INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

A New Approach To Protect Against Phising Attacks With Bogusbites

HarishBabu. Kalidasu
Asst.Professor
Priyadarshini Institute of Technology
& Science, Chintalapudi, Tenali

B.Prasanna Kumar
Assoc. Professor
Mandava Institute of Technology
& Science, Jaggayyapet.

Abstract:

In this paper, instead of preventing human users from "biting the bait", we propose a new approach to protect against phishing attacks with "bogus bites". We develop Bogus Biter, **a unique client-side ant phishing tool, which transparently feeds a relatively large number of bogus credentials into a suspected phishing site. Bogus Biter conceals a** victim's real credential among bogus credentials, and moreover, it enables a legitimate **web site to identify stolen credentials in a timely manner. Leveraging the power of client-side automatic phishing detection techniques, Bogus Biter is complementary to existing preventive anti-phishing approaches. We implement Bogus Biter as an extension to Firefox 2 web browser, and evaluate its efficacy through real experiments on both phishing and legitimate web sites. Many anti-phishing mechanisms currently focus on helping users verify whether a web site is genuine. However, usability studies have demonstrated that prevention-based approaches alone fail to effectively suppress phishing attacks and protect Internet users from revealing their credentials to phishing sites.**

Introduction

A phishing attack is typically carried out using an email or an instant message, in an attempt to lure recipients to a fake web site to disclose personal credentials. To defend against phishing attacks, a number of countermeasures have been proposed and developed. Server-side defenses employ SSL certificates, user selected site-images, and other security indicators to help users verify the legitimacy of web sites. Client-side defenses equip web browsers with automatic phishing detection features or add-ons to warn users away from suspected phishing sites. However, recent usability studies have demonstrated that neither server-side security indicators nor client-side toolbars and warnings are successful in preventing vulnerable users from being deceived [6, 21, 23, 26, 28]. This is mainly because (1) phishers can convincingly imitate the appearance of legitimate web sites, (2) users tend to ignore security indicators or warnings, and (3) users do not necessarily interpret security cues appropriately. Educational defenses [12, 16, 24] and takedown defenses [13, 18, 39] have also been studied. However, these defenses cannot completely foil phishing attacks and will take a long time to be effective on a large scale.

In this paper, we propose a new approach to protect against phishing attacks with "bogus bites" on the basis of the two observations mentioned above. The key feature of this approach is to transparently feed a relatively large number of bogus credentials into a suspected phishing site, rather than attempt to prevent vulnerable users from "biting the bait". These "bogus bites" conceal victims' real credentials among bogus credentials, and enable legitimate web sites to identify stolen credentials in a timely manner. Based on the concept of "bogus bites", we design and develop Bogus Biter, a unique client-side anti-phishing tool that is complementary to existing prevention-based mechanisms. Seamlessly integrated with the phishing detection and warning mechanisms in modern web browsers, Bogus- Biter is transparent to users. While leveraging the power of widely used client-side automatic phishing detection techniques, Bogus Biter is not bound to any specific phishing detection scheme. Thus, Bogus Biter can utilize the latest advances in phishing detection techniques such as blacklists and heuristics to protect against a wide range of phishing attacks. Moreover, Bogus- Biter is incrementally deployable over the Internet, and the fraud protection enabled at a legitimate web site is independent of the deployment scale of Bogus Biter. We implement Bogus Biter as a Firefox web browser extension and evaluate its efficacy through real experiments over both phishing and

legitimate web sites. Our experimental results indicate that Bogus Biter is a promising anti-phishing approach.

These different approaches are all preventive by nature. They endeavor to prevent users from being tricked into revealing their credentials to phishing sites. Nevertheless, these prevention-based approaches alone are insufficient to shield vulnerable users from "biting the bait" and defeat Phishers, as human users are the weakest link in the security chain. The ever-increasing prevalence and severity of phishing attacks clearly indicate that anti-phishing is still a daunting challenge. In response to this challenge, we have made two observations with respect to the acquisition of credentials by phishers and the automatic detection of phishing attacks on web browsers. First, currently the majority of those who have "bitten the bait" and fallen victim to phishing attacks are real victims, thus it is trivial for a phisher to verify the acquired credentials and trade them for money. However, if we can supply phishing sites with a large number of bogus credentials, we might be able to hide victims' real credentials among bogus credentials and make it harder for phishers to succeed. Second, although remarkable advances in client-side automatic phishing detection have empowered web browsers to identify the majority of phishing sites [4, 11, 17, 33, 36, 40], the possible false positives (legitimate web sites misclassified as phishing sites) make it hard for web browsers to directly block users' connections to suspected phishing sites. Thus, issuing warnings and expecting users to leave a suspected phishing site have become the most common actions employed by modern web browsers. However, instead of just wishing vulnerable users could make correct decisions, if we can effectively transform the power of automatic phishing detection into the power of automatic fraud protection, we will take a big step forward towards winning the battle against phishing.

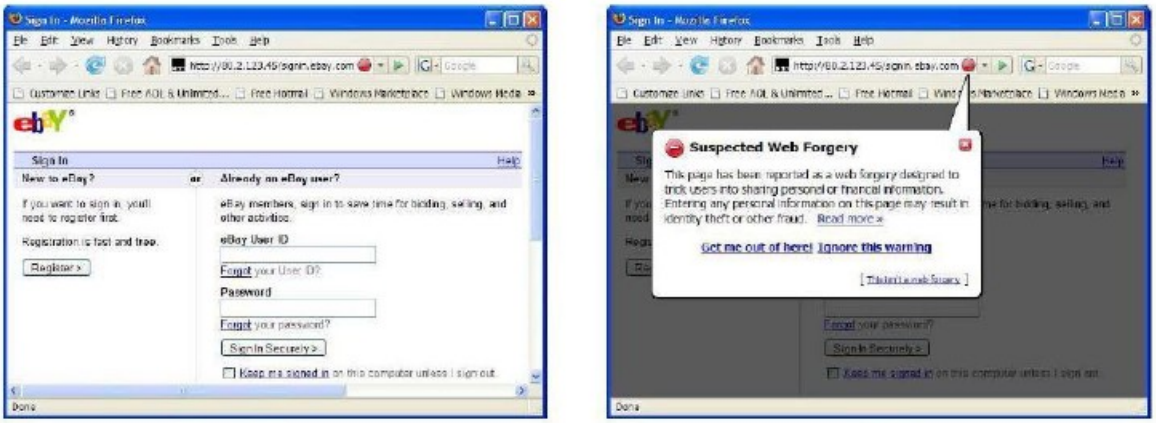
Background

Figure 1(a) illustrates a phishing site designed to attack eBay users. In a typical scenario, a user receives a spoofed email that appears to be sent from the real eBay, luring the user to log into the phishing site. Once the user believes this site is the genuine eBay web site and logs in, the user's username/password credential is stolen. Passwords have increasingly been targeted by harvesting attacks, as they protect online accounts with valuable assets [9]. While some phishing attacks may steal other types of credentials such as credit card numbers and social security numbers, the most common type of phishing attack attempts to steal account numbers and passwords used for online banking

[15]. Therefore, protecting a user's username/password credential is the primary focus of many client-side anti-phishing research work such as Spoof Guard [4], Dynamic Security Skins [5], PwdHash [22], Web Wallet [29], and Passpet [31]. Our work also focuses on protecting a user's username/ password credential. In the remainder of this paper, we use the terms credential and username/password pair interchangeably. While distinct from preventive anti-phishing mechanisms, Bogus Biter complements them in a natural way. In particular, Bogus Biter leverages the power of client-side automatic phishing detection mechanisms and takes advantage of the state-of-practice phishing warning mechanisms in popular web browsers to transparently protect vulnerable users. Among automatic phishing detection mechanisms, two commonly used techniques are blacklists and heuristics. Blacklist-based techniques generate close-to-zero false positives and can detect most phishing attacks [17, 32, 35, 37]. For example, Ludl et al: demonstrated that blacklists provided by Google (used by Firefox 2) can recognize almost 90% of live phishing sites. However, because some phishing sites may not be added into blacklists and the so-called zero-day attacks may occur, researchers have proposed various heuristic-based techniques to identify phishing sites in real time [4, 11, 17, and 33]. These heuristic-based techniques have obtained very encouraging results. For example, CANTINA, a content-based detection tool proposed by Zhang et al: [33] can identify 90% of phishing pages with only 1% false positives. A URL-based classifier proposed by Garera et al: [11] is another tool which can catch 95.8% of phishing pages with only 1.2% false positives. Currently, Firefox 2 primarily employs blacklist-based techniques while Internet Explorer (IE) 7 uses both kinds of techniques [36, 40]. Because Bogus Biter's design is independent of any specific detection scheme, it can leverage advances in both blacklist-based techniques and heuristic based techniques to combat the majority of phishing attacks. Regarding phishing site warning mechanisms, the state of- practice is to make it mandatory for a user to respond to the warning of a suspected phishing site. Figure 1(b) illustrates the warning given by Firefox 2 [36] after correctly identifying the example web site in Figure 1(a) as a phishing

site. A user is unable to enter the username and password without first interacting with the warning page. If the user clicks on the link "Get me out of here!", the user is redirected to a default page and is protected. Otherwise, if the user clicks on the link "Ignore this warning", the warning page disappears and the user is exposed to the risk of

credential theft. A similar warning mechanism is also used in IE 7 [40]. Both Firefox 2



(a) (b)

Figure 1: (a) A phishing site designed to attack eBay users, (b) Firefox 2 phishing warning mechanism.

and IE 7 might choose such a strong warning mechanism because: (1) issuing warnings simply through browser-based security indicators such as the address bar, the status bar, and various toolbars is ineffective [6, 21, 23, 26, 28], and (2) directly blocking users' connections to suspected phishing sites is unacceptable, due to inevitable false positives. Although using strong warning pages represents current best practice, the usability study of the IE 7 warning page conducted by Schechter et al: [23] demonstrates that over 50% of participants still ignore the warning and enter their passwords, despite the overtness of the warning page and its strong wording. Another usability study conducted by Egelman et al: [7] shows that over 20% of participants ignore the strong warnings.

Design

In this section, we first give an overview on the design of Bogus Biter, and then we detail the offensive line and defensive line of Bogus Biter.

Design Overview

Bogus Biter is designed as either a new component or an extension to popular web browsers such as Firefox 2 or IE 7. It integrates seamlessly with phishing detection and warning mechanisms of current web browsers to protect vulnerable users against phishing attacks.

<u>How It Works</u>

When a login page is classified as a phishing page by a browser's built-in detection component, Bogus Biter is triggered. At this point, Bogus Biter will perform differently

based on a user's response to the browser's phishing warning page. For a vulnerable user who clicks the "Ignore this warning" link and submits a real credential, Bogus Biter will intercept the victim's real credential, hide it among a set of **S-1 generated bogus credentials, and then submit the S credentials one by one to the phishing site within a few milliseconds. For a security-conscious**

user who clicks the "Get me out of here!" link on the warning page, Bogus Biter will **generate a set of S bogus credentials, and then feed them one by one into the phishing site in the same way as it does for a vulnerable user. These actions are completely transparent to both vulnerable and security conscious users.**

Design Assumption

We assume that a phisher does not have a complete list of valid usernames for a targeted legitimate web site, and cannot directly query a targeted legitimate web site for the validity of a specific username. Although this assumption may not be strictly correct

for email service web sites and community web sites, it is generally true for financial institutions, which are the main targets of phishing attacks. Financial institutions seldom have valid username lists publicly accessible. Meanwhile, for a failed login attempt, web sites often try to hide whether the failure is due to an incorrect username or due to an incorrect password by returning the same error message [3, 10], making it very hard to test the validity of a given username. Indeed, preventing the leakage of username validity information is necessary for protecting user privacy,

guarding users from invasive advertising and phishing, and defending against password guessing attacks. To enhance such a protection, the recent work by Bortz et al: [3] recommends that the response time of HTTP requests should be carefully controlled by some web sites to remove timing

vulnerabilities. Flor^encio et al: [10] further suggest that increasing username strength **could be more beneficial than merely increasing password strength.**

Design Objectives To Be Effective, Bogus Biter Has Two Key Design Objectives

Offensive objective: Bogus Biter should inject as many bogus credentials as possible into a phishing site, thus well hide victims' real credentials among bogus **credentials.**

Defensive objective: Given that a phisher is aware of Bogus Biter and is willing to assume the heavy burden of sifting out bogus credentials, Bogus Biter should enable a legitimate web site to exploit the filtering process initiated by the phisher for detecting victims' stolen credentials in a timely manner.

Offensive Line

To achieve its offensive objective, Bogus Biter needs to meet the following three requirements.

Massiveness: The number of bogus credentials fed into a phishing site should be large so that the overwhelming majority of credentials received by a phisher are bogus.

Indiscernibility: Without the credential verification at the legitimate web site, it is extremely difficult for a phisher to deterministically discern, either at credential submission time or afterwards, who are real victims and what are real credentials.

Usability: The usage of Bogus Biter at the client-side should not incur undue overhead or unwanted side effects, nor should it produce any security or privacy concerns.

Massiveness

We use the real-to-all ratio—the ratio between the number of real credentials being stolen and the total number of credentials being collected—to estimate how many bogus credentials could be fed into a phishing site to hide victims' real credentials. Without Bogus Biter, most or perhaps all credentials collected by a phisher are real credentials submitted by victims, thus the real-to-all ratio is close to one. A phisher can easily verify these credentials at the legitimate web site, assess their values, and ultimately use them to obtain money. With Bogus Biter equipped at each web browser, the realto- all ratio will be determined by two factors. The first is the set size S, i.e., the number of credentials submitted by Bogus Biter for each phishing site visit. The second is the cheat-to-click ratio, which is the ratio between the number of victims who reveal their credentials and the total number of users who visit the phishing site. The set size S is a parameter that we can configure, while the cheat-to-click ratio is related to the severity of phishing attacks. If all the phishing site visitors become victims, the cheat-to-click ratio equals one. Therefore, the upper bound of the real-to-all ratio is 1 /S . However, the experiments conducted by Jakobsson and Ratkiewicz [14] demonstrate that even with the effects of modern anti-phishing efforts, about 11 + 3% of users will read a spoofed email, click the link it contains, and enter their login credentials. In addition, Garera et al: [11] found that on average, 8.24% of users become victims after visiting phishing sites. If we use 10% as a realistic value for the cheat-to-click ratio, the real-to-all ratio becomes 1/10S . Thus, if the value of the set size S is 10, a real credential will be hidden among 100 bogus credentials. Moreover, it is plausible to assume that the cheat-to-click

ratio will decrease in the long run due to technical advances and educational efforts— a trend that favors Bogus Biter. Given the indiscernibility achieved by Bogus Biter, we now analyze the probability and the expected number of tries for a phisher to single out a certain number of real credentials by verifying them at the legitimate web site. Since each set of S credentials are submitted by Bogus Biter from a user's browser within a few milliseconds, a phisher can easily group the collected credentials by sets and verify them. If a set of S credentials is submitted from a victim's browser, the real credential will be singled out by a phisher with an expected number of (S+1)/2 tries. However, because a phisher cannot discern which set includes a real credential, the phisher has to verify all sets of the collected credentials in order to single out as many real credentials as possible. Considering the very low cheat-to-click ratio, without loss of generality, we simplify our analysis by mixing together all sets of the collected credentials. Let n be the total number of credentials collected at a phishing site, and m be the number of real credentials revealed by victims. Let Xk be the discrete random variable representing the number of tries performed by the phisher to single out k real credentials. The probability and expectation for Xk are described in Formula (1) and Formula (2), respectively, where

$$\text{where } P_{r}(X_k = i) = \frac{\binom{m}{k}\binom{n-m}{i-k}}{\binom{n}{i}} \quad \text{and } k = (1, 2, \ldots, m). \qquad (1)$$

$$E[X_k] = \sum_{i=k}^{n-m+k} i \, P_r(X_k = i) \qquad (2)$$

For example, we use 10% as the cheat-to-click ratio and 10 as the value of the set size S. If there are six real credentials hidden among all the collected credentials, the expected number of tries for a phisher to single out one real credential, i.e. E[X1], is 86, and the expected number of tries for a phisher to single out all the six real credentials is 515. This example indicates that Bogus Biter has the potential to feed a relatively large number of bogus credentials into a phishing site and well hide victims' real credentials among bogus credentials.

Indiscernibility

The indiscernibility requirement has two implications: the submission actions initiated from victims' browsers are indiscernible from the submission ac-tions initiated from security-conscious users' browsers, and victims' real credentials are indiscernible from

bogus credentials generated by Bogus Biter.

For a victim who ignores a browser's phishing warning, Bogus Biter first intercepts the credential submission HTTP request before it is sent out. Next, Bogus Biter creates S 1 bogus credentials based on the victim's real credential and spawns S 1 new HTTP requests based on the original HTTP request. Each of the S 1 spawned requests is exactly the same as the original request, except for carrying a bogus credential instead of a real one. Then, Bogus Biter inserts the original HTTP request into the S 1 spawned requests and sends out all the S requests within a few mil-liseconds. Finally, Bogus Biter interprets and properly pro-cesses the returned HTTP responses so that a phishing site cannot identify the differences between the S submissions.

For a security-conscious user who accepts a browser's phishing warning, Bogus Biter first imitates a victim's behavior by entering a generated bogus credential into the phishing page and submitting it. Next, similar to the above case for a real victim, Bogus Biter intercepts this original HTTP request, spawns S 1 new HTTP requests, and generates the corresponding S 1 bogus credentials as well. Finally, Bogus Biter sends out the S requests and processes the returned responses in the same way as it does for a victim, thereby making it hard for a phisher to distinguish these submissions from those initiated from a victim's browser. As for bogus credential generation, Bogus Biter uses the original credential as the template to generate the S – 1 bogus credentials. For a victim, the original credential is the victim's real credential and thus is ready to use. For a security-conscious user, the automatically generated original credential should be similar to a human's real credential. In current design, Bogus Biter randomly generates a username/password pair as the original credential. For the remaining S-1 bogus credentials, a specific rule should be followed to generate them so that neither a human nor a computer can easily discern which the original credential is and which are the rest.

Usability

In terms of usability, the major advantage of Bogus Biter is its transparency to users. Meanwhile, because Bogus Biter only needs to submit some extra bogus credentials to a suspected phishing site and does not contact any third-party service, it will not cause any security or privacy problems. The main usability concerns come from the scenario of a

false positive (i.e., a legitimate web site is wrongly classified as a phishing site). While the occurrence of false positives

is rare for Firefox 2, IE 7, and recent detection techniques as mentioned in Section 2, Bogus Biter should eliminate or reduce the possible side-effects on users' access to mis-classified legitimate web sites. The first side-effect is that submitting a set of S login requests and waiting for responses will induce an additional delay to users. To reduce the delay, Bogus Biter sends out all the S requests within a few milliseconds, so that the roundtrip times of the S submissions can be overlapped as much as possible. Accordingly, as long as the set size S is not too large, the additional delay incurred by Bogus Biter should be minimal and unobtrusive. Our experimental results in Section 5 confirm that the additional delays are negligible. The second side-effect is that a user's real account may be locked because multiple login requests are submitted from the user's browser to a legitimate web site within a few milliseconds. To defend against password guessing attacks, some web sites may lock a user's account for a period of time after several failed login attempts. However, because all the usernames are different for the S login requests sent out by Bogus Biter, the "account with many failed login attempts" alarm will not be triggered as discussed in [20]. Our experiments on 20 legitimate web sites confirm that account Locking is not a concern for Bogus Biter. The third side-effect is that a user may be asked to complete a CAPTCHA [5] test, for the same reason that multiple login requests are submitted from the user's browser within a few millisecond. Some web sites may resort to this mechanism to counter password guessing attacks or denial of service attacks. However, in our legitimate site experiments where false positives are assumed to occur, no CAPTCHA test is triggered if the set size S is not greater than 10, and only two of the 20 web sites ask a user to do a CAPTCHA test if the set size S is greater than 10.

Defensive Line

Given the indiscernibility of BoguBiter, phishers cannot single out real credentials without verifying the collected credentials one by one at legitimate web sites. Moreover, with the unique design of Bogus Biter, the forced verification process, either manually or automatically conducted, will help legitimate sites to detect victims' stolen credentials and provide fraud protection in a timely manner.

Working Mechanism

Bogus Biter makes such a defensive feature feasible by imposing a correlation requirement upon the generation of the S - 1 bogus credentials, in addition to the indiscernibility requirement.

Correlation Requirement: Based on the original credential, a specific rule is applied to generate the S – 1 bogus credentials. This rule must guarantee that the S credentials in a set are correlated: given any one of them, we can reversely derive a small superset that includes all the S credentials.

Implementation

We implemented Bogus Biter as a Firefox extension in JavaScript and C++, and seamlessly integrated it with the built-in phishing protection feature of Firefox 2 [36]. Bogus Biter consists of four main modules. The information extraction module extracts the username and password pair and its corresponding form element on a login page by analyzing Document Object Model (DOM) objects. The bogus credential generation module generates bogus credentials based on an original credential. The request submission module spawns multiple HTTP requests and submits them

to phishing sites. It uses XMLHttpRequest objects to create internal HTTP channels and submit HTTP requests behind the screen. By carefully performing request initialization, message body replacement, header fields setting, and header fields reordering, this module meets the indiscernibility and usability requirements of Bogus Biter. Finally, the response process module correctly matches responses to their corresponding requests and properly processes them.

Evaluation

We conducted three sets of experiments to evaluate the potential efficacy of the proposed anti-phishing approach and our reference implementation.

Test Bed Experiments

In the test bed experiments, we set up an Apache 2 web server in a Linux machine and hosted over twenty phishing web pages on it. We used Bogus Biter to send various login requests to these phishing web pages either directly or through proxies. By examining both request logs and request

contents at the web server, we verified that all the S requests in a set are exactly the same, except for the credentials carried in the request bodies.

Phishing Site Experiments

In the phishing site experiments, we ran Bogus Biter against 50 verified phishing sites chosen from Phish Tank [41]. For each phishing site, when it was online, we tested Bogus Biter with four different set sizes of 4, 8, 12, and 16. Our major experiential findings are summarized as follows. First, Bogus Biter is capable of attacking all the 50 phishing sites. Acting as either a victim or a security-conscious user, Bogus Biter always works correctly. It sends out all the S requests within 10 milliseconds, and then properly processes their responses. In rare cases that phishing sites were not correctly detected by Firefox 2, we manually corrected the detection results to trigger Bogus Biter. Second, the delay caused by Bogus Biter is minimal when the set size S is 4 or 8. Here the delay means the submission interaction time difference between using Bogus Biter and not using Bogus Biter. The submission interaction time is the time elapsed between the transmission of the first request and the reception of the last response. Figure 3 depicts the percentage of phishing sites versus the delay caused by Bogus Biter under four different set sizes. We can see that if the set size S is 4 or 8, for over 85% of phishing sites, the delay is less than 4 seconds. This delay measure is common to either a security-conscious user or a victim, but the delay effect is different. A security-conscious user is unaware of such a delay because the user is actually redirected to a default web page by Firefox. A victim may perceive this delay while waiting for the response from the phishing site. Nevertheless, it is worthwhile adding a small delay on revealing a victim's credential, in order to make it less likely for phishers to succeed.

Third, phishing sites take three different response actions after receiving a user's credential submission request. Among 50 phishing sites, 38 of them simply redirect a user to the invalid login pages of the targeted legitimate web sites; 11 of them keep a user at their local sites by using more faked web pages; and the last phishing site is very tricky because it verifies the received credential in real time at the legitimate web site and then sends back a response based on the verification result. All three types of response actions attempt to continue deceiving a victim and prevent the victim from realizing that an attack has happened, but the third type of response action is more deceptive. The defensive line of Bogus Biter indeed provides a good opportunity for a legitimate web site to defend against such attacks in real time.

Related Work

Basically the various client-side anti-phishing techniques can be classified into three different approaches. The first approach focuses on building tools or toolbars to enhance the security of a login process. Ye and Smith [30] designed a prototype of "Trusted Path" to convey relevant trust signals from a web browser to a human user. Dhamija and Tygar [5] proposed "Dynamic Security Skins" to allow a legitimate web site to prove its identity in a way that is easy for a user to verify but hard for a phisher to spoof. Ross et al: [22] designed PwdHash to transparently produce different passwords for different domains, so that passwords stolen at a phishing site are not useful at a legitimate web site. Wu et al: [29] introduced "Web Wallet" to direct an alternative safe path to a user if the user's intended web site does not match the current web site. Adida [1] proposed BeamAuth to use a secret token in a URL fragment identifier as a second factor for web-based authentication. These tools are very helpful, but users must be well trained to use them and must change some of their login habits.

The second approach focuses on improving the accuracy of automatic phishing detection techniques. Chou et al: [4] built Spoof Guard to compute spoof indexes using heuristics and to provide warnings for suspected phishing web sites. Recent work by Zhang et al: [33] and Garera et al: [11] demonstrate that heuristic-based techniques can correctly identify over 90% of phishing pages with about 1% false positives. Many other automatic phishing detection tools or toolbars have been developed, and both Firefox 2 and IE 7 have automatic phishing detection as a built-in feature. The evaluation of popular automatic phishing detection tools, toolbars, and web browser features can be found in [17, 32, 35, 37]. Researchers have also sought to develop non-preventive anti-phishing approaches. Florˆencio and Herley [8] proposed a password rescue scheme which relies on client-side reporting and server-side aggregation to detect and protect stolen credentials. However, this scheme can only make a detection decision after several users become victims, and it also raises privacy concerns by using an extra server to collect user activity information. Parno et al: [19] proposed a Phoolproof anti-phishing mechanism. Although their mechanism eliminates reliance on perfect user behavior, a trusted mobile device must be used to perform mutual authentications. Birk et al: [2] introduced an "active phishing tracing" method, which injects fingerprinted credentials into phishing sites to trace money laundering. Their method can support forensic

analyses and enforce judicial prosecutions, but it cannot directly protect phishing victims.

Conclusion

We introduced Bogus Biter, a new client-side ant phishing tool to automatically protect vulnerable users by injecting a relatively large number of bogus credentials into phishing sites. These bogus credentials hide victims' real credentials, and force phishers to verify their collected credentials at legitimate web sites. The credential verification actions initiated by phishers, in turn, create opportunities for legitimate web sites to detect stolen credentials in a timely manner. Bogus Biter is transparent to users and can be seamlessly integrated with current phishing detection and warning mechanisms on web browsers. We implemented Bogus Biter as a Firefox 2 extension and evaluated its effectiveness and usability. Phishing is a serious security problem today, and phishers are smart, economically motivated, and adaptable. We must therefore actively pursue different approaches and promote the cooperation of different solutions. The effectiveness of Bogus Biter depends on many factors, but we believe its unique approach will make a useful contribution to the anti-phishing research.

Reference

1. B. Adida. BeamAuth: Two-factor web authentication with a bookmark. In Proceedings of the CCS, pages 48–57, 2007.
2. D. Birk, M. Dornseif, S. Gajek, and F. Gr¨obert. Phishing phishers - tracing identity thieves and money launderer. Technical Report, Horst-G¨ortz Institute of Ruhr-University of Bochum, 2006.
3. A. Bortz, D. Boneh, and P. Nandy. Exposing private information by timing web applications. In Proceedings of the WWW, pages 621– 628, 2007.
4. N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell. Clientside defense against web-based identity theft. In Proceedings of the NDSS, 2004.
5. R. Dhamija and J.D.Tygar. The battle against phishing: Dynamic security skins. In Proceedings of the SOUPS, pages 77–88, 2005.
6. J. S. Downs, M. B. Holbrook, and L. F. Cranor. Decision strategies and susceptibility to phishing. In Proceedings of the SOUPS, pages 79–90, 2006.
7. S. Egelman, L. F. Cranor, and J. Hong. You've been warned: An empirical study of the effectiveness of web browser phishing warnings. In Proceedings of the CHI, pages 1065–1074, 2008.
8. D. Flor^encio and C. Herley. Password rescue: A new approach to phishing prevention. In Proceedings of the HOTSEC, 2006.
9. D. Flor^encio and C. Herley. A large-scale study of web password habits. In Proceedings of the WWW, pages 657–666, 2007.
10. D. Flor^encio, C. Herley, and B. Coskun. Do strong web passwords accomplish anything? In Proceedings of the HOTSEC, 2007.
11. S. Garera, N. Provos, M. Chew, and A. D. Rubin. A framework for detection and measurement of phishing attacks. In Proceedings of the WORM, 2007.
12. T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. Commun. ACM, 50(10):94–100, 2007.
13. M. Jakobsson and S. Myers. Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft. Wiley- Interscience, ISBN 0-471-78245-9, 2006.
14. M. Jakobsson and J. Ratkiewicz. Designing ethical phishing experiments: a study of (ROT13) rOnl query features. In Proceedings of the WWW, pages 513–522, 2006.

15. M. Jakobsson and A. Young. Distributed phishing attacks. In Proceedings of the workshop on Resilient Financial Information Systems, 2005.
16. P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nung. Protecting people from phishing: The design and evaluation of an embedded training email system. In Proceedings of the CHI, pages 905–914, 2007.
17. C. Ludl, S. McAllister, E. Kirda, and C. Kruegel. On the effectiveness of techniques to detect phishing sites. In Proceedings of the DIMVA, 2007.
18. T. Moore and R. Clayton. Examining the impact of website takedown on phishing. In Proceedings of the APWG eCrime Researchers Summit, 2007.
19. B. Parno, C. Kuo, and A. Perrig. Phoolproof phishing prevention. In0 Proceedings of the Financial Cryptography, pages 1–19, 2006.
20. B. Pinkas and T. Sander. Securing passwords against dictionary attacks. In Proceedings of the CCS, pages 161–170, 2002.
21. Rachna Dhamija and J.D.Tygar and Marti Hearst. Why phishing works. In Proceedings of the CHI, pages 581–590, 2006.
22. B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell. Stronger password authentication using browser extensions. In Proceedings of the USENIX Security Symposium, pages 17–32, 2005.
23. S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators: An evaluation of website authentication and the effect of role playing on usability studies. In Proceedings of the IEEE Symposium on Security and Privacy, pages 51–65, 2007.
24. S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge. Anti-Phishing Phil: the design and evaluation of a game that teaches people not to fall for phish. In Proceedings of the SOUPS, pages 88–99, 2007.
25. L. von Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using hard AI problems for security. In Proceedings of the Eurocrypt, pages 294–311, 2003.
26. T. Whalen and K. M. Inkpen. Gathering evidence: use of visual security cues in web browsers. In Proceedings of the conference on Graphics interface, pages 137–144, 2005.
27. M. Wu. Fighting Phishing at the User Interface. PhD thesis, MIT, 2006.
28. M. Wu, R. C. Miller, and S. L. Garfinkel. Do security toolbars actually prevent phishing attacks? In Proceedings of the CHI, pages 601–610, 2006.

29. M. Wu, R. C. Miller, and G. Little. Web Wallet: preventing phishing attacks by revealing user intentions. In Proceedings of the SOUPS, pages 102–113, 2006.
30. Z. E. Ye and S. Smith. Trusted paths for browsers. In Proceedings of the USENIX Security Symposium, pages 263–279, 2002.
31. K.-P. Yee and K. Sitaker. Passpet: convenient password management and phishing protection. In Proceedings of the SOUPS, pages 32–43, 2006.
32. Y. Zhang, S. Egelman, L. F. Cranor, and J. Hong. Phinding phish: Evaluating anti-phishing tools. In Proceedings of the NDSS, 2007.
33. Y. Zhang, J. Hong, and L. Cranor. CANTINA: A content-based approach to detecting phishing web sites. In Proceedings of the WWW, pages 639–648, 2007.
34. APWG: Phishing Scams by Targeted Company. http://www.millersmiles.co.uk/scams.php.
35. Firefox 2 Phishing Protection Effectiveness Testing. http://www.mozilla.org/security/phishing-test.html.
36. Firefox Phishing Protection. http://www.mozilla.com/en-US/firefox/phishing-protection/.
37. Gone Phishing: Evaluating Anti-Phishing Tools for Windows. http://www.3sharp.com/projects/antiphishing/gone-phishing.pdf.
38. Inaccessibility of CAPTCHA. http://www.w3.org/TR/turingtest/.
39. Know your Enemy: Phishing. http://www.honeynet.org/papers/phishing/.
40. Microsoft Phishing Filter. http://www.microsoft.com/protect/products/yourself/.
41. Phish Tank. http://www.phishtank.com/.