A Cryptography Application using Salt Hash Technique

Pritesh N. Patel¹, Jigisha K. Patel² and Paresh V. Virparia³

¹Institute of Science and Technology for Advanced Studies and Research (ISTAR), Vallabh Vidyaganagar, Gujarat, India
²³Department of Computer Science, Sardar Patel University, Vallabh Vidyaganagar, Gujarat, India

ABSTRACT

World Wide Web has changed the life of people lives. More than half of the globe uses Internet in a day and communicate with each other to share data and information. But there is a chance of licking the personal information while transferring those data over Internet. There are cyber stalkers who stole the user password and used it for harassing and reading the personal information of the victims. In this paper we present the solution of storing password and sensitive data using salt hash technique. There are various methods for securing data, but salt may prove dominant among other in some applications. Salt is a publicly readable string stored in the database for each user id to protect against the dictionary attacks. Salt hash password is used to reduce the dictionary attacks.

Keywords: Salt, hashing, salt password, cryptography, SHA, web security

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 [4]. 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 [2] 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 [2] 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 inputted 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 [2]. 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.

2. OVERVIEW OF SALT

The use of the word "salt" is probably a reference to warfare in ancient times, when people would salt the wells or farmland to make it less hospitable. The Romans are sometimes supposed to have done this to Carthage in 146 BC. In the context of passwords, a "salted" password is harder to crack [3].

In password protection, salt is a random string of data used to modify a password hash and can be added to the hash to prevent a collision by uniquely identifying a user's password, even if another user in the system has selected the same password [1]. Salt can also be added to make it more difficult for an attacker to break into a system by using password hash-matching strategies because adding salt to a password hash prevents an attacker from testing known dictionary words across the entire system [1]. The intruder has to produce every possible salt value, which would take both a significant amount of time and space. The length and complexity of the salt value directly affects the time taken for a rainbow table attack [2]. If the salt is
longer and more complex, the greater time is required to success the attack by the intruder. For every new hash value we must use new salt and for intruder a new dictionary has to be generated for each stored password.

3. METHOD OF GENERATING SALT

In general salt is a random block of data or string or bytes. Computer languages provide different random number generation classes or functions are used to generate random numbers and bytes, but these classes and functions are not able to generate cryptographically secure random numbers. They are pseudo random number generators (PRNG) algorithms which are used by classes and functions in any language because the random value is completely dependent on data used to initiate the algorithm.

So cryptographically secure pseudo random number generator (CSPRNG) algorithm is to be required which must produce statically random number and they must hold up against attack. In some highly secure application special hardware is used to produce true random number from a physical process such as noise produced by microphone or nuclear decay of a radioactive source [2]. After generating true random number called as salt value, it must be combined with the plane text to produce salted hash. To produce salted hash – use salt value as prefix to the plane text or appending to the plane text before calculating hash.

Steps to generate Salt Hash password:
1. Get password
2. Generate Salt using trusted random functions/method
3. Append salt to original password
4. Generate Salt Hash password using appropriate hash function
5. Store salt and salt hash in the database

Using salt password in web application we can prevent SQL Injection attack because many user uses same password for multiple site for login the account. In this paper we present a mechanism to secure password hashes using salt in languages like PHP and .NET.

GENERATING SALT USING .NET

In .NET System.Security.Cryptography namespace provides cryptographic services, including secure encoding and decoding of data, as well as many other operations, such as hashing, random number generation, and message authentication [6]. RNGCryptoServiceProvider class is used to generate high quality random number which can be used in cryptography. Secure Hash Algorithm (SHA-256) which produces 256 bit digests.

Following code snippets is used to generate salt password in .NET:

```csharp
private static string CreateSalt(int size)
{
    //Generate a cryptographic random number.
    RNGCryptoServiceProvider rng = new RNGCryptoServiceProvider();
    byte[] buff = new byte[size];
    rng.GetBytes(buff);
    // Return a Base64 string representation
    // of the random number.
    return Convert.ToBase64String(buff);
}

private static string CreatePasswordHash(string pwd, string salt)
{
    string saltAndPwd = String.Concat(pwd, salt);
    // Use Hash Algorithm in 256 bits
    HashAlgorithm HashProvider = new SHA256Managed();
    byte[] hashsalt = HashProvider.ComputeHash(Encoding.UTF8.GetBytes(saltAndPwd));
    // Generate hash salt password
    saltAndPwd = Convert.ToBase64String(hashsalt);
    return saltAndPwd;
}
```
Above code shows the generation of salt and hash password with salt size 4 and expected output is generated.

4. GENERATING SALT USING PHP

PHP provide various functions for hashing and are publicly available in mhash library and can be found through link in PHP manuals [5]. Following code snippets is used to generate salt password in PHP:

```php
function HashPassword($password)
{
    $salt = bin2hex(mcrypt_create_iv(32, MCRYPT_DEV_URANDOM)); //get 256 random bits in hex
    $salthash = hash("sha256", $salt . $password); //prepend the salt hash
    return $salthash;
}
```

We can store the Password Hash and salt in the database in the user's account to decrypt the password.

5. CONCLUSION

Salt hash password prevents the attacker by mean of - the attacker will now have to recalculate their entire dictionary for every individual account they're attempting to crack. But salt can only help against prebuilt dictionaries, if intruder gets access to our system and if uses brute force attack, than salt will not provide must security.

References

[2.] ASP. NET Security, By. Barry Dorrans
[5.] PHP5 and MySQL Bible, By Tim Converse, Joyce Park with Clark Morgan, Wiley-India
[8.] Cryptography and Network Security By Behrouz A. Forouzan and Debdeep Mukhopadhyay
AUTHOR

Pritesh Patel received his Master degree in Computer Application from Sardar Patel University in 2008. He is currently working as an Assistant Professor with the Department of Computer Science, Institute of Science and Technology for Advanced Studies and Research, Vallabh Vidyanagar, Gujarat Technological University, Gujarat, India. His research interests are mainly focused on Mobile and Web Technology.

Jigisha Patel received her Master degree in Computer Application from Sardar Patel University in 2008. She is currently working as an Assistant Professor with the Department of Computer Science, Sardar Patel University, Gujarat, India. Her research interests are mainly focused on Natural Language Processing and Web Technology.

Dr. PareshVirparia is working as a Director and Professor in the Department of Computer Science, Sardar Patel University, Vallabh Vidyanagar. He completed his MCA in 1989 from Sardar Patel University and Ph. D. in 2002 from Sardar Patel University. He is recognized Ph.D. guide in Computer Science at Sardar Patel University, V V Nagar, KadiVishvaVidyalaya, Gandhinagar, Charotar University of Science & Technology, Changa, R K University, Rajkot and UkaTarsadia University, Bardoli. THREE research scholars have completed their Ph.D. (Computer science) under his guidance. Currently, EIGHT students are doing their Ph. D. under the guidance of him. Also, three students have completed their M.Phil. (Comp.Sc.) under his supervision. His publications include 31 papers in International Journal, 14 papers in National Journals and 36 papers in national conferences/seminars. His research interests include the areas of Computer Simulation & Modeling, Networking and IT enabled services.