Bypassing Security Toolbars and Phishing Filters via DNS Poisoning

Saeed Abu-Nimeh and Suku Nair
SMU HACNet Lab
Computer Science and Engineering Dept.
Southern Methodist University
Dallas, TX 75275
{sabunime, nair}@engr.smu.edu

***Abstract*—Security toolbars are used to protect naive users against phishing attacks by displaying warnings on suspicious sites. Recently, web browsers have added built-in phishing filters mimicking the same functionality to detect phishing sites. The present study proposes a new attack to bypass security toolbars and phishing filters via DNS poisoning. Spoofed DNS cache entries are used to forge the results provided to security toolbars and thus misleading information is displayed to the victim. Although there are several studies that demonstrate DNS poisoning attacks, none to our best knowledge, investigate whether such attacks can circumvent security toolbars or phishing filters. Four well-known security toolbars and three reputable browser built-in phishing filters are scrutinized. None of the seven tools detect the attack. Worse still, security toolbars provide the victim with false confirmative indicators that the phishing site is legitimate.**

I. Introduction

The Anti-Phishing Working Group (APWG) [1] defines phishing as a form of online identity theft that employs both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials. Studies show a steady increase in phishing activities as well as the related cost. In December 2007 Gartner Group published results of a survey showing that in 2007 phishing attacks in the U.S. increased compared to the past two years. In 2006, approximately, 3.25 million victims were spoofed by phishing attacks. In 2007, the number increased almost by 1.3 million victims. Moreover, in 2007, monetary losses, related to phishing, were estimated by $3.2 billion.

Anti-phishing security toolbars are one of the widely used phishing detection tools these days by naive users due to their simplicity, interpretability, and configurability. These toolbars are added to web browsers to warn users about suspicious sites they visit. Despite their advantages, these toolbars suffer from several drawbacks, namely exposing victims to attacks carried by phishing sites and providing out of context information about the spoofed links. We discuss these drawbacks in further details in Section II.

Security warnings provided by these toolbars can be divided into two main categories; *positive* and *negative warnings*. *Positive warnings* are displayed when the toolbar detects a phishing site and provides the user with an indicator that the visited site is phishing. *Negative warnings* are displayed when the visited site is not phishing (legitimate) and the toolbar provides the user with confirmative information about the legitimacy of the visited site.

In this study we propose a new attack to bypass anti-phishing security toolbars and phishing filters using DNS poisoning. There exist several discussions on vulnerabilities in home routers and access points (AP) and how they are prone to domain name system (DNS) poisoning and pharming attacks [2], [3]. DNS poisoning involves exploiting a vulnerability in a DNS server and poisoning the table entries of the DNS server with false information. The information can be a false IP address in the table entry, hence when a user tries to resolve a URL, he would be directed to an incorrect IP address. Pharming [1], as a result, can be used to misdirect users to fraudulent sites or proxy servers, typically through this very technique. Although there are several studies that discussed DNS poisoning and pharming attacks, none to our best knowledge, investigated whether such attacks can circumvent security toolbars or phishing filters. Phishing attacks demonstrated in this study are not detected by any of these toolbars or even the latest (including beta releases) web browsers with built-in phishing filters, hence the tools do not provide any positive warnings on the attacks. More importantly, by adding forged entries to the DNS cache, the toolbars provide the user with false negative misleading warnings on phishing sites confirming that the phishing site is legitimate.

The rest of the paper is organized as follows. In Section II we discuss related work. Section III discusses the attack details and attack prevention. We conclude and motivate for future work in Section IV.

II. Related Work

In a research study by Stamm et al. [3], the authors showed that it is possible to gain access to a home router by tricking the user into clicking on a malicious link or by viewing a page that contains a malicious JavaScript code. The attack can be done by using cross site request forgery (CSRF). Upon successful access to the router or the AP, the attacker can change the DNS settings to perform DNS poisoning or pharming.

Wu et al. [4] evaluated the effectiveness of security toolbars in preventing phishing attacks. They performed experiments

on three security toolbars, the browsers address bar, and the status bar. A total of 30 subjects were included in experiments. They showed that all tested security toolbars were ineffective in preventing phishing attacks. Users were spoofed 34% of the time. 20 out of 30 users got spoofed by at least one phishing attack. 85% of the spoofed users thought that websites look legitimate or exactly the same as they visited before. 40% of the spoofed users were tricked because of poorly designed websites, especially when using improper redirections.

Cranor et al. [5] tested the effectiveness of 10 security toolbars and found that three of the 10 toolbars were able to identify over 75% of the phishing sites tested. However, four of the toolbars were not able to identify 50% of the tested sites. This shows that there is a problem in the design of these solutions and more work needs to done to improve the quality of results. In the following section we introduce the tools scrutinized in the study.

A. Security Toolbars and Phishing Filters

Four well-known anti-phishing toolbars and three reputable built-in phishing filters are scrutinized in the study. Note that we test the latest releases of the browsers, namely, IE version 8, Firefox version 3, and Opera version 9.5. Table I summarizes all tested tools. Due to space constrains, we do not provide screen shots of the security toolbars and phishing filters.

TABLE I
SCRUTINIZED SECURITY TOOLBARS AND PHISHING FILTERS.

Toolbar	Supported browser(s)	Warnings
Netcraft	IE and Firefox	Positive and negative
SpoofStick	IE and Firefox	Positive and negative
SpoofGuard	IE	Positive and negative
Google toolbar	IE and Firefox	Positive and negative
IE phishing filter	IE ver. 7 and 8	Positive
Firefox phishing filter	Firefox ver. 2 and 3	Positive
Opera phishing filter	Opera ver. 9.5	Positive

1) Netcraft Toolbar: Netcraft toolbar [6] is a free security toolbar that can be added to IE and Firefox browsers. The toolbar provides both positive and negative warnings. Once the toolbar detects a phishing site, it provides the user with a positive warning that the visited site is spoofed. If the user ignores the message, the toolbar displays statistics about the phishing site including; the month and year the site was established, the rank of the site, a link to provide a report about the site, the country where the site is hosted, and the hosting company. On the other hand, if a legitimate site is detected, the toolbar provides the user with the same previous statistics; however, this time with confirmative information about the legitimacy of the site, i.e. negative statistics. Therefore, if for any reason the toolbar did not detect the phishing site, the user would be able to detect the attack just by looking at the statistics. For instance, if the user found that "Bank of America" site was hosted in "China", it was established in 2007, and the hosting company was "Chinese Hosting Ltd.", this would raise suspicions about the legitimacy of the site.

2) SpoofGuard: SpoofGuard [7] is an open source security toolbar developed at Stanford University. The toolbar displays both positive and negative warnings as well. The tool gives a score to each message at the retrieval step. The score is given based on common characteristics of the previous detected phishing attacks. Examples of characteristics used: misleading patterns in URLs and password input fields in page with no secure connection. Based on the score, the tool provides an indicator (red, yellow, and green) along with the domain name of the site in the toolbar indicating if the page is spoofed or not. If the site is phishing, then the indicator displays a red light and provides a warning message to the user. If the toolbar is suspicious and cannot decide whether the site is phishing or not, it displays a yellow light and asks for the user input. If the visited site is legitimate, then the displayed light is green.

3) SpoofStick: SpoofStick [8] is a free security toolbar that can be added to both IE and Firefox browsers. The toolbar displays both positive and negative warnings as well. SpoofStick only displays the domain name that is hosting the visited site to the user This is useful when spoofed links contain multiple subdomains and the name of the phished site is also crafted in the link to lure victims. For example, `http://patrickbond.co.uk/w/www.chase.com/` displays `chase.com` to trick victims and make the link look legitimate. In the previous example, SpoofStick displays `patrickbond.co.uk` as the domain name for the user, so the user notices the actual hosting domain.

4) Google Toolbar: Google toolbar [9] is a multi-purpose toolbar. One of its features is to display the page rank (out of 10) of the visited site. The toolbar displays both positive and negative warnings. In case of phishing sites, the page will not be ranked. However, legitimate sites have higher ranks and the page rank indicator is green.

5) Internet Explorer: Internet Explorer version 7 [10] was introduced by Microsoft in 2006. IE7 users have the option to enable the phishing filter, as it is not enabled by default. The built-in phishing filter in IE has a downloaded list of "known-safe" sites. Furthermore, it does real-time checking for phishing sites by verifying URLs with an anti-phishing verification server. IE phishing filter only provides positive warnings if a phishing site is detected.

6) Firefox: In Firefox browser version 2 [11] there are two options to detect phishing sites using the built-in phishing filter. Users can either depend on a blacklist which Firefox stores on the user's computer locally, or can choose to check the visited site with Google. If users check with Google to detect phishing sites, Firefox uses the same Google safe browsing interface in Google toolbar to get the page rank and other information. Once a phishing site is detected, the page is blocked and a warning is displayed to the user. Firefox only provides positive warnings if a phishing site is detected.

7) Opera: Opera browser [12] has a built-in phishing filter. If a phishing site is detected, then the browser blocks the site. Similar to IE and Firefox, Opera only provides the user with positive warnings if a phishing site is detected.

III. Attack Details

A. Attack Scenario

Alice is having her morning coffee at "Starbucks". She uses "Starbucks" hotspot to connect to the Internet. *Bob*, next to her, is setting up a rogue AP (See Figure 1) using his laptop with a stronger signal range. *Bob* uses the same setup discussed in Section III-B. He is hosting many phishing banks and a "T-Mobile" *captive portal* to fake the "T-Mobile" login page required at Starbucks, so the attack does not look suspicious. Further, he has a script code to harvest the usernames and passwords entered to any page hosted at the rogue AP and another simple HTTP redirect to redirect victims to legitimate sites after the phish succeeds. By doing this, victims do not notice that their credentials are harvested or stolen. Now, *Alice's* laptop is associated with *Bob's* AP, she logs in to "T-Mobile's" captive portal and continues on to "chase.com" to pay some bills. Being knowledgable of potential phishing attacks, *Alice* makes sure that she types (not by clicking a link that came in email) "chase.com" in the browser address bar. Moreover, *Alice* uses security toolbars and phishing filters to protect herself against phishing. Since the local DNS in the AP is poisoned, *Alice* is directed to the phishing site hosted at the AP's local *Apache* server. A "Chase" phishing page opens to collect *Alice's* credentials. Furthermore, the security toolbars assure her that this site is legitimate and the built-in phishing filters do not provide warnings on the phishing site. Once she provides her credentials, she is redirected to the legitimate "chase.com" site and the security toolbars and phishing filters continue to assure her that she is on the legitimate "Chase" site. *Alice* finishes her coffee and leaves to work. Meanwhile, *Bob* is waiting for his next victim.

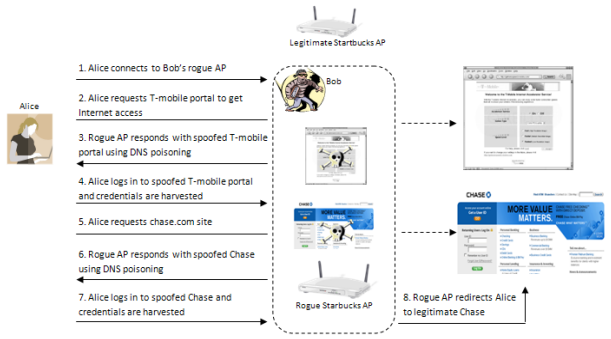

Fig. 1. Rogue AP attack.

B. Attack Setup

We build a rogue AP, also dubbed as *evil twin*, using a *FreeBSD 7.0* server. In order to enable the server to act as an AP, we use *HostAP 0.5.8*. In addition, we install *Apache 2.2* server to host the phishing site locally on the rogue AP. *Dnsmasq 2.40* is installed and used as a local DNS and DHCP server.

After building the rogue AP, we set up a Chase bank phishing site on the *Apache* server. We poison the DNS cache in *Dnsmasq* by adding `address=/chase.com/129.119.1.1` to the `dnsmasq.conf` file, where `129.119.1.1` is the IP address of the *FreeBSD* server. Using Apache virtual hosting, an attacker can host multiple phishing sites similar to the example shown in Figure 2.

```
NameVirtualHost *:80
<VirtualHost _default_:80>
 DocumentRoot /usr/local/www/apache22/data
 Options +Indexes
</VirtualHost> <VirtualHost *:80>
  ServerName chase.com
  ServerAlias www.chase.com
  ServerAdmin tester@unixtest
  DocumentRoot /home/tester/chase
  ErrorLog /home/tester/logs/error_log
  <Directory /home/tester/chase>
     Order Deny,Allow
     Deny from all
     Allow from 192.168.1
     Options +Indexes
  </Directory>
</VirtualHost> <VirtualHost *:80>
  ServerName bankofamerica.com
  ServerAlias www.bankofamerica.com
  ServerAdmin tester@unixtest
  DocumentRoot /home/tester/bofa
  ErrorLog /home/tester/logs/error_log
  <Directory /home/tester/bofa>
    Order Deny,Allow
    Deny from all
    Allow from 192.168.1
    Options +Indexes
  </Directory>
</VirtualHost>
```

Fig. 2. Apache virtual host configuration.

C. Packet Capture Analysis

We investigate the behavior of the security toolbars and phishing filters when a phishing site is detected or a legitimate site is visited. We analyze the traffic between the web browser, with the toolbars and filters enabled, and several legitimate and malicious sites. Note that we are interested in DNS queries to lookup suspicious and legitimate domain names. We use, Wireshark [13], a packet sniffier to analyze TCP requests, traversed servers, DNS queries, and the TCP responses.

Netcraft sends the URL of the visited site to a verification server at `http://toolbar.netcraft.com/check_url/http://sitename.com`. Checking the URL is not performed through a secure connection (HTTPS or SSL) which renders requests and responses prone to forgery via replay attacks. This is illustrated in more details in Section III-E. Once the verification server detects a phishing attack, it provides the toolbar with a response (See Figure 3) that includes; the month and year the site was established, the rank of the site, a link to provide a report about the site, the country where the site is hosted, and the hosting company.

The built-in phishing filter in IE checks the site in question against a downloaded list of "known-safe" sites. Also, it does real-time checking for phishing sites by verifying URLs with an anti-phishing verification server. According to [14], SSL encryption is used to help protect any queries sent from the

```
Since: <a href="http://toolbar.netcraft.com/
site_report?url=http://mizymiau.com">
Jun 2007</a>
Rank: <a href="http://toolbar.netcraft.com/
stats/topsites?s=#-">-</a>
<a href="http://toolbar.netcraft.com/
site_report?url=http://mizymiau.com">
Site Report</a> [US]
<a href="http://toolbar.netcraft.com/
netblock?q=SAGO-20040121-1400,
207.150.160.0,207.150.191.255">
Sago Networks</a>
```

Fig. 3. Netcraft toolbar response.

client to the anti-phishing server. After analyzing the packet capture, we find that, indeed, the anti-phishing filter connects to `65.55.157.59` to verify the domain name and all the traffic in between is encrypted. Interestingly, by having this encrypted channel, the anti-phishing filter in IE seems to be the only solution guarded against replay attacks.

Google toolbar checks the domain name by verifying it at `http://toolbarqueries.google.com`. The server sends back the page rank and other page information. Apparently, the communication with the verification server is not done through a secure connection.

As we mentioned earlier, to verify phishing sites Firefox can be configured to check against a blacklist that is stored on the user's computer locally, or can choose to check the visited site with Google safe browsing API. Since we check the site using the latter approach, the domain name is verified using the same procedure above.

The phishing filter in Opera browser sends the domain name of the visited site to a verification server at `http://sitecheck.opera.com/?host=site.com`. The verification server replies with a `XML` file (See Figure 4). Similar to the majority of the solutions above, the communication with the verification server is not done through a secure connection.

Unlike the other tools, SpoofGuard and SpoofStick do not perform any external domain name or IP address lookup on phishing sites. They merely display the domain name of the hosting site.

```
<?xml version="1.0" encoding="utf-8" ?>
<trustwatch version="1.0">
 <package>
  <action type="searchresponse">
   <trustlevel>V</trustlevel>
   <host>google.com</host>
   <partner>0</partner>
   <serverexpiretime>86400
   </serverexpiretime>
   <clientexpiretime>172800
   </clientexpiretime>
  </action>
 </package>
</trustwatch>
```

Fig. 4. Opera XML response.

D. Attack Description

We setup a rogue AP and host several phishing sites following the details in Section III-B. Thus, multiple clients are associated with the rogue AP. We enable security toolbars and phishing filters in clients' web browsers. Now, the clients visit financial sites, for instance `chase.com` and `bankofamerica.com`. We successfully harvest all credentials entered by associated clients as shown in Figure 5. Most importantly, none of the seven tools detect the attack. Worse yet, security toolbars confirm that the victims are in the legitimate site. Figure 6 shows screen shots of the seven tools bypassed by the attacks.

```
url = www.chase.com, username = victim1,
noerror = 1, password=foobar,
challenge = 1a5294eec0e104c3e734dd6a67d46054

url = www.bankofamerica.com, username = victim2,
noerror = 1, password=f00bar,
challenge = 1a5294eec0e104c3e734dd6a67d46054
```

Fig. 5. Harvested credentials.

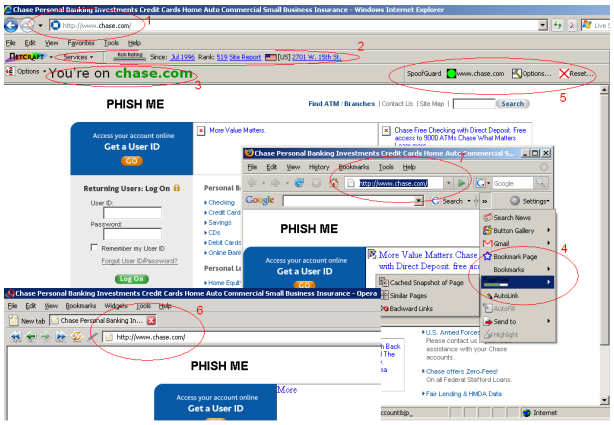

Fig. 6. Bypassing toolbars and filters.

E. Discussion

In the present study we scrutinized well-known security toolbars and browser phishing filters against DNS poisoning attacks. The security toolbars and built-in filters failed to detect the spoofed phishing sites and thus were successfully circumvented. Worse yet, they provided the victim with false confirmative indicators that these phishing sites are legitimate. The attack clearly demonstrates that there is a deficiency in the detection mechanism in the studied solutions. We argue that the following limitations in the toolbars and phishing filters significantly contributed to the attack's success.

1) The toolbars and filters only send the domain name of the phishing site to a verification server or site to check the domain name. Since the attack is done via DNS poisoning, the domain that is sent for verification is the same as the legitimate domain name. Therefore, the verification server or site does not detect the spoof.
2) None of the tools or filters send the IP of the phishing site along with the domain name for verification. They merely send the domain name to be resolved. If the tools and filters send the IP address along with the domain

name to be cross checked against potential legitimate IP addresses, the tools will detect any mismatch between the legitimate and spoofed IP address, thus detect the attack.

3) The communication between *the majority* of toolbars or filters and the verification server is not going through a secure connection, i.e. *SSL or HTTPS*. Now, assuming that our attack did not succeed and that the current solutions account for the two limitations mentioned above, by exploiting the fact that the traffic between the verification server and the client is not going through a secure connection, the attacker can trivially perform a replay attack using the compromised AP by forging the server's responses to lure the toolbar with false responses. Similarly, the attacker can forge the requests sent from the toolbar or filter to the verification server for checking.

F. Attack Prevention

In order to protect the associated clients against the proposed attack, we recommend protection metrics for both the users and the toolbars and filters developers.

1) Users simply can use a virtual private network (VPN) connection to guarantee end-to-end encryption. After connecting to any AP, be it in hotels, airports, or restaurants, users can establish a VPN connection to encrypt the traffic between the user and the VPN server. This not only provides traffic encryption, but also ensures that clients are not using the poisoned local DNS in the rogue AP. In this case, DNS queries will be routed through the VPN and the VPN server will handle them.
2) Similar to VPN, users can use web proxies to route all HTTP and HTTPS traffic through a proxy server. Using this very technique, users avoid looking up DNS queries through the local poisoned DNS in the AP; however, DNS queries will be routed through the web proxy and the proxy server will handle them.
3) Toolbars and filters need also to verify the IP address of the hosting site along with the domain name to be resolved. Should a mismatch occur between the potential legitimate IP addresses and the one provided, the tools and filters can easily detect the attack. Although sometimes sites change their IP addresses, verification servers can maintain a "white list" of potential IP addresses for legitimate sites and update them regularly.
4) Similar to IE, other web browsers need to use a secure connection (e.g. *SSL* or *HTTPS*) for the communication between the verification server and the client to guard against replay attacks. This assures that traffic in between cannot be altered or modified even if the AP is compromised.
5) Few ISPs and network administrators use OpenDNS [15] to block phishing websites. The idea is to block phishing sites at the DNS level, hence users will not need to use phishing filters and security toolbars. Using OpenDNS blacklist, if the domain is known to be a phishing site, it will be null routed or routed to an alter page. This is one possible fix if all clients associated with the AP explicitly choose not to use the DNS provided by AP's DHCP server and use their own DNS server. However, since the AP is compromised, an adversary can fake DNS replies using DNS response forgery and enforce all DNS requests and replies to go through the poisoned DNS.

IV. Conclusions

The present study demonstrated DNS poisoning attack to bypass security toolbars and browser filters used to detect phishing sites. The victim connects to a rogue access point (AP), in which a local web server is hosting phishing sites and a DNS server is forged with poisoned DNS entries. Although the victim types the correct URL in the address bar in the web browser, the victim is directed to the phishing site that is hosted at the local web server in the AP via the poisoned DNS. Interestingly, security toolbars and phishing filters in web browsers cannot detect such attacks. Worse yet, they provide the victim with false misleading confirmative indicators that the phishing site is legitimate. Three reputable web browsers including the latest beta releases of Internet Explorer, Firfox, and Opera with phishing filters enabled and four well-known anti-phishing security toolbars were scrutinized in the study and and none of them detected the attacks.

Since the AP is compromised and the traffic between the toolbars or filters and their corresponding verification servers is not encrypted, this motivates future work to explore various types of replay attacks that can be performed to deceive the verification server and the toolbars as well.

References

[1] Anti-Phishing Working Group, 2007. [Online]. Available: http://www.antiphishing.org/

[2] A. Tsow, M. Jakobsson, L. Yang, and S. Wetzel, "Warkitting: the drive-by subversion of wireless home routers," *The Journal of Digital Forensic Practice*, 2006. [Online]. Available: http://www.indiana.edu/ phishing/papers/warkit.pdf

[3] S. Stamm, Z. Ramzan, and M. Jakobsson, "Drive-by pharming," Symantec Inc., Tech. Rep., 2006.

[4] M. Wu, R. C. Miller, and S. L. Garfinkel, "Do security toolbars actually prevent phishing attacks?" in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006.

[5] L. F. Cranor, S. Egelman, J. Hong, and Y. Zhang, "Phinding phish: An evaluation of anti-phishing toolbars," CMU, Tech. Rep., 2006.

[6] Netcraft. [Online]. Available: http://toolbar.netcraft.com/

[7] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell, "Client-side defense against web-based identity theft," in *11th Annual Network and Distributed System Security Symposium (NDSS '04)*, 2004.

[8] SpoofStick. [Online]. Available: http://www.spoofstick.com

[9] Google Toolbar. [Online]. Available: http://toolbar.google.com/

[10] Internet Explorer. [Online]. Available: http://www.microsoft.com/windows/products/winfamily/ie/default.mspx

[11] Firefox. [Online]. Available: http://getfirefox.com

[12] Opera. [Online]. Available: http://www.opera.org

[13] Wireshark. [Online]. Available: http://www.wireshark.org

[14] Principles behind IE7's Phishing Filter. [Online]. Available: http://blogs.msdn.com/ie/archive/2005/08/31/458663.aspx

[15] OpenDNS. [Online]. Available: http://addiator.blogspot.com/2007/09/opendns-and-anti-phishing.html