A Novel Anti-Phishing Framework Based on Honeypots

Shujun Li
Department of Computer and Information Science
University of Konstanz
Universitätsstraße 10, D-78457 Konstanz, Germany
www.hooklee.com

Roland Schmitz
Department of Computer Science and Media
Stuttgart Media University
Nobelstrasse 10, D-70569 Stuttgart, Germany
schmitz@hdm-stuttgart.de

***Abstract*—As a powerful anti-phishing tool, honeypots have been widely used by security service providers and financial institutes to collect phishing mails, so that new phishing sites can be earlier detected and quickly shut down. Another popular use of honeypots is to collect useful information about phishers' activities, which is used to make various kinds of statistics for the purposes of research and forensics. Recently, it has also been proposed to actively feed phishers with honeytokens.**

In the present paper, we discuss some problems of existing anti-phishing solutions based on honeypots. We propose to overcome these problems by transforming the real e-banking system itself into a honeypot equipped with honeytokens and supported by some other kinds of honeypots. A phishing detector is used to automatically detect suspicious phishers' attempts of stealing money from victims' accounts, and then ask for the potential victims' reconfirmation. This leads to a novel anti-phishing framework based on honeypots.

As an indispensable part of the framework, we also propose to use phoneybots, i.e., active honeypots running in virtual machines and mimicking real users' behavior to access the real e-banking system automatically, in order to submit honeytokens to pharmers and phishing malware. The involvement of phoneybots is crucial to fight gainst advanced phishing attacks such as pharming and malware-based phishing attacks.

***Index Terms*—phishing; honeypot; honeytoken; phoneypot; phoneytoken; phoneybot; online banking; money mule;**

I. Introduction

The prevalence of e-commerce in today's digital world opens a door for various cyber crimes that we have never seen before. Among all the cyber crimes targeting e-banking systems, phishing attack has become one of the most serious threats [1]–[3]. In the main form of phishing attack, the criminals (called phishers) setup fake e-banking/e-payment web sites, and then send phishing emails to potential victims, who may be lured to access the phishing sites and expose their sensitive credentials to the phishers. The credentials harvested by the phishers normally include bank account numbers, passwords/PINs, e-banking TANs (Transaction Authentication Numbers), credit card numbers and security codes, social security numbers, and so forth. With the collected credentials, the phishers can login the genuine e-banking/e-payment system to steal the victim's money. There are also many other more advanced forms of phishing attack, such as the following:

- phishers get phishing sites indexed by some search engines (via some Search Engine Optimization tricks) and then wait for victims to visit them [4];
- phishers use cross-site-scripting (XSS) to inject links of phishing sites to legitimate sites [5], [6];
- *spy-phishing* (or *malware-based phishing*): phishers depend on spyware/malware like trojan horses and keyloggers to collect sensitive credentials [7];
- *pharming*: phishers misdirect potential victims to phishing sites through DNS poisoning [8], [9].

Phishers can also tailor the contents of the phishing mails and even those of the phishing sites for targeted victims, which is called *spear phishing* or *context-aware phishing* in the literature [1, Chapter 6]. This kind of phishing attack becomes much easier nowadays, because more and more personal information is publicly available at online social networks. It is expected that even more advanced forms of phishing attack may soon appear in near future.

To fight against phishing, a lot of countermeasures have been proposed by researchers, security service providers, financial institutes, related governmental authorities and also law makers. Among all the countermeasures that financial institutes can adopt, two-factor users authentication and SSL/TLS-based authentication of e-banking web sites are most widely deployed. Some financial institutes also adopt other forms of mutual authentication to enhance security [1, Chapter 9]. In addition, due to the need of fighting against money laundering, nowadays most financial institutes are maintaining AML (anti-money laundering) software as part of the e-banking system to monitor transactions and detect suspicious money laundering activities [10]. While all of these countermeasures help financial institutes against phishing attacks, they are not enough to frustrate more advanced attacks such as spy-phishing and pharming.

In this paper, we propose a novel anti-phishing framework based on a seamless incorporation of different kinds of honeypot-based anti-phishing techniques and the existing e-banking systems. Compared with traditional anti-phishing

 978-1-4244-4626-1/09/$25.00

solutions based on honeypots, our proposed framework can solve some problems as described in Sec. III of this paper. The combination of different kinds of honeypots makes it possible for a phishing detector embedded in the e-banking system to automatically determine suspicious phishing attacks and then seek the potential victims' re-confirmation. As a result, the victim is rescued timely, and the phishing attack is frustrated in the following senses: 1) the money a phisher can steal is drastically limited; 2) the risk a phisher or his money mule (financial agents) will be caught is considerably increased.

The rest of this paper is organized as follows. In the next section we briefly overview existing measures against phishing and some limitations about their performance. We also show why honeypots may be a good candidate for a better solution against phishing, and briefly introduce some existing solutions. Section III discusses some problems with existing anti-phishing solutions based on honeypots with some suggested solutions. In Sec. IV, the proposed anti-phishing framework is presented in detail, based on the solutions suggested in Sec. III. We show how different components of the framework work together to offer a better anti-phishing performance. The last section concludes the paper and gives our plan of future work. We also include an appendix to show how PIN/TAN, the most widely-used two-factor user authentication system in German banking industry, can be enhanced to help improve the performance of the proposed anti-phishing framework.

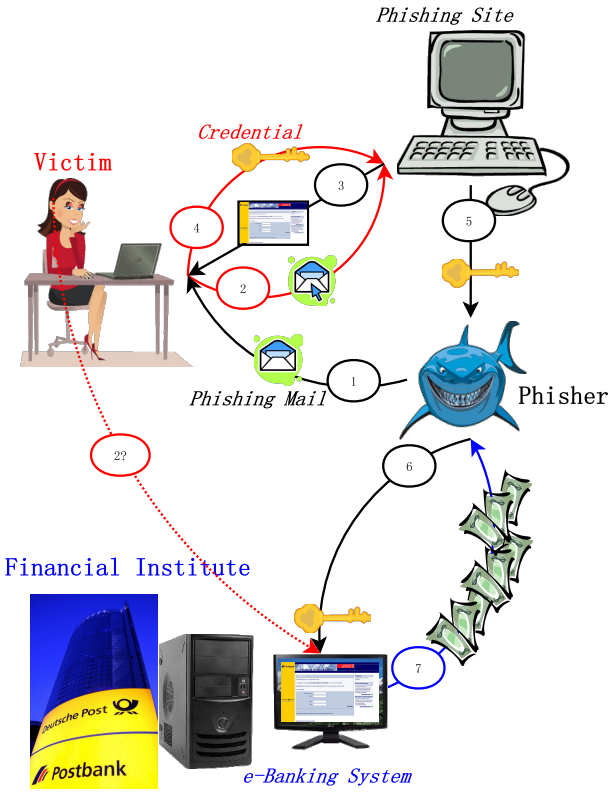

Fig. 1. Phishing information flow.

II. Related Work

A very good summary of existing measures against phishing can be found in [1]. Given the information flow of a typical phishing attack shown in Fig. 1, we can see seven different steps that can be cut down to stop a phishing attack. In the following, we give a list of some existing countermeasure working at different steps of the phishing information flow, and discuss the limits of their performance.

- *Step 1 – blocking the information flow from phishers to potential victims*: phishing email detection and filtering, email authentication, anti-malware software, cousin domain rejection, and so on.
  Limitations: In principle, no phishing email filter can block *all* phishing mails from potential victims. Email authentication is often less useful in practice since only very few financial institutes are using secure emails and most average users have no necessary knowledge to authenticate the source of an email. Anti-malware software has the same problem as phishing email filters.
- *Steps 2-4 – avoiding credential leakage*: user education, phishing site warning, inconsistent DNS information detection, cross-site/injected script rejection, mutual authentication, trusted path between user and web browser, delayed password disclosure [11], and so on.
  Limitations: Many countermeasures in these steps depend on end users to make a final decision. Because users are not very dependable to properly respond to security indicators/alerts about phishing [12]–[14], these countermeasures may not work as effectively as expected.
- *Step 5 – preventing phishers from getting stolen credentials*: early detection and quick takedown of phishing sites, fake credential submission, password rescue [15], and so on.
  Limitations: As recently reported in [16], commercial take-down services do not work very well if we see all the phishing sites as a whole. Fake credential submission is based on a specific kind of honeypots – honeytokens, whose problems will be discussed later in the next section. Password rescue focuses on a probably earlier detection of phishing sites, but still depends on other partners to quickly takedown the detected phishing sites.
- *Step 6 – making stolen credentials useless*: two-factor user authentication, password hashing [17], transaction monitoring and reconfirmation, and so on.
  Limitations: Two-factor user authentication and password hashing can frustrate phishers effectively, but they usually require extra software/hardware at the user side, which becomes a problem if some users are not willing to cooperate or when the software/hardware is damaged, lost, stolen or poisoned. Transaction monitoring and reconfirmation can be a good countermeasure, but it has the same problem of the phishing email filter.
- *Step 7 – preventing phishers from getting the stolen money, or catching phishers*: transaction authentication, intentional transaction delay, law enforcement, and so on.
  Limitations: Transaction authentication is a second de-

fence in addition to user authentication, which normally needs either an additional hardware device [18], [19] or an additional trusted channel (like the cellular network) [20], thus leading to higher implementation costs and worse usability. The intentional transaction delay cannot be very long to ensure the quality of e-banking services, so its performance is quite limited as long as the lifespan of some phishing sites is still longer than the delay. Law enforcement is the last hope of catching phishers (or their money mules [21]), but its performance is less effective than technological countermeasures [22], [23].

We notice the following two key problems that limit the performance of many countermeasures:

1) a 100% automatic detection of any statistical feature of phishing is theoretically impossible;
2) any countermeasures depending on end users' judgement and action may fail to some extent.

The first problem shows that we cannot completely depend on any statistical feature to detect phishing, and the second one implies that anti-phishing measures working at the user level should be used only as complements to a complete anti-phishing system. In other words, we need to find a better way to phishing detection and to make the kernel of an anti-phishing solution independent of end users' distinction. While algorithms based on statistical features cannot achieve a 100% detection rate, there does exist one anti-phishing measure which may achieve a detection rate very close to 100% – honeypots. It is not surprising because honeypots are information resources whose value fully lies in unauthorized/illicit use. Honeypots are also independent of end users, since they interact only with attackers.

According to the definition given in [24], "*a honeypot is a closely monitored computing resource that we want to be probed, attacked, or compromised*". A honeypot does not need to be a physical machine, but can be anything that is available to attackers as a computing resource such as a sub-network, a computer or a web service. A honeypot can also be a piece of electronic information (i.e., any digital entity), which is a special form of honeypot called *honeytoken* [25]. The most essential fact about all honeypots is that they are decoys to lure and probably also track attackers.

While honeypots and honeytokens can be used to lure any malicious attackers, it is worth nothing that they can also be used to lure phishers. In fact, one major way for early detection of phishing sites is to use spamtraps (i.e., honeypots against spams) to collect phishing emails [1, Section 5.4 and Chapter 11]. Some researchers have also suggested the use of *phoneytokens* ("phishing honeytokens"), which are sent to phishing sites as fake credentials to confuse phishers [26] and/or collect information about phishers' activities [27].

The BogusBiter system proposed in [28] does not use the term "honeytoken" or "phoneytoken", but the mechanism is the same. The BogusBiter system also has a server-side program helping detect stolen credentials. In order to submit phoneytokens to phishing sites, the end users have to install a special plugin in their web browsers, which may become a problem since some users may not cooperate.

Recently, a major step about the use of phoneytokens was reported in [29], [30], where it is proposed to setup a simulated e-banking system as a *phoneypot* ("phishing honeypot") to trace how phishers use phoneytokens, which are submitted by financial institutes to already known phishing sites. A phisher will be led to the phoneypot instead of the real e-banking system when they try to access the e-banking system with a phoneytoken. The collected profiling information about phishers are recorded for future forensic purpose. A notable feature of the design is the isolation of the phoneypot and the real e-banking system. To avoid being detected, the phoneypot shares the same domain name as the real e-banking system, and the load-balancing technique [31], [32] is used to redirect phishers to the phoneypot and legitimate users to the real e-banking system.

A similar idea to the one in [29], [30] was proposed by Herley and Florêncio [33] for the purpose of fighting against brute-force attack. The main goal is to make bulky guessing attack on a large number of user accounts less feasible. For each wrongly guessed PIN, the attacker will be led to a honeypot account, so he has to further verify if a "broken" account is a real one. This strategy cannot be used for anti-phishing, since phishers lure PINs directly from the legitimate users rather than randomly guess them.

In addition to the above academic research work, there are also commercial anti-phishing solutions based on honeypots. RSA Security Inc. has a security service called FraudAction, which exploits RSA's "Randomized Credentials Technology" (RCT) to feed dummy credentials to phishing sites and crimeware applications as baits to collect phishers' profiling information [34]. Similarly, MarkMonitor Inc. also has two services called Dilution and Phish Tagging, which can submit fake or bank-generated phoneytoken credentials to phishing sites for the purpose of confusing phishers or monitoring phishers' activities [35].

III. Anti-Phishing Honeypots: Problems and Solutions

Although all the existing anti-phishing measures based on honeypots are able to fight against phishers to some extent, there are still some problems to be further overcome. In this section, we discuss these problems and propose some practical solutions, which form the foundation of the novel anti-phishing framework described in Sec. IV.

A. Gap between spamtraps and phoneytokens

We notice that spamtraps and phoneytokens are often used separately. Generally spamtraps are used only as a tool to detect phishing emails (i.e., URLs of phishing sites included in the phishing emails), and submissions of phoneytokens are triggered after a phishing site is confirmed (often by a human inspector). It is very common that the spamtrap detecting

a phishing site is maintained on a machine different from the one on which the phoneytoken submission is done. In addition, there is often a considerable difference between the time a phishing site is detected by a spamtrap and the time a phoneytoken is submitted. In contrast, most potential victims of phishing behave in a different manner: after reading the phishing mail, they often immediately access the phishing sites and submit their credentials on the same machine. Therefore, phishers may make use of the differences to tell phoneytokens from genuine credentials apart. To this end, phishers need to know when and where a potential victim reads a phishing mail. Information about this can be obtained by embedding a web bug in each phishing mail [36]. The web bug can even be tailored to mark each phishing mail and thus each receipt.[1]

To meet this gap, we believe that spamtraps themselves have to be in charge of submitting phoneytokens to phishing sites. In other words, phoneytokens should be an essential part of spamtraps. In addition, to fully disable phishers' capability of detecting spamtraps, phoneytokens should be submitted by human monitors of spamtraps, who mimic the typical behavior of a phishing victim.

B. Online verification of phoneytokens

Simply submitting phoneytokens to phishing sites may not cause too much trouble to phishers. If the phoneytokens are just randomly generated credentials that cannot be used to login the real e-banking system, a phisher can simply write an automated script to verify all the harvested credentials. Although financial institutes can use CAPTCHAs ("Completely Automated Public Turing test to tell Computers and Humans Apart") [37] to make such automated tests more difficult, it is doubtful if CAPTCHAs will really work well. On one hand, it has been known that many CAPTCHA systems are not strong enough against automated attacks [38]–[42]. On the other hand, even if a CAPTCHA system is strong enough against all known automated attacks, phishers can still use the "stealing cycles from humans" tactic to circumvent CAPTCHAs. In 2007, a trojan horse was found to use this tactic against Yahoo!'s CAPTCHA system [43].

To solve the above problem, the real e-banking system should be "honeyed", i.e., phishers should be allowed to access the real e-banking system with phoneytokens. Furthermore, it should not be easy for a phisher to distinguish a phoneytoken and a genuine credential even after logging into the "honeyed" e-banking system. This problem is not trivial. In the next section, we will discuss how it can be solved in our proposed framework with greater detail.

C. Gap between the phoneypot and the real e-banking system

In most existing solutions, honeypots are used independently of the real e-banking systems. Although the framework proposed in [29], [30] runs a simulated e-banking system as a honeypot, it is also physically isolated from the real e-banking system. By separating the real e-banking system and the honeypot, there is always a risk that the phisher may figure out a way to distinguish the phoneypot. The simplest way is to test a collected credential by transferring a small amount of money from the tested credential to a known genuine bank account. To avoid this problem, the phoneypot must be able to communicate with the real e-banking system and show the transaction result correctly. In Section 6.2 of [30], the authors actually mentioned this approach of detecting a phoneypot, but it was not clarified how the phoneypot can interact with the real e-banking system. We argue that the phoneypot has to be the real e-banking system itself rather than an isolated simulation system. This will help prolong the lifespan of a honeytoken, so that more victims can be rescued and phishers can be defeated more easily. In the next section, we will explain how this can be done in our proposed anti-phishing framework.

D. Spamtraps can't defeat advanced phishers

While spamtraps are mainly used to lure phishing emails, they can do nothing against pharming, malware-based phishing, or other more advanced phishing attacks.

To better fight against advanced phishers, we propose to setup a special kind of phoneypots, which can actively feed phoneytokens to pharmers and phishing malware. These phoneypots are automated programs running in virtual machines without any anti-malware and anti-pharming protection. From time to time they log into the e-banking system with the phoneytokens assigned to them. To avoid that phishers (or phishing malware) detect any suspicious feature of these special phoneypots, they should mimic the web surfing behavior of an average bank customer. The average behavior of using the e-banking system can be stored by a dynamic file, which reflects the average behavior of all legitimate customers who are using the real e-banking system. A human manager is in charge of the maintenance of the dynamic file. In this paper we call such special phoneypots *phoneybots*, since they work like web robots. Phoneybots can work together with spamtraps so that they can also respond to phishing attacks based on emails.

E. Problems with outsourcing

It is a common practice of financial institutes to outsource some tasks to external contractors. However, we believe that outsourcing the whole anti-phishing task or part of it to a company is less efficient and may cause reaction delay when real phishing attacks are detected. Since the typical lifespan of most phishing sites is only several days [16], [44], it is crucial for financial institutes to make immediate action once active phishing attacks are detected by spamtraps. This suggests that there should be a direct link between the spamtraps and the e-banking systems, both of which should be under full control of the financial institutes instead of the outsourcing contractors. Since maintaining spamtraps is not very technically complicated, we believe financial institutes can save the overall costs by having their in-house maintenance team, which can be part of their security or information technology departments.

[1] Interestingly, web bugs can also be used against phishers in a very similar way as suggested in [27].

Another problem with outsourcing is the potential risk to customer privacy. The more tasks a financial institute outsources to external contractors, the more risks to customer privacy there will be. To maximize the performance of spamtraps and phoneybots, profiling information of users have to be shared with the outsourcing contractors, which may incur further concerns from the customer side. In our opinion, instead of offering anti-phishing services to financial institutes, security industry can develop anti-phishing software as add-value components of e-banking systems, sell them to financial institutes and provide technical support.

Furthermore, it is obvious that outsourcing will lead to a higher risk of insider attacks [45]. A very recent insider attack related to outsourcing is reported in [46]. An employee (i.e., insider) of Vodacom (a telecommunication service provider in South Africa) helped an SMS banking fraud syndicate to intercept SMS notifications sent from banks to customers. Many banks are subscribers of Vodacom, thus become victims of the insider attack. Although the SMS-based banking service may not be seen as outsourcing, it works in a way similar to outsourcing: the key security part of the system is under control of a third-party contractor (Vodacom in this case) rather than of the financial institute itself.

Although we argue that each financial institute should have full control of its anti-phishing solution, cooperation between different financial institutes and other organizations fighting against phishers should not be neglected. A recent report [16] has shown that lack of cooperation among different take-down service providers indeed harms the battle against phishers. We advocate establishing an information sharing network run by a cross-bank consortium, which should also embrace security service/software providers, related governmental authorities and international anti-phishing organizations like the Anti-Phishing Working Group (APWG). By anonymizing the information shared in the network, customers' privacy can be protected properly.

We see two important benefits of establishing such an information sharing network. Firstly, the shared information will allow all financial institutes to be better prepared for phishing attacks and thus rescue more victims when a real attack happens. It is obvious that the anti-phishing system deployed by each financial institute can adapt to new phishing information obtained from other partners of the cooperative network, thus leading to better performance. Such a network will help standardize anti-phishing measures and encourage a larger-scale deployment of anti-phishing systems, which will dramatically increase the successful rate of detecting phishers. Secondly, a cooperative network will help reduce the total anti-phishing costs of the whole banking industry and thus the average costs of each financial institute. For instance, the development and maintenance costs of anti-phishing systems can be reduced, since all the partners participating the network have access to much more useful information. In the next section, we will also show how different financial institutes and related authorities can cooperate to better fight against phishing in the proposed novel anti-phishing framework.

IV. The Anti-Phishing Framework

In the previous section we discussed some problems with existing anti-phishing based on honeypots and suggested some solutions to overcome those problem. These solutions lead to a novel anti-phishing framework that makes use of four different kinds of honeypots:

- a honeyed e-banking system as a phoneypot;
- a number of phoneytokens as fake credentials supported by the honeyed e-banking system;
- a number of spamtraps for attracting phishing emails and submitting phoneytokens to phishing sites;
- a number of phoneybots for submitting phoneytokens to pharmers and phishing malware.

Phoneytokens are the kernel of the framework, which are used by both the phoneybots and the honeyed e-banking system. A phishing detector is embedded into the honeyed e-banking system to automatically determine if a phishing attack is happening. Note that the honeyed e-banking system is not a pure honeypot according to the traditional definition, since it also has all the functions of a normal e-banking system. We can consider it as a *semi-honeypot* instead.

A diagram of the proposed anti-phishing framework is shown in Fig. 2. Different from the information flow in Fig. 1, now we consider a more complicated but more common model of money laundering of phishers – recruiting money mules to help them avoid being traced by financial institutes and authorities [2], [3]. In (the rare) case a phisher does not recruit mules, we can consider himself is the mule and the anti-phishing framework works even better.

A phisher may use some more sophisticated ways to launder the money [3, Chapter 6]. For instance, a phisher may make a big online purchase with the stolen credentials and then recruit mules to help him sell the goods out. The earned money is then sent to the phisher through a cash-delivery service like WesternUnion. However, this form of money laundering is mainly used by phishers who steal credit card numbers rather than credentials related to bank accounts. Since this alternative process of money laundering is much more complicated and last for a much longer time (thus with a much higher risk of being detected), we assume that very few (if any) phishers will use this way to launder the money stolen from bank customers. Therefore, we will not discuss this money-laundering model in the following part of the paper.

In the next subsection we first introduce all the steps of the proposed anti-phishing framework. Then, we discuss how an important step – user reconfirmation of suspicious transactions – should be implemented. Some interesting features of the framework are given in Sec. IV-C. Some performance factors of the anti-phishing framework are discussed in Sec. IV-D. Finally, in Sec. IV-E we explain how the phishing detector works in detail.

A. Step-by-Step Description

The anti-phishing framework is a systematic solution covering different steps of the whole phishing information flow. In

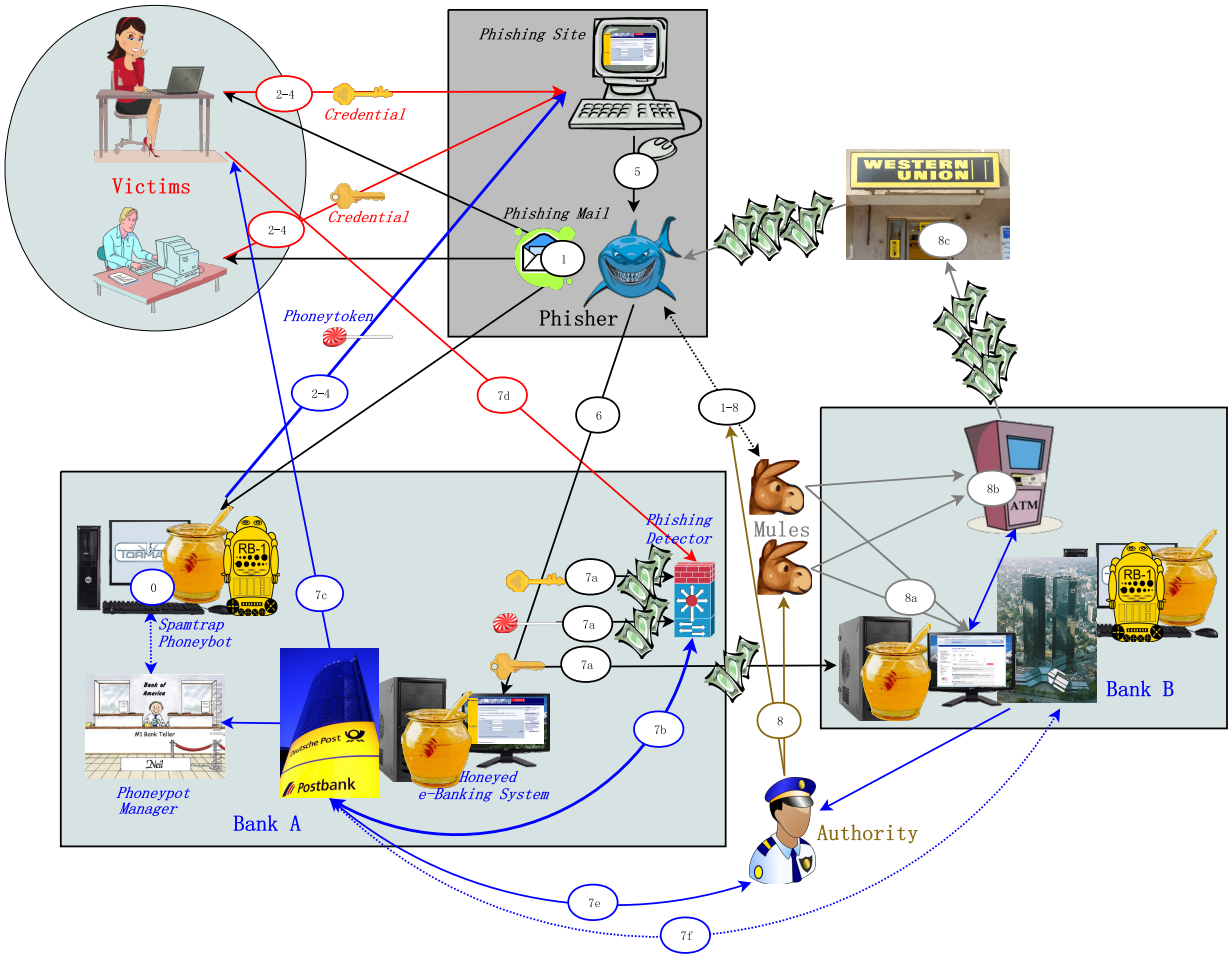

Fig. 2. The novel anti-phishing framework based on honeypots.

the following, we discuss steps involved in the proposed anti-phishing framework. All the steps are numbered to keep coincide with those shown in Fig. 1.

Step 0: The financial institute sets up $m \geq 1$ virtual machines running m phoneybots and m spamtraps. Each spamtrap/phoneybot is equipped with a phoneytoken. Following the average behavior of all bank customers, each phoneybot accesses the real e-banking system from time to time and makes virtual online transactions. A group of bank staff members manages the virtual machines, spamtraps and phoneybots, and monitors suspicious phishing emails.

Step 1: The phisher sends phishing mails to potential victims, or infects potential victims' computers with malware, or launches pharming attacks.

Steps 2–4: In a phishing attack based on phishing emails, the human manager submits the phoneytoken associated with the spamtrap to the phishing site. In an advanced phishing attack based on pharming or malware, a phoneytoken will be automatically collected by the phisher every time a phoneybot uses its phoneytoken to access the real e-banking system. Each phoneytoken should be submitted only once. A new phoneytoken will be created and assigned to each spamtrap/phoneybot by the human manager after the old one is consumed. Each spamtrap should also be updated after submitting a phoneytoken, because it may be detected in spear phishing attacks.

Step 5: The phisher collects some genuine credentials mixed with a number of phoneytokens from the phishing site, or from the phishing malware.

Step 6: The phisher logs in the e-banking system with the collected credentials/phoneytokens. He may then test the genuineness of each credential, since we have to assume (according to Kerckhoffs' principle) that he knows how the anti-phishing framework works.

Step 7a: The phisher tries to steal money from the victims' accounts, and also from the phoneytokens.

Step 7b: A phishing detector in the e-banking system monitors online transactions related to all phoneytokens. It issues an alert once it detects a phisher tries to transfer a considerable amount of money from a phoneytoken to a non-phoneytoken account. The receiver's account is then marked as highly suspicious "phishing account" (which is usually a mule's account opened at another financial institute).

Step 7c: The financial institute contacts potential victims (or

the victims contact the financial institute) for reconfirmation of all fund transfers from their bank accounts to any marked "phishing accounts". See Sec. IV-B for more discussions about this process.

Step 7d: The victims approve or reject the transactions. In the latter case, they ask for temporary locking their bank accounts, and then reset their credentials.

Steps 7e, 7f: The financial institute reports suspicious phishing activities to related authorities and/or its cooperative financial institutes (e.g., banks managing the accounts of the phisher's mules). The cooperative financial institutes may also offer feedback to help the phishing detector make a better decision for future phishing attacks.

Step 8: The phisher's mules check if they have received the money. If so, they withdraw the money at an ATM, and then remit the money to the phisher. During the period or even afterwards, the related authorities may start investigation according to the information provided by the financial institute in the previous step. Since this is the last step of the whole phishing information flow, the law enforcement is the only hope to get the stolen money back.

In the whole framework, detection of suspicious phishing accounts in Step 7b is the most important step, which depends on successful submission of phoneytokens in Steps 2–4.

B. User Reconfirmation

Since the user is not dependable in making cautious decisions, we should make the user reconfirmation process in Step 7c independent of the user's distinction, too. To this end, the reconfirmation of suspicious transactions should be done at the server side by the staff member of the financial institute, not at the client side by the user. Since the user's computer may not be secure in malware-based phishing attack, an out-of-band (OOB) channel should be used for the user reconfirmation process. Possible OOB channels include SMS, telephone, fax and post. The most practical and convenient OOB channel will be SMS or telephone. Given a specific OOB channel, the user reconfirmation process can be done as follows: the financial institute contacts the user via the OOB channel and asks her to explicitly re-confirm or reject the suspicious transaction. It is important that update of the contact information of the user should be forbidden in the e-banking system, otherwise the phishing malware will be able to change it in a man-in-the-middle attack. Instead, the contact information should be protected with a special mechanism based on an OOB channel. For instance, the financial institute can issue each user a special TAN list, which is used *only* for updating the contact information via telephone.

In case a user does not have a convenient way to be contacted or does not like to be contacted, the user reconfirmation process can be implemented in another way. First, the financial institute issues a special TAN list to the user in advance. When a reconfirmation is necessary, the e-banking system prompts the user (or actually the phisher) to call a toll-free telephone number of the financial institute, and press a specific TAN to re-confirm the transaction or press a random wrong TAN to reject it. The user may also be asked for other private information to further prove her identity over telephone.

C. Some Interesting Features

Compared with other anti-phishing solutions, our proposed framework has some interesting features.

Feature 1: A complete anti-phishing chain is constructed during the whole lifespan of a phishing attack (except for Steps 5 and 8, which are out of control of the financial institute).

Feature 2: The anti-phishing framework is independent of end users' distinction, but some modifications to the user interface may help enhance the performance of the framework (see the Appendix for an example).

Feature 3: Some other efforts can also (maybe drastically) improve the performance of the anti-phishing framework. For example, a new legislation can be made to allow the financial institutes to get back any money sent from a phoneytoken to a third party's bank account, which will help effectively reduce the potential costs the financial institutes have to bear.

Feature 4: Four different types of honeypots are involved and work closely to complement each other. Actually, there are two types of phoneytokens in the proposed anti-phishing framework: the normal phoneytokens and the shadow phoneytokens. The latter are short-lived phoneytokens as shadows of genuine bank accounts, which are used to delay exposure of normal phoneytokens and already rescued victims' accounts. In Sec. IV-E we report more details.

Feature 5: Instead of being completely dummy, phoneytokens have to support real transactions like genuine bank accounts, because a phisher may try to verify the genuineness of each collected credential (see Sec. IV-E2).

Feature 6: A phishing detector is embedded in the e-banking system to block suspicious money laundering activities of phishers via customer reconfirmation. It can be seen as an AML module enhanced by honeypots.

Feature 7: The phishing detector can also fight against more advanced forms of phishing attacks, as long as the phoneybots can successfully cheat phishers to collect the phoneytokens.

Feature 8: The phoneybots simulate the average behavior of bank customers according to information from human managers, so there is no chance for phishers to distinguish phoneybots from human users.

Feature 9: The phoneybots, the honeyed e-banking system and the phishing detector can collect profiling information of phishers. The information can be shared with different components of the anti-phishing framework to enhance the overall anti-phishing performance. It can also be shared with other anti-phishing bodies to help the whole society better fight against phishers.

Feature 10: The anti-phishing framework does not have any additional requirement (neither software nor hardware) at the user side. The user only needs to maintain an additional TAN list issued by the financial institute for updating contact information or user reconfirmation. In contrast, there must be some nontrivial changes at the server side, including support for phoneytokens, enhanced AML module with a phishing

detector and deployment of spamtraps and phoneybots. While these changes will definitely incur additional costs, we believe they deserve the efforts due to the enhanced performance against phishing attacks.

D. Performance Factors

The performance of the proposed anti-phishing framework depends on the following four main factors.

1. *How likely can a phishing mail be trapped so that a phoneytoken can be successfully submitted?*

2. *How likely can a phisher detect a phoneybot and then distinguish a phoneytoken collected from it?*

3. *How likely can a phisher distinguish a phoneytoken from a genuine credential?*

4. *How well can the phishing detector work to detect real phishing account and block a phisher's money laundering process successfully?*

The first factor is not a real problem, since phishers have to depend on a large-scale distribution of phishing emails to lure potential victims. If we agree with the estimation given in [47], the annual victimization rate of phishing attacks is about 0.37% or even less. Such a low rate forces phishers to spread their phishing mails as wide as possible. Therefore, as long as phishers have no effective way to detect spamtraps and phoneybots, a very high capture rate can be expected. In addition, since human managers are in charge of checking suspicious phishing emails, false positive and false negative rates can be effectively reduced.

The second factor is not a real problem, either. As we mentioned before, it is a human user (the phoneybot manager) who submits the phoneytoken to the phishing site as the response to a phishing mail, by mimicking the behavior of a typical victim. Thus, we believe that it is very difficult (if not impossible) for a phisher to distinguish phoneytokens from genuine credentials. While all the virtual machines running phoneybots can be physically hosted on a few number of computers of the financial institutes, the IP address of each virtual machine should look like an IP address of a common home PC. By using DSL or VPN (virtual private network) to connect each virtual machine to a public ISP, this requirement can be easily fulfilled. Furthermore, the phoneybot managers should always access the virtual machines from public ISPs rather than from an IP address of the financial institute to avoid leaving any observable trace to phishers.

The other two factors highly depend on how a phisher interacts with the e-banking system and how the phishing detector works. We discuss them in the next subsection.

E. The Phishing Detector

As we mentioned in Sec. IV-A, Step 7b is the most crucial step in the proposed anti-phishing framework. Only if the phishing detector works well, the financial institute can successfully frustrate phishers' efforts of stealing money from victims.

1) The basic idea

The basic idea of the phishing detector is actually very simple. Since a phoneytoken is simply a fake credential, no legitimate transaction will use a phoneytoken as the sender. In other words, once the phishing detector notices any attempt of transferring some money from a phoneytoken to another bank account, it must be due to a phisher's money laundering effort. Note that some honest customers may accidentally input an incorrect bank account number, so a phoneytoken may be the receiver of a legitimate transaction.

To make things simpler, let us first assume a phisher does not doubt the genuineness of a collected credential. In this case, he will simply try to transfer the maximally allowed amount of money from each bank account under his control to his mules. Then, the phishing detector can immediately mark the receiving account of each fund transfer from a phoneytoken as a suspicious "phishing account". The phishing detector then freezes all future fund transfers sent from any genuine account to a marked phishing account, and seeks reconfirmation from the owner of the affected account. Upon the approval of the real account owner, the stolen bank account can be locked. Note that the reconfirmation will be sought for all incomplete fund transfer from any genuine bank account to each suspicious phishing account, including those sent before the detection of the phishing account. This is possible due to the normal transaction delay.

To lure the phisher to continue using the marked phishing accounts for money laundering so that more victims can be rescued, the e-banking system dynamically creates a new phoneytoken as a shadow of the protected account. In this way, all rescued victims' accounts will be transformed into *shadow phoneytokens*. The phisher is allowed to continue accessing these shadow phoneytokens, and all further transactions made in the e-banking system will be showed and analyzed, but not be executed. Due to the usual transaction delay, the phisher will not be able to notice what is going on until he finds out a previous fund transfer fails several days later. Although the phisher will finally realize at least one of credentials he used in the past is a phoneytoken, it is too late for him to get any of the credentials back. After a phoneytoken is exposed, it will be removed from the honeyed e-banking system.

When the phisher is waiting for the transferred funds to reach his mules' accounts, the financial institute can contact the financial institutes managing the mules' accounts and related authorities to start further investigation. This may help catch the mules and rescue victims whose money was stolen well before the first phishing account is marked. Although it may be impossible to catch the phisher, the potential risk of being suspected and inspected by the financial institutes and related authorities may effectively deter a phisher's motivation to recruit mules and launch new phishing attacks in future.

2) Countering phoneytoken verification

The above simplified analysis assumes that the phisher does not doubt the genuineness of each collected credential. This assumption is, unfortunately, not true in reality. According to

the Kerckhoffs' principle, the phisher knows all the implementation details of the e-banking server, so he may consider verifying the genuineness of each collected credential before stealing money. It is obvious that the phishing detector should tolerate the verification attempts of a phisher. This makes the task of the phishing detector more complicated.

As already discussed in Sec. III-C, the simplest tactic a phisher can use for phoneytoken verification is to send a specific amount of money to an already known genuine account and (after a reasonable waiting time) to check if the verification fund transfer is successful. There are several classes of accounts a phisher may use: his mules' accounts, other victims' accounts under his control, third-party accounts which allow the phisher to check the results of fund transfers. The third class of accounts include bank accounts of charitable organizations, freeware developers, online shops, and so forth.

Since a phisher will not like losing too much money from any genuine bank accounts under his control, it is reasonable to assume that he will only send a small amount of money for the purpose of verification. This means that the phishing detector can define a threshold H and restrain from issuing an alert for all fund transfers below H. While it is difficult to know how large H should be, the phishing detector can adaptively determine its value according to behavioral profiling information collected from phishers and customers. A negative side effect is that the financial institute has to bear the costs incurred from phishers' verification attempts.

If a phisher can successfully recruit a large number of mules, he may be able to steal a victim's money by making a large number of small fund transfers to his mules, each of which is below the threshold H. In this case, the large number of small fund transfers itself becomes a salient feature and can be used to issue an alert. This is because a common user does not tend to have such an extraordinary behavior of online transactions. In addition, since most financial institutes have deployed two-factor user authentication schemes like the TAN systems for online transactions, it is not likely a phisher will be able to make a large number of fund transfers unless he has access to all dynamic TANs.

Since transferring a small amount of money cannot help a phisher tell a phoneytoken from a genuine account apart, he may turn to use a different tactic: to transfer a considerably large amount of money to a genuine bank account. In this case, we believe the phisher will not choose third-parties as the receivers, because there is a high risk of losing too much money from a real victim's account (in case the tested credential is not a phoneytoken). He will not consider his mules' accounts, either, because this actually means the verification process is dropped for the tested credentials. Then, the phisher only has one choice left: other victims' accounts. Note that this choice is also the best case from the banking institute's point of view, because some victims can be rescued as shown below. Since the phisher does not know which credentials are genuine, the only tactic he can use is to test the genuineness of all collected credentials by transferring a considerably large amount of money to each other.

In the above case, the phishing detector has to issue an alert if there is a relatively large fund transfer from a phoneytoken to a non-phoneytoken account, because there is no clue to exclude an attempt of real money laundering. After issuing an alert, the financial institute tries to contact the receiver of the fund transfer to see if it is a victim or a mule recruited by a phisher. If the receiver is a victim, she can be rescued and a shadow phoneytoken is created to fool the phisher as usual. Note that the rescue of one victim may lead to rescue of another victim, if the first victim's account is tested by a fund transfer sent to the second victim's account. Such an interesting "chain reaction" may cause rescue of even more victims. If the receiver is a mule, he will be warned about the phishing scam and his cooperation is sought. We expect in most cases the mule will cooperate, since he is often an innocent person who believes the phisher is offering a real job. The financial institute may also force the mule to cooperate via related authorities if he turns out to be an accomplice. If the receiver's account belongs to another financial institute, cooperation between the two financial institutes is a must.

For a tested phoneytoken, if the financial institute fails to reach the receiver or to get necessary cooperation, the phoneytoken will be detected by the phisher after the transaction delay. Fortunately, even in this case, some phoneytokens may still remain undetected. Assume that the phisher has collected N_g genuine accounts and N_p phoneytokens, and that for each tested account he randomly picks another account as the receiving account. Then, the detection probability of each phoneytoken is $P(1) = p \cdot N_g/(N_g + N_p - 1)$, where p is the probability that the financial institute fails to reach the owner of a genuine account or get cooperation. The phisher may increase this probability by making k different verification transactions for each tested account. If the availability of the owner of each receiving account is independent of each other, the phisher may be able to detect each phoneytoken with probability $P(k) = \sum_{i=0}^{\min(k,N_p-1)}(1-(1-p)^{k-i})\binom{N_p-1}{i}\binom{N_g}{k-i}/\binom{N_g+N_p-1}{k}$. The value of $P(k)$ depends on N_g, N_p and p. The financial institute has no control on N_g and p, but it can increase the value of N_p by deploying more phoneybots/spamtraps. It is obvious that $P(k)$ decreases as N_p increases. When $N_p = N_g + 1$, we can verify $P(k) \leq P(1) \leq p/2 \leq 1/2$. The number of deployed phoneybots/spamtraps $\hat{N}_p$ can be estimated from the victimization rate r of each phishing attack targeting the financial institute. Assuming the total number of customers is N, then a proper choice of $\hat{N}_p$ will be $\hat{N}_p = N \cdot r + 1$. Without loss of generality, assume that the average number of phishing attacks targeting the financial institute per year is n and the annual victimization rate of all the n phishing attacks is R, then $\hat{N}_p = N \cdot R/n + 1$. According to a recent phishing trends report [48], the total number of unique phishing URLs per targeted brand is 563 for the second half of 2008. We assume that each unique phishing URL corresponds to a unique phishing attack. To have a reasonable estimation of n, we set $n = 2 \times 500 = 1000$. We further assume $R = 0.01$ for a typical financial institute. Then, we can

get $\hat{N}_p = N \cdot 10^{-5} + 1$. This means that the financial institute only needs to deploy one phoneybot/spamtrap for for every 100,000 customers plus one more for all customers, in order to ensure each phoneytoken will be detected with probability less than 1/2. Taking Germany's largest bank – Deutsche Bank – as an example, in Year 2007 it had 13,800,000 customers [49], which corresponds to 139 phoneybots/spamtraps. This is apparently a manageable scale. Note that $P(k)$ is the probability of each phoneytoken is detected by a phisher. If we consider all the N_p phoneytokens, the detection probability will be $\leq 1/2^{N_p}$, which is negligible if N_p is not very small.

3) Practicality of phoneytoken verification

In the above analysis we assume a phisher will verify the genuineness of each credential before laundering the money. In a real phishing attack, this assumption may not be true. The main concern of a phisher is about the relatively short lifespan of a phishing attack. On one hand, the average lifespan of a phishing site is only several days [16], [44]. On the other hand, even if a phishing site can survive a long term, as long as a financial institute can detect a phishing attack in time, all of its customers can be immediately reminded about this fact and then the victims will take action to lock their accounts. In addition, a victim may by accident notice the phishing scam. All these facts imply that a phisher may not have enough time to perform the verification process. The average processing time of an electronic fund transfer is normally 2 or 3 banking days. This means a phisher needs 4 to 6 banking days to verify a credential and confirm the later money laundering is successful. Since there are only 5 banking days each week in most countries, the waiting time is often longer than one week. Considering most phishing sites cannot survive more than one week, a phisher may not be willing to take the risk to verify each collected credential.

Some financial institutes do offer real-time fund transfer services, but only for fund transfers between accounts belonging to the same financial institute. From a phisher's point of view, this is not a good situation, since the whole process of fund transfer is under the surveillance of the targeted financial institute. However, the convenience of quickly verifying the genuineness of collected credentials may encourage phishers to target these banks more often in future. The real-time fund transfer service can also dramatically influence the performance of the phishing detector, since in this case the phishing detector will become merely a detector of mules' accounts, and further rescue of more victims will not be possible. Due to this concern, we argue that the real-time fund transfer service should not be offered to customers if we want to better fight against phishers. This is obvious since without a reasonable time period we cannot do anything against phishers. Or we can say "phishers love faster banks" – a very interesting phenomenon that deserves more research in future.

4) Confusing the phishing detector

In addition to the tactics we discussed above, a phisher may use another tactic to confuse the phishing detector: transferring a considerable amount of money (above the threshold H) to a number of randomly selected bank accounts NOT under his control. This will lead to false positives of the phishing detector and cause investigation on innocent customers. Apparently, a phisher's mules may argue they are innocent customers, too. However, it is doubtful if the phisher is willing to distribute a large amount of stolen money to an unknown person. Even if this happens, the financial institute will still be able to get the stolen money back, since both the innocent users and the phisher's mules have to admit that the incoming money does not belong to them. Since a phisher can not get direct benefit from this tactic, we assume that this tactic will not cause too much trouble to our proposed anti-phishing framework.

5) Summary

To sum up, the phishing detector sets a threshold H and monitors fund transfers from all phoneytokens to non-phoneytoken accounts. An alert will be issued for any fund transfer above H, and the receiver's account will be marked as a suspicious "phishing account". Depending on the tactic a phisher may use, there will be both false negative (missing phishers or their mules) and false positive (annoying innocent customers) alerts. How to reduce both false negative and false positive rates is an open topic for future research on the anti-phishing framework.

V. Conclusions

In this paper, we have identified some problems with existing anti-phishing solutions based on honeypots and propose a new anti-phishing framework to overcome these problems. This framework cannot prevent phishing in itself. Rather, it is designed to hit phishers' financial motivation and to make phishing more risky and less beneficial, thereby discouraging phishers from launching new phishing attacks. We believe our proposal opens a new way to a probably ultimate solution to the problem of phishing. To support this claim, we would like to quote the following texts extracted from [2, Chapter 5]:

> "*There's one really good way to stop phishing and identity theft. It's hard, though: make it unprofitable. Criminals don't generally waste their time with unprofitable scams. At least the smart ones don't; and those are the ones you really have to worry about.*
> *Cutting the connection between stolen identities and cash – whether by making identities harder to steal or by making money harder to obtain – will be difficult and time-consuming. However, it's the only way to really stop phishing.*"

Although our proposed anti-phishing framework is mainly focused on e-banking systems, the basic idea could be generalized to other potential targets of phishing attacks such as e-payment platforms.

In the future, we plan to develop a prototype system to demonstrate our idea and test the effectiveness of the proposed anti-phishing framework. The prototype system will be a virtual e-banking system that allows users to simulate

most common e-banking practices. Virtual phishers and virtual victims will be added to test how to tailor the phishing detector to get an optimized performance. The prototype system can also be considered as a serious game for the purpose of user education against phishing. We are also in the process of contacting some financial institutes for possible cooperation.

Acknowledgments

Shujun Li was supported by a fellowship from the Zukunftskolleg of the University of Konstanz, Germany, which is part of the "Excellence Initiative" Program of the German Research Foundation (DFG). Both authors would like to thank Prof. Walter Kriha of the Stuttgart Media University for valuable discussions and for commenting on an early draft of the paper.

Appendix: Enhancing PIN/TAN System

Almost all e-banking systems in Germany are using PIN/TAN as the two-factor authentication method. Generally, a customer has to input a PIN for login and a TAN for any online transaction made in the e-banking system. There are mainly two forms of TANs: iTANs (indexed TAN) [50] and mTANs (mobile TAN) [20]. The iTANs are a list of n randomly-generated TANs indexed by the numbers 1 to n, which are sent to each customer in paper form in advance. The mTANs are TANs generated in a real-time manner for online transactions, and sent to each customer's registered mobile phone via SMS. There are also TANs generated by hardware tokens [51], [52].

In this appendix, we consider how the iTAN system can be enhanced to better work with the anti-phishing framework proposed in this paper. We do not intend to argue iTAN is better than mTAN[2], but just want to show that, even with quite a simple TAN system like iTAN, the performance of our proposed anti-phishing framework can be easily enhanced.

In Sec. III, we have shown that a phisher may try to verify the genuineness of each credential. Without the protection of TANs, a phisher can make as many online transactions as he likes for the purpose of verification. However, in case a TAN has to be input for each transaction, the phisher may have difficulties making successful transactions.

It has been known that the iTAN is not secure against man-in-the-middle attack [53], [54], so the phisher may depend on the user to input the PINs required for the online transactions. The dependence on the user means that the phisher has no freedom to perform credential verification and money laundering at any time he wants. In case the user notices one previous online transaction failed, she will be able to detect the man-in-the-middle attack, and then fix her computer accordingly to remove the source of future man-in-the-middle attacks (i.e., poisoned DNS entry or phishing malware). When the phisher wrongly verify a phoneytoken as a genuine credential, the user may also be informed by the financial institute about the man-in-the-middle attack as we discussed in Sec. IV-B.

The above analysis shows that the man-in-the-middle attack becomes less powerful due to the combination of iTAN and the honeypot-based anti-phishing framework. As a result, instead of launching man-in-the-middle attacks, a better strategy of the phisher will be to lure some iTANs from the user and then perform the credential verification/money laundering attempts freely. In this case, it is obvious that the anti-phishing performance depends on how many unused iTANs a phisher can get from a victim and how many successful online transactions the phisher can make with these iTANs. In the following, we make a rough probabilistic analysis on this strategy.

Although many financial institutes clearly remind their customers that NO TANs will be asked for on the login page of their e-banking systems, many customers simply do not pay any attention to such notices. This makes it possible for a phisher to ask for a reasonable number of iTANs (see Fig. 8.4 of [1] for an example). Let us assume that the total number of unused iTANs is $n' \leq n$ and a phisher has successfully got $1 \leq k \leq n$ iTANs from a victim together with the account number and the PIN. Note that the phisher may get some used iTANs, since he has no information about the indices of previously used TANs. Assuming the number of unused iTANs the phisher has got is $0 \leq k' \leq k$, then we know that averagely $k' = k \cdot n'/n$. Then, the probability that the phisher passes the first online transaction he tries in the e-banking system will be $k'/n' = k/n$. Since asking for too many iTANs will definitely raise suspicion of the potential victim, we believe a proper value of k cannot be greater than 10. When $n = 100$, the phisher will pass the next transaction with probability ≤ 0.1. Considering most e-banking systems allow the user to make 3 consecutive errors, the probability that the phisher can make the first online transaction is $1-(1-k/100)^3 \leq 1-0.9^3 = 0.271$. While this is not a very small probability, keep in mind that the phisher has to verify each credential and then transfer the money to his mules. This means at least two successful online transactions are necessary, and the second one is more important from the phisher's point of view. Therefore, the probability that a phisher can successfully verify a credential and then steal the money is not greater than $0.271^2 \approx 0.07$. Recalling the example we discussed for the Deutsche Bank in Sec. IV-E, this probability means that, averagely, in each phishing attack, the phisher can only steal money from $138 \times 0.07 < 10$ victims. Due to the anti-phishing function offered by the phishing detector, the success rate of the phisher will be further reduced. A direct consequence of the above analysis is: most phishers will simply skip the verification step and always try to send the money in all stolen accounts to his mules. This is very good for our anti-phishing framework, since the phishing detector works perfectly if a phisher does not verify genuineness of collected credentials.

The above analysis shows the simple iTAN system has already offered an acceptable performance against phishers. By slightly modifying the existing iTAN system, we can

[2]In fact mTAN can resist advanced phishing attacks based on pharming and malware, because transaction authentication is also included in the SMS sent by the the financial institute to the customer.

do even better. In the following, we propose two possible modifications to the existing iTAN system and show they can enhance the system's anti-phishing performance. Note that the following two modifications enhance the current iTAN system in two different ways: the first modification focuses on reducing the probability a phisher gets unused iTANs, and the second one focuses on increasing the phisher's difficulties of making successful online transactions. They can be combined to provide even better performance.

Advanced iTAN

Most existing implementations of iTAN systems in Germany are using 1 to 100 as the indices of the 100 iTANs on the TAN list. This makes it possible for a phisher to ask a potential victim to expose a number of iTANs and then use the obtained iTANs to manipulate the victim's account. If we change the indices of the 100 iTANs to something a phisher does not know, he has to guess them randomly, which will lead to a much lower success rate even at the stage of luring iTANs. In addition, the potential victim may immediately recognize the ongoing phishing scam once she notices an invalid index on a phishing site. We believe most users will definitely be able to notice *all* invalid indices, because they have to look for the corresponding iTANs in the iTAN list by comparing the indices with those given by the phisher.

Let us consider a simple example. Instead of using $\{1, \ldots, 100\}$ as the indices, we may turn to use 100 integers randomly selected from a set $\{1, \ldots, 1000000\}$. Then, if a phisher wants to lure k iTANs from a potential victim, he has to show the potential victim k valid indices. Since the phisher has no prior information about the valid indices, he can only randomly pick k indices from $\{1, \ldots, 1000000\}$. Then, the probability that all the k indices are valid will be

$$\begin{aligned} \frac{\binom{100}{k}}{\binom{1000000}{k}} &= \frac{100 \cdot 99 \cdots (100-k+1)}{1000000 \cdot 999999 \cdots (1000000-k+1)} \\ &< \left(\frac{100}{1000000}\right)^k = 10^{-4k}. \end{aligned}$$

This means that the phisher has a very tiny success rate to spoof the potential victim even when he asks for only one iTAN. In other words, the potential victim has a very high probability to recognize she is interacting with a phishing site rather than the real e-banking web site. Now the victim herself becomes a very reliable and powerful tool against phishers, which sharply contrasts with the unreliability of human users in many other anti-phishing measures designed at the user interaction level. To further reduce the success rate of a phishing attack, we may use even more complicated index like a word composed of English characters and digits.

Double-TAN Confirmation

Instead of asking for only one iTAN for each online transaction, we can ask the user to input two iTANs. All other operations such as login and changing password ask for no or only one iTAN. This change only slightly decreases the usability of the iTAN system, but will dramatically reduce the success rate of a phishing attack: when a phisher tries to do the verification, the success rate of the phishing attack will become $0.07^2 = 0.0049$, and if he chooses not to do the verification, the success rate will be 0.07.

References

[1] M. Jakobsson and S. Myers, Eds., *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. John Wiley & Sons, Inc., 2007.

[2] R. Lininger and R. D. Vines, *Phishing: Cutting the Identity Theft Line*. Wiley Publishing, Inc., 2005.

[3] L. James, *Phishing Exposed*. Syngress Publishing, Inc., 2005.

[4] D. Hartley, "Search engines disguise phishing sites," 2009. [Online]. Available: http://www.itexaminer.com/search-engines-disguise-phishing-sites.aspx

[5] Imperva, Inc., "Securesphere defense note: Phishing and cross-site scripting," 2004. [Online]. Available: http://www.securitytechnet.com/resource/rsc-center2/vendor-wp/imperva/20050606_Imperva_SecureSphere_Defense_Note-Phishing.pdf

[6] Nexus, "Applying XSS to phishing attacks," 2007. [Online]. Available: http://www.playhack.net/view.php?type=1&id=20

[7] J. Yaneza, "Spy-phishing: A new breed of blended threats," in *Proc. Virus Bulletin Conference 2006*, 2006. [Online]. Available: http://www.trendmicro.com/NR/rdonlyres/AC48648F-50D0-49F5-8438-7AEE516C501E/21322/spyphishing_102006.pdf

[8] G. Ollmann, "The pharming guide: Understanding & preventing DNS-related attacks by phishers," 2005. [Online]. Available: http://www.ngssoftware.com/papers/ThePharmingGuide.pdf

[9] S. Stamm, Z. Ramzan, and M. Jakobsson, "Drive-by pharming," in *Information and Communications Security (Proc. ICICS'2007)*, ser. Lecture Notes in Computer Science, vol. 4861. Springer, 2008, pp. 495–506.

[10] US Office of the Comptroller of the Currency, "Money laundering: A banker's guide to avoiding problems," 2002. [Online]. Available: http://www.comptrollerofthecurrency.gov/moneylaundering2002.pdf

[11] M. Jakobsson and S. Myers, "Delayed password disclosure," *ACM SIGACT News*, vol. 38, no. 3, pp. 56–75, 2007.

[12] M. Wu, S. Garfinkel, and R. Miller, "Users are not dependable: How to make security indicators that protect them better," Presented at 1st Workshop on Trustworthy Interfaces for Passwords and Personal Information (TIPPI'2005), 2005. [Online]. Available: http://crypto.stanford.edu/TIPPI/first/slides/min_wu.ppt

[13] M. Wu, R. C. Miller, and S. L. Garfinkel, "Do security toolbars actually prevent phishing attacks?" in *Proc. CHI'2006*. ACM, 2006, pp. 601–610.

[14] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proc. CHI'2006*. ACM, 2006, pp. 581–590.

[15] D. Florêncio and C. Herley, "Password rescue: A new approach to phishing prevention," in *Proc. USENIX HotSec'2006*, 2006, pp. 7–11.

[16] T. Moore and R. Clayton, "The consequence of non-cooperation in the fight against phishing," in *Proc. APWG eCRS'2008*. IEEE Computer Society, 2008.

[17] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell, "Stronger password authentication using browser extensions," in *Proc. USENIX Security'2005*, 2005, pp. 17–31.

[18] IBM, "IBM Zone Trusted Information Channel (ZTIC): A banking server's display on your key chain," 2008. [Online]. Available: http://www.zurich.ibm.com/ztic

[19] G. Starnberger, L. Froihofer, and K. M. Goeschka, "QR-TAN:

Secure mobile transaction authentication," in *Proc. ARES'2009*. IEEE Computer Society, 2009, pp. 578–583.

[20] Postbank, "mTAN now free for all customers," 2008. [Online]. Available: http://www.postbank.com/pbcom_ag_home/pbcom_pr_press/pbcom_pr_press_archives/pbcom_pr_press_archives_2008/pbcom_pr_pm1063_19_05_08.html

[21] Irish Times, "Suspended term for €12,000 bank 'phishing' scam," 2009. [Online]. Available: http://www.irishtimes.com/newspaper/ireland/2009/0214/1233867937480.html

[22] R. L. B. Stevenson, "Plugging the phishing hole: Legislation versus technology," *Duke Law & Technology Review*, vol. 2005, no. 0006, 2005. [Online]. Available: http://www.law.duke.edu/journals/dltr/articles/2005dltr0006.html

[23] D. Whitlock, "INTERNET FRAUD: Preventing and responding to phishing and spoofing scams," *New Hampshire Bar J.*, vol. 42, no. 2, pp. 30–33, 2008.

[24] L. Spitzner, *Honeypots: Tracking Hackers*. Addison Wesley, 2002.

[25] F. Pouget, M. Dacier, and H. Debar, "Honeypot, honeynet, honeytoken: Terminological issues," Institut Eurécom (EURECOM), Sophia Antipolis, France, Research Report RR-03-081, 2003. [Online]. Available: http://www.eurecom.fr/util/publidownload.fr.htm?id=1275

[26] M. Chandrasekaran, R. Chinchani, and S. Upadhyaya, "PHONEY: Mimicking user response to detect phishing attacks," in *Proc. WoWMoM'2006*. IEEE Computer Society, 2006, pp. 668–672.

[27] C. M. McRae and R. B. Vaughn, "Phighting the phisher: Using web bugs and honeytokens to investigate the source of phishing attacks," in *Proc. HICSS'2007*. IEEE Computer Society, 2007, p. 270c.

[28] C. Yue and H. Wang, "Anti-phishing in offense and defense," in *Proc. ACSAC'2008*. IEEE Computer Society, 2008, pp. 345–354.

[29] D. Birk, S. Gajek, F. Grobert, and A.-R. Sadeghi, "Phishing phishers—observing and tracing organized cybercrime," in *Proc. ICIMP'2007*. IEEE Computer Society, 2007.

[30] S. Gajek and A. Sadeghi, "A forensic framework for tracing phishers," in *The Future of Identity in the Information Society*, ser. IFIP International Federation for Information Processing, vol. 262. Springer, 2008, pp. 23–35.

[31] G. Roth, "Server load balancing architectures, Part 1: Transport-level load balancing," 2008. [Online]. Available: http://www.javaworld.com/javaworld/jw-10-2008/jw-10-load-balancing-1.html

[32] ——, "Server load balancing architectures, Part 2: Application-level load balancing," 2008. [Online]. Available: http://www.javaworld.com/javaworld/jw-10-2008/jw-10-load-balancing-2.html

[33] C. Herley and D. Florêncio, "Protecting financial institutions from brute-force attacks," in *Proc. IFIP SEC'2008*, ser. IFIP International Federation for Information Processing, vol. 278. Springer, 2008, pp. 681–685.

[34] RSA Security Inc., "RSA® FraudAction℠: Advanced external threats protection service," 2008. [Online]. Available: http://www.rsa.com/products/consumer/datasheets/9933_FRAGOV_DS_1208.pdf

[35] MarkMonitor Inc., "Rock phishing: The threat and recommended countermeasures," 2007. [Online]. Available: http://www.markmonitor.com/download/wp/wp-rockphish.pdf

[36] R. M. Smith, "The web bug FAQ," 1999. [Online]. Available: http://w2.eff.org/Privacy/Marketing/web_bug.html

[37] L. von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," *Comm. ACM*, vol. 47, no. 2, pp. 57–60, 2004.

[38] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA," in *Proc. CVPR'2003*, vol. 1. IEEE Computer Society, 2003, pp. 134–141.

[39] J. Yan and A. S. E. Ahmad, "Breaking visual CAPTCHAs with naïve pattern recognition algorithms," in *Proc. ACSAC'2007*. IEEE Computer Society, 2007, pp. 279–291.

[40] ——, "A low-cost attack on a Microsoft CAPTCHA," in *Proc. ACM CCS'2008*, 2008, pp. 543–554.

[41] S. Hocevar, "PWNtcha: Pretend we're not a Turing computer but a human antagonist." [Online]. Available: http://caca.zoy.org/wiki/PWNtcha

[42] C. Chesnut, "aiCaptcha: Using AI to beat CAPTCHA and post comment spam," 2005. [Online]. Available: http://www.brains-n-brawn.com/aiCaptcha

[43] BBC News, "PC stripper helps spam to spread," 2007. [Online]. Available: http://news.bbc.co.uk/2/hi/technology/7067962.stm

[44] T. Moore and R. Clayton, "Examining the impact of website take-down on phishing," in *Proc. APWG eCRS'2007*. ACM, 2007, pp. 1–13.

[45] V. L. Caruso, "Outsourcing information technology and the insider threat," AFIT/GIR/ENG/03-01, Department of Electrical and Computer Engineering, Graduate School of Engineering and Management, Air Force Institute of Technology, Air University, USA, 2003. [Online]. Available: http://handle.dtic.mil/100.2/ADA415113

[46] IT-Online, "World-first SMS banking scam exposes weaknesses," Web page, July 2009. [Online]. Available: http://www.it-online.co.za/content/view/1092105/142/

[47] C. Herley and D. Florêncio, "A profitless endeavor: Phishing as tragedy of the commons," in *Proc. NSPW'2008*. ACM, 2008, pp. 59–70.

[48] Anti-Phishing Working Group, "Phishing activity trends report, 2nd half / 2008," 2009. [Online]. Available: http://www.antiphishing.org/reports/apwg_report_H2_2008.pdf

[49] Deutsche Bank, "Deutsche Bank at a glance," 2008. [Online]. Available: http://www.db.com/presse/en/download/DB_at_a_glance_2008.pdf

[50] Sparkasse Heidelberg, "Online-banking: Increased security with iTAN," 2006. [Online]. Available: http://www.sparkasse-heidelberg.com/spk-hd_en/pk/banking/itan.html

[51] Berliner Sparkasse, "chipTAN: Listen werden überflüssig," 2005. [Online]. Available: http://www.berliner-sparkasse.de/anzeigen.php?tpl=privatkunden/konten_karten/online_banking/tan_generator.html

[52] Volksbank Solling eG, "Sm@rt-TAN-plus," 2009. [Online]. Available: http://www.volksbank-solling.de/flycms/de/html/913/-/Smart+TAN+plus.html

[53] RedTeam Pentesting GmbH, "New banking security system iTAN not as secure as claimed," Advisory rt-sa-2005-014, 2005. [Online]. Available: http://www.redteam-pentesting.de/advisories/rt-sa-2005-014

[54] Arbeitsgruppe Identitätsschutz im Internet, "A-I3 Pressemeldung: iTAN nur in Verbindung mit SSL sicher (Update)," 2005. [Online]. Available: https://www.a-i3.org/content/view/411/28