ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

ENHANCED METHODOLOGY TO IMPROVE INFORMATION SECURITY SYSTEM USING THE COMBINATION OF ETHICAL HACKING AND CAPTCHA TECHNOLOGY

Arshad Kamal*, Mohammad Danish
*M Tech Scholar Department of Computer Science and Engineering,Al-Falah University, Faridabad, Haryana-India-121005.
Assistant Professor Department of Computer Science and Engineering,
Al-Falah University, Faridabad, Haryana-India-121005

ABSTRACT

Information Technology has become an integrated and fundamental part of the business, and with the increased automation of business processes in all the industries and sectors, the Information Technology users are not limited to only IT and Software Company. With the increased usage and penetration of Internet and application access on handheld mobile devices, the IT user community has increased many times. In fact, for every interaction or transaction what happens today, some way or the other IT system has got a place there. The usage is being tracked and captured by organizations to do analytics to find the opportunities for growth.

The business process automation requires organizations process and data as the input to the IT systems. The data in the transaction has become very important, and it contains private data and should be kept secure understanding the criticality of data privacy. In fact, the entire process must meet the basic requirements of security i.e. Availability, Confidentiality, and Integrity.

KEYWORDS: Information security; Privacy; Ethical Hacking; Requirement Engineering; IT Project Management; Risk Management; SQL Injection; CAPTCHA;.

INTRODUCTION

We see a high similarity in the human need model defined by Maslow to the current requirement on the information security in the internet age. Maslow defines Basic needs like Food, Clothing, and Shelter etc. as the primary need and is the first step in the Maslow model hierarchy. Security and Safety comes second. Similarly, we see the trend in the Information Technology and Internet evolution. As the internet users has increased and it has found place in all the activities we do. The basic fundamentals of information security i.e. Availability, Confidentiality and Integrity becomes very important.

The India Risk Survey 2015 [01] analyses and quantifies 'potentially destructive' risks to business enterprises in the country. It provides a referral to understand the complexity of these new risks across the spectrum of stakeholders, i.e., policymakers, corporate and members of the civil society [01].

Information security is a growing concern keeping in mind the technological advancements and the tendency of corporate houses to create more and more intellectual property and competitive strategies [02-03].

The numbers of security vulnerabilities that are being found today are much higher in applications than in operating systems [02]. This means that the attacks aimed at web applications are exploiting vulnerabilities at the application level and not at the transport or network level like common attacks from the past. At the same time, quantity and impact of security vulnerabilities in such applications has grown as well. Many transactions are performed online with various kinds of web applications.

This work is an effort to highlight and reinforce the criticality of the requirement of being security driven in all the tasks we do, in all the moves we take, in all the transactions we do, in all the interactions we have, through the summarization of the previous research works in the field of information security. The work should be useful for the industry and academic professionals i.e. developers, testers, analyst, project managers, students, technical trainers etc. Ethical hacking is a growing practice within risk management area in the organizations, to proactively find the vulnerabilities in the system using the tools and techniques as used by hackers.

The work has been demonstrated through the use of SQL Injection- as an example- to highlight the security gaps in coding practices and CAPTCHA- as an example- to highlight the gaps in third party services integration methods, resulting in vulnerabilities.

SQL Injection is one of the top 10 techniques used by hackers to exploit the websites to take the confidential data for different purpose. The solution proposed by experts in designing an application is to keep the minimum target surface area and control procedures.

CAPTCHA is one of the techniques used by the application developers to ensure that the website services are not put down by automated scripts by malicious users. CAPTCHA is based on Turing Test and here the system differentiates between a human interface and machine interface.

CAPTCHA has been very successful and it has found its application and uses in almost all the websites we use. There have been many research activities to increase the complexity and strength of CAPTCHA. In the current time, highly complex and strong CAPTCHA is available as a third party service which can be integrated in application.

Objectives of Cyber Policy 2013

- Establish a secure cyber eco-system in the country.
- Framework for design of security policies and compliance to global security standards and best practices.
- Strengthen the Regulatory Framework.
- Enhance & create National & Sectorial level 24x7 mechanisms for obtaining strategic information regarding threats.
- Improve visibility of integrity of ICT products and services.
- Provide fiscal benefit to businesses for adoption of standard security practices and processes.
- Protection of information during process, handling, storage & transit.
- Effective prevention, investigation and prosecution of cybercrime and enhancement of law enforcement capabilities. [01]

RELATED WORK

OWASP Top 10

The primary aim of the OWASP Top 10 is to educate developers, designers, architects, managers, and organizations about the consequences of the most important web application security weaknesses. The Top 10 provides basic techniques to protect against these high risk problem areas – and also provides guidance on where to go from here.
List of the Top 10 risk areas with their comparative ranking in current and previous studies [29]:

1. Injection
2. Broken Authentication and Session Management
3. Cross-Site Scripting (XSS)
4. Insecure Direct Object References
5. Security Misconfiguration
6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross-Site Request Forgery (CSRF)
9. Using Components with Known Vulnerabilities
10. Unvalidated Redirects and Forwards

OWASP Top 10 – 2010 (Previous)	OWASP Top 10 – 2013 (New)
A1 – Injection	A1 – Injection
A3 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A2 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References	A4 – Insecure Direct Object References
A6 – Security Misconfiguration	A5 – Security Misconfiguration
A7 – Insecure Cryptographic Storage – Merged with A9 →	A6 – Sensitive Data Exposure
A8 – Failure to Restrict URL Access – Broadened into →	A7 – Missing Function Level Access Control
A5 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
<buried in A6: Security Misconfiguration>	A9 – Using Known Vulnerable Components
A10 – Unvalidated Redirects and Forwards	A10 – Unvalidated Redirects and Forwards
A9 – Insufficient Transport Layer Protection	Merged with 2010-A7 into new 2013-A6

Vulnerability analysis

Vulnerability analysis, also known as vulnerability assessment, ―is a process that defines, identifies, and classifies the security holes (vulnerabilities) in a computer, network, or communications infrastructure‖. In addition, vulnerability analysis can forecast the effectiveness of proposed countermeasures and evaluate their actual effectiveness after they are put into use. It consists of following steps [02]:

- Define and classify target system.
- Assign relative levels of importance to the target system resources.
- Identify potential threats to each resource.
- Develop or setup a method to deal with the most serious potential problems.
- Define and implement procedures to minimize the consequences if the attack for the target system resource.

System Threats

System threats refer to misuse of system services and network connections to put user in trouble and can be used to launch program threats on a complete network called as program attack. Some of the well-known system threats are Worm, Port Scanning, and Denial of Service [07].

Program Threats

If a user program made Operating system's processes and kernel process do malicious tasks then it is known as Program Threats. Common examples of program threat is a program installed in a computer which can store and send user credentials via network to some hacker. Some of the well-known program threats are Trojan horse, Trap Door, Logic Bomb, and Virus [07].

Objectives of Security requirements

1. Ensure that users and client applications are identified and that their identities are properly verified.
2. Ensure that users and client applications can only access data and services for which they have been properly authorized.
3. Detect attempted intrusions by unauthorized persons and client applications.
4. Ensure that unauthorized malicious programs (e.g., viruses) do not infect the application or component.
5. Ensure that communications and data are not intentionally corrupted.
6. Ensure that parties to interactions with the application or component cannot later repudiate those interactions.
7. Ensure that confidential communications and data are kept private.
8. Enable security personnel to audit the status and usage of the security mechanisms.
9. Ensure that applications and centres survive attack, possibly in degraded mode.
10. Ensure that centers and their components and personnel are protected against destruction, damage, theft, or surreptitious replacement (e.g., due to vandalism, sabotage, or terrorism).
11. Ensure that system maintenance does not unintentionally disrupt the security mechanisms of the application, component, or center. [03]

Kinds of security requirements to meet the Security objectives

1. Identification Requirements,
2. Authentication Requirements,
3. Authorization Requirements,
4. Immunity Requirements,
5. Integrity Requirements,

6. Intrusion Detection
7. Requirements,
8. Nonrepudiation Requirements,
9. Privacy Requirements,
10. Security Auditing Requirements,
11. Survivability Requirements,
12. Physical Protection Requirements,
13. System Maintenance Security Requirements.[03]

Software Development Security

Software is usually developed for functionality, not security. To get the best of both worlds, security and functionality would have to be designed and integrated into the individual phases of the development life cycle.

- Software should be developed with potential risks in mind, and many types of threats models and risk analyses should be invoked at different stage of development.
- The goals are to reduce vulnerabilities and the possibility of system compromise.
- The controls can be preventive, detective, or corrective.
- If an application is purely proprietary and will run only in closed trusted environments, fewer security controls may be needed than those required for applications that can connect business over the internet and provide financial transactions [04].

Usual Trend of dealing with Security

1. Buggy software is released to the market to beat the competition.
2. Hackers find new vulnerabilities and weakness in the software.
3. Websites post these vulnerabilities and how to exploit them.
4. Vendors develop and releases patch to fix vulnerabilities.
5. The new patch goes on the stack of software patches that all network administrator need to test and install [04].

System development Life Cycle

A system has its own development life cycle, which is made up of the following phases:

- Initiation: Need for a system is defined.
- Acquisition/Development: New system is either created or developed
- Implementation: New system is installed into the production environment
- Operation/Maintenance: System is used and cared for.
- Disposal: System is removed from the production environment [04]

Software Development Life Cycle

The life cycle of software development deals with putting repeatable and predictable processes in place that help ensure functionality, cost, quality, and delivery schedule requirements are met.[04]

WEB APPLICATION SECURITY

There are three major areas of Computer, or Cyber, Security.

- Network or Physical Level
- Operating System Level
- Application Level [19]

WEB VULNERABILITIES

There are ten broad categories of security flaws or vulnerabilities which are common to almost all web based applications. These vulnerabilities can be lumped into four major areas of concern:

- Authentication weaknesses
- Code weaknesses
- Cryptographic weaknesses
- Administration weaknesses

There are four basic approaches to testing web applications for security flaws:

- Manual penetration testing
- Automated vulnerability scanning
- Manual code review
- Automated code analysis

The most common approach to finding vulnerabilities is to analyse the running application. The two techniques are "vulnerability scanning" (using tools and signature databases) and "penetration testing" (custom testing by experts). The most cost-effective approach to application security is a "combined" or "integrated" approach. The assessor should be encouraged to use the most appropriate tool to find problems in the most cost-effective manner. For example, an assessor may notice a potential vulnerability during a penetration test, automatically scan the code for possible instances of the problem, and then confirm using code review [19].

Ethical Hacking

Ethical hacking is an assessment to test and check an information technology environment for possible weak links and vulnerabilities. Ethical hacking describes the process of hacking a network in an ethical way, therefore with good intentions.

The working of an ethical hacker involves the under mentioned steps:

1. Obeying the Ethical Hacking Commandments:
2. Working ethically:
3. Respecting Privacy:
4. Not crashing your systems:
5. Executing the plan:

The overall hacking methodology consists of certain steps which are as follows: Reconnaissance, Probe and Attack, Listening, First Access, Advancement, Stealth, Takeover, Cleanup. In the field of Ethical hacking, the approach is to proactively find the vulnerability in the application using the tools and techniques deployed by hackers, to have early fix and reduce risk. The vulnerabilities can be of many types, however we have focused on SQL Injection understanding its criticality based on OWASP 2013 Top 10 report. [08, 09]

SQL INJECTION: PREVENTION TECHNIQUES

Protection products which can be used to prevent SQL Injections in online applications, in the form of Web Application Firewalls (WAFs), HIPS solutions and Database Extrusion Protection (DBEP) systems.

Eight recommended mitigation techniques that can prove very effective against SQL Injections in online applications (e.g. web sites) and other environments [28].

1. Always apply the "Least Privilege" rule: set up low-privileged database accounts for applications that access the DBMS.
2. Always validate user-supplied data – as well as any data obtained from a potentially unsafe source – on the server side. Client-side input validation can be useful (mostly for user experience) but cannot in any case be relied upon.
3. Do not return SQL error messages to users as they contain information useful for attackers, such as the query, details about the targeted tables or even their content. This can be easily prevented in Java using exception handling: simply catch all SQLExceptions.
4. Enforce data types for all inputs. Type-specifying regular expressions can be used to validate the data. Types can also be enforced via pre-compiled statements with binded variables (e.g., JDBC's PreparedStatement interface. Also check boundaries to prevent buffer overflows and truncation errors which could lead to a crash of the DBMS.
5. Encode text input fields likely to contain problematic characters into an alphanumeric version using a two-way function such as Base64.
6. Filter all input data via a 2-step process. First, apply white-list filtering at user input collection (e.g., web forms): allow only field-relevant characters, string formats and data types; restrict string length. Then, black-list filtering or escaping should be applied in the data access layer before generating SQL queries: escape SQL metacharacters and keywords/operators.
7. Validate dynamically-generated database object names (e.g. table names) with strict white-list filtering. For instance with Oracle, allow only alphanumeric characters, '_', '$' and '#'.

8. Avoid quoted/delimited identifiers as they significantly complicate all whitelisting, black-listing and escaping efforts [28].

CURRENT IMPLEMENTATION AND FUTURE WORK

During the study on Cyber Security Risk and Ethical hacking practices, we observe that SQL Injection is a known vulnerability for many years, and still it's ranked as number one risk. Some of the observations on SQL injections are:

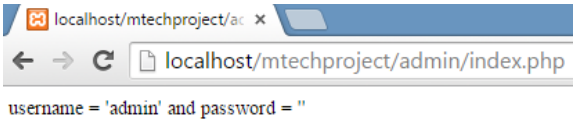
- Previous research regarding SQL-injections has focused on procedures to automate the search for vulnerabilities using different tools, and different coding techniques for defence.
- There is also research that describes in depth how the code in different programming languages should be constructed, and how to write code in order to prevent or stop SQL-injections and other types of attacks.
- The objective of the current work is to highlight the SQL Injection vulnerability and the risk associated with it.
- There have been earlier research and guidelines to suggest on the SQL Injection and Prevention techniques, however the best practices are not in common use. The objective of the current work is to highlight on the best practice and prevention techniques to protect the application from SQL injections.
- We have highlighted the need of security at each layer of software development life cycle based on available research works.
- We have demonstrated SQL Injection techniques to show the weakness in a PHP-MySQL application, where SQL statements are created dynamically.

Example to demonstrate SQL Injection in PHP-MySQL application

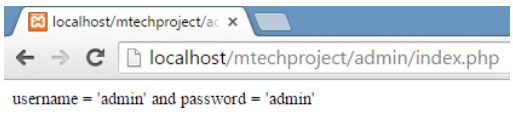
```
$condition = "username = '".$_POST['adminUserName']."' and password = '".$_POST['adminPassword']."'";
echo  $condition;
```

Prompt to Enter User Name and Password

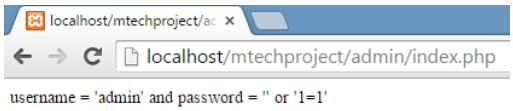
Print SQL string to demonstrate SQL injection techniques
SQL string when only user name is given as input and no password.



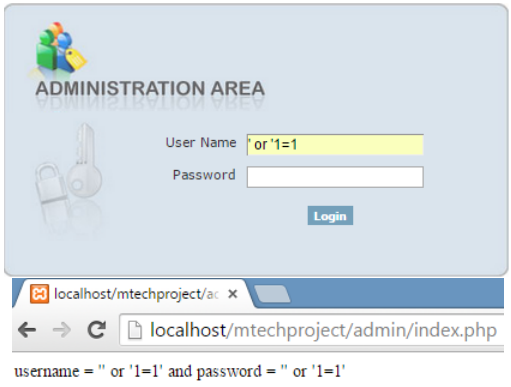
SQL string when both user name and password is given as input.



Login happens successfully and Control Panel screen opens.
SQL string when SQL injection 1=1 is done in password



SQL string when SQL injection 1=1 is done both in user name and password



In the both scenarios above, the SQL string formed is a valid SQL, and login happens successfully.

CONCLUSION

Information security is not for only the Information Technology Professional. In fact, information security is end user driven. As more and more services are moving to internet and web applications user are increasing, the security awareness has to increase. Most faults and weaknesses can be prevented by carefully and consistently implementing security during the application's requirements, architecture and design phases.

In the End, we would like to conclude that security is not in the isolation. It has to be multi-layered. Security Awareness is the key. Every user needs to become more conscious towards security. Application development and maintenance needs to keep the right balance between functionality and security requirements.

REFERENCES

[1] India Risk Survey 2015 Pinkerton and Federation of Indian Chambers of Commerce and Industry (FICCI)
[2] Modern Approach for WEB Applications Vulnerability Analysis, Rami M. F. Jnena, The Islamic University of Gaza, Deanery of Graduate Studies, Faculty of Engineering, Computer Engineering Department
[3] Engineering Security Requirements, JOURNAL OF OBJECT TECHNOLOGY, Donald G. Firesmith, Firesmith Consulting, U.S.A. Published by ETH Zurich, Chair of Software Engineering ©JOT, 2003 Vol. 2, No. 1, January-February 2003
[4] Software Development Security, Reference: Chapter 10, CISSP Exam Guide, Shon Harris
[5] Improving Security and Privacy of Integrated Web Applications, A Dissertation Presented to the Faculty of the School of Engineering and Applied Science University of Virginia, In Partial Fulfillment of the requirements for the Degree Doctor of Philosophy (Computer Engineering) by Yuchen Zhou May 2015
[6] SQL-Injections A wake-up call for developers, A study about a major threat and issue for companies and organizations worldwide, Martin Flodström & Oskar Vikholm , Uppsala University Department of Informatics and Media Bachelor Thesis Spring 2013
[7] http://www.tutorialspoint.com/operating_system/os_se curity.htm
[8] A Closer Look at Ethical Hacking and Hackers, Marilyn Leathers, East Carolina University ICTN 6865
[9] ETHICAL HACKING: A TECHNIQUE TO ENHANCE INFORMATION SECURITY, Gurpreet K. Juneja1, Lecturer, Department of Computer Science & Engineering, Guru Nanak Dev, Engineering College, Ludhiana, India, International Journal of Innovative Research in Science, Engineering and Technology
[10] WHITEPAPER: NUCAPTCHA & TRADITIONALCAPTCHA, www.nucaptcha.com
[11] How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation, Elie Bursztein, Steven Bethard, Celine Fabry, John C. Mitchell, Dan Jurafsky, elie@cs.stanford.edu, bethard@stanford.edu, mitchell@cs.stanford.edu, jurafsky@stanford.edu, Stanford University
[12] CAPTCHAS: SURVEY OF EXISTING TECHNIQUES AND A NEW APPROACH, Vaishakh B. N., Harish G., Department of Computer Science and Engineering R.V College Of Engineering, Bangalore

[13] CAPTCHA Smuggling: Hijacking Web Browsing Sessions to Create CAPTCHA Farms, Manuel Egele, Technical University Vienna, Austria, Leyla Bilge, Engin Kirda, Christopher Kruegel
[14] Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI, Luis von Ahn
[15] Oracle Tutorial on Defending against SQL injection
[16] sql-injection-attacks-and-their-prevention-practices (securityresearch.in)
[17] WHITEPAPER: NUCAPTCHA & TRADITIONALCAPTCHA www.nucaptcha.com
[18] A CAPTCHA Implementation Based on 3D Animation,Jing-Song Cui, Jing-Ting Mei, Xia Wang, Da Zhang, Wu-Zhou Zhang, College of Computer Science, Wuhan University
[19] NASS Whitepaper Web Application Security, ManTech International Corporation
[20] McAfee White Paper on Bypassing CAPTCHAs by Impersonating CAPTCHA Providers, Gursev Singh Kalra
[21] McAfee White Paper on Attacking CAPTCHAs for Fun and Profit, Gursev Singh Kalra
[22] Text-based CAPTCHA Strengths and Weaknesses, Elie Bursztein, Stanford University
[23] Understanding Captcha: Text and Audio Based Captcha with its Applications, International Journal of Advanced Research in Computer Science and Software Engineering, Sarika Choudhary, Ritika Saroha
[24] Image Recognition CAPTCHAs International Information Security Conference (ISC 2004), Springer, Monica Chew and J. D. Tygar, UC Berkeley
[25] SURVEY ON CAPTCHA SYSTEMS Journal of Global Research in Computer Science, Rizwan Ur Rahman Maulana Azad National Institute of Technology, Bhopal, M.P, India
[26] Survey of Different Types of CAPTCHA, International Journal of Computer Science and Information Technologies, Ved Prakash Singh, Preet Pal, School of Computer Science, Lovely Professional University
[27] The Robustness of Google CAPTCHAs, School of Computer Science,Newcastle University, UK, Ahmad S El Ahmad, Jeff Yan, Mohamad Tayara
[28] Preventing SQL Injections in Online Applications: Study, Recommendations and Java Solution Prototype Based on the SQL DOM, Etienne Janot, Pavol Zavarsky, Concordia University College of Alberta, Department of Information Systems Security, 7128 Ada Boulevard, Edmonton, AB, T5B 4E4, Canada, OWASP Application Security Conference 2008
[29] https://www.owasp.org