Journal of Informatics
Electrical and Electronics Engineering, 2020,
Vol. 01, Iss. 02, S. No. 003, pp. 1-11
ISSN (Online): 2582-7006

Web Application and Penetration Testing

Saurabh Yadav[1], Pawan Singh[2]

[1, 2]Department of Computer Science and Engineering, Amity University Uttar Pradesh, Lucknow Campus, India
[1]saurabhyadav970@gmail.com, [2]pawansingh51279@gmail.com

How to cite this paper: S. Yadav, P. Singh (2020) Web Application and Penetration Testing. *Journal of Informatics Electrical and Electronics Engineering,* Vol. 01, Iss. 02, S. No. 3, pp. 1-11, 2020.

https://doi.org/10.54060/JIEEE/001.02.003

Received: 27/10/2020
Accepted: 14/11/2020
Published: 20/11/2020

Abstract

In the present scenario, the usage of internet is enormous and is escalating day by day. Internet facilities are employed in almost every field of work and people are becoming depending on it, with the increasing dependency on the internet, concern regarding information security has been increased. Because most of the work, e-commerce, chatting, payment of the bill, etc. are work through over the internet. That is why security is most important for any web site. Basically, such security concern is high in the field of organizations, institutions, and the financial sector. This paper aims to address the top most vulnerability concerns and how to overcome them. This paper addresses most of the popular vulnerabilities, which are amongst the top 10 according to OWASP and addresses the precautions to be taken to deal with these vulnerabilities. This paper provides a better understanding in a simple and easy way. When the entire world is behind new technologies and everything is moving towards the internet, the need for security increases. One has to be sure about the security of their website as well as the security and privacy of the end users. So, when the world is demanding for new technologies there will be an increase in demand for security testing. Every application or website is considered good only when it is secure and it can only be done by a web tester. This paper explores the vulnerabilities in a precise manner.

Keywords

Web application, Penetration testing, OWASP

1. Introduction

Web security is just not a thing you can actually ignore. Everyone makes applications and website and on the other hand the consumers want to enjoy the features but they care about their privacy and don't want their private or sensitive information to go in wrong hands. Sensitive information may contain their email, passwords or some bank related information that can really turn their worlds upside down. Sensitive information can be easily compromised if not properly managed.

Web penetration and application testing is a necessary procedure that every website or application must go through in order to ensure the privacy of their end customers. Web penetration and testing have some methods that check the sites or applications for vulnerabilities that can be exploited by someone who knows the hooks and crooks of how to steal information.

2. Literature Survey

Web penetration and application testing is a necessary procedure that every website or application must go through in order to ensure the privacy of their end customers. Sandeep Kumar, Renuka Mahajan, Naresh Kumar, Sunil Kumar Khatri [1] presented a paper on web application security which only dealt with three crucial vulnerabilities. Nuno Antunes , Marco Vieira [2] presented a paper on enhancing penetration testing with attack signatures and interface monitoring for the detection of injection vulnerabilities in web services which discussed about the vulnerability assessment and the SQL injection into the web environment. Prashant S. Shinde, Shrikant B. Ardhapurkar [3] conducted a study on vulnerability assessment and penetration testing which discussed about VAPT and few vulnerabilities mentioned in OWASP. Jose Fonseca, Marco Viera, Henrique Madeira [4] presented a paper on testing and comparing web vulnerabilities scanning tools for SQL injection and XSS attacks which dealt with the details on SQL injection and procedure of testing. Yarsolav Stefinko, Andrian Piskozub, Roman Banakh [5] presented a paper on manual and automated penetration testing which discussed about the approach of testing for vulnerabilities present. This study provides an overall of all the major vulnerabilities that can be present and helps to grasp an overview of how to avoid these vulnerabilities. The study is easy to grasp and revolves around the top vulnerabilities according to the OWASP.

3. Objective

This paper aims to put some light on few of the vulnerability concerns that are present in the web and to provide a few measure on how to avoid them. This paper identifies some vulnerability concerns in a systematic and easy to understand manner.

4. OWASP

Open Web Application Security Project is a non-profit global organization that focuses on providing information to help improve web application security. Open Web Application Security Project has developed an awareness document called the Open Source Application Security Project top ten.

4.1. OWASP Top 10-2013

A1- Injection
A2- Broken Authentication and Session Management
A3- Cross-site Scripting (XSS)
A4- Insecure Direct Object Reference
A5- Security disfigure
A6- Sensitive data exposure
A7- Missing function level access
A8- Using components with known vulnerabilities
A9- Using components with known vulnerabilities
A10- Non-validated redirects and forwards

4.2. OWASP Top 10-2017

A1- Injection
A2- Broken Authentication
A3- Sensitive Data Exposure

A4- XML external entities (XXE)
A5- Broken Access control
A6- Security Disfigure
A7- Cross site scripting (XSS)
A8- insecure deserialization
A9- Using components with known vulnerabilities

4.3. OWASP Methodology

1. **Planning:** Planning includes what are the ways to test the web or application and it includes the objectives of the test and how much time will it take and the time period for which the web or application will be under process.
2. **Intelligence Gathering:** Intelligence gathering includes a view of the application or web. A walk through of the web or application to know more about its working and to have a better understanding of the overall work.
3. **Vulnerability analysis:** Vulnerability analysis includes an analysis or walk through of all the vulnerabilities that have been encountered and analyzing all the possible effects and dangers of the vulnerabilities.
4. **Reporting:** Reporting is a systematic and well designed format of presenting the vulnerability with all the danger it causes and to help with its solution and how to prevent or be somewhat immune to it.

5. Report Making

Report making is an essential part of being a tester. You need the company or the owner of the website or application to understand how vulnerable their application or website is to a particular attack and to understand the criticality of the attack and how much data and user security is at stake.

The report should be clear and should be easy to understand and should deal with almost everything important that makes sense to the other person like the name of the vulnerability, the cause of the vulnerability, the kind of attack it is vulnerable to, what all data would be on stake and how to prevent that attack from taking place.

5.1. Report Format

A general report format: NAME OF VULNERABILITY

Table 1. Report format

Severity level	High/medium/low
Description	
Observation	
Ease of exploitation	Easy/Moderate/Difficult
Impact	
Recommendation	
Location	
Reference	

6. Vulnerabilities

Every business is under constant threat from a multitude of sources. From the biggest Fortune 500 companies down to the smallest of mom-and-pop stores, no business is 100% safe from an attack. The simple fact is that there are too many threats out there to effectively prevent them all.

But, malware isn't the only threat out there; there are many more cybersecurity threats and network vulnerabilities in existence that malicious actors can exploit to steal your company's data or cause harm. To put it in the most basic terms, a computer system vulnerability is a flaw or weakness in a system or network that could be exploited to cause damage, or allow an attacker to manipulate the system in some way.

6.1. Brute Force

Brute force is also known as the dictionary attack. This attack is used to bypass for example login pages to crack through the password. The passwords are tried to bypass through using 0-9, a-z etc. For brute force, a combination of passwords can be saved in notepad which can be used to bypass the password.

Different intruder attack types:

1. Sniper
2. Battering ram
3. Pitchfork
4. Cluster bomb

a) **Sniper:** The sniper attack replaces position of one field at a time, the sniper attack uses one set of payload, regardless of the number of positions and the sniper attack uses the original values for all positions that have no payload

b) **Cluster bomb:** The cluster bomb attacks two lists firstly, the lists runs against every word to bypass the password and secondly, tries all possible combinations.

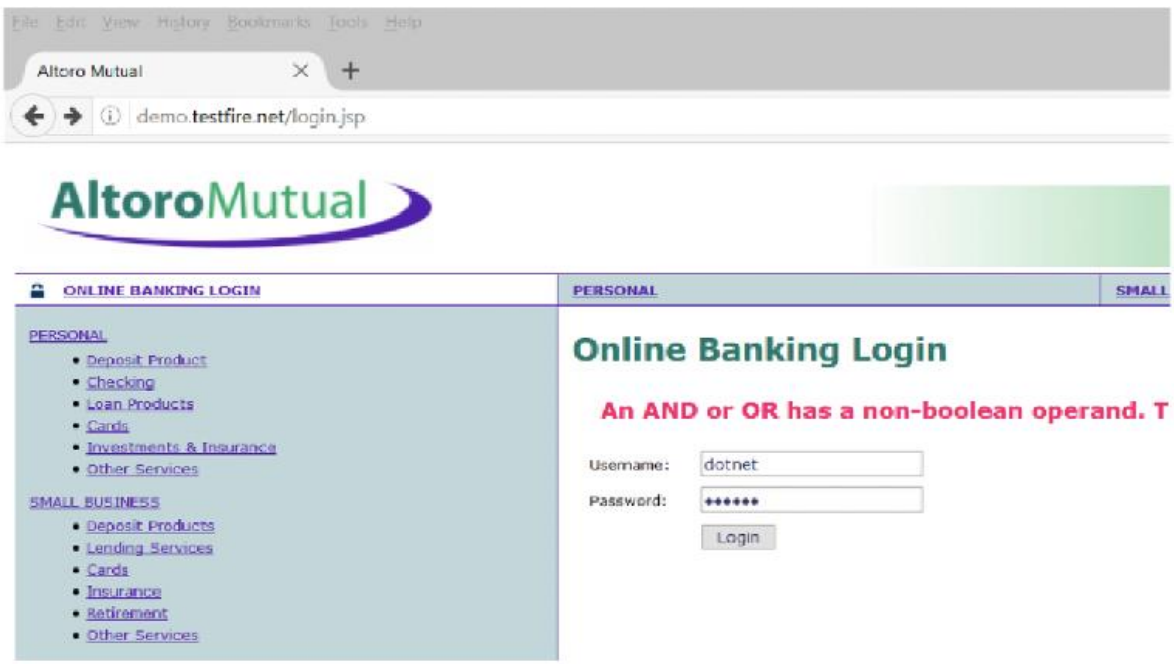

Figure 1. Brute Force on Altoro Mutual

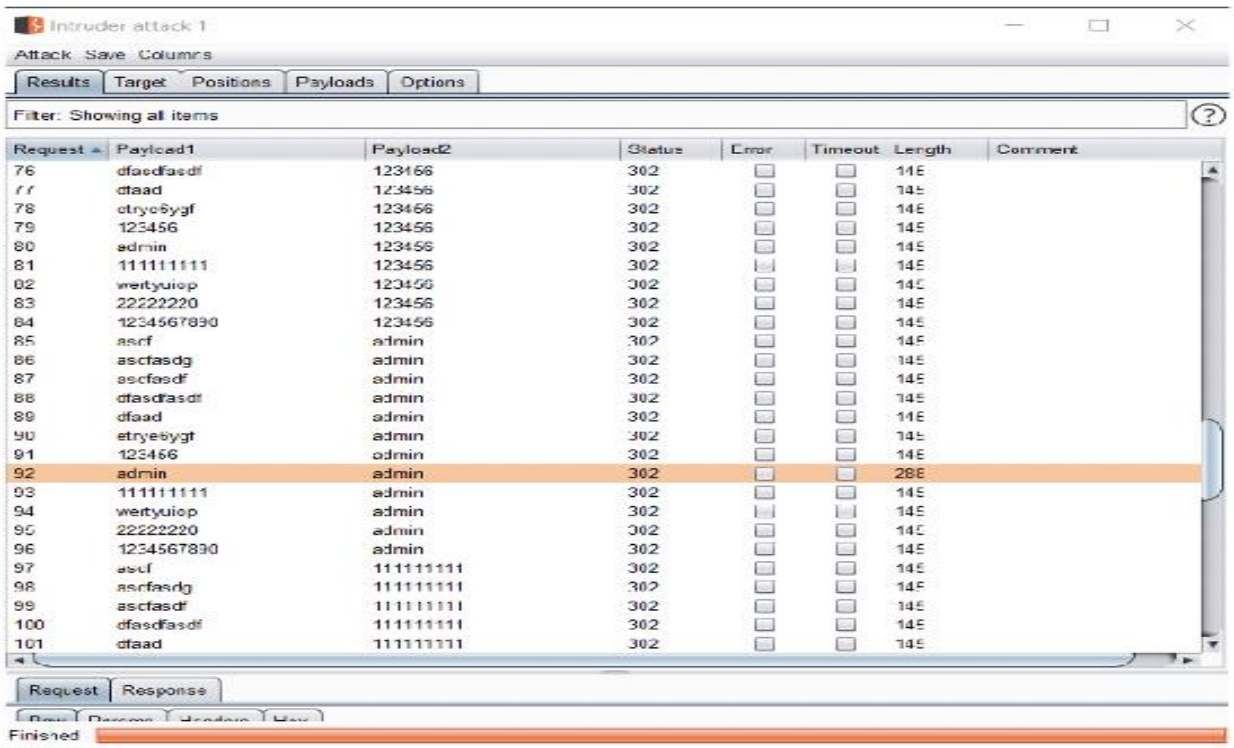

Figure 2. Combination used on the page

Figure 3 demonstrates how to prevent brute force attack. The most simple way to prevent a brute force attack is to limit the number of incorrect passwords, incorporate the use of captcha, two factor authentication. The brute force attack is only possible if there are unlimited numbers of failed logins allowed. There is no other way to identify the user except the regular password validation method. Figure 4 on the other hand demonstrates the simple use of password validation with unlimited number of failed login attempts.

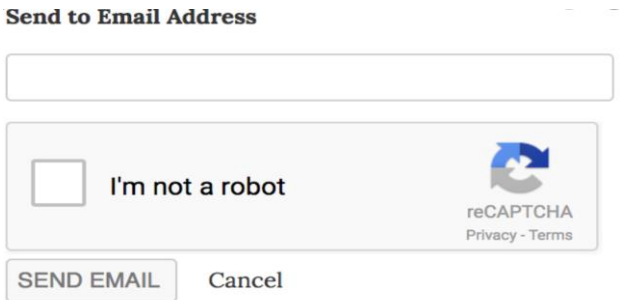

Figure 3. Captcha to prevent brute force attack

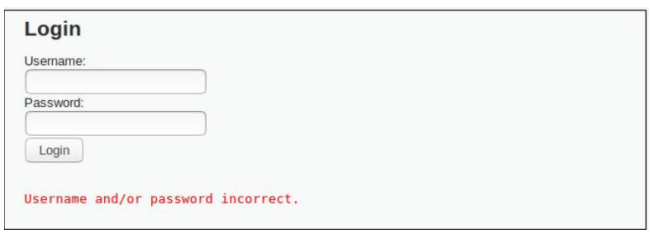

Figure 4. Brute force vulnerable

6.2. Credentials Transported over Unencrypted channel

1. **HTTP:** HTTP keeps the sensitive information in plain text which compromises the security and anyone can intercept and read the data.

2. **HTTPs:** HTTPs keeps the sensitive information in encrypted form while transmitting it to server or the client which makes it difficult for the hacker to decode the sensitive data and thus boosts the security.

This attack is also known as the man in the middle attack. The man in the middle works as when the hacker intercepts the information transported over the HTTP channel which is not encrypted.

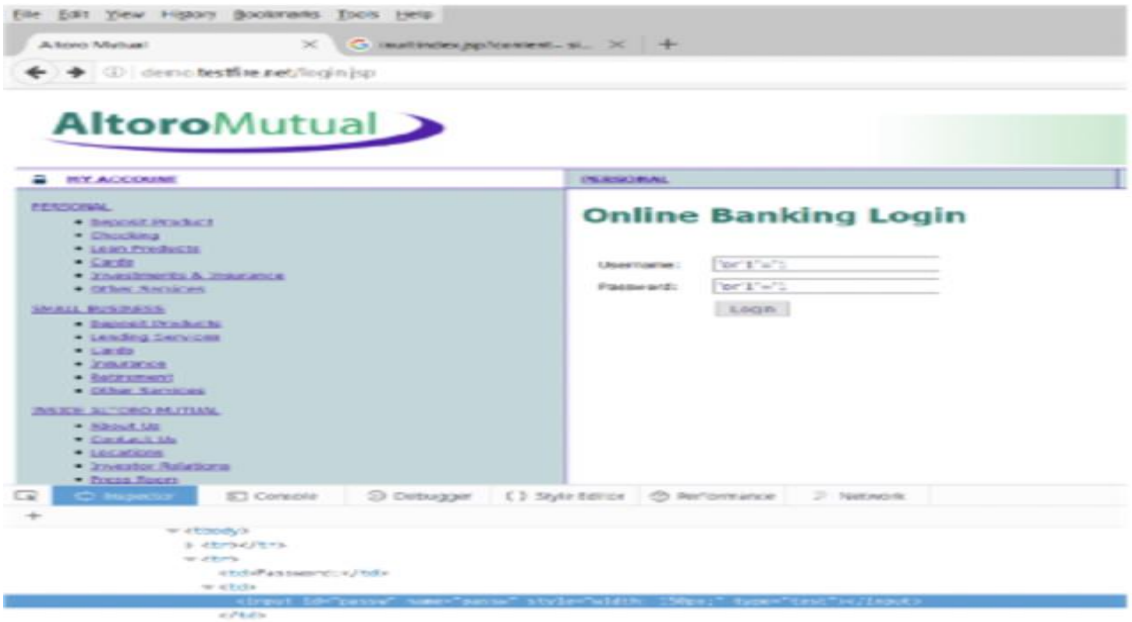

Figure 5. Credential transported over an unencrypted channel on Altoro Mutual

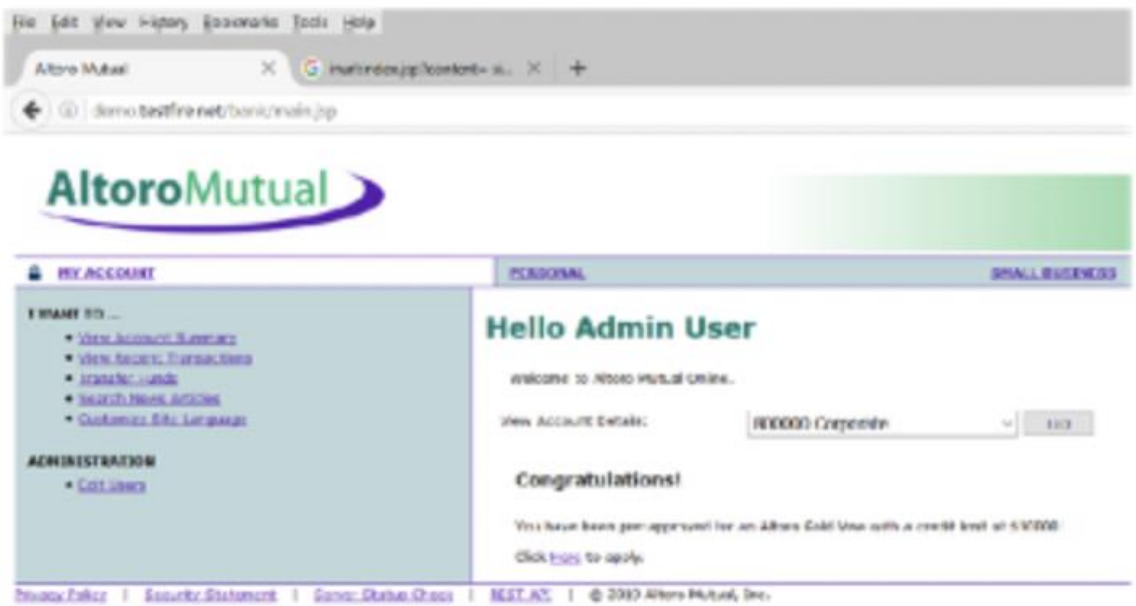

Figure 6. Sensitive information visible in plain text

Figure 7 demonstrates the password that is passed over the HTTPs channel which encrypts the password so that any sensitive information is not clearly visible which makes it easily accessible to the hackers. Figure 8 on the other hand demonstrates the use of HTTP channel which is not secure and send the sensitive information as a plain text which allows the hackers to easily access the information and misuse it.

Figure 7. Encrypted password over HTTPs

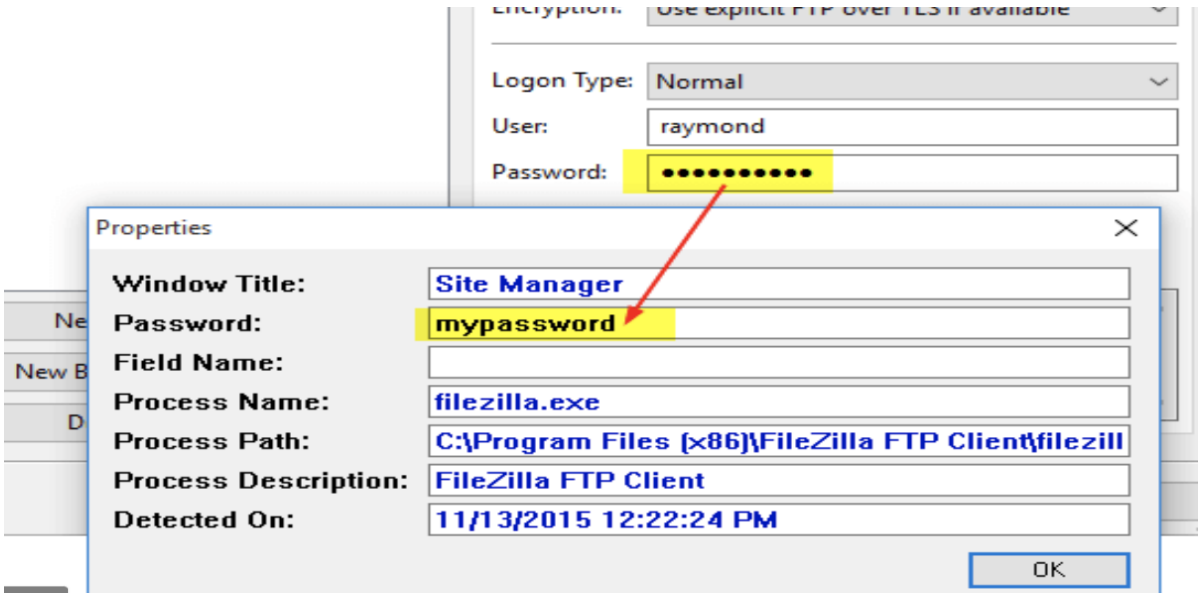

Figure 8. Password in plain text

6.3. Broken Authentication and Session Management

Unsecured login and logout processing can allow attackers to compromise authentication token, password and keys. If the login and logouts are not secured the sensitive information of the user can be compromised.

Applications vulnerable by this when:

1. Permits default, weak or common passwords
2. Weak security questions
3. Exposes session ids in the URL
4. Permits automated attacks such as credential stuffing.

Testing for default credentials: Whenever a login page is considered the developer might have allowed logging in using default credentials which lets you go in the home page after logging in without actually having a registered account or being a registered user. A login page can have various default credentials for example the username and password both can be admin or user.

Prevention for broken authentication:

1. Password should not be too short
2. Password should not be easy to guess and should not be common
3. The username and password should not be allowed to be kept the same
4. Protection against brute force login

6.4. Sensitive Data Exposure

Sensitive data exposure vulnerabilities can occur when a web application does not properly protect sensitive information from being disclosed to attackers. The sensitive data can be then used for the unsuitable things that the user doesn't want.

Prevention:

1. The pages should be HTTPs secured.
2. Default credentials should not be allowed.
3. Secure socket layer and transport layer security.
4. Brute force should not be allowed

6.5. Cross Site Scripting (XSS)

Cross site scripting happens whenever an application takes the data that is not trusted and sends it to the client (browser) without validation.

Prevention:

1. Untrusted data should not be allowed unless it is required at a certain point
2. HTML escape: The web framework should have HTML escaping for characters
3. Attribute escape: untrusted data input in fields like name, value etc.

Figure 9 demonstrates how an attacker sends miscellaneous scripts to the user disguised as something informative, and as soon as the user clicks on the link the script in the background passes all the data to the hacker and not to the original website. if the web page allows the users to post programs and allows <scripts> anything that is posted in scripts will be considered as JavaScript and treated normally. Once the users opens such scripts, user becomes the victim of the script.

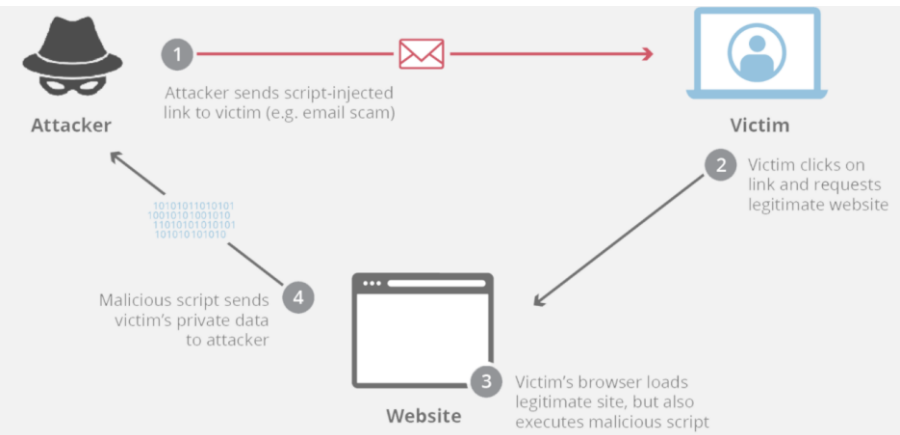

Figure 9. Cross site scripting

6.6. SQL Injection

A website is vulnerable to SQL injection if we can use the commands the SQL commands to make changes in the database. The syntax used is: 'or_'1'='1. This can be used in both the username and passwords fields to bypass the login pages.

Prevention:

1. All the input by user should be validated via function
2. No queries should be formed by using the input provided by the user
3. Don't have any additional database functionality which is not needed.
4. Do not disclose any information related to errors than required.
5. Firewall

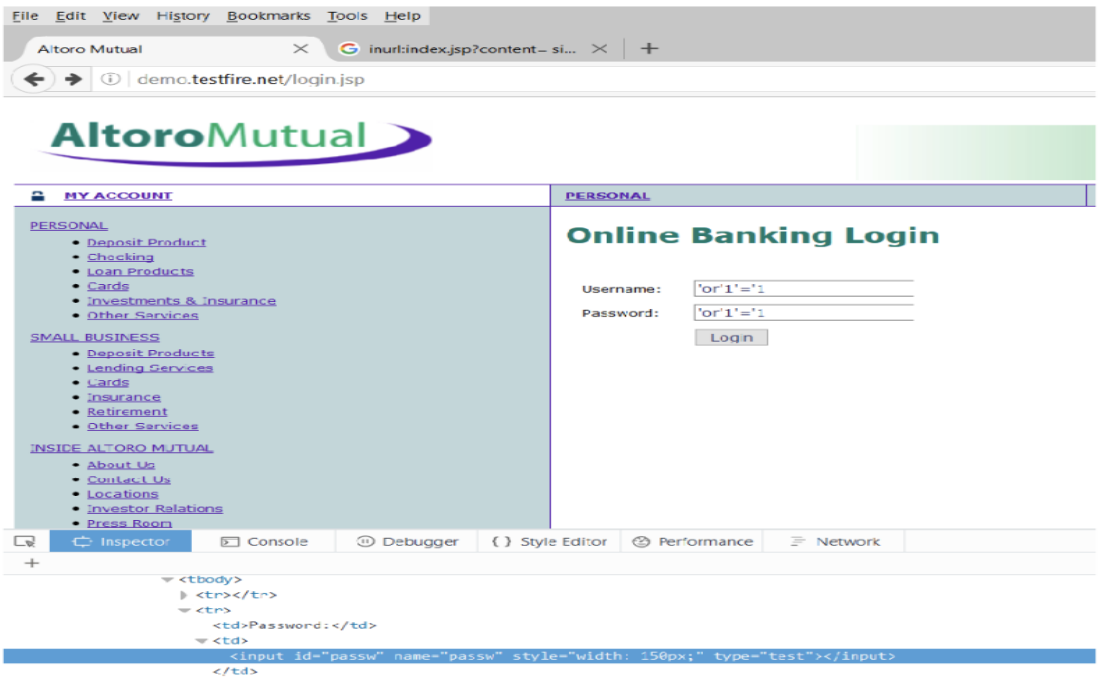

Figure 10. SQL Injection on Altoro Mutual

6.7. POST SQL Injection

Post sql injection can be used to find out the database from the username and password fields. Post sql injection attack can be made using sql map master by using a series of commands that are used to exploit the database.

The commands can be the used to then access information in the database and use them for any purpose intended.

Prevention:

1. All the input by user should be validated via function
2. No queries should be formed by using the input provided by the user
3. Don't have any additional database functionality which is not needed.
4. Do not disclose any information related to errors than required.
5. Firewall

Figure 11 demonstrates the queries allowed by the website. post SQL injection is posting SQL queries in the database to get information of the users or any sensitive information that should not be known otherwise to the people outside.

The figure below shows how the id tuple is selected to get all the information from the id tuple to extracted.

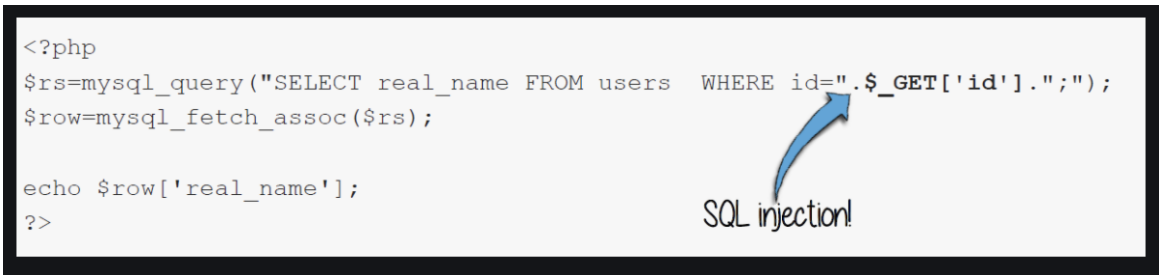

Figure 11. Post SQL queries

6.8. Clickjacking

Clickjacking attack is mostly used on login or registration pages. The websites that allow setting of iframe are vulnerable to clickjacking.

The iframe can be added to the page using HTML. It can be used to display a fake page to the user which gets him into thinking that it is the original page and the fake page then captures the information of the user. Clickjacking can be used then to spread any type of viruses, stealing private information.

Prevention:

1. The webpage should not allow setting of iframe in the source code of the page.
2. We can use the X-Frame options response to defend against clickjacking.
3. The exploitation of the XSS filters.

Figure 12. Clickjacking on Altoro Mutual

7. Results

This pool of study includes a combined study of web penetration and application testing to help grasp an overall view. This also provides a precise and informative view of the vulnerabilities present in the web with certain measures to avoid them. The types of vulnerabilities present in this domain and mentioned int OWASP.

8. Conclusion

This review paper provides an overall view of the vulnerabilities present as well as the necessary precautions to be taken. This review paper has covered the top vulnerabilities in accordance with the vulnerabilities mentioned by OWASP. The review paper addresses all the vulnerabilities in a simple way which is easy to understand. The result of the study can help to grasp an overview of the vulnerabilities present in the web and helps to avoid them.

Acknowledgements

When I look at the precious knowledge, I have gained till date in my B. Tech journey, I realize how much time and effort spent toward the completion of this work; not only by me but also by key individuals whom I feel very indebted to. I gratefully acknowledge the support of my supervisor Prof. Dr. Pawan Singh. I would not imagine completing this work without all his advices. I acknowledge the time he spent in our weekly meetings, even with his busy schedule and other commitments. I am deeply indebted to him for his patience. He assisted in all aspects of this work from discussing new ideas to writing and completing this project. I would like to thank Amity University, Lucknow Campus for providing me a wonderful platform for this work.

References

[1] S. Kumar, R. Mahajan, N. Kumar, et al, “A study on web application security and detecting security vulnerabilities,” in 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) pp. 451-455. Sept. 2017.

[2] N. Antunes and M. Vieira, “Enhancing penetration testing with attack signatures and interface monitoring for the detection of injection vulnerabilities in web services,” in IEEE International Conference on Services Computing, pp. 104-111, July 2011.

[3] P. S. Shinde and S. B. Ardhapurkar, “Cyber security analysis using vulnerability assessment and penetration testing,” in World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), pp. 1-5, March 2016,

[4] J. Fonseca, M. Vieira, and H. Madeira, “Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks,” in 13th Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 365-372, Dec. 2007.

[5] Y. Stefinko, A. Piskozub, and R. Banakh, “Manual and automated penetration testing. Benefits and drawbacks. Modern tendency,” in 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), pp.488-491, Feb. 2016.