Appendix A

Pilot Study: Appreciative Inquiry Technique

In this study, an accounting field has been selected as the case study in question, due to the fact that this field is informative and full of tasks; which must be implemented and have too much requirements to be achieved in the future software. Will use the same questions recommended by (Cooperrider,2008).

A) *Discovery Phase*

- What were your hopes and dreams, when you chose this project?

  A: To deliver an effective system to the client.

- Based on your past experiences, what was the greatest experience you have had with a project, when were you most successful and satisfied?

  A: The trust by the customer, and working in front of them, with their participation are the greatest achievements. It gives us immense satisfaction, when we would satisfy the customers with minimal effort and time, within the International Financial Accounting (IFA).

- Did you get any help from your friends/colleagues? Were you able to help them in return?

  A: Yes we did receive assistance from our friends and colleagues, and of course we reciprocated the help, based on our experience, we were able to solve the problems of our colleagues.

- Did you experience any unexpected incidents or face a difficult challenge? What did you learn from those?

  A: Yes, every difficult challenge we face was our learning curve, above all it boosts our courage of facing any other bigger challenge. For example (Real case), converting manual accounting entries to the code example: the sales code is 01, this leads to solve a big challenge for our organization. Because the traditional (manual) process needs a lot of time, effort and margin of error is high, also checking in this case is difficult.

- What conditions are contributed to that extraordinary level of success and satisfaction?

  A: The most important aspect is manually migrating data, this process demands a great deal of time and hard work, this process must be carried out with high precision, which is very challenging in our field. But we are now very much relieved now, as the new software can migrate the data after each movement and update records and data, the budget with high accuracy, and at low cost.

- What do you value most about yourself and your capabilities, as a member of the team, or as a contributor of the project?

  A: We can get involved and play a significant role in building the foundation of this software, by using our expertise and skills, to transform the manual processes towards the computerized processes. This will help us to get over the complications faced by us, especially subtracting the process of exports, and store the outcomes again in table; the new software will also help us to, make a query, related to inventory any time we wish, to know the financial situation in the organization, and business movements at any point of time.

- What do you value most about yourself as a member of the organization and/or member of the team?

  A: I value both, as a member in the organization I perform my duties to solve problems, and provide best services to our customers; and at the level of team, we focused on the software development, which is essential in putting things required of the system.

- What are the most important attributes that support your highest levels of success and satisfaction?

  A: Apply all kinds of accounting transactions in future system, inform the customers about all the operations in page of its own, to get their opinions about the tables and results.

B) *Dream Phase*

- What results do you expect from a team or a project?

  A: I expect all the following: Income list, List cash expenses, Budget page and customers Feedback.

- What is your vision as an ideal project in the future after many years (when, you have grand children)?

  A: To be the most powerful and largest accounting firm in the Market

  The next phases will use answer these questions to know "how can be and what will be".

C) *Design Phase*

- How can it be?

  A: Software Size: all types of accounting transactions, notify the customers about all the operations.

  Strength Identification: better services, customers trust, working in front of customers with their participation, work under International Financial Accounting (IFA), automatically migrate the data, high accuracy, make a query regarding inventory at any moment, account all the following: Income list, List cash expenses, Budget page and customers Feedback.

  Organization Roles: customer participation, get account customer feedback, outsider's viewpoint.

D) *Destiny Phase*

- What will be?

a) Requirements: (convert manual processes to computerized processes).

1) Converting manual accounting entries to the code.

2) Migrating data after each movement, and updating records and data.

3) Migrating data up to the budget.

4) Storing the results of the operations in table.

5) Making a query about inventory sat any time

6) Showing business movements at any moment.

7) Applying all kinds of accounting transactions.

8) Informing the customers about all the operations and results in isolated page.

b) Timeline: This system will be built within six months.

c) Resources: Old data, users, stakeholders, developers and customers. After completing this case study, I use the appreciative inquiry method, another technique is adopted to study the same case, to prove the efficiency of the appreciative inquiry in finding unique requirements, as against the other techniques: It is practically impossible for the researchers to gather users' requirements, just with the observation technique alone, so the interview technique is used in association with the observation, to obtain the user requirements as much as possible, by consuming the same amount of time used to perform the appreciative inquiry processes. Only four user requirements have been obtained. Table 24 below illustrates the results of the case study.

TABLE 24: Techniques comparison

Attribute / Technique	**Appreciative Inquiry**	**Observation**	**Interview**	**Observation & Interview**
User participation	Yes, Too Much	No	Yes	Yes
Unique requirements	Yes	Some Times	Some Times	Some Times
Requirement number	8	0	3	4
Understand system	Yes, Too Much	No	Yes	Yes

The beneficial factors of the appreciative inquiry are: this approach is built upon the positive aspects of an organization or group, it recognizes factors, which are well

executed; therefore, a very effective and optimistic effect on the state of mind, assurance, and value of individuals and groups can be attained. Contributors are motivated, because they feel very much appreciated.

Appendix B

Pilot Study: IslamTag Social Network

The social network "www.islamtag.com" has been nominated to apply this case study to quantify security requirements as follow:

Firstly, both of vulnerabilities and errors indexes which result from calculating an input used to quantify or calculate security index. Secondly, a step of joining the Security Requirements (SR) with Security Index (SI) requires obtaining the quantification of security requirements which is called later the Security Requirements Index (SRI).

Calculate Vulnerabilities Index

Every software has its own vulnerabilities, which are considered as the weak points that allow attackers to either reduce a system's information assurance or deny the service. In this section, the vulnerabilities index that may happen in the system will be calculated. According to Khan (2008) the VI depends on:

1. Percentage of vulnerability occurrences in each software; i.e. every vulnerability maybe happen one time or more than one time in particular software (likelihood of each vulnerability being exploited).
2. Percentage of the damage that can happen because of this vulnerability to the software.
3. Percentage of the asset important in particular software. Every software has assistance and each assistance has a different value in importance, each single asset may be affected by at least one of the system vulnerabilities.

The security experts want to find all vulnerabilities effects (for every single vulnerability related to the system) in the system. Will define a universe U be described as follows

$U=\{V_1,V_2,V_3,V_4,V_5\}$.

Where V denote "vulnerabilities".

After interviewing the website developer "www.islamtag.com" and investigate all vulnerabilities related to this type of systems according to SANS list which supported by NIST, *Vulnerabilities* are:

1. (SQL Injection (*SQLI*).
2. OS Command Injection (*OSCI*)
3. Classic Buffer Overflow (*CLBO*)
4. Cross-site Scripting (*CR-sS*)
5. Download of Code Without Integrity Check (*DOCWIC*)).

$U=\{SQLI, OSCI, CLBO, CR\text{-}sS, DOCWIC\}$.

Let $E_u = \{E_u\}$ be a set of decision parameters related to the above universe U "vulnerabilities" according to the security experts.

$Eu = \{eu,1, eu,2, eu,3, eu,4, eu,5,\ldots eu,n\}$.

$Eu = \{e_{u,1}$= *occurrence one time*, $e_{u,2}$= *occurrence two times*, $e_{u,3}$= *high damage*, $e_{u,4}$= *low damage*, $e_{u,5}$= *effect on very important asset*, $e_{u,6}$= *effect on important asset*, $e_{u,7}$= *effect on medium asset*, $e_{u,8}$= *effect on less important asset*$\}$.

In this case, eight parameters (E_u) have been used and values that have been given according to the security experts' perspectives. For instance if a vulnerability X occurs two times in the system, it should have a high value- see $e_{u,2}$ and vice versa. In other cases, more or less parameters and different values can be adopted.

$$E = \left\{ \frac{e_{u,1}}{0.5}, \frac{e_{u,2}}{1}, \frac{e_{u,3}}{1}, \frac{e_{u,4}}{0.5}, \frac{e_{u,5}}{1}, \frac{e_{u,6}}{0.7}, \frac{e_{u,7}}{0.5}, \frac{e_{u,8}}{0.2} \right\}$$

Now, the function for each parameter i.e. F $(eu,1)$ will be found, which means the parameter $(eu,1)$ which is "occurrence one time" and it has (0.5) value in this system. According to security experts' opinion, all vulnerabilities that occur one time in this

system will be part of this function for instance (*SQLI, CLBO*) vulnerabilities appearing in this case study.

$F(e_{u,1})$ = F(Occurrence One Time) = {*SQLI, CLBO*}.
$F(e_{u,2})$ = F(Occurrence Two Times) = {*SQLI, OSCI, CR-sS*}.
$F(e_{u,3})$ = F(High Damage) = {*SQLI, CR-sS, DOCWIC*}.
$F(e_{u,4})$ = F(Low Damage) = {*OSCI, CLBO*}.
$F(e_{u,5})$ = F(Effect On Very Important Asset) = {*SQLI*}.
$F(e_{u,6})$ = F(Effect On Important Asset) = {*OSCI, CLBO*}.
$F(e_{u,7})$ = F(Effect On Medium Asset) = {Ø}.
$F(e_{u,8})$ = F(Effect On Less Important Asset) = {*CR-sS, DOCWIC*}.

Then, the fuzzy soft set (F, E) can be viewed as consisting of the following collection of approximations:

(F, E) = {($e_{u,1}$, ({*SQLI, CLBO*})), ($e_{u,2}$, ({*SQLI, OSCI, CR-sS*})),
($e_{u,3}$, ({*SQLI, CR-sS, DOCWIC*})), ($e_{u,4}$, ({*OSCI, CLBO*})),
($e_{u,5}$, ({*SQLI*})), ($e_{u,6}$, ({*OSCI, CLBO*})), ($e_{u,7}$, ({Ø})), ($e_{u,8}$, ({*CR-sS, DOCWIC*}))}.

The Cagman, Citak and Enginoglu Algorithm (CCEA) applied to the first fuzzy soft part in (F, E) to take the decision from the availability set U incorporating the choice values.

$$\mu_F{}^d{}_X = \frac{1}{5}[\mu_X(e_{U,1})_{xfx}(e_{U,1})(V_1) + \mu_X(e_{U,2})_{xfx}(e_{U,2})(V_1) + \mu_X(e_{U,3})_{xfx}(e_{U,3})(V_1) + \mu_X(e_{U,4})_{xfx}(e_{U,4})(V_1) + \mu_X(e_{U,5})_{xfx}(e_{U,5})(V_1) + \mu_X(e_{U,6})_{xfx}(e_{U,6})(V_1) + \mu_X(e_{U,7})_{xfx}(e_{U,7})(V_1) + \mu_X(e_{U,8})_{xfx}(e_{U,8})(V_1)]$$

$$\mu_F{}^d{}_X = \frac{1}{5}[(0.5 \times 1) + (1 \times 1) + (1 \times 1) + (0.5 \times 0) + (1 \times 1) + (0.7 \times 0) + (0.5 \times 0) + (0.2 \times 0)]$$

$$= \frac{1}{5}[0.5 + 1 + 1 + 1] = 0.7$$

$$\mu_F{}^d{}_X = \frac{1}{5}[\,\mu_X(e_{U_1})_{\forall fx}(e_{U_1})(V_2) + \mu_X(e_{U_2})_{\forall fx}(e_{U_2})(V_2) + \mu_X(e_{U_3})_{\forall fx}(e_{U_3})(V_2) + \mu_X(e_{U_4})_{\forall fx}(e_{U_4})(V_2) + \mu_X(e_{U_5})_{\forall fx}(e_{U_5})(V_2) + \mu_X(e_{U_6})_{\forall fx}(e_{U_6})(V_2) + \mu_X(e_{U_7})_{\forall fx}(e_{U_7})(V_2) + \mu_X(e_{U_8})_{\forall fx}(e_{U_8})(V_2)\,]$$

$$\mu_F{}^d{}_X = \frac{1}{5}[(0.5 \times 0) + (1 \times 1) + (1 \times 0) + (0.5 \times 1) + (1 \times 0) + (0.7 \times 1) + (0.5 \times 0) + (0.2 \times 0)]$$

$$= \frac{1}{5}[1 + 0.5 + 0.7] = 0.44$$

And so on for all vulnerabilities.

Then ${}_F{}^d{}_X$ is represented by.

$${}_F{}^d{}_X = \left\{\frac{0.7}{V_1}, \frac{0.44}{V_2}, \frac{0.34}{V_3}, \frac{0.44}{V_4}, \frac{0.24}{V_5}\right\}$$

i.e. the *Fuzzy* decision for this vulnerabilities (VI) are:

1. SQL Injection (*SQLI*) = 0.7
2. OS Command Injection (*OSCI*) = 0.44
3. Classic Buffer Overflow (*CLBO*) = 0.34
4. Cross-site Scripting (*CR-sS*) = 0.44
5. Download of Code Without Integrity Check (*DOCWIC*) = 0.24

The five vulnerabilities above are part of the most dangerous software and web applications' vulnerabilities in the SANS and NIST institute list (NIST, 8 September 2013; KLOCWORK, 12 November 2013). It has been noted that the maximum value of $\mu FdX = 0.7$ among all vulnerabilities; this means that the *SQL Injection* vulnerability receives the most influence in the proposed website under this parameters (*Eu*), as well as in SANS and NIST institute list *SQL Injection* is the most dangerous software vulnerability. On other hand, the errors that cause these vulnerabilities have a direct impact on the web applications. So, the next sub section will discuss this issue in further details.

Calculate Errors Index

Every single error or more led to one vulnerability or more (M. Khan & Zulkernine, 2008), which is manipulated by the attackers to control the system, and to read, modify or destroy it. In this sub section the Errors Index (EI) that may be available in the system will be quantified. According to Khan (2008) the EI depends on:

1. Percentage of error occurrences in each software; i.e. every error may happen one time or more than one time in particular software (likelihood of each single error being exploited).
2. Percentage of the error's effects; i.e. every single error may affect one vulnerability or more, for instance, the percentage of effects error *X* which is led to three different vulnerabilities more than of the percentage of effects error *Y* that led to one vulnerability.
3. Percentage of the error dangers in particular software. Every software has vulnerabilities, every single vulnerability influenced by one or more errors that affects different levels of the danger on vulnerabilities.

Now, the same algorithm (3) is recalled, as it is used in the previous sub section to calculate the EI. The security experts want to find all errors effects (for every single errors related to the system) in the system. A universe U will be described as follows according to SANS and NIST list (NIST, 8 September 2013; KLOCWORK, 12 November 2013).

$U=\{R_1, R_2, R_3, R_4, R_5, R_6 R_7, R_8, R_9, R_{10}\}$.

Where R denote "errors".

After interviewing the website developer "www.islamtag.com" and investigating all errors related to this type of systems according to SANS and NIST list (NIST, 8 September 2013; KLOCWORK, 12 November 2013), *errors* are:

1. Poor SQL commands are used to check user names and passwords (*PSQL*).
2. The program that is executed allows arguments to be specified within an input file or from a standard input (*PEAASI*).

3. Code path includes a Buffer Write Operation (*CPBWO*).
4. Buffer is as large as you specify (*BLS*).
5. Replace unbounded copy functions with analogous functions that support length arguments (*RUCFSLA*).
6. Set the session cookie to be not only Http (*SSCH*).
7. Assume all inputs are not malicious (*AIM*).
8. Input Validation did not consider all potentially relevant properties (*IVPRP*).
9. Did not use proper output encoding, escaping, and quoting (*DPOEEQ*).
10. Encrypt the code with a reliable encryption scheme before transmitting (*ECREST*).

In this case a set of errors containing ten errors is used, depending on SANS and NIST lists (NIST, 8 September 2013; KLOCWORK, 12 November 2013), and system nature, in other cases more or less errors can be used, according to the system type. All errors used must have a relationship with the vulnerabilities which are calculated in previous sub section, otherwise, security requirements cannot be quantified..

U={*PSQL, PEAASI, CPBWO, BLS, RUCFSLA, SSCH, AIM, IVPRP, DPOEEQ, ECREST*}.

Let Eu ={Eu} be a set of decision parameters related to the above universe U "errors" according to the security experts.

Eu ={eu_1= *occurrence one time,* eu_2= *occurrence two times,* eu_3= *occurrence three times or more,* eu_4= *effect on one vulnerability,* eu_5= *effect on two vulnerabilities,* eu_6= *effect on three vulnerabilities or more,* eu_7= *effect on very dangerous vulnerability,* eu_8= *effect on dangerous vulnerability,* eu_9= *effect on normal vulnerability*}.

In this case nine parameters (E_u) have been used and values were given according to security experts' perspectives. For instance if an error X occurs three times in the

system, it should have a high value; see $e_{u,3}$ and vice versa. In other cases more or less than this number of parameters and different values.

$$E = \left\{ \frac{e_{u,1}}{0.33}, \frac{e_{u,2}}{0.66}, \frac{e_{u,3}}{1}, \frac{e_{u,4}}{0.33}, \frac{e_{u,5}}{0.66}, \frac{e_{u,6}}{1}, \frac{e_{u,7}}{1}, \frac{e_{u,8}}{0.66}, \frac{e_{u,9}}{0.33} \right\}$$

Now, the function for each parameter i.e. $F(e_{u,1})$ means the parameter $(e_{u,1})$ which is "*occurrence one time*" has (0.33) value in this system. According to the security experts' opinion, all errors that occur one time in this system will be part of this function like (*ECREST, PEAASI)* errors.

$F(e_{u,1})$ = F(Occurrence One Time) = {*ECREST, PEAASI*}.
$F(e_{u,2})$ = F(Occurrence Two Times) = {*CPBWO, IVPRP, DPOEEQ, AIM*}.
$F(e_{u,3})$ = F(Occurrence Three Times Or More) = {*PSQL, BLS, RUCFSLA, SSCH*}.
$F(e_{u,4})$ = F(Effect On One Vulnerability) = {*PSQL, CPBWO, BLS, RUCFSLA, SSCH, AIM, DPOEEQ, ECREST*}.
$F(e_{u,5})$ = F(Effect On Two Vulnerabilities) = {*PEAASI*}.
$F(e_{u,6})$ = F(Effect On Three Vulnerabilities Or More) = {*IVPRP*}.
$F(e_{u,7})$ = F(Effect On Very Dangerous Vulnerability) = {*PSQL, PEAASI, IVPRP*}.
$F(e_{u,8})$ = F(Effect On Dangerous Vulnerability) = {*PEAASI, IVPRP, SSCH, DPOEEQ, AIM*}.
$F(e_{u,9})$ = F(Effect On Normal Vulnerability) = {*CPBWO, BLS, RUCFSLA, ECREST*}.

Then, the fuzzy soft set (F, E) can be seen as consisting the following collection of approximations:

(F, E) = {($e_{u,1}$, ({*ECREST, PEAASI*})), ($e_{u,2}$, ({*CPBWO, IVPRP, DPOEEQ, AIM*})),
($e_{u,3}$, ({*PSQL, BLS, RUCFSLA, SSCH*})),
($e_{u,4}$, ({*PSQL, CPBWO, BLS, RUCFSLA, SSCH, AIM, DPOEEQ, ECREST*})),
($e_{u,5}$, ({*PEAASI*})),($e_{u,6}$, ({*IVPRP*})),($e_{u,7}$, ({*PSQL, PEAASI, IVPRP*})),
($e_{u,8}$, ({*PEAASI, IVPRP, SSCH, DPOEEQ, AIM* })),

$(e_{u,9}, (\{CPBWO, BLS, RUCFSLA, ECREST\}))$,}.

Cagman, Citak and Enginoglu Algorithm (CCEA) applied the first fuzzy soft part in (F, E) to take the decision from the availability set U The choice values are then incorporated.

$$\mu_F{}^d{}_X = \frac{1}{10}[\mu_X(e_{U_{-1}})_{\gamma fx}(e_{U_{-1}})(R_1) + \mu_X(e_{U_{-2}})_{\gamma fx}(e_{U_{-2}})(R_1) + \mu_X(e_{U_{-3}})_{\gamma fx}(e_{U_{-3}})(R_1) +$$
$$\mu_X(e_{U_{-4}})_{\gamma fx}(e_{U_{-4}})(R_1) + \mu_X(e_{U_{-5}})_{\gamma fx}(e_{U_{-5}})(R_1) + \mu_X(e_{U_{-6}})_{\gamma fx}(e_{U_{-6}})(R_1) +$$
$$\mu_X(e_{U_{-7}})_{\gamma fx}(e_{U_{-7}})(R_1) + \mu_X(e_{U_{-8}})_{\gamma fx}(e_{U_{-8}})(R_1) + \mu_X(e_{U_{-9}})_{\gamma fx}(e_{U_{-9}})(R_1)]$$

$$\mu_F{}^d{}_X = \frac{1}{10}[(0.33 \times 0) + (0.66 \times 0) + (1 \times 1) + (0.33 \times 1) + (0.66 \times 0) + (1 \times 0) +$$
$$(1 \times 1) + (0.66 \times 0) + (0.33 \times 0)]$$
$$= \frac{1}{10}[1 + 0.33 + 1] = 0.233$$

$$\mu_F{}^d{}_X = \frac{1}{10}[\mu_X(e_{U_{-1}})_{\gamma fx}(e_{U_{-1}})(R_2) + \mu_X(e_{U_{-2}})_{\gamma fx}(e_{U_{-2}})(R_2) + \mu_X(e_{U_{-3}})_{\gamma fx}(e_{U_{-3}})(R_2) +$$
$$\mu_X(e_{U_{-4}})_{\gamma fx}(e_{U_{-4}})(R_2) + \mu_X(e_{U_{-5}})_{\gamma fx}(e_{U_{-5}})(R_2) + \mu_X(e_{U_{-6}})_{\gamma fx}(e_{U_{-6}})(R_2) +$$
$$\mu_X(e_{U_{-7}})_{\gamma fx}(e_{U_{-7}})(R_2) + \mu_X(e_{U_{-8}})_{\gamma fx}(e_{U_{-8}})(R_2) + \mu_X(e_{U_{-9}})_{\gamma fx}(e_{U_{-9}})(R_2)]$$

$$\mu_F{}^d{}_X = \frac{1}{10}[(0.33 \times 1) + (0.66 \times 0) + (1 \times 0) + (0.33 \times 0) + (0.66 \times 1) + (1 \times 0) + (1$$
$$\times 1) + (0.66 \times 1) + (0.33 \times 0)]$$
$$= \frac{1}{10}[0.33 + 0.66 + 1 + 0.66] = 0.265$$

And so on for all errors.

Then ${}_F{}^d{}_X$ is represented by:

$${}_F{}^d{}_X = \left\{\frac{0.233}{R_1}, \frac{0.265}{R_2}, \frac{0.132}{R_3}, \frac{0.166}{R_4}, \frac{0.166}{R_5}, \frac{0.199}{R_6}, \frac{0.165}{R_7}, \frac{0.332}{R_8}, \frac{0.165}{R_9}, \frac{0.099}{R_{10}}\right\}$$

i.e. the *Fuzzy* decision for this errors (Errors Index) is established below:

1. Poor SQL commands are used to check user names and passwords (*PSQL*) = 0.233
2. The program that is executed allows arguments to be specified within an input file or from standard input (*PEAASI*) = 0.265

3. Code path includes a Buffer Write Operation (*CPBWO*) = 0.132
4. Buffer is as large as you specify (*BLS*) = 0.166
5. Replace unbounded copy functions with analogous functions that support length arguments (*RUCFSLA*) = 0.166
6. Set the session cookie to be not only Http (*SSCH*) = 0.199
7. Assume all input is not malicious (*AIM*) = 0.165
8. Input Validation did not consider all potentially relevant properties (*IVPRP*) = 0.332
9. Did not use proper output encoding, escaping, and quoting (*DPOEEQ*) = 0.165
10. Encrypt the code with a reliable encryption scheme before transmitting (*ECREST*) = 0.099

It is noted that the maximum value of $\mu_F{}^d{}_X = 0.332$ among all errors; meaning that "*Input Validation did not consider all potentially relevant properties*" error receives the most influence in the website under this parameters (E_n) and that all of the above ten errors are part of SANS and NIST error list (NIST, 8 September 2013; KLOCWORK, 12 November 2013).

The errors index causes the system's vulnerabilities. Vulnerabilities and errors have a direct impact on this web application, but all of them need to find security index and security requirements index. So, the next sub-section will discuss in depth both the SI and security requirements index.

Calculate Security Index

After calculating both of the vulnerability and error indexes which are related to the web application in this case study, the next step is to calculate the SI, then finally, to connect SI contents with the security requirements to obtain the result, which is the security requirements index. There is a relationship between error and vulnerability indexes; this relationship gives us the security index. Using below equation the SI can be calculated.

$$SI^{Vi} = \sum_{i,j \in I} \frac{V_i \; E_j}{j}$$

Where;

SI^{Vi} : is security index for vulnerability i .

Now, for calculating the SI, need VI and EI shall be calculated in these last two sub sections.

Vulnerabilities Index:

1. SQL Injection (*SQLI*) = 0.7
2. OS Command Injection (*OSCI*) = 0.44
3. Classic Buffer Overflow (*CLBO*) = 0.34
4. Cross-site Scripting (*CR-sS*) = 0.44
5. Download of Code Without Integrity Check (*DOCWIC*) = 0.24

Errors Index:

1. Poor SQL commands are used to check user names and passwords (*PSQL*) = 0.233
2. The program that is executed allows arguments to be specified within an input file or from standard input (*PEAASI*) = 0.265
3. Code path includes a Buffer Write Operation (*CPBWO*) = 0.132
4. Buffer is as large as you specify (*BLS*) = 0.166
5. Replace unbounded copy functions with analogous functions that support length arguments (*RUCFSLA*) = 0.166
6. Set the session cookie to be not only Http (*SSCH*) = 0.199
7. Assume all input is not malicious (*AIM*) = 0.165
8. Input Validation did not consider all potentially relevant properties (*IVPRP*) = 0.332
9. Did not use proper output encoding, escaping, and quoting (*DPOEEQ*) = 0.165
10. Encrypt the code with a reliable encryption scheme before transmitting (*ECREST*) = 0.099

Every single vulnerability consists of one error or more, and to calculate the SI for any security requirements the relationships between vulnerabilities and errors, the SI which must be found uses equation (5). Now, the SI will be located by using both the VI and EI as follows:

$SQLI = \{ PSQL, PEAASI, IVPRP \}$
$OSCI = \{ PEAASI, IVPRP \}$
$CLBO = \{ CPBWO, BLS, RUCFSLA \}$
$CR\text{-}sS = \{ SSCH, AIM, IVPRP, DPOEEQ \}$
$DOCWIC = \{ ECREST \}$

This indicates that the "SQL Injection (*SQLI*)" vulnerability consists of "Poor SQL commands which are used to check user names' and passwords' (*PSQL*)" error, "The program be executed allows arguments to be specified within an input file or from standard input (*PEAASI*)" error and "Input Validation did not consider all potentially relevant properties (*IVPRP*)" error. This applies to all vulnerabilities as mentioned above. Now the security index for each vulnerability alone using equation (5) is calculated as follows:

$$SI^{Vi} = \sum_{i,j \in I} \frac{V_i \; E_j}{j}$$

$$SI^{SQLI} = \frac{SQLI \times PSQL + SQLI \times PEAASI + SQLI \times IVPRP}{3}$$

$$= \frac{(0.7 \times 0.233) + (0.7 \times 0.265) + (0.7 \times 0.332)}{3}$$

$$= \frac{0.1631 + 0.1855 + 0.2324}{3}$$

$$= \frac{0.581}{3}$$

$$SI^{SQLI} = 0.1936$$

$$SI^{OSCI} = \frac{OSCI \times PEAASI + OSCI \times IVPRP}{2}$$

$$= \frac{(0.44 \times 0.265) + (0.44 \times 0.332)}{2}$$

$$SI^{OSCI} = 0.1313$$

$$SI^{CLBO} = \frac{CLBO \times CPBWO + CLBO \times BLS + CLBO \times RUCFSLA}{3}$$

$$= \frac{(0.34 \times 0.132) + (0.34 \times 0.166) + (0.34 \times 0.166)}{3}$$

$$SI^{CLBO} = 0.0525$$

$$SI^{CR\_sS} = \frac{CR\_sS \times SSCH + CR\_sS \times AIM + CR\_sS \times IVPRP + CR\_sS \times DPOEEQ}{4}$$

$$= \frac{(0.44 \times 0.199) + (0.44 \times 0.165) + (0.44 \times 0.332) + (0.44 \times 0.165)}{4}$$

$$SI^{CR\_sS} = 0.0947$$

$$SI^{DOCWIC} = \frac{DOCWIC \times ECREST}{1}$$

$$= \frac{0.24 \times 0.099}{1}$$

$$SI^{DOCWIC} = 0.0237$$

This works in the same way for all vulnerabilities. Concerning the security index for all the vulnerabilities, fuzzy numbers means that when the number approaches to one, the seriousness of these vulnerability increases; so it should be removed to a high priority, and vice versa. The vulnerabilities are arranged according to their dangerous degrees as follows:

Security Index:

1. SQL Injection (*SQLI*) = 0.1936
2. OS Command Injection (*OSCI*) = 0.1313
3. Cross-site Scripting (*CR-sS*) = 0.0947
4. Classic Buffer Overflow (*CLBO*) = 0.0525
5. Download of Code Without Integrity Check (*DOCWIC*) = 0.0237

Security Requirements Index

At the end of the issue, all SI contents will be connected with all security requirements, which cover all software/business requirements (Assets). So, each SI should be connected with related security requirements; sometimes such security requirements connects two or more of SI contents.

1. Command Injection Flaws: (*SQLI*) and (*OSCI*) = 0.3249
2. Data and Input Validation: (*SQLI*) and (*OSCI*) = 0.3249
3. OS Command Injection Flaws: (*OSCI*) = 0.1313
4. Cross Site Scripting (*XSS*): (*CR-sS*) = 0.0947
5. Buffer Overflows: (*CLBO*) = 0.0525
6. Authentication: (*DOCWIC*) = 0.023

Appendix C

Sequence Diagram and Collaborative Diagram for Real Case Study: Online E-Business Website (PRICE).

FIGURE 31: Sequence diagram of login use case.

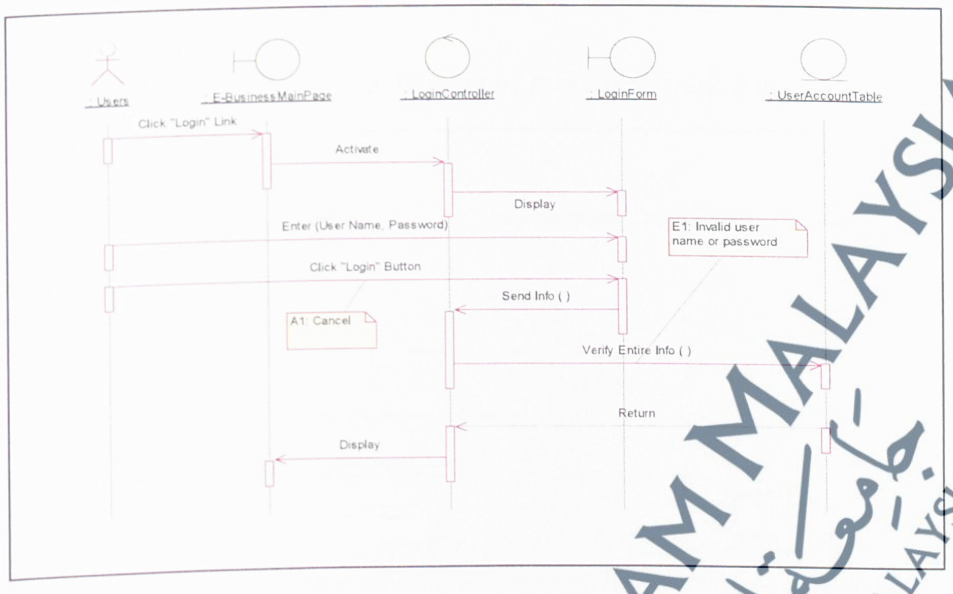


FIGURE 32: Collaborative diagram of login use case.

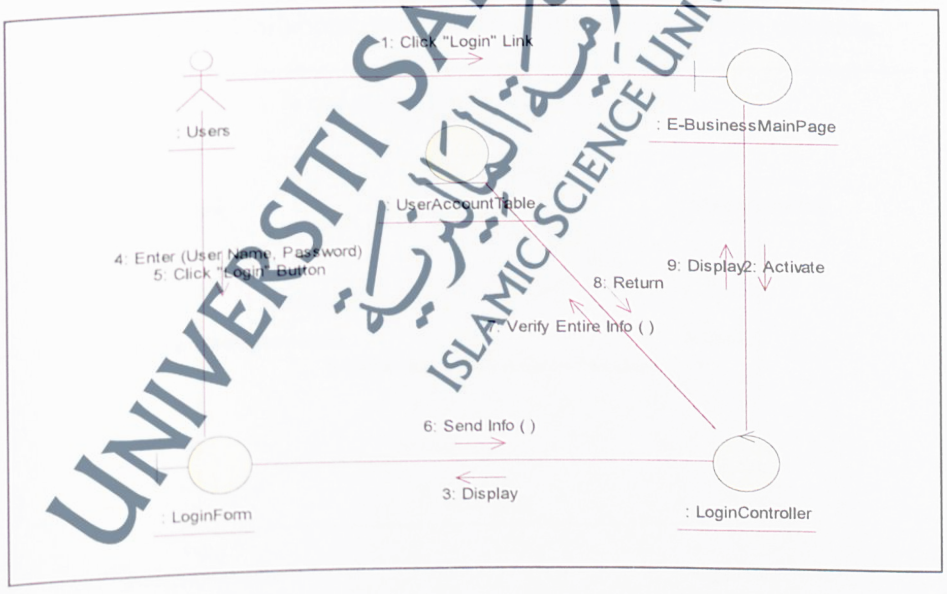


FIGURE 33: Sequence diagram of make registration use case.

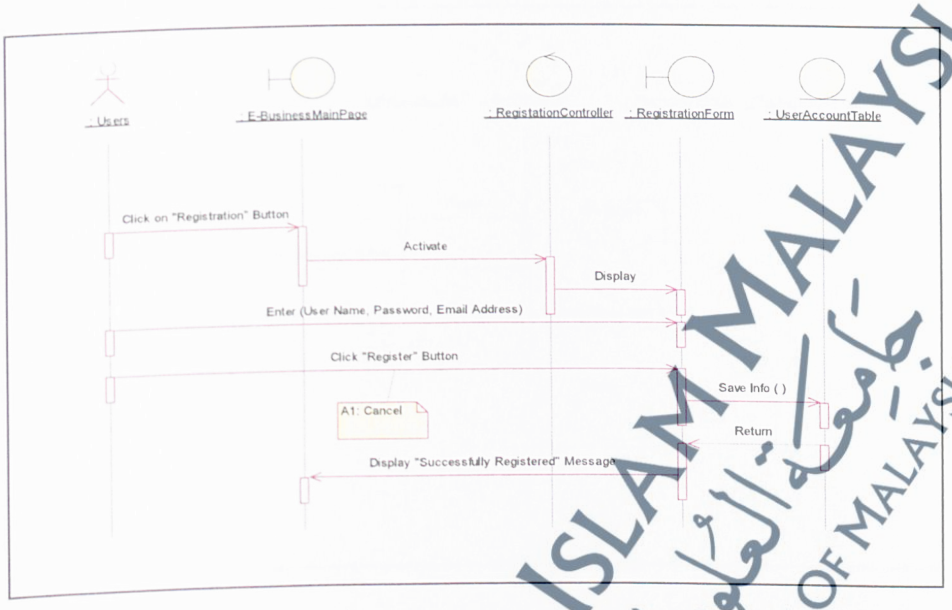


FIGURE 34: Collaborative diagram of make registration use case.

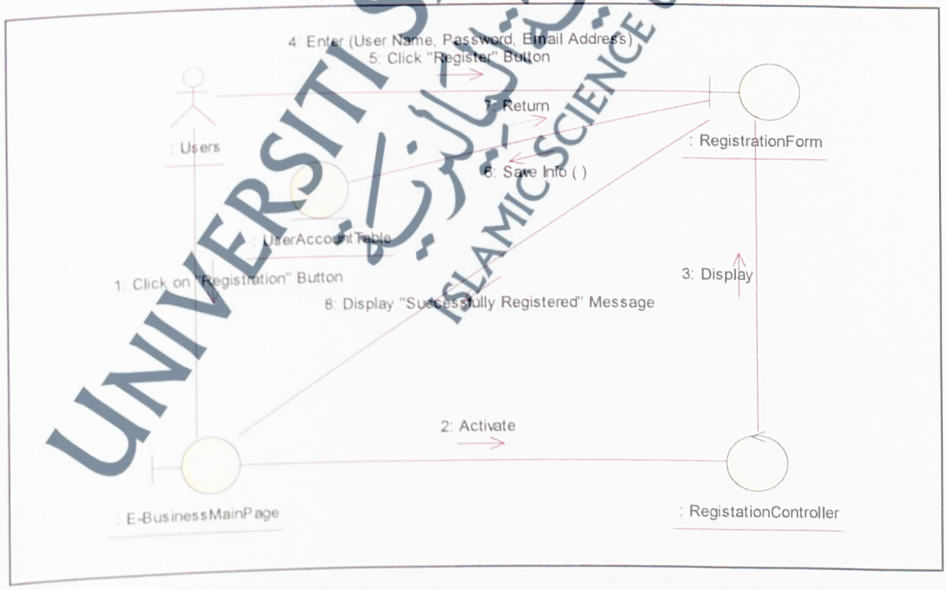


FIGURE 35: Sequence diagram of purchase services use case.

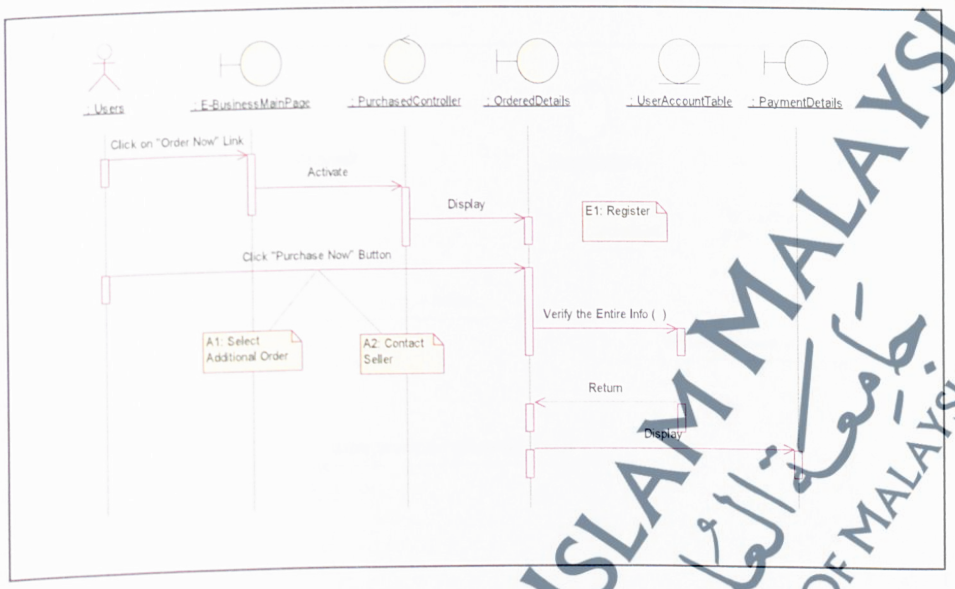


FIGURE 36: Collaborative diagram of purchase services use case.

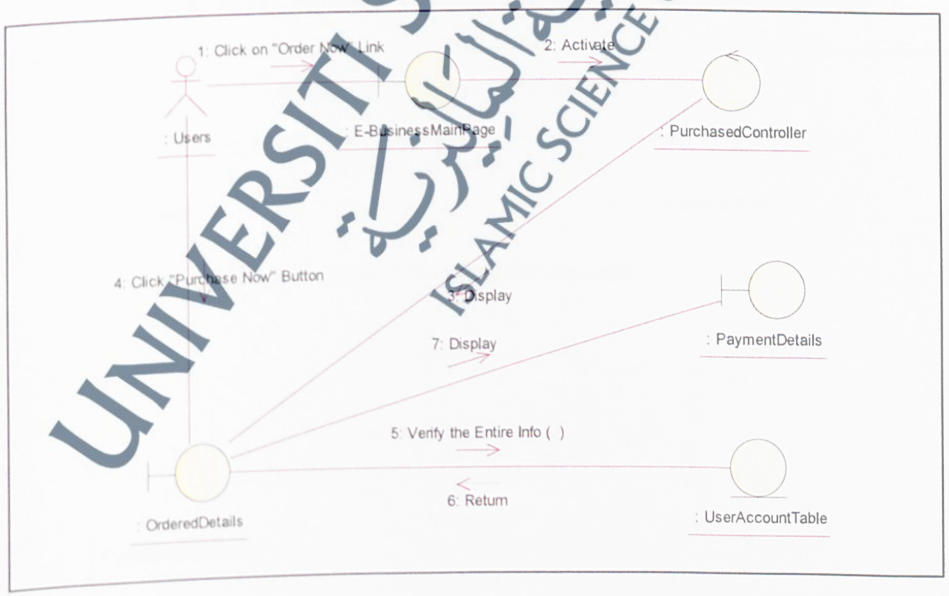


FIGURE 37: Sequence diagram of make payment use case.

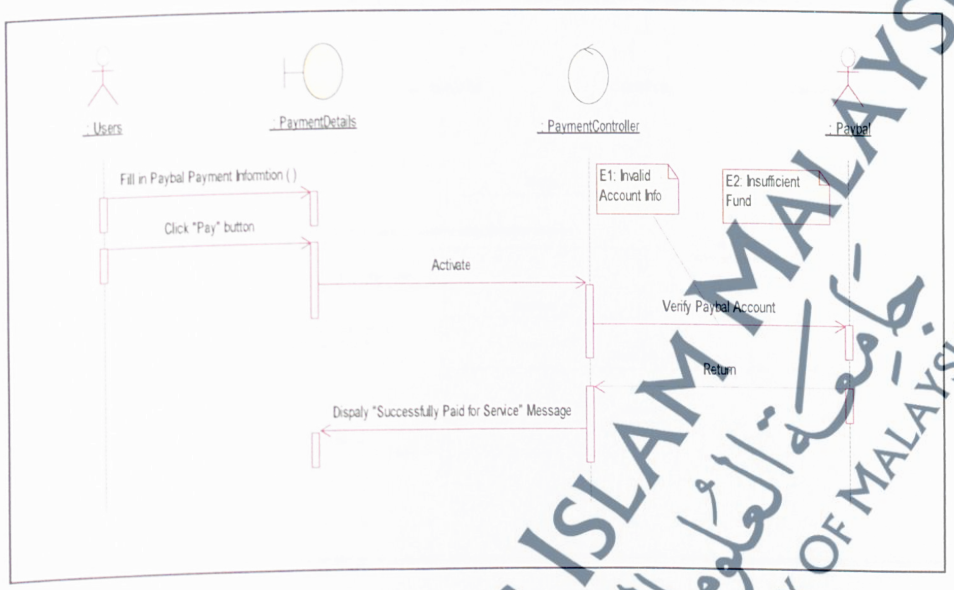


FIGURE 38: Collaborative diagram of make payment use case.

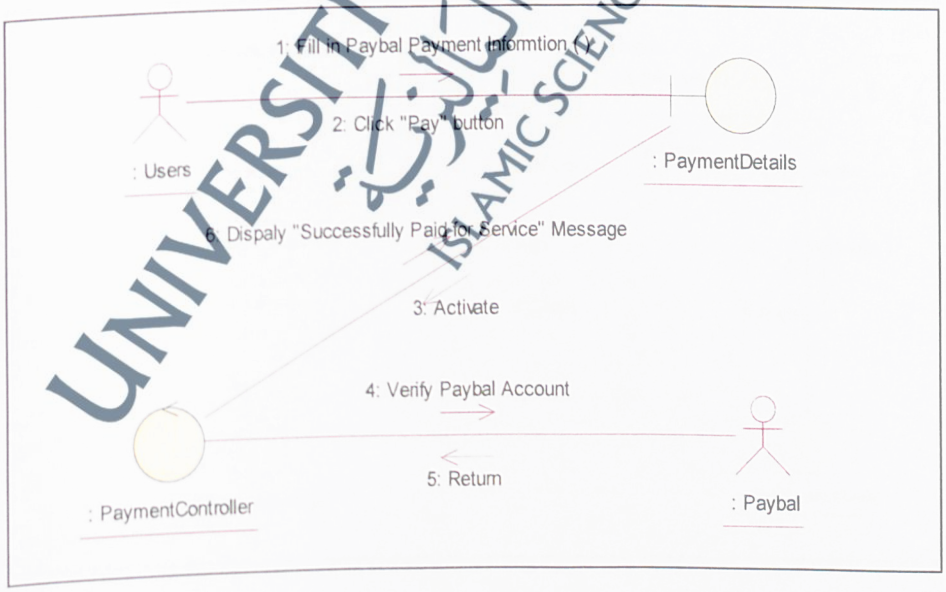


FIGURE 39: Sequence diagram of brute force login use case.

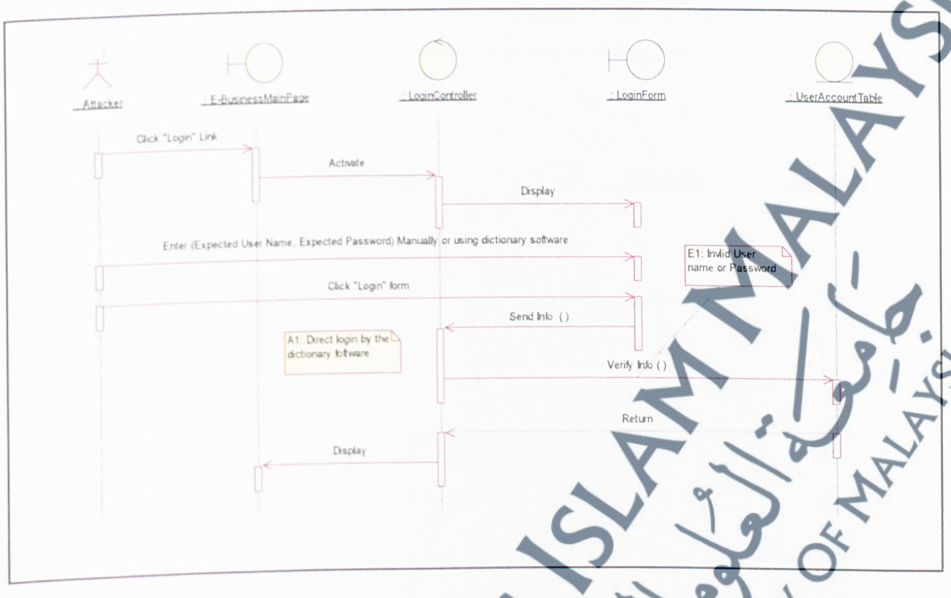


FIGURE 39: Collaborative diagram of brute force use case.

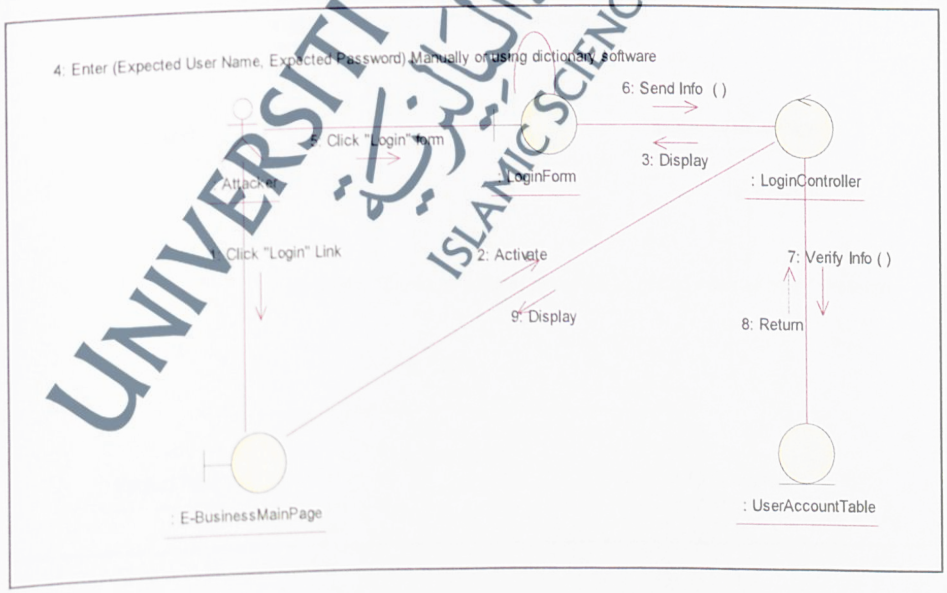


FIGURE 40: Sequence diagram of malicious code injection use case.

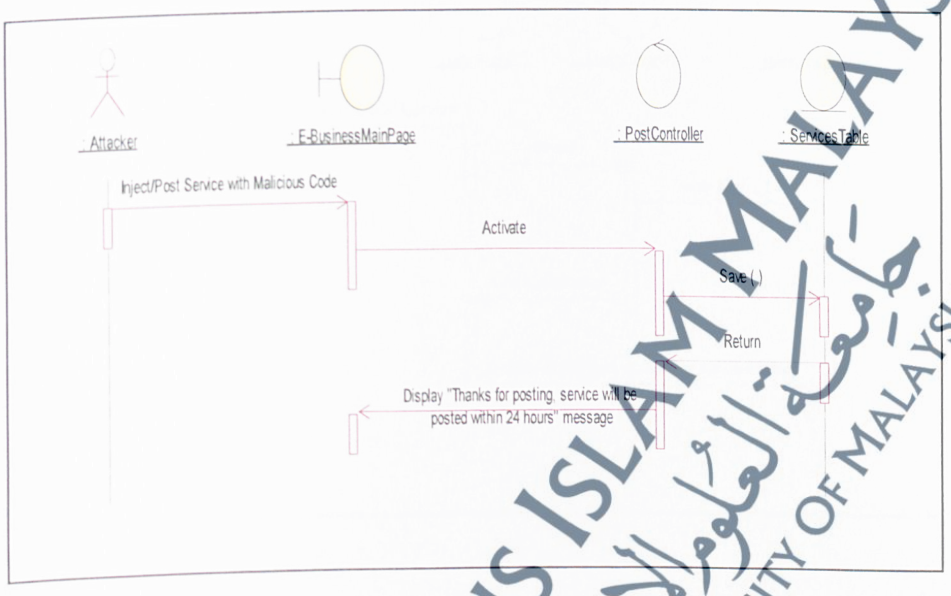


FIGURE 41: Collaborative diagram of malicious code injection use case.

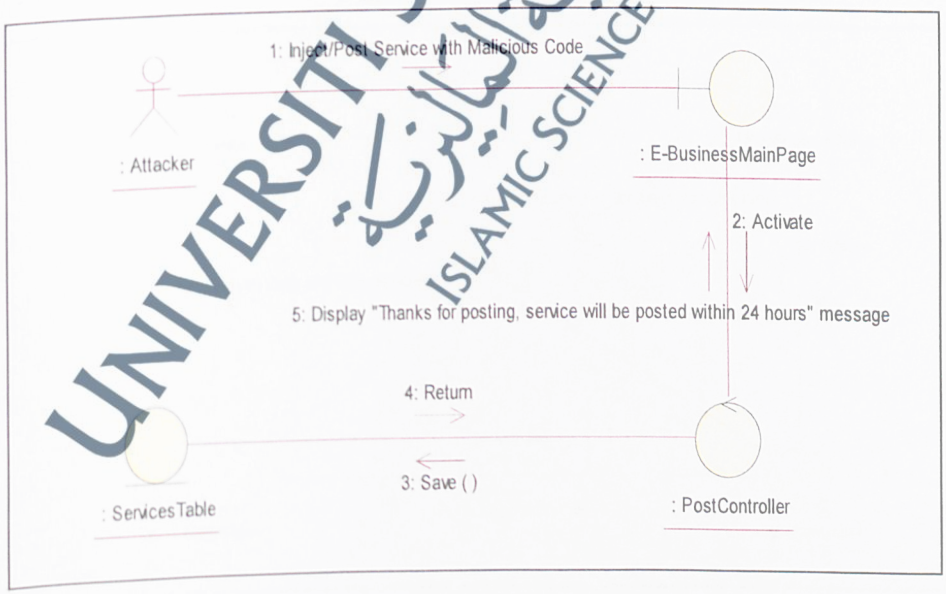


FIGURE 42: Sequence diagram of disclose user information use case.

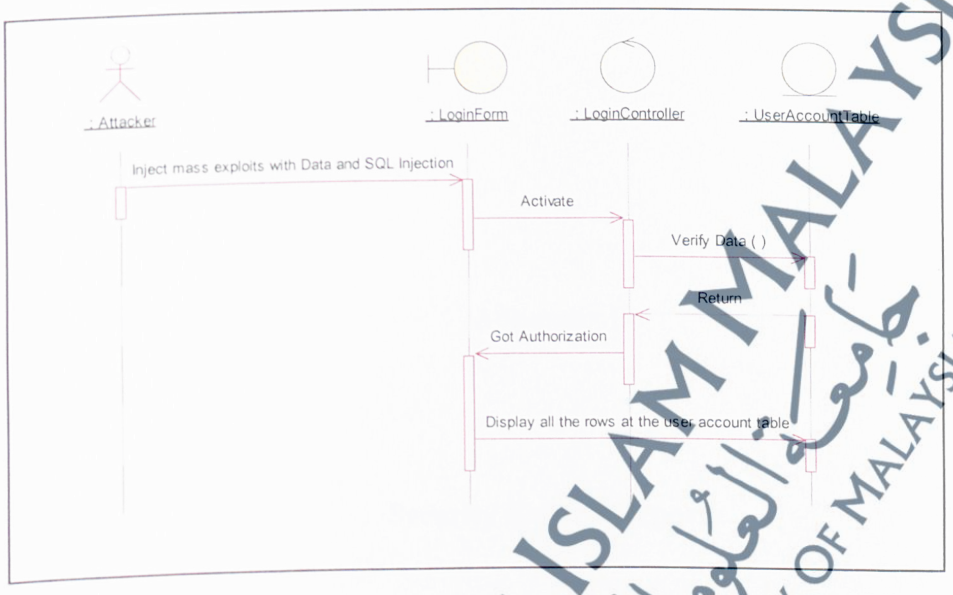
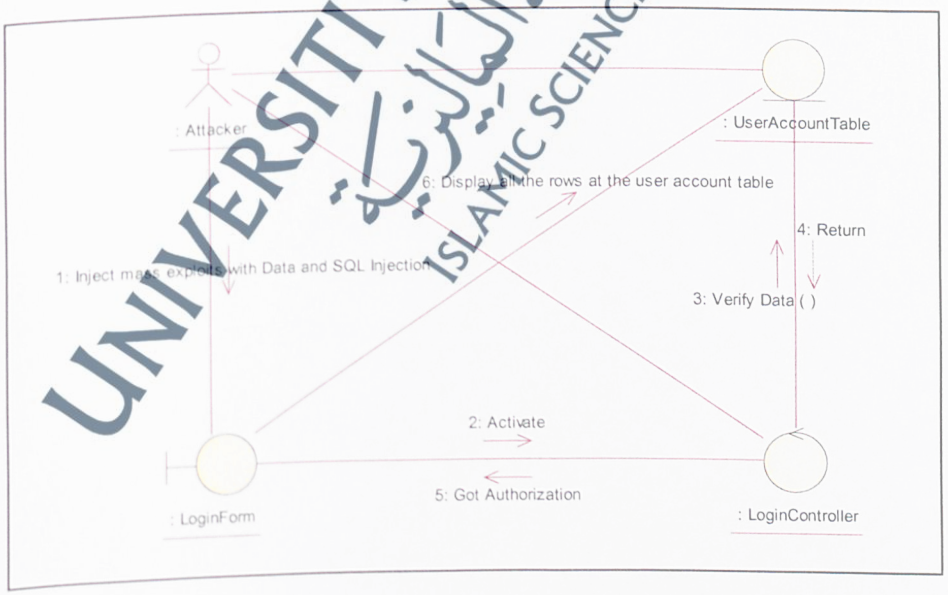


FIGURE 43: Collaborative diagram of disclose user information use case.



Appendix D

Security Experts Reports

1) Dr. KAMARUDIN BIN SAADAN

Senior Fellow

Faculty of Sciene and Technology

Universiti Sains Islam Malaysia

Date : 2 January 2014
Time : 2 pm
Venue : FST, A2-055

Comments:

Based on my understanding on the write-up on the topics the research was quite interesting and the content/steps of the SAIFQT cover all security activities, which help developers to capture security requirements. The proposed technique SAIFQT achieves the policy aspects in the first phase (Discovery Phase) besides eliciting security requirements this point is important and it is one of the benefits of this technique. Another benefit makes risk assessment the proposed system before building it.

The following are some of the points to be considered to improve the proposed technique: Information on the outcome of the study should be highlighted in literature review to justify the selection of SQUARE and CLASP in this new technique.

The algorithm should be outlined (at least framework). How the formulae 1,2,3, 4 (Cagman et al. 2011) mentioned in the write-up are used or utilized in the algorithm should be highlighted and explained clearly by applying a case study. This is important since this part is the main contribution to the research. The intelligent characteristics of this technique should be also highlighted and explained.

2) Dr. MADIHAH BINTI MOHD SAUDI

Faculty of Sciene and Technology

Universiti Sains Islam Malaysia

Date : 9 January 2014
Time : 10 am
Venue : FEM, A2-061

Comments:
It is a good technique, because it elicits and quantifies security requirements at the same time with using the same technique especially, the process of quantifying security requirements before building the system, this unique feature is one of the most important and useful features in this proposed technique SAIFQT. Overall, the research was quite interesting and the contribution is very good.

3) Dr. MOHD ZALISHAM BIN JALI

Faculty of Sciene and Technology

Universiti Sains Islam Malaysia

Date : 9 January 2014
Time : 3 pm
Venue : UKP, A2-PIPP

Comments:

The main step which is eliciting security requirements in conjunction with eliciting software/business requirements have done in logically way and correct strategy in the SAIFQT, because the need for security aspects was to protect business requirements. So, no need to conduct extra non-useful security. Thus, need extra time, efforts and money (waste extra resources). The SAIFQT succeeded in eliciting security requirements without wasting extra resources. on the other hand, even after eliciting all security requirements, sometimes the system owners can't pay too much of money to cover all elicited security requirements, in this case the solution is security requirements prioritization aspect which is taken into account of the SAIFQT, to choose the most dangerous vulnerabilities which must be covered in the system using this aspect (security requirements prioritization).

4) Dr. Aysh Alhroob

Faculty of Science and Information Technology

Isra University, Jordan, Amman

Head of Software Engineering

Aysh@ipu.edu.jo

00962775701222

Date : 30 January 2014
Time : 6 pm
Venue : Skype and E-mail

Evaluation Report

It is a good idea to integrate Security Quality Requirements Engineering (SQUARE) and Comprehensive, Lightweight Application Security Process (CLASP) to improve both of them. The integration makes it easy to minimize the illegitimate requirements that will influence in a bad manner. The Appreciative Inquiry method (AI) technique presents a good logic in the result but I need more clarifications about the relation between the previous work and the proposed technique.

The Author talked about elicits user requirements, elicit security requirements and quantify security requirements as a result to the proposed technique. If all expected results are founded in the end of the work, this means the proposed work succeeds to introduce the aims, but I am not sure that it could be, these doubts came from the lack of clarity of the approach presentation. So, there is a need to conduct a case study by using this technique to prove the final results.

Comments:

1- The introduction should introduce the problem, why it is important, what exists is not sufficient, what problems will be addressed (and therefore what are the contributions).

2- The introduced results seem good enough, but no clear prove to get such a like result (need to conduct a case study).

3- The figure introduces the proposed technique in a good manner, but no clear technique is used to identify how the "what" questions are performed (need to conduct a case study).

5) Dr. Lena Khalid Ahmad

Department of Software Engineering

Faculty of Information Technology

Zarqa University, Jordan, Amman

Editor in the Journal of Communications and Information Sciences, which is related to the Advanced Institute of Convergence IT. Member of the International Board of Reviewers of the Journal of Information Technology Education and one of the committee members of the International Arab Journal of Information Technology (IAJIT).

lena_khaled@yahoo.com

Date : 9 February 2014
Time : 4 pm
Venue : Skype and E-mail

Comments:

Strength

I think that the research has a good and a new idea if the methodology is described well and then applied. It will be then very useful for those who want to know the security of the system from the beginning (from the requirement stage).

Suggestions

Despite the fact that I am interested in the idea for this technique I think it needs to answer some questions like:

- What are the main factors that the developer needs to make decision on the most appropriate security requirements?
- If the idea works on business requirements then it must take into account how to measure the cost. It must show the reason for choosing this specific development method (the waterfall)?

6) Msc. Saleh Atiewi. and Zaid alhalhouli

Computer Center, Programmer

Al Houssain Ben Talal University, Jordan, Maan

atiewi@yahoo.com

Date : 15 February 2014
Time : 5 pm
Venue : Skype and E-mail

Comments:

Login Script with Various Functionalities

We have used three fields Email, Password and Google recaptcha:

User enters his email, password, and recaptcha text ,where we uses validations both server as well as client side validations:

1. Client side validations uses javascript. We have used email field for javascript validation it saves time and bandwidth. People who visit your site may use an old version or may use a disabled javascript which will break client's validation client and server-side validations that complement each other, and as such, they really should not use independently.

2. Server Side Validation: Checks that user is approved or not. We have used Pwd Hash function to decrypt the password. That it is a script to verify that good values have been sent to the script. It is more secure and works seamlessly with all browsers, but it does so at the cost of slightly higher server load and slower feedback for users.

3. Cookies: To remember password at user machine. We have created cookies:
   - User id
   - User key
   - User name

Defines a cookie to be sent along with the rest of the HTTP headers. Like other headers, cookies must be sent *before* any output from your script (this is a protocol restriction).

7) Msc. Zaid alhalhouli

Computer Center, Programmer

Tafelah Technical University, Jordan, Tafelah

Zaid_halhouli@yahoo.com

Date : 17 February 2014
Time : 9 pm
Venue : Skype and E-mail

Comments:

When you are working with an application, you open it, do some changes and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. However, on the internet there is one problem: the web server does not know who you are and what you do because the HTTP address does not maintain state.

A PHP session solves this problem by allowing you to store user information on the server for later use (i.e. username, shopping items, etc). However, session information is temporary and will be deleted after the user has left the website. If you need a permanent storage, you may want to store the data in a database.

Filter for inputs: We have uses trim, get_magic_quotes_gpc, stripslashes and mysql_real_escape_string functions to filter input fields. A PHP filter is used to validate and filter data coming from insecure sources. To test, validate and filter user input or custom data is an important part of any web application. The PHP filter extension is designed to make data filtering easier and quicker.

Google recaptcha: To prevent brute force login. Brute force attack will exponentially take longer and longer. It could lock a user output it will not count a failed login whilst trying to login during the lockout period. This should minimize it.
Register field contains Name, Address, Account Type, Country, Fax, Website, Phone, Username, password, Confirm Password, Email and Image verification fields to prevent brute force login. Which created field on database users table.

1. User id.
2. Md5_id (password encryption form).
3. Full name.
4. User name.
5. Telephone.
6. Fax.
7. Address.
8. User email.
9. User level (checks user level for login).
10. Password.
11. Country.
12. Date.
13. User IP (system IP).
14. Approved (approved user or not).
15. Activation code (will send email to users to activate his account).
16. Banned (admin can ban users).
17. Ckey (login key).
18. Ctime (user login time).
19. Account type fields (customer or seller).

PayPal gateway used is independent of the website system. It takes user to PayPal system protected with SSL layer. Once user makes the purchase, he comes back to the website URL set under 'return' parameter.

Appendix E

Penetration Test for the Two Prototypes

There are two penetration-testing reports for the two prototypes in this section; the first one is associated with the prototype, which is built under the proposed technique SAIFQT, and the second report has to do with the second prototype, which is built using normal SDLC.

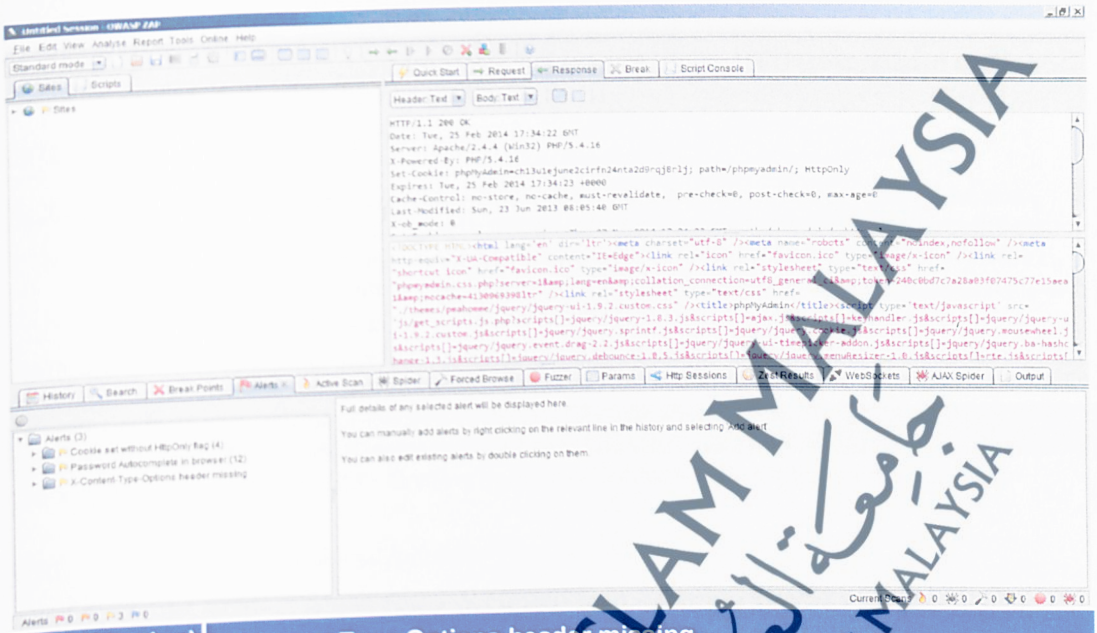
The tool that proves to be helpful for this penetration test is OWASP ZAP version 2.2.2, used to assess the integrated technique. The two reports below show the extent to which the website (prototype) built using the integrated technique is well-protected following the NIST and SANS Standards (NIST, 8 September 2013; KLOCWORK, 12 November 2013).

Here is the specification of the system:

PHP 5.2.4 or greater

MySQL 5.0 or greater

1) Report for Prototype Built Under SAIFQT.



Low (Warning)	X-Content-Type-Options header missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'
URL	http://localhost
Solution	This check is specific to Internet Explorer 8 and Google Chrome. Ensure each page sets a Content-Type header and the X-CONTENT-TYPE-OPTIONS if the Content-Type header is unknown
Reference	

Informational (Warning)	X-Frame-Options header not set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks
URL	http://localhost
Solution	Most modern Web browsers support the X-Frame-Options HTTP header, ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY).
Reference	http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx?Redirected=true

Low (Warning)	X-Content-Type-Options header missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'
URL	http://localhost/?lang=fr
Solution	This check is specific to Internet Explorer 8 and Google Chrome. Ensure each page sets a Content-Type header and the X-CONTENT-TYPE-OPTIONS if the Content-Type header is unknown

Reference	

Informational (Warning)	X-Frame-Options header not set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks
URL	http://localhost/?lang=fr
Solution	Most modern Web browsers support the X-Frame-Options HTTP header, ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY).
Reference	http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx?Redirected=true

Low (Warning)	X-Content-Type-Options header missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'
URL	http://localhost/?phpinfo=1
Solution	This check is specific to Internet Explorer 8 and Google Chrome. Ensure each page sets a Content-Type header and the X-CONTENT-TYPE-OPTIONS if the Content-Type header is unknown
Reference	

Informational (Warning)	X-Frame-Options header not set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks
URL	http://localhost/?phpinfo=1
Solution	Most modern Web browsers support the X-Frame-Options HTTP header, ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY).
Reference	http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx?Redirected=true

Low (Warning)	Cookie set without HttpOnly flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	http://localhost/sqlbuddy/
Parameter	PHPSESSID=1t8q3k43qkafnhhi2gt596tka5; path=/
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	www.owasp.org/index.php/HttpOnly
WASC Id	13

Low (Warning)	Password Autocomplete in browser

Description	AUTOCOMPLETE attribute is not disabled in HTML FORM/INPUT element containing password type input. Passwords may be stored in browsers and retrieved.
URL	http://localhost/phpmyadmin/
Parameter	Input
Attack	<input type="password" name="pma_password" id="input_password" value="" size="24" class="textfield" />
Solution	Turn off AUTOCOMPLETE attribute in form or individual input elements containing password by using AUTOCOMPLETE='OFF'
Reference	http://msdn.microsoft.com/library/default.asp?url=/workshop/author/forms/autocomplete_ovr.asp
CWE Id	525

Low (Warning)	**X-Content-Type-Options header missing**
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'
URL	http://localhost/?lang=en
Solution	This check is specific to Internet Explorer 8 and Google Chrome. Ensure each page sets a Content-Type header and the X-CONTENT-TYPE-OPTIONS if the Content-Type header is unknown
Reference	

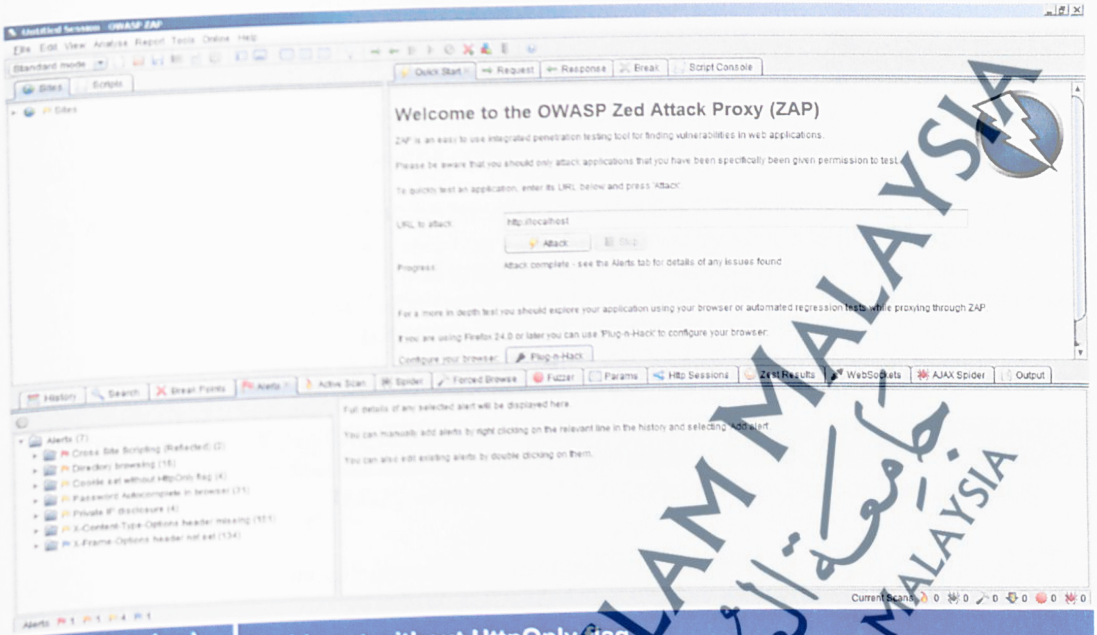
Informational (Warning)	**X-Frame-Options header not set**
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks
URL	http://localhost/?lang=en
Solution	Most modern Web browsers support the X-Frame-Options HTTP header, ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY).
Reference	http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx?Redirected=true

Low (Warning)	**Cookie set without HttpOnly flag**
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	http://localhost/sqlbuddy/login.php
Parameter	PHPSESSID=k9smnuap7k3sf12nkekeeu3le7; path=/
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	www.owasp.org/index.php/HttpOnly
WASC Id	13

Low (Warning)	**Cookie set without HttpOnly flag**

Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	http://localhost/sqlbuddy/login.php
Parameter	PHPSESSID=digc650lng98le1jc3uga537s2; path=/
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	www.owasp.org/index.php/HttpOnly
WASC Id	13

2) Report for Prototype Built Under Normal SDLC.



Low (Warning)	Cookie set without HttpOnly flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	http://localhost/sqlbuddy/
Parameter	PHPSESSID=frmdiq1n5v2mmfrl4602i3is17; path=/
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	www.owasp.org/index.php/HttpOnly
WASC Id	13

Low (Warning)	Password Autocomplete in browser
Description	AUTOCOMPLETE attribute is not disabled in HTML FORM/INPUT element containing password type input. Passwords may be stored in browsers and retrieved.
URL	http://localhost/phpmyadmin/
Parameter	input
Attack	<input type="password" name="pma_password" id="input_password" value="" size="24" class="textfield" />
Solution	Turn off AUTOCOMPLETE attribute in form or individual input elements containing password by using AUTOCOMPLETE='OFF'
Reference	http://msdn.microsoft.com/library/default.asp?url=/workshop/author/forms/autocomplete_ovr.asp
CWE Id	525

Low (Warning)	X-Content-Type-Options header missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to

	'nosniff'
URL	http://localhost/phpmyadmin/
Solution	This check is specific to Internet Explorer 8 and Google Chrome. Ensure each page sets a Content-Type header and the X-CONTENT-TYPE-OPTIONS if the Content-Type header is unknown
URL	http://localhost/sqlbuddy/js/mootools-1.2-core.js?ver=1_3_3
Solution	Most modern Web browsers support the X-Frame-Options HTTP header, ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY).
Reference	http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx?Redirected=true
Low (Warning)	**X-Content-Type-Options header missing**
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'
URL	http://localhost/sqlbuddy/js/movement.js?ver=1_3_3
Solution	This check is specific to Internet Explorer 8 and Google Chrome. Ensure each page sets a Content-Type header and the X-CONTENT-TYPE-OPTIONS if the Content-Type header is unknown
Reference	
Informational (Warning)	**X-Frame-Options header not set**
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks
URL	http://localhost/sqlbuddy/js/movement.js?ver=1_3_3
Solution	Most modern Web browsers support the X-Frame-Options HTTP header, ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY).
Reference	http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx?Redirected=true
Low (Warning)	**X-Content-Type-Options header missing**
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'
URL	http://localhost/sqlbuddy/themes/bittersweet/css/ie.css?ver=1_3_3
Solution	This check is specific to Internet Explorer 8 and Google Chrome. Ensure each page sets a Content-Type header and the X-CONTENT-TYPE-OPTIONS if the Content-Type header is unknown
Reference	
High (Warning)	**Cross Site Scripting (Reflected)**
Description	Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a

	software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology. There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash.
URL	http://localhost/phpmyadmin/index.php?collation_connection=utf8_general_ci&db=javascript%3Aalert%281%29%3B&lang=en&table&token=75296e71ffb200fd7d3269cec581cda5
Parameter	db
Attack	javascript:alert(1);
Solution	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket. Phases: Implementation; Architecture and Design Understand the context in which your data will be used and the encoding that will be expected. This is especially important when transmitting data between different components, or when generating outputs that can contain multiple encodings at the same time, such as web pages or multi-part mail messages. Study all expected communication protocols and data representations to determine the required encoding strategies. For any data that will be output to another web page, especially any data that was received from external inputs, use the appropriate encoding on all non-alphanumeric characters. Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed. Phase: Architecture and Design

	Phase: Implementation Ensure that you perform input validation at well-defined interfaces within the application. This will help protect the application even if a component is reused or moved elsewhere.
Reference	http://projects.webappsec.org/Cross-Site-Scripting http://cwe.mitre.org/data/definitions/79.html
CWE Id	79
WASC Id	8
High (Warning)	**Cross Site Scripting (Reflected)**
Description	Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology. There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing the code.
URL	http://localhost/phpmyadmin/index.php?db=javascript%3Aalert%281%29%3B&lang=en&table&token=a38d5febb8baed29ed9aa8fb0fac06b7
Parameter	db
Attack	javascript:alert(1);
Solution	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket. Phases: Implementation; Architecture and Design Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed. Phase: Architecture and Design Phase: Implementation

	To help mitigate XSS attacks against the user's session cookie, set the session cookie to be HttpOnly. In browsers that support the HttpOnly feature (such as more recent versions of Internet Explorer and Firefox), this attribute can prevent the user's session cookie from being accessible to malicious client-side scripts that use document.cookie. This is not a complete solution, since HttpOnly is not supported by all browsers. More importantly, XMLHTTPRequest and other powerful browser technologies provide read access to HTTP headers, including the Set-Cookie header in which the HttpOnly flag is set. Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a blacklist). However, blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue." Ensure that you perform input validation at well-defined interfaces within the application. This will help protect the application even if a component is reused or moved elsewhere.
Reference	http://projects.webappsec.org/Cross-Site-Scripting http://cwe.mitre.org/data/definitions/79.html
CWE Id	79
WASC Id	8
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/phpmyadmin/doc/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options

	. http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/phpmyadmin/doc/html/_sources/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/phpmyadmin/doc/html/_static/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48

Medium (Warning)	Directory browsing
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/phpmyadmin/js/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48

Medium (Warning)	Directory browsing
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/phpmyadmin/themes/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48

Medium (Warning)	Directory browsing
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.

URL	http://localhost/phpmyadmin/themes/pmahomme/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/phpmyadmin/themes/pmahomme/img/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/phpmyadmin/themes/pmahomme/jquery/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.

Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/sqlbuddy/css/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/sqlbuddy/js/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess:

	. http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/sqlbuddy/themes/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/sqlbuddy/themes/bittersweet/css/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548

WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/sqlbuddy/themes/bittersweet/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000058.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/webgrind/img/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/webgrind/js/

Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48
Medium (Warning)	**Directory browsing**
Description	It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which be accessed to read sensitive information.
URL	http://localhost/webgrind/styles/
Attack	Parent Directory
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	For IIS, turn off directory browsing. For Apache, use the 'Options -Indexes' directive to disable indexes in directory or via .htaccess: . http://httpd.apache.org/docs/mod/core.html#options . http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html . or create a default index.html for each directory.
CWE Id	548
WASC Id	48