VOL. 10, NO. 13, JULY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

PROVIDING PASSWORD SECURITY BY SALTED PASSWORD HASHING USING BCRYPT ALGORITHM

P. Sriramya and R. A. Karthika
Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha University, Chennai, India
E-Mail: sriramya82@yahoo.com

ABSTRACT

World Wide Web has become a popular medium to search information, business, trading and so on. Various organizations and companies are also employing the web in order to introduce their products or services around the world. E-commerce is any type of business or commercial transaction that involves the transfer of information across the internet. A huge amount of information is generated and stored in the web services. This document is intended for System end-users, System architects and System developers and Software Testers. This project focuses on providing security to user's data by using Salted Password Hashing Technique. Shopping Online can be so vulnerable, since the user information are saved as a plain text in their database. To overcome this scenario hashing is used. This project focuses on saving an encrypted user data to the database rather their saving as a plain text. To provide more security to user data Bcrypt algorithm is implemented. Bcrypt algorithm can encrypt the data up to 512bits which provides a longer encryption key and give hashed value of the user data. Hash functions are primarily used in hash tables, to quickly locate a data records.

Keywords: bcrypt algorithm, salt, lookup table, rainbow table.

1. INTRODUCTION

WEB security is a procedure, practices and technologies for assuring the reliability, predictable operation of the web server, web browser, other programs that communicate with web server and the surrounding internet infrastructure [17]. Various encrypt ion algorithm, cryptographic protocol and cryptography algorithm are used to secure password and data transfer.

If storing password in a plain text or is compromised through easy encryption method then there are possibilities of decrypting of password and stolen. It may be result in fake login and loss of privacy.MD5 (Merkl-Damagerd), SHA1 (Secure Hash Algorithm) and RIPEMD (RACE Integrity Primitives Evaluation Message Digest) algorithm are considered as broken algorithm and we should not use it our new application code [18] from cryptography. To secure data and password SHA256, SHA512, RipeMD, and WHIRLPOOL are cryptographic hash functions can be used. Hashing password is better method then encryption of password because hashing is a one way function – we cannot discover plain text value from its hash [18] means the plain password that construct hash cannot be regenerated from its hash value.

Hashing is sensitive to the dictionary attack. Dictionary attack is a method of recovering password from known password. So it is possible to crack hash password by using pre-calculated hash value or using hash dictionary. Hashing algorithms are very deterministic as they produce same hash value for same input text. Raw hashes are also vulnerable to rainbow tables, a method of balancing a need for pre - computation of hashes and the obviously large storage is necessary to keep an entire dictionary of hashes [18]. To avoid these problem, salting comes to our rescue. But while implementing salt hashing don't reuse the same salt in hash function, too small salt and don't hard code it in the program for password Web security is a procedure, practices and technologies for assuring the reliability, predictable operation of the web server, web browser, other programs that communicate with web server and the surrounding internet infrastructure [17].

Various encryption algorithm, cryptographic protocol and cryptography algorithm are used to secure password and data transfer. If storing password in a plain text or is compromised through easy encryption method then there are possibilities of decrypting of password and stolen. It may be result in fake login and loss of privacy. MD5 (Merkl - Damagerd), SHA1 (Secure Hash Algorithm) and RIPEMD (RACE Integrity Primitives Evaluation Message Digest) algorithm are considered as broken algorithm and we should not use it our new application code [18] from cryptography. To secure data and password SHA256, SHA512, RipeMD, and WHIRLPOOL are cryptographic hash functions can be used. Hashing password is better method then encryption of password because hashing is a one way f unction – we cannot discover plain text value from its hash [18] means the plain password that construct hash cannot be regenerated from its hash value. Hashing is sensitive to the dictionary attack. Dictionary attack is a method of recovering password from known password. So it is possible to crack hash password by using pre-calculated hash value or using hash dictionary.

Hashing algorithms are very deterministic as they produce same hash value for same input text. Raw hashes are also vulnerable to rainbow tables, a method of balancing a need for pre - computation of hashes and the obviously large storage is necessary to keep an entire dictionary of hashes [18]. To avoid these problem, salting comes to our rescue. But while implementing salt hashing don't reuse the same salt in hash function, too small salt and don't hard code it in the program for password creation.

Hash is a kind of process, signature, function

which is responsible for translating information into a cryptic value. The concept of hash and encryption is almost same. In practical view Hash is an algorithm that takes an arbitrary block of data and returns a fixed-size bit string. Hashing is also known for its unidirectional process because it does not require rehashing or decrypting to get back data. In hashing the data which is needed to be encoded is often called the "message," and the outcome of hash value after processing is sometimes called the message digest or simply digests.

While hashing message, an algorithm is utilize which work is to map input values to a series of known output values. So given the same series of input values, a hash algorithm always produces the same output values. Hashing is an industry supported standard similar to encryption. While creating and designing hash functions, we generally come across with ways:

Case 1: When we have no idea about the distribution of the incoming keys.

- Hash function evenly distributes the key range across the hash table.

Case 2: When we have a little bit idea and information about the distribution of the incoming keys.

- Distribution-dependent hash function that can avoid assigning clusters of related key values to the same hash table slot.

2. LITERATURE SURVEY

Yu-Chi Chen, Gwoboa Horng, Chang-Chin Huang, describes the blind decoding schemes are proposed as tools for protecting customers' privacy in on-line shopping for electronic documents such that the company has no way of knowing which documents the customers have purchased. Most of the blind decoding schemes suffer from the oracle problem. Schemes utilizing the transformability of digital signatures were proposed to ensure the correctness of the requests from the customers. In this paper, a secure blind decoding scheme based on RSA scheme is proposed. It does not utilize the transformability of RSA digital signature [12].

Niels Provos and David Mazeries, proposes ways of building systems in which password security keeps up with hardware speeds. We formalize the properties desirable in a good password system and show that the computational cost of any secure password scheme must increase as hardware improves. We present two algorithms with adaptable costeksblowsh a block cipher with a purposefully expensive key schedule and bcrypt_ a related hash function. Failing a major breakthrough in complexity theory these algorithms should allow password based systems to adapt to hardware improvements and remain secure well into the future [13].

Pritesh N, Jigisha K and Paresh V, describes ways of building systems in which password security keeps up with hardware speeds. We formalize the properties desirable in a good password system and show that the computational cost of any secure password scheme must increase as hardware improves. We present two algorithms with adaptable cost a block cipher with a purposefully expensive key schedule and bcrypt a related hash function. Failing a major breakthrough in complexity theory these algorithms should allow password based systems to adapt to hardware improvements and remain secure well into the future.

Thulasimani Lakshmanan and Madheswaran Muthusamy, proposes Hash functions are the most widespread among all cryptographic primitives, and are currently used in multiple cryptographic schemes and in security protocols. The basic design of SHA- 192 is to have the output length of 192.The SHA-192 has been designed to satisfy the different level of enhanced security and to resist the advanced SHA attacks. The security analysis of the SHA-192 is compared to the old one given by NIST and gives more security. The SHA-192 can be used in many applications such s public key cryptosystem, digital sign encryption, message authentication code, random generator and in security architecture of upcoming wireless devices like software defined radio etc [14].

Janaka Deepakumara, Howard M. Heys and R. Venkatesan, describes message authentication is an essential technique to verify that received messages come from the alleged source and have not been altered. The input message may be arbitrarily large and is processed in 512-bit blocks by executing 64 steps involving the manipulation of 128-bit blocks. It is reasonable to construct cryptographic accelerators using hardware implementations of HMACs based on a hash algorithm such as MD5. Two different architectures, iterative and full loop unrolling, of MD5 have been implemented using Field programmable Gate Arrays (FPGAs). The performance of these implementations is discussed [15].

3. EXISTING SYSTEM

The data which is given in database will be encrypted to 128 bits.

The drawback of this method is its sensitivity to the presence of model identification text or other objects above or below the vehicle that can disturb the texture histogram to differentiate between text and other image types to the license plates.

The main drawback of these segmentation techniques was their intensive computational demand and also sensitivity to the presence of other text such as bumper stickers or model identification.

4. PROPOSED SYSTEM

BCRYPT is a key derivation function for passwords designed by Niels Provos and David Mazières, based on the Blowfish cipher, and presented at USENIX Besides incorporating a salt to protect against rainbow table attacks, BCRYPT is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power.

The bcrypt function is the default password hash algorithm for BSD and many other systems. The prefix

"$2a$" or "2y" in a hash string in a shadow password file indicates that hash string is a bcrypt hash in modular crypt format. The rest of the hash string includes the cost parameter, a 128-bit salt (base-64 encoded as 22 characters), and the 192-bit hash value (base-64 encoded as 31 characters).

Blowfish is notable among block ciphers for its expensive key setup phase. It starts off with sub-keys in a standard state, then uses this state to perform a block encryption using part of the key, and uses the result of that encryption (which is more accurately a hashing) to replace some of the sub-keys. Then it uses this modified state to encrypt another part of the key, and uses the result to replace more of the sub-keys. It proceeds in this fashion, using a progressively modified state to hash the key and replace bits of state, until all sub keys have been set.

Provos and Mazières developed a new key setup algorithm for Blowfish, dubbing the resulting cipher "Eksblowfish" ("expensive key schedule Blowfish"). The key setup begins with a modified form of the standard Blowfish key setup, in which both the salt and password are used to set all sub-keys. There are number of rounds in which the standard Blowfish keying algorithm is applied, using alternately the salt and the password as the key, each round starting with the sub-keys state from the previous round. Crypto-theoretically, this is no stronger than the standard Blowfish key schedule, but the number of rekeying rounds is configurable; this process can therefore be made arbitrarily slow, which helps deter brute-force attacks upon the hash or salt.

BCRYPT is currently the secure standard for password hashing. It's derived from the Blowfish block cipher which, to generate the hash, uses look up tables which are initiated in memory. This means a certain amount of memory space needs to be used before a hash can be generated. This can be done on CPU, but when using the power of GPU it will become a lot more cumbersome due to memory restrictions. Bcrypt has been around for 14 years, based on a cipher which has been around for over 20 years. It's been well vetted and tested and hence considered the standard for password hashing.

There is actually one weakness, FPGA processing units. When bcrypt was originally developed its main threat was custom ASICs specifically built to attack hash functions. These days those ASICs would be GPUs (password brute forcing can actually still run on GPU, but not in full parallelism) which are cheap to purchase and are ideal for multithreaded processes such as password brute forcing. FPGAs (Field Programmable Gate Arrays) are similar to GPUs but the memory management is very different. On these chips brute forcing bcrypt can be done more efficiently than on GPUs, but if you have a long enough password it will still be unfeasible. The iteration count is a power of two, which is an input to the algorithm. The number is encoded in the textual result.

Finally displays or records the resulting. Figure-1 gives the block diagram for the fundamental sequence involved in WEB APPLICATION system.

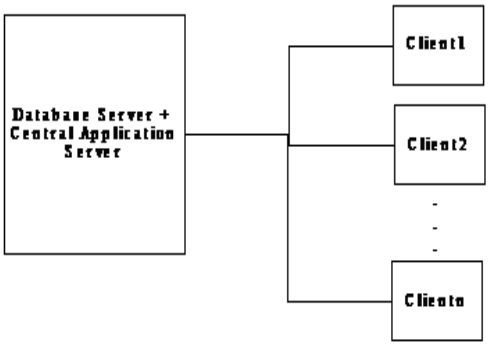

Figure-1. Sequence of web application system.

The functional requirements define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. For each function, specify requirements on inputs, processing, and outputs. These are usually organized with these four sub-paragraphs:

(1) Purpose of the function: Provide rationale to clarify the intent of the function.

(2) Inputs: Sources, valid ranges of values, any timing concerns, operator requirements, special interfaces.

(3) Operations to be performed: Validity checks, responses to abnormal conditions, types of processing required.

(4) Outputs: Destinations, valid ranges of values, timing concerns and handling of illegal values, error messages, and interfaces required.

a) Salted password hashing

The most important aspect of a user account system is how user passwords are protected. User account databases are hacked frequently, so you absolutely must do something to protect your users' passwords if your website is ever breached. The best way to protect passwords is to employ salted password hashing.

b) Password hashing

Hash algorithms are one way functions. They turn any amount of data into a fixed-length "fingerprint" that cannot be reversed. They also have the property that if the input changes by even a tiny bit, the resulting hash is completely different. This is great for protecting passwords, because we want to store passwords in a form that protects them even if the password file itself is compromised, but at the same time, we need to be able to verify that a user's password is correct.

The general workflow for account registration and authentication in a hash-based account system is as follows:

1. The user creates an account.
2. Their password is hashed and stored in the database. At no point is the plain-text (unencrypted) password ever written to the hard drive.

3. When the user attempts to login, the hash of the password they entered is checked against the hash of their real password (retrieved from the database).
4. If the hashes match, the user is granted access. If not, the user is told they entered invalid login credentials.
5. Steps 3 and 4 repeat every time someone tries to login to their account.

In step 4, never tell the user if it was the username or password they got wrong. Always display a generic message like "Invalid username or password." This prevents attackers from enumerating valid usernames without knowing their passwords.

It should be noted that the hash functions used to protect passwords are not the same as the hash functions you may have seen in a data structures course. The hash functions used to implement data structures such as hash tables are designed to be fast, not secure. Only cryptographic hash functions may be used to implement password hashing. Hash functions like SHA256, SHA512, RipeMD, and WHIRLPOOL are cryptographic hash functions.

It is easy to think that all you have to do is run the password through a cryptographic hash function and your users' passwords will be secure. This is far from the truth. There are many ways to recover passwords from plain hashes very quickly. There are several easy-to-implement techniques that make these "attacks" much less effective.

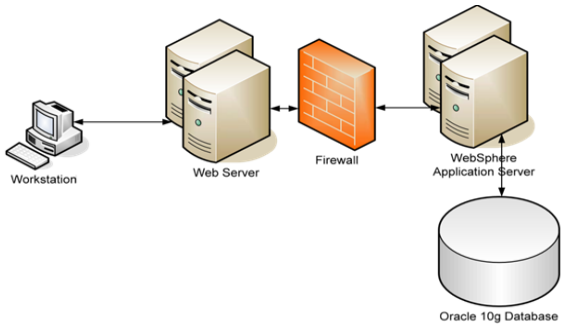

Figure-2. Topology diagram.

In the above Figure-2 Web server is responsible for serving web pages, mostly HTML pages, via the HTTP protocol to clients. The Web server sends out web pages in response to requests from browsers. A page request is generated when a client clicks a link on a web page in the browser. Application server hosts the online screening application and hosts the business logic and the business model classes of applications. It serves requests for dynamic HTTP web pages from Web servers. Oracle 10g Online Screening database stores the screening data, program and question information, audit trails of screening application in relational format. Hyper Text Transport Protocol is the communication protocol used to connect to servers on the World Wide Web.

The primary function of HTTP is to establish a connection with a Web server and transmit HTML pages to the user's browser. Java Database Connectivity is an application program interface (API) specification for connecting programs written in Java to the data in popular databases. The application program interface lets you encode access request statements in structured query language (SQL) that are then passed to the program that manages the database. It returns the results through a similar interface. XML is a programming language/ specification developed by the W3C, for organizing and tagging elements of a document so that the document can be transmitted and interpreted between applications and organizations.

5. APPLICATION ARCHITECTURE

Application architecture defines various components and their interactions in context of a whole system. Application architecture is the critical software that bridges the architectural gap between the application server and the application's business logic, thereby eliminating the complexities and excessive costs of constructing, deploying and managing distributed enterprise applications.

Online Screening Tool will have a layered application architecture which provides some of the key features as below:

- **Structure:** Organizing applications along business-level boundaries and not technical boundaries
- **Speed and flexibility:** Making application changes through configuration and not programming
- **Control:** Modifying, extending or overwriting any architectural element.
- **Reuse:** Achieving greater reusability and integration by loosely coupling application logic to infrastructure.

At a conceptual level, they represent distinct and cohesive aggregations of functionality. Online Screening Tool design is based on a tiered approach. A tier is a logical partition of the separation of concerns of the system. Each tier is assigned its unique responsibility in the system. Each tier is viewed as logically separated from one another and is loosely coupled with the adjacent tier. Online Screening Tool architecture can be represented in the below Figure-3.

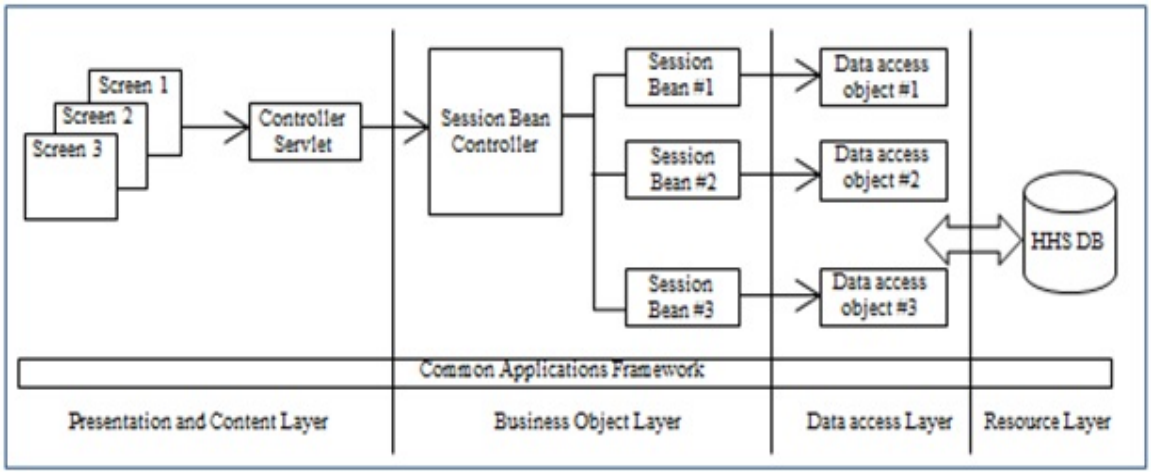

Figure-3. Online screening tool architecture.

6. CONCLUSION AND FUTURE WORK

Privacy and authenticity are the two concerns of the global network users. These issues are solved based on the science of cryptography. Password storage security is one important aspect of data security as most systems nowadays require an authentication method using passwords. Hashing algorithms are commonly used for encrypting plaintext passwords into strings that theoretically cannot be deciphered by hackers due to their one-way encryption feature. However, with time, attacks became possible through the use of dictionary tables and rainbow tables. In this paper, we discussed different methods to thwart these attacks: (1) the use of a strong password to reduce the probability of it existing in a dictionary, (2) using salts, (3) key stretching and iteration hashing to make the bcrypt computation slower, (4) chaining method, where the output of one iteration is used in the input of the next iteration and the use of a different initialization vector for each password. In this paper, we have addressed the Salted Password Hashing Technique using bcrypt algorithm for providing the user's privacy when shopping online.

In future, this application can be used in many social websites to provide security to their users. Databases are most commonly needed for any kind of applications, to safeguard the personal data or information in database bcrypt hashing is more applicable. This will make the any kind of data more secure. The case off many decryptors is also interesting and deserves further investigation.

REFERENCES

[1] Antony G. Robertiello and Kiran A. Bandla. 2005. "Attacks on MD5 Hashed Passwords," Technical Report, George Mason University, USA.

[2] C.H. Chen, G. Horng and C.H. Hsu. 2009. A novel private information retrieval scheme with fair privacy in the user side and the server side, Int. J. Innovat. Comput. Inf. Control, Vol. 5, No. 3, pp. 801–810.

[3] C. Percival and S. Josefsson. 2012. "The scrypt PasswordBased Key Derivation Function," Internet Draft, Internet Engineering Task Force, September.

[4] D. Boneh, E. Kushilevitz, R. Ostrovsky and W. Skeith III. 2007. Public key encryption that allows PIR queries, in: A. Menezes (Ed.), Advances in Cryptology – Crypto 2007, Lecture Notes in Computer Science, vol. 4622, Springer-Verlag, pp. 50–67.

[5] Eastlake, Donald E. 3rd, and Jones, Peter. 2001. US Secure Hash Algorithm 1 (SHA1) (online), Internet Engineering Task Force.

[6] M. Al-Fayoumi and S. Aboud. 2005. Blind decryption and privacy protection, Am. J. Appl. Sci. Vol. 2, pp. 873–876.

[7] M. Bellare, C. Namprempre, D. Pointcheval and M. Semanko. 2003. The one more-RSA-inversion problems and the security of Chaum's blind signature scheme, J.Cryptol. Vol. 16, pp. 185–215.

[8] M. Dürmuth, T. Güneysu, M. Kasper, C. Paar, T. Yalçin, and R. Zimmermann. 2012. "Evaluation of Standardized Password-Based Key Derivation against Parallel Processing Platforms," in Computer Security – ESORICS 2012, pp. 716–733.

[9] Sotirov, Alexander, Stevens, Marc, Appelbaum, Jacob, Lenstra and Arjen *et al.* 2008. MD5 considered harmful today (online), Technische Universiteit Eidhoven.

[10] Wang, Xiaoyun and Yu, Hongbo. 2005. How to Break MD5 and Other Hash Functions, Lecture Notes in Computer Science, Vol. 3494, pp 561-577.

[11] Wiemer F. Horst gortz inst. For it-security (hgi), Ruhr-univ. Bochum, bochum, germany Zimmermann, R., High-speed Implementation of bcrypt password search using special-purpose hardware, December 2014.

[12] Y.C. Chen, G. Horng and C.C. Huang. 2009. Privacy protection in on-line shopping for electronic documents, in: 5th International Conference on Information Assurance and Security, Vol. 2, pp. 105–108.

[13] Niels Provos and David Mazeries. 1999. "A future – adaptable Password Scheme", Proceedings of the FREENIX Track: 1999 USENIX Annual Technical Conference, Monterey, California, USA, June 6–11,

[14] Thulasimani Lakshmanan and Madheswaran Muthusamy. 2012. "A Novel Secure Hash Algorithm for Public Key Digital Signature Schemes", The International Arab Journal of Information Technology, Vol. 9, No. 3, May, pp. 262-267.

[15] Janaka Deepakumara, Howard M. Heys and R. Venkatesan. "FPGA Implementation of MD5 Hash Algorithm".

[16] Tamimi R. 2013. "The application of web usage mining in E-commerce security", IEEE Xplore Digital Library , e-Commerce in Developing Countries: With Focus on e-Security (ECDC), 2013 7th Intenational Conference on 17-18 April, pp. 1-8.

[17] Web Security, Privacy & commerce - 2nd Edition, By Simson Garfinkel with Gene Spafford , O'Reilly SearchSecurity,http://searchsecurity.techtarget.com/definition/salt, Retrived 15th October, 2011.