Alice, what did you do last time? Fighting Phishing Using Past Activity Tests

Nikos Nikiforakis, Andreas Makridakis
Elias Athanasopoulos, and Evangelos P. Markatos

Institute of Computer Science (ICS)
Foundation for Research & Technology Hellas (FORTH)
{nikifor,amakrid,elathan,markatos}@ics.forth.gr

Abstract

Phishing attacks are one of the most crucial modern security threats in the current World Wide Web. An adversary may clone a legitimate Web site and lure a user to submit her credentials to the malicious construct. The adversary may then use the stolen credentials to the authentic site. In this paper we present a novel idea to fight phishing using Past Activity Tests (PACTs). In a nutshell, PACTs take advantage of the fact that the user has accessed at least once her account in the past, contrary to the phisher who accesses the user's account for the first time. Thus, a user can answer a question relative to her past activity, but the attacker can not.

Keywords: Web Security, Phishing

1 Introduction

Phishing attacks exploit social engineering techniques in order to lure users and convince them to give their account information, such as their login and their password, in a fraud web site. A phishing attack is held by building a web site identical of the authentic one. This process does not require cloning the legitimate web site in whole, but only the pages involved in the login process of a user, in order to give the impression to the victim that she is visiting the authentic web site and not the cloned one.

Phishing attacks are considered as a major security threat in modern Internet. In Phishing Attack Trends Report, published by Anti-Phishing Working Group (APWG) [3] in April 2007, it was reported that the number of unique phishing web sites detected by APWG rose to 55,643 in April 2007. A massive jump of nearly 35,000 from March was detected, resulting from aggressive subdomain phishing tactics by which phishers started using the tactic of putting a large number of phishing URLs on the same domain. A large number of banks were targeted in April with seven of the most-targeted 20 brands in that month belonging to European banks. In addition, one of the top 20 was a Canadian financial institution.

Phishing is usually connected with the cloning of sites that are active in e-commerce, like bank services or on-line stores. However, a phishing attack may target sites that are not involved directly with money transactions. These sites may still embed private social information which is sensitive and must not be leaked to third parties. Some examples are, blog services, e-mail services or content hosting services, like Flickr.com[1]. These sites are expected to be phishing targets with a lower probability than e-commerce sites, which may not be willing to invest a large amount of money for an anti-phishing approach that requires an extensive effort and modifications to the site's infrastructure.

In this paper, we present a novel anti-phishing approach, which:

- is easily deployable in the server side of a web site,
- does not need client-side modifications to a web browser,
- is low cost,
- is user friendly.

Our anti-phishing solution adds an extra authenticating step between the username/password form and the complete authentication of the user. It is based on the fact that *the phisher is accessing the victim's account for the first time in contrast with the legitimate user, who has accessed her account in the past.* The rest of this paper is organized as follows. In section 2 we present our anti-phishing approach by introducing for the first time Past Activity Tests (PACTs). In section 3 we evaluate theoritically the PACT approach and we highlight PACTs' limitations. In section 4 we present two real deployments. We present related work in section 5 and we conclude in section 6.

2 PACT architecture

This section highlights the basic components of an anti-phishing architecture, which is based on Past Activity Tests (PACTs). We assume Alice is a registered user to an on-line service and Trudy an adversary, who tries to steal Alice's credentials using phishing.

2.1 PACT Definition

We define a PACT as follows:

> **PACT definition**: *A PACT is a dynamic test with N possible answers but only one solution. The solution is correlated with Alice's past activity of her account. The correct solution can only be given by the user with probability 1, or by Trudy with probability* $\frac{1}{N}$.

Following directly from PACT definition, the idea we are building over our anti-phishing solution, relies on the following observation:

[1] A popular site, which hosts a user's pictures in private or public areas

Alice has visited her account at least once in the past. On the contrast, Trudy has access in Alice's active account for the first time.

Using PACT we ask Alice to give an answer to an obvious question, which is based on her past activity to her account. This answer can only be guessed by Trudy, since she is not the real owner of the account and thus, has no knowledge of its past activity.

2.2 Example PACTs

In order to deploy PACT in a real on-line service, the service must create dynamic puzzles based on its subscribers profiles. These puzzles must be solved upon a user is authenticating herself to the on-line service. In order to build a PACT a service must have the following information:

- A summary of Alice's past actions.
- A pool of abstract actions to the service performed by a speculative subscriber.

For example in an e-mail service, Alice may be asked to choose from a pool of e-mail addresses, an e-mail address with which she had contact in the near past. In Table 1 we list a series of PACT examples.

Service	PACT
E-mail	Select an e-mail address you had contact with in the near past.
E-commerce	Select an item you have purchased in the past.
Content host	Select a picture you have uploaded in the near past.[2]
Instant Messaging	Select a user, who is in your contact list.[3]

Table 1. Example of PACTs for different possible on-line services.

3 PACT evaluation

In this section we present a theoretical evaluation of PACTs and highlight their limitations.

[2] The majority of popular content host providers allow you to maintain a private area with your content. We assume that PACT's solution is taken from Alice's private area.

[3] This is a case where PACT is used to an on-line service that is not necessarily hosted in the World Wide Web.

3.1 PACT Resistance

Following directly from PACT definition, the probability of solving a PACT is $p_s = \frac{1}{N}$, where N denotes the number of possible answers. An interesting property of PACTs is that they can be combined so as the probability p_s to degrade exponentially. For example, by combining m PACTs we have $p_s = \frac{1}{N^m}$. Assuming that the on-line server suspends an account after a false attempt, a brute force approach for by-passing a PACT is considered unrealistic. Between the exploiting of an on-line account, by malicious users, and the temporary suspension of it, the latter is preferable.

3.2 PACT Suspension Policy

PACTs can adapt their suspension policy, according to the type of service provided and the available amount of data from the user.

In an e-mail service a user has probably sent more than one e-mails in the near past. So, if she fails to answer the first PACT puzzle, she is provided a new puzzle with a different correct answer. Thus, the account will be suspended after two unsuccessful attempts.

On the contrast, in a bidding service a user will more likely find the correct answer to the first PACT puzzle, since the question concerns an item bought by her in the past. Thus, the account can be suspended after the first unsuccessful attempt.

3.3 PACT Limitations

PACT can resist to a phishing attack since the probability for an adversary for solving a PACT is quite low. However, PACT can not resist to a Man-in-the-Middle attack. If Trudy can set up a Phishing Proxy between Alice and the on-line service, then Trudy, upon reaching the PACT, can redirect the test to Alice and get the solution.

Although PACT can not resist to a MiM attack, it actually prohibits the attacker of creating a collection of stolen accounts that she can later use and/or distribute. Even if an attacker manages to authenticate a session of a user, she can only use it at that time, until the session expires (e.g. bank sessions expire within hours). When the session expires, a new PACT puzzle will be introduced and the attacker will not be able to solve it. So mass-phishing is doable but cumbersome.

Fighting the Phishing Proxy threat model is beyond the scope of the PACT solution. However, there are other countermeasures for malicious Proxies. The most widely adopted is the use of Cryptographic Certificates. It is assumed, that the malicious Proxy can not have access to the authentic server's certificate. Also, it is not easy for an attacker to register a security certificate that obviously tries to impersonate a valid well-known company (e.g. citybank vs citybanc). Then, it is a matter of the adversary to convince the user to accept the forged certificate.

We also acknowledge that an implementation of PACTs in an e-mail service may pose a minor privacy issue, because one out of N presented e-mail addresses will be a valid contact. But without PACTs the attacker already has full access to the victim's mailbox. So, though we may not completely solve the privacy problem, we are increasing the overall security of the service provided.

4 Case Studies

In this section we present two real deployments of on-line services, which use PACT as an anti-phishing countermeasure.

4.1 A PACT enabled e-mail service

This case study refers to a secure way of management an e-mail account. Our goal is to prevent Trudy from accessing Alice's e-mail account. When Alice authenticates herself to the e-mail service, then the service requires from her to answer the following PACT: "Please choose an e-mail address, in which you have sent an e-mail in the near past". A list of ten e-mail addresses is presented and Alice must choose the valid one.

In the left frame of Figure 1 we present a screenshot of the e-mail service, which has been developed using PHP [13]. In order to store the information needed for the dynamic creation of PACTs, we used a PostgreSQL DataBase [14] (version 8.1.8).

Our implementation scheme consists of three tables. The first one, named "*users*", has four fields: *userid, username, password* and *suspended*. Every tuple of this table correspond to a specific e-mail user. The second table, named "*emails*", has three fields: *id, address* and *valid*. The tuples of this table represent valid and invalid e-mail addresses. The last table, named "*emailsFromUsers*", has two fields: *user_id* and *email_id*. It contains both valid and invalid (fake) e-mails which will be used by PACT.

In order to test the service we collected 900 e-mail addresses from Yahoo public lists [2], using a script written in Python. 85,3% of these addresses correspond to yahoo domain. For the purposes of our experiment, we assumed that 800 of the above e-mails are invalid and the rest 100 are valid. All these e-mail addresses were inserted, as tuples, in table "emails". The "valid" field is false for the 800 tuples, which represent invalid e-mails and true for the rest 100 valid e-mail addresses.

Alternatively, construction of invalid e-mail addresses can be used instead of harvesting them off the Internet. According to the results of a quick experiment, 8 out of 10 real e-mail addresses do not produce results in Google [8]. So, a malicious user can not easily find out the correct answer of a PACT puzzle in time.

In the initial page of the e-mail service, Alice must insert into the login form the correct username and password. In case of valid input, a list of ten e-mail addresses is presented. Nine addresses are randomly selected tuples from table

e-mails, where field "valid" is false, and the other address is randomly selected from the same table, where field "valid" is true. If Alice selects the valid e-mail address, a new list of ten addresses will be presented. The new list is created with the same mechanism, as mentioned before. Each time, the ids of these 10 addresses, along with the proper user id, are inserted in table "emailsFromUsers". All the previous tuples of this table, that correspond to the user trying to login, are deleted. Since Alice must choose twice a valid e-mail from a list of ten e-mails, the probability Trudy to efficiently log in Alice's e-mail account is $\frac{1}{10} * \frac{1}{10}$, namely $\frac{1}{100}$.

In the case where Alice inserts into the login form her valid username and password, her account is temporally suspended. This is achieved via the assignment of value "true" in the field "suspended" of table "users", in the tuple that corresponds to Alice. Thus, Trudy can not observe some of the puzzles in order to infer the list of valid e-mail addresses. Alice's account becomes unsuspended, exclusively if she answers correct both questions. However there are possible situations where Alice:

1. fails to answer one of the two questions, selecting an invalid e-mail address
2. answers any of the two questions, but in a time interval longer than 60 seconds
3. never answers the puzzle
4. visits another webpage and then forwards again on the webpage of the questions.

In all the above cases, Alice's account remains suspended and she must contact the site's administrator, in order to unsuspend her account.

For administration purposes, the e-mail service keeps log files, where information about visiting users is appended. Particularly, information is appended in the following cases:

- the login form is made out with invalid username and/or invalid password
- the login form is made out with valid username and valid password, so that user's account becomes temporally suspended
- a user tries to log in a suspended account
- a user answers correct each of the two questions, so that her account becomes unsuspended
- a user fails to answer one of the two questions
- a user answers one of the two questions in a time interval longer than 60 seconds.

This idea could be implemented in real webmail services, such as Hotmail [9] and Google mail [7]. In a real implementation, the list of valid e-mail addresses can be extracted from sent-mail list. The set of invalid e-mail addresses can be constructed using an algorithm that produces plausible e-mail addresses.

4.2 A PACT enabled e-commerce service

For our second case study, we chose to implement our new security measure on a bidding site.

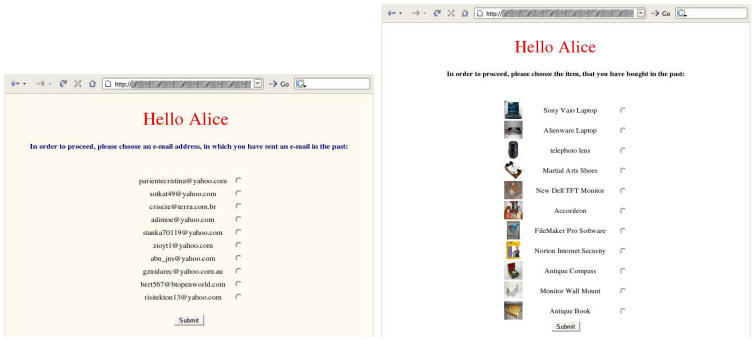

Fig. 1. Screenshots from the two prototype PACT enabled services deployed.

In this case study we wanted to prevent the misuse of a bidding account, even if a prior phishing attack was successful, and the malicious user had obtained a valid username/password pair for entrance in the authentic site.

The PACT used in this case study is: "Select which item(s) you have bought in the past". The list of items, from which Alice has to pick the correct one, is dynamic, meaning that both the items and their position on the list, change every time Alice's logs in. The knowledge of an item bought at the past, is one that, theoretically, only Alice possesses. For the users that have not yet bought an item, we supplied them with a random "dummy purchase" at the time of their registration, so that they too, can be protected by our extra security measure. If the user trying to login, chooses the wrong item, or makes more than 60 seconds to decide, the account is suspended. Our core for the experiment is pretty much the same PHP pages and SQL tables used in our first case study of the e-mail service. In the right frame of Figure 1 we present a screenshot of our bidding site.

4.3 Results

In this section, we present the experimental evaluation of our prototype implementation of Past Activity Tests (PACTs). We created two fake accounts, in Flickr [5] and in Google mail [7]. Also, two accounts were created, for our two case studies (the e-mail service and the bidding service). All of the above accounts correspond to a specific user, thus the username and password are the same. The details of these accounts were stored in a text file, which was shared in the Gnutella P2P network [1]. The purpose of this action was to invite attention of prospective attackers, in order to test our services' implementation. Respecting to the e-mail service, two specific users tried to log in, but in the first question, an invalid e-mail address was selected. In the bidding service, the same users tried to log in plus another one. The first user successfully logged in, with his first effort. However, in his second try, a wrong item-product was selected. Then, the attacker tried again to log in, but his account was suspended,

so the bidding service could not effieciently process his requirement. Concerning the second attacker, he picked an invalid product, thus his account remained suspended. Lastly, the thrird attacker tried to log in four times. His first try was successful but the rest were not. The three attackers used the pair of username and password, stored in the shared text file, as mentioned above.

5 Related Work

There are many approaches previously proposed for fighting phishing attacks.

The first approach is a browser plug-in, PwdHash [15], that transparently transforms user passwords to domain specific ones. This is done by substituting user's password with a hash of the password and the domain name of the web page that the password is going to be submitted. A main problem of this approach is the implementation in web sites that have multiple domain names.

A second approach is to use dynamically generated virtual skins [4]. Alice may customize her index page in Bob's site and store her preferences to Bob's database. Thus, an attacker can not clone Alice's index page in Bob's site, since he has not access to Alice's preferences, which are stored in Bob's database. However, this approach requires the developing of new habits from users, such as customization of the Internet services they use everyday.

AntiPhish [10] tries to protect users from giving away their sensitive data to untrusted web sites. Users have to specify what they consider sensitive data and which web sites are trusted. However, situations where an attacker tricks Alice to submit her credentials via another media, for example via e-mail, and not in a cloned web site can not be defeated by AntiPhish.

Another interesting approach, illustrated in [16], is based on visual similarity. A legitimate owner of a web page can use this technique to search the Web for suspicious web pages, which are visually similar to the true web page. The visual similarity between two web pages is based on three metrics: block level similarity, layout similarity and overall similarity. A web page is considered as a phishing suspect if any of these similarities to the true web page is higher than a threshold.

An approach, similar to PACTs, is the use of graphical passwords instead of text-based passwords. In an authentication procedure a user is presented with a set of images and he passes the authentication by recognizing and identifying the images he had selected during the registration stage. "Passface" is a technique, based on graphical passwords, developed by Real User Corporation [12]. Through this technique a user will be asked to choose four images of human faces from a database as his future password. In the authentication stage, the user sees a grid of nine faces, consisting of one face previously chosen and eight decoy faces. The user recognizes and clicks on the known face. This procedure is repeated for several rounds, thus the user is fully authenticated if he correctly identifies the four faces. Passfaces replaces or works in conjunction with text passwords. However, passface-based login process takes longer than text passwords. Also, another drawback is that passwords created using this technique

have obvious patterns among them. This makes the passface-based password somewhat predictable.

Apart from the above, there is a series of academic papers [6, 11], which try to detect phishing Web sites based on visual similarities or content based similarities [17]. Past Activity Tests (PACTs) is a very interesting but completely different approach to fight phishing.

6 Conclusion

In this paper we presented Past Activity Tests (PACTs) as a countermeasure against phishing attacks. PACTs rely on the idea that a user has accessed her account in the past, but an adversary is accessing it for the first time. Thus, a user can answer easily a question in regards to her past activity, but the adversary can not.

We presented two real deployments of PACT enabled on-line services. We created accounts for our on-line services and injected their credentials in P2P file sharing networks so as to lure possible adversaries to access our on-line services. The results showed that PACT could resist in an attempt from a third party trying to access the service with stolen credentials.

PACTs are not only an anti-phishing solution, but also a more secure way of general web-based authentication.

Part of our future work is to explore user-friendly PACTs for various on-line services which are not covered in this paper.

References

1. Gnutella. http://www.gnutella.com.
2. Yahoo people search. http://people.yahoo.com.
3. APWG. Anti-phishing working group. http://www.antiphishing.org.
4. R. Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 77–88, New York, NY, USA, 2005. ACM Press.
5. Flickr. Photo sharing service. http://www.flickr.com.
6. A. Y. Fu. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd). *IEEE Trans. Dependable Secur. Comput.*, 3(4):301–311, 2006. Senior Member-Liu Wenyin and Senior Member-Xiaotie Deng.
7. Gmail. A new kind of webmail. http://mail.google.com.
8. Google. Search engine. http://www.google.com.
9. Hotmail. Free e-mail with security by microsoft. http://www.hotmail.com.
10. E. Kirda and C. Kruegel. Protecting users against phishing attacks with antiphish. In *COMPSAC '05: Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05) Volume 1*, pages 517–524, Washington, DC, USA, 2005. IEEE Computer Society.
11. W. Liu, X. Deng, G. Huang, and A. Y. Fu. An antiphishing strategy based on visual similarity assessment. *IEEE Internet Computing*, 10(2):58–65, 2006.
12. Passfaces. Patented graphical passwords for enterprise. http://www.passfaces.com.

13. PHP. Hypertext preprocessor. http://www.php.net.
14. PostgreSQL. The world's most advanced open source database. http://www.postgresql.org.
15. B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell. Stronger password authentication using browser extensions. In *SSYM'05: Proceedings of the 14th conference on USENIX Security Symposium*, pages 2–2, Berkeley, CA, USA, 2005. USENIX Association.
16. L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng. Detection of phishing webpages based on visual similarity. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1060–1061, New York, NY, USA, 2005. ACM Press.
17. Y. Zhang, J. Hong, and L. Cranor. CANTINA: A Content-Based Approach to Detecting Phishing Web Sites. In *Proceedings of the 16th International World Wide Web Conference (WWW2007)*.