ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 1, pp: (1-4), Month: January - March 2016, Available at: www.researchpublish.com

Security Skins: Embedded, Security Indicators

[1]Ms. Madhavi Joshi, [2] Ms. Yeole Madhavi, [3]Ms. Bhondave Snehalata, [4] Mr. Y.K. Patil

[1, 2, 3] BE Student, Department of Computer Science & Engineering, JSPM Wagholi, Pune, India
[4]Assistant Professor, Department of Computer Science & Engineering, JSPM Wagholi, Pune, India

Abstract: **In Dynamic security skin, Passwords are the key to all digital secrets. Passwords stay on the most widely used confirmation method even with their well-known security weakness. Text password is the most accepted form of user authentication on websites due to its relieve and ease. This paper gives a check on different method that were introduce for safe guard of password on a network. This paper present a scheme, Dynamic protection Skins, that allows a remote web server to prove its self in a way that is easy for a human user to verify & hard for an invader to skit. These schemes include a new password authentication & key-exchange protocol suitable for authenticating users & exchanging keys over an un-trusted network. In small DSS decrease user memory wants. As the final result the DSS provide toughest security.**

Keywords: **substantiation, domino result, glossary attacks, phishing.**

1. INTRODUCTION

In this paper, we studied the case of users authenticate web sites in the context of phishing attacks. Code word authentication protocol come in many flavour, but they all solve the same problem: One party must somehow show to one more party that it knows some password P, typically set in advance. Such protocols range from the trivial to the very complex, [1] and many of them present some form of defence from a diversity of attacks mount by hateful or extremely curious third parties.

In a phishing attack, the attacker spoofs a website (e.g., a financial services website). The enemy draw a casualty to the rascal website; occasionally by embed a linkage in message & heartening the user to snap on the bond.

Security is a division of in a row safety that deals exclusively with sanctuary of websites, web application and web forces. It is the main step to admission to any websites and hence the hackers always try to sneak through the usual security events and gain illegitimate access. Due to the general use of passwords, many security fear are involved in this area. Nowadays, even banking dealings are perform using passwords. Different researchers studied different types of passwords, their profit and drawback.

2. INFORMATION ANALYSIS

In this project, we inspect the case of users authenticate websites in the situation of phishing attacks. [4] The figure of unique phishing news submitted to APWG in the third section of 2009 reached an all time high of 40,621 in August, a number nearly 5.5 percent higher than the previous record high of 38,514 reported in September 2007.

● Unique Phishing reports submit to APWG, 2009 reached a record 40,621 in August, 5.5 percent more than the previous trace in September, 2007.

● Single Phishing websites reported to APWG reach a record 56,362 in August, displacing the previous record of 55,643 by 1.3 percent in April, 2007.

● The number of hijack brand rise to a proof 341 in August, up more than 10 percent from the earlier record of 310 in March 2009.

● Monetary air force rise back to the top of most under fire business sectors in 2009 after a short disarticulation by expense Services in 2009.

● Entire number of infected computers dropped to 11,001,646 in 2009, representing more than 48.35 percent of the total sample of scanned computers.

The Phishing difficulty show that we as security designers have a distance to travel. Because both attackers and designers use user interface tools, [2] examining this problem yields near into usability design for other time alone and safety areas. We inspect safety properties that make phishing a testing design difficulty.

3. REFUGE PROPERTIES

➢ **Partial being skills assets:**

This aim appears clear, but it implies a diverse come close to towards the design of security systems , humans are not universal principle computer. They are restricted by their natural skills and ability.

➢ **Universal point graphics assets:**

Condition, we are structure a scheme with the aim of is calculated to oppose spoofing we must presuppose that standardized realistic designs can be without problems copied.

➢ **Fair arch assets:**

We should leave to unusual lengths to check populace beginning mechanically transmission hope based on logos on your own. This model applies to the point out of refuge needle with icons while all right.

➢ **Unenthusiastic customer assets:**

Defence is regularly a resulting purpose, Generally users rather to meeting point on their main tasks, with so designers cannot wait for users to be highly enthused to direct their refuge.

➢ **Hangar access assets:**

This goods encourages us to plan systems that place a high right of way on helping users to keep susceptible data before it trees their run.

4. DYNAMIC SECUTITY SKINS

Energetic safety Skins is a scheme that allows in the sticks web browser to prove its self in such a way which is easy for being user to know & hard for enemy to satire.

We chose to mechanically recognize real web pages and their contented with aimlessly generated images. In this section we describe some approaches.

1. Browser-Generated Random Images:

Assume that the browser generate a random number at the start of every verification deal. This number is known only to the browser, and is used to generate a unique image that will only be used for that transaction. The generated image is used by the browser to create a patterned window border. Once a server is effectively authenticated, the browser presents each webpage that is generated by that server using its own unique window border. The pattern of the window border is at the same time displayed in the user's trusted window. To validate a particular server window, the user only needs to make sure that two patterns match. All non-authenticated windows are displayed by the browser using a radically different, solid, non-patterned border, so that they cannot be mistaken for valid windows.

2. Server-Generated Random Images:

We describe an come up to for the server to make images that can be used to mark trusted content. To achieve this, we take gain of some properties of the SRP protocol . In the last step of the protocol, the server presents a hash value to the user, which proves that the server holds the user's verifier. In our scheme, the server uses this value to generate an abstract image, using the visual hash algorithm described above. The user's browser can independently reach the same value as the server and can compute the same image (because it also knows the values of the verifier and the random

values supplied by each party). The browser presents the user with the image that it expects to receive from the server in the trusted password window. Neither the user nor the server has to store any images in advance, since images are computed quickly from the seed.

3. Reliable trail towards the key Window:

Our addition provides the user with a **trusted password window** that is dedicated to password entry and display of security information. We establish a trusted path to this window by assigning [5] each user a random photographic image that will always appear in that window. We refer to this as the user's ***personal image*.** The user should easily be able to recognize the personal image and should only enter his password when this image is displayed. As shown in Figure 1, the personal image serves as the background of the window. The personal image is also transparently overlaid onto the textboxes. This ensures that user focus is on the image at the point of text entry and makes it more difficult to spoof the password entry boxes.

Figure 1: Trusted Password Window

4. Verifier Based Protocol:

We present a simple impression of the protocol to give an insight for how it works. To start, Carol chooses a password, picks a chance salt, and applies a one-way function to the code word to make a verifier. She send this verifier and the salt to the server as a one-time operation. The server will store the verifier as Carol's "password".[3] To login to the server, the only data that she needs to provide is her username, and the server will look up her salt and verifier. Next, Carol's client sends a random value to the server chosen by her client.

The server in turn sends Carol its own random values. Each party, using their knowledge of the verifier and the random values, can reach the same session key, a common value that is never shared. Carol sends a proof to the server that she knows the session key (this proof consists of a hash of the session key and the random values exchanged earlier). In the last step, the server sends its proof to Carol (this proof consists of a hash of the session key with Carol's proof and the random values generated earlier). At the end of this interaction, Carol is able to prove to the server that she knows the password without revealing it.

5. Secure Remote Password Protocol:

Our goal is to get verification of the user and the server, without significantly altering user password performance or increasing user recall burden. We chose to implement a verifier-based protocol. These protocols differ from square shared-secret confirmation protocol in that they do not entail two parties to share a secret key to authenticate each other. SRP allows a user and server to validate each other over an un-trusted system. We chose SRP because it is lightweight, well analyzed and has many useful properties. Namely, it allow us to preserve the familiar use of passwords, with no require the user to send his password to the server.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 1, pp: (1-4), Month: January - March 2016, Available at: www.researchpublish.com

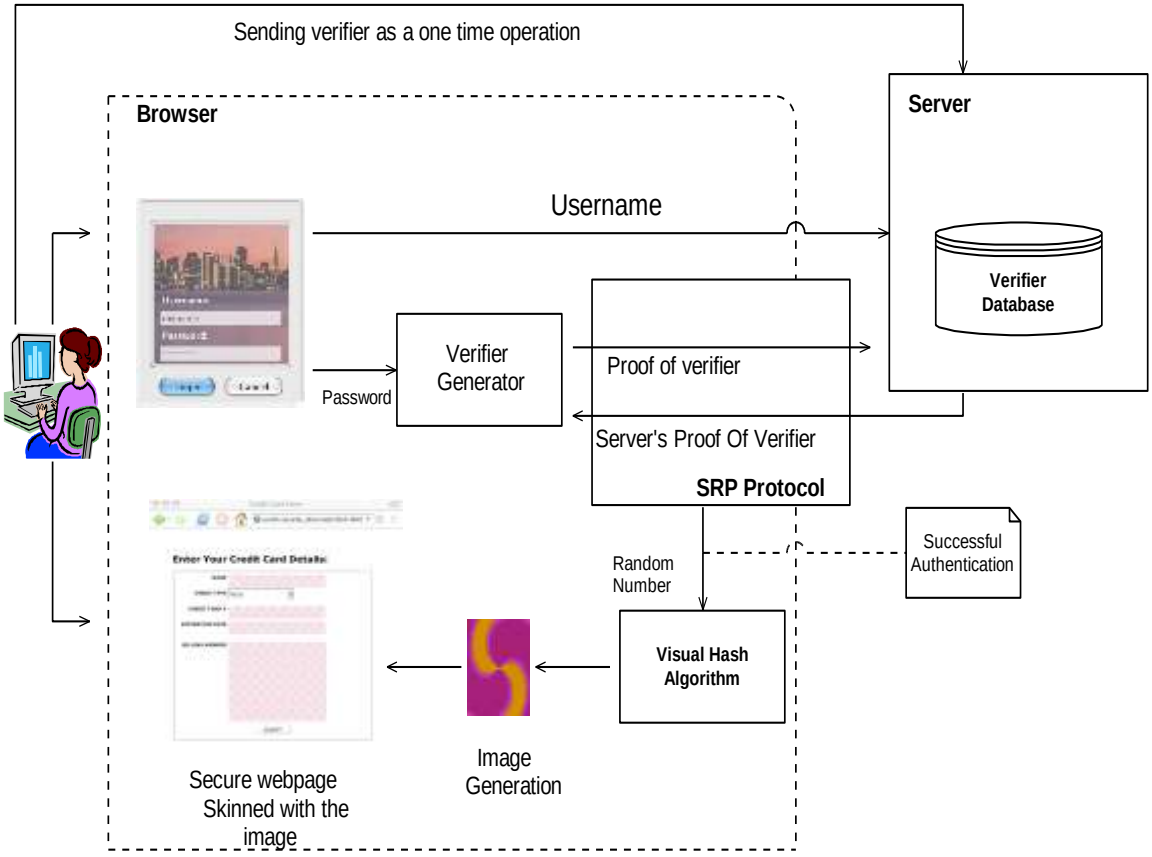

Figure 2: Block Diagram of DSS

5. CONCLUSION

Our whole project is java based and we know that java is most secured & reliable platform. As we are beginner so we didn't have money to invest in software and java is free to use, we choose database according to that only.

SRP is an improvement[6] on standard login/authentication programs. While adding any type of authentication can be seen as an improvement on current systems, SRP offers the advantages of being both unobtrusive to the day to day lives of users as well as not being too complex (or large) to be implemented in hardware or software, as we have shown.

Finally we are very proud to say that we worked as a team during the whole project work and we tried our best to complete the project according to requirement.

REFERENCES

[1] Rachana Dhamija & J.D.Tygar,"The Battle Against Phishing: Dynamic Security Skins"

[2] Tom Wu,"Secure Remote Password (SRP) Authentication", *Stanford University*.

[3] RFC 2944 – Telnet Authentication: SRP.

[4] RFC 2945 – The SRP Authentication and Key Exchange System.

[5] Patrick Naughton and Herbert Schildt,"Java 2: The Complete Reference",*Osborne/McGraw-Hill.*

[6] "SRP Design Specifications ,SRP JavaScript Demo",*www.srp.standford.edu*

[7] William Stallings, "Cryptography and Network Security: Principles and Practices", Pearson Education, Third Edition "java information", *www.sun.java.com*

[8] "phishing information", *www.apwg.org*