Phishing 101

Alexandru Catalin COSOI*. Carmen Maria COSOI,**

*Senior Researcher at BitDefender AntiSpam Labs (*email: acosoi@bitdefender.com*)
**Senior Researcher at Nielsen Romania (*email: carmen.cosoi@nielsen.com*)

Abstract: Phishing can no longer be considered a new and emerging phenomenon. Fake websites impersonating both national and international financial institutions can now be created in a matter of seconds and can be hosted anywhere in the world. Large spam attacks, which will populate these websites, can be generated only with a few clicks by renting a network of infected machines. In the context of the current financial crisis, the uncertainty regarding job security and the potential rise in unemployment, might determine a considerably increase in the illegal online activities, as a resort to obtain secondary sources of income. Having this in mind, this paper will try to deal with this problem and provide a possible solution for protection at the browser level by combining both content based and contend independent technologies and also portray an overall picture of this new and growing phenomenon.

1. INTRODUCTION

Phishing is the process of enticing people to visit fraudulent websites and persuading them to enter identity information such as usernames and passwords. The information is then used to impersonate victims in order to empty their bank accounts, run fraudulent auctions, launder money, and so on.

Pharming is a hacker's attack aiming to redirect a website's traffic to another, bogus website. The most common ways for pharming attacks is either by changing the host's files on a victim's computer or by exploitation of vulnerability in DNS server software.

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications, which allows code injection by malicious web users. Examples of such code include HTML code and client-side scripts. Vulnerabilities of this kind have been exploited to craft powerful phishing attacks and browser exploits.

Wireless Phishing is an attack method in which a hacker poses as an access point using a wireless capable device such as a laptop. By posing as a fake access point, the hacker becomes a portal where unwitting Internet surfers may divulge logon credentials to financial websites.

All the techniques presented above and many other are part of the arsenal phishers throw at us every day, and even though APWG's statistics don't indicate a huge difference in the number of unique phishing attacks, we believe that this number will rise consistently in the following months.

We base our assumption on the fact that unemployed people will have to find other sources of income and also on the fact that phishing technologies have evolved considerably from the early stages of their launch. By renting an army of infected machines, one can generate an attack in only a matter of minutes.

As an example, there were only 27 phishing attacks in Romania during the past year. In 2009, by the end of February alone, there are already 98 unique attacks targeting only one financial institution and counting, using 3 of the most recent phishing techniques: attached HTML page (in order to confuse blacklists), Java Encoded WebPage (most toolbars aren't able to parse encoded HTMLs) and also a technique called "the never ending loading page".

Nowadays, current AntiSpam technologies have obtained competitive detection rates on phishing emails, but since recently phishers are advertising their fake websites via a plurality of communication methods (e.g. email spam, instant messaging, social networks, blog posts and even SMS (Cosoi & Petre, 2008; Hatlestad, 2006)), and having some starting information about their victims from social network profiles, (Jagatic et al., 2005) they can easily social engineer their way to the user's trust, which means that a browser level protection must be assured in order to prevent the user to access the website, even though he was persuaded to access the fake URL.

Current browser based technologies employ whitelists, blacklists, various heuristics to see if a URL is similar to a well-known URL, community ratings and content based heuristics (Cranor et al., 2006) and lately visual similarity (Wenyin et al., 2005, Wu 2006).

Most anti-phishing technologies check the URL against a list of known phishing sites known as a blacklist. There are several large phishing blacklists on the Internet and many anti-phishing technologies check each URL users visit against the list to see if the site they are visiting is a phishing site. One problem is that these databases must be kept up to

date which is not an easy task with the current amount of phishing websites created each day. (Owen, 2008).

Blacklisting worked great so far, but the timeframe needed for a URL to become worldwide blacklisted is in most cases overlapping with the time in which the phishing attack is most successful. Also, current content based solutions, mostly blacklists and body heuristics (Cranor et al., 2006) do not always make use of whitelists, which sometimes might cause the filter to consider eBay's official website as a phishing website (Owen, 2008 and Wu et al., 2006, Cosoi, 2008).

2. THE BITDEFENDER APPROACH

In our approach of dealing with the phishing phenomenon, we first tried to address it by making use of content-based algorithms. This primarily aims at detecting the clone sites, which might bring some sort of prejudice to the users. The current antiphishing method implemented in the BitDefender Lab emerged from the hypothesis that in a given language, the number of possible rephrases of a given text that transmits the same or similar information (e.g. We would like to inform you that we are currently carrying out scheduled maintenance of banking software, that operates customer database for BankName OnLine users. Customer database is based on a client-server protocol, so, in order to finish the update procedure, we need customer direct participation. Every BankName OnLine customer has to complete a BankName Customer Form. In order to access the form, please use the link below") and not considering obfuscation purposes, is limited by the speaker's common sense (e.g. the information will be phrased in a simple readable and understandable form). In other words, we assume that all English log-in pages of financial institutions will have a large set of commons words, since they share common purposes and specialized financial vocabulary (Landauer et al., 1998; Kelleher, 2004; Shin & Choi, 2004; McConnell-Ginet, 1973; Merlo et al., 2003; Biemann & Quasthoff, 2007). It is obvious that this behavior is language independent.

Having this in mind, we also postulate that if we consider two web-pages A and B, the number of common words will be less or equal than the number of all words (common and not common) contained by the two documents. This can be translated in mathematics as $|A \cap B| \leq |A \cup B|$. We consider this to be the key element in our method, because it indicates that the necessary amount of memory usage needed to keep structures of the form *(word, document, number of occurrences)* is significantly smaller on similar documents than on different documents.

Following, we defined a similarity indicator between two documents, known as the Jaccard Distance[1] for sets.

$$d = 1 - \frac{|A \cap B|}{|A \cup B|}$$

On identical documents, this distance will have a null value, while in case of almost similar documents, it will be close to 0. Since these are not standard sets (e.g. in ordinary sets, identical elements appear just once, while in this set, we decided that each element (word) appears as many times as it is found in the document), the distance actually provides an acceptable similarity value, judging by the number of words.

Based on this initial background, our proposed method can be better understood from Figure 1. First, the presented webpage is verified against a blacklist (local and RBL) and a whitelist. Afterwards, some simple heuristics are tested against the webpage's content, to check whether this page would actually try to mimic an official log-in page (e.g. contains a submit button, input forms or words like eBay, PayPal, etc). We introduced this step for speed optimization purposes and diminish the duration of the analysis.

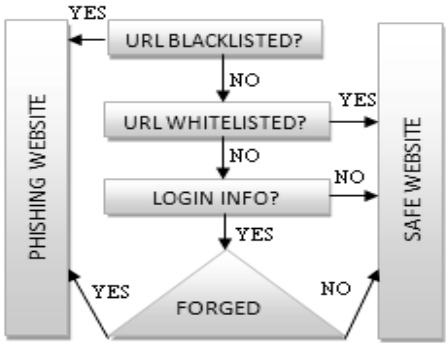

Figure 1 – Toolbar Algorithm

If we consider that it is necessary to run the presented forgery filter on the target webpage, we then start computing the Jaccard distance for each institution on which the filter has trained on (e.g. the words[2] from learned webpages are to be found in the database). Our research revealed that the lowest distance obtained, indicates the highest similarity (judging by the specified distance) between the target webpage and one reference webpage from our database. If the computed distance is smaller then a predefined threshold, we will consider this website a forged page.

When dealing with this technology, using an up to date whitelist is a necessity, because after this filter has learned the original website, it will score a perfect match when visiting the target webpage. An up to date whitelist will inhibit running the forgery filter on original websites in order to avoid false positives (Cosoi, 2008).

Right now, this technology offers protection to about 600 websites (financial institutions, online gaming, webmail accounts) from US, Romania, Germany, Italy, Spain and France. Each time this filter will detect a website as a phishing website (even though it may be online or stored locally in the user's personal computer) it will report the URL to us.

[1] | | means the number of elements of set A

[2] Only visible words will be inserted in the database

We have a 99.8% detection rate on phishing websites that mimic original web-banking pages, and, of course, if they were seen by our filter (Cosoi, 2008). It is obvious that if a phishing website is totally different from the bank's website, our method is useless, and this is the main reason we also use a real time blacklist.

3. CORRELATED PHENOMENON

In addition to our current approach which aims to solve one aspect of the complex phishing attack, our goal would be to also address additional aspects which appear when dealing with phishing, meaning: the lack of trust of some users in the recommendations given by security providers.

In "*Do Security Toolbars Actually Prevent Phishing Attacks?",*the authors note that many users rely mostly on the web content to decide if a site is authentic or phishing. The web content has a large display area and is in the center of the user's attention. It can make itself very convincing since this is the main area where both phishers and users focus firsthand. Most of the time, the web appearance does reflect the site's identity because of the low phishing rate in the real world. What's more, in the early days of phishing, phishing attacks frequently had poor grammar and spelling mistakes. In our study, simulated phishing sites had high-fidelity content. As a result, even though the security toolbar and other security indicators in the browser tried to alert the user, many users disregarded the security toolbars because the content looked considerably similar to the authentic one.

Trustbar makes secure web connections (SSL) more visible by displaying the logos of the website and its certificate authority (CA). This is useful against phishing because many legitimate websites use SSL to encrypt the user's sensitive data transmission, while the majority of phishing sites do not.

eBay's Account Guard shows a green icon to indicate that the current site belongs to eBay or PayPal, a red icon to indicate a known phishing site found on a blacklist maintained by eBay, and a gray icon for all other sites.

SpoofGuard calculates a *spoof score* for the current web page using a set of heuristics derived from previous phishing attacks. It then translates this score into a traffic light: red for spoof scores above a threshold, indicating the page is probably hostile; yellow for scores in the middle; and green for low scores, indicating that the page is probably safe.

Netcraft Toolbar displays information about the site, including the domain's registration date, hosting country, and popularity among other toolbar users.

SpoofStick displays the website's real domain name, in order to expose phishing sites that obscure their domain name. An attack might use a legitimate-looking domain name as a sub-domain, *e.g.,* www.paypal.com.wws2.us to fool users; SpoofStick would display this domain as wws2.us.

A possible solution which could help dealing with this type of issues would be to show the user the original website itself and ask the user to determine whether that wasn't actually the website he desires to visit, or even better, we suggest that the user should be automatically redirected towards the banks original website. Although difficult since maintaining a database of IP addresses of official institutions for each DNS server is a task comparable with the one of maintaining a real time blacklist, we believe that this approach will decrease the number of ignored phishing warnings.

In "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites" (Zhang, 2007), the authors present a similar idea in which they use a TF-IDF algorithm for determining the actual website the user desires to visit. They examine the content of the web-page and create a fingerprint which is then sent to a search-engine. If the web-pages is in the top results than it is considered to be legitimate, otherwise it is a phishing website.

For instance, our suggestion of automatically redirecting the user to the correct website (both address and IP) could actually solve pharming attacks. Also, we suggest spreading antiphishing toolbars not alone, but bundled with a virus security solution in order to prevent direct local pharming attacks.

We believe that the number of phishing attacks, especially spear phishing (targeted, or in other words custom phishing) will start to grow as a response to the increased number of botnets. If so far spear phishing was not a very profitable method, this was due to the fact that it usually had a small number of victims (towards 0) since the phisher didn't have the property of large scale popularization of the URL's. We think this is going to tip since the rise of social search engines like http://www.whostalkin.com/[3] and the computational power of the current botnet armies could easily generate automated spear phishing attacks, leaving the user totally unprotected.

4. CONCLUSIONS

Since phishing websites are no longer advertised on just email spam, we believe that it is time for companies to invest more in research and development on browser level antiphishing protection. This research should focus both on new detection technologies, but also on ways of making the user to not ignore the warnings received.

The current BitDefender approach to web-site clones showed good results on both lab testing and also market testing. This shows that this is a viable method to provide forgery detection to official financial institutions websites. Also, it is not necessary to run this system on all the pages visited by the user, focusing just on the ones that require sensitive information submission, and thereby, highly increasing the

[3] WhosTalkin.com is a social media search tool that allows users to search for conversations surrounding the topics that they care about most. Whostalkin.com can help join in on the conversations that one cares about most. WhosTalking.com search and sorting algorithms combine data taken from over 60 of the internet's most popular social media gateways.

user's tolerance level by decreasing the time spent for analysis.

5. REFERENCES

L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, X. Deng (2005). *Detection of phishing webpages based on visual Similarity*, WWW2005, May 10-14, Chiba, Japan, ACM 1-59593-051-5/05/0005

L. Cranor, S. Egelman, J. Hong, Y. Zhang (2006). *Phinding Phish: An Evaluation of Anti-Phishing Toolbars*, November 13, 2006, CMU-CyLab-06-018

T. Jagatic, N. Johnson, M. Jakobsson, F. Menczer (2005). *Social Phishing*, School of Informatics, Indiana University, Bloomington, December 12, 2005

M. Wu, R. C. Miller, S. L. Garfinkel (2006). *Do Security Toolbars Actually Prevent Phishing Attacks?*, MIT Computer Science and Artificial Intelligence Lab, CHI 2006, April 22-27, 2006, Montréal, Québec, Canada

A.C. Cosoi, G. Petre (2008). *Spam 2.0. Workshop on Digital Social Networks*, SpamConference 2008, Boston, MIT

C. Biemann, U. Quasthoff (2007). *Similarity of documents and document Collections using attributes with low noise*, Institute of Computer Science, NLP department, Universityof Leipzig, Johannisgasse 26, 04103 Leipzig, Germany

P. Merlo, J. Henderson, G. Schneider, E. Wehrli (2003). *Learning Document Similarity Using Natural Language Processing*, Geneva

S. Shin, K. Choi (2004). *Automatic Word Sense Clustering Using Collocation for Sense Adaptation*, KORTERM, KAIST 373-1 Guseong-dong, Yuseong-gu, Daejeon, Republic of Korea

D. Kelleher (2004). *Spam Filtering Using Contextual Network Graphs,* PDF extracted on 10 Jan 2008 from https://www.cs.tcd.ie/courses/csll/dkellehe0304.pdf

S McConnell-Ginet (1973). *Comparative Constructions in English: A Syntactic and Semantic Analysis*, University of Rochester

T. K. Landauer, P. W. Foltz, D. Laham (1998). *An introduction to Latent Semantic Indexing*, Department of philosophy, University of Colorado at Boulder, *Discourse Processes,* 25, 259-284.

L. Hatlestad (2006). *McAfee's Avert Labs is warning of a new threat from hackers: phishing via SMS.* VARBusiness August 31, 2006

M. Ceglowski, A. Coburn, J. Cuadrado (2003). *Semantic Search of Unstructured Data Using Contextual Network Graphs*

V. Prakash, C. Abad, J. de Guerre (2006). *Cloudmark's unique approach to Phishing*. WhitePaper – Antiphishing working group. Extracted from: *http://www.antiphishing.org/sponsors_technical_papers/cloudmark_unique_approach.pdf*

R. Dhamija, J. D. Tygar, M. Hearst (2006). *Why Phishing works.* Proceedings of the SIGCHI conference on Human Factors in computing systems, Montréal, Québec, Canada

R. Dhamija, J. D. Tygar (2005). *The battle against phishing: Dynamic Security Skins.* Proceedings of the 2005 symposium on Usable privacy and security, Pittsburgh, Pennsylvania

G. Tally, R. Thomas, T. V. Vleck (2004). *AntiPhishing: Best Practices for Institutions and Consumers.* McAfee Research, Technical Report – AntiPhishing Working Group WhitePaper.

Ka-Ping Yee (2006). *Designing and Evaluating a Petname Anti-Phishing Tool.* - University of California, Berkeley, Berkeley, CA 94720

K. Hall (2005). *Vulnerability of First-Generation Digital Certificates and Potential for Phishing Attacks and Consumer Fraud*. GeoTrust WhitePaper published on AntiPhishing Working Group website

L. Li, M. Helenius (2007). *Usability evaluation of antiphishing toolbars.* Journal in Computer Virology, Eicar 2007 Best Academic Papers

B. Rudd (2004). *An analysis of Phishing and Possible mitigation strategies.* SANS Institute 2004.

M. Wu, R. C. Miller, G. Little (2006). *Web Wallet: Preventing Phishing Attacks by Revealing User Intentions*, Symposium on Usable Privacy and Security (SOUPS 2006).

M. Wu (2006). *Fighting Phishing at the User Interface.* PhD Thesis, Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering at the Massachusetts Institute Of Technology

C. Jackson, D. R. Simon, D. S. Tan, A. Barth (2007). *An Evaluation of Extended Validation and Picture-in-Picture Phishing Attacks*. Proceedings of Usable Security (USEC'07), February, 2007

A. Cosoi (2008). *Fighting Phishing at the Browser Level.* Virus Bulletin, Spam Supplement, Dec. 2008

Y. Zhang, J. Hong, L. Cranor (2007). *CANTINA: A Content-Based Approach to Detecting Phishing Web Sites,* from www.cs.cmu.edu/~jasonh/ presentations/www2007-cantina.pp

W. Owen (2008). *Examining the effectiveness and techniques of the antiphishing technology in leading web browsers and security toolbars.* SpamConference 2008