

Laboratory Activity: Identifying Social Engineering Attacks

Data File

Identifying Security Threats\social_engineering_script.ps1

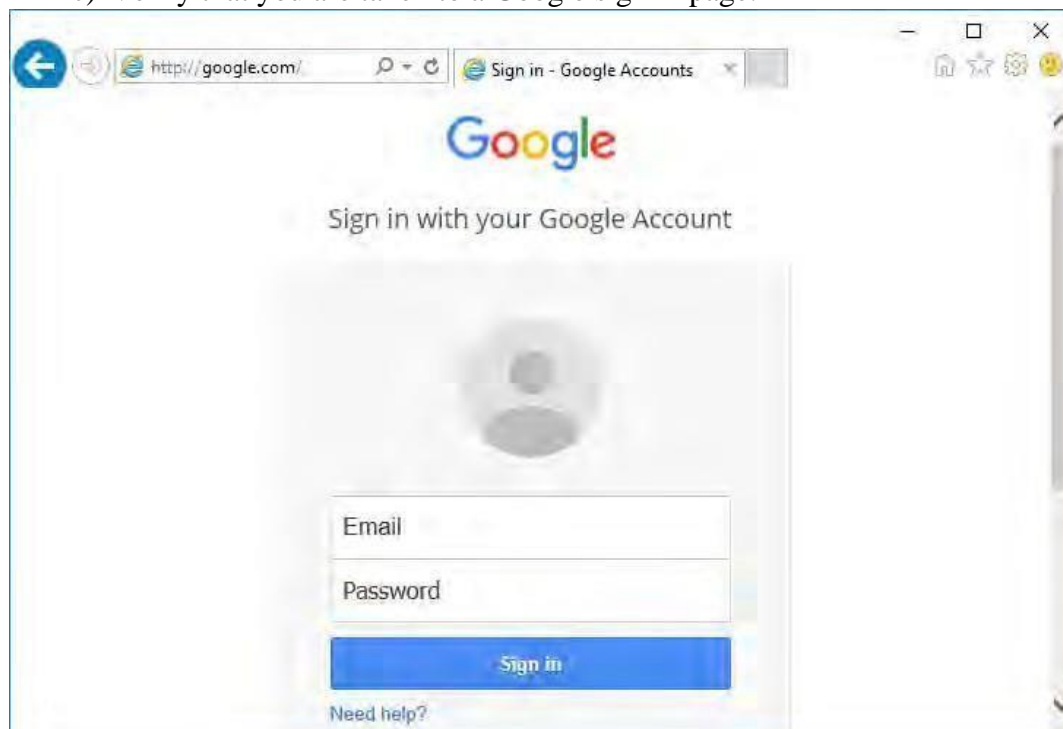
Scenario

You've received reports of users complaining that someone has hacked their Google accounts. The users claim that this "hack" occurs when they go to sign in to Google. You decide to investigate the situation to determine the issue.

-
1. Simulate the attack being set up on the server.
 - a) Navigate to folder **Identifying Security Threats**.
 - b) Right-click **social_engineering_script.ps1** and select **Run with PowerShell**.
 - c) At the prompt, type **y** and press **Enter** to change the execution policy.
 - d) Press **Enter** to exit.

This script sets up the simulated attack that you'll see in a moment.

2. Investigate the issue by logging in to Google.
 - a) Open Internet Explorer.
 - b) In the address bar, type **google.com** and press **Enter**.
 - c) Verify that you are taken to a Google sign in page.



- d) In the **Email** text box, type **youremail@gmail.com**
- e) In the **Password** text box, type **!Pass1234**
- f) Select **Sign in**.
- g) Verify you are taken to a page that indicates your account credentials are compromised.

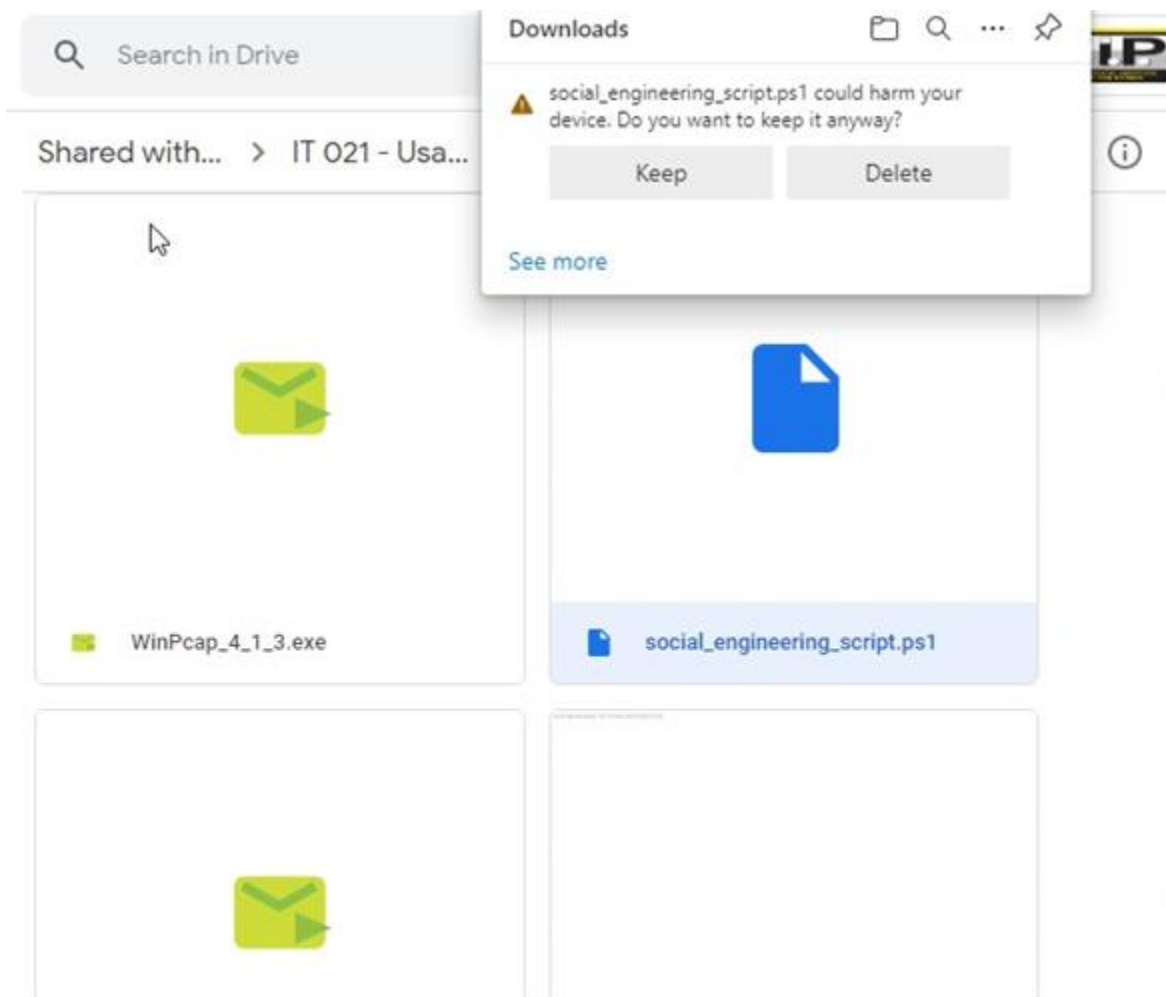


Even though you entered a legitimate URL, you were taken to a page that only appeared to be the Google sign in page. In reality, it was designed to be a convincing fake that would trick users into entering their Google credentials.

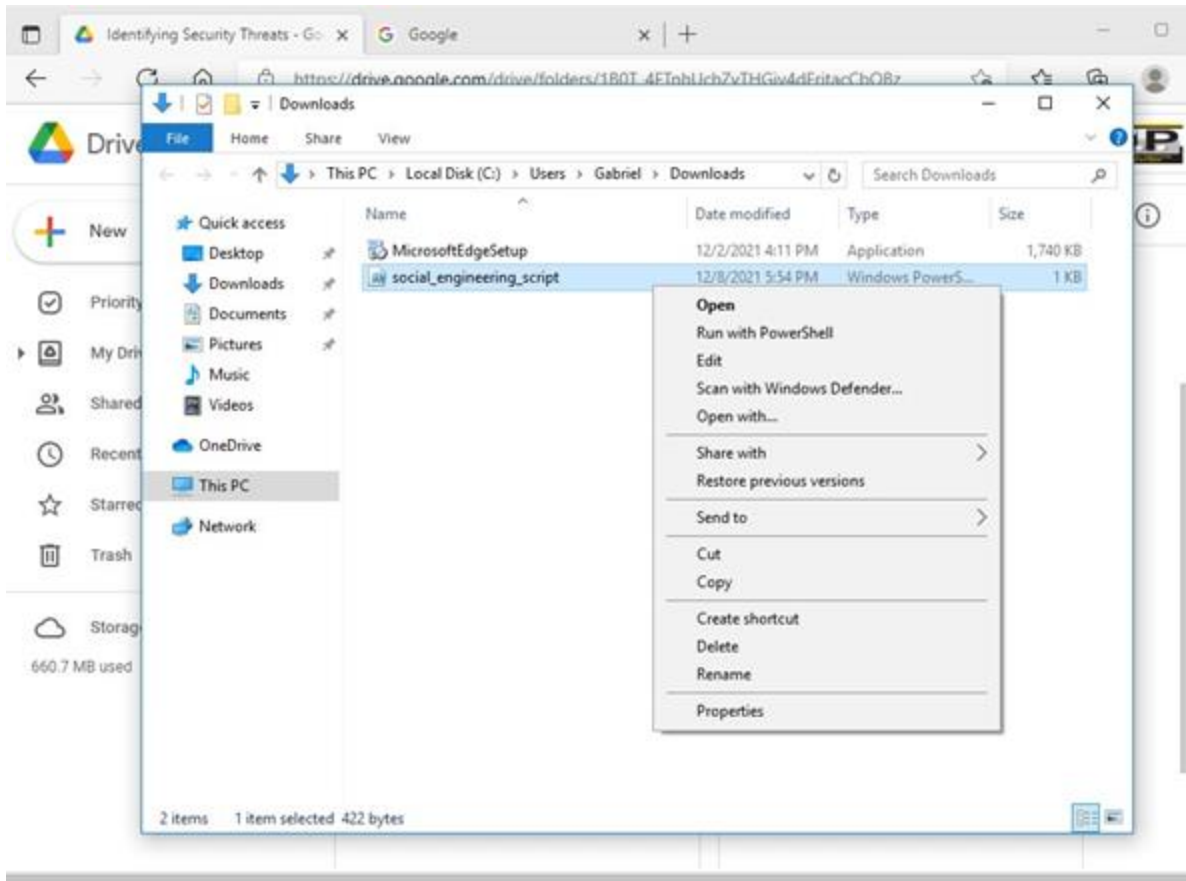
- h) Close the browser.

Instructions:

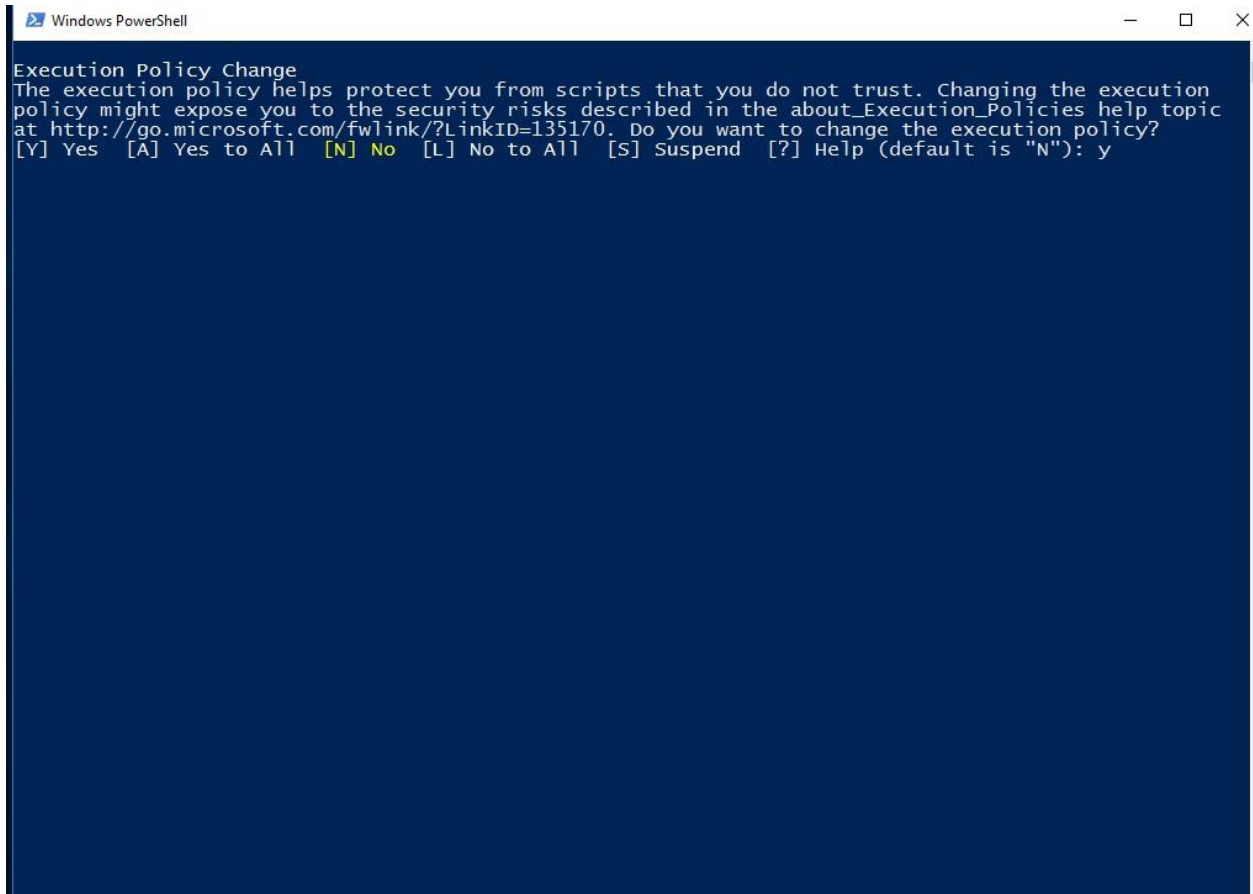
1. Document this process/output through screenshots and provide brief description of it.



- First, we downloaded the script in the google drive. This script will run the attack that will happen to your virtual workstation.



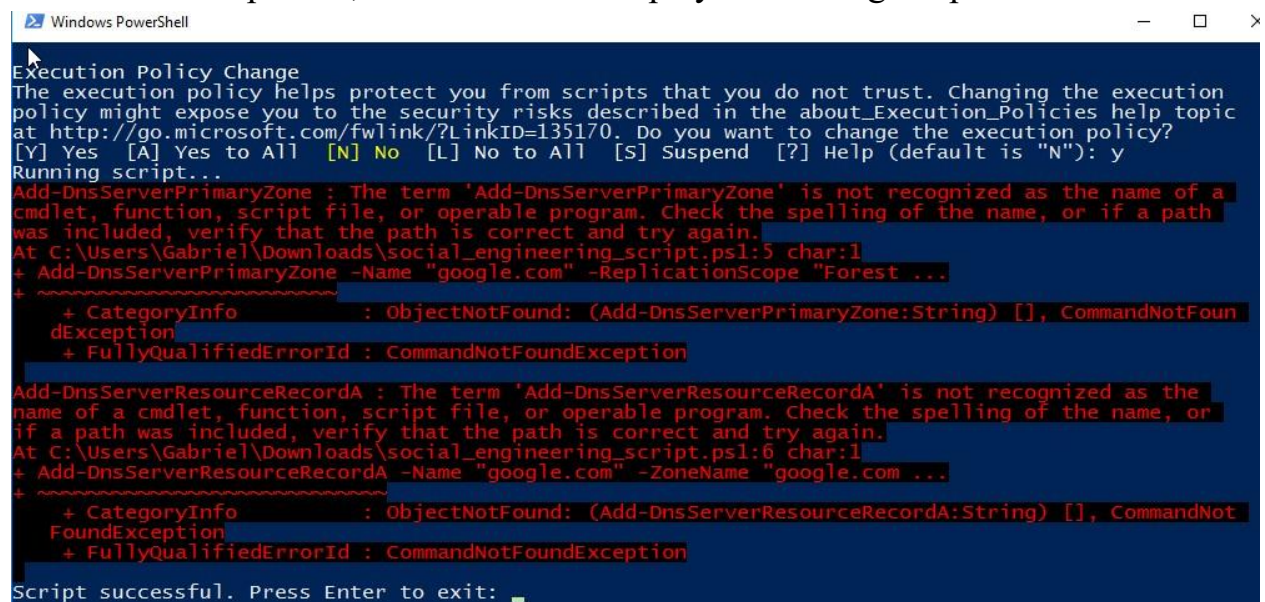
-After downloading the script, we located it in downloads and run it with PowerShell. By doing this, it will execute the attack shortly after the script runs successfully.



-After running it will prompt you to the PowerShell and display these choices.

```
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution
policy might expose you to the security risks described in the about_Execution_Policies help topic
at http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): y
Running script...
```

-After pressing y which is Yes, it will execute the script. As you can see in the picture, the PowerShell displayed Running script...

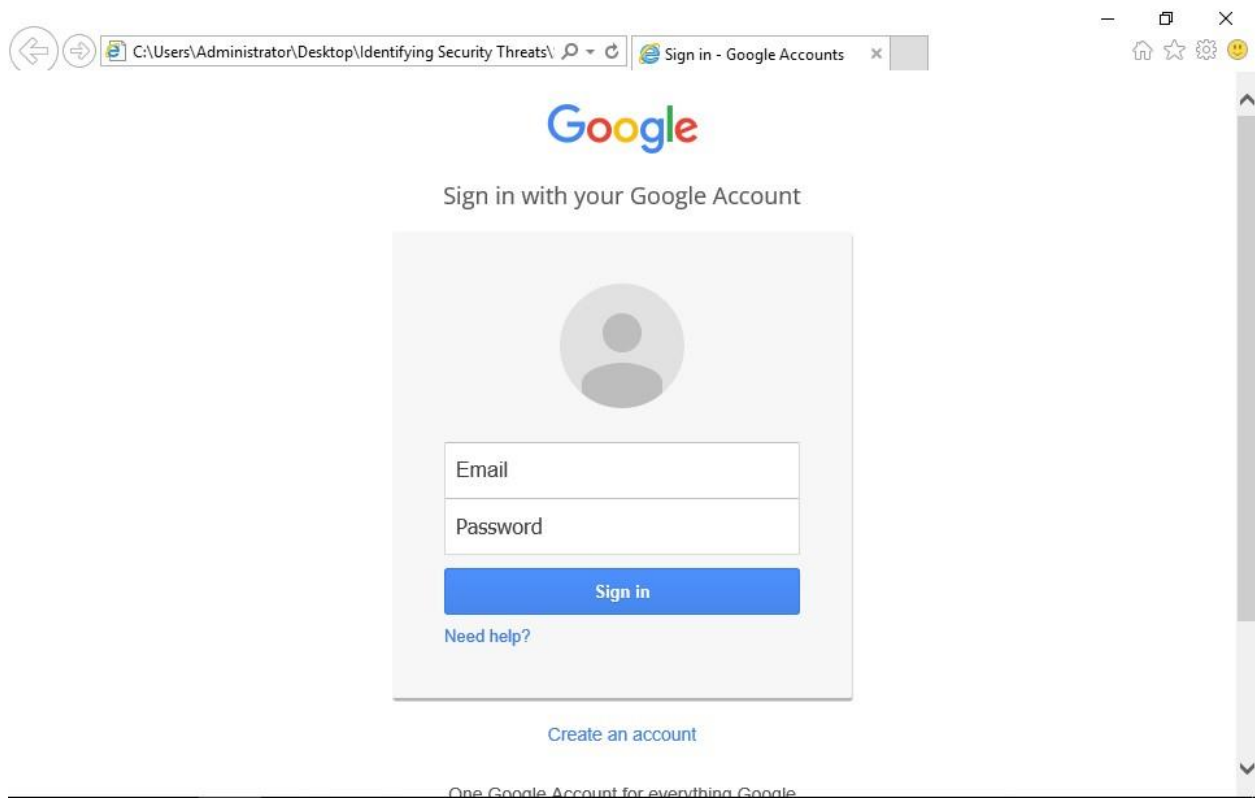


```
Windows PowerShell
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution
policy might expose you to the security risks described in the about_Execution_Policies help topic
at http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): y
Running script...
Add-DnsServerPrimaryZone : The term 'Add-DnsServerPrimaryZone' is not recognized as the name of a
cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path
was included, verify that the path is correct and try again.
At C:\Users\Gabriel\Downloads\social_engineering_script.ps1:5 char:1
+ Add-DnsServerPrimaryZone -Name "google.com" -ReplicationScope "Forest ...
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Add-DnsServerPrimaryZone:String) [], CommandNotFoun
dException
+ FullyQualifiedErrorId : CommandNotFoundException

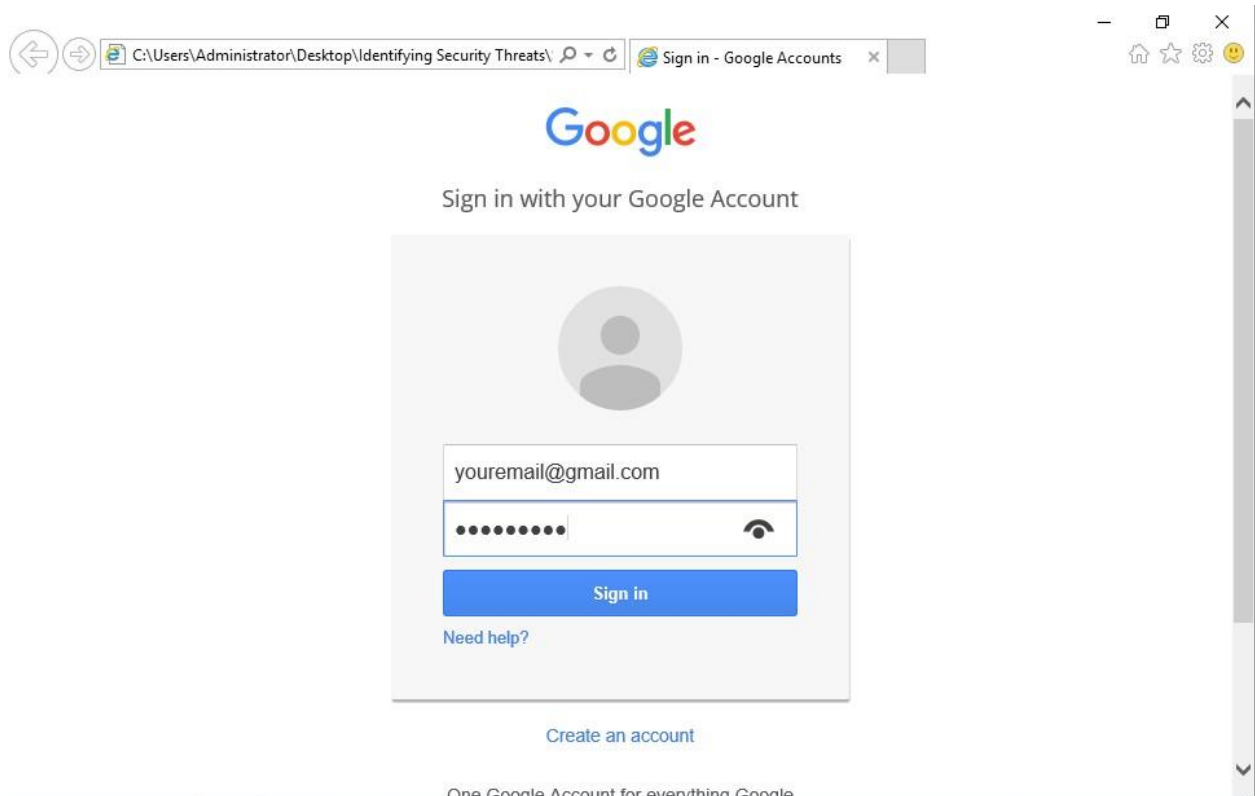
Add-DnsServerResourceRecordA : The term 'Add-DnsServerResourceRecordA' is not recognized as the
name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or
if a path was included, verify that the path is correct and try again.
At C:\Users\Gabriel\Downloads\social_engineering_script.ps1:6 char:1
+ Add-DnsServerResourceRecordA -Name "google.com" -ZoneName "google.com ...
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Add-DnsServerResourceRecordA:String) [], CommandNot
FoundException
+ FullyQualifiedErrorId : CommandNotFoundException

Script successful. Press Enter to exit: _
```

-After running the script this will be displayed in the PowerShell. This proves that the script run successfully.



-After closing the script, we went ahead and typed the google.com in internet explorer's address bar. After loading, we were asked to login our google account.



-We used youremail@gmail.com as username and !Pass1234 as password so our original account will not be compromised.



You've been tricked!

Thanks for your user name and password!



-After entering our login information this type of text displayed.

"We affirm that we have not given or received any unauthorized help on this assignment/practice exercise, and that this work is our own."