Prototype Expert System to Estimate the Risk level of an Application

GRADUATE PROJECT REPORT

Submitted to the Faculty of the Department of Computing Sciences Texas A&M University-Corpus Christi Corpus Christi, Texas

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

By

Suliman Alrumaih
Spring 2017

Committee Members

Dr. Mario Garcia  Committee Chairperson

Dr. David Thomas  Committee Member
ABSTRACT

In order for organizations to determine the extent of a potential risk and threat associated with an IT application, they need to perform a risk assessment on such application. To perform such task, it is necessary for the assessor to obtain detailed information regarding the applications’ business purposes and what kind of data are processed and stored in each application.

This project, is an expert system that is capable of categorizing the organization’s IT applications and segregating them depending on the potential security risk into: critical, high, medium or low risk. The system will give each application an overall risk rating based on easy to understand questions to help the organization on its mission to securing its existing and all upcoming application. The knowledge base for this expert system will be acquired from two sources- personal expertise in the industry applying all the knowledge the developer gained from implementing and maintaining Information Security Management System (ISMS), as well as the work of others by following the best information security and risk standards such as ISO 27001, COBIT, and NIST as a guideline for the implementation and to determine the level of potential risk each application imposes to the organization.
TABLE OF CONTENTS

ABSTRACT.................................................................................................................. ii

TABLE OF CONTENTS............................................................................................... iii

LIST OF FIGURES ...................................................................................................... iv

LIST OF TABLES ......................................................................................................... vii

1. BACKGROUND AND RATIONALE ......................................................................... 1
   1.1. Introduction ..................................................................................................... 1
   1.2. Expert Systems ............................................................................................... 2
   1.3. Expert Systems in Information Security ......................................................... 2

2. NARRATIVE ............................................................................................................. 4

3. SYSTEM DESIGN ................................................................................................... 6
   3.1. Knowledge Base .............................................................................................. 6
   3.2. Inference Engine ............................................................................................ 7
   3.3. Database .......................................................................................................... 7
   3.4. Flow of Execution ......................................................................................... 8
   3.5 User Interface ................................................................................................. 15

4. TESTING AND EVALUATION ............................................................................ 22
   4.1. Risk Rating Test Cases .................................................................................. 22
   4.2 Registration Test Cases .................................................................................. 29
   4.3. Login Test Cases ........................................................................................... 33
   4.4. Add Application Form Test Cases ................................................................. 33
   4.5. OWASP Application Vulnerability Test ......................................................... 35
5. CONCLUSION AND FUTURE WORK .............................................................. 37

BIBLIOGRAPHY and REFERENCES ............................................................ 38
LIST OF FIGURES

Figure 3.1. System Architecture ..................................................................................6
Figure 3.2. System Class Diagram ..............................................................................8
Figure 3.3. System Login Page ....................................................................................16
Figure 3.4. System Registration Page .........................................................................16
Figure 3.5. System Dashboard Page ...........................................................................17
Figure 3.6. Add New Application Page .......................................................................18
Figure 3.7. Application Details Page ..........................................................................19
Figure 3.8. All Application Page ................................................................................19
Figure 3.9. Departments Page ....................................................................................20
Figure 3.10. Actions Page ..........................................................................................21
Figure 4.1. Risk Rating Low ......................................................................................26
Figure 4.2. Risk Rating Medium ...............................................................................27
Figure 4.3. Risk Rating High .....................................................................................28
Figure 4.4. Risk Rating Critical ...............................................................................29
Figure 4.5. Empty Filed Error ....................................................................................30
Figure 4.6. Valid Email Error ....................................................................................31
Figure 4.7 (a). Password Less Than 8 Error ..............................................................32
Figure 4.7 (b). Password Complexity Error ..............................................................32
Figure 4.8. Login Error .............................................................................................33
Figure 4.9. Form Empty Error ..................................................................................34
Figure 4.10. Form Email Error ................................................................................34
Figure 4.11. Form Selection Error ..................................................................................35

Figure 4.12. Reported Vulnerabilities..........................................................................36
LIST OF TABLES

Table 1. List of Test Cases

...........................................................................................................11
1. BACKGROUND AND RATIONALE

1.1 Introduction

There are many reasons which motivate organizations to take a proactive approach when addressing information security concerns. From preserving customers’ trust and expectation to legal and regulatory requirements, organizations regardless of size or mission need to mitigate all risk effectively and in a timely manner to avoid experiencing a high-profile application security breaches that result in the compromise of personally identifiable information (PII).

The first step in securing any origination is to identify and quantify the risks of the organization’s information assets. This process can be done using what is known as risk profiling. Originations currently are using a manual process (typically using an excel spreadsheet) to collect the information needed to assess the risk of an application. This creates a manual and inconsistent data collection which leads to an ambiguous and unreliable interpretation of risk. Another issue with the current process is the lack of central risk repository or historical data as well as lack of business context and no holistic view of risk which leads to uncorrelated and redundant data included in risk analysis and inability to measure improvements and predict threats. This manual process cost organizations valuable time and resources in addressing noncritical risks. Therefore, this project’s goal is to solve these problems and help organizations in their risk assessment.
1.2 Expert Systems

Expert systems (ES) is a form of Artificial Intelligence that uses the human specialized expertise and knowledge to solve complex problems or give advice. It could take any form such as procedures, data related to domains or rules. The use of ES becomes very critical and useful where the amount of data to be processed is massive. An ES is usually composed of a knowledge base, inference engine, and a user interface [12].

1.3 Expert Systems in Information Security

1.3.1 AudES - an Expert System for Security Auditing

Gene Tsudik and Rita Summers from the IBM Los Angeles Scientific Center present their expert system named AudES. AudES is an Expert System for Security Auditing that takes on the challenge of automating the manual security auditing procedures and to alleviate the burden on human auditors. AudES is interposed between a human auditor and Resource Access Control Facility (RACF), a popular security mechanism for IBM mainframe systems. RACF reports are reviewed individually for each system. With the use of an expert system (AudES), a correlation between systems’ activity can be performed easily removing the potential of hackers distributing their attempts among several systems.

AudES was developed using Expert System Environment (ESE) tool and focuses on automating the process of detection of violations. After learning the current manual procedures for auditing systems in the origination, the system then will acquire the logs from RACF get all systems and rate the violations by looking at different variables such as VIOLATION TYPE, USER TYPE, LOCATION, DATE/TIME, and THRESHOLD. AudES then will apply the following rules:

**IF** (resource-violation-counter > resource-threshold)

**AND** (user-type IS CUSTOMER)

**AND** (resource-type IS INTERNAL)

**THEN** recommended-action =”Contact the resource owner and report incident to Customer Assistance”
After that, AudES will display the suspected violations along with all the necessary user data as well as the information on commands, resources, affiliations and, most importantly, appropriate actions that an auditor must take according to the local security auditing guidelines [13].

### 1.3.2 An Expert System for Risk Assessment of Information System Security Based on ISO 27002

Sihwi, S.W., Andriyanto, F., and Anggrainingsih, R. (2016) in their work on using Expert System for Risk Assessment of Information System Security managed to successfully integrate between the risk assessment and expert system which lead to a creation of an expert system that is able to predict the security position in different companies, determine whether or not those companies need to conduct a security audit and also provide key information for the decision makers in determining where the critical risks are and prioritizing them and also provide optimized solution for tackling these risks. They based their risk assessment standard on ISO 27002 and used the forward chaining method to determine the rules and scoring in the Expert System.
2. NARRATIVE

One of the most used applications is “Practical Threat Analysis” or “PTA”. PTA is a threat risk assessment tool that assists security consultants and analysts in assessing the operational and security risks in their systems as well as helping them in creating an appropriate risk mitigation policy [9]. However, because PTA’s process is manual -using excel spreadsheet- which leads to uncorrelated and redundant data included in risk analysis and inability to measure improvements and predict threats, this project was created to solve these problems.

This project is a web based application that provides organizations with an automated and consistent data collection risk repository for web applications in their environment. This application will ask the user to enter the application’s information such as application’s name, application’s owner’s contact information, application’s URL, and a brief description of the application’s function. After that, it will ask the user 10 easy to understand questions about the application. For example, it will ask: “What is the most sensitive classification of information within the application?” And it will present the user with a number of choices (Public, Private, Confidential, and Restricted). Another question would be: “What is the availability requirement for the supported system? (Recovery Time Objective)”, and the user will have to choose the time that’s applicable to the application. This question is to measure the potential impact of system’s unavailability resulting in loss of revenue and operational inefficiencies, which means compromising availability. Another question that is important to determine the level of potential risk of an IT application is to know the number of interfaces to internal Company’s systems. In this case, Interfaces represent the points of interconnection between two systems or subsystems. The more interfaces there are, the higher the risk.
Another question is: “How can the system be accessed?” and the user will have four options to choose from (Internal Network, Extranet, Internet with Controls, or Internet Public). Systems that can be accessed through the company’s internal network only, had less threat than being hosted on a publicly accessible server.

In a nutshell, the system will calculate the application’s overall risk rating by using the user’s input from answering all required questions. Each answer will be scored from 0 – 3 and then will be multiplied by the weight factor allocated to that question. The total risk rating for the application then will be calculated by adding up each question’s risk valuation.

The system then will take the final result and determine the risk category of that application to be either: critical, high, medium or low. After that, the system will recommend actions need to be taken for each application based on their risk rating and will provide a cost estimate for each action. Finally, the system will present a dashboard of charts and tables that show the overall risk level in the organization.
3. SYSTEM DESIGN

Figure 3.1 represent the architecture the system.

![System Architecture Diagram]

**Figure 3.1 System Architecture**

3.1 Knowledge Base

The knowledge in the knowledge base is represented in a set of if-then rules [12].

The knowledge for this system was acquired from following resources:

1) COBIT (Control Objectives for Information and Related Technologies) [1].


3.2 **Inference Engine**

The inference engine uses the knowledge from the knowledge base to derive conclusions. Using Forward chaining method, the inference engine searches the rules provided by the knowledge base until it finds one where the antecedent is true [12]. For example, the inference engine will go through the following rules to determine the risk score for data classification:

- If Data classification is "Restricted", then the risk score for Data classification = 75
- If Data classification is “Confidential”, then the risk score for Data classification = 50
- If Data classification is “Private”, then the risk score for Data classification = 25
- If Data classification is “Public", then the risk score for Data classification = 0

3.3 **Database**

All the relational table schemas which are called models are defined in the database layer [3]. User can retrieve from the database a single object representing one row in a particular table or the user can also retrieve a series of objects satisfying certain conditions that are previously specified. These queries are done by using a query set that initially contains the information about the query. It will hit the database and will be evaluated only when actually the results would be needed from it [3]. Query set offers a quick approach to make a query that even involves complex relationships between tables.
(e.g. foreign key or many-to-many relationships). Figure 3.2 show the system’s class diagram.

**Figure 3.2 System Class Diagram**

### 3.3 Flow of Execution

**Step 1: Add a new Application**

To add a new application to the system, the user is asked to enter the following details about the application.

1) **Application Name**: application’s name is used to identify the application. This has to be a unique entry.
2) Application Owner: name of the person responsible for this application in the organization

3) Application Owner Email: this has to be an email address. The system will reject any entry that does not match an email format

4) Application Department: name of the department that uses the application. The user has to select a department from a dropdown menu

5) Application URL/hostname: the web address for the application

6) Asset Description: a brief description of the application’s function

7) Data classification: the user has to select an option that corresponds to what is the most sensitive classification of information within the application? The choices are (Public, Private, Confidential or Restricted).

8) Reputational impact: the user has to indicate if there will be a potential reputational impact to the company if the application is compromised. The choices are (Yes or No)

9) Number of users: How many users access the system? The choices are (0-50, 51-250, 251-1000, or >1000).

10) Recovery Time Objective: What is the availability requirement for the supported system? The choices are (< 24 Hours, 24 - 48 Hours, 48 - 72 Hours, 4-7 Days, or 8+ Days)

11) System Access: How can the system be accessed? The choices are (Internal Network, Extranet, Internet with Controls, or Internet Public)

12) Fraud impact: Does the application have a fraud impact (financial or Sox)? The choices are (Yes or No).
13) Hosting location: is the application hosted outside of the company network?
   The choices are (Yes or No).

14) Number of interfaces: How many systems does this application interact with?
   The choices are (0, 1-2, 3-4 or > 4).

15) Regulatory impact: Is there a regulatory impact related to this application (i.e. PCI, ITAR, etc.)? The choices are (Yes or No).

16) Application content: Type of content within the application. The choices are (Static, Dynamic, or E-commerce).

After filling the form completely and clicking on the submit button, the system will then store the data into the database and call the risk score calculation activity.

**Step 2: Calculating Risk score:**

In this step, the inference engine will retrieve the application’s data entered by the user in step one from the database. Then, it will go through the questions one by one and match the answer to a rule from the knowledge base. Starting with the first question, the system will follow these steps:

1) Data classification score (allocated weight for this question=25):
   a. if Data classification is "Restricted", then the risk score for Data classification = 75
   b. if Data classification is "Confidential", then the risk score for Data classification = 50
   c. if Data classification is "Private", then the risk score for Data classification = 25
d. if Data classification is " Public ", then the risk score for Data classification = 0

2) Reputational impact score (allocated weight for this question=10):
   a. if Reputational impact is "YES", then the risk score for Reputational impact = 30
   b. if Reputational impact is "NO", then the risk score for Reputational impact = 0

3) Number of users score (allocated weight for this question=10):
   a. if Number of users is "0-50", then the risk score for Number of users = 0
   b. if Number of users is "51-250", then the risk score for Number of users = 10
   c. if Number of users is "251-1000", then the risk score for Number of users = 20
   d. if Number of users is ">1000", then the risk score for Number of users = 30

4) Recovery Time Objective score (allocated weight for this question=3):
   a. if Recovery Time Objective is "8+ Days ", then the risk score for Recovery Time Objective = 0
   b. if Recovery Time Objective is "4-7 Days ", then the risk score for Recovery Time Objective = 0
   c. if Recovery Time Objective is "48 - 72 Hours ", then the risk score for Recovery Time Objective = 3
d. if Recovery Time Objective is "24 - 48 Hours ", then the risk score for Recovery Time Objective = 6

e. if Recovery Time Objective is "< 24 Hours ", then the risk score for Recovery Time Objective = 9

5) System Access score (allocated weight for this question=10):

a. if System Access is " Internal Network ", then the risk score for System Access = 0

b. if System Access is " Extranet ", then the risk score for System Access = 10

c. if System Access is " Internet with Controls ", then the risk score for System Access = 20

d. if System Access is " Internet Public ", then the risk score for System Access = 30

6) Fraud impact score (allocated weight for this question=10):

a. if Fraud impact is "YES", then the risk score for Fraud impact = 30

b. if Fraud impact is "NO", then the risk score for Fraud impact = 0

7) Hosting location score (allocated weight for this question=8):

a. if Hosting location is "YES", then the risk score for Hosting location = 24

b. if Hosting location is "NO", then the risk score for Hosting location = 0

8) Number of interfaces score (allocated weight for this question=8):

a. if Number of interfaces is "0", then the risk score for Number of interfaces = 0
b. if Number of interfaces is "1-2", then the risk score for Number of interfaces = 8

c. if Number of interfaces is "3-4", then the risk score for Number of interfaces = 16

d. if Number of interfaces is "> 4", then the risk score for Number of interfaces = 24

9) Regulatory impact score (allocated weight for this question=8):

a. if Regulatory impact is "YES", then the risk score for Regulatory impact = 24

b. if Regulatory impact is "NO", then the risk score for Regulatory impact = 0

10) Application content score (allocated weight for this question=8):

a. if Application content is " Static ", then the risk score for Application content = 0

b. if Application content is " Dynamic ", then the risk score for Application content = 16

c. if Application content is " E-commerce ", then the risk score for Application content = 24

The system will then calculate the total risk rating for the application by adding up all the risk values for each question and store the total in the database.

**Step 3: Risk Rating**

In this step, the inference engine will retrieve the application’s risk rating calculated in step two from the database. Then, it will try to match the result to one of the following rules:
1) if risk rating is <= 50 then risk rating is “Low”

2) if risk rating is >50 AND <= 100 then risk rating is “Medium”

3) if risk rating is >100 AND <= 150 then risk rating is “High”

4) if risk rating is >150 then risk rating is “Critical”

The system will store the risk rating value in the database.

**Step 4: Recommended Actions and Estimate cost**

In this step, the inference engine will retrieve the application’s risk rating from the database. Then, it will try to match it to one of the following rules:

1) if risk rating is “Low” the recommend the following actions:
   - i. “Self-Risk Assessment”

2) if risk rating is “Medium” the recommend the following actions:
   - i. “Black Box vulnerability Scan”
   - ii. “Self-Risk Assessment”

3) if risk rating is “High” the recommend the following actions:
   - i. “Information Security Risk Assessment”
   - ii. “White Box vulnerability Scan”
   - iii. “Database Scan”

4) if risk rating is “Critical” the recommend the following actions:
   - i. “Information Security Risk Assessment”
   - ii. “White Box vulnerability Scan”
   - iii. “Database Scan”
   - iv. “Application Penetration Testing”
   - v. “Security Architecture Review”

The system will store the recommended actions in the database. Each action has an
associated cost provided by the user. The system will add these cost based on what action has been recommended and store them into the database.

**Step 5: Display Result**

In this step, the system will display the following results to the user:

1) Application Information :
   a. Application name
   b. Application Owner
   c. Application Owner Email
   d. Application Department
   e. Application URL/hostname
   f. Asset Description

2) Application Data Classification

3) Application Risk Rating

4) Recommend Actions

5) Estimated Cost

3.5 **User Interface**

3.5.1 **Login page**

Figure 3.4 shows the first screen shown when the user navigate to the applications’ URL. In the login page, the user needs to enter his/her user ID and password to login to the app or they can go to the registration screen if they click on register.
3.5.2 Register page

The screens in Figure 3.5 shows the registration form used by the new user to create an account. The user has to enter a username, Email and a password. As soon as the registration is done, the student is logged in to the app.
3.5.3 Dashboard

The dashboard page is the home page for this application. In this page, the user is presented with 3 dynamic charts. The first chart is a pie chart that shows risk rating of all applications in the system. The second chart shows the total anticipated cost for each department in the organization. And the third chart shows the number of actions required for all applications. On top of these chart, we have 3 boxes that shows the total number of application in the system, the estimated total cost for all applications and the total number of actions required for all applications. On the left of the page, we have the navigation options which will take the user to another page. The first option is the dashboard page, the second all apps page, the third is the department page and the fourth is actions page. On the top right corner, we have add new application tap and the logout tap. Figure 3.6 shows the dashboard page.

![Figure 3.5 System Dashboard Page](image-url)
3.5.4 Add a new application

In this page, the user is presented with a form that they need to fill to add a new application. Once the user submit the form, they will be taken to the application detail page. Figure 3.7 shows add new application page.

![Add New Application Page](image)

Figure 3.6 Add New Application Page

3.5.5 Application detail

In this page the user will be presented with the application details along with the calculated risk rating, estimated cost (that’s calculated based on each recommended action’s cost) and the required action for the application. Figure 3.8 shows the application detail page.
3.5.6 All Apps page

In this page, the user is presented with a table that includes all the applications in the system. The table includes the application ID, name, department, rating, and cost. Figure 3.9 shows the All Apps page.
3.5.7 Departments page

In this page, the user is presented with a pie chart that shows the risk rating along with the total cost for each individual department. Figure 3.10 shows the department page.

![Figure 3.9 Departments Page](image)

3.5.8 Actions page

In this page, the user is presented with a table that includes all the Actions required for all applications in the system. The table include the action, application name, department, and rating. Figure 3.11 shows the Actions page.
**Figure 3.10 Actions Page**

<table>
<thead>
<tr>
<th>Action</th>
<th>Application Name</th>
<th>Department</th>
<th>Rating</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self Risk Assessment</td>
<td>Desk</td>
<td>Information Technology (IT)</td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>Self Risk Assessment</td>
<td>NetSuite</td>
<td>Operations</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Self Risk Assessment</td>
<td>Zoho Books</td>
<td>Finance</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Self Risk Assessment</td>
<td>SysAid</td>
<td>Information Technology (IT)</td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>Self Risk Assessment</td>
<td>Quick Legal</td>
<td>Legal</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Self Risk Assessment</td>
<td>iCIMS Talent Platform</td>
<td>Human Resources (HR)</td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>Self Risk Assessment</td>
<td>MasterControl</td>
<td>Operations</td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Self Risk Assessment</td>
<td>AnswerHub</td>
<td>Operations</td>
<td>Low</td>
<td></td>
</tr>
</tbody>
</table>
4. TESTING AND EVALUATION

Testing and evaluation includes testing all the functionalities of the application. The application is tested using Ubuntu 16.04 Server with 512 MB memory. Each module of the application is tested with all the possible test cases.

4.1 Risk Rating Test Cases

The purpose of the risk rating test case is to validate the outcome of every possible answer the user select from the application questioner. This test was done manually on each question to make sure the application will be classified correctly either critical, high, medium or low, based on the calculated score. All 32 options were tested individually to see if the calculated score is correct. This was done by adding a printout statement after each option is selected by the user. For example, if the user selected “Privet” as the confidentiality option, the system should assign 25 as risk score for confidentiality. This way we eliminate the need for checking all possible combinations that the user can choose from.

The results of some of the test cases are shown below in Table 1.

<table>
<thead>
<tr>
<th>Test Case No.</th>
<th>Category</th>
<th>User Input</th>
<th>Expected Output</th>
<th>Result Obtained from the Expert System</th>
<th>Test Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data</td>
<td>Public</td>
<td>Low</td>
<td>Low</td>
<td>Successful</td>
</tr>
<tr>
<td>Classification - Reputational Impact</td>
<td>- NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Number of Users</td>
<td>- 251-1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Availability Requirement</td>
<td>- 24 - 48 Hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- System be Access - Internal Network</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fraud Impact</td>
<td>- No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Hosting - No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Location - No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Number of Interfaces - 1-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Regulatory Impact - No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Application’s Content - Static</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| 2 | - Data Classification - Private |
|   | Medium | Medium | Successful |
|   | - Reputational Impact - NO |
|   | - Number of Users - 0-50 |
|   | - Availability Requirement - 48 - 72 Hours |
|   | - Internet with |</p>
<table>
<thead>
<tr>
<th></th>
<th>Controls</th>
<th>High</th>
<th>High</th>
<th>Successful</th>
</tr>
</thead>
<tbody>
<tr>
<td>- System be</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Access</td>
<td>- Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fraud Impact</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Hosting</td>
<td>- 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Location</td>
<td>- No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Number of Interfaces</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Regulatory Impact</td>
<td>- Static</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Application’s Content</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Data</td>
<td>- Confidential</td>
<td>High</td>
<td>High</td>
<td>Successful</td>
</tr>
<tr>
<td>- Classification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Reputational Impact</td>
<td>- YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Number of Users</td>
<td>- 0-50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Availability</td>
<td>- &lt; 24 Hours</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Requirement</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- System be</td>
<td>- Internet with Controls</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Access</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fraud Impact</td>
<td>- No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Hosting</td>
<td>- No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>Number of Interfaces</td>
<td>Regulatory Impact</td>
<td>Application’s Content</td>
<td>Data Classification</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------</td>
<td>-------------------</td>
<td>-----------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Location</td>
<td>- 0</td>
<td>- No</td>
<td>- Dynamic</td>
<td>- Confidential</td>
</tr>
<tr>
<td>- Application’s Content</td>
<td>- Dynamic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1. List of Test Cases**

Figure 4.8 shows the result screen from test case #1.

![Application's Profile](image)

**Figure 4.1 Risk Rating Low**
Figure 4.9 shows the result screen from test case #2.

**Figure 4.2 Risk Rating Medium**

Figure 4.10 shows the result screen from test case #3.
Figure 4.3 Risk Rating High

Figure 4.11 shows the result screen from test case #4.
In this case, the user will be asked to choose a username and a password as well as entering a valid email address. The user will be prompted to choose a password that is
at least 8 characters long, with at least one upper case letter, one lower case letter and one special character. The system will validate if the password selected meets the minimum required length (8 characters). If it does meet the requirement the system should prompt the user that the password selected does not meet the minimum requirement for a strong password and they need to select another one that is at least 8 character long.

4.2.1 Empty Filed

The first case was to test empty fields in the registration form. If a user clicks on the Register button without entering any data in the registration fields, the application displays an error message as shown in Fig 4.1(a) indicating that it is a mandatory field.

![Figure 4.5 Empty Filed Error](image)

4.2.2 Valid Email

The second case is to check if the user have entered an email address in the Email field. The system will check if the entered text is in an Email format. If not, then an error message will displayed asking the user to enter a valid email address as shown in Fig 4.2.
31

Figure 4.6 Valid Email Error

4.2.3 Complex Password

The third case is to test if the password selected by the user meet the minimum complexity requirements. If not, the system will display an error message based on the missing requirements as shown on Fig 4.3(a) 4.3(b).
Figure 4.7(a) Password Less Than 8 Error

- Must be more complex (must contain 1 or more unique uppercase characters, 1 or more unique lowercase characters, 1 or more non unique special characters)

Figure 4.7 (b) Password Complexity Error
4.3 Login Test Cases

When the user enters invalid credentials, or leave one or more of the fields empty, the application will show message invalid login. Figure 4.4 shows the test case with invalid user login credentials.

![Login Error](image)

**Figure 4.8 Login Error**

4.4 Add Application Form Test Cases

The first case was to test empty fields in the Add Application form. If a user clicks on the Submit button without entering any data in the application form fields, the application displays an error message as shown in Fig 4.5 indicating that it is a mandatory field.
The second case is to check if the user have entered an email address in the Email field. The system will check if the entered text is in an Email format. If not, then an error message will displayed asking the user to enter a valid email address as shown in Fig 4.6.

![Application's Profile](image)

**Figure 4.9 Form Empty Error**

The third case is to test the selection filed. If the user does not select an option the application displays an error message as shown in Fig 4.7 indicating that they need to select an option.

![Application's Profile](image)

**Figure 4.10 Form Email Error**
4.5 OWASP Application Vulnerability Test

In this test, The OWASP Zed Attack Proxy (ZAP) tool was utilized to scan the code and verify it against OWASP Top 10 Most Critical Web Application Security Risks. This tool automatically find security vulnerabilities in web applications while they are in the development and testing stages. The tool reported that this system has a total of 6 vulnerabilities, one of which is considered medium risk and the other five are low risk vulnerabilities. Due to time constrain, these issue will be fixed in future work. Fig 4.12 shows the reported vulnerabilities.
ZAP Scanning Report

Summary of Alerts

<table>
<thead>
<tr>
<th>Risk Level</th>
<th>Number of Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>0</td>
</tr>
<tr>
<td>Medium</td>
<td>1</td>
</tr>
<tr>
<td>Low</td>
<td>5</td>
</tr>
<tr>
<td>Information</td>
<td>0</td>
</tr>
</tbody>
</table>

Alert Detail

<table>
<thead>
<tr>
<th>Medium (Medium)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-Frame-Options Header Not Set</td>
<td>X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.</td>
</tr>
<tr>
<td>Cookie No HttpOnly Flag</td>
<td>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.</td>
</tr>
<tr>
<td>Cross-Domain Javascript Source File Inclusion</td>
<td>The page includes one or more script files from a third-party domain.</td>
</tr>
<tr>
<td>Web Browser XSS Protection Not Enabled</td>
<td>Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server.</td>
</tr>
<tr>
<td>Web Browser XSS Protection Not Enabled</td>
<td>Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server.</td>
</tr>
<tr>
<td>Password Autocomplete in Browser</td>
<td>The AUTOCOMPLETE attribute is not disabled on an HTML FORM/INPUT element containing password type input. Passwords may be stored in browsers and reused.</td>
</tr>
<tr>
<td>X-Content-Type-Options Header Missing</td>
<td>The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'text/html'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type if one is set, rather than performing MIME-sniffing.</td>
</tr>
</tbody>
</table>

Figure 4.12 Reported Vulnerabilities
5. CONCLUSION AND FUTURE WORK

This project is a web application that provide organizations with an automated and consistent data collection risk repository for web applications in their environment. The system will present to the user a dashboard that shows the overall risk level in the organization. The user will be able to drill down to see the risk score for each application and the recommended action needed for each application based on the risk score that been given to by the system.

Future Work:

This system can be improved by adding following functionalities:

- Fix the security vulnerabilities reported by The OWASP Zed Attack Proxy (ZAP) tool.
- Add an automated task assignment where the application can assign a task to the team responsible for performing the action recommended by the application.
- Integrate the system with a Security vulnerabilities scanning tool where the user can perform a security vulnerability scan from the system interface and attach the report to the application detail page.
- User a more complex rules that provide more accurate risk rating. This can be done by implementing a vulnerability scoring rules that uses the Common Vulnerability Scoring System (CVSS) and integrate them with the current rule set.
BIBLIOGRAPHY and REFERENCES


