Linnæus University
Sweden

Independent Project's Thesis

Web developers awareness of web-security risks

From the perspective of OWASP's Top 10 Most Critical Web Application Security Risks

Author: Mosa Kasem Rasol
Supervisor: Supervisor
Examiner: Johan Leitet
Semester: Spring 2019
Subject: Computer Science
Level: First Level
Course code: 1DV50E, 7.5 credits

Linnæus University
Sweden

Linnæus University
Sweden

Abstract

Cybercrime is relentless, undiminished and unlikely to stop. An attack on web application can take more time to resolve and can result in bankruptcy for organizations and make hundreds of millions of people victims. The Open Web Application Security Project (OWASP) focused on improving the security of software and offer recommendations to prevent the top-10 most critical web application security risks. Are web-developers aware of the web-security risks, it is hard to know for certain. The aim of this thesis was therefore to clarify and get an in-depth understanding of the situation.

Keywords: Web-developers, OWASP, Web-developers awareness of web-security risks, web-security

Sammanfattning

Cyberbrottslighet är obeveklig, obefintlig och osannolikt att sluta. En attack på en webbapplikation kan ta tid att lösa och kan leda till konkurs för organisationer och göra hundratals miljoner människor till offer. Open Web Application Security Project (OWASP) fokuserar på att förbättra programvarans säkerhet och erbjuda rekommendationer för de 10 mest kritiska säkerhet risker i webbapplikationer. Är webbutvecklare medvetna om webbsäkerhetsriskerna, svårt att veta säkert. Syftet med denna studie var att klargöra och få en djupgående förståelse av situationen.

Acknowledgements

There is no affliction worse than a lack of intellect - Imam Muhammad al-Baqir

First and foremost, I would like to thank, the biggest source of my strength, my family, relatives and specially my mother.

My gratitude and thanks to my teachers at the Linnaeus university and special thanks to **Johan Leitet**, for sparing his valuable time whenever I approached him, asking for assistance, he showed me the way ahead. Thank you sir.

Mosa Kasem Rasol.

Contents

1 Introduction

Web technologies are growing epidemically in the last decade [1], leading to websites becoming the prime target of attacks for criminals [2], companies often forget or leave behind the most important features that shelters the service from attacks. Millions of websites regulate access to highly sensitive data, from social security numbers to credit card numbers, names, addresses, and phone numbers, to preserve the integrity of that data, a number of 3,418,647,753 total records containing personal and other sensitive data that have been compromised between 2017 to 2019 [2]. Security professionals rightfully point out that a single flaw in practically any program can result in a devastating security compromise [3]. The importance of creating a secure website and website security is there for increasing rapidly.

"In addition to a significant increase in cyber crime, we also see new and more sophisticated threats with the risk of further escalation" Richard Oehme[4]

In this thesis, web-developers will be interviewed to get an insight whether or not they are aware of the security risks that exist in the web applications from the security perspective of the Open Web Application Security Project, also known as OWASP [5]. The objective is to examine to what extent the web-developers in Kalmar are aware of the top 10 most critical web application security risks.

1.1 Background

The main goal of security is to monitor and to protect the data and act as a deterrent to cyberattacks [6], the result is winning the confidence or trust of a user to continue being a customer to the organization that his/her data is safe. Different organizations have different valuable data which always is a valuable target for an attack. In this modern and fast-paced world, security is more important than ever. The result in a report dated back to 2016 by Web Security Scanner, revealed that 84% of web applications have one or more medium-severity vulnerability and 55% have at least one high-severity vulnerability in web applications [7]. According to another scanner, the Edgescan report dated back to 2018, revealed that 20% of all vulnerabilities discovered are High or Critical Risk [8].

Lack of awareness of the risks and their consequences can lead to a catastrophic outcome, as losing a colossal number of users very fast, facing class-action lawsuits, regulatory fines, reputation damage, or even risk facing bankruptcy [9]. Some companies are overly confident and do not invest in proper security, therefore risking the valuable information carelessly. Unlikely to be targeted, some companies are unaware that most hackers use automated tools to find vulnerabilities in the application or they are generally aware of the security issue but do not connect those threats with their own business, so precaution of attacks is put aside.

The goal of the study is to gain insight into what extent the web-developers have of web application security risks.

1.1.1 Web Security

Web security is any effort or application taken to ensure the website data is not exposed to threats, or to prevent the exploitation of any website in any way [6].

1.1.2 OWASP

OWASP is an open community providing reports, documentation, and tools to help improve web application security and to raise awareness of the attacks [5]. Every third year, a list of top 10 risks in the web applications is published, unfortunately, as when writing this thesis, the 2019 top 10 report is yet to be released, the thesis will proceed from the latest published document that is available from 2017.

1.1.3 OWASP Top 10 Vulnerabilities

Below is a brief introduction to the top 10 most critical web security risks vulnerabilities in the 2017 report from OWASP as well as recommendations to prevent them [5]:

1. **Injection** - The Injection attack occurs when untrusted data is sent to a code interpreter through a form or some other data submission to a web application [10]. The attacker in this case could enter a SQL database code into the a form that expects plain text, tricking the interpreter to access unauthorized data.

   Injection attack can be prevented by validating [1] or sanitizing [2] user-submitted data.

2. **Broken Authentication** - A Broken Authentication system can give attackers access to user accounts or even an entire web application system should the attacker get access to an admin account [5].

   The system is vulnerable to such an attack when it permits brute-force [3] or other automated attacks, missing or not having an effectual multi-factor authentication, exposing sessions IDs in the URL or permitting weak passwords like "admin" [5].

   Some strategies to mitigate this type of vulnerability in authentication, are to require 2-factor authentication to prevent automated, credential stuffing brute-force and stolen credential re-use attacks [5] as well as to delay repeated login attempts and to log failed attempts and to alert administrators when credential stuffing, brute-force or other attacks are detected [5], and to manage sessions on server-side using a secure built-in session manager.

3. **Sensitive Data Exposure** - Web applications without proper protection puts sensitive data such as financial information and passwords at risk, attackers can then gain access to that data. One popular method for stealing sensitive information is using a man-in-the-middle attack [4].

   Data exposure risk or man-in-the-middle attack does not not have all-in-one solution, because such an attack have numerous ways of being carried out. One of the fundamental way to protect against man-in-middle attack is to adopt SSL/TLS protocols which will establish a secure connection between the web services and the users. Furthermore, by encrypting all data in transit as well as disabling caching for responses that contain sensitive data [5].

[1] Validation means, to reject suspicious data.

[2] Sanitization refers to cleaning up the suspicious part of the data.

[3] Brute-force attack is an activity which involves repetitive successive attempts of trying various password combinations to break into any website [11].

[4] The "man in the middle" intercepts traffic from the source and forwards it to the destination, the attacker places himself between two devices (web browser and a web server) and hence gaining the ability to collection information as well as impersonate either of the two agents [12].

4. **XML External Entities (XXE)** - XML stands for Extensible Markup Language, and is intended to be both human-readable and machine-readable, and this attack is against XML-based web service that parses XML input. The input for the XML parser can be exploited to send data to an unauthorized external entity, which can pass sensitive data directly to an attacker [5]. It is also possible that the XML input can be used to perform a denial-of-service attack on the service [5].

   To prevent XXE attacks is to patch the XML parser and disable the use of external entities in an XML application, an even better alternative would be to have the web application accept less complex type of data such as JSON [5] [5].

5. **Broken Access Control** - The access control is referring to the system that controls access to information or functionality, such as users can not be outside of their intended permissions. Broken access control leaves a gap for attackers to bypass authorization and execute tasks as thought they were an administrator, e.g. bypassing access control by modifying the URL to access the admin panel [5].

   To secure access control is to ensure the web application uses authorization tokens with tight controls [6]. Issue an authorization token when users log in and is invalidated on the server after user logs out [5], this is a secure way for users to authenticate who they are and to avoid having users constantly entering their login credentials.

6. **Security Misconfiguration** - The security misconfiguration can be the result of using default configuration, unnecessary features installed, displaying overly informative error messages which may reveal vulnerabilities in the application [5].

   This can be mitigated by removing unused features and displaying error messages in a more general context [5].

7. **Cross-Site Scripting (XSS)** - XSS vulnerabilities occur when web application takes untrusted data and sends it to a web browser without proper validation, allowing the attacker to execute scripts in the victim's browser which can hijack user sessions [5].

   To protect against XSS vulnerabilities, include escaping untrusted HTTP request [5], and also to validate and sanitize user-generated content. Using framework that automatically escape XSS by design like Ruby on Rails and React [5].

8. **Insecure Deserialization** - This exploit is often the result of deserialization data from untrusted sources and it leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attack, injection attacks and privilege escalation attacks [5].

   Steps can be taken to try and catch the attackers, such as monitoring deserialization and implementing type checks, e.g. in coming type is not the expected type [5]. The better and more direct way to protect against this exploit is to prohibit the deserialization of data from untrusted source.

9. **Using Components with Known Vulnerabilities** - Many modern web application use components such as libraries, frameworks and other software modules that help developers avoid redundant work. As an example, the front-end framework React, where attackers looked for vulnerabilities in the component which they can use to orchestrate attacks [5]. Some of the more popular components are used on hundreds of thousands of websites, a single security flaw in one of these components could have hundreds of thousands of sites vulnerable to exploit.

   To minimize the risk of running components with known vulnerabilities, developers should remove unused components from their projects, as well as ensuring that they are receiving components from a trusted source and ensuring they are up to date [5].

[5] JSON is a language which can be easily understood by both developers and machines,it has become the most popular format to send API requests and responses over the HTTP protocol [13].

[6] Tight controls is referring OWASP [5] suggestions to preventing broken access control.

10. **Insufficient Logging & Monitoring** - Many web applications do not take enough steps to detect data breaches. The average discovery time for a breach is around 200 days [5]. That is enough time to cause significant amount of damage.

    The OWASP recommendation is to implement logging and monitoring as well as incident response plans such as login failures, access control failures and server-side input validation failures to ensure they are made aware of attacks on their application [5].

1.2 Related Work

There have not been any studies regarding this type of research, or about the awareness of web-developers on the web application security risks.

1.3 Problem formulation

A secure system is one that is protected against undesired outcomes. Vulnerabilities or bugs in web apps may enable cybercriminals to exploit the system, to financial, data and identity theft. Are web-developers able to provide protection to the system and to the users and their data from cybercriminals attacks. The top 10 most critical security risks that exists on web applications, are web-developers aware of them?

In response to this problem, our study proposes to investigate several options in order to find what is the perspective of web-developers on web security, how much do web-developers know of the web application security risks, furthermore do they know how these risks occurs? Lastly what actions or steps are taken to prevent or minimize the risks in web applications.

Possible issues that could take place during the research:

1. Participants might not want to answer the questions.
2. Vague response, participants might not be so clear or specific in the expressing their answers during the interview.
3. The possibility of the participants falling sick, and not being able to attend the interview.

1.4 Justification and delimitation

"Cyber threats are already one of the greatest social problems of our time. It is high time that politicians and business now do common things to find solutions for a sustainable digital society." Jakob Bundgaard, Cyber Security Leader, PwC [14].

The focus will be the perspective and awareness of web-developers on web security and web application security risks.

The scope of the study is specific within Kalmar, a city south of Sweden. The objective is to interview at least **3-6 web-developers** from **3 or more different IT-companies**.

2 Method

Quantitative research might not get the full truth from the participants, for it is possible that the participant would search in the internet for the correct answer, therefore it was rejected. To answer the research question, a qualitative study design was selected, a semi-structured interview with web-developers, an important role in the development of web applications or applications that run over the HTTP-layer. It is of the essence that the questions are directed to this work group. The qualitative method was used as they can reveal new information, it can also be more useful to provide an insight into whether or not there is an awareness of the top-10 most critical web application security risks from OWASP's [5] perspective.

2.1 Reliability and Validity

The author performed the interviews with the web-developers. This type of initiative should include a qualitative design to enable in-depth exploration of participants awareness of the security risks in web applications. The aim of this study was therefore a qualitatively explore and to understand to what extent, are the risks understood with three IT-companies.

Face-to-face, structured interviews were considered the most suitable primary data collection tool to access this knowledge and to enable flexible, in-depth exploration of the issue. Discussion or interview of qualitative research evolve through participant responses. The study consisted of 3 IT-companies. The IT-companies were suggested by the thesis supervisor.

The participants were unaware what the interview was about and it was kept that way so no prepping up was done on the subject for that might minimize or otherwise eliminate the purpose of the study.

2.2 Ethical Considerations

The participants were suggested by each IT-companies themselves.

During the interview, the participants will be informed of the study, that is when the interview begins. With the participants consent, in order to generalize the answers, the interview was recorded by audio. We used purposive questions sampling to identify the web programmers experience in programming field, however no personal or sensitive information regarding the participants will not in any way be published or disclosed, this will apply to the IT-companies details as well.

Identifying the participants or the IT-companies is of no interest in this study and is not needed to perform the research, rather it is a specific subject within web-development about the current awareness of the web application security risks and it is kept that way to ensure the participants human rights are not violated.

The participants are well informed to answer the question freely, meaning the answers did not have to be related to OWASP in any way for it is very likely and it is anticipated that some web-developers might not have heard of OWASP or read their documentation, and it would be unfair to them. With help of semi-structured interview and by having some question to be wider instead of many detailed questions one can bring the conversation more naturally and let the participant themselves to some extent to elaborate and discuss on their own what they know, at their own pace which led to a more relaxed interview.

2.3 The Questions

This section contains all the questions which is asked to the participants during the interview, and also the purpose behind each question.

1. How long have you been a web-developer for?
    - The purpose of the question is to find out the web-developers experience in web-development field
2. What does web security mean to you?
    - What is the perspective of web-developers on web security, is it means of protection for the users, data, servers, or to protect all three?
3. What are the advantages of implementing web security?
    - Web security is good for what? The question is aimed to find out from the web-developers what they believe is the advantage of web security, this is to gain further understanding of the web-developers perspective on subject of web security and furthermore to assess credibility in results.
4. Should a web-developer (in this case you) trust the end user? Why, why not?
    - Note that this question contains a follow up question. The end users could have a financial gain, in other words, meaning the theft of credit card numbers, or corporate espionage, when hackers seek to steal sensitive information, is it wise, then, to trust the end-user when such possibility exists?
5. Do you recognize OWASP? What is it that they do?
    - The Open Web Application Project, or OWASP [5] is an international organization dedicated to web application security as previously introduced 1.1.2, this question will help the interviewer on how to formulate the next upcoming question, if the participant does not recognize OWASP.
6. Can you name a security risks in the web application that is in the top-3?
7. How does this type of attack/risk occur?
    - This is to find out if they're aware of the web security risk that is most critical in web applications and further more if they are able to identify why the vulnerability exists or how it happens.
8. What other security risks in web applications do you know?
9. How does this type of attack/risk occur?
    - Same purpose as before from previous question, however, this time it is not restricted to OWASP top 10 or just one attack, here the participants share what they know about web application security risks, whether it be from OWASP or not.
10. What can you as a web-developer, in this case do, in order to minimize or counteract these risks/attack?
    - For many web-developers, the driving goal centers around getting the application to work, focusing less on whether or not it is secure. So the question is to find what steps do they take to ensure the application are secure from risks? They are free to answer this question however they like, they could relate it to the previously discussed security risks that they mentioned or not.

3 Results

This chapter will include the results from the interview, that includes the participants generalized response. The participants will be categorized with ID in the alphabetical order.

3.1 Interviewees Response

1. **How long have you been a web-developer for?**
    - Company A
        - Participant A-A: Around 12-14 years.
        - Participant A-B: Around 1-2 years.
        - Participant A-C: Around 2 years.
        - Participant A-D: Around 10 years.
    - Company B
        - Participant B-A: Around 4-5 years.
        - Participant B-B: Around 1-2 years.
        - Participant B-D: Around 1-2 years.
    - Company C
        - Participant C-A: Around 10 years.
        - Participant C-B: Around 15 years.
        - Participant C-C: Around 2-3 years.

2. **What does web security mean to you?**
    - Company A
        - Participant A-A: "To validate, sanitize data, to use tokens and to protect against Cross-Site-Scripting (XSS) attack."
        - Participant A-B: "That the users data to be stored in a secure way, no plain texts, the data that is sent should be encrypted."
        - Participant A-C: "It depends on what the type of data is that you are working with, the data should not be public. To ensure the users that their data is safe. No data leakage."
        - Participant A-D: "To build secure applications."
    - Company B
        - Participant B-A: "To build secure applications, to not allow unauthorized users access."
        - Participant B-B: "Everything from security on the client all the way through to the database, ensuring the users data is safe."
        - Participant B-C: "Security on the web."
    - Company C
        - Participant C-A: "Integrity for the users, to secure the data so it does not leak out, managing passwords and bank information."
        - Participant C-B: "To protect the users data from unauthorized access."
        - Participant C-C: "Access to the actual data content of the web-site and performance."

3. **What are the advantages of implementing web security?**
    - Company A
        - Participant A-A: "Protecting the data."
        - Participant A-B: "To make sure that the users data is handled correctly, this builds confidence in the users and in a way gain their respect. Minimising the risk of the data to be handled in a wrong way."
        - Participant A-C: Response from interviewee was obscure.
        - Participant A-D: "This is up to the client, whether he wants these advantages."
    - Company B
        - Participant B-A: "Unauthorized users can not get in the system, the data should not leak outside. The application is not exploitable."
        - Participants B-B and B-C: Both interviewee responded with, "protecting the end-user".
    - Company C
        - Participant C-A: "Security for the users, and also to us."
        - Participant C-B: The interviewee gave same response that was given to the previous question.
        - Participant C-C: "No users get access to unauthorized content, a content that does not belong to them or a content that they should not be allowed to access. Also to ensure the data is not malicious, by validating it."

4. **Should a web-developer trust the end-user? Why or why not?**
    - Company A
        - Participant A-A: No! The interviewee in response to the follow up question said, "to protect ours and the users content. Protect the servers. To exclude files that we do not want in our system or database".
        - Participant A-B: No! The interviewee in response to the follow up question said, "the end-users knowledge can be very varied. There are those that know their way around the web and are able to navigate online, and know how to manage their data, while there are also those who are new to the internet, and they do not know how to handle their data, so therefore it is important that we manage their data in a right way for them".
        - Participant A-C: No! The interviewee in response to the follow up question said, it is very likely that someone will want to try to get inside.
        - Participant A-D: No! The interviewee in response to the follow up question said, "it depends on the application you are building, if it is a public web application then you assume that there are groups of people who have bad intentions to harm".
    - Company B
        - Participant B-A: No! The interviewee in response to the follow up question said, "some individuals have good knowledge on what to do to exploit, you have to always keep in mind, there is always that someone who wants to cause damage".
        - Participant B-B and B-C: Both interviewee said no! Also, in response to the follow up question said same thing, "always to be skeptical, there are always those who seek to exploit things".
    - Company C
        - Participant C-A: Maybe! The interviewee in response to the follow up question said, "it depends on who the end-user is, if it is someone that I know".

  - Participant C-B: No! The interviewee in response to the follow up question said, "you never know how the end-users are going to use your application, there is a lot to consider because of all the possibilities of how they might use the application".
  - Participant C-C: No! The interviewee in response to the follow up question said, because it could be the end-user have little knowledge or experience to know what they are doing or they have that and know what they are doing.

5. **Do you recognize OWASP? What is it that they do?**
   - Company A
     - Participant A-A: Yes! The interviewee was able to identify what they do and said they are working with web security in web applications, providing guidelines, as well as how to prevent the risks.
     - Participant A-B: No!
     - Participant A-C: Little bit! The interviewee was able to identify one of their project and said they present ten points and these ten points is what you need to consider in order to make your application secure.
     - Participant A-D: Yes! The interviewee said, it is web security related, where you can get access to what you can do.
   - Company B
     - Participant B-A: Yes! The interviewee put it as: that they had to do with web security identifying, and offer documentation.
     - Participant B-B: Yes! The interviewee was only able to identify their release for documentation for the most critical web security risks.
     - Participant B-C: Yes! The interviewee was able to recognize they had to do with web security but was not able to elaborate more on the subject.
   - Company C
     - Participant C-A: No!
     - Participant C-B: Yes! The interviewee was able to identify what they do and said they offer documentation and have a list of top-10 vulnerabilities and solutions on how to protect the application.
     - Participant C-C: No! After explaining what it is they do, the interviewee recognized the description, but was not so fully aware of what they have to offer.

6. **Can you name a security risks in the web application that is in the top-3?**
   - Company A
     - Participant A-A: The interviewee said Injection Attack.
     - Participant A-B: The interviewee said Injection Attack.
     - Participant A-C: The interviewee was not able to name a security risk from top-3.
     - Participant A-D: The interviewee said Injection Attack.
   - Company B
     - Participant B-A: The interviewee said XSS (Cross-Site Scripting).
     - Participant B-B: The interviewee said Injection Attack.
     - Participant B-C: The interviewee said XSS (Cross-Site Scripting).
   - Company C

- Participant C-A: The interviewee said Injection Attack.
- Participant C-B: The interviewee said Injection Attack.
- Participant C-C: The interviewee was not able to name a security risk from top-3.
  -

7. **How does this type of attack/risk occur?**
   - Company A
     - Participant A-A: "when the data comes from the client, the data comes in through query parameters without any validation or it comes through an input-field, the data is not taken care of appropriately before it is executed against a database."
     - Participant A-B: "That the user inserts into an input-text form an SQL query, that is not validated, and it gets injected straight into the database."
     - Participant A-C: Since the interviewee was not able to name a security risk, this follow up question was therefore not asked to the interviewee.
     - Participant A-D: "Making a post, injecting an SQL command, or scripting command, and if the command is not sanitized then it is executed against the server, you could gain access to an entire database."
   - Company B
     - Participant B-A: "n example if you post JavaScript code and it is not validated and it gets injected into the web application and when users visit the website, the JavaScript code on execution gives the attacker access to the users cookie. The script that was injected sends the users cookies to the attacker server."
     - Participant B-B: "if I do not sanitize the input data from the end-user, and it goes straight to the database, it causes damage."
     - Participant B-C: The interviewee simply put it as: "hijacking the cookie."
   - Company C
     - Participant C-A: "to post a script into a web-server"
     - Participant C-B: "the input data comes from the end-user and when without being sanitized, it gets executed against the database."
     - Participant C-C: Since the interviewee was not able to name a security risk, this follow up question was therefore not asked to the interviewee.

8. **What other security risks in web applications do you know?**
   - Company A
     - Participant A-A: The interviewee did not mention a name but explained how the attack occurs at next question.
     - Participant A-B: Vague response from the interviewee.
     - Participant A-C: The interviewee said Injection attack
     - Participant A-D: Interviewee mentioned something about faking your identity, the attacker is sitting in the middle.
   - Company B
     - Participant B-A: The interviewee said Injection attack.
     - Participant B-B: The interviewee said session Hijacking.
     - Participant B-C: The interviewee was unable to name other risks.
   - Company C

  - Participant C-A: The interviewee said Brute-force attack.
  - Participant C-B: The interviewee said Cross-Site Scripting (XSS)
  - Participant C-C: The interviewee said Brute-force attack

9. **How do these type of risk/attack occur?**
   - Company A
     - Participant A-A: The interviewee put it as: the attacker can gain access to the system by uploading a file through file uploading service, that gets executed on the server.
     - Participant A-B: Vague response from the interviewee, unable to interpret.
     - Participant A-C: The interviewee did not describe the SQL injection, but went on describing another threat about taking advantage of the CPU optimization of code execution.
     - Participant A-D: The interviewee said something about setting up a fake network, where you (referring to the attacker) have access to others data traffic as it passes through your computer.
   - Company B
     - Participant B-A: The interviewee said, sending query strings without escaping it, to the system or database, without any validation and it gets executed.
     - Participant B-B: The interviewee was not able to confirm it but said the attacker some how manages to get in malicious content into the web application, so the scripts get executed when users visit the web application.
     - Participant B-C: The interviewee was not able to name other risks or attacks but was able to elaborate a type of attack and said, when visiting a website, you can set up an i-frame over the actual website. The user sees the ordinary page but the content in the background is not the same, e.g. the attacker sees what the user types in the input field.
   - Company C
     - Participant C-A: The interviewee said, an automated bot that either tries to guess the users passwords or have a list of common passwords and tries them until correct combination is found.
     - Participant C-B: The interviewee said, an end-user injects a script through a form of an input field and is able to steal other users-cookie on the script execution.
     - Participant C-C: The interviewee put it as: a bot of some sort that makes excessive posts that the server can not manage.

10. **What can you as a web-developer, in this case do, in order to minimize or counteract these risks/attacks**
    - Company A
      - Participant A-A: The interviewee said to look at the mime-type, to validate the files we are receiving. Set policies on the files
      - Participant A-B: The response from the interviewee was: Keep myself updated with what type of attacks that exists. To read the documentation of OWASP, and find out what they recommend in order to secure a web service.
      - Participant A-C: The response from interviewee was: It is good to keep an eye out on OWASP list.

  - Participant A-D: The interviewee said, to read a lot to gain knowledge, find out what you can do and what you can not do.

- Company B
  - Participant B-A: The interviewee said to use well tested frameworks and software. It is better this way because you might miss a risk if you are not fully aware of all the vulnerabilities. Follow best practises.
  - Participant B-B: The interviewee said to always be skeptical of the end-users intentions. To read the OWASP top 10 documentation, and to follow best practises.
  - Participant: The interviewee said, to talk about web security, develop web security things, or look at OWASP recommendations.
- Company C
  - Participant C-A: The interviewee responded with solution to the earlier attacks that were discussed. In regards to brute-force attack the interviewee said to speed up and see where the traffic is coming from in order to block it. In regards to the SQL injection, the interviewee said to validate and sanitize user input.
  - Participant C-B: The interviewee said to validate the input both on the client and server side, and to keep up to date with OWASP list. Use secured frameworks that offer security against risks.
  - Participant C-C: The interviewee said, it is hard for someone individually to be up to date with the risks. It is very good to use the resources like what OWASP had to offer and take guidance from it. To make use of software that exists out there that helps with finding these loophole in the web application.

4 Analysis

In this chapter the results will be analyzed. The information that was gathered from the participants will be divided into three parts: the perspective of web-developers on web security, awareness of web security risks as well as understanding how these risks occur, and lastly what course of action is taken to minimize the risks.

4.1 Web-developers perspective on web security

The definition of web security is referenced in 1.1.1, however this is not to say that this is the right or wrong answer, it is only to gain an insight whether the perspective of the web-developers is on the same page or is close to the definition of web security.

Results from second questions revealed that half of the participants response in general was to protect the application from exploitation while the the other half was about protecting the data, could they both have meant the same thing? If we now turn to the results from third questions in which less than half of the participants suggest that it is an advantage to protect the application. It is hard to determine but together these results suggests that it is possible that they are referring to the same thing, and that is to protect the data, is to protect the application, it goes hand in hand together.

4.2 Web-developers awareness of web application security risks

The OWASP top-10 most critical security risks is introduced in 1.1.2. The questions this thesis aims to answer is if web-developers are aware of application security risks and in order to answer this question, these set of questions were devised to answer the problem:

- Should a web-developer trust the end-user? Why or why not?
- Can you name a security risks in the web application that is in the top-3?
- How does this type of attack/risk occur? *Follow up question to previous question*
- What other security risks in web applications do you know?
- How do these type of risk/attack occur? *Follow up question to previous question*

Results shows that more than half of the total participants recognized the OWASP, some could identify what they do better than others. The result for if a web-developer should trust the end-user, only one said maybe. which is somewhat counterintuitive, and is a risk, however the other responses are very reassuring, and in general their reason for not trusting the end-users was because the end-users could try to gain unauthorized access, initiate an attack or exploit the application. Another reason was because the end-users might not know how to protect themselves, an example would be the end-user choosing a weak password, unaware that cybercriminals are using automated attacks such as brute force attack to try various combinations of usernames and password until it eventually gets it right, so the end-user does not consider a strong password, and it becomes the duty of the web-developer to set policies to deal with weak passwords, enforcing the end-user to choose a strong password for their account.

Concerning the top-3 most critical web application security risks, only 6 participants responded correctly and said Injection attack, two participants said, Cross-Site Scripting which was in the top 3 in the OWASP list of top-10 back in 2013 [5]. Last two participant could not name any risks.

The results of how the attack occurs, the majority could describe how the attack takes place, it is due to not validating the incoming data from the client, and it results in the attacker gaining unauthorized access to the data.

The overall results are varied but promising none the less, the brute-force attack and Cross-Site Scripting as well as Injection attack are mentioned and the results on how these attacks occurs it is mostly correct, some gave more detailed and accurate response. Worth keeping in mind that the brute-force attack [11] is not a web application risk but a method used by attackers to exploit some of risks that are mentioned in the OWASP [5] top-10 list, a threat none the less and also requires protection, both gave two different responses to brute-force attack and both are correct according to [11]. Overall it shows there is a concept that is followed, that is validating things all the way from the client side to the server side and that is a reassuring thing.

4.3 Web-developers on minimizing web application security risks

This is answered by the last question that is presented to the web-developers and the results are very promising from two aspects, firstly the majority of the participants agreed on one thing and that is to read OWASP recommendations, that is positive to hear because that is one way to keep up to date with the most critical web security risks that exists in web applications and how to prevent them, secondly whats reassuring is the A-A, B-A, C-A, C-B and C-C participants responded to minimizing risks that they mentioned, this could also indicate they are well aware of the security risk and are able to prevent it.

5 Discussion and Conclusion

A brief definition of web security is introduced 1.1.1 in order to help us with with the first set of questions to gain an insight into web-developers point of view on web security, these findings raise intriguing questions regarding whether or not they share the same concept on web security, and the findings suggests that they share the same concept, that is if protecting the data is the same as protecting the application or the other way around, and this could also suggest that there is a concern among web-developers to protecting the sensitive data.

The next set of questions was to answer the main question: *Web developers awareness of web security risks*. Are web-developers aware of the top-10 most critical web application security risks, and the OWASP offers just that, and researched needed to be done on these vulnerabilities to confirm what the web-developers had to say. This thesis initially states the definition of web security and the top-10 most critical security risks in web application and then presents the methodology for the research.

The results is somewhat disappointing, due to not a single participant managed to name 4 or more web security risks. The awareness bar for web security risks is not as it should be. It is troubling, and that is from the perspective of OWASP, while almost all participants could describe how some attacks or risks occurs, it is still somewhat troubling due to the small number of security risks mentioned.

Injection attack is mentioned 8 times, this findings may help us to understand that there is some concern or awareness for web security risks. The Injection attack is very easy to pull off, and the results can be very catastrophic, for both the company and the users who are effected by it, if the web application is not properly protected, and have some validate checks for user-input, but that is not the only risk which can be catastrophic for both parties.

Why is it that the web-developers could not name 4 or more of the most critical security risks in web application, is it possible that the web-developers are depended on frameworks to do the job of managing security? Some frameworks have built-in security features that web-developers can make use of to help secure the application against cyberattacks. This way the web-developer can focus on building the application without having to worry so much about the security part.

It is also possible that they are simply not aware of the security risks in web applications, however it is very reassuring should the web-developer keep up to date with the OWASP top 10 list and get familiar with the web security risks as they said they would in response to the last question, for it would help in mobilizing the application against cyberattacks and that will benefit both the users, and the IT-company in avoiding bad reputation, law-suits, or worst case scenario, bankruptcy.

In conclusion, the web-developers awareness of the security risks in web applications is less than half, of the top-10 security risks list from OWASP perspective, it is worth keeping in mind, perhaps, a bigger scale for the research area might produce a different sets of results.

5.1 Future Work and Final Words

It is hard to fully determine the reality of the situation on this scale, it will benefit many to know how the situation really is, for it would raise a great amount concern depending on the situation, if it is critical or not concerning the web-developers awareness of web security risks, otherwise it would be unfair for the researcher, web-developers, the IT-companies and also the reader, the researcher can only offer from what was gathered. Therefore, and this is an important issue for the future work, the next initiative should be to research on the subject on a larger scale for a better clarity of the situation. Future studies on the current topic are recommended to use the qualitative method to get honest results through discussions with the participants. Furthermore it is recommended not to reveal what the subject is about until

the interview begins with the interviewee, that way, no prepping up is done, leading to the response from the participants will be honest and true, that too will benefit the research as well.

References

[1] J. Murphy and M. Roser, "Internet", *Our World in Data*, 2019. [Online]. Available: https://ourworldindata.org/internet (visited on 2019-06-01).

[2] P. R. Clearinghouse, *Data breaches*, 2019, 2018, 2017, 2016. [Online]. Available: https://www.privacyrights.org/data-breaches (visited on 2019-06-01).

[3] S. L. Garfinkel, "The cybersecurity risk", *Commun. ACM*, vol. 55, no. 6, pp. 29–32, Jun. 2012, ISSN: 0001-0782. DOI: 10.1145/2184319.2184330. [Online]. Available: http://doi.acm.org.proxy.lnu.se/10.1145/2184319.2184330.

[4] R. Oehme, *Cyber security threat and crime development in sweden*, Mar. 2018. [Online]. Available: https://www.pwc.se/sv/cyber-security/sakerhetsgalan.html (visited on 2019-05-11).

[5] OWASP, "OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks", [Online]. Available: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf (visited on 2019-05-06).

[6] J. Tammany, *What is website security?* [Online]. Available: https://www.sitelock.com/blog/2018/10/what-is-website-security/ (visited on 2019-05-06).

[7] Acunetix, *Web Application Vulnerability Report*, Mar. 2016. [Online]. Available: https://d3eaqdewfg2crq.cloudfront.net/resources/acunetix-web-application-vulnerability-report-2016.pdf (visited on 2019-05-15).

[8] *2018 vulnerability statistics report*, 2018. [Online]. Available: https://www.edgescan.com/wp-content/uploads/2018/05/edgescan-stats-report-2018.pdf (visited on 2019-06-01).

[9] M. S.Jalali, *The trouble with cybersecurity management*, Aug. 2018. [Online]. Available: https://sloanreview.mit.edu/article/the-trouble-with-cybersecurity-management/ (visited on 2019-06-06).

[10] M. Bravenboer, E. Dolstra, and E. Visser, "Preventing injection attacks with syntax embeddings", in *Proceedings of the 6th International Conference on Generative Programming and Component Engineering*, ser. GPCE '07, Salzburg, Austria: ACM, 2007, pp. 3–12, ISBN: 978-1-59593-855-8. DOI: 10.1145/1289971.1289975. [Online]. Available: http://doi.acm.org/10.1145/1289971.1289975.

[11] I. U. Rehman, *What Is A Brute Force Attack?* [Online]. Available: https://www.cloudways.com/blog/what-is-brute-force-attack/.

[12] M. R. Franco Callegati Walter Cerroni, *Man-in-the-Middle Attack to the HTTPS Protocol*, vol. 7, pp. 78–81.

[13] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, "Foundations of json schema", in *Proceedings of the 25th International Conference on World Wide Web*, ser. WWW '16, Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 2016,

pp. 263–273, ISBN: 978-1-4503-4143-1. DOI: 10.1145/2872427.2883029. [Online]. Available: https://doi.org/10.1145/2872427.2883029 (visited on 2019-04-23).

[14] J. Bundgaard, *Increase in cyber crime against swedish companies in 2018*, Apr. 2018. [Online]. Available: https://www.pwc.se/sv/cyber-security/hackerattack.html (visited on 2019-06-02).