND HAVE FUN!

Deliverables Summary/Requirements

There are 3 targets in total worth 70 points, and the write-up is worth an additional 30 points.

Please submit your deliverables on T-square as separate files. Do **NOT** zip them. Failure to follow this rule will result in a **50% penalty**.

Filename	**Description**
t1.html	Crafted HTML page for Target 1
t2.html	Crafted HTML page for Target 2
t3.html	Crafted HTML page for Target 3
report.pdf	Please include your full name and your Georgia Tech username (e.g. jdoe3) at the top of the report. This should contain the required responses to the Epilogue section, as well as any of the partial credit responses you wish to include for each target.

Not following the file naming convention results in an automatic failure (0 points).

Disclaimer

This project is solely for educational purposes. Professor Wenke Lee and the people affiliated with his teaching and research are NOT responsible in the event of any criminal charges brought against any individuals misusing the information in this project to break the law. When in doubt, please consult the TAs or Professor Lee regarding any questions or issues you may have.

Target 1: XSRF (20 points)

You have stumbled upon the Georgia Tech payroll website and discovered a vulnerability. Suppose a user, say Alice, is already logged in the Georgia Tech payroll site. You noticed that you can craft a web page so that when Alice visits your web page, she gets redirected (NO popups) to the Georgia Tech payroll page with her account number and routing number set to some values of your choice.

Poor and living off of ramen, you decide to give it a try and craft a web page to set the banking information to yours.

You forgot your bank account information, but luckily, you remember storing them inside a script you wrote a long time ago.

To fetch your bank account number and routing number, run the **get_bank_info** script inside the VM and pass in your Georgia Tech username (e.g. jdoe3):

```
get_bank_info jdoe3
```

Here is an example of what the script will print out:

```
Username: jdoe3
Account number: 962362227
Routing number: 2113956237
```

Double check that you entered your Georgia Tech username. This is the username you use to login to T-square. It is **NOT** your 9 digit student number. If you enter the wrong username, which generates a different account and routing number, your exploit will fail our scripts, and you will receive zero points for this part.

The user must **NOT** see the contents of your crafted page! However, a split second due to browser rendering is acceptable.

Deliverables

- t1.html
- report.pdf (optional)

Sample t1.html deliverable

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>XSRF</title>
  </head>
  <body onload="document.forms[0].submit()">
    <form action="" onsubmit="" method="POST">
    <!--
      Your exploits here
      You may also want to change some form attributes
    -->
    </form>
  </body>
</html>
```

report.pdf format

Target 1 Partial Credit

The vulnerable code is in <filename>:<line>

<Explanation of why the code is vulnerable>

<Provide screenshot if applicable>

Milestones

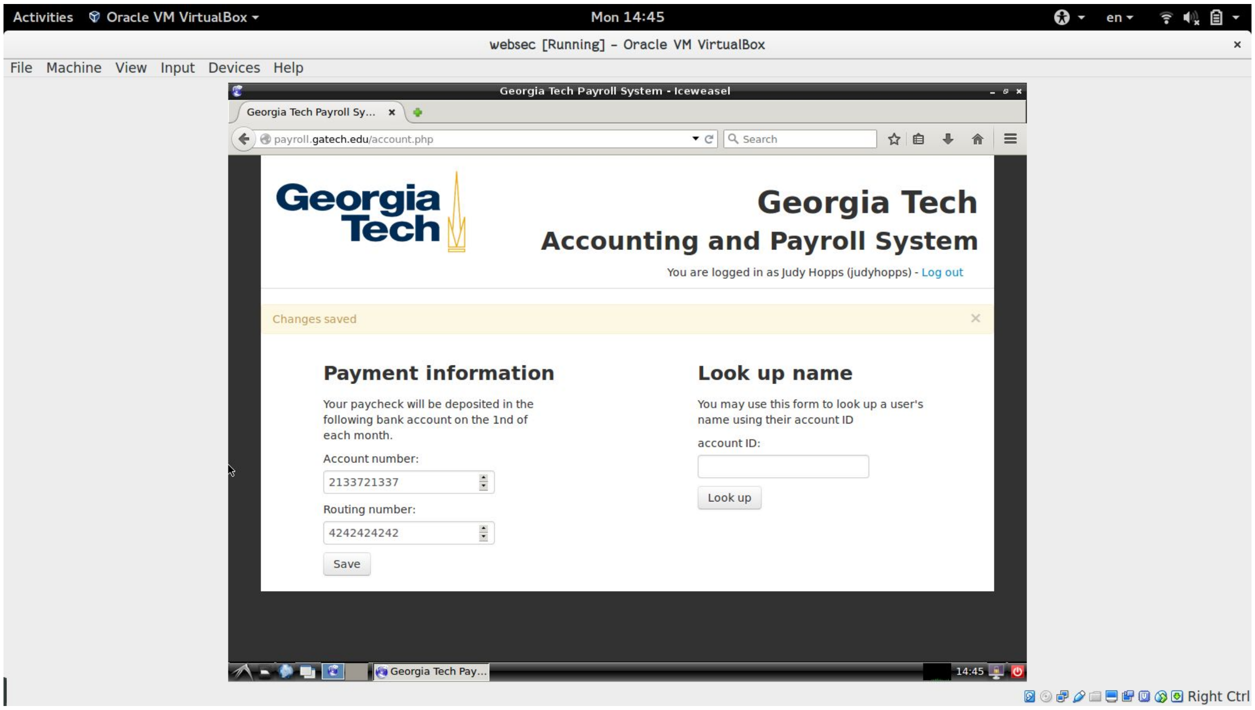
A successful attack earns 20 points automatically.
If you are unable to complete the task, you will earn partial credit as follows:

Points	Milestone
6	Properly identify the vulnerability and explain why it is vulnerable. Provide your response in report.pdf.
7	You see the “XSRF prevented” message with your exploit.
7	Able to change the account number and routing number without extra browser tabs or popups.

Note

You can visit your web page by entering the path of your file in the browser URL bar. For example, this would be **file:///home/user/t1.html** assuming that your exploit lives in **/home/user/**.

Example of Successful Exploit


Activities
Oracle VM VirtualBox
Mon 14:45
en
websec [Running] – Oracle VM VirtualBox
File Machine View Input Devices Help
Georgia Tech Payroll System - Iceweasel
Georgia Tech Payroll Sy...
payroll.gatech.edu/account.php
Search
Georgia Tech
Georgia Tech Pay...
14:45
Right Ctrl

Georgia Tech
Accounting and Payroll System

You are logged in as Judy Hopps (judyhopps) - Log out

Changes saved

Payment information

Your paycheck will be deposited in the following bank account on the 1nd of each month.

Account number: 2133721337

Routing number: 4242424242

Save

Look up name

You may use this form to look up a user's name using their account ID

account ID:

Look up

Target 2: XSS Password Theft (30 points)

You got caught! The good news is that Georgia Tech InfoSec is curious if you can find another vulnerability that is more severe. They will let you off the hook if you help them out. You noticed that you are able to steal a user's browser cookies. You can craft a web page such that whenever a victim, say Bob, visits the page, it will redirect (NO popups) him to **http://payroll.gatech.edu/**

The web page should look as if Bob visited the site directly. When Bob types in his username and password and hits the login button, an email with all of his browser cookies will be sent. Georgia Tech administrators would like you to demonstrate the attack and pay you accordingly. You will have to send the email to the local **user** account on the virtual machine as a proof of concept.

This attack requires an email to be sent to **user** on the system. The good news is that you can use **hackmail**:

```
http://hackmail.org/sendmail.php
```

Open this URL from within virtual machine for instructions on how to send emails via your attack script. Any mail that the **user** account receives will appear in **/var/mail/user**

Requirements

- The attack must be performed using XSS. Providing a phishing web page will result in 0 points. The browser URL bar should contain the domain **payroll.gatech.edu** and not a phishing URL.
- The email payload should be ALL of the user's browser cookies for the payroll.gatech.edu domain unformatted. Do not perform any encoding and/or special formatting.
    - The sender of the email should be set to **haxor**.
    - Failure to follow this format will result in 0 points for this part.
- The redirected page must be **cosmetically identical** to the original page. The web page source can be different as long as the user cannot tell without looking at the source.

Deliverables

- t2.html
- report.pdf (optional)

Sample t2.html deliverable

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>XSS</title>
  </head>
  <body onload="document.forms[0].submit()">
    <form action="" onsubmit="" method="POST">
    <!--
      Your exploits here
      You may also want to change some form attributes
    -->
    </form>
  </body>
</html>
```

report.pdf format

Target 2 Partial Credit

The vulnerable code is in <filename>:<line>

<Explanation of why the code is vulnerable>

<Provide screenshot if applicable>

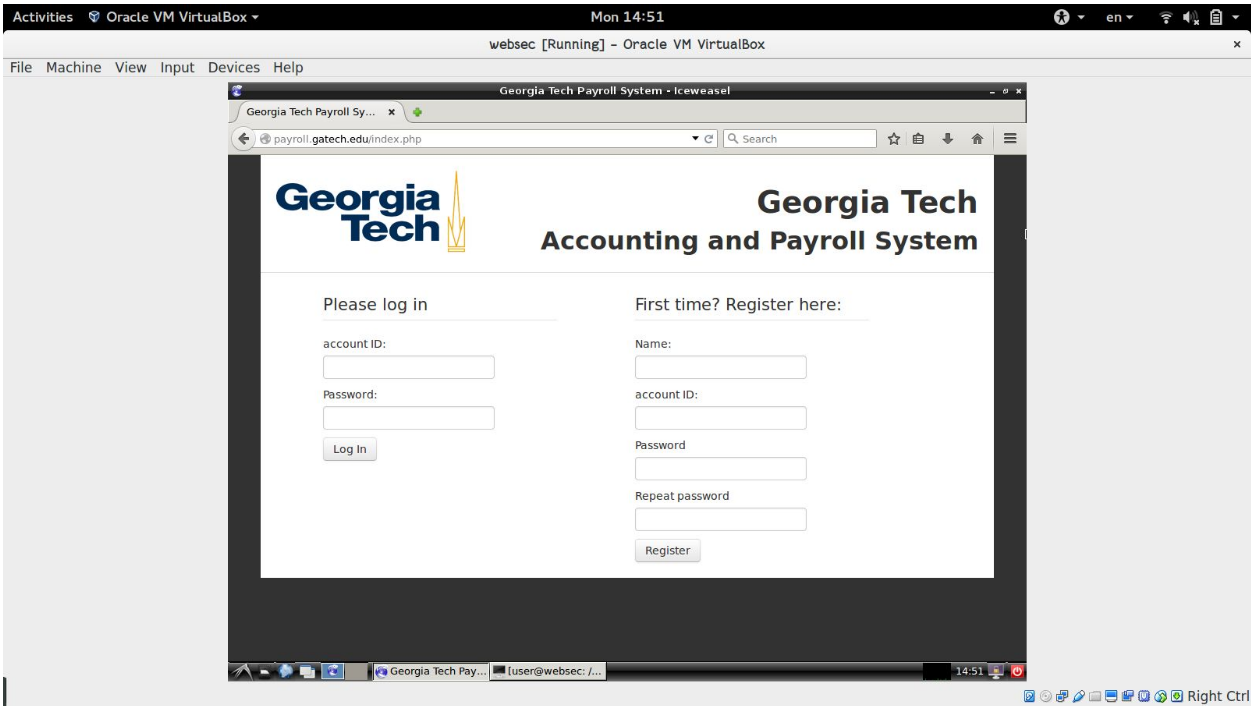
Milestones

A successful attack earns 30 points automatically.
If you are unable to complete the task, you will earn partial credit as follows:

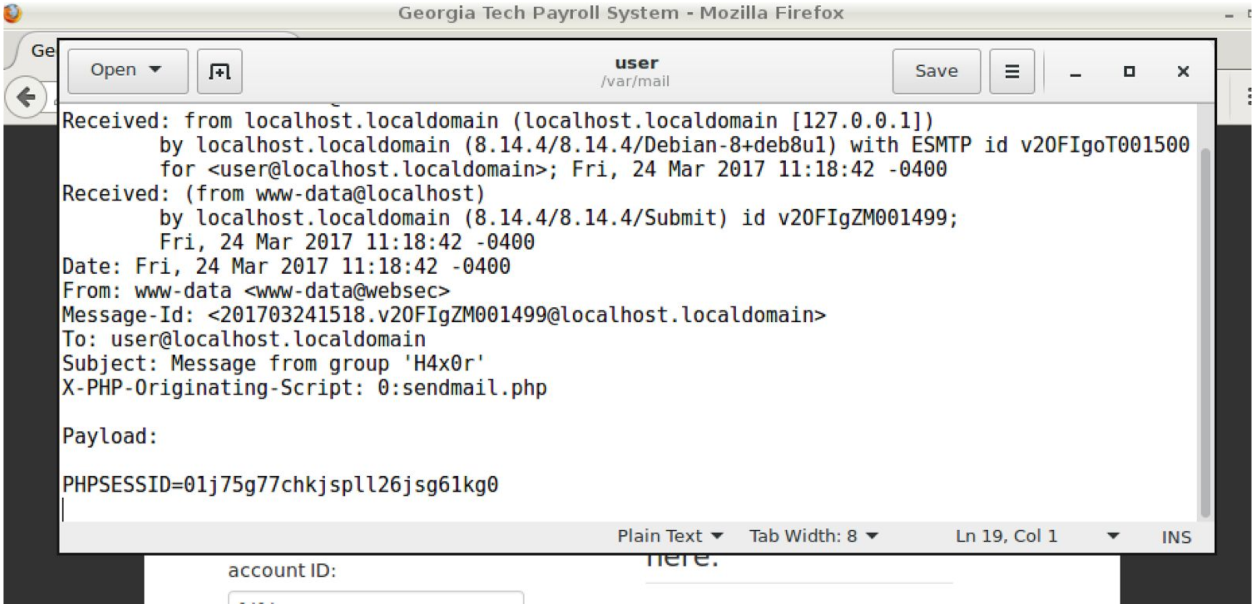
Points	Milestone
6	Properly identify the vulnerability and explain why it is vulnerable. Provide your response in report.pdf.
17	Steal the user’s browser cookies and send them to the **user** account via email.
7	The exploited web page is cosmetically identical to the original website.

Example of Successful Exploit

After visiting t2.html, the web page should look exactly the same as the legitimate site.



After user types in his or her credentials, the login should succeed, and an email should be sent.



```
Received: from localhost.localdomain (localhost.localdomain [127.0.0.1])
        by localhost.localdomain (8.14.4/8.14.4/Debian-8+deb8u1) with ESMTP id v2OFIgoT001500
        for <user@localhost.localdomain>; Fri, 24 Mar 2017 11:18:42 -0400
Received: (from www-data@localhost)
        by localhost.localdomain (8.14.4/8.14.4/Submit) id v2OFIgZM001499;
        Fri, 24 Mar 2017 11:18:42 -0400
Date: Fri, 24 Mar 2017 11:18:42 -0400
From: www-data <www-data@websec>
Message-Id: <201703241518.v2OFIgZM001499@localhost.localdomain>
To: user@localhost.localdomain
Subject: Message from group 'H4x0r'
X-PHP-Originating-Script: 0:sendmail.php

Payload:

PHPSESSID=01j75g77chkjspll26jsg61kg0
```

Target 3: SQL Injection (20 points)

H4x0r0rg has heard about your feat in making tons of money from Georgia Tech by changing other people's payroll account. They contacted you and gave you a job, a job with a hefty sum you cannot resist. Your task is to create an HTML webpage, and the requirements are:

- The crafted page has a text field for the username and a submit button.
  - NO password field!
- The user of this page is not logged into Georgia Tech payroll system, but when he or she enters a valid Georgia Tech payroll registered username (for example, judyhopps) and hits submit, the user is redirected to **http://payroll.gatech.edu/account.php** and logged in as judyhopps.
- Do NOT execute destructive SQL commands such as DROP tables. System administrators can easily detect data loss!
- The id of the input field must be set to **targetlogin**, and the button id must be **exploit**. Example:

```
<input name="login" id="targetlogin" value="username" />
<button id="exploit">Hold onto your butts!</button>
```

Deliverables

- t3.html
- report.pdf (optional)

Sample t3.html deliverable

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>SQL Injection</title>
    <script>
      <!--
        Your exploits here
      -->
    </script>
  </head>
  <body>
    <form action="" onsubmit="" method="POST">
      <input name="login" id="targetlogin" value="username" />
      <button id="exploit">Hold onto your butts!</button>
    </form>
  </body>
</html>
```

report.pdf format

Target 3 Partial Credit

The vulnerable code is in <filename>:<line>

<Explanation of why the code is vulnerable>

<Provide screenshot if applicable>

Milestones

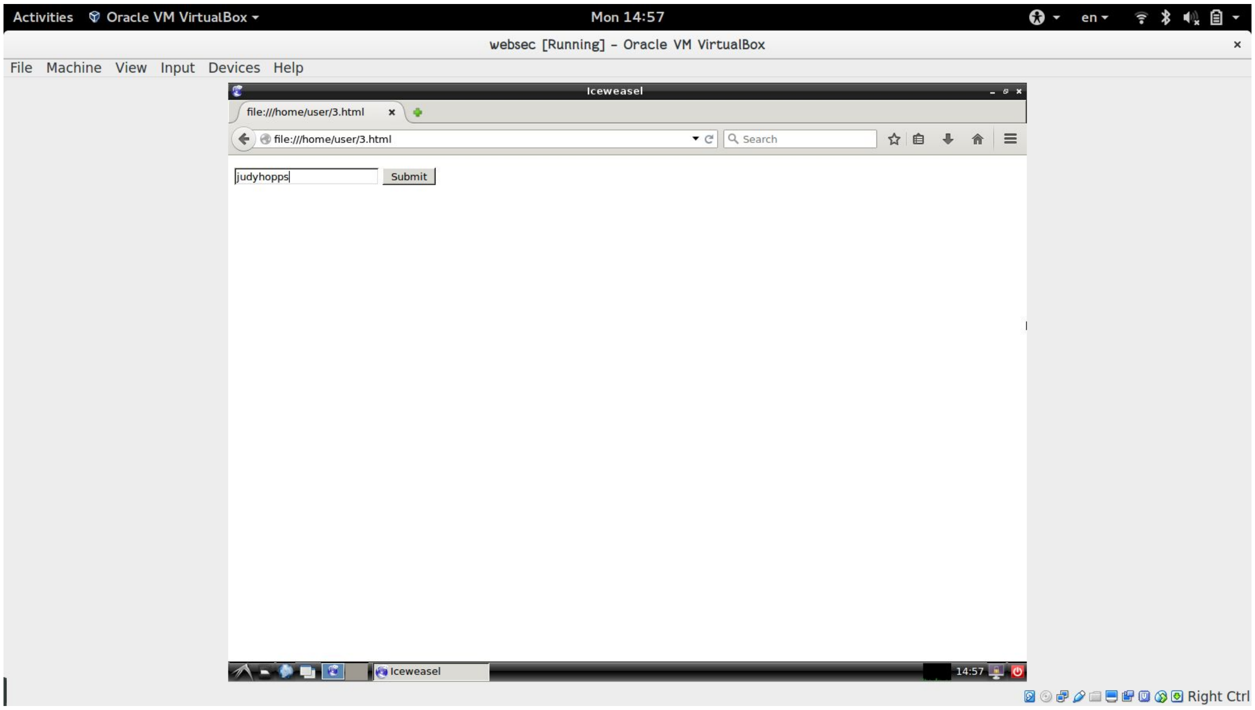
A successful attack earns 20 points automatically.
If you are unable to complete the task, you will earn partial credit as follows:

Points	Milestone
5	Properly identify the vulnerability and explain why it is vulnerable. Provide your response in report.pdf.
10	Steal the username and password and send the credentials to the **user** account via email.
5	The exploited web page is cosmetically identical to the original website.

If you implemented the attack with a destructive SQL command that causes our scripts to fail to grade your other targets, you will not be getting any partial credit for those failed targets.

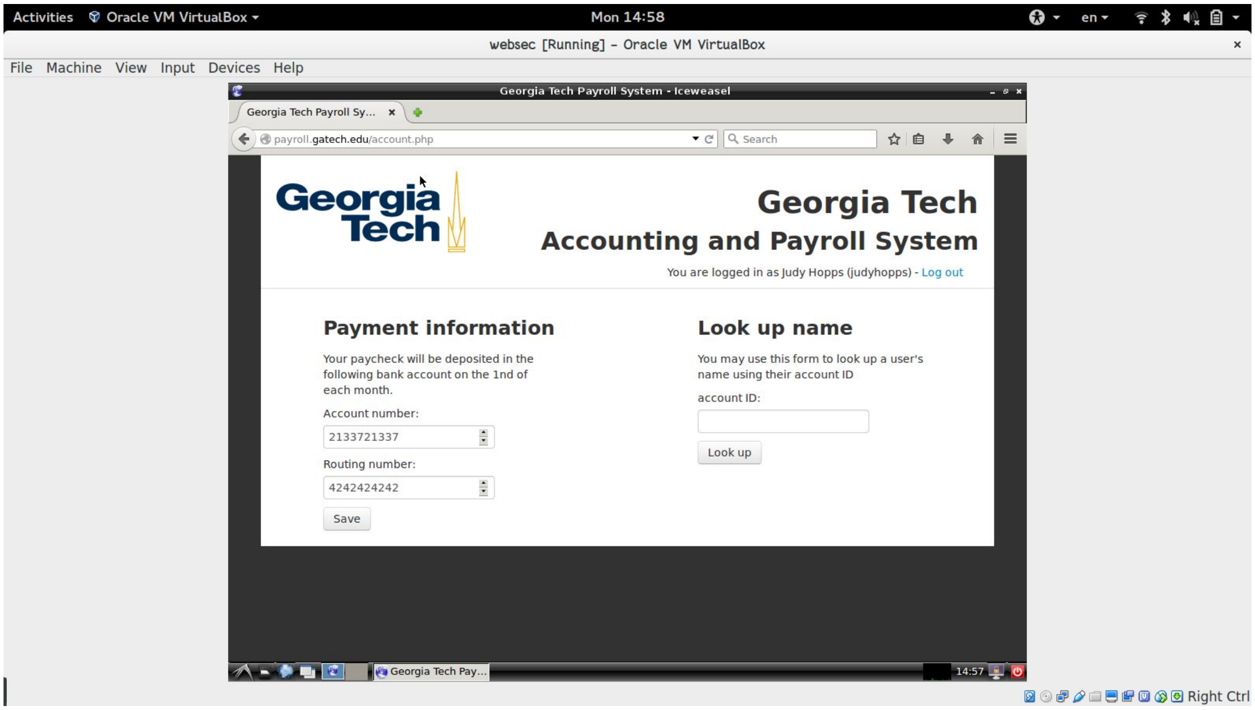
Example of Successful Exploit

After visiting t3.html, the page displays an input field for the attacker. Please feel free to prettify it as long as it complies with the requirements.



judyhopps

After typing in the username of an existing user in the payroll system, you should be successfully logged in. The site should function as if logged in legitimately.



payroll.gatech.edu/account.php

Georgia Tech

Accounting and Payroll System

You are logged in as Judy Hopps (judyhopps) - Log out

Payment information

Your paycheck will be deposited in the following bank account on the 1nd of each month.

Account number: 2133721337

Routing number: 4242424242

Save

Look up name

You may use this form to look up a user's name using their account ID

account ID:

Look up

Epilogue (30 points)

You delivered your exploit to H4x0r0rg. They seemed quite happy, and so are you. Just the thought of not having to work for the rest of your life seems quite enticing. However, you suddenly hear the FBI knocking on your door. It turns out that H4x0r0rg was just a law enforcement honeypot!

The FBI is willing to let you off the hook this time, but only if you do some work in restitution first. For each of the three targets, describe briefly in **report.pdf** how to fix the vulnerabilities that you found so that they can no longer be exploited. You do not need to include actual patched code. However, your descriptions should be sufficiently detailed that they would be actionable.

Deliverables

- report.pdf

report.pdf format

Target 1 Epilogue

<Explanation of why the code is vulnerable and how to fix it>

<Provide screenshot if applicable>

Target 2 Epilogue

<Explanation of why the code is vulnerable and how to fix it>

<Provide screenshot if applicable>

Target 3 Epilogue

<Explanation of why the code is vulnerable and how to fix it>

<Provide screenshot if applicable>

Helpful Readings and Hints

This assignment requires submitting forms. If you do not know how to do so, you may consult http://www.w3schools.com/html/html_forms.asp

This assignment requires writing JavaScript. Only a very basic knowledge of JavaScript is needed. You may find http://eloquentjavascript.net/ useful if you are completely new to JavaScript.

You do not need to have extensive SQL knowledge to complete Target 3. However, it requires some observations and thinking.

The sample HTML deliverables are there for your benefit and convenience. You are not required to follow the format unless specifically called out in the target. As long as the exploits work according to the requirements, you will receive full credit.

Acknowledgement

Special thanks to Professor Vitaly Shmatikov for the inspiration of this project and his permission to modify and reuse his materials. You may find more about him and his research at: http://www.cs.cornell.edu/~shmat/