The Reality of Applying Security in Web Applications in Academia

Mohamed Al-Ibrahim
College of Basic Education, PAAET
Kuwait

Yousef Shams Al-Deen
Telecommunication & Navigation Institute, PAAET
Kuwait

***Abstract*—Web applications are used in academic institutions, such as universities, for variety of purposes. Since these web pages contain critical information, securing educational systems is as important as securing any banking system. It has been found that many academic institutions have not fully secured their web pages against some class of vulnerabilities. In this empirical study, these vulnerabilities are focused and their existences in the web sites of the academic institutions are shown. The degree of securing web pages in education systems is measured. The differences among academic institutions on protecting their web applications are discussed. Recommendation on ways of protecting websites is addressed.**

Keywords—Web applications; Security; Education systems

I. INTRODUCTION

A web application is an application that is accessed with a web browser over a network such as the Internet or an intranet. Web applications are popular due to the ubiquity of the browser as a client. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity. Web applications are used to implement various sort of applications including E-commerce, online banking, webmail, business applications and many other functions [15].

Since the Internet is open systems and the web applications are increasingly used to deliver critical services, they become a valuable target for security attacks. The security of the web applications become a main concern to many users of the web applications, especially when the web application is interactive and requires the exchange of sensitive information such as financial, health, or credit cards numbers. If these web applications were not secured, then the entire database of sensitive information is at serious risk. Therefore, there was great effort in both the research and industry community to provide secure communication services to web applications. A great deal of attention has been given to network-level security, such as port scanning, and great achievements have been accomplished at this level as well. However, it was found that about 75% of attacks were targeted to application-level, such as web servers [8].

One of the important sectors that exploit the web technology in their services is the education sector such as research institutions, universities, training organizations …etc. Web application and web sites are heavily used in education for information dissemination, lectures, assignments, collaborations, discussions, conferences, grading, training, distance learning, research activities and many others. Web applications in education sector usually hold sensitive information, such as faculty-members researches, student grades, staffs accounts ...etc. These data or information need to be secured from non-authorized users. Unfortunately, the sense and awareness of securing these data have not received great attention from academicians. While securing enterprise data is usually focused on financial, military or demographic organizations, it is often neglected in education organizations.

Goals and Contributions:

The main goals behind this research paper are twofold. First, is raising the digital security awareness among academicians in education, scientific, or research centers. Second, is to identify the main security vulnerabilities in web applications in education system. Also, to measure the variation of security level of the education organization from the standard levels of security set by known organizations. Further, to study why education institutions differ in terms of securing their web pages, i.e. what are the factors (budget, specialists, technology,…, etc) that affect implementing security procedures.

The methods includes auditing web application security for the interactive web site of several academic institutions in State of Kuwait during the years 2013 and 2014, including universities, colleges, and research institutes. The results reveal a set of vulnerabilities in web applications that are commonly found in educational systems. It also exposes the degree of using security technologies in protecting the web application against a set of known threats.

We suggested some defend techniques as counterattack. We also list a number of recommendations as security policy. The methodology and tools described later in this paper could be used as guideline for similar studies. The main lesson to address is that educational systems have to revise their web-based applications against sort of vulnerabilities.

Paper Structure:

The paper is organized as follow. Section II provides a brief technical background on the security of web technology as well as a literature review on research papers in web security. Section III describes the methodology and tools used in data gathering. Section IV present the results obtained and analyzed the outcomes. Section V discusses the factors the affect applying security in institutions. Finally, Section VI concludes with recommendations.

II. BACKGROUND

It is important at this stage to start defining some security terminologies used frequently in this paper. First, a *threat* is a danger that could affect the security (confidentiality, integrity, availability) of assets in an organization, leading to a potential loss or damage. *Vulnerability* is the existence of a weakness in design or implementation error that can lead to an unexpected, undesirable event compromising the security of the system. While an *Exploit* is a software bug, or feature, that allows access to a computer system beyond what was originally intended by the operator or programmer. Last, *attack* is an action that violates security carried out by an adversary, or an unauthorized entity, trying to carry out a hostile action against a system in a way that may compromise the system security. The Web platform is a complex ecosystem composed of a large number of components and technologies, including HTTP protocol, web browser (e.g., Explorer, Chrome), server applications (e.g., PHP,ASP) and client technologies (e.g., Javascript, Flash).

A. Why the need to secure web applications?

Website security is today's most overlooked aspect of securing the enterprise and should be a priority in any organization. Increasingly, hackers are concentrating their efforts on web-based applications – shopping carts, forms, login pages, dynamic content, etc. Accessible twenty-four hours a day, seven days a week from anywhere in the world, insecure web applications provide easy access to backend corporate databases and also allow hackers to perform illegal activities using the attacked sites. According to a report conducted by Web Application Security Consortium WASC [13] reveals that about 49% of the web applications being reviewed contain vulnerabilities of high risk level and more than 13% of the website can be compromised. A victim's website can be used to launch criminal activities such as hosting phishing sites or to transfer illicit content, while abusing the website's bandwidth and making its owner liable for these unlawful acts. Another study by Gartner Group [5] reveals that 75% of cyber-attacks are launched at the web application level. Website security is today's most overlooked aspect of securing the enterprise and should be a priority in any organization. Increasingly, hackers are concentrating their efforts on web-based applications – shopping carts, forms, login pages, dynamic content …etc.

On the other hand, hackers already have a wide repertoire of attacks that they regularly launch against organizations including SQL Injection, Cross Site Scripting, Directory Traversal Attacks, Parameter Manipulation (e.g., URL, Cookie, HTTP headers, web forms), Authentication Attacks, Directory Enumeration and other exploits. Moreover, the hacker community is very close-knit; newly discovered web application intrusions, known as Zero Day exploits, are posted on a number of forums and websites known only to members of that exclusive group. Postings are updated daily and are used to propagate and facilitate further hacking.

B. Why are web applications vulnerable?

Although most of the originations try to protect their intranet system by firewalls and SSL, firewalls and SSL provide no protection against web application hacking, simply because access to the website has to be made public. Web applications often have direct access to backend data such as customer databases. Most web applications are custom-made and, therefore, involve a lesser degree of testing than off-the-shelf software. If web applications are compromised, hackers will have complete access to backend data of the institution even though its firewall is configured correctly and its operating system and applications are patched repeatedly. Also, network security defense provides no protection against web application attacks since these are launched on port 80 which has to remain open to allow regular operation of the business. It is therefore imperative that the institution regularly and consistently audit its web applications for exploitable vulnerabilities.

C. Web Application Security Organizations

Due to the increase number of incidents of security attacks to web applications, many software vendors had fair efforts to clarify the web application security awareness, and type of vulnerabilities on the web sites to customers. Nevertheless, special, non-profit, charitable organizations have established solely to promote to the concept of web application security. The most two important organizations in this area are the Open Web Application Security Project OWASP [9], and the Web Application Security Consortium, WASC [13]. OWASP is dedicated to finding and fighting the causes of insecure software. Everything in OWASP is free and open source. OWASP provides an awareness document that describes the top ten web application security vulnerabilities. The OWASP Top-Ten represents a broad consensus about what the most critical web application security flaws are. Also, they provide OWASP Guide Project, a massive document covering all aspects of web application and web service security. Among other documentation and video presentations, a complete list of their projects can be found in their project home page OWASP.

D. Literatre Review

In the last few years, application-level vulnerabilities have been exploited with serious consequences: Hackers have tricked e-commerce sites into shipping goods for no charge, usernames and passwords have been harvested, and confidential information (such as addresses and credit-card numbers) has been leaked. Researchers start to investigate new tools and techniques which address the problem of application-level web security from multiple directions: pre, within, and post. Glisson,and Welland in [6] argue that security should be started first before the application development process upfront through an independent flexible methodology that contains customizable security components. Scott and Sharp in [10] described a scalable structuring mechanism when developing an application facilitating the abstraction of security policies from large web-applications developed in heterogeneous multiplatform environments; and presented a set of tools which assist programmers in developing secure applications which are resilient to a wide range of common attacks. Seo, Kim, Cho and Cha in [11] developed web Intrusion Detection System (IDS) that uses anomaly-based intrusion detection and application-level IDS tailored to web services to detect any security anomalies in web application. On the other hand, Grier, Tang and King in

[7] noticed that web browsers itself are not secure enough, so they focused on building a new secure web browser that prevent various vulnerabilities that exist in current browsers. Other papers presented different ideas (e.g., [2]; [3];[4]. Later a substantial amount of research effort have been devoted to hardening web applications and mitigating the attacks. Many of these techniques make assumptions on the web technologies used. Li and Xue [14] argued that a secure web application should preserve three security properties: *Input validity* means the user input should be validated before it can be utilized by the web application; *state integrity*, means the application state should be kept untampered; and *logic correctness* means the application logic should be executed correctly as intended by the developer.

III. METHODOLOGY

A. Target Destinations.

We targeted twelve higher-education, academic and research institutes in State of Kuwait who are involved under the umbrella of Ministry of Higher Education (MOHE). These are divided into two categories: governmental & private institutes. The governmental institutes are those non-profit organizations which their budgets are funded directly from the government as well there policies. These institutions three in total including Kuwait University (KU), Public Authority for Applied Education and Training (PAAET) and Kuwait Institute for Scientific Research (KISR). The private institutes are profit-based organizations and partially directed to government regulations include nine authorized private universities licensed from the Private Universities Council (PUC) which belongs to (MOHE). These colleges or universities includes (in abbreviations without extension) : ACK, ACM, AUM, AUK, AOU, KILAW,BHCK, GUST, and KBMS.

The targeted destinations of both categories are basically the application software's that provide services in shape of web-application. The main services in academia are student Information System (SIS). Campus-solution-systems such as PeopleSoft, Campus Vue, River Vue, Banner, Academia,…,etc are examples for on-shelf SIS software's. Due to system limitations in these applications, some colleges or universities prefer developing in-house applications for SIS using web technologies, such as ASP, PHP, .Net. to build dynamic and interactive websites applications and storing their data in databases.

B. Tools

The software specialist in finding security holes or vulnerability in websites is called *Scanner*. Web Scanners launches an automatic security audit of a website. It consists of two phases: first is *Crawling*, the process of building the site's structure. It enumerates all files and is vital to ensure that all the files on the website are scanned. Second is *Scanning*, the process of inspection intensely to find security vulnerabilities. By default, scanning process involves crawling.

Scanners are used to find crackers and possible problems in the applications. First it collects essential information about the web application such as web-server, Operating-System type, their version and any patches were installed; this information usually appears in system banner and is helpful to discover well-known vulnerabilities on the server [12]. Therefore, it is wise to hide such information from non-authorized. We used a web vulnerability scanner tools named *Acunetix* [1]. This software is used to check a wide range of vulnerabilities in a web site, and it includes many innovative features such as:

1) Automatic JavaScript analyzer

2) Industry's most advanced and in-depth SQL injection and Cross-site scripting testing

3) Visual macro recorder makes testing web forms and password protected areas easy

4) Extensive reporting facilities including OWASP Top 10 vulnerabilities

5) Multi-threaded and lightning fast scanner crawls hundreds of thousands of pages

6) Intelligent crawler detects web server and application language types

7) Crawls, analyzes web sites including flash content

C. Process

The followed methodology, in this research, to determine the degree of security in web application servers involved the following steps. First, scanning through the websites of each targeted destination and list all found vulnerabilities. Then, segregate the found vulnerabilities into four types according to their degree of severity, namely: *High*, *Medium*, *Low* and *Informational*. Later, we identified the vulnerabilities of each type and list them in separate groups according to their severity, and a table for each type was built. Fig.1 is snapshot of a session in a scanning process. Tables 2 through 5 list all vulnerabilities that were found of each type of vulnerability. Finally, each type of vulnerability was cross-checked with the list of top-ten vulnerabilities of OWASP [9] and if any of the vulnerabilities were matched, then a 10 percent number was added.

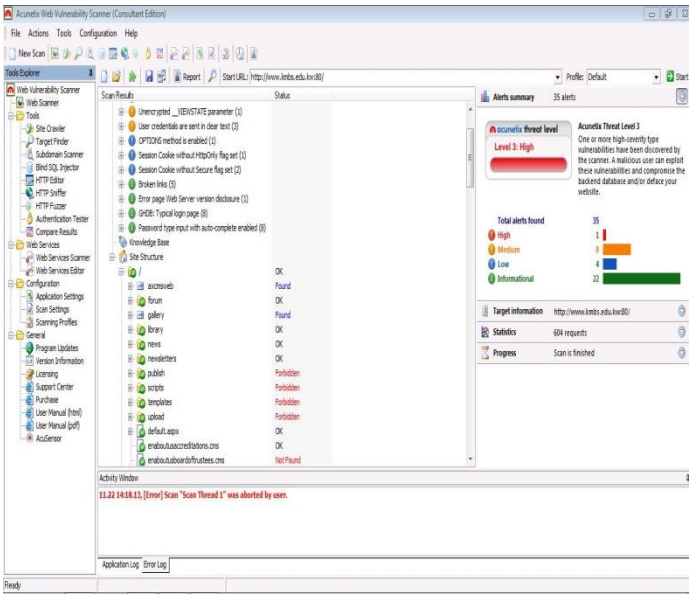

Fig. 1. Snapshot of scanning process

IV. ANALYSIS

The scanning tool of Acunetix reveals abundant information on the targeted destination under examination that may discloses valuable information useful for tactician the method of attack. Examples of basic exposed information are the following: the used web technology in the host, the operating system running the web server, the versions of system software's … etc. Other advanced diagnosing information includes: distribution of the total alerts for each type of threat levels (namely High, Medium, Low, and Informational), a list of file extensions found and the number of files per extension (file extensions can provide information on what technologies are being used on attacked websites), a distribution of top ten files that has lowest response times measured during the crawling process (the average response time for each host is computed in milliseconds and these files could be targeted in denial of service attacks), a distribution of the list of client scripts that contain Javascript code referenced from the website (Javascript is potential threat for many types of attacks), list of the external hosts that are linked from the organization websites, and finally, a list of email addresses found on the targeted host.

TABLE I. DISTRIBUTION OF VULNERABILITIES IN INSTITUTIONS

Inst / Level	High	Medium	Low	Information
ACK	1	0	0	1
ACM	2	2	6	4
AOU	4	11	15	221
AUK	5	19	4	199
AUM	6	8	2	2
BHCK	4	9	22	19
KBMS	3323	796	13	172
KILAW	2	9	9	56
GUST	1	2	0	29
PAAET	2	144	12	5
KU	113	24	11	4
KISR	0	7	2	4

After scanning tool analyzed target destinations, huge amount of data was accumulated. The total number of different threats found in all target destinations for each level of severity was as the following: High 14, Medium 15, Low 8, and 9 threats for informational. Table 1 provides statistical summary on the number of vulnerabilities found for each type in the websites of each institution of target destination.

Information revealed from figures Fig. 2 through Fig. 5 illustrate the frequencies of attacks of each type. It is easy to note from the graphs the common vulnerabilities that mostly appeared in the scanned website and their percentages of appearance according to the total number of found vulnerabilities of each type. We can figure out several remarks of each type of severity as we detail their discussion in the following subsections.

A. HIGH

The vulnerabilities of this type of severity are the most dangerous sort of threats which put a site at maximum risk for hacking and data theft. It has direct effect on the security, integrity, privacy of the information of the websites. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface the website. The total number of vulnerabilities of all websites that scanned destinations were limited to fourteen threats. Table 2 lists these vulnerabilities. From the table, we can define a shortlist of the most serious attacks that commonly found in education sector are H1 (ASP.NET Padding Oracle Vulnerability), H2 (Slow HTTP DOS attack) and H3 (Cross Site Scripting) with 18% appearance each. These three vulnerabilities occupy more than 50% of the most potential serious attacks. To analyze these three attacks in particular, as a sample for type 'High' of severity, a brief description of the attack and its direct implication as well as quick remedy for this threat are shortly described. Fig.2 below presents its appearance frequency.

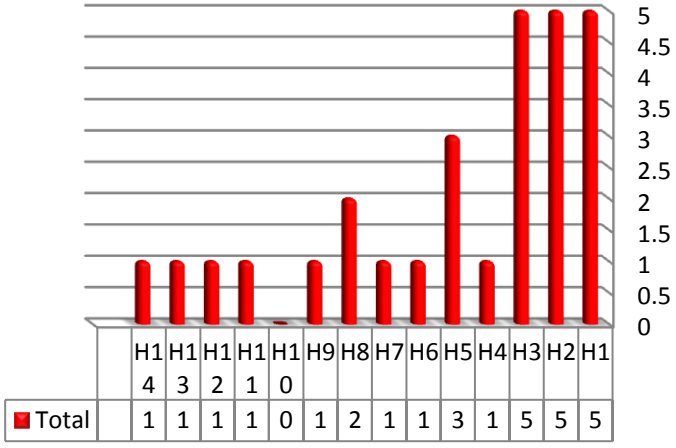

Fig. 2. High risk vulnerabilities

First, H1, ASP.Net uses encryption to hide sensitive data and protect it from tampering by the client. However, a vulnerability in the ASP.Net encryption implementation can allow an attacker to decrypt and tamper with this data. This vulnerability exists in all versions of ASP.Net. A direct result of this attack that an attacker who exploited this vulnerability could view data, such as the View State, which was encrypted by the target server, or read data on the server, such as web.config. This would allow the attacker to tamper with the contents of the data. By sending back the altered contents to an affected server, the attacker could observe the error codes returned by the server. One of the recommendations to stop this threat is to apply Microsoft patches solely for this problem.

Second, Slow HTTP POST DoS attacks rely on the fact that the HTTP protocol, by design, requires requests to be completely received by the server before they are processed. If an HTTP request is not complete, or if the transfer rate is very low, the server keeps its resources busy waiting for the rest of the data. If the server keeps too many resources busy, this

creates a denial of service. The impact is that a single machine can take down another machine's web server with minimal bandwidth and side effects on unrelated services and ports. One of possible solutions to this problem is that web server administrators can isolate or abort the traffic from the source of the attack.

Third, Cross site scripting (also referred to as XSS) is a vulnerability that allows an attacker to send malicious code (usually in the form of Javascript) to another user. It is a cause of the lack of input validity property to web applications. This is because a browser cannot know if the script should be trusted or not, it will execute the script in the user context allowing the attacker to access any cookies or session tokens retained by the browser. The implication is an attacker can steal the session cookie and take over the account, impersonating the user, and it is also possible to modify the content of the page presented to the user. The remedy to this threat is that scripts sent from a user as input should filter the metacharacters, i.e. a character that has a special meaning (instead of a literal meaning) to a computer program such as \ or ; or . (dot) or $ or ? ..etc.

TABLE II. HIGH RISK VULNERABILITIES

No	High	Total	%
H1	ASP.NET Padding Oracle Vulnerability	5	**18**
H2	Slow HTTP DOS attack	5	**18**
H3	Cross Site Scripting	5	**18**
H4	Apache Tomcat version older than 6.0.35	1	**4**
H5	Microsoft IIS tilde directory enumeration	3	**11**
H6	WebDAV Directory with Write Permissions	1	**4**
H7	WebDAV Remote Code Execution	1	**4**
H8	Blind SQL Injection	2	**7**
H9	FCKeditor spellchecker.php Cross Site Scripting	1	**4**
H10	jQuery Cross Site Scripting	0	**0**
H11	Spellchecker.php Cross Site Scripting	1	**4**
H12	HTTP Parameter Pollution	1	**4**
H13	HTML form without CSRF protection	1	**4**
H14	CRLF injection/HTTP response splitting	1	**4**

It is possible to detect short names of files and directories which have MS 8.3 file naming scheme equivalent in Windows by using some vectors in several versions of Microsoft IIS. For instance, it is possible to detect all short-names of ".aspx" files as they have 4 letters in their extensions. This can be a major issue especially for the .Net websites which are vulnerable to direct URL access as an attacker can find important files and folders that they are not normally visible. The severity of this threat stem from the potential for possible disclosure of sensitive information.

One interesting observation can be concluded from the result is that SQL Injection threat was appeared only 8%, although it was the top threat for many years according to OWASP statistics. This gives an indication of spread of web-security awareness among web developers against this threat.

B. MEDIUM

Vulnerabilities of this type are caused by server misconfiguration and site-coding flaws which facilitate server disruption and intrusion. The error messages of this type may disclose sensitive information. These information can be used to launch further attacks. Table 3 list the found vulnerabilities.

TABLE III. MEDIUM RISK VULNERABILITIES

No	Medium	Total	%
M1	Application error message	6	**19.4**
M2	Error message on page	4	**12.9**
M3	HTML form without CSRF protection	3	**9.7**
M4	User credentials sent in clear text	5	**16.1**
M5	Web Application Firewall detected	1	**3.2**
M6	OPTIONS method is enabled	1	**3.2**
M7	Possible Virtual Host found	1	**3.2**
M8	Session Cookie without Http only flag set	1	**3.2**
M9	Session Cookie without Secure flag set	1	**3.2**
M10	Apache http Remote Denial of Service	1	**3.2**
M11	Apache httpOnly Cookie Disclosure	1	**3.2**
M12	FCKeditor Arbitrary File Upload	1	**3.2**
M13	HTML form without CSRF protection	3	**9.7**
M14	Unencrypted __VIEWSTATE parameter	1	**3.2**
M15	SSL weak ciphers	1	**3.2**

The highest three threats of this type are M1, M2 and M4 are interestingly common in similarity. The three threats share the vitality of system messages for malicious users. First, M1 represent the problem that error/warning message may disclose sensitive information that could lead the adversary to some facts about the system application. It is usually originated to guide the system administrator to solve the problem, such as the location of the file that produced the unhandled exception, but it may used by adversary to better plan for an attack. Second, M4 reveals the problem of not encrypting user credentials such as input text data such as usernames or passwords that make it easy for malicious users to launch further attacks. This piece of information should always be transferred via an encrypted channel (HTTPS) to avoid being intercepted by adversaries. Third, M2 has similar cause and impact as M4. Fortunately, these the three threats despite its spread are easy to deal with by applying encryption on captured text and directing error messages to a designated log console. Fig. 3 shows a distribution of this type.

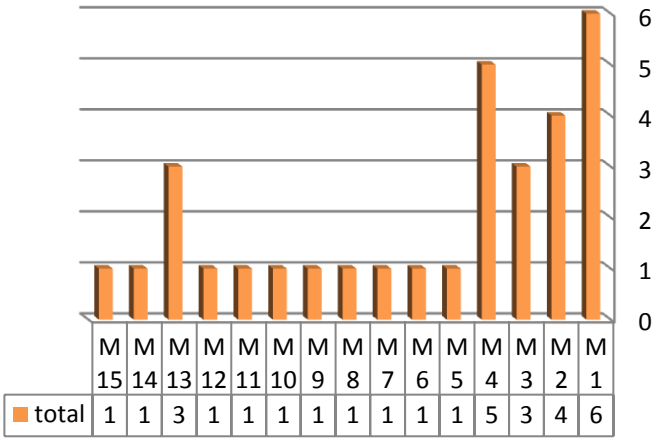

Fig. 3. Medium risk vulnerabilities

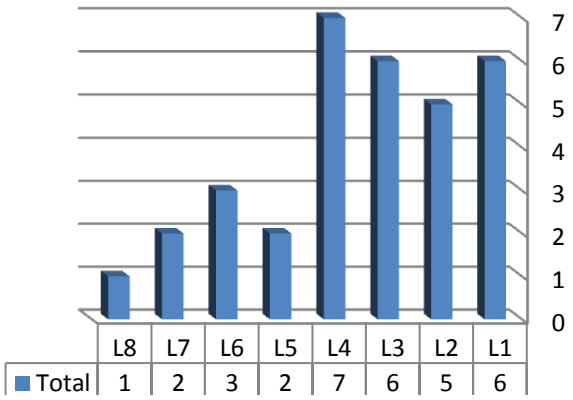

Fig. 4. Low risk vulnerabilities

C. Low

These vulnerabilities are derived from lack of encryption of data traffic, or directory path disclosures. In this type of attacks, the set of highest three appearance of attacks are L4, L3 and L1. First, L4 reflects the security status for an online session that is connected to the web in which its cookie does not have the Secure flag set. When a cookie is set with the Secure flag, it instructs the browser that the cookie can only be accessed over secure SSL channels. This is an important security protection for session cookies but does not have serious impact. Second, L3 represent a threat of a slow response time of a webpage when its response time is below the average response time of its site. This types of files can be targeted in denial of service attacks. An attacker can request this page repeatedly from multiple computers until the server becomes overloaded. Third, L1 threat indicates that the OPTIONS method is enabled on this web server and it provides a list of methods that are supported by the web server, it represents a request for information about the communication options available on the request/response chain identified by the Request-URI. The OPTIONS method may expose sensitive information that may help an malicious user to prepare more advanced attacks. Therefore, it's recommended to disable OPTIONS method on the web server. Fig. 4 presents distribution of low vulnerabilities

TABLE IV. Low Risk Vulnerabilities

No	Low	Total	%
L1	OPTIONS method is enabled	6	**18.8**
L2	Possible sensitive directories	5	**15.6**
L3	Slow response time	6	**18.8**
L4	Session Cookie without Secure flag set	7	**21.9**
L5	Session Cookie without HttpOnly flag set	2	**6.3**
L6	Login page password-guessing attack	3	**9.4**
L7	File upload	2	**6.3**
L8	TRACE method is enabled	1	**3.1**

D. Informational

This type of threats reveal information through Google hacking search strings, or email address disclosure. Threat I1, Broken Links, alone form 30% of this type of attacks. It refers to any link that should take user to a document, image or webpage, that actually results in an error. It indicates that a page was linked from the website but it is inaccessible anymore. It may cause problems navigating the site. Second, I2 represent the threat of exposure of email addresses that may not be needed to be exposed and it is the source of the majority of spam problems. Third, I6 represent a threat when a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved. Thereafter, when the form is displayed, the name and password are filled-in automatically or are completed as the name is entered. An attacker with local access could obtain the clear-text password from the browser cache. The set of threats I1, I2 and I6 represent 60% of threats of this type, but fortunately they are easy to solve or prevent. It seems that the systems administrator do not have enough tools to discover these threats. Fig. 5 presents the distribution of this type of threats.

TABLE V. Informational Risk Vulnerabilities

No	Informational	Total	%
I1	Broken links	9	**30**
I2	Email address found	5	**16**
I3	Microsoft Frontpage Configuration Information	3	**10**
I4	GHDB: Frontpage extensions for Unix	3	**10**
I5	Possible username or password disclosure	3	**10**
I6	Password type input with auto-complete enabled	4	**13**
I7	Files listed in robots.txt but not linked	1	**3**
I8	Content type is not specified	1	**3**
I9	Error page web server version disclosure	1	**3**

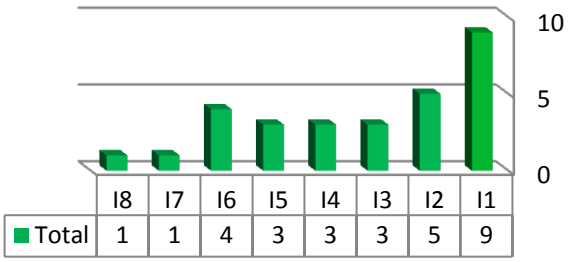

Fig. 5. Informational risk vulnerabilities

V. Discussion

It was obvious from the analysis section early presented in Table I regarding the distribution of vulnerabilities in the targeted institutions that most institutions have some weakness in their web security. There is also big disparity among the four levels of vulnerabilities, i.e. some have big number of High-level vulnerabilities while having small number of Informational-level vulnerabilities, and vise versa. This raises some questions: why this phenomenon occurs? What are the factors that affect enforcement of security in these institutions? To answer these questions, a survey was prepared and distributed to the I.T. managers in the institutions. The main affecting factors raised in the survey are: budget, expertise, tools, policies, management support, equipments, and awareness. Statistical outcome of each factor is as follow:

A. Budget

This factor reflects the fact that the lack of enough budgets may affect possessing cutting-edge technology. This hypothesis is important to investigate since there is difference in budgets between private and governmental institutions. All governmental institutes in the survey indicated that the budget supported for I.T. is generous, but among the private universities 30% declared that they don't have enough budgets dedicated to apply security techniques. On the other hand, governmental institutes has slower routine process due to the long documentary cycle in the government for purchasing makes the ordered technology sometimes become obsolete by the time it arrive, but it is faster in private universities which don't follow this routine.

B. Expertise

This hypothesis reflects the fact whether the lack of expertise specialist in network security form a deficiency. Almost all organizations have I.T. department, but few has a section, unit, or at least specialists in information security. With the diversity and complexity of security problems from application layer to physical layer, it becomes essential to have specialists with profound experience in digital security to manage and solve diverse and emerging security issues. Thus, the existing of threats or vulnerability in a system may give a clue of non-awareness in dealing with it. In private universities, 40% indicated not having security specialist, while 50% in governmental institutes indicated not having security specialist.

C. Awareness

In case the institution does not have specialist or experts in information security, the I.T. specialist must have the basic knowledge in web security in particular. Web application developer should educate themselves with latest threats in web technology. Several online resources and organizations exist nowadays that frequently update their websites with the recent knowledge or statistics of threats, attacks, or vulnerabilities in web technology. OWASP, WASP are examples for such non-profit organizations. 56% of total responses were not aware of embedding security methods in coding in-house applications. This high figure reflects the fact of obscurity of security principles among many programmers and system analysts when coding software. This has to be thought in early stages of computer curriculums of programming subjects in colleges and institutes.

1) Equipments

Special security devices such as firewalls and anti-virus form the first defense line of security. Establishing DMZ within network equipment also plays crucial role in guarding and saving the enterprise assets. 90 % of response indicated having sophisticated security technology and tools such as firewalls and anti -virus, only 30 % indicated having penetration tools for self diagnosing and testing such as Sniffer (networking tool) or Acunetix (security tool).

D. Management Support

The hypothesis in this item states that the upper management in institutions may not give security of information a priority when the decision reaches to allocate budget for devices or training in security technology. The case is opposite in financial organizations, such as banks, where the upper management appreciates the safety of their monetary assets. This awareness related to upper management should be shifted to scholars and managers in education sector to protect their records and files that may hold vital information such as students grade, ongoing researches, or classified data. This hypothesis found to be true 57% of total responses indicated that upper management is not aware the importance of this issue.

E. Policies

Deploying security policies enhance overall security in any organization. 78% of total participants indicate deploying security policies. With further investigation, it was found that many of security policies were concentrated only on forcing password changes. In fact, the concept of security policies is more than this portion. The document in [16] details major security policy standards for information systems technology.

VI. Recommendations

The methods and techniques to protect the web applications can vary from administrational to technical, from prevention to protection, from coding-level to monitoring-level. In this section, suggested ideas are presented to make deploying web technology in education more secure:

A. Administrational

We propose establishing a central authority for the higher education institutions to ensure the safety of digital

information that has the authority and power to impose security standards on web technology and its applications among higher education institutes and research centers. Since the information held by these destinations are critical and its integrity is para important, such as the academic level of students (marks, grades, GPA), or could be of nation security interest (military and intelligence research), or technology competence (between companies or research centers) …etc, therefore, it is very important that this authority monitors the web security of their affiliated organizations. This authority is supposed to have the right not to provide license to institutions without passing the security standards of its digital information. Also, it has the right to revoke the accreditation of a university that found to have security breaches in their digital systems. This principle is actually very much adopted in the financial sector. For example, we can notice how the central banks in many countries monitor the monetary and interest rates in banks to preserve the stability of economy of the country. In state of Kuwait, as in this research took place, the potential organization to take this role is the PUC, which has the authority to give the licenses to open new private colleges and universities in the country, while MOHE can take same role for governmental and research institutes that their budgets are directly funded by the government. Other countries also have similar organizational authorities with this regard. Assuring quality and accreditation organizations such as Accreditation Board for Engineering and Technology (ABET) could put digital security assurance among its evaluation factors to grant accreditation to its evaluated institutions.

B. Technical

Among important issues for any system administrator is to perform the following tests that are solely related to security of their web technology:

1) Test Web Messages or regular basis
2) Test for Web Storage SQL injection.
3) Check SSL versions, Algorithms, Key length.
4) Check for Digital Certificate Validity (Duration, Signature).
5) Test for user enumeration.
6) Test for authentication bypass.
7) Check if data which should be encrypted.
8) Check for wrong algorithms usage depending on context.

C. Prevention

System administrators can do some precaution methods to prevent possible attacks by closing points of potential exploits. One of the primitive and essential tasks for any system administrator is to update their system software's on regular basis. This includes updating the operating system for advanced editions or any patches and service-pack provided by the vendor, also, updating their servers and application software's, drivers. Yet, the administrator has the responsibility to gather information about the site under control to manually explore the sites to find any holes or bugs especially for special kind of spider or crawl for missed content or hidden source of threat. There are many tools that can do this task even built by some operating systems. Moreover, system administrator has to make regular configuration management test to check for commonly used application and administrative URLs, and to check for old, backup or unreferenced files. System administrator has also to perform regular session management by establishing how sessions are handled in the application, check session tokens for cookies flags ...etc.

D. Protection

If an attack launched and discovered, it is possible to take some actions to stop the impact of it**.** The Denial of Service attack, for example, can be stopped by testing for anti-automation and test for account lockout**.** Also, system administrator should test the proper authorization are done in proper way. It is important to test for path traversal, test for bypassing authorization schema.

E. Construction

Many threats can be eliminated in early stages when developing the application. SQL injection, for example, is a threat that caused by improper coding which allows taking input from user that can later be exploited to masquerade in the database. Also, test for stored Cross Site Scripting (XSS). Many of security problems can be solved from the root if proper security mechanism were embedded in web applications to ensure that no potential vulnerabilities exist within the application. Robust program verification in early stage against a vector of security vulnerabilities that can expose them can dramatically reduce potential attacks.

VII. CONCLUSION

Testing web applications for security vulnerabilities something that needs be taken seriously. There are neat tools and interesting ways to take Web application hiccup, crash or otherwise give out information one should not be able to see. On the other hand, there are tools and ways to expose these vulnerabilities. The results of this study reveal a set of vulnerabilities in web applications that are commonly found in educational systems. These vulnerabilities range in risk from high, medium, low to informational threats. It also exposes the degree of security technologies in protecting the web applications against a set of known threats. We studied the possible reasons behind weakness of security in academic organizations. We suggested some defend techniques as counterattack. The main lesson to address is that educational systems holds sensitive digital data and information that is seductive for intruders, and therefore, have to revise their web-based applications against certain vulnerabilities and potential risks.

ACKNOWLEDGMENT

This research paper was funded by the Research Department at PAAET based on contract number BE-12-07.

REFERENCES

[1] Acunetix. Auditing your web site security with Acunetix web vulnerability scanner. Retrieved March 15, 2013, from website: http://www.acunetix.com/.

[2] Cao, M., Xing, T., & Wang, C.. Implementation of web security & identity scheme based on session & online table. Proceeding of the 4th ICCSE '09, pp.1278-1283, 2009.

[3] S. Chong, J. Liu, A. C. Myers, X. Qi, K. Vikram, L. Zheng, and X. Zheng, "Secure web applications via automatic partitioning," in SOSP '07: Proceedings of the 21st ACM SIGOPS symposium on operating system principles, 2007, pp31-44

[4] Dai, S. & Du,Y. **(2009).** Design and implementation of dynamic web security and defense mechanism Based on NDIS intermediate driver, Proceeding of APCIP '09,1, 506 –509.

[5] Gartner, www.gartner.com

[6] Glisson, W. & Welland, R. Web development evolution: the assimilation of Web engineering security, Proceeding of Third Latin American Web conference, 5 pp. 2005, doi: 10.1109/LAWEB.2005.48

[7] Grier, C., Tang, S. & King, S.T., (2008). Secure web browsing with the OP web browser, Proceeding of IEEE Symposium on Security and Privacy, 402-416. doi 1109/SP.2008.19

[8] Livshits, B., & Lam, M. Finding security vulnerabilities in Java applications with static analysis, Proceedings of the 14th conference on USENIX Security Symposium, 14, Retrieved 2009 from website http://www.portal.acm.org/, 2005.

[9] OWASP. Open Web Application Security Project . Retrieved from http://www.owasp.org/ index.php/ OWASP_Top_Ten_Project.

[10] Scott, D. & Sharp, R.. Developing secure web applications, Journal of Internet Computing, IEEE Publication, 6 (6), 38-45, 2002.

[11] J. Seo, H. Kim, S.Cho, & S. Cha . Web server attack categorization based on root causes and their locations, Proceedings of ITCC'04, 1, 90-96. doi: 10.1109/ITCC.2004.1286431, 2004

[12] Vieira, Antunes, & Madeira, Using web seurity scanners to detect vulunerabilies in web services . In IEEE/IFIP International conference Conference on Dependable Systems & /networks, 2009,DSN'09, ESOTRIL (2009)

[13] WASC, Classes of attacks, Retrieved from website:http://www.webappsec.org/projects/threat/classes_of_attacks.html

[14] Xiaowei Li & Yuan Xue, " A Survey on Web application Security", ACM Transactions on Computing Surveys, Vol. V, No. N, November, 2013

[15] Zhou, X., Zhang, Y., & Orlowska, E. (Eds.). Web technologies and applications, Proceedings of 5th Asia-Pacific Web Conference, Lecture Notes in Computer Science, Springer. 2003

[16] Technical Security Standard for Information Technology, http://www.iwar.org.uk/comsec/resources/standards/canada/tssit97e.pdf, Canadian federal government , 1997