A dual approach to detect pharming attacks at the client-side

Sophie Gastellier-Prevost, Gustavo Daniel Gonzalez Granadillo, Maryline Laurent

A dual approach to detect pharming attacks at the client-side

Sophie Gastellier-Prevost, Gustavo Gonzalez Granadillo, and Maryline Laurent

Abstract— **Pharming attacks – a sophisticated version of phishing attacks – aim to steal users' credentials by redirecting them to a fraudulent website using DNS-based techniques. Pharming attacks can be performed at the client-side or into the Internet, using complex and well designed techniques that make the attack often imperceptible to the user. With the deployment of broadband connections for Internet access, personal networks are a privileged target for attackers. In this paper, we propose a dual approach to provide an anti-pharming protection integrated into the client's browser. Our approach combines both an IP address check as well as a webpage content analysis, using the information provided by multiple DNS servers. We present first experimental results and we discuss about future works and limitations of our approach.**

Keywords— **Pharming, phishing, DNS, client-side, webpage, security, attack.**

I. Introduction

PHISHING attacks are a major concern for preserving Internet users privacy. By combining social engineering and website forgery techniques, phishing attacks spoof the identity of a company (typically a bank or an auction site), to trick Internet users to reveal confidential information (e.g. login, password, credit card number). The perfect phishing attack creates a website very similar to the legitimate one by using the same logos, images, structure, etc. However, if the user examines attentively the URL displayed in the address bar of the web browser, he should notice that the URL (especially the domain name) is not the usual one. Other kinds of phishing attacks – i.e. the pharming attacks – are much more complex to detect because both the visited URL and the website are similar to the legitimate site. Pharming attacks aim to corrupt DNS information to redirect users to a fraudulent website under the control of the attacker. DNS vulnerabilities can be exploited at the client-side - by corrupting the user/company computer or the border router -, but also in the ISP network or at the server-side - by intercepting, modifying or spoofing DNS exchanges as well as using content-injection code techniques -.

As DNSSec protocol is not fully deployed today over the whole Internet infrastructure to provide end-to-end secured DNS exchanges, we can hardly protect the user from DNS corruptions, especially for the attacks that occur in his own network. Stamm *et al.* study [1] demonstrates how an internal network can be attacked by modifying the border router configuration.

This paper presents our framework to detect pharming attacks at the client-side. It is organized as follows: Section 2 introduces the different types of pharming attacks. Section 3 details our framework and gives first experimentations results. Then, section 4 discusses future work and details limitations and drawbacks of the proposed framework.

Manuscript received September 28, 2010.

S. Gastellier-Prevost is with the CNRS Samovar UMR 5157, Institut Telecom, Telecom SudParis, Evry, FRANCE (phone: +33-1-60764195; fax: +33-1-60764291; e-mail: sophie.gastellier@it-sudparis.eu).

G. Gonzalez Granadillo is with the CNRS Samovar UMR 5157, Institut Telecom, Telecom SudParis, Evry, FRANCE (e-mail: gustavo.gonzalez_granadillo@telecom-sudparis.eu).

M. Laurent is with the CNRS Samovar UMR 5157, Institut Telecom, Telecom SudParis, Evry, FRANCE (e-mail: maryline.laurent@it-sudparis.eu).

II. Description of Pharming Attacks

Pharming attacks defeat the integrity of the lookup process for a domain name, by exploiting DNS vulnerabilities. Many types of DNS-based attacks have been already identified [2]. In this section, we classified the attacks that aim to corrupt websites for identity theft purposes, according to the location they are implemented (see Table 1). This includes DNS-based pharming attacks as well as some close attacks that are very difficult to detect for the user (i.e. the URL and the webpage content look very similar to the legitimate ones).

A. Client-side

Some pharming attacks are performed at the client-side to modify the local lookup settings. We can distinguish the following attacks:

-- ***Local host attack*** statically modifies the victim's operating system host files to redirect the user's traffic to a domain under the attacker's control.

-- ***Browser proxy configuration attack*** overrides the victims' web browser proxy configuration options, using DNS spoofing or poisoning techniques, to redirect all web traffic to a fraudulent proxy server that is under the control of the attacker. Another type of browser attack - DNS rebinding attack - tends to convert the user's web browser into an open network proxy [3], e.g. the client's browser can visit a

malicious website that embeds a Flash movie which opens a socket to an arbitrary port number rebounded by the attacker. As a result, the attacker is enabled to read arbitrary documents, compromise internal machines, hijack IP address of innocent clients, etc.

-- ***Rogue DHCP.*** The attacker uses malicious softwares to install a rogue DHCP on the client's network to control the DHCP local options. The objective is to modify the DNS server of the user to provide incorrect host resolutions.

-- ***Home or border router attack*** aims to access and compromise the home or border router so that, by adding or modifying DNS entries, the traffic of the user is redirected to the attacker's server. Stamm *et al.* [1] describes several attacks scenarios to compromise home routers.

TABLE I
PHARMING ATTACKS

Location	Attacks
Client	- Local host - Browser proxy configuration - Rogue DHCP - Home or border router
ISP or server	- Domain hijack - Similar domain name - Search engine - Content-injection code - Transparent proxies - Cache poisoning - DNS spoofing - Dynamic pharming

B. ISP (Internet Service Provider) network and Server-side

Other ways to steal the user's credentials, whereas he is surfing on the web, can be performed by corrupting the visited website or URL, or by hijacking the communication through the ISP network using Man-in-the-Middle techniques:

-- ***Domain hijack.*** Through this technique**,** a domain that has just expired is purchased by someone else with malicious purposes e.g. building a new website to imitate the previous version and deceive users that connect to the site.

-- ***Similar domain name.*** The attacker can register multiple spelling permutations of the targeted domain name in order to lure users. For instance, an attacker can register a domain name that adds an extra TLD to the legitimate domain name, e.g. www.mybank.us.com can be used to fake the Mybank's site www.mybank.com.

-- ***Search engine attack.*** The attacker purchases sponsored links or similar services, taking advantage of the flexibility of some search engine providers, in order to place their hyperlinked resources (fake websites) at the top of a user search page response.

-- ***Content-injection code attack*** compromises a web server to insert malicious content into a legitimate webpage. This allows the attacker to gain access to sensitive information maintained by the browser of the user such as stored credentials, cookies, etc. In addition, if the attacker inserts a fake form into a legitimate webpage, the user will connect to the legitimate site and type his credentials in a fake frame under the control of the attacker.

-- ***Transparent proxies*** can be installed in the Internet to force the client's outgoing traffic to be redirected through the attacker's server.

-- ***DNS cache poisoning*** takes advantage of DNS servers caching vulnerabilities in the Internet to add multiple fake resolution entries to hosts, so that a DNS query for a particular domain name resolves into the attacker's IP address. A way to implement this attack is to compromise the authoritative servers to fake resolution responses.

-- ***DNS spoofing attack*** is performed when an unauthorized host successfully inserts incorrect resolution information (IP address) into an Internet DNS server, to redirect users from a legitimate site to one under the control of the attacker.

-- ***Dynamic pharming attack*** compromises Internet DNS servers to attack a legitimate server. Karlof et al [4] explain that a typical dynamic pharming attack delivers a compromised web document to the victim, containing malicious JavaScript code, and then exploits DNS rebinding vulnerabilities into the Internet to force the victim's browser to connect to the legitimate server in a separate window or frame. Once the victim is authenticated, the attacker hijacks the victim's session, enabling him to eavesdrop sensitive content, to fake transactions, to capture passwords, etc.

III. OUR PROPOSAL

The core idea of our framework is to authenticate a website at the client-side. Our approach combines both an IP address check as well as a webpage content analysis, using the information provided by multiple DNS servers.

Our approach intends to be integrated within the web browser of the user, so that a notification is displayed in case of a suspicious website (see Fig. 1).

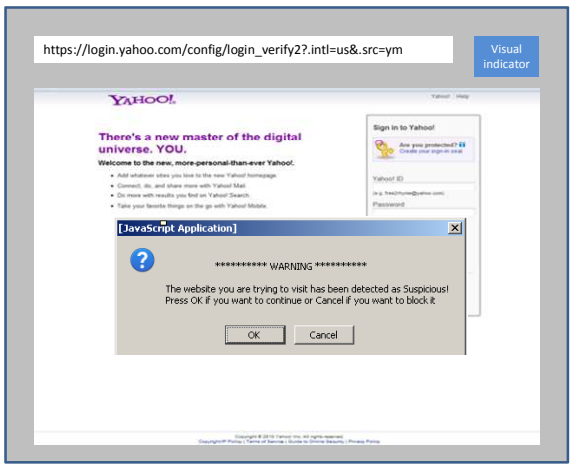

Fig.1. Framework integration into the web browser

Warning alerts: a visual indicator (V.I) is integrated into the web browser to continuously indicate the level of trust. As active warnings are more efficient to protect the user [5] from identity theft attacks, we also integrated a pop-up to alert the user in case of suspicious site.

A. IP address check

Each time the web browser accesses a URL, the domain name of the visited website is checked out. Then, a DNS request is sent to two DNS servers: the default one and a third-party one, in order to compare the IP addresses returned for the evaluated domain name. Even if the DNS response from the default server returns the IP address of the site displayed in the web browser (further named "default IP address"), the same request to a third-party DNS server can return one or several IP addresses (further named "third-party IP addresses"), including or excluding the one used by the browser.

If the default IP address is included in the third-party IP addresses, the site is considered as legitimate. Otherwise, the webpage content analysis is performed (see Fig. 2).

Third-party server definition*:* when installing the anti-pharming solution, the user is asked to choose the third-party DNS server among a pre-defined list of DNS servers (e.g. OpenDNS, Google DNS, etc.). We recommend the user to choose a third-party DNS server different from his ISP.

To validate our proposal, we conducted a first set of experimentations[1] from 6 continents (North America, South America, Europe, Africa, Asia and Australia). The same third-party DNS server was asked to resolve many homepages of legitimate domain names, in order to check the IP addresses changes.

Legitimate domain names selection*:* we selected 226 domain names as follows: 100 sites from the most popular websites[2] in the world, 100 sites from the most popular websites in France and 26 additional bank websites in the world. We selected websites using different languages and different TLD's in the domain name.

For most of the domain names under study, we noticed that the third-party DNS server responses can greatly vary according to the location from which the DNS query was launched.

Table 2 gives few results obtained from the OpenDNS server for the same domain name. We can conclude that for many legitimate websites, the IP address check cannot be used as a single criterion to ensure the homepage legitimacy of a domain name.

[1] Implementation was made using Java code.

[2] The most popular websites list includes e-commerce platforms, social networks, news websites, phone directories, banks, forums, on-line games, etc.taken from the Google top 1000 most visited sites, the Alexa top 500 Global sites, and the Netcraft database.

On the other hand, we conducted the same experimentations with login pages instead of domain name homepages. For many of them, as previously identified in Cao *et al.* study [6], the IP addresses returned by the same third-party server are reported as more stable, regardless of the geographic location. Therefore, the IP address check seems to be more significant when applied to login pages.

TABLE II
OPENDNS RESULTS FROM 4 DIFFERENT LOCATIONS

	France	Tunisia	Mexico	Turkey
www.facebook.com	66.220.146.25	66.220.153.19	66.220.146.11	66.220.153.11
images.google.fr	66.102.9.99	66.102.9.105	209.85.225.106	74.125.39.105
	66.102.9.103	66.102.9.99	209.85.225.105	74.125.39.147
	66.102.9.147	66.102.9.106	209.85.225.103	74.125.39.104
	66.102.9.106	66.102.9.103	209.85.225.147	74.125.39.99
	66.102.9.105	66.102.9.147	209.85.225.99	74.125.39.106
	66.102.9.104	66.102.9.104	209.85.225.104	74.125.39.103
www.amazon.com	72.21.210.250	72.21.210.250	72.21.207.65	207.171.166.252
www.comcast.net	67.215.65.132	213.155.157.49	63.97.94.10	93.158.110.107
		213.155.157.16	63.97.94.58	93.158.110.144

B. Webpage content analysis

In the second part of our approach, we analyze the HTML source code of the visited webpage. Previous works already compared the visited webpage against a reference database [7], [8], [9], but the main drawbacks of these solutions are to maintain an up-to-date database as well as to protect it against any compromising attacks. To avoid this scheme, our framework aims to compare the webpages on-the-fly, without any storage of their content.

As a preliminary step of the webpage content analysis, the IP address check is performed. If the IP address returned by the default DNS server (IPdef) is included in the third-party server reply (IPref), no HTML request is sent. If not, the source code of the two following webpages is downloaded:

- The "visited webpage", given by the IPdef server, is the classical webpage downloaded by the web browser when asking for a URL.
- The "reference webpage" is obtained from a new generated URL based on the third-party server reply, i.e. the domain name part of the visited URL is replaced by IPref. For instance, a visited URL is https://www.amazon.com/gp/yourstore?ie=UTF8&ref_=pd_irl_gw&signIn=1. The default DNS server returns the IP address 72.21.210.250 (IPdef) whereby the visited webpage is downloaded, while the third-party DNS server returns the IP address 207.171.166.252. As such, the reference webpage is downloaded from the following URL: https://207.171.166.252/gp/yourstore?ie=UTF8&ref_=pd_irl_gw&signIn=1.

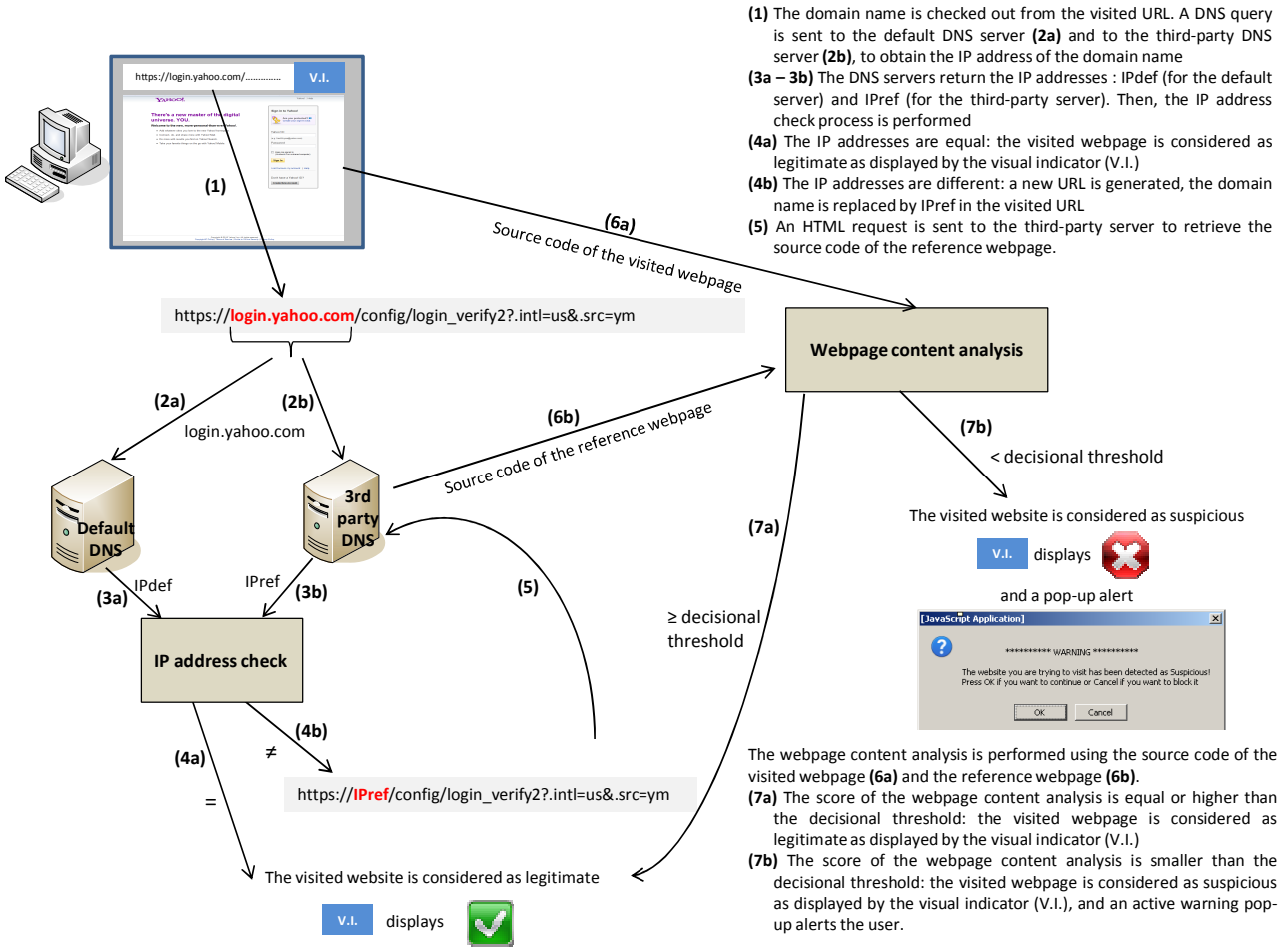

Fig. 2. The evaluation process of the framework

Then, the content of the visited and the reference webpages are compared (see Fig. 2).

Our webpage content analysis refers to a preliminary study we conducted over several legitimate and phishing sites to determine the characteristics and the variability of legitimate websites, depending on the location and/or the downloaded time. This study identified the following difficulties for content comparison:

-- **Webpages content is more and more dynamic**, by integrating ads, RSS feeds, etc.

-- **Phishing and legitimate sites use both absolute and relative paths** for images, links, etc.

-- **Attackers create phishing/pharming site very similar to the legitimate one**, by using mirroring tools and keeping links to the legitimate site as much as possible. They modify minimal part of the legitimate site to lure as many users as possible

-- **Additional script can be added to the HTML content depending on the web browser** of the user (Internet Explorer, Firefox, Opera, …)

-- **HTML structure of the same webpage can be very different** in terms of organization, links, depending on the location where the webpage is downloaded.

We can conclude that analyzing a webpage based on its structure and type of links can give high false positives rates. Therefore, we focused our webpage comparison on the HTML content using character and word approaches:

-- **Character approach** (based on N-gram approach [10]): a score is calculated for each webpage (p) depending on the occurrence frequency (occ) of each character (i), as follows:

$$Score(p) = \sum occ(i) * valAscii(i)$$

where *valAscii(i)* represents the Ascci value of *i*.

Then, we determine the percentage of similarity between two webpages – by comparing their score – and we use it as the "decisional threshold".

We implemented this approach using Java language and we tested it over 10 login websites such as Bank of America,

Paypal, Hotmail, HSBC, etc. We used Phishtank database [13] to find valid phishing sites that looks very similar to the legitimate one. The average rate of similarity between a legitimate webpage and its associated phishing mirror was about 80,5%, fluctuating from 59,1 to 91,9%.

-- Word approach (based on Diff approach [11], [12]): the "decisional threshold" is determined as the percentage of similarity between two webpages. It is obtained by comparing the words it contains and their location in the HTML document. Then, we determine how many words are unchanged, deleted, added or modified.
This approach, implemented using Java language, was tested over 11 login websites such as Facebook, Paypal, HSBC, eBay, Hotmail, etc. We compared the legitimate pages from 4 different locations in the world and we obtained an average similarity rate of 96,8%, fluctuating from 94,9 to 99,1 %. Then, we compared the legitimate pages with some valid phishing sites, extracted from the Phishtank database. The average rate of similarity obtained was about 31,9%, fluctuating from 3,2% to 89,4%.

Future experimentations will tend to confirm these figures and associated decisional thresholds to distinguish legitimate from fraudulent sites.

IV. Conclusion

Our framework proposes an anti-pharming protection for the user in order to detect DNS attacks at the client-side.

Future works will focus on further analysis and experimentation, especially on the webpage content approach. Even if this protection does not help authenticating a website - as it is possible to obtain the same scores for two unrelated webpages -, it can help defining decisional thresholds to differentiate legitimate from phishing sites.

Other limitations of our proposal are related to its location (into the browser) and its implementation. First, it might be subject to web browser vulnerabilities as well as web browser implementation issues, such as the integration of JavaScript language - to design an appropriate interface for the user - and Java language - to make multiple DNS requests - both at the client-side. Second, the network connection of the user might defeat the IP address check of our proposal because of DNS queries filtering.

Finally, we expect the proposed framework to be integrated into a global solution that combines protection against both phishing – such as an anti-phishing toolbar – and pharming.

References

[1] S. Stamm, Z. Ramzan, et Jakobsson Markus, "Drive-By Pharming," *Proceedings of the 9th international conference on Information and communications security*, Zhengzhou, China: ACM, 2007, p. 495-506.

[2] G. Ollman, "The Pharming Guide," Jul. 2005; http://www.ngssoftware.com/papers/ThePharmingGuide.pdf.

[3] C. Jackson, A. Barth, A. Botz, W. Shao, et D. Boneh, "Protecting browsers from DNS rebinding attacks," *ACM*, vol. 3, Issue 1, Jan. 2009.

[4] C. Karlof, U. Shankar, J. Tygar, et D. Wagner, "Dynamic pharming attacks and locked same-origin policies for web browsers," *Proceedings of the 14th ACM conference on Computer and communications security*, Alexandria, Viriginia, USA: ACM, 2007, p. 58-71.

[5] S. Egelman, L. Cranor, et J. Hong, "You've been warned: an empirical study of the effectiveness of web browser phishing warnings," *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, Florence, Italy: ACM, 2008, p. 1065-1074.

[6] Y. Cao, W. Han, et Y. Le, "Anti-phishing Based on Automated Individual White-List," *Proceedings of the 4th ACM workshop on Digital identity management*, Alexandria, Viriginia, USA: ACM, 2008, p. 51-60.

[7] M. Hara, A. Yamada, et Y. Miyake, "Visual similarity-based phishing detection without victim site information," Nashville, Tennessee, USA: IEEE, 2009, p. 30-36.

[8] A.P.E. Rosiello, E. Kirda, C. Kruegel, et F. Ferrandi, "A layout-similarity-based approach for detecting phishing pages," Nice, France: IEEE, 2007, p. 454-463.

[9] E. Medvet, E. Kirda, et C. Kruegel, "Visual-Similarity-Based Phishing Detection," *Proceedings of the 4th international conference on Security and privacy in communication networks*, Istanbul, Turkey: ACM, 2008, p. Article No. 22.

[10] W.B. Cavnar et J.M. Trenkle, "N-Gram-Based Text Categorization," *Proceedings of SDAIR*, 1994.

[11] E.W. Myers, "An O(ND) Difference Algorithm and Its Variations."

[12] S. Wu, U. Manber, G. Myers, et W. Miller, "An O(NP) sequence comparison algorithm," *Information Processing Letters*, Sep. 1990, p. 317-323.

[13] Phishtank, "PhishTank home"; http://www.phishtank.com/.