International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 11, ISSN No. 2455-2143, Pages 8-13
Published Online March 2020 in IJEAST (http://www.ijeast.com)

EVALUATION OF WEB APPLICATION SECURITY IN AFGHANISTAN

Abdullah Hamidi (MSc.)
Database / Information Systems Department
Computer Science Faculty, Herat University
Herat, Afghanistan

Mohammad Mustafa Naier (MSc.)
Computer Engineering Department
Computer Science Faculty, Polytechnic University
Kabul, Afghanistan

Mohammad Rafi Bahez (MSc.)
Software Engineering Department
Faculty of Computer Science, Kabul University
Kabul, Afghanistan

***Abstract*— Security is an important concern to be considered in web application development. Some web applications are developed and maintained by the developers yearly and published on the World Wide Web. Due to the nature of web applications, they are located on the internet and accessed by different people. Sometimes, it happens that a group of people wants to access information stored by the web applications, remove their data or access their software systems without having authorized access. Many students are graduated from computer science faculties in Afghanistan yearly. However, they introduced to the most known technologies in software and web application development, but they are not completely aware of security issues, threats, and vulnerabilities of their software systems. Therefore, they are graduated and start developing various types of applications for different private and public sector organizations. It is vital to the educational institutes to provide more information and guidelines about different types of vulnerabilities, threats, and ways to overcome security challenges in their software and web application development courses. To get a better idea about the current situations among students and developers different surveys and interviews conducted. The results show, most of the students, graduates and developers are suffer from lack of information about most known security risks to consider in the development of web applications. Therefore, recommendation and suggestions are provided to mitigate and overcome the problems of security concerns in the application development in Afghanistan.**

Keywords— **web application, vulnerability, scanner, application, awareness, attack, injection**

I. INTRODUCTION

A common way of information sharing is happening through websites and web applications nowadays. Web applications provide a great interface for the users through web pages. As the number of web applications is increasing, it is important for the developers as well as the users to consider the security of information stored in these applications. Therefore, web applications and websites have to be secure against unauthorized access and unauthorized users. Attackers use different techniques to exploit web applications such as SQL injection, cross-site scripting, cookie-session theft, and self-propagating worms in users' emails. Attacks happen if the web application or website suffers from poor authentications, poor data encryption, and use of cracked plugins including destructive codes.

Afghanistan is a developing country which is growing in ICT every day. Many web applications, mobile applications, and websites are designed and developed by students and graduates of computer science in this country. But the security of these software applications would be a major concern in the future. Computer Science students from different Afghan universities have been graduated and are developing various types of web applications for different government and private organizations. The problem arises when the security of such applications come into a deep discussion. In this research over 1000 students and graduates from different universities who are actively developing web applications in the job market of Afghanistan are surveyed. The most important issue is that developers are not considering web application security as a major concern. It is because students graduate with a poor web applications security background as it is not considered as an important issue in academia. To overcome the problem, a major change in academia as well as job market is required.

II. IMPORTANCE OF SECURITY IN WEB APPLICATIONS

A web application directly accessed and interpreted by clients' browsers on the internet. From 1994 the use of the Internet became popular. Users from different locations used to access web pages. The participation of different organizations was tremendous, organizations and developers used to put web

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 11, ISSN No. 2455-2143, Pages 8-13
Published Online March 2020 in IJEAST (http://www.ijeast.com)

application available to the public. From its begging to date millions of requests are generated to access different web applications' data. Web applications have become an integral part of the recent industry. In recent decades significant online businesses are emerging, including universities, social networking, online banking, online money transactions, and online shopping. The sensitive data that web applications can handle are credit card numbers and shopping activity information, online transactions and different authentication mechanisms. In order to be defensible and secure, web applications must protect user's data from unauthorized access, disclosure, disruption, modification, and destruction [1].

The architecture of web applications is client-server. Mainly, the server is responsible for the processing of data and responds the client's requests. The only difference that makes the web application isolated from other software developments is, that web application developers mainly use technologies that are accepted by the World Wide Web (WWW). Additionally, standardized network protocols are used such as Hypertext Transfer Protocol (HTTPS) [2].

In recent years, the Afghanistan government provided different online services such as online passport requests, online E-Tazkera registration which contain very sensitive data. If the data is such a case is compromised, public information security comes in a serious problem. Therefore, web application security is very important to the government as well as to private organizations.

III. Vulnerability in Web Applications

Vulnerability in web applications can be defined as a weakness which could be exploited by hackers. A hacker can use web application weakness to compromise organizations' assets into serious danger [4]. Compromising organizations' assets for any purpose is a cyber-crime [3]. The hacker who committed such crime is called cybercriminal. Some of the common ways used by an attacker are discussed here.

A. Injection

"Injection weaknesses, such as SQL, OS, XXE, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query" [5]. It could happen if the user privilege and user permission to access sensitive data is unprotected. The attackers use weaknesses and execute their own codes as a command or part of the query to access the sensitive data via an interpreter. These weaknesses are very common and usually found in "SQL, LDAP, XPath or NoSQL queries". The most common weaknesses can be found easily [5].

Example:

1. An application use untrusted data in the construction of the following vulnerable SQL call:

```
String query = "SELECT * FROM accounts WHERE custuer_id = ' " + request.getParameter("id")+" ' ";
```

2. Similarly, an application's blind trust in frameworks may result in queries that are still vulnerable, for example Hirbernate Query Language:

```
Query HQLQuery = session.createQuery (FROM accounts WHERE customer_id = "' + request.getParameter("id")+" ' ");
```

The Query returns always true so this is very dangerous because it may invoke all data from the database.

B. Broken Authentication and Session Management

Incorrect authentication and session management allows hackers to acquire access to the sensitive data such as passwords, keys, session token, and some other exploits [5].

Example:

1. A travel reservations application supports URL rewriting, putting session IDs in the URL:

```
http://example.com/sale/salesitem;jsessionid=20PQOD045DSEHKILE5L4064LF0?dest=Hawaii
```

2. Application's timeouts aren't set properly. Users uses a public computer to access site instead of selecting logout the user simply closes the browser tab and walks away. An attacker uses the same browser an hour later, and that browser is still authenticated.

C. Cross-site Scripting (XSS)

XSS weaknesses occurs whenever an application includes untrusted data in a new web page without proper validation. The attackers can use this weakness to execute there exploit "victim's browser ", or they can direct their victim to a vulnerable website or destroy the style of the website [5].

Example:

The application uses untrusted data in the construction of the following HTML snippet without validation or escaping:

```
(String)page += "<input name='creditcard' type='text' value='" +request.getParameter("cc")+ "'>";
```

The attacker modifies the 'CC' parameter in his browser to:

```
'><script>document.location='http://www.attacker.com/cgi-bin/cookie.cgi?foo='+document.cookie</script>'.
```

This attack causes the victim's session ID to be sent to the attacker's website allowing the attacker to hijack the user's current session.

D. Broken Access Control

Sometimes attackers can access sensitive data by using the broken access control "weakness". broken access control is

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 11, ISSN No. 2455-2143, Pages 8-13
Published Online March 2020 in IJEAST (http://www.ijeast.com)

"restrictions on what authenticated users are allowed to do are not properly enforced". This flaw allows attackers to get different access privileges such as modifying some files and accounts, changing the privileges of other users, access sensitive data and some other functionalities [5].
Example:

The application uses unverified data in a SQL call that is accessing account information:

```
Pst.setString(1, request.getParameter("sample"));

resultSet results = pst.executeQuery();
```

An attacker simply modifies the 'sample' parameter in the browser to send whatever account number they want. If not properly verified, the attacker can access any user's account information.

```
http://example.com/app/accountinfo?sample=notmysample
```

E. Security Misconfiguration

These flaws allow attackers to get access to the default accounts or some other files that are not protected properly or some unsecured directories. this flaw occurs in different levels of web application such as "application framework, application server, web server, database server and platform". "Good security" needs to define a valid and admirable "configuration" [5].
Example:

The app server admin console is automatically installed and not removed. Default accounts aren't changed. Attacker discovers the standard admin pages are on your server, logs in with default passwords, and takes over.

F. Sensitive Data Exposure

One of the most common web application vulnerabilities is Sensitive Data Exposure. This vulnerability allows the attackers to steal the valuable data. It happens when we use weak algorithms, have a weak password, untrusted management of the keys, and so on [5].
Example:

A site simply doesn't use TLS for all authenticated pages. An attacker simply monitors network traffic (like an open wireless network), and steals the user's session cookie. The attacker then replays this cookie and hijacks the user's session, accessing the user's private data.

G. Insufficient Attack Protection

Insufficient attack protection is the ability of "automatically detecting, logging, responding" vulnerable inputs that are sent by the hackers in applications. if anyone who is connected to the internet sends a vulnerable or invalid input to your application, does your application prevent and detect that? But "the majority of application and APIs lack the basic ability to detect, prevent and respond to both manual and automated attacks" [5]. To ensure the application security we must prevent and block such attacks automatically [5].
Example:

Attacker uses automated tool kike OWASP ZAP or SQLMap to detect vulnerabilities and possibly exploit them.
Attack detection should recognize if the application is being targeted with unusual requests and high volume of data. Automated scans should be done to distinguish these requests from normal traffic.

H. Cross-site Request Forgery (CSRF)

CSRF is a vulnerability that makes it possible for an attacker to enforce a user naively perform some actions that attacker wants, the hackers usually enforced "victim's browsers to send a forged HTTP request, including the victim's session cookie" and add some exploit to it and then redirect the victim to a malicious address. detecting this vulnerability is easy by testing and analyzing the source code [5].
Example:

The application allows a user to submit a state changing request that does not include anything secret. For example:

```
http://example.com/app/transferFunds?amount=1000&destinationAccount=7245651246
```

So the attacker constructs a request that will transfer money from the victim's account to the attacker's account, and then embeds this attack in an image request or iframe stored on various sites under the attacker's control:

```
<imgsrc="http://example.com/app/transferFunds?amount=1000&destinationAccount=attackersAcount#" width="0" height="0" />
```

I. Using components with Known Vulnerabilities

Sometimes the components of our application are the cause of vulnerability such as "libraries, frameworks, and other software modules. Because they may run with the same privileges as the application does". The hackers often identify the vulnerable parts of the application by using some tools and execute their exploit and may down part of the application or server [5].

J. Under protected APIs

Mostly, all applications use the APIs which are located on client applications such as JavaScript in the browser and mobile apps. These APIs mostly contain numerous flaws and vulnerabilities which used by the attackers to attack the web application [5].

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 11, ISSN No. 2455-2143, Pages 8-13
Published Online March 2020 in IJEAST (http://www.ijeast.com)

K. Graduates Awareness about Web Application Security

In this research about 1000 computer science students and graduates from different universities who are active web application developers in the market of Afghanistan are participated.

IV. Evaluation Results

A. Graduates Awareness of Web Application Security

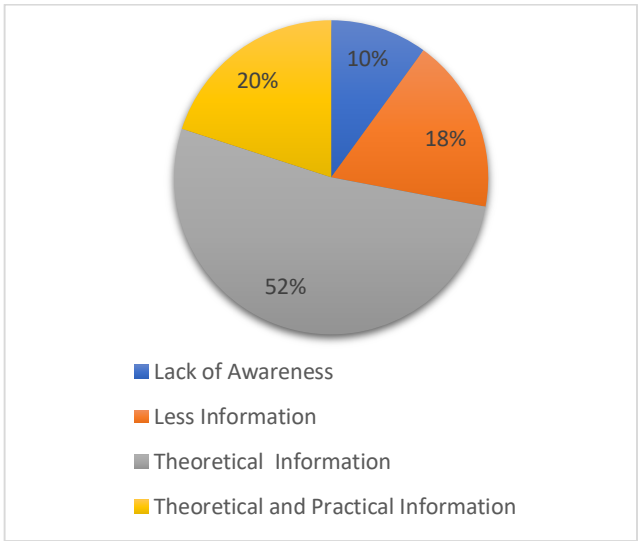

Fig. 1. Graduates Awareness of Web Application Security

According to the above diagram, about 70% of the graduates have information about web application security. But most of them have only a theoretical knowledge and suffer to implement security issues in web applications. The result is, they develop week applications in the market of Afghanistan.

B. Graduates Awareness about Vulnerabilities

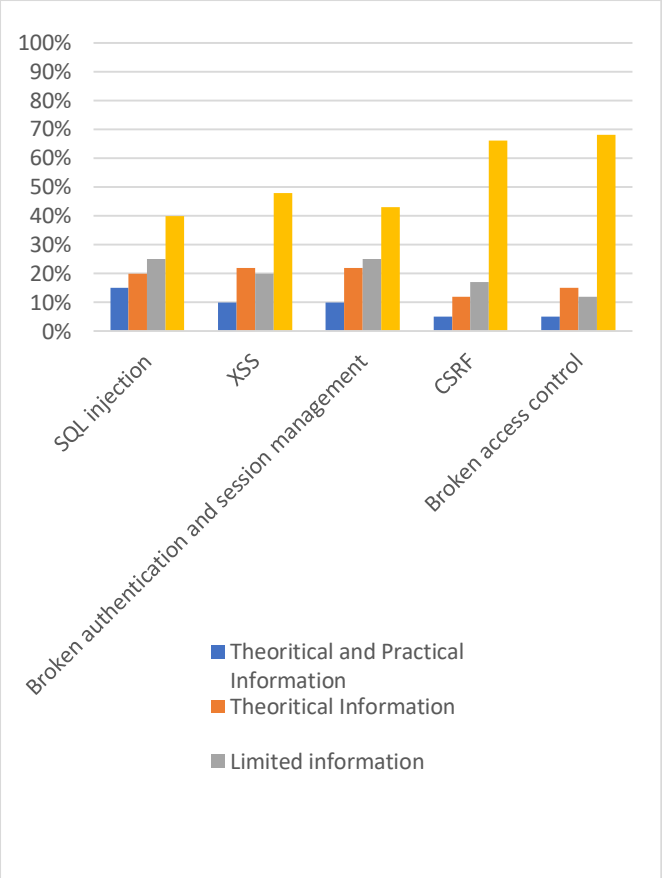

Fig. 2. Awareness of Graduages on Top 5 Vulnerabilities

According to the above diagram, most of the students and graduates didn't have enough information about different vulnerabilities specifically SQL Injection, XSS, Broken authentication and session management, CSRF and Broken access control.

C. Graduates and Students' Awareness of Testing Tools

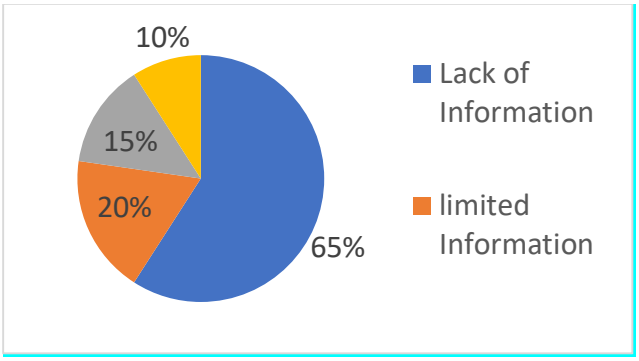

Fig. 3. Graduates and Students' Awareness of Testing Tools

According to the above diagram, most of the students and graduates didn't have enough information about testing tools so that they could test their websites to find the vulnerabilities.

V. RECOMMENDATIONS

Although, security related topics are beyond the scope of bachelor degree but based on the market demand of Afghanistan it is highly recommended to have such subjects. In the curriculum of academia (Universities), subjects in bachelor should be added to deal with security challenges. In the four-year studies to complete bachelor degree, there are limited topics which deals with security challenges.

Based on our evaluation, we highly recommend for universities to add the OWASP or similar methodologies in their bachelor degree program. The students must have theoretical and practical knowledge of security related challenges based on OWASP. Furthermore, there should be awareness training workshop in the market place of Afghanistan for stakeholders.

It is necessary to analyze the source code for detecting and repairing the security flaws [11]. Sometimes it is needed to check the source code or test the web applications for vulnerabilities using on-site penetration testing tools [10]. Therefore, software applications must be checked with application security (Vulnerability) scanners before publicizing them. There are open source and close source application security scanners to scan any web application which provide a comprehensive report. For instance, Vega [7] and Acunetix [8] are very popular and powerful vulnerability scanners. Besides, developers have to consider the security of their database and apply different security policies to keep the database application more secure through WWW [12].

Vega vulnerability scanner

Vega vulnerability scanner is very helpful to find web application vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and other popular vulnerabilities. Vega scanner can automatically scans web application and provide a comprehensive report of vulnerabilities. Furthermore, Vega scanner categorize vulnerabilities into High, Medium, Low, and Info. High and Medium categorizes of vulnerabilities are very risk and can damage the assets of any organization [7].

Acunetix vulnerability scanner

Acunetix is a popular and powerful vulnerability scanner which a lot of companies [8] are currently using it. Acunetix is used for vulnerabilities such as "SQL Injection, XSS, XXE, SSRF, Host Header Injection, and over 3000 other web vulnerabilities" [9].

VI. CONCLUSION

The most important aspects of secure web applications are the ability to bind an incoming request or invalid inputs. In this research paper, a complete survey is conducted to evaluate students, graduates and developer's knowledge about web application security, web application vulnerability scanners, and top five web application vulnerabilities. The result shows, lack of security subjects in universities, lack of practical implementation security challenges and lack of using different web application vulnerabilities among developers. To overcome the problems, major changes in academia and markets are necessary. The curriculum of academia should be design based on demand of market which requires to add security related subjects more. Security awareness training workshops are necessary for organization to discuss web application security and top ten OWASP vulnerabilities.

VII. ACKNOWLEDGEMENT

We thank the Ministry of Higher Education, Herat, Kabul and Polytechnic Universities for providing us the research facilities. Besides, we thank the anonymous reviewers for their helpful feedback and our students to contribute in our research processes.

VIII. REFERENCES

[1] Jazayeri, M. (2007, May). "Some trends in web application development. In Future of Software Engineering" (FOSE'07) (pp. 199-213). IEEE.

[2] Scambray, J., & Shema, M. (2006). "Hacking Exposed Web App". Tata McGraw-Hill Education.

[3] Stuttard, D., & Pinto, M. (2011). "The web application hacker's handbook: Finding and exploiting security flaws". John Wiley & Sons.

[4] DuPaul, N. "Application Security Vulnerability: Code Flaws Insecure Code".

[5] OWASP. (NA). "OWASP Top Ten. OWASP". https://owasp.org/www-project-top-ten/. Retrieved 2019, Dec 12

[6] OWASP. (NA). "Category:Vulnerability Scanning Tools – OWASP". OWASP. https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools. Retrieved 2019, Dec 15

[7] Infosec Institute. (NA)."14 Best Open Source Web Application Vulnerability Scanners". http://resources.infosecinstitute.com/14-popular-web-application-vulnerability-scanners/. Retrieved 2019, Dec 10

[8] Acunetix. (NA). "Website security - keep in check with Acunetix". https://www.acunetix.com/. Retrived 2019, Dec 17.

[9] N0Where. (NA). "Web Application Security Scanner: Netsparker, CyberPunk". https://n0where.net/web-application-security-scanner-netsparker/. Retrieved 2019, Sep 17

[10] M. Shema. (2011). "Web application security for dummies. Chichester". John Wiley and Sons.

[11] Microsoft. (NA). "Ten Tips for Designing, Building, and Deploying More Secure Web Applications". https://technet.microsoft.com/en-us/library/cc512638.aspx. Retrieved 2019, Oc 10.

[12] Lew, A., & Mauch, H. (2006). "Dynamic programming: A computational tool". (Vol. 38). Springer.