# Preventing Phishing Attacks: A Novel Approach

Tushar Goyal, Alay Vakil, Dhrumil Parmar, Rishit Jain
Department of Computer Engineering
S V National Institute of Technology
Surat, India.

Devesh C. Jinwala
Department of Computer Engineering
S V National Institute of Technology
Surat, India.

## ABSTRACT
Phishing is the process of acquiring sensitive information by masquerading as a sensitive entity. Such attacks in turn make it possible for an adversary to orchestrate Denial of Service (DOS) attacks or have sensitive data leaked from an application. With increasing reliance of people on internet based transactions, phishing attacks have also become more sophisticated and have caused large-scale material and trust losses. Hence, dealing with phishing attacks has become a critical issue. Many anti-phishing approaches that are either client-centric or server-centric involving either toolbars, databases or blacklisting have been proposed in the literature. However, we observe that there is a need for an approach that involves both the client and server, and integrates security with the primary task of the user. In this paper, we propose and experiment with an anti-phishing approach that includes server authentication in the client login process. To the best of our knowledge, ours is a novel approach involving server authentication to prevent phishing attacks successfully.

## General Terms
Phishing Attacks, Anti-phishing Techniques.

## Keywords
Security, Phishing Attack, Authentication.

## 1. INTRODUCTION
Phishing is a mechanism that tantamounts to a crime employing both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials [1]. With the advancement of the internet and anti-phishing techniques, the number and types of phishing attacks are also on the rise. The phishing attacks, too, have become more sophisticated. According to the Annual Phishing report published by Anti-Phishing Working Group(APWG) 49480, unique Phishing attacks were reported in July 2013 with 390 unique brands having been compromised [1]. This accounts for a 300% increase with respect to the December 2005 [1].

Phishing has caused numerous prospective e-commerce clients to lose confidence in online transactions and prefer other physical means – thereby, impeding the expansion of e-commerce. Phishing has also triggers other potential attacks that lead to online identity theft, loss of financial security and privacy of internet users. Therefore, it is the need of the hour to keep up with evolving technologies and devise sophisticated methodologies to deal with a spectrum of phishing attacks.

The phishing attacks may be of two broad types viz. either spoofing emails or spoofing of websites or a combination of both [2]. Our focus in this paper, however is only on website based phishing attacks. The website based phishing attacks typically make use of the well-known vulnerabilities in popular web browsers to collect the sensitive information about a user and orchestrate a phishing attack. The problem is aggravated by the fact that the most popular methodology for exchanging information online is by the use of forms. However, using forms involves certain caveats that make phishing attacks difficult to prevent [3] [4].

Firstly, users attribute legitimacy of a webpage by how it looks. An adversary can easily spoof the visual cues of web pages and forms. As a result, user easily gets tricked into entering sensitive information. Secondly, forms are used for both sensitive and insensitive data. A user is generally not aware if an SSL connection is active. A browser cannot really understand the semantics of the data and hence cannot guard the user from sensitive data submission.

Many contemporary Anti-Phishing techniques employ browser extensions and toolbars to display several warning messages. Browser indicators like the URLs and the "SSL Lock" are also encouraged to be noticed. However, newer sophisticated phishing attacks circumvent these measures easily [4].

Unfortunately, as it happens with most of the protocols devised in the Computer Science and Information Technology domain, incorporating security attributes in Web technology has also been a result of an afterthought. Security presently is not a part of the main task the user needs to perform and hence security ends up taking a backseat. Making security a separate task that the user needs to remember is not effective.

In addition, the service providers too, do not follow proper security practices. Moreover, the security warnings flashed to the users do not provide satisfactory explanations. This ultimately causes the user to think that the software is erroneous and not treat the warning seriously. The users are just given a warning and not given any alternatives. Hence, the users are forced to proceed with the dangerous option.

Motivated by the same, in this paper, we propose a new anti-phishing technique that involves both the server as well as the client side functionalities. As we show using our experimentations, our technique indeed successfully mitigates the phishing attacks at reasonable overhead.

The rest of the paper is organized as follows: in the next section, we discuss a brief introduction to the problem. In the third section, we survey the current anti-phishing techniques with their merits and demerits. In the section after that, we discuss our proposed algorithm with following sections discussing another improved version of the algorithm that offers choices with respect to the hashing algorithm and that makes the algorithm suitable to used for the mobile devices. In the section after that we discuss our implementation rsults

with respect to a comparison of universal hash algorithm and SHA2 512 algorithms. We conclude with our observations and their impact in the last section.

We ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download the template, and replace the content with your own material.

## 2. THE PROBLEM FORMULATION

Let us consider a legitimate user who encounters an email in his inbox, which appears to be from his bank that he needs to change his password. User clicks link in the email to change the credentials and he is redirected to a page asking the current password and the new password. The user enters his information believing the webpage. However, the email redirected the user to a malicious URL that tricked the user. The user ended giving away his password to a malicious entity himself. Such webpages not only emulate the look and feel of the original webpage but also have a similar URL that generally goes unnoticed. The phishers or attackers employ wide range of stratagems to attack a user. These maneuvers include installing a malware or Trojan in the host and poisoning host files, injection of a malicious script in the web browser and sending plausible emails.

Security is like a war, or a race between attackers and defenders. The war will never stop. No human can be immune from all virus or bacteria attacks, the same goes for computers. As phishers employ more advanced techniques, the anti-phishing cosmos follows suite. The methodologies implemented to protect the users also evolve alongside attacking approaches. Our next section discusses the widely used anti-phishing techniques with their advantages and disadvantages.

## 3. ANTIPHISHING TECHNIQUES

In this section, we discuss various anti-phishing techniques already published in the literature. We also discuss how our proposed approach improves upon the existing state-of-the-art.

### 3.1 Attribute-based Antiphishing

Attribute phishing based technique employs different kind of checks such as image attribute check, certificate checks and URL checks. It checks whether the URL allocated to page is relatively new. Image attribute cross checks the similarity of suspected webpage to legitimate webpage. The certificate validity and credibility is checked and verified with trusted certificate authority.

The merit of the attribute checking approach is its ability to detect new and unknown phishing attacks. The scheme also detects more number of false websites, hence has a high flase positive rate.

The flipside of this technique is that it is expensive. Multilevel checking for a plethora of cues is slow hence causing high response time even for an authentic website. An example of the use of this approach is PhiishBouncer [5].

### 3.2 Genetic Algorithm-based Antiphishing Techniques

The genetic algorithm based technique mainly works on idea of finding out certain traces in phishing webpage that help in classifying it as illegitimate. The genetic algorithm evolves a simple rule on the basis of principles like natural selection. These evolved rules differentiate between original and phished webpages.

The advantage of Genetic Approach is the detection of phishing email before even user has opened the email. The features like malicious link detection can be provided.

The disadvantage lies in the making up the complex rules. The probability of false positives and the accuracy of the genetic algorithm has to be considered. An example of this technique is presented in [6].

### 3.3 The Character-based Anti Phishing Approach

The Character Based Anti Phishing Approach is based on the Uniform Resource Identifier (URI) structure. A hyperlink is defined as <a href = “hidden destination”> visible text </a>.The information in these hyperlinks are used to classify the emails. If the hidden destination IP is in dotted decimal for-mat, it can be classified as suspicious. If the hidden destination and visible text correspond to different destinations then it can be classified as a phishing attack.

The advantage of this approach is that it can detect both known and unknown phishing attacks. The disadvantage of this approach lies in the false positives because in many places using dotted decimal IP addresses is favourable.Not every phishing links have destination website names in their visible fields in the hyperlinks. This approach may fail for such cases. This technique has been illustrated in [7].

### 3.4 Content Based Anti-Phishing Approach

The main working of Content Based Phishing is the fact that a phishing webpage is short lived and less popular than the original. Hence, they acquire a low rank on search engines. The content based phishing makes use of this rule. It takes the content of a website to be checked and gets its rank in a search engine. Then the algorithm decides the legitimacy of webpage with respect to the rank in search results.

The merit of content based approach is the detection of zero day attacks and very less probability of false positives.

The disadvantage in the content based phishing is due to time taken in getting search results and the probability of attacks on Search Engine’s page rank algorithm. This approach is used in the tool Goldpolish in [8].

### 3.5 Identity Based Anti-Phishing Approach

An antiphishing technique in which the some identity of a website is used as the basis for prevention of the attack is known as Identity based Anti-phishing approach. The technique may use mutual authentication too, wherein a user and a website mutually authenticate each other with a handshake This approach has been used in [9].

## 4. THE PROPOSED APPROACH

The approach that we propose in this paper is based on Client-Server authentication method. Instead of entering a password string in the login page presented to the user, the client is made to enter a hash digest. The hash digest is generated with the help of user password and the random number sent by the server. The hash digest generation is by the help of One Way Hash Functions (OWHF) and random key, hence making the string random and indecipherable at each instance. This approach has been implemented by giving user a lightweight application which facilitates the hash digest generation.

The basic principle behind our approach is as follows: the client side enters the hash digest instead of the plaintext password. Hence, even if adversary sniffs the hash digest, it

cannot get hold of user plaintext password in absence of SSL connection. Since the security generation is with the random number embedded into the process, hence the hash digest generated is different at every instance. This makes cracking the string computationally unfeasible. We show the sequence diagram of the proposed approach in Fig. 1.

Most toolbar based approaches allow the user to ignore the security indicators. However, the new authentication scheme is a part of the login process; a user cannot proceed to do his primary task unless authentication is complete.

We now discuss the proposed algorithm that ensures two way authentication. Hence, each user now can be sure of the server it is communicating with and is hence protected from revealing plaintext password to a phisher. We describe our algorithm in client-server model. The server is presenting the webpage to user and client server interacts by entering data in the forms.

We define the terms which we will be using to explain our algorithm.

Server: A server is a system (software and suitable computer hardware) that responds to requests across a computer network to provide, or help to provide, a network service [10].

Client: A client is a computer system which accesses the services made available by the user [11]. Client side computer system is also provided with a lightweight user application which helps the user generate the hash digest.

Hash Function: A hash function is any function that maps data of arbitrary length to data of a fixed length [12].

Hash Digest: The value returned by a hash function is called hash digest [12].

Random Number: Random number is the 10 byte string with seed value as timestamp.

User Application: The client side application used to run the hash function that takes random number and password from the user and gives the hash digest as password back to the use

## 4.1 The Client Side

The client is presented the login page by server. Here, there is a random number and form to enter username and password. Client calculates the hash of concatenation of Random String and Password. The hash digest is entered as password. This data is sent to server..

## 4.2 Choosing our hash function

A cryptographic hash function is a hash function that takes an arbitrary block of data and returns a fixed-size bitstring, the cryptographic hash value, such that any (accidental or intentional) change to the data will (with very high probability) change the hash value. The data to be encoded are often called the message, and the hash value is called the message digest [13].

The ideal cryptographic hash function has four main properties:

- It is computationally feasible to compute the hash value for any given message.
- It is infeasible to generate a message that has a given hash.
- It is infeasible to modify a message without changing the hash.
- It is infeasible to find two different messages with the same hash.

In the above mentioned Anti Phishing application one of these Hash functions can be used for authenticating the user and to ensure that adversary cannot decode the password submitted by the user. There are many hash functions that have been designed but the most commonly used are viz. the MD5, SHA-1 and SHA-2. The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of security applications. It is also commonly used to check data integrity. In 1996 a flaw was found in the design of MD5 [14]. While it was not a clearly fatal weakness, cryptographers began recommending the use of other algorithms, such as SHA-1.

SHA stands for ”secure hash algorithm”. SHA-1 is a cryptographic hash function designed by the United States National Security Agency. SHA-1 produces a 160-bit(20-byte) hash value. A SHA-1 hash value is typically expressed as a hexadecimal number, 40 digits long. In 2005, cryptanalysts found attacks on SHA-1 suggesting that the algorithm might not be secure enough for on-going use [15].

These hash functions were compared according to our requirements and the results have been explained as follows:

### *4.3.1 The Security*

The security of the MD5 hash function is already compromised. A collision attack exists that can find collisions within seconds on a computer with a 2.6 GHz Pentium 4 processor (complexity of 224.1). Further, there is also a chosen-prefix collision attack that can produce a collision for two chosen arbitrarily different inputs within hours, using off-the-shelf computing hardware (complexity 239). However, that does not preclude its use in applications that do not require collision resistance. Indeed, MD5 is often still used in applications where the smaller key size and speed are beneficial. That said, due to its flaws, researchers recommend the use of other hash functions in scenarios as ours [16].

SHA1 has a flaw that allows collisions to be found in theoretically far less than the 280 steps a secure hash function of its length would require. The attack is continually being revised and currently can be done in 263 steps - just barely within the current realm of computability. For this reason the use of SHA1 is being phased out, with the SHA2 family being used after 2010 [16].

Currently there are no known attacks against SHA2 functions.

### *4.3.2 The Performance*

SHA-1 is noticeably slower than SHA-2 256 therefore SHA-2 is definitely a better choice keeping in mind the security and performance.

**Table 1: Performance Analysis Of Hash Functions [13]**

| Algorithm & its variant | Hashes/millisecond |
|---|---|
| MD5 | 510 |
| SHA-1 | 520 |
| SHA-2 256 | 810 |
| SHA-3 512 | 150 |

Note: Tested on Intel(R) Core(TM) i3-2350M CPU @ 2.30GHz

On the other hand if one wants added precaution one can use SHA-2 512 which is highly secure but compromises on performance.

### *4.3.2 Universal Hash Functions*

According to the KPCB Internet Trends Report 2013 [17] more than 15% of the Internet traffic is from Mobile devices. This indicates a huge volume of internet users using resource constrained mobile devices for a variety of online jobs including online transactions. These users are as susceptible to Phishing attacks as any other user. Hence, anti-phishing techniques need to be applied to protect such users too. The Anti-Phishing technique that can be deployed on mobile devices needs to have better time and space efficiency. Only then can one ensure that the mobile user is protected without affecting the other operations of the mobile device.

We have used the SHA-2 Hash algorithm. Its effective but bulky as far as time and space usage is considered. Therefore we considered other lightweight hash functions. Ultimately we decided to use an approach that combined nine extremely lightweight General Purpose Universal Hash Functions [19]. The light weight hash functions that we have used are PJWHash, ELFHash, BKDRHash, SDBMHash, DJBHash, DEKHash, APHash, BPHash, FNVHash.In resource constrained environments a Bloom filter based checking approach may be used to ensure authentication.

## 5. GENERATING THE RANDOM STRING

The random string generated by the server side consists of two parts. The first part consists of the latitude and the longitude of the user. The location of the user is achieved using Geolocation API in JavaScript. The second part has 10 characters consisting of upper case characters and special characters. The server side script returns a pseudo-random integer that uses the current timestamp as its seed value. Both these parts are concatenated together to form the final Random String.

## 6. PERFORMANCE RESULTS AND ANALYSIS

We compared the performance of both our selected hash functions for different environments: The SHA-2 512 algorithm and the universal hash function made using the lightweight universal hash functions. The simulation of SHA-2 512 algorithm was done using PHP with MySQL server backend. The client application given to user was developed using Python with Qt framework.

For each password length from 1 - 100 we generated hash digests using both the algorithms for 1000 random instances. We found that the universal hash function had approximately half the execution time than that of SHA-2 512. Hence the universal hash function is more suitable for limited resource systems.

## 7. CONCLUSION

Phishing is an important problem that results in identity and critical information theft. Though simple, it is highly effective and accounts for losses of almost billions of dollars. The efficacy of these attacks is due to simplicity of execution and the existence of a lot of gullible users online. These attacks are hence important and surely need to be rectified.

In this research we surveyed the different Phishing and Anti-Phishing Techniques .We then devised an algorithm involving two directional authentications. Both the Server and Client authenticate each other using hash digests of pre-shared knowledge. However, with the phenomenal growth of mobile internet users it has become necessary to protect mobile users too. Mobile devices are resource constraint. Therefore, we adapted our approach to mobile environment with lightweight universal hash functions and location based random Number generation.

Our solution becomes a part of the primary work pathway of the user. This makes sure that security does not take a backseat nor does it hinder the actual work of the user. However our future work includes protecting or alerting the user of phishing attack attempts even when the client machine has been remotely compromised.

## 8. REFERENCES

[1] Anti-Phishing Working Group. Phishing Activity Trends Report, Third Quarter 2013. URL: http://antiphishing.org/resources/apwg-reports/ Last Accessed: April 2015

[2] Engin Karda, Christopher Kruegel. "Protecting Users against Phishing Attacks" The Computer Journal, pp. 1-8, Vol 00, Issue 0, The British Computer Society, 2005

[3] Min Wu , Robert C. Miller , Greg Little WebWallet:Preventing Phishing Attacks by Revealing User Intentions, Symposium On Usable Privacy and Security (SOUPS) 2006, July 12-14, 2006, Pittsburgh, PA, USA.

[4] Min Wu, Robert C. Miller, Simson L. Garfinkel Do Security Toolbars Actually Prevent Phishing Attacks?, CHI 2006, April 22-27, 2006, Montral, Qubec, Canada.

[5] Michael Atighetchi, Partha Pal, "Attreibute-based prevention of Phishing Attacks", Proceedings of the 8th IEEE International Symposium on Network Computing and Applications, 2009.

[6] V. Shreeram, M Suban, P Shanthi, K Manjula, "Antiphishing detection of phishing attacks using genetic algorithm", Proceedings of the International Conference on Communication Control and Computing Technology, pp. 447-450, 2010

[7] Juan Chen, Chuan Xiong Guo "Online Detection and Prevention of Phishing Attacks" , Proceeings of the First International Conference on Commmunication and Networking in China, Beijing, pp. 1-7, 2007.

[8] Matthew Dunlop, Stephen Groat, David Shelly, " Goldpolish: Using Images for Content-based Phishing Analysis, In Proceedings of the Fifth Internationnal Conference on Internet Monitoring and Protection, Barcelona, pp. 123-128, 2010.

[9] Hicham Tout, William Hafner "Phishpin: An identity-based anti-phishing approach" in proceedings of international conference on computational science and engineering, Vancouver, BC, pages 347-352, 2009 .

[10] URL: http://en.wikipedia.org/wiki/Server (computing). Last Accessed: April 2015

[11] URL: http://en.wikipedia.org/wiki/Client (computing). Last Accessed: April 2015

[12] URL:http://en.wikipedia.org/wiki/Hashing function. Last Accessed: April 2015

[13] URL: http://en.wikipedia.org/wiki/CryptographicHash Function. Last Accessed: April 2015

[14] URL: http://en.wikipedia.org/wiki/Md5. Last Accessed: April 2015

[15] URL:http://en.wikipedia.org/wiki/SHA-1. Last Accessed: April 2015

[16] URL:http://www.not-implemented.com/comparing-hash-algorithms-md5-sha1-sha2/. Last Accessed: April 2015

[17] Mary Meeker, Liang Wu KPCB. Internet Trends Report 2013, Internet Trends D11 Conference 29th May 2013.

[18] URL:http://www.slideshare.net/kleinerperkins/kpcb-internet-trends-2013. Last Accessed: April 2015

[19] Arash Partow Python and PHP Implementations for Lightweight Hash functions URL: http://www.partow.net/programming/hashfunctions. Last Accessed: April 2015.

## 9. APPENDIX

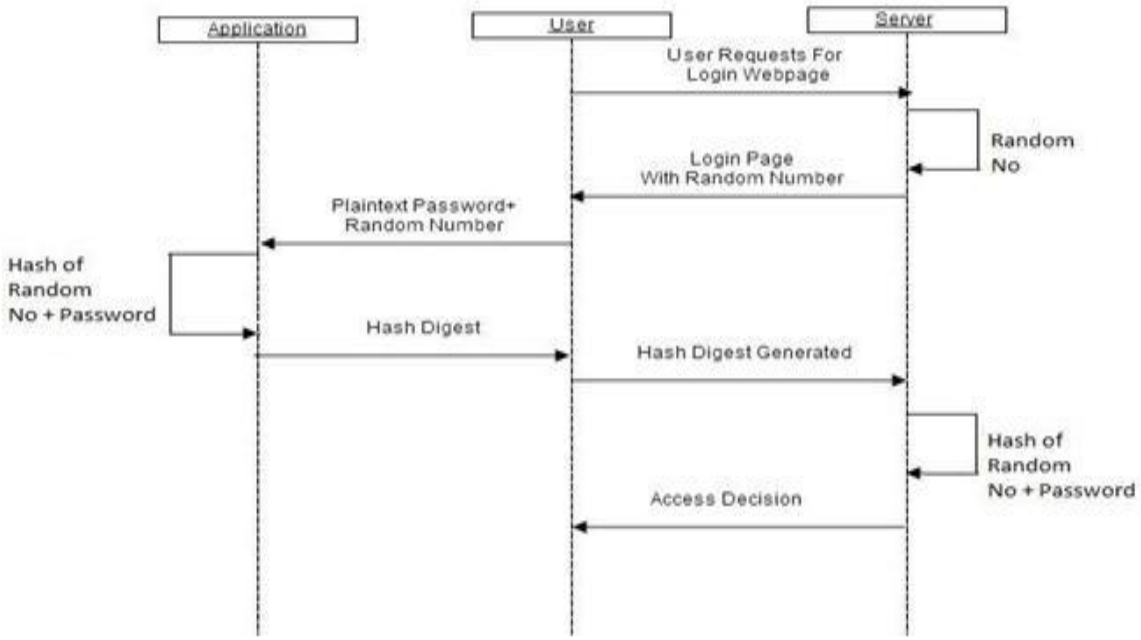


**Fig 1. Sequence Diagram of the Proposed Approach**