APPROACHES TO PHISHING IDENTIFICATION USING MATCH AND PROBABILISTIC DIGITAL FINGERPRINTING TECHNIQUES

Jonathan Zdziarski
CipherTrust, Inc.
jonathan.zdziarski@ciphertrust.com

Weilai Yang
CipherTrust, Inc.
weilai.yang@ciphertrust.com

Paul Judge
CipherTrust, Inc.
paul.judge@ciphertrust.com

ABSTRACT

Phishing is a malicious form of Internet fraud with the aim to steal valuable information such as credit cards, social security numbers, and account information. This is accomplished primarily by crafting a faux online presence to masquerade as a legitimate institution and soliciting information from unsuspecting customers. Phishing attacks involving websites are among the most commonplace and effective types of online fraud, having the potential to cost both victims and targeted organizations in privacy, reputation, and monetarily. Due to the malicious nature of phishing attacks, identifying them bears higher demands in detection than filtering spam or other nuisance content. This paper establishes some requirements for phishing identification and explains various approaches to detection by looking for copying of web site layout and structure through source code (and optionally image) fingerprinting. This enables us to perform a number of exact-match comparisons to genuine websites or to other known attacks. Lastly, we also explore techniques to correlate different attacks to a single likely source.

Categories and Subject Descriptors

I.7.m [**Document Text and Processing**]: Miscellaneous

General Terms

Algorithms

Keywords

fingerprinting, phishing, copying, contextual, classification, text, fraud, detection

1.INTRODUCTION

Successful phishing attacks are based on a form of copying, or reengineering, a website's design and layout in order to pass themselves off as a genuine (targeted) website. A malicious website is crafted which looks and feels like the original site, convincing unsuspecting users that they are giving personal information to a trusted organization. Users are frequently drawn to the sites by forged emails designed to look like legitimate correspondence and may even copy the body from real email, but when the user clicks a link to visit the website, they will be directed to the malicious site instead. The more convincing a phishing attack appears - or rather, the more genuine a malicious website looks - the more success the attack will have in extracting personal information. Some phishing attacks go so far as to create faux websites for which there is no legitimate counterpart; e.g. a page prompting users for personal information the organization wouldn't have otherwise asked for.

In this section, we will discuss the challenges and goals of a successful anti-phishing solution. In section 2, we will cover fingerprinting as a technique to detect phishing attacks and cover various approaches to printing itself. We'll provide an illustration of the match process and cover a probabilistic approach towards detection when using smaller-scale fingerprints. Finally, section 3 will cover source correlation and how different attacks can be correlated back to a single likely source.

1.1 Background

Phishing attacks are a presence whose existence is under false pretenses, and with the goal of convincing the user to enter personal information such as credit card numbers, social security numbers, or any other information that is valuable to the attacker. The primary problem in detecting phishing attacks is that their delivery mechanism bears such a close resemblance to legitimate correspondence that many content-based detection techniques provide only fuzzy results at best. On top of this, it is the actual attack, and not the email, that poses the real threat and the added demand of correlating the attack to the targeted site makes mere content-based detection incomplete.

Fingerprint detection is a technique used to identify copying between documents, in this case faux websites. Applying fingerprinting technique to this problem allows for the detection of copying between a legitimate institution's website or of another, similar attack. This allows us to train previous attacks, legitimate web pages, or a mixture of both. By identifying this copying, we identify the attacks. The benefits of fingerprinting are two-fold. First, in precise identification of an attack, but secondly by forcing scammers to further obfuscate their layout and source code to avoid printing against an original site or a previous attack. In doing so, this increases the risk that the site will appear disgenuine.

1.2 Detection Goals

The goals for identifying phishing can be slightly more demanding than those of identifying spam. Identifying a message as undesirable is relatively easy to accomplish using many methods, but because phishing attacks require a different takedown strategy, it's also necessary to identify messages that are specifically phishing. The burden of enforcement also rests primarily on the targeted organization rather than the service provider, and so institutions need to know when they are targeted so they can involve law enforcement. Once law enforcement has made arrests, proving that copying actually took place (as opposed to a confusion by unsavvy users) can prove useful in prosecution. Finally, managing the problem requires knowledge of which sources are responsible for the largest number of attacks in order to focus efforts and avoid wasting time on minor offenders, or to increase the number of charges against an individual that has already been caught. To summarize, an effective anti-phishing solution should meet the following requirements:

1. Distinguish nuisance spam/spoofing from phishing
2. Identify the organization being targeted in the phishing attack
3. Provide evidentiary examples of copying
4. Correlate different attacks to individual sources

1.3 Existing Detection Approaches

Many approaches exist today in the field of email filtering, some of which are used with varying levels of efficiency in identifying phishing. Most operate on some characteristics of the message content or envelope. Some attempts have been made to apply these techniques to website content; however the existing approaches fail to satisfy many of the goals specific to phishing identification, as illustrated in Fig. 1.1.

1.3.1 Content-Based Filtering

Content-based filtering, such as Bayesian content analysis, has shown to be extremely effective at identifying unwanted content, and can even identify many phishing attacks as spam. Content-based filtering operates on the message content to weigh key words or phrases of interest and assign an overall probability for a particular category. These categories, however, must be predefined and therefore it is infeasible to use content-based filtering as a mechanism for identifying specific organizations being targeted. Phishing messages also bear a significant resemblance to both spam and legitimate mail, making it much more difficult to classify than the linearly separable problem of spam detection. A content-based filter's efficiency in identifying phishing messages can be significantly lower when the user also receives legitimate mail from the target organization, making it more difficult for the filter to tell the difference between the two, nevertheless the results are usually acceptable.

Other content-based approaches, such as KNN (K-Nearest Neighbor) provide clustering methods capable of dynamically creating groups of like messages, however mapping these clusters back to individual organizations (as well as the existing issues discussed of content identification) can be fuzzy and require significant maintenance. Content-based filtering provides no true evidentiary proof of copying, as it can only deliver the number of hints it found interesting, which may be sparse words or phrases spread throughout the document. This same limitation makes it difficult to correlate different attacks from similar sources, again, as only various sparse pieces of text are used.

1.3.2 Envelope Heuristics

Envelope heuristics involve performing various tests on the headers of a message to determine whether or not they have the characteristics of a forged message. It is relatively easy to identify forged messages whose domain name does not match the reverse DNS of the sending mail server, however with many caveats. Not all businesses run their own mail server, and in fact many businesses are still "virtual", whose employees use whatever mail server their ISP provides (making tools such as sender-policy framework infeasible).

More importantly, however, identifying forgery doesn't distinguish spoofed spam from actual phishing attacks. Because a large percentage of spam also forges the sender address, there is no way to identify, with any level of certainty, which messages are phishing. Envelope heuristics also lack the ability to provide definitive information about the targeted organization. While many phishing attacks spoof the target domain, many also do not. For example, many PayPal phishes use a sender address of support@gaypal.com, or historically, support@paypai.com. Only source address information can be used to correlate attacks, and so different attacks (which are likely to originate from different source addresses) cannot be correlated to each other.

1.3.3 Reputation Systems

Reputation systems provide collaborative intelligence about source nodes on a network and provide invaluable information about sources of network phenomenon, serving as a correlation piece. Some examples of these systems include CipherTrust TrustedSource[6] and Project Lumos[7]. As a support system, however, reputation systems can only base their decisions on what other detection mechanisms tell them, and due to the vast network of delivery mechanisms overlapping in both spam and phishing (such as botnets, stolen shells, and mass distribution markets), the source address cannot, with certainty, determine if the message is likely to be spam or phishing. Reputation systems also fail to meet other goals of phishing detection, such as organization identification. Because different attacks typically originate from different source addresses, correlation between different attacks is very limited.

	Content-Based	**Envelope Heuristics**	**Reputation Systems**
Distinguish spam/spoofing from phishing	Yes	No	No
Identify targeted organizations	Partial	Partial	No
Provide evidentiary examples of copying	No	No	No
Correlate different attacks to a source	No	Partial	Partial

Fig. 1.1 Existing Detection Technique Goals

2. FINGERPRINTING AS A DETECTION TECHNIQUE

Digital fingerprinting is a widely accepted approach for detecting copying among documents. Unlike probabilistic content-based models of detection, fingerprinting techniques provide exact matches to a source document and can also track multiple plagiarisms back to a single document or set of documents.

2.1 Goals of Digital Fingerprinting

Applying digital fingerprinting to the detection of phishing attacks meets the goals we've defined for phishing identification:

1. *Distinguishing nuisance spam from phishing*. Fingerprinting can identify specific copying between documents, and is therefore able to distinguish nuisance spam (which does not attempt to copy a legitimate website) to phishing (which does).
2. *Identifying targeted organizations*. Because fingerprinting is exact, it can be used to match a copied document to an exact source, allowing us to identify the target organization it was copied from.
3. *Provide evidentiary examples of copying.* Because fingerprinting performs exact matches of content, examples of copied content can be provided as credible evidence to refute any claims that the site was not intended to be malicious.
4. *Correlate different attacks to individual sources*. As we'll illustrate, the intersection of fingerprints allows us to match unique attributes copied throughout a particular scammer's series of attacks on an organization.

2.2 How Fingerprinting Works

Digital fingerprinting algorithms typically operate by generating a series of digital context markers (or signatures, hashes, etc.) to identify portions of content in a document as unique. Depending on the specific approach used, these are typically hash values, alphanumeric checksums, or some other form of token to denote unique content. A fingerprinting algorithm appropriate for phishing detection has three important properties: white space insensitivity, noise suppression, and position independence. Most fingerprinting algorithms function in a similar way. An input text is provided and a series of digital markers are returned as the output. For example, given the text:

```
'Tis better to have loved and lost than never to have loved at all
```

May result in one particular fingerprinting algorithm returning the following hash markers:

```
0x07a3ca26 0x1a25a210 0x1b47235f
```

The goal of fingerprinting is to identify copying, even if some of the text has been modified. Should the input text be slightly altered to read:

```
'Tis better to hate love and lose and never to have loved at all
```

Both texts still share many similarities and therefore the hash markers will overlap, identifying the relationship between the two:

`0x07a3ca26` `0x0be4f7b3` **`0x1b47235f`**

The detection techniques we'll discuss use this approach to generate a series of context markers which are then recorded to identify document copying through comparison with presented test documents, namely websites. As long as the fingerprinting approach employed follows the basic input/output paradigm shown above, any technique will fit into this application nicely, leaving the format of the context markers fairly irrelevant.

2.3 Various Approaches to Fingerprinting

Using fingerprinting to identify phishing attacks is algorithm-independent. Many different approaches have been proposed over the years including the techniques below.

Karp-Rabin String Matching
Karp and Rabin's algorithm[4] for fast substring matching is the earliest version of fingerprinting based on k-grams. The solution is to find occurrences of a particular string s of length k within a much longer string. The idea is to compare hashes of all k-grams in the long string with a hash of s. It has some optimized hash calculation to reduce the execution time.

All-to-all Matching
All-to-all matching compares all pairs of k-grams in the collection of documents. A simple way is to select every ith hash of a document, but this is not robust against reordering, insertions and deletions. Manber[2] chose to select all hashes that are *0 mod p*. In this way fingerprints are chosen independent of their position, and if two documents share a hash that is 0 mod p it is selected in both documents. Heintze[3] proposed choosing the n smallest hashes of all k-grams of a document as the fingerprints of that document. By fixing the number of hashes per document, the system would be more scalable as large documents have the same number of fingerprints as small documents. The price for a fixed-size fingerprint set is that only near-copies of entire documents could be detected.

SCAM (Stanford Copy Analysis Mechanism)
The idea behind SCAM[5] was that if the distance between feature vectors representing the two documents is small, these two documents are considered similar. The features are words, and the notion of distance is a variation on standard information-retrieval measures of similarity. The similarity measurement is based solely on word frequencies in documents.

COPS
COPS[1] shares similar concepts as SCAM, however feature vectors are constructed based on sentences. The comparison between SCAM and COPS shows that SCAM provides better document overlap detection, however resulting in more false positives.

Winnowing
Winnowing[4] detects similarities whenever the size of a single common substring exceeds a threshold (W). The algorithm guarantees any matches that are more than W characters in length to be detected. The choice of hash depends only on the contents of the window, and doesn't depend on any external information about the position of the window in the file or its relationship to other windows. Winnowing, however, assumes uniformity of input.

2.4 Detection Technique Applying Match Fingerprinting

This section explains the application of fingerprinting to detection and will cover the first two goals: distinguishing phishing and binding them to a specific target. Here we will first illustrate the detection technique using an exact fingerprint matching approach.

2.4.1 Document Retrieval and Preparation

The detection goal is to match copying between two documents, those namely being the suspect site and the legitimate version of the site, or the suspect site and a known phishing attack. Before any detection can occur, it will be necessary to retrieve and prepare the website sources for fingerprinting. The web pages of the suspect site can be extracted from the email itself and retrieved using an HTTP GET.

While the stumbling block for spam filters is widely agreed to be the tokenizer, fingerprinting's ability to identify copying may hinge upon the document retrieval process. Understanding basic scammer trickery designed to prevent users from obtaining the source code will help in ensuring the full source code of the website is retrieved. Some common tricks can include:

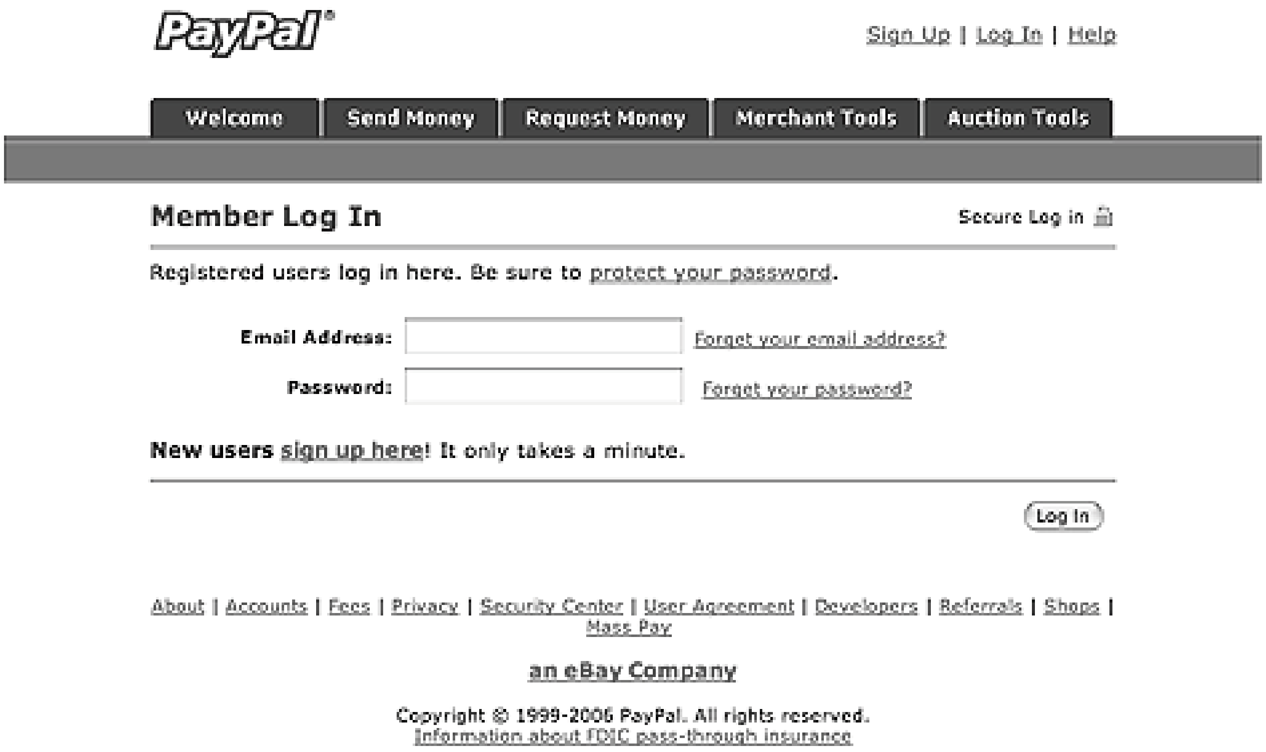
a. Using framesets or IFRAMEs to reference additional pages
b. Incorporating meta refreshes to 'bounce' users to other pages
c. Using callouts such as XMLHTTP to load remote pages
d. Other disgusting and dastardly obfuscations

A successful fingerprinting engine will need to account for these and other tricks to ensure the document is retrieved properly. In keeping with the requirements of fingerprinting, the document must be prepared to eliminate white space and noise. Various scrubbing techniques common to fingerprinting may be employed to accomplish this, the simplest of which involves eliminating any non-alphanumeric characters.

In contrast to content-based filters, the ability to extract the actual content from the web page is irrelevant, and in fact an obfuscated version of the page may serve as a better means of detection if the malicious site itself is used as a training document. Since we also don't know what, if any, encoding or script the source website is using, performing any type of decoding may lead to a higher level of false negatives.

2.4.2 Detection by Exact Matching

Once the classification criteria has been retrieved and scrubbed, it can be fingerprinted. This will generate a series of checksums or hashes, depending on which fingerprint algorithm is used. These hashes will be based on the HTML content and the specific tunables of the fingerprinting approach. In Fig. 2.1, we see an example phish and sample hashes generated by a fingerprinting algorithm.



Member Log In

Registered users log in here. Be sure to protect your password.

Email Address: Forget your email address?

Password: Forget your password?

New users sign up here! It only takes a minute.

About | Accounts | Fees | Privacy | Security Center | User Agreement | Developers | Referrals | Shops | Mass Pay

an eBay Company

Copyright © 1999-2006 PayPal. All rights reserved.
Information about FDIC pass-through insurance

0x0026fa80	0x004129ec	0x00da62a0	0x03f3db0e	0x06fa9461	0x06d4bb73	0x024c3386
0x0069040f	0x00f25530	0x00281f20	0x015bee8b	0x026a0340	0x003379df	0x018ec907
0x039228f5	0x0090b551	0x0121402d	0x03283fb1	0x00ebc2b2	0x03a4dcea	0x001d815b
0x01c779eb	0x0028de1b	0x00e95017	0x00f47a83	0x00baa5cb	0x01293c6b	0x0132f10e
0x001b2b34	0x00dfa027	0x018720f7	0x0335e45a	0x03719720	0x0052cc93	0x015a3b15
0x0028de1b	0x03bb7d39	0x04016646	0x028b9318	0x00334ba1	0x000ad731	0x02906079
0x009f88d3	0x00f48df6	0x0491cdfb	0x01b0d1b5	0x0411a87a		

Fig. 2.1. An example phish and fingerprint hashes

We now fingerprint the original Paypal website and perform a hash comparison. If the number of matches exceeds our threshold, we have a definitive copying event:

0x016ebbee	0x02546530	**0x00da62a0**	0x02079626	**0x03f3db0e**	0x01a4214f	0x003cab67
0x0055d8f4	0x00272399	0x00be53bb	0x009274ec	0x0322e14e	0x025f6aa9	0x0564a046
0x033902f8	0x00b0788c	0x01777ee5	0x01b08076	0x0116e6f3	0x02546b57	0x01b08076

```
0x01523295  0x00ab0a5d  0x015c0529  0x005924ee  0x0009bbcd  0x03b56dab  0x02c04286
0x023b5f86  0x00baa5cb  0x0162ebde  0x03dee023  0x030d9946  0x01a0cd6a  0x017196bd
0x091e583c  0x0139b2e1  0x006f60b4  0x06679235  0x0156d882  0x0023b5e3  0x025896e7
0x003f4a71  0x0637e497  0x01316dfc  0x0100f5d9  0x024536f1  0x000ad731  0x02906079
0x009f88d3  0x00114c2c  0x0051eb2f  0x00f48df6  0x00248175  0x02162f4c  0x006986d9
0x01413ce3  0x000b2636  0x03347a16  0x0113cb62
```

Fig. 2.2. Hashes generated of the original website, using the same fingerprinting algorithm

Depending on the fingerprinting approach used and the specific tunables, a single match may or may not be sufficient to identifying copying. This hinges heavily on the amount of text considered in forming each fingerprint. Variables such as window and hashing size can dramatically affect the amount of text covered by a single hash. In 2.5, we'll discuss a probabilistic model designed to accommodate fingerprint hashes generated on smaller, less-precise lengths of data.

2.4.3 Source Identification and Printing Repository

Training for phishing attacks doesn't necessarily require the source website be used, as the attacks themselves can be used as training data to identify future attacks. This creates an unsupervised training model which makes the challenge of evasion increasingly difficult for each new attack. However, finding and printing the source for all sites to be protected can provide immediate detection without first training any phishing attacks.

In order to perform detection against an original website in an automated environment, it's necessary to identify the source to serve as a means of comparison. A Paypal phish is obviously not going to match to any other website but Paypal, or previously printed attacks against them. The simplest, however least effective approach to identifying the source is to use links to other website content (such as images) embedded within the message or the website source. This will provide a source domain, which can then be printed. But well constructed phishing scams don't always link to their original source, nor do they always copy the front page of the website. A more complete and server-side solution is required to locate and fingerprint source websites.

A source-printing repository is one approach to providing the comparison medium and correlation to a particular organization. Establishing a printing repository involves maintaining a database of source pages and their accompanying fingerprint hashes. When a comparison is performed, positive matches to documents in the repository are then returned. This allows for a more practical and large-scale implementation of fingerprint identification without costing the legitimate sites additional bandwidth overhead whenever they are targeted. Such a repository can be maintained by humans or automated with search bots or a registration process to allow for anyone to place their organization in the repository.

The process of building a repository is very basic:

1. Obtain a list of organizations and the login page URLs on their website
2. Fingerprint each page and store the hash values in a database
3. Subsequent printing of suspect websites can now be compared to the values in the database.

Though fingerprinting works particularly well on today's phishing scams, participating organizations can take additional steps to make fingerprint work in their favor:

a. Use at most one or two sign-on pages that legitimate users can become familiar with
b. Make these sign-on pages complex, requiring a higher degree of copying in order to emulate them
c. Use proprietary and distinguishing text and images to serve as identifying markers
d. Use at least a few common controls (check boxes, submit buttons, etc), which would appear different on various browsers and operating systems if a screenshot was used in the phishing attack.

2.5 Probabilistic Detection

If the fingerprinting approach used can guarantee a solid enough of a match with one or two fingerprint hashes, then exact matching will provide very precise results. Some fingerprinting algorithms, however, are designed (or can be tuned) to operate on small chunks of text that are likely to overlap across two or more different websites. In cases where this type of fingerprinting is uses, a probabilistic approach to matching can be applied to incorporate accurate, but more weighed results. This approach is more indigenous to that of biometric fingerprinting, where many individuals share similar features.

Using the same approach to building a repository as described in section 2.4, probabilistic detection now assigns probabilities to each source hash (that is, hashes belonging to the sites we're storing in the database) based on the spread across multiple organizations. For example, if one particular fingerprint hash has been found in half of the overall organizations, then that fingerprint hash is only 50% reliable and should be combined with other matches in order to determine a more certain result. One formula for determining the probability of a hash is $P=1-N_M/N_T$, where N_M represents the number of organizations matching the hash and N_T represents the total number of organizations. The reasoning behind assigning probabilities based on organizations, and not pages, is due to the likelihood that the same organization may have multiple login pages sharing the same hash. If the hash is indigenous to that one specific organization, it is still a 100% match.

A simple Bayesian algorithm[8] or similar algorithm can then be applied to combine the probabilities to determine a final result. Through experimentation, and depending on the formula used to compute the probability, a threshold between 0.7 – 0.9 may provide a good indicator of a confident result. For example, given three fingerprint hits with probabilities 0.75, 0.98, and 0.46, the following combination might be applied:

$$\frac{(0.75)(0.98)(0.46)}{(0.75)(0.98)(0.46) + (1-0.75)(1-0.98)(1-0.46)} = 0.9920$$

3. SOURCE CORRELATION

We've illustrated the general fingerprinting approaches to identifying phishing. In this section, we'll discuss applying these techniques to identifying attacks sharing unique information copied between each other, showing a high likelihood of originating from the same malicious source.

3.1 Hash-Intersection

In the same fashion that hashes can be matched to legitimate websites to identify copying, an intersection of these hashes can easily be used to identify copying of unique characteristics among different phishing attacks. If different phishing attacks prove to be copied from one another, then there is a likelihood that they have also originated from the same source. To perform this comparison, we compare the fingerprint hashes of two different phishing attacks and then subtract matching hashes from the target website, as shown in Fig. 3.1.

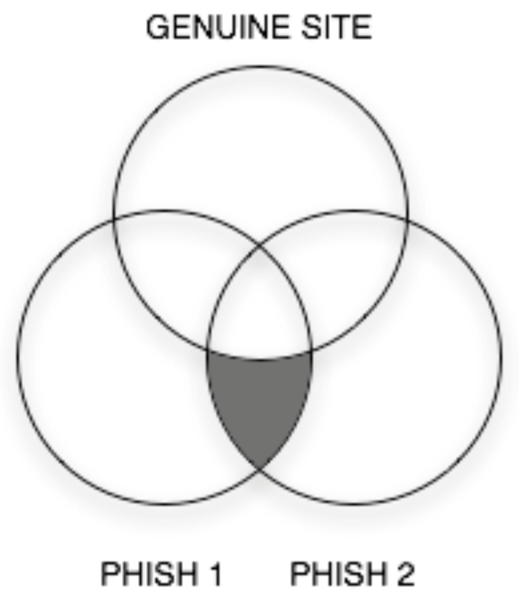


Fig. 3.1. Intersection and subtraction of fingerprint hashes

As we see in Fig. 3.1, the hashes shared between both phishing attacks which are NOT shared by the genuine site (if any) illustrate copying between attacks.

3.2 Fingerprinting Complex Attacks

Just as we can fingerprint legitimate sites, we can also fingerprint complex attacks that fail to match against the legitimate site, or all attacks to increase the difficulty in evading detection. In order to evade a system using such a training paradigm, the attacker would need to craft their attack in such a way that it not only failed to fingerprint against the target website, but against all previous attacks. Should similar documents be used in subsequent attacks, we can then correlate them with the original attack and not only tie them to the same likely source, but also to the legitimate site if we included this in manual training.

This provides the benefit of pushing scammers to not only obfuscate their attack from the original in a way that it will not fingerprint against it, but to also obfuscate each attack from its predecessor to avoid detection. Eventually, the scammer will be forced to alter the layout of the web page so drastically that it will be an obvious fake.

4. SUMMARY

This paper has covered several techniques in identifying phishing attacks by detecting copying between web pages. In detecting copying, not only can matches be made to source documents, but between two existing attacks. Because phishing attacks hinge on formatting attacks to closely resemble the target, there are a finite number of ways in which this data can be represented. By learning new attacks and correlating the copying between them, each new false negative reduces the number of approaches available to future attacks. Eventually, the attacker will be forced to obfuscate their attacks in such a way that the quality of the mockup site is too low to convince users, or to forego obfuscation and surrender to detection. Future work will involve measuring the efficacy of this approach against various evasion techniques, which are likely to develop, and metrics to compare the overall tracking and correlation of progressive attacks.

5. REFERENCES

[1] S Brin, J Davis, H G-Molina. Copy detection mechanisms for ditial documents. In *Proceedings of the ACM SIGMOD Annual Conference*, San Francisco, CA 1995.

[2]: N. Heintze. Scalable document fingerprinting. In *1996 USENIX Workshop on Electronic Commerce*, Nov 1996.

[3]: U Manber. Finding similar files in a large file system. In *Proceedings of the USENIX Winter 1994 Technical Conference*, pp 1-10, San Francisco, CA USA 1994

[4]: S. Schleimer, D. Wilkerson, and A. Aiken. Winnowing: Local algorithms for document fingerprinting, In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2003, pp76-85.

[5]: N Shivakumar and H G-Molina. SCAM: A copy detection mechanism for digital documents. In *Proceedings of the Second Annual Conference on the Theory and Practice of Digital Libraries*, 1995.

[6]: TrustedSource, http://www.trustedsource.org

[7]: Brondmo H., Olson M., Boissonneault P. Project Lumos http://www.espcoalition.org/Project_Lumos_White_Paper.pdf

[8] Graham, P. A Plan for Spam, August 2002 *http://www.paulgraham.com/paulgraham/spam.html*