Web Phishing: An Archive and Analysis

Kristen Morse Mentor: Calvin Ardi
kmorse@usc.edu calvin@isi.edu

Introduction

Web phish are websites created to look exactly like other commercial websites in an attempt to steal personal data from users of the internet: this may include credit card numbers, banking information, account login credentials, or any other information sensitive to the user's identity. The victims of web phishing attacks are customers and users of commonly known businesses, banks, and websites. In order to fool its victims, web phishing sites try to maintain a trustworthy appearance while stealing credentials and other information from a user through input forms on their site. They create the appearance of a login form on a company website that the user belongs to, influence the user to input information, and steal that information without the customer's knowledge. As these attacks are becoming more prevalent, it is important to create countermeasures in an attempt to protect the sensitive data of internet users. In order to do so, it is necessary to analyze how phish behave: what code is being written to copy the original website, what credentials are being stolen, and whether or not phish maintain a facade of trust after the initial submission of credentials by the user. The problem; however, is that phish sites are created, steal information, and are shut down within a short period of time. As a result, a more concrete tool for analysis of phish needs to be maintained in order for current researchers in the web phishing field to analyze the behavior of phish long after they expire. Currently, the best public tool for this is a website called Phishtank, which maintains a list of valid phish and provides a development starter kit for downloading a valid phish database including URL, submission time, verification time, whether or not the phish is verified, whether or not the phish is online, and the target company. It provides a user friendly UI for searching through snapshots of the original phish before it expires. While Phishtank provides meaningful data about phish and allows its users to download this content, it is necessary to add to the value of this tool by producing more meaningful data useful to phish analyzers and consistently maintaining an archive of this data that can be easily evaluated by developers. In addition, companies like Google create tools for avoiding phish, so it is likely an archive for phish already exists in the private sector; however, researchers do not have access to private company archives. It is necessary to create an open source tool for collection of phishing data. This research builds upon the Phishtank solution by creating a publicly available phish archive with Phishtank database information on each valid phish as well as: a screenshot of the phish, http responses, source code, and similar information about the next page after a credential submission. Next, a simple analysis is made of target companies and "click through," or login submission results. For this tool to be effective, the data within the archive needs to be enough for future researchers to evaluate. It needs to be updated as frequently as possible so that as few phish will expire in the time it takes to archive them. Furthermore, the code must be reconstructable, easy to read, and widely available to the public. These factors play into the archive's design. The data collected: source code, http response, and so on are all useful information for the developer/researcher to use when conducting analysis. Screenshots are useful to see what a phish looks like, source code is a key data element to use for determining how a phish creates its appearance, and http response is useful for determining a page's status. The timeline of collecting data is hourly: the Phishtank database can be pulled on an hourly basis. As a result, this work creates a script that can pull that data and scrape the phish hourly. Lastly, the scripts for this project are on a GitHub repository, open to the public for development and analysis. This work benefits the future research of phish by building a more complete tool for future researchers to utilize while analyzing phish behavior. It provides a way for researchers to recreate the archive, analyze its evolution over time, and edit code to capture any additional relevant data.

Section 2 Related Work:

In "Framework for Detection and Measurement of Phishing Attacks"[Garera], Garera explains how Google detects phish. He talks about using a black list of phish, a white list of targets of phish attacks, and crawling these lists to find similarities between the two. He discusses a regression classifier for choosing which sites are phish. I found this research interesting and was not surprised that Google had such an in depth study of phishing attacks. My research is geared toward the same goal, yet my project focuses on the "black list" section of phish attacks. My work allows researchers to keep an updated version of phish archived in order to detect trends for longitudinal studies. It would even be possible for Google to use this archive and modify it to make a white list archive in order to have a more in depth analysis of what is happening when phish recreate original websites.

In "Characteristics and Responsibilities involved in a Phishing attack",[Van der Merwe], Alta Van der Merwe discusses how the existence of phish change life for users of the internet as well as developers. He discusses the need for security measures and re-engineering as a result of these attacks and talks about weaknesses in phishing attacks. While my project is not intending to characterize phish in depth, it does create a place for researchers like Van der Merwe to analyze phish attacks and test their theories of characterization on a large data set.

In "Who Falls for Phish?:a Demographic Analysis of Phishing Susceptibility and Effectiveness of Interventions"[Sheng], the researchers take surveys to see how humans react to web phishing and what types of people are more likely to fall for the tricks. In contrast to this study, I feel that my research is more relevant to the targets of the attack on the corporate side, but it is interesting to see that there is research to be done on the victims of the attack: both corporate and consumer.

"Detection of Phishing Attacks: A Machine Learning Approach,"[Basnet] is an in depth study of what techniques phish use and how they try to avoid countermeasures. The research discusses using IP-based URLs, HTML emails, sub domain usage, Javascript, form tags, URL images, increased number of links, and keywords to create a phishing attack. I found this paper fascinating. It seems like there are many sure ways of detecting a phish, but as attackers realize how their phish can be recognized, they may quickly create new ways of fooling the public eye. My research will help show how phish have changed throughout the years and allow researchers to see how phish will react to the countermeasures against them over time.

In "Click Trajectories: End to End Analysis of the Spam Value Chain," [Levchenko] Levchenko and his researchers analyze relationships between elements of the spam value chain. The paper looks at resource dependencies within different spam campaigns and how those dependencies can lead to weaknesses or payment bottlenecks. This research focuses on the attacks made through spam emails. A main evaluation of this spam included using "click trajectories," in which the researchers would evaluate the follow through results after clicking on a link within a spammer email. Similar to this work, my research focuses on finding weaknesses within phish. By creating a tool for evaluation, techniques and behavior of phish can be analyzed, inspiring new methods for countermeasures. In addition, I drew inspiration from the "click trajectories" in my process of collecting next page data on "click through" of credential submission.

Project A and B Goals and Status of Completion

Project A Goals:

1. Understand how phish are able to trick users into using their version of a trusted website. This goal was completed by studying the research surrounding phishing: papers on who victims are, how phish are detected, and how phish try to recreate a website. This goal was continued throughout all projects.
2. Create a plan for the semester project. The initial plan was completed; however, aspects of the focus of this research changed throughout the projects. The plan involved archiving live phishing sites and their metadata and then evaluating phishing behavior in depth. Initially, I focused more on planning an analysis for the phish techniques; however, my final project focused on creating a more complete version of the archive, deciding it was more important to create a valuable tool than have an in depth analysis of a small amount of data.

Project B Goals:

1. Design an archive in a well-known format. In this project, I decided the format I would use for the archive was a WARC (Web ARChive) format (as suggested by Calvin). The WARC format is similar to the Internet Archive's ARC file format, which is a commonly used format for data retrieved from web crawling. WARC improves upon ARC by making it easier to harvest, access, or exchange information within the archive. With this archive format, it is possible to associate a website with blocks of information (like source code). The archive would include source code, http responses, and eventually click through results from login submissions.
2. Gather data to populate the archive. I gathered Phishtank's database of valid phish using a developer key provided by Phishtank. This provided me with a set of information that I could scrape and input into the archive.

3. Run a web scraper on collected data. My web scraper grabbed each phish from the database and collected screenshots of them. This goal was difficult to complete within the time frame, as I had to open a browser for each of the 18000 phish. I instead decided to do this for 100 phish and then use those results for analysis.
4. Analyze a manual sample of the data. This phase was about discovery and exploration. I manually looked through the 100 screenshots I had gathered and examined them for similarities and differences. I found that most URLs lead to corporate websites like Google and PayPal [72 URLs], while some were banking websites [3 URLs] , websites with credit card input [3 URLs], or sites that asked for information to “resolve issues” [17 URLs]. I noticed my sample was not random enough and decided to continue analysis on this in the next phase of the project. Additionally, I followed the submission of these websites and found that click through results could end in login errors, a valid next page, or a dummy page. More click through analysis would be done in the project C.

Project C

Project C Goals

1. Project C focused on taking the basic archive produced in Project A and B and extending them into an easily readable, runnable, and reconstructable archive and analysis tool
2. Data collection was extended to source code and http responses in this section and click through results were built into a script for automatic login (using specified types of URLs)
3. Analysis in this section was to be more in depth using a random sample of data to allow for more accurate results.
4. Getting this project into the hands of the public, with easily readable code and a Wiki for how to run the system was a clear goal for this project.

Methodology/Approach

- Environment: the scripts to pull the database, scrape each URL within it for details, and populate the archive were running on the Oracle VM Virtual Box with Ubuntu running so that possible infections from these malicious sites would not hinder my computer. I also ran the click through trials on this VM.
- Input phish: As in project B, the input was a database from Phishtank; however the script was modified to allow for hourly data pulls from this source. Phishtank was a valuable source for this project because rather than creating a phish detection algorithm, I could rely on this well trusted database in order to initialize my system with known valid phish. The database download contains around 18,000 entries.
- System Design: As seen in the Figure 1 below, the system pulls the Phishtank database every hour. It puts this data into a database on the local host. Next, a web scraper runs on the database and grabs screenshots, http responses, and source code of each phish. From there, this updated information is put into a new database with these results as well as into the archive. By keeping a database of the WARC archive, it is possible to have a

backup of information as well as a place for the results of click through analysis to go. The initial phish database is in place so that phish can be put into the system and processed as soon as possible. When a phish id is in the initial database and not the scraped database, this means the phish has not been processed yet. As click through results are performed for each entry, the results from the click through results shall also be updating the WARC archive.

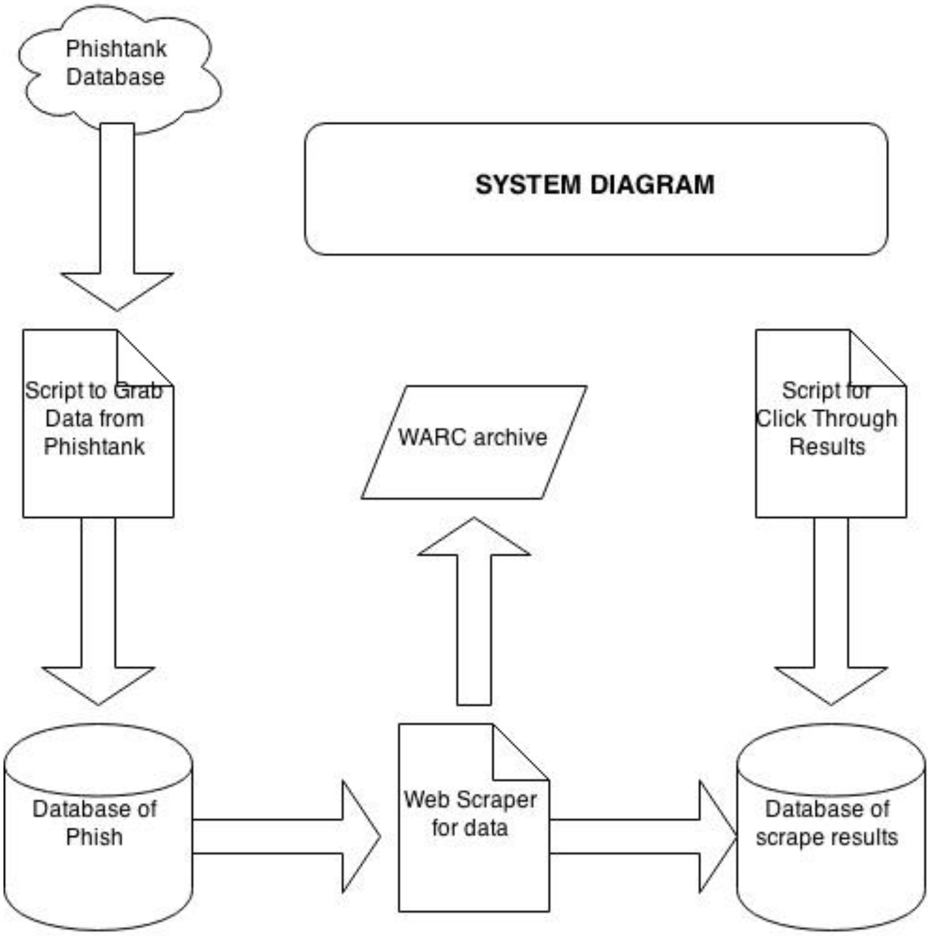
Phishtank Database
SYSTEM DIAGRAM
Script to Grab Data from Phishtank
WARC archive
Script for Click Through Results
Database of Phish
Web Scraper for data
Database of scrape results

Figure 1: Diagram of System

Data Collection/Analysis

Data Collection for this section needed to improve upon project B's data collection. Consequently, a random sample was made of 100 websites to detect targets of phish attacks (what types of web sites were copied) and an automated click through script was written for a selection of PayPal-similar phish. (This click through analysis was the first step in automating this script for the entire database of phish. As of now, the PayPal phish were singled out in order to maintain similarities between the code and see the differences in resulting pages of similar phish).

- Collected: http response, screenshot, source code, click through response, source code, screenshot. Phishtank database information collected (phish id, phish URL, submission time, verification, verification time, whether the phish is online, target of phishing).
- Samples: From the initial database, a sample of 100 random phish were chosen. From the Click through sample, 50 PayPal websites were crawled.

Results and Evaluation

- Archive Tool: The results of this project are an open source repository on GitHub at https://github.com/kmorse/WebPhishing containing all three scripts as well as information on how to run each of these scripts in order to recreate the original archive. As a result, other researchers in the field can run these scripts and maintain information in the archive for analysis purposes. As the scripts are all written in an easy to understand language (Python), modifications of the scripts are simple to develop and researchers can gear the data collection toward their own purposes (capturing data relevant to their own purposes). Eventually, I intend this archive to have enough data relevant to phishing research that modifications will not be the main use of this repository. I am hoping that as time passes, this archive will become a much better tool and will provide a way of visualizing how phish evolve over a period of time (possible years). My evaluation of this system relies on the improvement this tool makes upon existing systems, specifically Phishtank. My tool is geared toward providing key information and scripts to developers, while Phishtank provides a fancy UI for the public to look through. Analysis will be much simpler using my system: grabbing data from the scripts and running algorithms for analysis will be unproblematic. The resulting archive improves upon Phishtank by providing more useful information and allowing that information to be easily processed. Additionally, any developer can pull from my repository and begin analysis or improvement of the tool as soon as possible without requiring a development key.
- Data Collection Evaluation: A simple analysis was made on the data and for future research much more in depth and time consuming analysis is necessary (with larger sample sizes and more automated code) in order to have a better idea of how phish behave and find techniques as well as similarities and differences within the phish. Eventually, a good analysis of this data could lead to new ways of fighting against phishing sites. From the 100 random sample of phish, 19 were banks, 45 were commercial sites such as Google, Yahoo, or Ebay, and 36 were commercial sites with banking related content, which were mostly PayPal sites. These results seem relevant. Most websites I ran into were very commonly known targets. Google and PayPal especially were targeted in the analysis from project B and project C. Google most likely was targeted due to its massive user base, while PayPal was probably targeted because of its extremely sensitive customer information (credit cards). While banks were a target as well, it makes sense that less of the targets are banks, since people are often more careful about giving out banking information or credentials. Click through results (what happens when a user submits invalid credentials into a login form and hits the submit button)

resulted in 28 “valid” pages, showing the phish attempted to maintain a facade of trust even after submission of data. 19 phish returned a login error, asking for resubmission from the user. This indicated that the phish were looking for valid credentials and must have checked the user’s input against the original system’s login. It would be useful to know what happens when entering valid credentials into these sensitive sites (PayPal or WellsFargo) in order to know whether or not a login error page results even with correct information. The last result I found was 404 pages. After submission, 3 phish sent me to a 404 or blank page. The results of this section are useful. It seems the majority of phish try to maintain trust with the user. This makes sense when considering that a user would look into countermeasures once he had determined an attack had been made. This evaluation is too simple; however, to come to any concrete conclusions about the behavior of phish since the sample of click through data is only run on PayPal similar sites and the analysis can only be done using invalid credentials. It is very interesting to see the results of the invalid credential submission, but it will be necessary to enter valid credentials as well.

Conclusion

Overall, this research provides a new tool for archiving and analyzing phishing data open to the public for recreation, manipulation, and analysis. I am excited to see the next direction this research takes and I hope this tool will be a valuable asset to the scientific community.

Future Work

Beyond the scope of this project, there are a few interesting avenues this project may lead to. As is, the archive can be used for analysis, extended for a different scope of work, or improved.

Discussion with Others (Visibility and Feedback)

If I were to continue this project, the first thing I would do is reach out to other researchers around me in order to see whether this archive has sufficient data. I would take this work to a phishing or security related conference or event and get feedback from others in the field working on this type of analysis. By having discussions with other current researchers, I could understand what data is most useful to have within the archive and improve the code to capture these key elements. Furthermore, talking to others in the field would give my project more visibility, giving it a better chance to be a tool that others use. This is especially important because once this code is being maintained by others in the field, I will be confident that this archive will continue and evolve as expected.

Discussion with Phishtank and Other Corporate Businesses

On the same note, I would contact Google, Yahoo, Phishtank, and others working on phishing at a corporate level and notify them about the existence and benefits of my archive. This would give me more visibility in the corporate sector and may influence others to use this code for improving upon their own work in this area (especially Phishtank). In addition, if willing, different businesses may be interested in discussing with me what data they find most important for analysis when creating anti-phishing countermeasures (Google would be especially helpful). Furthermore, if it were possible to set up a dummy bank account with WellsFargo or PayPal, it would be possible to see what happens on click through results with valid credentials. This would help support the analysis already made and extend it.

Avenues for a Valid Phish Database(Where Shall Input Come From)

This project pulls from the Phishtank database once an hour. At this point, the hourly data collection is acceptable for the scope of this project; however, one of the first things I would like to improve within this archive is pulling in data at a much quicker rate. I would like to see a phish become valid, processed, and put in the archive within a matter of seconds, not hours. This hinges on Phishtank at first. They allow developers to pull data once an hour; however, I would like to contact Phishtank find out whether an exception can be made for the purpose of this research. If so, the archive would be able to capture phish right as they are validated and would not run into as many expired entries. If Phishtank does not approve this access, the next step is to look for other work that has classified phish as valid or start a classifier for the archive to pull from. Phishtank is useful as a source for this project, but the code may pull from any archive. Reaching out may be a valuable tool in this archive's life and effectiveness. Perhaps Google would be willing to partner with this research.

Archive Extensions

In order to improve this archive, one important development addition would be to make it possible for every URL to experience a click through analysis. At the moment, the click through script is not automatically run on each URL, but rather manually picked PayPal variant URLs. While this was useful for specific analysis, the point of the click through element of this research is to extend those results to the whole database of phish. Having some click through results was valuable when trying to understand whether or not collecting data on the follow through page of a phish is relevant; however, after coming to the conclusion that the data is valuable, it is time to collect this data on every phish. Furthermore, invalid credentials were used during the click through analysis; however, dummy accounts may be set up in the next phase in order to see whether or not the phish behave the same as when invalid credentials are entered. I would like to see both of these aspects of the project extended as a high priority for the next researchers who pick up this work.

Analysis Extensions

For future analysis, this project should prove to be a useful resource. The source code data alone should provide researchers with valuable information to be used for determining how phish behave. First, by choosing a target company, researchers may grab the original website from the company, and then all phish related to that target in order to look for differences within the code. Does the phish copy source code directly from the original page? Is Javascript used to mimic the page? Is the phish hacked together or well written? Are images taken directly from the original website? These questions bring about different techniques for finding solutions. For example, determining whether the images on the original website were copied by the phish or whether the phish uses screenshots of the original site are interesting computer vision problems. If behavior could be determined in this way, this comparison may be used to detect phish.

Next Phases

This project will be passed along to the research mentor Calvin Ardi for further development or analysis. In addition it is posted on GitHub for public recreation, development, and use. If possible, I will also continue to maintain the archive and make necessary improvements on the web scraping code.

Acknowledgement

A special thank you to the instructor of this Computer Communication course, Ethan Katz-Bassett, for providing detailed feedback on each deliverable of the project as well as lectured on topics in class related to this work and also to Calvin Ardi, the mentor of this project, who was consistently available for meetings and influenced a direction for this project, key advice throughout the duration of the project, and feedback on all vital elements.

Bibliography

Basnet, Ram, Srinivas Mukkamala, and Andrew H. Sung. "Detection of Phishing Attacks: A Machine Learning Approach." *Soft Computing Applications in Industry*. Springer Berlin Heidelberg, 2008. 373-383.

Whittaker, Colin, Brian Ryner, and Marria Nazif. "Large-Scale Automatic Classification of Phishing Pages." *NDSS*. 2010.

Van der Merwe, Alta, Marianne Loock, and Marek Dabrowski. "Characteristics and Responsibilities Involved in a Phishing Attack." *Proceedings of the 4th international symposium on Information and communication technologies*. Trinity College Dublin, 2005.

Levchenko, Kirill, et al. "Click Trajectories: End-to-End Analysis of the Spam Value Chain." *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 2011.

Liao, Qinyu, and Xin Luo. "The phishing hook: issues and reality." *Journal of Internet Banking and Commerce* 9.3 (2004): 1.

Garera, Sujata, et al. "A Framework for Detection and Measurement of Phishing Attacks." *Proceedings of the 2007 ACM workshop on Recurring malcode*. ACM, 2007.

Sheng, Steve, et al. "Who Falls for Phish?: a Demographic Analysis of Phishing Susceptibility and Effectiveness of Interventions." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010.

Blythe, Mark, Helen Petrie, and John A. Clark. "F for fake: Four Studies on How We Fall for Phish." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011.

Purkait, Swapan. "Phishing Counter Measures and their Effectiveness–Literature Review." *Information Management & Computer Security* 20.5 (2012): 382-420.