IEEE CS Security and Privacy Workshops

Towards a Semantics of Phish

Hilarie Orman
Purple Streak, Inc.
Woodland Hills UT, USA
Email: hilarie@purplestreak.com

***Abstract*—Phishing constitutes more than half of all reported security incidents on the Internet. The attacks cause users to erroneously trust websites and enter sensitive data because the email notifications and the website look familiar. Our hypothesis is that familiarity can be defined formally using history data from the user's computer, and effective presentation of the data can help users distinguish phishing messages from trustworthy messages.**

I. Introduction

Phishing has become a huge security problem on the Internet. As with many computer security problems, the attacks require active involvement of a human being. Even knowledgable, security conscious people can fall into a phishing trap [1].

The prevalence of the attacks is astonishing. From [2]: "More than 100,000 security incidents were reported last year to the federal agency that acts as a clearinghouse for cyber security information, most of them dealing with phishing attacks. Some 56,579 phishing incidents were reported to the U.S. Computer Emergency Readiness Team (US-CERT) last year, according to a report released March 23 by the federal Office of Management and Budget. That's 52.7 percent of the 107,439 incidents flagged with CERT by federal and state agencies, commercial enterprises, U.S. citizens and similar agencies in other countries."

A phishing attack usually involves an email message purporting to be from a legitimate business, such as a bank [3] or Amazon. The messages can look official, and it can contain html links to a website that strongly resembles the website of a legitimate business. The message will generally offer some service via an html link. The user must decide if this it is safe to visit the website and possibly enter sensitive information. What criteria should be used?

We argue that the correct answer to is to draw on the same kind of information that makes phishing successful: familiarity. The perpetrators of phishing attacks draw on visual familiarity because it is easy to imitate. Our approach is to draw on the digital artifacts of personal online interactions to help a user understand if a message is based on familiar Internet places or a dangerous combination of familiar and unknown. We compute a "familiarity index" for Internet places; this index is an inferred semantic notion built on the memes of Internet usage.

The hypothesis of this work is that the familiarity index can be a reliable aid for human to use in making a decision to "trust" a website by visiting it or entering personal or sensitive information. If a site is familiar because a person has used it several times in the past, the likelihood of it being trustworthy increases.

Phishing emails often combine familiar and unfamiliar website references in a single message as a way of confusing the recipient. Because the actual content of an html link isn't obvious when rendering the content, a user may click on an unfamiliar link because it is surrounded by familiar content.

Our definition of an Internet place is a "TLD" or "top-level domain" as defined in [4]. The TLD is simply the last two components of a hostname on the Internet. For example, "www.google.com" is a name with the TLD "google.com"; "fbcdn-profile-a.akamaihd.net" has the TLD "akamaihd.net".

We have constructed a message analysis tool that determines the familiarity index for each domain in an email message. The analysis also determines the degree to which familiar and unfamiliar domains are mixed. There are several components to the message analysis, and we evaluated them using the Weka ([5], [6]) system for building clusters.

We trained the analysis evaluation on a set of email messages identified as having a high probability of being phishing attacks. We then tested it on 559 messages that came into an inbox during the course of a few days. Manual inspection was used to determine the effectiveness of the analysis.

Our work also led to the creation of a profile for each domain that combines text and graphics to help the user understand what past usage of the site has been and how it has been used. For messages with an ambiguous familiarity index, the user can easily examine each domain and decide if the risk of visiting an unfamiliar place might be warranted. Because risky sites are often hidden by html, exposing them through the message analysis is often surprising. A message that appears to be from Facebook might hide a url for domain registered to an unrelated company with a shady

DOI 10.1109/SPW.2012.12

contact information.

II. Message Evaluations

The html “trick” that underlies phishing is an exploitation of the difference between information displayed to a browser user and information that the browser uses for navigation. In this minimalist example, the user will see “Company A” but will navigate to domainofevil.com:

```
<A HREF=http://domainofevil.com>Company A</A>
```

We parse each message into mime parts and analyze text and html (the mime parsing was arguably the most difficult software task, and now use it for our regular email client).

The data that we collect for each message is

- A list of all urls in the message, parsed into domains (DNS names)
- The presence of malformed urls in the message, i.e., text that is syntactically incorrect when typed into a browser address bar, or urls that are all numeric (e.g. 144.68.32.121).
- A simple heuristic evaluation the text, ascertaining if it appears to be a request for sensitive information.

Although an email message may appear to the user to contain information about a trusted domain, such as their mortgage holder’s, the that may not be where the browser will be directed to when the user clicks on the link in the message. That information is embedded in the html, and that is what we parse.

For each of the DNS names in the message, we construct an extensive profile, described in the next section. For the purpose of creating classifiers for phishing evaluations, we use four pieces of information:

- Has the user ever visited the domain via a browser?
- Is there a potential login url in the visit history?
- Does the user have any email contacts with the domain?

For the domains mentioned in a message, we compare how often previously visited domains are mentioned vs. how often unvisited domains are mentioned. The result is an heuristic called the “imbalance predicate”.

Finally, each message is characterized by 6 predicates and one derived numeric value:

- Is the ratio of known to unknown urls “imbalanced”?
- Does it appear to request sensitive information?
- Is there a domain that has never been visited?
- Does at least one domain in the message have a login url in the browsing history?
- Does it contain a malformed url?
- Does the user have an email contact in any of the TLDs?
- The sum of the previous attributes using 1 point for each that is true (the “phish score”)

The following text is an example of a message that is probably a phishing attempt. A browser-based email system would display it as in Figure 1, but the message analysis reveals that the message is something other what it appears to be:

```
Message ID:
721af9da76b2ec436525421d1a627baf@notifierfacebook.com
  Phish score: 3
  Contains a trusted domain
  Contains an unvisited domain
* facebook.com, used 4 times, visited 2536 times,
   4887 urls in the domain, 0 emails
evil1.example.com, used 4 times,
          visited 0 times,
          0 urls in the domain,
          0 emails
evil2.example.com, used 1 times,
   visited 0 times,
   0 urls in the domain,
          0 emails
```

The message links the text ”How to get back your lost messages on Facebook” to the domain “evil1.example.com”. Those domains (which are sanitized versions of the actual urls in a real message), are registered to an individual with residential address in a small town. They seem unlikely to have a corporate relationship to facebook.com.

A. Limitations

Various syntactic factors can lead to obfuscation of the effect of clicking on elements of a displayed message. The actual url can be changed by the “base” tag, as in this example:

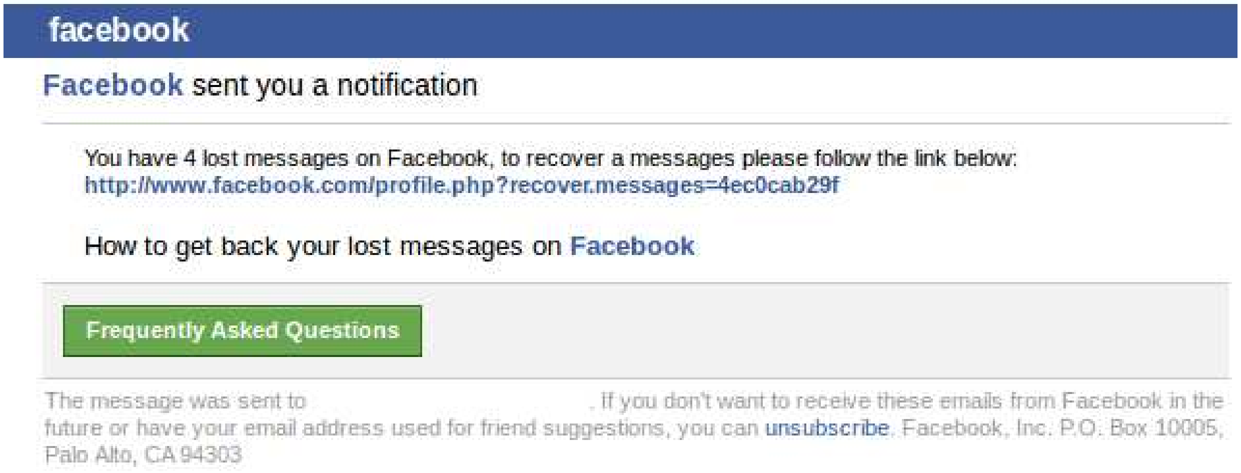
```
<head>
<base href="http://www.example.com/images/"
   target="_blank" />
</head>
<body>
<img src="stickman.gif" />
```

The image loaded via the “src” tag has an abbreviated url that does not include the domain. Although our code does handle this case, because it parses the “base” tag, it cannot find similar uses done through javascript. The following javascript code will, on some browsers, reset the url base in a way that we cannot detect:

```
<script type="text/javascript">
function setbasehref(basehref) {
var thebase = document.getElementsByTagName("base");
thebase[0].href = basehref;
} </script>
```

Although we see occasional uses of the html base tag, the javascript manipulation is unusual. Nonetheless, it demonstrates the difficulty of ensuring that the url analysis is complete.

The visit history is kept by the browser, but the user might have other software that visits urls. A remote email client, a secondary browser, or software that utilizes the “wget” application — any of these could



Facebook sent you a notification

You have 4 lost messages on Facebook, to recover a messages please follow the link below:
http://www.facebook.com/profile.php?recover.messages=4ec0cab29f

How to get back your lost messages on Facebook

The message was sent to . If you don't want to receive these emails from Facebook in the future or have your email address used for friend suggestions, you can unsubscribe. Facebook, Inc. P.O. Box 10005, Palo Alto, CA 94303

Figure 1. A phishing message may appear innocent

contribute to a visit history. Our work currently uses only the Firefox sqlite database in the user's home directory.

Some domains, such as yahoo.com, host millions of email accounts and public web pages. The domain name is of little use in determining trustworthiness; everyone knows someone with a yahoo.com email address, and most people have visited Yahoo websites. A finer-grained url examination would yield better results for domains of major email providers. A similar situation exists with respect to country codes (e.g. "uk" or "fr").

Users who deliberately erase their browsing and email history cannot be helped by a familiarity analysis.

DNS registration information can be helpful, but only if the user has some general knowledge about the entity that he associates with a domain. For example, at the current time, there are only a few widely used social networking (SN) sites. Most people understand what country operates their favorite SN site, and if they realized that the purported url from that site was actually registered on a different continent, they might suspect a phishing attack. But, the Internet promotes businesses without borders, geographic information may become largely irrelevant for multi-national corporations.

III. Domain Familiarity

We attempt to present to the user some idea of the familiarity of a domain by summarizing everything we can find about the user's history of interactions with it.

For each DNS name that occurs in an email message, we construct a domain profile based on its Top-Level Domain (TLD). The profile shows a histogram of visits over the past year and other relevant data.

The Firefox browser keeps an sqlite database of information about visited urls. The ones entered by the user into an address bar are particularly important because they indicate that the user had some reason to trust the url.

For our experiments, we have assumed that the browsing database contains only trusted domains. Prior contamination would be a problem. Because these methods are history-based, they are not suitable in cases where there is no history or a largely untrustworthy history, as on a shared public computer.

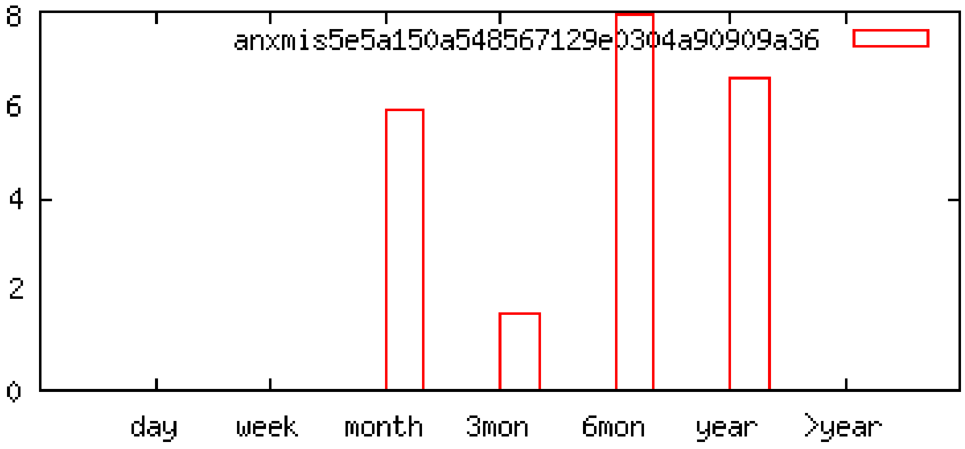
Another indication of trust is a prior visit to a login portal for a website. For this project we used a very simple lexical analysis of urls to make a guess about whether or not it was a login site; more complicated and more accurate methods would use the browser password history, for example. In presenting domain information to the user, we use the character string "log" as an indicator of a login url.

Under Internet governance rules, domain names have degree of accountability. Information about the domain is often available through the "whois" command. We use "whois" results as part of the presentation to the user so that they can see the administrative contact name and address and use that as part of their decision making. Although our system does not set policy based on this information, it could be added. Some decisions are complex, though. Facebook uses a domain registered from the UK for its content distribution urls.

Figure 2 is an example showing part of the display available to users by clicking on a domain name in a message summary:

User visible aspects of domain familiarity:

- Total number of urls visited
- Total number of visits across all urls
- Most commonly visited url
- Histogram of visits during the past year or more
- Possible visit to login url
- Number of email correspondents



```
You have visited 1337 urls in this domain ( example.com ) , a total of 2941
Most visited url is 512 ; 204 visits
```

Figure 2. Example: histogram of visits to a url (visits vs. time) and summary of domain visit info

- “Whois” information

IV. The Experiment and Results

We applied the clustering capabilities of the Weka system to analyze two datasets. Both sets were email messages received by the author. The sets were an aggregation from accounts by a variety of email providers, and one account was not subjected to any spam filtering by the email provider.

The first was a collection of 132 email messages identified by the author as being possible phishing attacks. Not much effort went into creating this collection; in reading through a day’s email, 3 or 4 messages were noted as “might be phish” and recorded as such. The collection period was a few months. The second dataset held 559 messages received by the author over a period of a few weeks; most obvious spam messages had been removed based on keywords in the subject line.

The WEKA algorithm for Expectation Maximisation is the one used for this analysis, with the default parameters of “maxIterations = 100”, “minStdDev = 1.0E-6”, “numClusters = -1”, and “seed = 100”.

The set of messages “likely to be phish” fell into 4 clusters, three of which accounted for 88% of the messages. The three classes had an average “phish score” of 2 or higher. The most important attributes were an unvisited domain and a login domain. There was an obvious correlation between login domains and known email correspondents. The linkage isn’t universal, though, because the most widely used social networking sites require login but do not provide email addresses.

The standard deviation on the cluster characterized by “unvisited domain” was high; by reweighting that attribute, we could catch more phish, at the expense of a higher false positive rate.

The 559 emails that were “not likely to be phishing” fell into 3 clusters, two of which accounted for 86% of the messages. Half of the messages had no phishing indicators 40% had only one indicator. Messages with three or more indicators constituted only 3% of the total. The standard deviation of the “phish score” was small, and a score of less than 2 was an accurate characterizer for this group.

Unexpectedly, there was a cluster in both data sets that had two characterizing attributes: messages with only two urls, one an unvisited domain and the other a trusted domain. Manual inspection of the results shows that a common false positive identification of a message as “phish” is when a friend recommends an internet link. Because these messages tend to be simple and devoid of links to third-party graphical content, they might warrant a new attribute to reduce confusion. However, email source attribution is notoriously unreliable, and even if a link really is from a friend, that person might unwittingly pass along a phish attack. Whether or not familiarity analysis is useful for these cases is an open question.

In summary, these initial results show that the false positive rate for familiarity analysis is about 3% and the rate of correctly identified phish is better than 50%. Further analysis is necessary to understand the latter number because not everything in the phish collection is actually phish, and although most of it is spam, some is not (e.g., some Facebook messages).

Our contention is that having a warning that is correct 75% of the time would be a great help to users because these warnings that will cause them to hesitate before clicking, and to take a moment to examine the graphical presentation of domain usage and provenance. In most cases, we contend, even a naive user would conclude that the risk was too high. Legitimate messages from trusted service providers are unlikely to be caught in a phish trap, but this is something that warrants study with a much larger dataset.

V. Other Methods

Some earlier work also used email analysis. [7] applied machine learning to email features such as numeric domains, number of domains, the presence of javascript or html, etc. They reported a 90% success rate in 2007. One problem with this kind of analysis is that email structure changes with the times, and html has become so common as to be an unreliable indicator today.

Most browsers today come with phish filters that can tap into global information about the trustworthiness of domains, and this provides some protection for users who read email with a web browser. While these methods have value and may produce similar results to ours, they are less effective in dealing with zero-day attacks against new Internet services or rapidly evolving attacks. With our methods, a user can establish familiarity with a legitimate new service without being confused by phishing attacks against that service before it has a globally established reputation.

A new method reported in 2011 [8] uses observations about the visual appearance of a website to judge its "phish" potential. In contrast to this, our methods do not require any accesses to the suspicious websites.

VI. Privacy

Were these methods to be aggregated over the behavior of many users, or if the computations were done by a third party, using information from a user's browsing history, there would be grave privacy concerns. The fact that a user is familiar with a domain gives clues to his identity, his habits, and his interests. Most users want to control the dissemination of such information, and they should be wary of trusting a third-party with the data. Beyond privacy, though, is the possibility that the user could become the victim of a highly targeted phishing attack ("spear-phishing") that could lay bare his passwords.

The familiarity index used in this work is deliberately based on local information, under the control of an individual computer user. The index does not rely on external data collection nor does it need any data other than normal activity logs.

The queries that build the descriptions of DNS domains might leak some information about the user's email. If a phisher registered a unique domain name for each of his targeted users, such as "phish-johnqdoe-299482.net" and carefully watched his DNS server for queries about that TLD, he could determine an his phishing email had been opened, even if the user never visited the domain.

VII. Conclusions

The familiarity index shows promise as a guide to phishing alerts. Because it is based on interactions of a single user over a period of time, is it automatically customized to that person's habits. A strength of the method is that the user be aware of attempts to divert him to a business with which he has no prior relationship, even if it is a legitimate business. This is especially important for a common class of phish that claim to be from "your email provider".

Our experiments with the familiarity index has been limited, and our results are tentatively positive. The evaluation methods need refinement, and we hope that by delving further into a user's interaction history we can derive more attributes that are semantically meaningful and useful for decision making.

Our plans for future work include obtaining a much larger set of training and testing examples, developing more contextual analysis that could be used for a formal grammar of phishing or other mal-messaging. We would also like to compare the analysis to browser filters.

References

[1] R. Lucky, "Clickphobia," *IEEE Spectrum Magazine*, 2011. [Online]. Available: http://www.boblucky.com/reflect/jan11.html

[2] "Phishing tops CERT incidents in 2010," Government Security News, Tech. Rep., March 2011. [Online]. Available: http://www.gsnmagazine.com/article/22782/phishing_tops_cert_incidents_2010

[3] "Malware targets bank accounts, 'gameover' delivered via phishing e-mails," FBI, Tech. Rep., January 2012. [Online]. Available: http://www.fbi.gov /news/stories/2012/january/malware_010612

[4] D. Eastlake and A. Panitz, "IETF Request for Comments: 2606; BCP: 32," IETF Network Working Group, Tech. Rep., June 1999. [Online]. Available: http://tools.ietf.org/html/rfc2606

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, *The WEKA Data Mining Software: An Update*, ser. SIGKDD Explorations, 2009, vol. 11.

[6] I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham, "Weka: Practical machine learning tools and techniques with Java implementations," pp. 192–196, 1999.

[7] I. Fette, N. Sadeh, and A. Tomasic, “Learning to detect phishing emails,” in *Proceedings of the 16th international conference on World Wide Web*, ser. WWW ’07. New York, NY, USA: ACM, 2007, pp. 649–656. [Online]. Available: http://doi.acm.org/10.1145/1242572.1242660

[8] S. Afroz and R. Greenstadt, “Phishzoo: Detecting phishing websites by looking at them,” in *2011 Fifth IEEE International Conference on Semantic Computing (ICSC)*, September 2011, pp. 368 – 375.