Phishing Attacks and 2FA: Moving Towards a More Secure Authentication Infrastructure

By Sam Driver and Matt Querdasi

In today's modern world, Cyber Security is an ever-growing field that is constantly evolving. The proliferation of ways to share and store information online has created a veritable gold mine for cyber attacks and malicious activity. One of the newest and most popular developments in the cyber security field is two-factor authentication for password protection. In this study, we conducted interviews, sent out a survey to the Wesleyan community, and did some practical examination into two factor authentication.

One of the primary types of cyber-attacks on the modern web is a phishing attack. Phishing is a type of social engineering attack, often used to steal login passwords, or credit card details. For our purposes, we will be discussing phishing attacks that are designed to glean login passwords, as this will allow us to discuss more fundamental issues with current methodologies for authenticating users. One of the most common phishing attacks are spoof emails. For instance, someone would receive an email stating that there is something wrong with their account, and that they should follow the attached link and log in. The link would lead to a fake login page, and when the victim logged in with their username and password, it would be recorded so that the attacker could use it to access the account at their leisure. There are countless variations on this attack but that is a general example, and we believe that it shows the problem with the traditional login structure. A stolen password means a stolen account.

Before moving on to our experimental findings, it is important to establish exactly what two factor authentication is. When one normally logs in to an online service, they are only required to enter a username (or email) and a password. This is considered single factor authentication. The problem with this, is that passwords are generally unreliable. People are tempted to use short passwords that they can easily remember, re-use passwords between sites, and write down passwords in places that may not be secure. All of these present vulnerabilities for information stored on password protected sites. One way to fight against this is to enforce strict password protocols (the National Institute of Standards and Technology have recently released a new set of such protocols). What we propose to be, and what we believe has shown to be, a better solution is to move towards multi factor authentication. This would mean that in order to log in with a given username, the user would be required to not only insert their password but also provide some secondary form of proof of account ownership. Some examples of a second factor are: a code provided through text/email/an app, some form of biometric data (fingerprint/face scan), a magnetic stripe card, etc. Requiring a password and one of these would be considered two factor authentication, which is what most sites have been moving towards. However, sites that require even more protection, most notably government sites, are moving towards three or more factors for authentication.

The first step in our process towards understanding two-factor authentication was to conduct an interview with Wesleyan's CIO (Chief Information Officer), Joseph Bazaely. He described his job as "keeping Wesleyan out of the press." This includes making sure that no sensitive information is leaked through electronic means. He let us know that Wesleyan is currently undergoing a process of integrating a two-factor login and walked us through a lot of the steps that make up that process. Essentially, there are three options for integrating institution wide multi factor authentication. The first option is to build your own system. Generally, this is not done, as it requires a lot of coding hours, and as we have discussed in class, it is rarely worth it to reinvent the wheel for security. As with encryption, It is better to try and use established and safe practices and products than make something yourself that could have unknown workarounds. The preferred approach for institutions is to purchase the service from one of two market leaders: Duo Authentication, or Microsoft Authentication. Based on our discussion and research, companies choose Microsoft if they are already integrated with other Microsoft products such as Teams and Outlook etc., otherwise Duo is the go to choice. Wesleyan will be using Duo Authentication for their 2FA service.

It was made clear to us that there are a lot of costs behind 2FA that might not be expected by the average user. We will discuss some of them with Duo as the model. First of all you have to buy a license, allowing your institution to register a certain number of accounts with the Duo service. The costs don’t stop there however. If you want to use text messages to deliver the codes to users, there is a cost of a couple cents for each time a code is sent. This may seem small at first, but with several thousand users all signing in several times a day, some of which incurring an extra cost for being international numbers, it can add up to a significant amount of capital. Text message authentication can also suffer from sim cloning attacks, where text messages can be intercepted by a third party. The solution to this is to not allow text messages, and require the second factor to be a code provided through the Duo app. This has its own issue however, as It requires a smartphone, which not all members of the institution may own. In this case you can provide physical cards to those members of your institution that will generate a code whenever needed, but those aren't free either.

Wesleyan's solution to keep costs low is to only provide two-factor authentication for faculty and employees, but not students. The reasoning here is that generally, employees will have access to more vulnerable information and data (large number of: grades, health records, etc.) than students will. By keeping the amount of registered accounts low, Wesleyan can afford to provide physical cards to those who need them, as well as can keep the overall price for their subscription to the service on the lower side. While it may seem an imperfect solution to only protect the employees and not the students, it is important to remember that Wesleyan's email is implemented through Gmail. Google provides two-factor authentication for Gmail free of charge, so Wesleyan is recommending that concerned students turn on that option for their email accounts.

The other topic that we discussed with Joe was the general attitude towards two-factor authentication within the Wesleyan community. Apparently, Wesleyan has been discussing plans to transition to a more secure authentication architecture for several years, but kept putting off making any sweeping changes. Once Joe Bazaely was hired last year, he made it a priority to make this a higher priority for the school. He said that it is easier to convince people to implement a more secure authentication infrastructure when you frame it in regard to what it protects. If you tell someone you want to switch to multi-factor to prevent stealing pictures of their dog, they won't take it seriously, but if you make it clear that this is protecting their paycheck they will work with you to improve it. He said in recent years attitudes have become more accepting and knowledgeable about two-factor authentication, but most people who use it, don't always understand what it is. There is a common misconception within the public that having two passwords would be an implementation of two-factor authentication. Clearly, this is not the case. Two-factor requires two different factors, not two of the same. The more secure that the two factors you choose to use are, the more secure your implementation will be, but obviously you have to balance security and usability. The NIST provides guidelines for password security, and one can only assume that as more research on multi-factor is performed, organizations like the NIST will provide guidance towards more secure multi factor authentication as well.

Data

Through a research questionnaire, we were able to learn a tremendous amount about the greater Wesleyan populations attitudes towards and knowledge of cyber security. When participants were asked whether they knew what a phishing attack was 63% responded that they did, and provided what they believed to be an accurate definition, almost all of which were in fact correct. In contrast, 90% of participants said that they knew what two-factor authentication was. However, some of the provided definitions for two-factor authentication were inaccurate. Most commonly people claimed that having two passwords, instead of one, counted as two-factor authentication which it does not.

The next section of our form, quizzed the participants using 4 screenshots of webpages with edited URLs to simulate a phishing attack. Averaged out, participants were correct 62.5% of the time. This lines up with our original breakdown of people who knew what a phishing attack was, which was 63% of people. This lets us conclude that generally, people who knew what a phishing attack is are able to detect when a site has been tampered with, and those who did not were prone to falling victim. Overall, the site with no changes and the Wescam site with a url with ".biz" in it had the highest percentage of correct answers on whether or not they were safe to visit.

Next, we asked a yes or no whether the participant would be in favor of wesleyan switching to two-factor authentication for all users. Surprisingly to us, and contrary to what our discussion with Joe Bazaely had led us to believe, 62% of participants said they would be against the switch. This seems to contradict what we had heard about the growing positive outlook on two-factor authentication. Due to the anonymity of our form, we are unable to follow up with any of these participants, but we would be curious whether the people saying no were students or faculty, and to get more detailed information as to why they would be against wesleyan switching. There is a possibility that respondents believed that this would be a mandatory switch, despite the fact that was not outlined in the question. We would be curious to see whether giving people optional 2FA would lead to a more favorable response. A two thirds majority is a significant opposition, and in our opinion is worthy of Wesleyan discussing this change more openly with the Wesleyan population.

In the last section of our form we presented participants with several opinion questions, and allowed them to gauge their agreement with the statement from 1-10. When asked how important cyber security is to them, participants responded an average of about 7.5. This is higher than we had expected. Our initial instinct was that the average would be around a 5 or 6, and those heavily involved with technology would rank it higher, but it seems that cyber security is on the front of more people's minds than expected. There was also an interesting contrast to be found when participants were asked how frightened they were of being the victim of a cyber attack. Results for that question were very polarizing, but ended up with an average of 4.9. There were a significant number of 3s and also a significant number of 7s. This leads us to believe that either people think they are safer than they are, or are informed enough about their security that they feel secure. Based on the high average for how often people think about cyber security, we are leaning towards the idea that people who are informed tend to be less afraid. Our final question was a yes or no response as to whether the participant had been a victim of a cyber attack. 20% of respondents said that yes, they had been a victim of a cyber attack. This could be one of the reasons that we had so many high fear responses, as it stands to reason that if you have been a victim once, you would be more frightened of being a victim again.

Overall, our data was very enlightening. We were slightly limited by the anonymity of our survey, and would be interested in discussing some of these questions at a greater length with members of the wesleyan community if our research was to go further. Some visuals of the data can be seen on the site.

Website

The website portion of the project was created to give us hands on experience implementing two factor authentication, and help us learn more about the implementation process. The site offers a login and register feature similar to most account based sites, and upon both registration and login requires you to scan a randomly generated QR code with the Google Authenticator app to generate a six digit code. Once you correctly enter the Google Authenticator code, you will then be directed to our sites homepage, which contains all of our project information.

The two-factor authentication element of the site uses the Google Authenticator library that regenerates a six-digit secure pin every 30 seconds for maximum security using a QR code. We implemented this by using an open source library to generate a random string, then we input the string into a QR code generator to get a QR code that generates a six digit pin upon entry into the Google Authenticator app. We then check the users input code against our original string, using a function from our open source library to authenticate the user. All backend and database management was done by us, and we implemented the Google Authenticator library using an open source program called PHP Gangsta.

Our implementation of two-factor authentication has a few limitations. The main problem we ran into is that while providing essentially an added login with the QR code page, there is no way to confirm that the user is scanning the code with their device. For instance, if an attacker steals someone's username and password, they could simply scan the QR code on our site using the Google Authenticator app on their personal phone. In order to prevent this, we would’ve had to implement a Google specific login (i.e. users on our site enter their gmail credentials), and then link this to the Google Authenticator API. We discovered that this would potentially cost us a lot of money, as well as a timeline that was beyond our capacity. Therefore, our implementation simply references the Google Authenticator library, but in fact has no way to confirm the user between login pages.

As for the site itself, we created the website using HTML/CSS to style and create a shell for our login feature. For username and password storing, we used an SQL database that we query using PHP. For form submission and user input we use a combination of jQuery and Ajax. Although not the direct focus of the project, we hash user passwords upon registration for secure storage, and are string escaped to prevent SQL injection attacks. In addition, we implemented session cookies for all privilege level changes to prevent path traversal attacks.