Steve Erdelyi & William Williams

Iman Vakilinia

CS 445: Internet Security

May 10, 2019

Go Phish

For our CS 445 project we decided to explore and develop our own phishing attack. These attacks are common-place in our modern technology based world where many of our day-to-day activities on the internet involve some sort of login credentials. These login credentials are the focus of phishing attacks and are often directed towards getting bank account, social media, or other high-profile website credentials. For our project, we decided to create a phishing attack that would attempt to retrieve the credentials of a person attempting to login to UNR online resources such as myNevada or Webcampus. The basic idea of our attack is that the target will be directed to a fake UNR NetID page from a phish email. The user would then input their UNR credentials into this fake NetID login page in which we will then have their credentials and save them into our database. Since these same credentials are often used to login-to most of the UNR services the attacker would then be able to gain access to the target's UNR online services using these stolen credentials.

To create the fake UNR NetID login page, we used the same UNR logos and color theme to design our own webpage that resembled that of the real NetID page as closely as we could. For this task we utilized HTML along with PHP to create a webpage that without close observation, would easily pass for the actual NetID login page. Since this

is simply a simulation of how an attack could be played out, we decided it would be best practice for us to also include a message on our fake NetID page that indicated that it was indeed fake and for only demonstration purposes of our phishing attack. Besides this small addition, our fake NetID mimicked the real-one exactly including the text entry boxes and submission button along with working forgotten password and NetID links. The scripts for our fake UNR NetID page can be viewed on our Github repository under the unr-sso folder.

We also created a web interface to run our sample phishing attack from. This web interface allowed us to send the phishing email with the phish link to the fake NetID login page. This web interface was connected to the database where we held the emails that are to be used for the phishing attack. This file can be viewed on the Github repository in the folder titled process and file name is emails-ticker.php. Another web interface was designed to display the contents of the database such as the username, password, email, and timestamp of all the phishes. This file is in the process folder and is titled phishie-ticker.php. For our simulated phishing attack, we used a free database program created by Sergey Tsalkov to satisfy our database needs. For more information on this program, the code is on the github repository in the folder process and is titled meekrodb.2.3.class.php.

The main web interface to run the simulated phishing attack is titled index.php and can be found on the main page of the Github repository. This script creates a webpage that allows the attacker to specify the email to send the phish to. The phish is then sent to the email that is inputted. When the target clicks the link and puts there credentials into

the form, those credentials would then be available to be viewed by the attacker. We have left opportunity for expanding our sample phishing attack to Facebook, Twitter, and Instagram. To do this we would need to create web pages that emulate those that are used to login to these popular social media platforms. The functionality to allow the attacker to specify the type of phish to send to the email is currently deactivated in this main web interface.

Steve's background in web design made this an interesting path to pursue due to the prevalence of these types of attacks in our modern time. Due to William's lack of knowledge in the realm of web design and its languages, he approached this assignment as an opportunity to learn the languages and how to use them. So that only clean, working, functional code was uploaded onto the Github repository, Steve and William worked on parts of the project as a team. Steve uploaded all of the code onto the repository after he ensured its functionality and operability. By doing this Steve caught and fixed many mistakes that William made while developing the code for this project. After completing this project, Steve has gained experience on how a phishing attack is created and executed while William has gained experience in creating web pages along with working with experience in using a database.