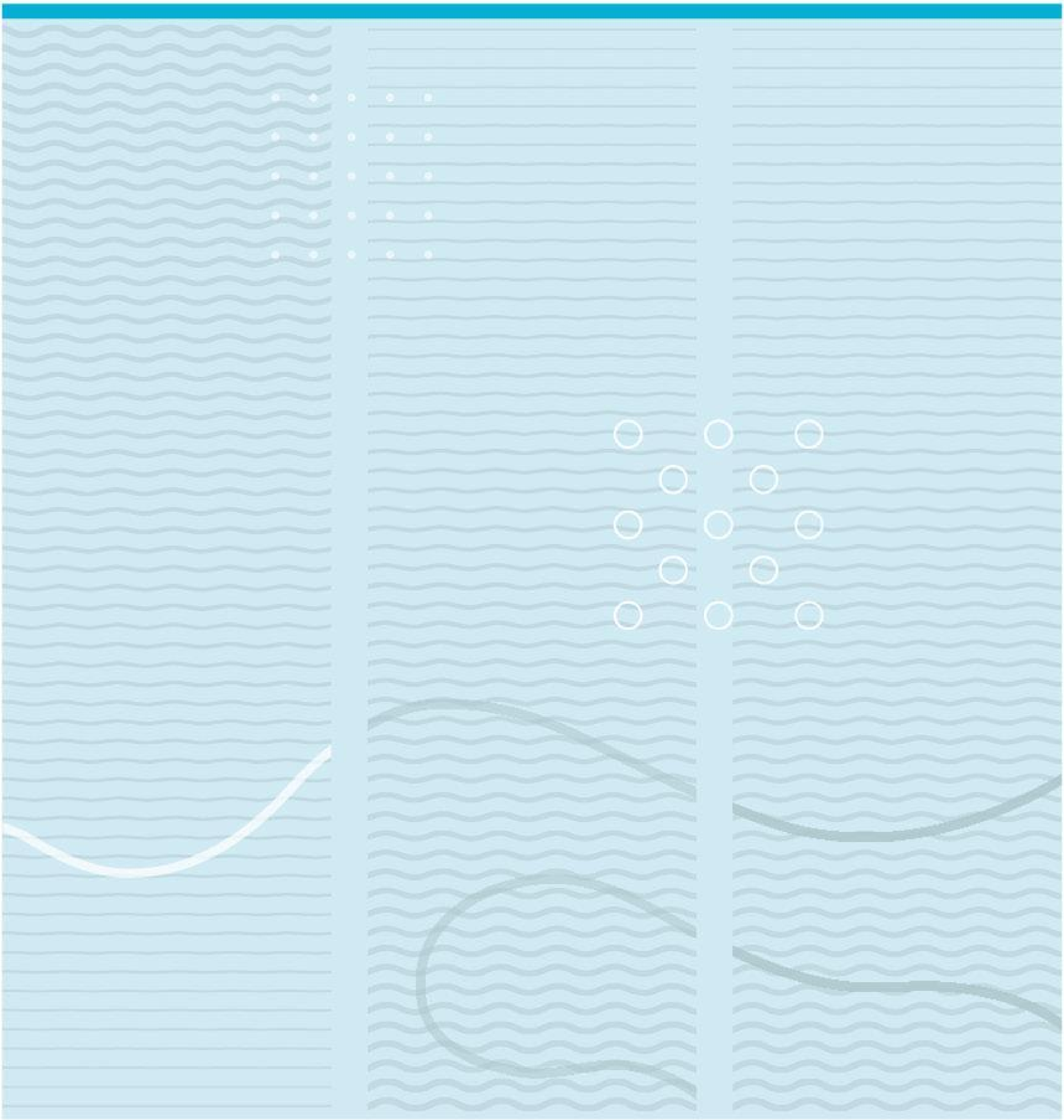
University of South-Eastern Norway
USN School of Business
Department of Economics, Marketing and Law
–

Master’s Thesis
Study Programme: Management Information System
Autumn 2022

Pratik Sapkota & Anjeela Shova Sthapit

Analysis of Web Application Security Management in Context of Nepal’s Organizations



University of South-Eastern Norway
USN School of Business
Department of Economics, Marketing and Law
PO Box 235
NO-3603 Tonsberg, Norway

http://www.usn.no

This thesis is worth 30 study points

Foreword

The University of South-Eastern Norway's School of Business required the completion of this thesis as part of the requirements for the Master of Science in Management Information Systems degree. It has been a long road to get to this thesis. Numerous people we have encountered along the journey have motivated us and given us unexpected encouragement. Too many names exist to list them all. This section is devoted to a few of those we have encountered on the journey.

We want to thank our mentors and supervisors, Associate Professor Rania EI-Gazzar and Professor Anh Nguyen Duc, for their dedication and encouragement. They demonstrated remarkable dedication to and interest in our master's thesis and have always been readily available. Also, we would like to express our gratitude to all the lecturers of the University of South Eastern Norway. We would also like to appreciate the respondents for taking the time out of their busy schedules to participate in the research. The feasibility of this study would not have been achievable without their involvement and input. We also want to express our gratitude to all our classmates and friends for their support of our education. Moreover, without our parents' support, we never would have begun the road that led to this thesis. They seeded the value of perseverance in us from a young age. They have motivated us to maintain our curiosity, continue exploring, and seek understanding.

Lastly, we have made an incredible team and want to thank one another for our successful teamwork as students at the University of Southeast Norway. We have made an equal contribution to every part of this thesis. For better or worse, we balanced one another out and always engaged in frank, direct, and critical conversation. For better or worse, we balanced one another out, and we constantly engaged in frank, direct, and critical conversation. Having someone to depend on has been beneficial and safe, mainly through the challenging portions of the process. After a difficult and intriguing effort, we are pleased with the outcome. This thesis was interesting to work in part because of the topic we have explored. We now have grown in knowledge and belief to have a solid understanding of the subject we chose to explore.

Pratik Sapkota & Anjeela Shova Sthapit
Hønefoss, 9 October, 2022.

Abstract

We use web-based applications in various aspects of our lives, including banking, healthcare, sports, entertainment, media, learning, commerce, and so much more. As a result, it has increased the use of web applications for many tasks and daily activities. These applications contain sensitive and essential data that needs to be safeguarded. In numerous sectors of Nepalese society, cyberattacks and threats have been gradually increasing. Security has been mostly neglected despite being a crucial aspect while developing web applications. This thesis aims to study how the Organizations of Nepal perceive and practice web application security management. This thesis investigates the use of Open Web Application Security Project's (OWASP) related security practices among security experts in Nepal to understand how different practices, approaches, and mitigation of security vulnerabilities are employed in Organizations in Nepal. This thesis includes the study of both technical and non-technical aspects related to web application security management. The study followed a mixed method approach, i.e., a sequential explanatory research approach. A survey was conducted first, in which eighty-seven valid responses were obtained. Then interviews with six security experts were conducted to understand the context better. We found that many Organizations do not follow standard security practices and lack the necessary experience in secure coding, which might lead to security-related issues. We also discovered that Organizations did not consistently consider security throughout each stage of the software development cycle. We found different factors that have been affecting secure practices in the context of Nepal's Organizations which were Human Factors (Security Knowledge and Training, Security Awareness, Attitudes, Beliefs, and Behavior, Motivation, and National Culture), Policies, Organizational Communication, Experiencing a Security Incident, Technology Advancement, and Resources Constraint (Budget, Time, Manpower). This study contributes to the field of information security research.

Keywords: *Web application, Security, Management, OWASP, Vulnerabilities, Top 10 Vulnerabilities, Nepal, Secure Software Development Lifecycle;*

Table Of Content

1. Introduction **1**
1.1 Motivation of the Research 3
1.2 Problem Statement 4
1.3 Research Aims and Objectives 5
1.4 Research Question 6
1.5 Significance of Study 6
1.6 Thesis Structure 7

2. Theoretical Background **8**
2.1 Web Application 8
2.1.1 Evolution of Web Applications 8
2.1.2 Working Mechanism of Web Applications 9
2.1.3 Vulnerability Assessment and Penetration Testing (VAPT) 9
2.2 Security 10
2.2.1 People, Process, and Technology 11
2.2.2 Web Application Security 12
2.3 Open Web Application Security Project® (OWASP) 12
2.3.1 OWASP Top 10 Web Application Vulnerabilities 13
2.3.1.1 Broken Access Control 13
2.3.1.2 Cryptographic Failures 14
2.3.1.3 Injection 14
2.3.1.4 Insecure Design 14
2.3.1.5 Security Misconfiguration 15
2.3.1.6 Vulnerable and Outdated Components 15
2.3.1.7 Identification and Authentication Failures 15
2.3.1.8 Software and Data Integrity Failures 15
2.3.1.9 Security Logging and Monitoring Failures 16
2.3.1.10 Server-Side Request Forgery 16
2.3.2 Vulnerabilities Countermeasures 16
2.3.3 Secure Coding Practices 19
2.3.4 Secure Software Development Life Cycle 22

3. Related Works **25**
3.1 Research Gap 29

4. Methodology **31**
4.1 Research Design 31
4.2 Research Strategy 32
4.3 Research Approach 32
4.3.1 Quantitative Approach 32

4.3.2 Qualitative Approach 33
4.3.3 Mixed Method Approach 33
4.3.4 Ethical Considerations 35
4.4 Data Collection 36
4.4.1 Sampling Procedure 36
4.4.2 Quantitative Approach 37
4.4.2.1 Survey Questionnaires: Validity and Reliability 38
4.4.2.1.1 Phase one: Validity testing 39
4.4.2.1.2 Phase two: Pilot Study 40
4.4.3 Qualitative Approach 41

5. Data Analysis 44
5.1 Quantitative Data Analysis 44
5.1.1 Descriptive Statistics of Demographic Sample 44
5.1.2 Descriptive Analysis of the Survey Answers 50
5.2 Qualitative Data Analysis 62
5.2.1 Descriptive Analysis of the Interview Answers 63
5.3 Validity and Reliability 71

6. Discussion 72
6.1 Summary of Findings 72
6.1.1 Best Practices versus Current Practices 72
6.1.2 Factors Affecting Current Security Practices 79
6.2 Limitations 84
6.3 Implication for Practice 84
6.4 Implication for Research 85

7. Conclusion 86

References 87

Appendix A: Survey Questionnaires 100

Appendix B: Interview Questions 108

Table of Figures

Figure 1: Working Mechanism of Web Applications 9
Figure 2: People, Process, and Technology 11
Figure 3: OWASP Top Ten Vulnerabilities 13
Figure 4: Software Development Life Cycle (SDLC) testing workflow 23
Figure 5: Research Workflow Design 31
Figure 6: Sequential Explanatory Design Model 35
Figure 7: Pie Chart Representing the Primary Role of Respondents 44
Figure 8: Bar Chart representing the number of years of work experience 46
Figure 9: Pie Chart Representing Organization's primary industry 46
Figure 10: Bar Chart representing the number of people working in an Organization, either as employees or consultants 47
Figure 11: Size of respondents' web application development team in the Organization 47
Figure 12: Bar Chart representing the number of years respondents' Organizations have been practicing web application security 48
Figure 13: Pie chart showing how frequently participants' Organizations assess the security of their web applications 48
Figure 14: Bar Chart representing Respondents' Organization Spending 49
Figure 15: Pie Chart representing the proportion of Organizations where outside researchers can report security flaws in their Web Application 49
Figure 16: Analysis of General Practices 50
Figure 17: Participants' familiarity with OWASP top 10 Vulnerabilities 51
Figure 18: Analysis of Broken Access Control Vulnerability 54
Figure 19: Analysis of Cryptographic Failures Vulnerability 55
Figure 20: Analysis of Injection Vulnerability: 56
Figure 21: Analysis of Insecure Design Vulnerability 57
Figure 22: Analysis of Security Misconfiguration Vulnerability 58
Figure 23: Analysis of Vulnerable and Outdated Components Vulnerability 59
Figure 24: Analysis of Identification and Authentication Failure Vulnerability 59
Figure 25: Analysis of Software and Data Integrity Failures Vulnerability 60
Figure 26: Analysis of Security Logging and Monitoring Failures Vulnerability: 61
Figure 27: Analysis of Server-side Request Forgery Vulnerability 62
Figure 28: Familiarity of OWASP top ten vulnerabilities in Nepal 73

List of Tables

Table 1: OWASP Top 10 Vulnerabilities and their Countermeasures. 16

Table 2: OWASP Secure Coding Checklist 19

Table 3: Primary Roles of Respondents in Organization 45

Table 4: Scoring range of Likert scale of the survey 51

Table 5: Overall Participants’ familiarity with OWASP top 10 54

Table 6: Demographics of the Interviewed Experts. 63

Abbreviations

API	Application Programming Interface
ASP	Active Server Pages
ATM	Automated Teller Machine
CD	Continuous Delivery/Continuous Deployment
CI	Continuous Integration
CIA	Confidentiality, Integrity, and Availability
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identity Document
IPA	Information-Technology Promotion Agency
ISACA	Information Systems Audit and Control Association
IT	Information Technology
JS	Javascript
JWT	JSON Web Token
NIST	National Institute of Standards and Technology
ORM	Object-Relational Mapping
OS	Operating System
OWASP	Open Web Application Security Project
PHP	Hypertext Preprocessor
QA	Quality Assurance
SDLC	Software Development Life Cycle
SMTP	Simple Mail Transfer Protocol
SPSS	Statistical Package for Social Sciences
SQL	Server-site Request Forgery
SSL	Structured Query Language
SSRF	Secure Sockets Layer

TLS	Transport Layer Security
UML	Unified Modeling Language
URL	Uniform Resource Locator
VAPT	Vulnerability Assessment and Penetration Testing
XML	Extensible Markup Language
XSS	Cross-site Scripting

1. Introduction

Nowadays, web-based applications have become a widely implemented option for companies in developing products to offer a range of services to engage customers in almost every sector. It is implemented in their business processes, including enterprise resource planning (ERP), IT management, supply chain management, Customer Relationship Management (CRM), people engagement and many more. (Hassan, Ali, Bhuiyan, Sharif & Biswas, 2022; Jovanovic, Kruegel & Kirda, 2010). These web applications have to store, retain and manage customer data, which is regularly the target of cybercriminals. By taking advantage of vulnerabilities, attackers can access confidential information or knock down the system, causing a substantial financial loss for the company (Leite & Albuquerque, 2018). Vulnerabilities are security faults or weaknesses (Zhang, Li, Ren & Huang, 2021). Web applications are browser-based programs that process data while interacting with client-side and web servers in the user's web browser (PCMag, n.d.). Organizations are incorporating firewalls and other security measures into their web applications as they quickly expand to address web application risks and vulnerabilities. Despite using numerous defensive measures, web applications target 75% of attacks, increasing the trend (Zhang et al., 2021). Due to the complexity of testing methods and many testing instances, there are occasionally problems while developing unfamiliar systems, even when Organizations apply security measures.

According to a report by (Positive Technologies, 2021), cybercriminals can exploit consumers in 98% of web applications, leading to malware distribution, redirecting to a phishing site, or data theft via social engineering. User IDs were the most often revealed sensitive data (84% of incidents), while breaches of sensitive data happened in 91% of web applications. About half of the applications had user credentials leakage, and two-thirds of them had personal data breaches. In 84% of web applications, illegal access was discovered, and in 5% of those instances, complete control over the target website was attained (Positive Technologies, 2021).

In order to understand how different practices and approaches to analyze and mitigate security vulnerabilities are employed in Organizations, this thesis attempts to explore the use of the Open Web Application Security Project's (OWASP) related security practices among Organizations in Nepal. The OWASP top 10 is a guideline for developers on security issues.

Since it is the most recent edition to be published, the 2021 edition of the OWASP Top 10 is considered in this study (OWASP, 2021). To complete this study, we analyzed security experts' knowledge, familiarity, and experience with OWASP-related security practices and processes. According to OWASP top 10-2021, vulnerabilities in web applications can be classified into 1. Broken Access Control, 2. Cryptographic Failure, 3. Injection, 4. Insecure Design, 5. Security Misconfiguration, 6. Vulnerable and Outdated Components, 7. Identification and Authentication Failures, 8. Software and Data Integrity Failures, 9. Security Logging and Monitoring Failures, 10. Server-Side Request Forgery (OWASP, 2021).

Many developers and security personnel lack the necessary experience in secure coding, which leads to security-related issues and becomes one of the main reasons for web application vulnerability (Medeiros, Neves & Correia, 2016). Improvements in security knowledge, skills, and awareness among developers, security experts, IT managers, customers, software managers, and chief information officers have been stressed by Organizations like MITRE (Martin et al., 2010), SANS Institute (Dhamankar et al., 2009), and OWASP (2010). These Organizations achieve this by posting lists of the most recent and frequent programming mistakes. Additionally, the security research community developed tools and techniques to enhance web application security. Although many stakeholders have worked hard to make web applications safer and more secure, we lack enough qualitative and quantitative evidence that the efforts have increased web application security over time. Global digitization is causing an increase in daily dependence on the internet, mainly in least-developed nations like Nepal. In Nepal, there are 20 million internet users as of 2019. In Nepal, cyber security is at high risk because of the government's lack of implementation of the laws and regulations, lack of proper cyber security awareness, and poor security mechanisms and policies on cyber-attacks (Acharya & Dahal, 2021).

This paper seeks to understand the top 10 Open Web Application Security Project (OWASP) vulnerabilities, the mitigation measures that should be implemented, the effects of an attack on enterprises, and if security should be treated seriously. Also, we chose to study how the Organizations of Nepal perceive and practice web application security management concerning OWASP's top 10 security vulnerabilities. It includes the study of both non-technical aspects along with some technical aspects related to web application security vulnerabilities. Moreover, we focus on identifying various vulnerabilities in the web applications developed by different Organizations in Nepal and analyzing the risks of those security loopholes and their impact on the Organizations.

1.1 Motivation of the Research

Security industry analysts suggest that over 70 percent of attacks come through web applications (Sullivan & Liu, 2011). Many big banks in Nepal have been hacked recently, and the money was stolen. Initially, it was just ATM fraud, but the swift server itself was compromised (Schwartz, 2017). After this incident, banking became difficult for some time (Sapkota, Submitted Mandatory Assignment, 2021). Moreover, hackers stole Nepalese Rupees (Rs) 47.3 million from the Agriculture Development Bank of Nepal and Rs 1.2 million from Panas Remittance (Dhungana, 2019). Likewise, some hackers could hack about 68 ATMs in different parts of the country (Lamsal, 2019). Moreover, in reaction to the COVID-19 pandemic, one of the primary triggers of the increased attacks is thought to have been the accelerated introduction of remote workers (Vijayan, 2020).

On 2017, a gang of hackers infiltrated the Department of Passport's official website and vandalized it with a threatening message to expose the government's data. One of the most incredible hacks in Nepal's history again occurred in 2017, when 58 official websites were purportedly taken down (Ghimire, 2017). The NIC Asia Bank, one of the most famous banks of Nepal's SWIFT system, was purportedly compromised by an anonymous hacker in the same year. They stopped the flow of USD 4.4 million to six different countries from the user accounts. Again, in the same year, 2017, a third party gained access to Onlinekhabar, one of the most well-known Nepalese online news sites, and installed a JavaScript mining program to mine the cryptocurrency. On 2020, a hacker breached the food delivery web application and leaked the personal information of 50,000 individuals (Onlinekhabar, 2021). Another recent incident revealed the weakness of the data at Nepal's first and most well-known online payment service provider, which runs a mobile wallet and allows its users to make online and offline payments using their mobile devices or websites. This incident compromised the data of at least twenty users (Republica, 2020). According to the survey findings by Dongol & Chatterjee (2019), Nepal lacks national IT security guidelines and standards that may be used to identify threats and weaknesses in the technology, process, and people. So, in Nepal’s context, it appears that Security is one aspect of the implementation that has been overlooked in terms of web application which has been the biggest motivation for us to choose this topic.

1.2 Problem Statement

A problem statement will often list the drawbacks of the current circumstance and explain why this is important. The use of web applications is increasing. Almost everything is built as a web-based database application. The World Wide Web, or simply the Web as we commonly call it, has a significant role in our daily lives. Our lives have been made simpler using the Web, from a simple search engine to a sophisticated private banking web application. The increasing use of the Internet has undoubtedly transformed and empowered us. The development of the Internet has significantly altered how software has evolved, leading to a more significant presence of information systems in web contexts and, as a result, a rise in security flaws and dangers. In this situation, safe applications have become essential to the market's information systems (R. Silhavy et al., 2019). However, developing a web application may be challenging since the development team has to have exceptional knowledge of the most common security vulnerabilities that web applications might have. These days, different vital infrastructures are compromised by unreliable applications as the complexity of maintaining system confidentiality rises rapidly. Rouge Hackers can steal cookies and session tokens or compromise the database, leading to massive financial and reputation loss (Sapkota, Submitted Mandatory Assignment, 2021).

Many websites do not apply any security policy. Most of those who apply still make mistakes and are still vulnerable to attacks like SQL injection, Cross-site scripting, etc. Most of the time, the vulnerability exists due to a coding mistake. In other instances, it is caused by an outdated dependency. In rare cases, they are 0days (unfixed vulnerability in an application). Hackers take advantage of these poorly coded applications, and access to unauthorized data occurs. The flaws in web applications are prevalent, usually easy to exploit, and easy to fix (Sapkota, Submitted Mandatory Assignment, 2021). Compared to other cybercrimes, carrying out web application attacks is comparatively simple, especially since several attacks may be automated and deployed simultaneously against thousands of distinct targets (Preston, 2022).

Users frequently utilize web applications to access features and data provided by them. The situation may get worse if these services were hacked and all the sensitive data they were holding ended up in the hands of an evildoer. (Dangol & Kautish, 2019) conducted a study in Nepal by surveying consumers about their experiences with e-commerce web applications.

The findings showed that insufficient security practices harm customers' perceptions of the vulnerable. Customers are, therefore, more inclined to refrain from engaging in web applications if they have more knowledge of or experience with cyber-fraud occurrences. In order to guarantee that services will be delivered to consumers with the highest level of security, to secure the integrity of the system, and to guarantee the privacy of online users, there are growing worries about the reliability and security of the built web applications in Nepal. According to the report by (Giri & Shakya, 2020), the danger of a cyberattack and threat has been unexpectedly rising in several sectors of Nepalese society. Making a unified plan is crucial to lowering the rising technical risk associated with cyber. Cybercrime, danger, and attack risk are all relatively high. According to Khatiwada (2021), even the most popular government websites and Organizations in Nepal are prone to severe intrusions; any outside incursion might severely disrupt the whole web application network.

1.3 Research Aims and Objectives

The aim of this study examines how Nepalese Organizations perceive and practice Web application security management. This thesis also seeks to operate as a manual to select suggestions for the most recent best practices that raise awareness and help development teams create more secure Web Applications. The following objectives are formulated to meet the desired aim of this research:

- To study how familiar Organizations of Nepal are with OWASP's top 10 Web application vulnerabilities.
- To analyze to what extent the Organization in Nepal practices web application security and regards security as an essential factor in web application development.
- To investigate the implementation of OWASP-related security practices in the Organizations of Nepal and identify security loopholes within their systems.
- To investigate if Organizations in Nepal focus on deploying security in each phase of Software development life cycle (SDLC)
- To understand Organizations' perspectives on evaluating web application security, the difficulties they face, and the assistance Organizations offer to their employee.

- To find out factors affecting Security Practices in Nepal and find ways to solve them.
- To explore the best security practices and map those practices with the findings.

1.4 Research Question

To analyze and understand how the Organizations of Nepal perceive and practice web application security management, this thesis addresses the research question below:

- How do Organizations in Nepal Perceive and Practice Web Application Security Management?

1.5 Significance of Study

This research aims to contribute to the security field of information systems by addressing vulnerabilities in a way that limits the impact of web application attacks. The findings of this study could benefit the academic field by opening up opportunities for more research into web application security management. Our contribution to scientific knowledge through the large-scale survey includes publicly available survey data for other researchers to explore, the presentation and interpretation of survey analysis findings, and a list of practical recommendations for practitioners and industrial cybersecurity educators. Moreover, for aspiring software developers, this thesis aims to provide a reference to the most recent security practices and how to incorporate strong security principles into their projects. Our study can be used as a benchmark by businesses currently utilizing these methods to measure their progress. This thesis is written to comprehend the current state of web application security issues in Nepal and figure out ways to mitigate them. It will then create a solid baseline for security assessment that will thwart intrusion attempts and malicious adversaries who take advantage of opportunities. Numerous Nepali Organizations might anticipate helping to achieve a more secure Nepal through this report. Moreover, the findings of this study can serve as a guide and literature for policymakers from Nepal and other least developing countries.

1.6 Thesis Structure

This thesis is organized into six sections, each section addressing a different element of the study.

Section 1: Introduction - This section describes the motivation of the research, the problem statement, the study's aims, and objectives, along with the research questions that are introduced in this chapter. It also specifies the significance of the study and describes how this thesis is organized.

Section 2: Theoretical Background - This section describes all the theoretical aspects related to the thesis that serves as the roadmap for our arguments.

Section 3: Related Works - This section includes all the similar pieces of literature done by other researchers and finds out the actual research gap.

Section 4: Methodology - This section presents the research design used in our study, describes the strategy used in our research and describes the methodology used for gathering data.

Section 5: Data Analysis - In this section, we summarize the results of our investigation by analyzing our data.

Section 6: Discussion - This section provides a more in-depth analysis of the findings, discussing the findings in light of the theoretical background described in Chapter 2, their practical applications, and a discussion of the study's limitations.

Section 7: Conclusion - This section summarizes the key supporting ideas discussed throughout the work.

2. Theoretical Background

This section describes all the theoretical aspects of the thesis that serve as our arguments' roadmap.

2.1 Web Application

A web application is any program that may be launched through a web browser over the Internet. Web application development has been responsive, over time, to adopt software engineering tools, practices, and methodologies. Web applications used to be largely static and nothing more than a collection of web pages, but new web technologies, languages, and processes now enable the development of dynamic applications that offer a new paradigm for user participation and collaboration (Jazayeri, 2007). In essence, a web application is divided up into many parts. These components include a web server, the application, and generally a backend data store that the application can access and interact with popular web applications like Google Mail, Facebook, YouTube, Twitter, and Office 365 (Hhs, 2022).

2.1.1 Evolution of Web Applications

Early web-based applications were referred to as "websites," and they consisted of a collection of papers with static content connected by hyperlinks (Jazayeri, 2007). These websites or pages are located on a web server, and a web client application, most often the web browser, can be used by the user to request them. Later, these websites added client and server-side processing languages, including JavaScript (JS), PHP, and server-side development languages like JAVA. As a result, these websites have developed into web applications with server-side and client-side application components that can process and create dynamically adapted content (Shklar & Rosen, 2003). With continuous improvements, web applications offer a framework for delivering features of some other services, which may be obtained from data networks, such as email and file sharing via web protocols. With all these features, web services and applications have developed into very sophisticated, powerful, and complicated sets of entities (Dissanayake & Dias, 2017). A web application can serve multiple users concurrently with less difficulty than desktop programs. Early web applications for mobile devices had some requirements; however, newer technologies minimize these obstacles and quickly develop (techopedia, 2020; Bianco, Lewis, Merson & Simanta, 2011).

2.1.2 Working Mechanism of Web Applications

A browser is used to surf many websites and applications (Rahalkar, 2016). The web page or application loads in the browser shortly after the URL of the website is entered. Based on how it functions, one may then engage with it. Although this procedure appears simple, there is a lot of backend processing and interactions between many entities.

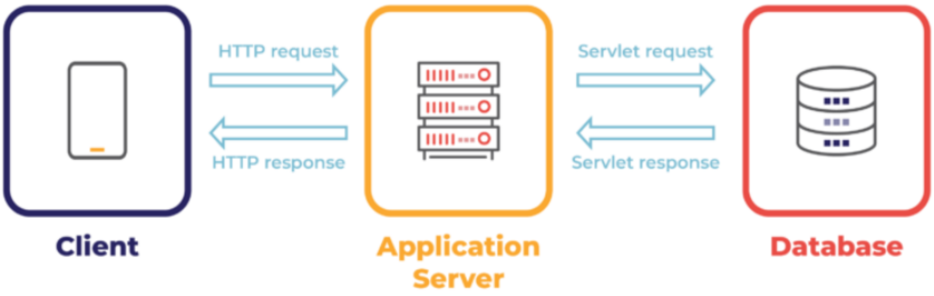


Figure 1: Working Mechanism of Web Applications (Ingalls, 2021)

Here, the user first types in the URL of the website they want to visit into the browser's address bar. The Domain Name System turns the domain name into an IP address (Rahalkar, 2016). After receiving the request, the web server looks at the requested document's extension (such as HTML, PHP, or ASP). If the user requests an HTML document, the web server executes the request and provides the requested HTML page back to the user. The web server will transfer the request to the application server that can handle the requested document if it has an extension that the web server cannot handle. The application server handles the request and could also need information from the database. When processing is complete, the web server returns the result set to the user, and the data is obtained from the database (Rahalkar, 2016).

2.1.3 Vulnerability Assessment and Penetration Testing (VAPT)

Vulnerability is a software system flaw that frequently results from engineering mistakes or unforeseen functionality when combining different components. A hacker can harm the software system unintentionally owing to this kind of flaw (Hoffman, 2020). An exploit is often a chunk of code or a list of commands that may be used to take advantage of a vulnerability (Hoffman, 2020). Vulnerability assessment is a procedure using a proactive and methodical approach to find vulnerabilities. Addressing known and unidentified issues with the system is a common practice (Shinde & Ardhapurkar, 2016). Penetration testing is critical for implementing secure code, especially for consumers testing in Web services contexts where internals are inaccessible (Antunes & Vieira, 2011). Web application security assessments can be carried out either manually or automatically. Organizations should aim for an integrated manual and automated testing methodology to maximize the accuracy of identifying vulnerabilities in web application logs. (Nagpure & Kurkure, 2017). In order to keep protected from the growing cyber threats, (Shinde and Ardhapurkar, 2016) introduced a Vulnerability Assessment and Penetration Testing (VAPT) technique that helps evaluate the effectiveness and ineffectiveness of the security measures of web applications (Shinde & Ardhapurkar, 2016). Security professionals test any organization's security arrangements using the VAPT technique, which simulates an attack and tests the target website's resistance to attacks (Nagpure & Kurkure, 2017). VAPT prevents network downtime caused by breaches and identifies ways hackers can attack the network (CWE List Version 2.9, 2016). Web Application Penetration Testing is expected to reveal vulnerabilities related to OWASP's top ten issues (Nagpure & Kurkure, 2017).

2.2 Security

In the world today, security is essential. One must safeguard a company's intangible assets in addition to its hard assets, which include servers, workstations, network components, and data, as a corporation's reputation, branding, and overall corporate image can be significantly impacted by security breaches (Andress, 2003). The essential security needs are met by what we refer to as the CIA or the security trifecta. The CIA Triad is a foundational cybersecurity paradigm that serves as a basis for creating security regulations to safeguard data. Here, the letter C stands for Confidentiality, I for Integrity, and A for Availability. One must ensure that only individuals with permission can access sensitive data to maintain confidentiality. Integrity signifies that only those with the required authorization may change this private information. Integrity may also assure data authenticity, but only if the data comes from or is received by the designated source. Additionally, availability refers to assuring that one may access computing resources and data without interruption and that data and system will be available (learncisco, 2015). The CIA Triad is so essential to data security that it can result in data breaches or several other security issues if one or more of the three fundamentals are broken.

2.2.1 People, Process, and Technology

According to Andress (2013), "Security is a ubiquitous, continuing process constantly being reviewed and revised in response to changes in the corporate and business environment." People, Processes, and Technology are regarded as the three elements that should be used to create a solid, efficient security architecture that offers layered, defense-in-depth security for an organization (OWASP, 2021).

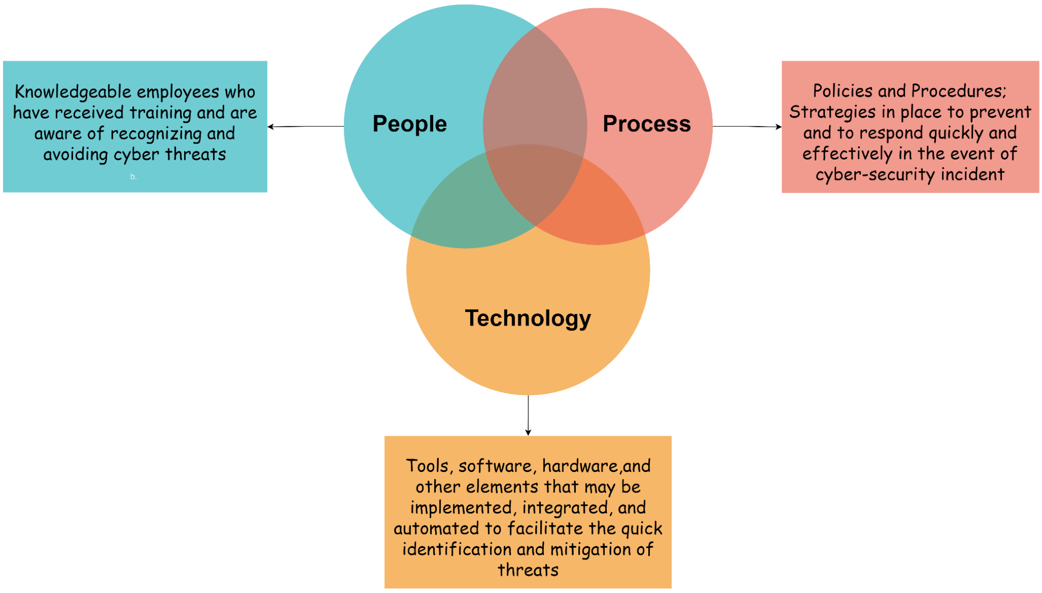

Knowledgeable employees who have received training and are aware of recognizing and avoiding cyber threats
People
Process
Policies and Procedures; Strategies in place to prevent and to respond quickly and effectively in the event of cyber-security incident
Technology
Tools, software, hardware,and other elements that may be implemented, integrated, and automated to facilitate the quick identification and mitigation of threats

Figure 2: People, Process, and Technology (The Human Firewall, 2019)

Even the best technology cannot reduce the danger of attack or compromise without adequately trained people and well-designed procedures/processes. (OWASP, 2021) sees addressing application security as a people, process, and technology challenge since the most successful methods of application security need changes in all areas. Therefore, the proper integration of "people," "process," and "technology," as well as adequate adherence to security policies, are essential components of information security management success (Eminağaoğlu, Uçar & Eren, 2009). Many attacks/breaches have well-known technical solutions. However, they are rarely used efficiently or adequately because people inside organizations are essentially the first and most important line of defense against information security hazards, as technology cannot address them (Tipton & Krause, 2007). Any firm that attempts to reduce information security threats solely through technical remedies will inevitably fail (Mitnick & Simon, 2003). Managers, security professionals, and all other decision-makers now place a greater emphasis on people than technology, reversing this faulty dependence on technology. Several controls on the "human element," most of which involve learning and training. Awareness about information security training for IT professionals and other awareness programs are now considered "must-haves" for everyone. This criterion has been made mandatory by international information security management system standards and related best practices (Eminağaoğlu et al., 2009; OWASP, 2021). People are the cornerstone of information security management in organizations' success or failure.

2.2.2 Web Application Security

The idea entails a set of security measures built into a web application to safeguard its resources from attackers (Preston, 2022; Synopsys, 2022). As internet usage increases, web applications have a rising need for security and reliability (Huang & Lee, 2005). Like any software, web applications inherently have flaws. These flaws are genuine vulnerabilities that may be used against organizations, increasing risk. Such flaws are protected from web application security.

2.3 Open Web Application Security Project® (OWASP)

A nonprofit group called the Open Web Application Security Project® (OWASP) works to improve software security. Through OWASP, like-minded security specialists collaborate to develop a leading practice response to a security issue. The OWASP Testing Project has been under development for many years. The project aims to simplify understanding of the why, when, where, and how of web application testing (OWASP, 2021). The OWASP top ten is a standardized awareness guideline for developers and web application security which reflects a broader understanding of web applications' most significant security vulnerabilities. OWASP continues to be a current, dependable, and instructional address for people and businesses because of its sizable and open community. The foundation's goal is to make the guidelines free and accessible to everyone. It allows everyone to understand how to look for common security problems. It believes the ability to manage security should not be reserved for a small group of people. Not just security experts should be able to utilize it; technical managers, QA, and developers should also be able to (OWASP, 2021).

2.3.1 OWASP Top 10 Web Application Vulnerabilities

Web applications continue to have security vulnerabilities, allowing attackers to obtain sensitive data and use reliable websites as a distribution point for malware (Atashzar, Torkaman, Bahrololum & Tadayon, 2011). Web applications are attacked more than 75% of the time, according to the Information-Technology Promotion Agency (IPA) (ipa, 2011).

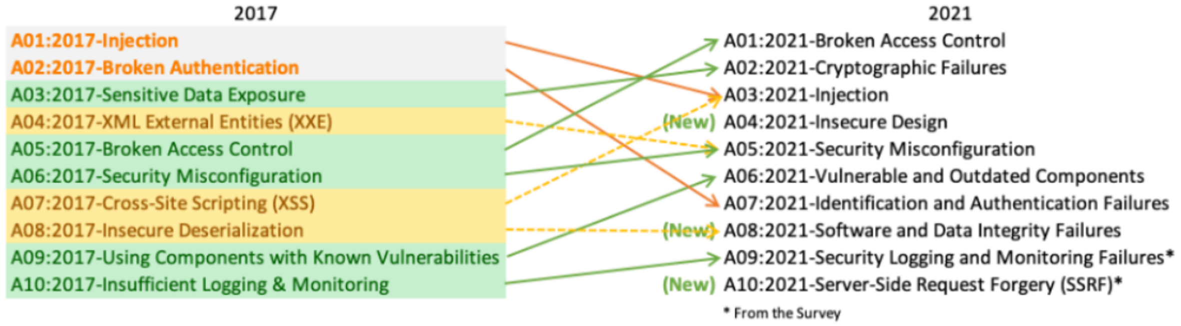

2017
A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring
2021
A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New) A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New) A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New) A10:2021-Server-Side Request Forgery (SSRF)*
* From the Survey

Figure 3: OWASP Top Ten Vulnerabilities (OWASP, 2021)

The OWASP Top 10 Web Application vulnerabilities are described below:

2.3.1.1 Broken Access Control

According to the Open Web Application Security Project (OWASP), Broken Access Control is the most severe vulnerability for web applications since it can lead to session hijacking, which could cause a substantial financial loss and damage the brand of the Organization (OWASP, 2021; Hassan et al., 2022). It includes numerous potential attack vectors, such as sidestepping access control measures, modifying the accounts of other users, raising permissions, which permits unauthorized access to restricted APIs, manipulating metadata through access control tokens or accessing unauthorized websites as an underprivileged user, which can give attackers control over business functions or the potential for attackers to affect simple operations (Hassan et al., 2022). It is advised to use access control lists and server-side code to restrict functionality so that hackers cannot access or manipulate metadata (OWASP, 2021; Velasco, 2019). Lack of compliance with secure designing practices, such as carrying out appropriate input validation, taking precautions to restrict sensitive data disclosure, secure session configuration and management, ensuring control on directory readability, etc. while developing web applications, is the primary cause of a Broken Access Control vulnerability in the application (Velasco, 2019; Hassan et al., 2022).

2.3.1.2 Cryptographic Failures

Attackers usually target sensitive data that is not safeguarded, such as passwords, credit card numbers, and personal information. Cryptographic Failure results in the exposure of sensitive data or system breaches (OWASP, 2021). Keeping cryptographic keys in the source code can occasionally lead to significant security problems because it is easy to find and distribute source code in an enterprise setting. Automatic database encryption is used by an application to encrypt credit card numbers in a database. However, when the data is obtained, it is automatically decrypted, allowing a SQL injection bug to extract the credit card numbers in clear text (Hhs, 2022). The following actions can lead to a cryptographic failure vulnerability:

- Maintaining or sending data in plain text
- Saving data using outdated or inadequate encryption.
- Filtering or hiding data in transit inappropriately (F5, 2022)

2.3.1.3 Injection

A web application can be defended in several ways when it is developed. Attackers can transmit malicious data to an interpreter through injection vulnerabilities, frequently detected in OS commands, XML parsers, SMTP headers, expression languages, and ORM queries. Even if it is presumed that the input came from a trusted source, every function in a web application should validate all input first (OWASP, 2022). Even if it is presumed that the input came from a trusted source, every function in a web application should validate all input first. Data loss, corruption, disclosure to unauthorized persons, loss of accountability, denial of access, and occasionally complete host takeover are all possible outcomes of injection (Coutinho & Pinheiro, 2021). Input sanitization is a cybersecurity solution that involves verifying, cleaning, and filtering data inputs from users, APIs, and web services of undesirable characters and strings to avoid the introduction of hazardous codes into the system (Webopedia, 2020).

2.3.1.4 Insecure Design

Generally, Insecure Design results in remote code execution when the program deserializes user-controllable data. A more incredible threat modeling, safe design patterns and principles, and reference architectures is required if the Organization is actually to defend (Coutinho & Pinheiro, 2021). A flawless implementation cannot address an Insecure Design since necessary security safeguards were never developed to prevent particular attacks (OWASP, 2021). Consider a system of movie theatres that offer discounts for group reservations but requests a deposit for parties larger than twenty. Attackers can use threat modeling to analyze this flow to determine if they can reserve hundreds of seats at other locations in the chain, costing the movie theatre a significant amount of money (Hhs, 2022).

2.3.1.5 Security Misconfiguration

Security misconfiguration carries dangers ranging from unauthorized access to some system data or functionality to a compromise of the entire system (Eshete, Villafiorita, & Weldemariam, 2011). The fact that many web applications are usually hosted in a single (perhaps dangerous) environment when employing shared technologies makes this situation much worse (e.g., a shared web server). A configuration could also be replicated unsafely, which could have risks (OWASP, 2021). The emergence of the platform, infrastructure, and software as a service in the cloud, which necessitates configuring numerous virtual machines there, is a pertinent illustration of this (Eshete, Villafiorita & Weldemariam, 2011).

2.3.1.6 Vulnerable and Outdated Components

Known vulnerabilities in sections, modules, libraries, or software packages are referred to as vulnerable and outdated components (Fortinet, 2021). A development team could not be familiar with or understand all the components involved in their application due to the large number of components used in the development (OWASP, 2021). Some of those components might be outdated and thus vulnerable to attack (Hhs, 2022).

2.3.1.7 Identification and Authentication Failures

Failures in identification, authentication, and session management which are crucial for defeating authentication-related attacks are referred to as Identification and authentication failures (Kumar, 2011). This type of failure is brought on by improperly implemented authentication and session management features, which enable attackers to steal passwords, keys, or session tokens and use them to access user identities (Fortinet, 2021).

2.3.1.8 Software and Data Integrity Failures

Code and infrastructure susceptible to integrity violations are software and data integrity failures. One such instance is an application that depends on plugins, libraries, or modules from unreliable sources and repositories that are improperly checked and may have been modified with or corrupted (OWASP, 2021). It focuses on CI/CD pipelines, essential data updates, and making assumptions about them without checking their integrity. Once the malicious code has been uploaded, this may result in allowing attackers to use the application for their gain. The SolarWinds 2020 supply chain attack, which had a significant global impact on thousands of enterprises, was primarily motivated by this (Fortinet, 2021).

2.3.1.9 Security Logging and Monitoring Failures

Suspicious activity logging is a crucial function of every security system. Lack of or inadequate integration with incident response and insufficient logging and monitoring permits attackers to carry out more attacks on systems, maintain persistence, switch to different systems, and alter, extract, or destroy data (Ridgesecurity, 2021). These mistakes occur when an application cannot recognize and react to security concerns (Fortinet, 2021; Ridgesecurity, 2021).

2.3.1.10 Server-Side Request Forgery

When a web application retrieves data from a remote resource based on a user-specified URL without verifying the URL, the server-side request forgery (SSRF) vulnerability takes place (Jabiyev, Mirzaei, Kharraz & Kirda, 2021). Bypassing security measures, the attacker can make the application submit requests which lead to data theft, sensitive data leaking, and exfiltration of data (Fortinet, 2021; Ridgesecurity, 2021). This category depicts the situation in which members of the security community inform OWASP that this category is crucial even though the evidence at this time does not demonstrate it (OWASP, 2021).

2.3.2 Vulnerabilities Countermeasures

Table 1: OWASP Top 10 Vulnerabilities and their Countermeasures

S/N	OWASP Top 10 Vulnerabilities	Countermeasures
1.	**Broken Access Control**	● Use a token for authorization of users like Json Web Token (JWT) (as they can be signed, they're a good means to securely send information between parties. ● Always deny public access by default except in rare cases for some resources that needed to be accessed publicly (Softwaretestinghelp, 2022). ● Reuse of access control mechanism ● Deny access to private data and information (Hhs, 2022)

2.	**Cryptographic Failures**	● Classify sensitive data according to the privacy laws and business needs ● Protect client and server-side communication by only using stronger ciphers ● Employ forward secrecy (FS) ciphers (an encryption system that changes the keys used to encrypt and decrypt information frequently and automatically) and secure protocols like Transport Layer System (TLS) to encrypt all data in transit (OWASP, 2021) ● Enable cookie encryption (Fortinet, 2021)
3.	**Injection**	● Apply input validation, which checks if the input meets a set of criteria based on a whitelist ● Apply input sanitization, which is a cybersecurity solution that involves verifying, cleaning, and filtering data inputs from users, APIs, and web services of any undesirable characters and strings (OWASP, 2021; Webopedia, 2020). ● Limit user privileges ● Use Templating system (automatically hides all user input before redisplaying it) (OWASP, 2021) ● Enable machine learning for anomaly detection to protect against zero-day injection attacks ● Use parameterized queries to bind all user-supplied data (Karande, 2017)
4.	**Insecure Design**	● Avoid Exposing Direct Object References ● Integrate plausibility checks at each tier of application (from frontend to backend) ● Establish and use a library of secure design patterns or paved-road, ready-to-use components. ● Integrate security language and controls into user stories ● To confirm that all important flows are resistant to the threat model, create unit and integration tests. ● Limit resource consumption by user or service (OWASP, 2021)
5.	**Security Misconfiguration**	● Apply the principles of least privileges for developers, Quality Assurance (QA), and other production environments. ● Have an automated process to verify the effectiveness of the configurations and settings in all environments. ● Limiting unnecessary features, components, and documentation (Hhs, 2022) ● Apply Security Headers on Response ● Protect Cookies by Using the httpOnly and Secure Flags (Karande, 2017)
6.	**Vulnerable and Outdated Components**	● Enable FortiWeb (Web application Firewall) attack signatures to detect attempts to exploit known Common Vulnerabilities and Exposures (CVEs). ● Regularly scan applications for known vulnerabilities using standard vulnerability assessment tools.

		● Continuously inventory the versions of both client-side and server-side components (e.g., frameworks, libraries) and their dependencies using tools like versions, OWASP Dependency Check, retire.js, etc. ● Limit or increasingly delay failed login attempts (Hhs, 2022) ● Scan for vulnerabilities regularly and subscribe to security bulletins related to the components that you use (OWASP, 2021).
7.	**Identification and Authentication Failures**	● Enable credential stuffing protection (the protection against the automated injection of stolen credentials into website login forms) to verify users aren't logging in with previously identified breached credentials ● Have a strong password policy ● Implement multi-factor authentication where possible ● Enable session fixation protection and enforce session timeout ● Enable cookie security "signed" or "encrypted" to prevent session hijacking (Fortinet, 2021). ● When attacks are discovered, administrators should be informed and all log all failures (OWASP, 2021)
8.	**Software and Data Integrity Failures**	● Use digital signatures to verify the software or data source ● To lessen the possibility of malicious code or configuration entering your software pipeline, make sure there is a review procedure in place for modifications to configuration and code (OWASP, 2021) ● Disable autocomplete on sensitive form fields ● Prevent browsers from caching sensitive information (Karande, 2017)
9.	**Security Logging and Monitoring Failures**	● Ensure log data is encoded correctly to prevent injections or attacks on the logging or monitoring systems ● Ensure that logs are generated in a format that log management solutions can easily consume (Hhs, 2022) ● Use threat analytics to increase threat alerts across all web applications (Fortinet, 2021) ● Use digital signatures or similar mechanisms to verify the software or data is from the expected source.
10.	**Server-Side Request Forgery**	● Enable attack signatures to protect against attacks in known applications (Fortinet, 2021) ● Enforce "deny by default" firewall policies or network access control rules to allow essential intranet traffic only ● Sanitize and validate all client-supplied input data (Hhs, 2022; OWASP, 2021).

2.3.3 Secure Coding Practices

According to the US Department of Homeland Security, program coding and design mistakes account for 90% of software vulnerabilities. The cost of these flaws in industrial automation systems alone has been estimated at more than 380 million USD. Software engineers add these errors to the source code since they write software. Conversely, secure coding guidelines prevent software developers from writing unsafe code (Gasiba, Lechner, Pinto-Albuquerque & Mendez, 2021). Developers may create applications and software safely, adhering to secure coding standards and best practices. These guidelines ensure that programmers create secure applications without leaving any potential threats to exploit. It can be used as a security protocol for all software development life cycles and deployment platforms to reduce risks related to poor coding standards (SecureCoding, 2021). The software development lifecycle may be combined with a set of generic software security coding standards that OWASP has described in a checklist style (OWASP, 2010). Some of the strategies and methods offered by OWASP include the following:

Table 2: OWASP Secure Coding Checklist (OWASP, 2010; SecureCoding, 2021)

1. Input Validation: It is a test conducted on input provided by users or the application (SecureCoding, 2021).	● Validate data only on reliable systems (example: Server). ● Check the input's length, data type, and range to a list of permitted characters to validate the data. ● Verify that values in headers of requests and responses are only in ASCII characters. ● Before the data is processed, all client/user-provided data should be validated. (OWASP, 2010).
2. Output Encoding: Output encoding entails translating user input and hazardous data so that it cannot be executed as code by a browser or application when displayed to prevent the execution of untrusted data in a format that the interpreter cannot comprehend (SecureCoding, 2021).	● Conduct all encoding on a trusted system (e.g., The server) ● Unless they are confirmed to be safe for the target interpreter, all characters should be encoded. ● Contextually sanitize all output from SQL, XML searches that use untrusted data. ● Sanitize all operating system command output from untrusted data sources. (OWASP, 2010)
3. Authentication and Password Management: The authentication process allows for the identification of a user's identity. In order to maintain the security and accessibility of some crucial assets/resources, users must adhere to a set of rules and	● All websites and resources that link to CIA Triad should demand authentication. ● Stop using the same password repeatedly. ● Notify users when their passwords are changed. ● Upon the user's subsequent successful login, inform them of the number of failed login attempts. ● Keep temporary passwords with a limited

practices when storing passwords (SecureCoding, 2021).	expiration date so they can be updated at the next login. ● Mandate that passwords be created in accordance with the password complexity policy (OWASP, 2010).
4. Session Management: Web applications use a variety of ways to store and validate credentials for a predetermined amount of time in order to avoid requiring continuous authentication for each page of a website or service. Session management refers to these processes (SecureCoding, 2021).	● Upon logging out, sessions and connections need to always get totally ended. ● The same User ID should not be used for multiple logins. ● The session inactivity timeout interval should be as brief as possible, taking into account the risks and business objectives (OWASP, 2010).
5. Access Control: According to their power level, users are granted the right to access resources or systems under the security approach known as access control (SecureCoding, 2021).	● Access to protected URLs, services, application data, user data, characteristics, etc. should only be granted to authorized users. ● The implementation of account audits and the deletion of unused accounts are both necessary. ● The least privilege should be granted to accounts connected to various internal or external systems that are mostly utilized for non-critical operations. ● A user or device's total number of transactions should be restricted for a specific amount of time (OWASP, 2010).
6. Cryptographic Practices: Usually, sensitive data is encrypted using cryptographic processes to ensure that only the designated users can access it and alter it, retaining confidentiality (SecureCoding, 2021).	● To ensure the privacy of sensitive data on the application, cryptographic functions should be implemented using a reliable mechanism. ● An authorized random number generator should be used when creating random numbers, file names, and strings. ● Utilizing processes and policies should be done when managing cryptographic keys. ● Unauthorized access should be prevented while using master keys (OWASP, 2010).
7. Error Handling and Logging: Procedures for handling unexpected output, which typically occurs when a program or piece of software is given an aberrant input, are referred to as error handling. One can monitor software or application changes by using logging (SecureCoding, 2021).	● Use error handlers that don't discard debugging information in the case of unexpected input. ● Memory needs to be correctly released when error situations arise. ● Logs shouldn't contain private data about systems, sessions, etc. ● Keep track of and carefully examine the logs for input validation errors, failed authentication attempts, access control, system exceptions, unusual data changes, and modifications to security parameters (OWASP, 2010).

8. Data Protection: It is the process of preventing crucial data from being altered, compromised, or lost (SecureCoding, 2021).	● Limit user rights and privileges to the systems, information, and usability necessary to accomplish the necessary goals and objectives by adhering to the principle of least privilege. ● GET requests found in HTTP should not include sensitive information. ● Protect server-side code and make sure that the average user cannot access it. ● Turn off auto-complete forms that are likely to include sensitive information, such as authentication (OWASP, 2010).
9. Communication Security: The use of encryption safeguards all sensitive data. Connections can be secured using Transport Layer Security (TLS), which can be used in conjunction with various types of encryption (SecureCoding, 2021).	● TLS connections that fail shouldn't simply rely on insecure protocols. ● Use TLS to protect sensitive information while using external sources. ● Character encoding needs to be provided for every connection (OWASP, 2010).
10. System Configuration: It is the systems engineering that specifies the many devices, processes, and computer hardware that make up the overall system and its limits (SecureCoding, 2021).	● Make sure that all system components, frameworks, and systems are running the most recent updates. ● Applying the least privilege should be done on web servers, processes, and service accounts. ● Only pertinent information should be included in HTTP response headers. Information about the OS, web server version, and software frameworks shouldn't be published. ● Environments for testing and development should be separated from those used for production
11. Database Security: Database security entails taking security precautions to resist various dangers to databases (SecureCoding, 2021).	● The program should access the database with the least amount of privilege possible. ● The default passwords must be changed right away. When choosing passwords for accounts, password management should be considered. ● Where appropriate, enable multifactor authentication. Deactivate any default accounts that are not required for business purposes (OWASP, 2010).
12. File Management: It is the process and action of putting information into a structured format for quick access (SecureCoding, 2021).	● Before uploading a file to the server, make sure it is authenticated. ● The file headers should be checked to ensure the validity of any files uploaded to the server. ● The directories where files are uploaded ought to not have execution privileges enabled. ● Never give the client an absolute file path (OWASP, 2010).

13. Memory Management: It involves using the limited resources at hand to allocate the memory needed for a program's objects and data structures and recycling the used memory for when it is no longer needed (SecureCoding, 2021).	● Avoid using vulnerable printing and copying functions like print, strcat, strcpy, etc. ● For overflows, the buffer size should be checked. ● Before using operations like copy and concatenation, input strings should be correctly shortened (OWASP, 2010).

2.3.4 Secure Software Development Life Cycle

Standard business procedures are used to develop software applications, known as the software development life cycle. The typical six to eight processes include planning, requirements, design, build, document, test, deployment, and maintenance. OWASP's (2021a) goal is to offer specific advice that is independent of any particular development technique, not to propose one. Instead, they have provided a general development model that one should use following their business's procedures. The following actions should be performed as part of this testing framework: "prior to the start of development, during definition and design, during development, during deployment, and during maintenance and operations." According to OWASP (2021a), "Businesses should review their whole SDLC to make sure security is a crucial step in the development process. To guarantee that security is effectively addressed and policies are effective throughout the development process, SDLCs should incorporate security tests."

Phase 1: Before Development Begins

1.1 Define an SDLC: Security must be built into the SDLC definition at each level to be effective.

1.2 Review Policies and Standards: Ensure the necessary standards, procedures, and documentation are in place. Documentation is crucial since it provides development teams with rules and regulations that they can attach to.

1.3 Plan the measurement program: By specifying the metrics that must be measured, it gives insight into both process and product flaws (OWASP, 2021a).

Phase 2: During Definition and Design

2.1 Review Security Requirements: Testing here refers to checking the premises supported by the requirements and determining whether the definitions of the requirements are complete.

2.2 Review Design and Architecture: As outlined in the requirements, the required level of security must be enforced through the design and architecture. Finding security issues early on in the design process might be one of the most cost-effective solutions and one of the most acceptable methods to put changes into practice.

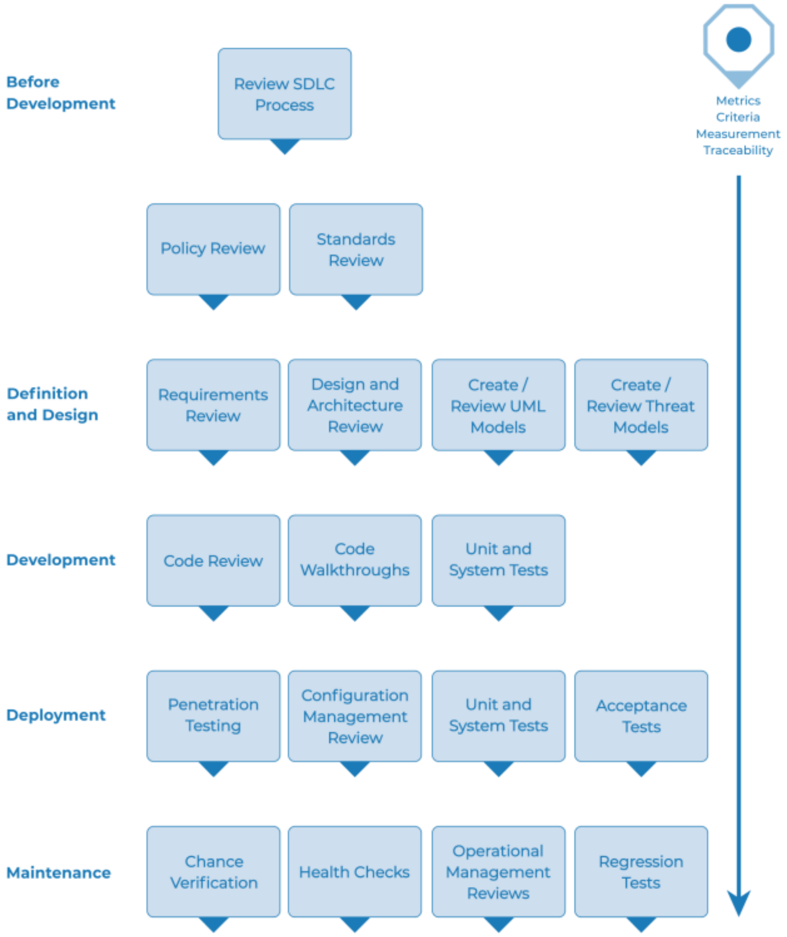


Figure 4: Software Development Life Cycle (SDLC) testing workflow (OWASP, 2021a)

2.3 Create and Review Unified Modeling Language (UML) Models: Construct application-specific UML models (to offer a standard method for visualizing system design) that detail the application's operation. If flaws are found, the system architect should be informed so that they can propose workarounds.

2.4 Create and Review Threat Models: Create plausible risk scenarios. Examine the architecture and design to ensure these risks have been considered. If there are no mitigation techniques, have the systems architect review the design and architecture and make any necessary changes (OWASP, 2021a).

Phase 3: During Development

Code Walkthrough: Developers can explain the rationale and flow of the implemented code by taking a high-level "code walkthrough" look at the code. It enables the developers to explain why specific features were created in a particular method and gives the code review team a basic grasp of the code.

Code Reviews: The tester can now review the actual code for security flaws after understanding the code's Organization and the rationale behind particular decisions (OWASP, 2021a).

Phase 4: During Deployment

Application Penetration Testing: After the application has been launched, it is subjected to penetration testing as an additional check to ensure nothing has been overlooked.

Configuration Management Testing: No matter how little, it is crucial to verify configuration settings to ensure that none are left at default settings that could be exploited (OWASP, 2021a).

Phase 5: During Maintenance and Operations

Conduct Operational Management Reviews: A procedure describing how the application and infrastructure's operational aspect must be in place.

Conduct Periodic Health Checks: The application and infrastructure should undergo monthly or quarterly health checks to ensure no new security risks have been introduced and that the degree of security is still in place.

Ensure Change Verification: Every modification must be reviewed to ensure the security level has not been compromised after it has been approved, tested in the QA environment, and deployed into the production environment (OWASP, 2021a).

3. Related Works

A literature review of the prior relevant studies is an essential technique for determining the goals of the planned study and the actual research gap. This section includes all the similar pieces of literature done by other researchers and finds the actual research gap.

State of Web Application in Nepal: (Dangol & Kautish, 2019) conducted a study in Nepal by surveying consumers about their experiences with e-commerce web applications. The findings showed that insufficient security practices hurt customers' perceptions of the vulnerabilities. Customers are, therefore, more inclined to refrain from engaging in e-commerce if they have more knowledge of or experience with cyber-fraud occurrences. In order to guarantee that services will be delivered to consumers with the highest level of security, to secure the integrity of the system, and to guarantee the privacy of online users, there are growing worries about the reliability and security of the built websites and applications in Nepal. Similarly, (Giri & Shakya, 2020) did a study to examine Nepal's significant risk of cyberattacks, crime, threats, and upcoming issues. According to the report, the risk of a cyberattack and threat has been unexpectedly rising in several sectors of Nepalese society. Hence, the author suggests that a unified plan is crucial to lowering the rising technical risk associated with cybercrime; danger and attack risks are all relatively high. (Upadhyaya, Shakya & Pokharel, 2012) helped Nepal by suggesting a cost-effective security framework. By developing a comparative and suggested framework for awareness, clarification, and examination of the security concerns involved in strengthening e-government security in technologically underdeveloped nations, this paper contributed to the literature on e-government.

Web Application Developers' Perspective: (Braz & Bacchelli, 2022) seek to comprehend developers' perspectives on evaluating software security during code review, their difficulties, and the assistance Organizations and projects offer. They performed two-part research, asking 182 practitioners about software security assessment during code review and speaking with ten professional developers. They found that developers felt inadequate training and security awareness as their biggest obstacles while looking for security flaws. Furthermore, they concluded that because developers make assumptions about the security dynamics of the applications they create, security may be overlooked during evaluations. Venson, Alfayez, Gomes, Figueiredo, & Boehm (2019) claimed that more research is necessary to fully understand the views of the various experts about security efforts in Software Development projects.

Web Application Security Testing: The study by (Rexha, Halili, Rrmoku & Imeraj, 2015) attempts to correlate and assess the effectiveness of security measures used by web developers to guard against vulnerabilities in web applications. More than 110 regional websites conducted penetration testing as part of research to figure out how much security measures were relevant during web application development. They found several flaws in these websites and connected them to the survey findings. The study by (Søhoel, Jaatun, & Boyd, 2018) examined five startups IT firms that provided services online to see how well-versed and capable they were in the OWASP top 10 web application vulnerabilities. The top 10 web application security vulnerabilities are controlled and anticipated using the OWASP Top 10 Guide for developers and security teams. These different flaws make it simple for hackers to insert malware, look for data, or completely control the website (Pandya & Patel, 2016).

(Santos & Santos, 2019) Showed that agile teams generally lacked basic security understanding and relied excessively on pen testers. The paper's main objective was to help general testers professionals understand the types of approaches, techniques, strategies, and technical knowledge that can be used to guarantee fewer security failures from the perspective of security specialists to improve testers' abilities in Web Security Testing. The author could also observe that the security team was improved by adding other experts as software testers. Moreover, Willberg (2019) tested Web Application Security with OWASP Top 10 - 2017 Framework. This study aimed to strengthen the security of the targeted web application. The author suggested that future security testing should be ongoing or at least done at each major release and suggests that security needs to be given more thought while developing software.

Developers' familiarity with Web Application Vulnerabilities: The study by (Sahin, Ünlü, Hébert, Shepherd, Coull, & Mc Lean, 2022) aimed to comprehend how familiar developers are with various web attack and defense mechanisms. (Coutinho & Pinheiro, 2021) Also did a familiarity test among 46 developers. 29 (63.04%) claimed to use OWASP practices in their software development process, while the other 17 (36.96%) developers claimed not to use such practices. In addition, 29 (63.04%) subjects have more than three years of experience in software development, and 17 (36.96%) have less than three years of experience in software development. (Søhoel, Jaatun, & Boyd, 2018) studied how familiar startup firms were with the OWASP top ten. They studied how familiar those firms were with the OWASP top ten. The finding was that none of the startups used a systematic approach to ensure security, and the measures being taken seemed somewhat arbitrary.

Web Application Security Practices: An exploratory research was carried out among Malaysian software professionals by (Mohamed, Baharom, Deraman, Yahaya, & Mohd, 2016) to examine their practices and experiences with the secure software process in actual projects. The study's results, which included 93 software professionals, showed that software practitioners are becoming increasingly aware of the value of safe software processes. However, they are not adequately putting such processes into practice. The majority of Organizations were not ready and would be adversely impacted by cyberattacks, according to research by the United States, Britain, Germany, Spain, and the Netherlands (Mardisalu, 2019).

Coutinho and Pinheiro (2021) conducted a qualitative and quantitative study to examine processes in the day-to-day activities of 46 professional software developers through a survey that assesses their level of knowledge on this topic based on the OWASP Top 10. They also aimed to learn more about the professional software developers' development process and the motivations behind using or abusing specific techniques. The findings indicated that using security procedures lengthens the development process. It was also noted that businesses significantly influence whether or not employees should engage in these behaviors. The study by (Søhoel et al., 2018) examined five startups IT firms that provide services online to see how well-versed and capable they are in the top 10 OWASP web application vulnerabilities.

The human element, the software developer, and secure code, particularly secure coding principles, were the main topics of the research by Gasiba et al. (2021). Bird & Kim (2014) performed a survey on application security programs and practices with participants of 488 firms on their training initiatives, where they found that 27% of the Organizations had no secure coding training programs, while 26% had to continue secure coding training programs that effectively reduced common application risks. They discovered that companies could not advance effectively due to a lack of knowledge and skills. The study by (Sahin, Ünlü, Hébert, Shepherd, Coull, & Mc Lean, 2022) aimed to comprehend how familiar developers are with various web attack and defense mechanisms by carrying out two distinct experiments: In order to understand the perceived attack surface and the different security controls that are frequently taken into consideration, they first used a questionnaire. Second, they created a Capture the Flag competition to encourage participants to find as many attack sites on a particular web application as possible. In the study by (Rao, 2016), numerous statutes, regulations, policies, guidelines, and standards issued by government agencies for the design, development, and deployment of web-based applications were examined for the case in India to illustrate various concerns with money, workforce, and code.

Secure Software Development Life Cycle: To investigate real-life software security procedures at each stage of the development lifecycle, the author (Assal & Chiasson, 2018) interviewed developers engaged in the business world. Maher, Shah, Chan-dio, Mohadis & Rahim (2020) conducted a qualitative study by interviewing senior professionals at Malaysian software development Organizations to determine the variables impacting developers' desire to embrace secure software development techniques. The aggregate interview findings demonstrate that most software industries have not adopted secure software development techniques at a suitable level. In order to undertake an internal audit and confirm the security of web applications against cyber-attacks before and after they are deployed onto an Organization's network, the research goal of the thesis by Narayana (2022) was to develop security auditing and testing guidelines and principles. By looking at and studying academic papers, security standards, expert community recommendations documents, and best practices, the thesis developed the Secure Software Development Life Cycle (SDLC) and ISACA's unified revolutionary security auditing architecture and principles for IIoT web applications. The findings by Narayana (2022) demonstrated that implementing the security principles checklist and the security analysis framework, including GDPR policies, authentication, and secure data transmission to audit, assisted in identifying security issues prior to each secure SDLC phase and enabled the deployment of IIoT web applications that are resistant to significant cyberattacks after being deployed to Organizations' networks. The survey performed by (Elahi, Yu, & Liu, 2011) found that among software developers, security needs are not constantly surveyed and recorded in the early phases of development but are taken into account at the implementation stage. The study (Geer, 2010) intended to identify Organizations that were not adopting software security practices and understand why. According to a survey by Errata Security, just 30.4% of the Organizations surveyed were implementing the Secure Software Development Life

Cycle, even though 81% had heard of them. When asked why they did not utilize them, 23.9% said it took too much time, 15.2% said it needed too many resources, and 4.3% said it was too expensive (Geer, 2010). The survey found that smaller Organizations used safe software development life cycles more frequently than larger ones which make sense since smaller Organizations are more flexible because there are fewer regulations and decision-making procedures to follow (Shoel et al., 2018).

3.1 Research Gap

Some literature has covered OWASP's top 10 vulnerabilities and their impact on web application security. They have tried to cover different aspects of understanding the status of web application security by covering best practices, testing approaches, the perspective of the developer, and other topics. However, there has not been enough research or investment in the web application security area in the context of Nepal. Although the consensus among experts is clear, studies demonstrate that Organizations in Nepal frequently do not apply best practices. Previous research (Dangol & Kautish, 2019; Giri & Shakya, 2020) has examined security in some Organizations in Nepal. Even though its significance has been recognized, very few studies have been done on its present application in web security. Since there was no prior research on the top 10 OWASP vulnerabilities in Nepal, this study can provide a unique and crucial purpose. In the most recent open literatures, this has not gotten much attention. Most literature focuses on qualitative or quantitative research methods, whereas this study covers both. We used the quantitative method in our initial data collection and then performed the qualitative method to understand the topic deeply. Another gap is that most research focused on technical aspects of web application security. We found a gap in the literature that focuses on non-technical aspects. We have covered both technical and non-technical aspects. Mostly, the literature covered the study of technological companies and lacked studies on companies that are not heavy in technology. To fulfill this gap, we have included diverse Organizations, including both governmental and private Organizations in Nepal.

There is nevertheless a rise in reports of these malicious attacks despite the many tools and procedures that can assist find, mitigating, or otherwise eliminating vulnerabilities in these programs (Deepa & Thilagam, 2016). Guidelines, best practices, and specialist knowledge, primarily undocumented, are crucial to application security (De Win, Scandariato, Buyens, Grégoire, & Joosen, 2009; Jose, 2020). Moreover, the OWASP community published a new Top Ten (2021) list following four years of work on the OWASP Top Ten (2017) categories. It demonstrates the emergence of new technology, process, and awareness. Additionally, new risks and gaps arise as users migrate and use new software stacks (Yergaliyev, 2022).

Furthermore, most of the studies were exploratory; ours is explanatory. Exploratory research examines the study topic but does not provide definitive or conclusive solutions to current issues. In contrast, explanatory research connects many concepts to comprehend the nature of cause-and-effect interactions in order to explain why particular events occur (Hasa, 2021).

4. Methodology

The methodology utilized to address the research issues in this thesis are described in this section. The various information-gathering methods will be covered in the following sections to accomplish this.

4.1 Research Design

The research workflow is presented in the figure below.

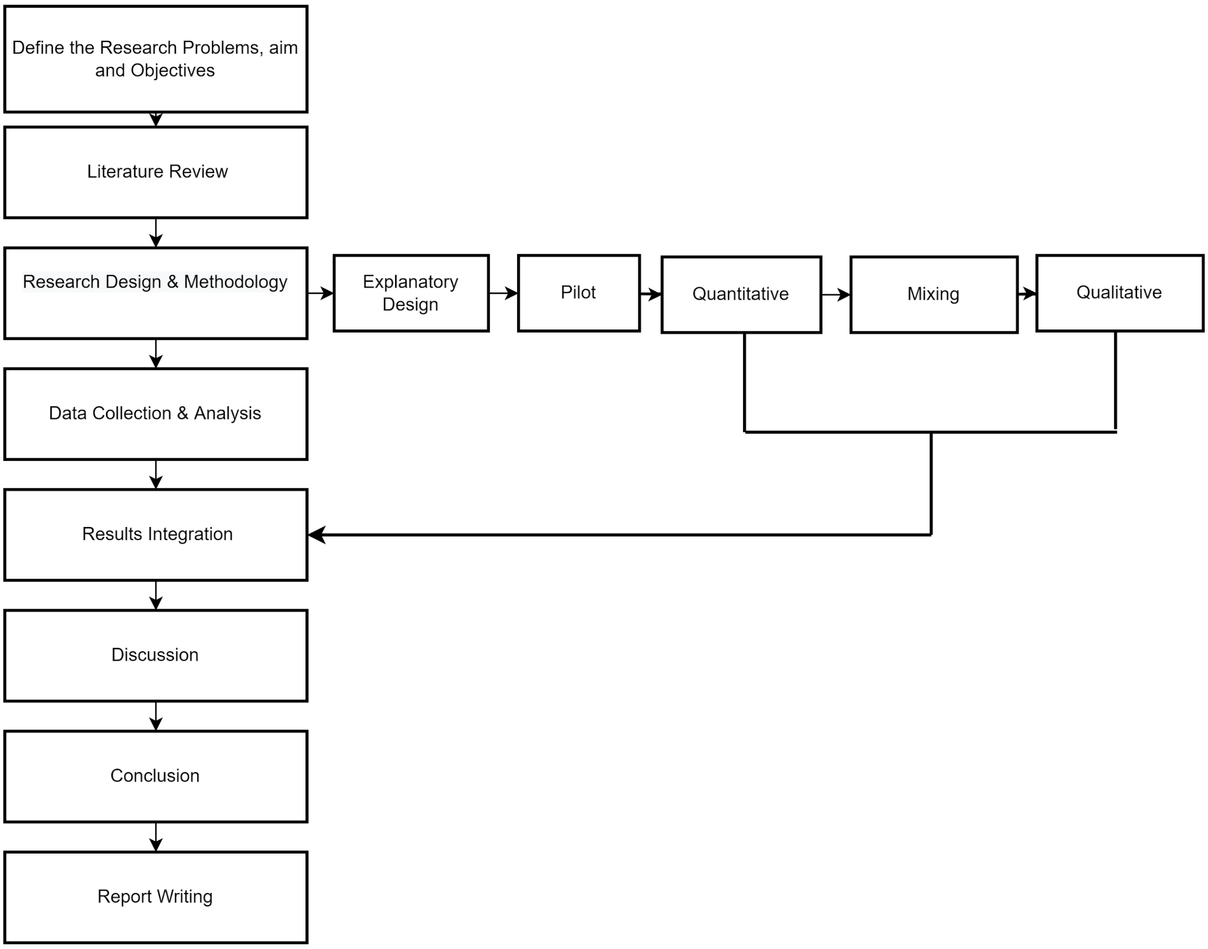


Figure 5: Research Workflow Design

As part of the procedure, the research started with a desktop study that entailed developing the research question, refining it through a literature review, and then organizing and carrying out the pilot study. The series of descriptive studies were pursued only if the pilot research findings fulfilled the minimal validity standards; otherwise, a repeat of the pilot study would be required. The research was based on a sequential explanatory design, therefore, before beginning the interview, the quantitative methodology was applied first, and the data was gathered and analyzed. It starts with gathering and analyzing quantitative data before moving on to qualitative gathering and analyzing data to understand the quantitative results in a deeper level. While the study workflow ended with the report compilation, the outcomes from both quantitative and qualitative methodologies interacted during integration and subsequent discussion. Detailed information on research design is presented in the following subsection.

4.2 Research Strategy

Research strategy is one of the elements of research methodology that sets the research direction and includes the process in which research is carried out (Magusiak, 2019). It is a subset of research design, includes data collection and interpretation elements, and emerges from the research purpose and question. Ultimately, a good research strategy answers the research questions (Malhotra, 2022). A survey is used as a research strategy here. A survey is a technique for acquiring data from a sample of people by asking pertinent questions to understand populations.

4.3 Research Approach

Research methodologies are divided into two categories: qualitative and quantitative. However, combining qualitative and quantitative research designs or techniques is feasible to increase the quality and performance of the drawn conclusions and their generalizability. This study uses the "Mixed Methodology Approach", which combines quantitative and qualitative research techniques. Both of the methods are described briefly below, followed by the explanation of the "Mixed Method Approach."

4.3.1 Quantitative Approach

Quantitative research methods are fundamentally concerned with gathering and evaluating structured data that may be represented quantitatively (Hoy & Adams, 2015). The quantitative research approach is used in our study to examine the data and produce insightful survey results. Since quantitative research concentrates on quantifiable data, it is particularly effective for our thesis to quantify the degree to which Nepalese Organizations are vulnerable by analyzing the process they are following. We have used questionnaires to collect the quantitative data. A questionnaire is a list of questions made to people to gather statistics on a particular subject. Questionnaires can be essential for making claims about particular individuals, groups, or entire populations when they are well designed and implemented. They are an effective way to gather a variety of information from a large group of people, often known as respondents (Roopa & Rani, 2012).

4.3.2 Qualitative Approach

The objective of qualitative research aims to comprehend social trends and how individuals interpret their experiences in the real world (Hancock, Ockleford, & Windridge, 2001; Cropley, 2015). Qualitative research aims to gain insight into human and interpersonal situations based on emotions and experiential components (Allwood, 2011). The methods for gathering qualitative data are unstructured, and the contexts encourage respondents to speak freely about topics that influence their lives (Mahoney 2009). The sample size limits generalization in qualitative research because analyzing and interpreting spoken data may be difficult and time-consuming (Mahoney, 2009). This form of research gathers evidence, provides findings that are not predetermined, and has implications beyond the scope of the study. For this study, semi-structured and open-ended interviewing is used since it gives the interviewer and respondent greater flexibility and response to themes identified (Jackson, Drummond & Camara, 2007).

4.3.3 Mixed Method Approach

This study uses "mixed methods" which are research methodologies that combine quantitative and qualitative approaches at different phases of the research process and result from the pragmatic paradigm (Tashakkori & Teddlie, 2008, p.22). Together, the two paradigms of methodology yield knowledge that neither quantitative analysis nor qualitative analysis can produce separately (Bell et al., 2019). Nevertheless, the two techniques can be combined to improve the quality and generalizability of the conclusions reached. As a result, research using a combination of approaches offers a more complete and in-depth understanding of the research issue.

The study's use of hybrid approaches was mainly motivated by two factors. First, there is the issue of theory development and measurement in quantitative research, which inadequate and unfinished theoretical conceptions can cause, such as a failure to recognize explanatory factors, among other things. Second, because of weak operationalization processes, qualitative research suffers from case selection and transferability problems, inhibiting the potential of qualitative research to be generalized (Alloghani, 2019). However, the study's use of both guaranteed its limitations were addressed. Moreover, with an in-depth interview, as the interviewee is physically present, there will be less "not applicable" replies and the interviewer can dig into deep information by asking follow-up questions. When necessary, interviewers can provide clarification for respondents too. In addition, interviews can be combined with surveys to provide a deeper insight of respondents' opinions (Alloghani, 2019).

Depending on the design, data might be collected using hybrid approaches concurrently or sequentially. In this case, either data type may be given priority, or both may be considered. In order to converge or corroborate findings, it enables researchers to broaden an understanding from one method to another (Creswell et al., 2012). This research method combines the depth of in-depth comprehension provided by qualitative research with the breadth of generalization provided by quantitative research (Terrell, 2012). According to Creswell (2013), the basic mixed methods designs include convergent design, explanatory sequential design, and exploratory sequential design where,

- **Convergent Design:** A Convergent design collects qualitative and quantitative data concurrently, analyzes them individually, and then merges them. As researchers, our primary goal is to determine whether our qualitative and quantitative data are meaningfully convergent.

- **Explanatory design:** Quantitative data collection and analysis are the first steps in an explanatory design. Qualitative data is then gathered and analyzed after having quantitative results. Quantitative outcomes are explained using qualitative data.

- **Exploratory Design:** In contrast, the qualitative phase of an experimental design is followed by the collection and interpretation of quantitative data. When adopting an exploratory approach, we typically test the generalizability of our qualitative conclusions using our quantitative data.

The primary focus of this design is to explain quantitative results by using qualitative data to explore specific results in more detail or help explain unexpected results (e.g., using follow-up interviews to understand the results of a quantitative study better) (Terrell, 2012).

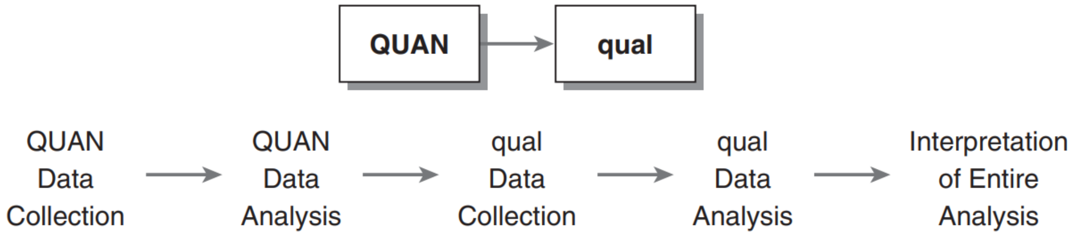


Figure 6: Sequential Explanatory Design Model based on (Creswell, Clark, Gutmann, & Hanson, 2003).

The sequential explanatory design's findings can be discussed in two stages (quantitative and then qualitative), with the final stage occurring when the results are compiled in the discussion section. Moreover, when a quantitative researcher is interested in further exploring the results, the sequential exploratory design might be helpful. However, the sequential exploratory design includes drawbacks in addition to its advantages. First, since simultaneous data collection is not supported, gathering and analyzing the two distinct data sets takes much longer. The time limit may be problematic, mainly when the two stages are equally important (Alloghani, 2019).

4.3.4 Ethical Considerations

According to Terrell (2012), a researcher should be aware of some ethical issues while adopting a mixed approach. In order to develop this thesis, the following ethical factors were taken into account.

- Participants participated voluntarily.
- Participants were aware of the study's objectives and methods.
- Participants knew that they were entitled to a copy of the outcomes.
- Participants were aware of the possible advantages of the study, and respect was shown for their privacy.
- Confidentiality will be preserved during analysis.
- The information of the study has been stated correctly in the report itself to provide readers the chance to evaluate the study's ethical standing independently.

4.4 Data Collection

Data collection is the systematic method of collecting and analyzing information on variables of interest to answer specific research questions, test theories and analyze results. Data collection is organized into two broad categories: qualitative and quantitative data collection (Kabir, 2016). Primary data collection is a method where the data is gathered firsthand by the researcher using a systematic procedure suitable to the research problem (Parveen & Showkat, 2017). Secondary data collection is a method where the data collected and compiled by someone are accessible to the public. For this study, we have used primary data from a survey and interviews with security experts from different Organizations in Nepal. Interviews were performed with prior consent from the participants. Likewise, secondary data were obtained from various articles, journal papers, books, websites, and other resources. Previous research on similar topics and studies were studied to get relevant information.

For the secondary data, the literature used for this study was taken from different platforms like Google Scholar, IEEE Explore, USN Library, and Springer. It was also essential to include blogs, frameworks, library documentation, and other grey literature to gain more profound knowledge. The generic Google search engine was also integrated to look for a literature review. However, the research, which was based on articles published as part of a thesis or dissertation, in a journal, a book, or on pertinent websites, and Peer-reviewed publications, was given priority. The OWASP Foundation and NIST publications were also employed as knowledge sources due to their practical and concrete security advice. Depending on the situation, different terms were utilized as search strings. The terms "Web Application" and "Vulnerability" were combined with one of the following keywords for the research on vulnerabilities: "OWASP," "Top 10," or "Security." The keyword "Nepal" was employed to learn about Nepal's situation. Other strings were searched for along with the words "Detection," "Prevention," and "Attacks," to identify mitigating measures.

4.4.1 Sampling Procedure

The population subset chosen for the study is referred to as a sample. It is a segment of the general public. According to Bell et al. (2019), through sampling, we can examine a subset of the larger group and draw conclusions that are likely to apply to the entire group. Probability and non-probability can be used as the basis for the selection process. This study's population comprises individuals with experience in web application development, so it was necessary to decide the sample size given the limited resources available. The sampling of such a population cannot be random since ordinary Nepalese may not use or be aware of web application development. As a result, the sampling method used is non-probability sampling. (Saunders et al., 2012) define non-probability sampling as "the case when probability of each case being selected from the total population is unknown and it is impossible to answer research questions or to address objectives that require you to make statistical inferences about the characteristics of the population." They include the most commonly used convenience/purposive, quota, snowball, etc. (Acharya, Prakash, Saxena & Nigam, 2013). In this study, convenience sampling was used. The investigator's convenience plays a role in the selection of the sample. Frequently, respondents are chosen because they are present at the appropriate time and place. As there was a time crunch, convenience sampling was advantageous because it is very efficient, simple, and cost-effective.

Security Analyst/ Administration, Security Manager/ Director, IT Manager/ IT Director, Network/ System Administration, Software Developer, QA/Tester/Test Manager, QA/Tester/Test Manager, and Risk Manager having expertise in Web Application Security from Nepal were the target group. Before being provided surveys, potential respondents were vetted for recent personal experience with web application security. The survey was voluntary and offered no monetary compensation to those who participated. The researchers believe this could have biased the selection of participants as only participants who could spare enough time or were interested in the incentive might have participated. Informants received guarantees that their privacy would be respected and that no published study would divulge their names or the identities of their Organizations. The participants for the research were approached via social media, including the messaging app WhatsApp, Facebook, E-mail, direct authors' contacts from their professional networks, and the professional networking site LinkedIn, which makes it easier to search for members by job title.

4.4.2 Quantitative Approach

Questionnaires are one of the most effective methods for data collection in survey methodology (Saunders et al., 2012). Every participant fills out an identical questionnaire, making it possible to obtain significant data samples quickly. The distribution of the survey took place at https://nettskjema.no. Due to respondents' convenience, an online questionnaire assures a better response rate than a typical paper-based survey. Moreover, this survey tool exports data straight to an excel file and SPSS to reduce errors in data transmission from data entry to data collection. Participants in the survey had further assurances of confidentiality and anonymity, and they were not forced to give their identities or those of their companies. As a result of the removal of all situational circumstances that drive individuals to respond in a socially acceptable manner, the authors believe bias should be eliminated (Donaldson & Grant-Vallone, 2002; Furnham, 1986).

For this study alone, the authors were the ones who originally designed the questionnaires. Because there was no previous work for the same purpose, this choice was taken. OWASP was used to build the questionnaire that was used in this investigation. This study examined how Nepalese Organizations perceive and practice Web application security management. The questionnaires were assessed using five-point Likert scales, which measured the answer under investigation using a five-point (1-5) ranging from strongly disagree to agree strongly. We added the "not-applicable" option to stop forcing respondents to answer questions they did not feel comfortable with. The survey layout is described below, along with how the participant answers each block of questions. Each block corresponds to at least one different page, and returning to previous pages is also allowed.

A) Starting Page: On the first page of the survey, we provide participants with information about the study. We inform participants about the data handling policy and inform them that they are allowed to drop out at any time. We ask participants questions to collect demographic information and confounding factors, such as years of professional experience (all questions are available in Appendix A). This information is mandatory to fill in as collecting helps us identify the respondent's role, company size, and Organization's primary industry represented by our respondents. Most demographic questionnaires were used, but we did not ask any personal questions to maintain the anonymity of the participants.

B) OWASP top 10 Vulnerabilities Questionnaires: The demographic questionnaires were then followed by the questions related to some general web application security practices, followed by the OWASP top 10 vulnerabilities questionnaires. At the end of the questionnaires, we asked for the participant's feedback on the survey.

4.4.2.1 Survey Questionnaires: Validity and Reliability

Following our explanatory approach, the methodology for this work is broken down into three phases (Parsons et al., 2014). The pre-testing step, also known as the validity phase, was developed to ascertain our questionnaires' internal, content, and face validity. As a part of the second phase, a pilot study was conducted to enhance and evaluate the reliability of our questionnaires. These stages supported the major investigation's execution (third phase) and provided preliminary evidence of the questionnaires' validity and reliability.

4.4.2.1.1 Phase one: Validity testing

Before the main study started, pre-testing methods were used further to evaluate the validity and reliability of the survey questionnaire. Before a respondent debriefing, a survey expert with expertise in survey design was asked to complete it. Three participants/judges who were experts in the information security field were chosen. One participant worked as a Security Analyst, another participant worked as a full stack web developer, and the other participant was also a Software Developer. These experts were questioned following the methodology outlined by DeMaio & Rothbeg (1996) concerning their comprehension of words, the clarity of instructions, and any other potential areas of confusion. The experts then participated in cognitive testing, which combines verbal questioning and think-aloud practices (Draugalis et al., 2008; Fowler, 1995). In each technique, the respondents were questioned to learn how they filled out a survey or a self-completion questionnaire. The interviewer uses the probing technique, asking specific questions or making inquiries to learn how the responder came up with the response. The think-aloud technique instructs the respondent to "think-aloud" as she or he answers the question or completes the form (Collins, 2003).

In our situation, this necessitated the experts completing the survey while researchers were present and verbalizing any thoughts that occurred as they responded (Willis, 2004). Probes were employed to gather extra data since the researchers felt that the think-aloud method had not fully detailed how the respondent understood, mentally processed, and responded to survey topics (Parsons K et al., 2014). Collins' (2003)' cognitive probes were used as a starting point. Collins (2003) used open-ended questions like, "How did you approach addressing that question? What are you thinking, please? What were you thinking about when you paused before responding? How simple or challenging was it for you to respond to this question? Why do you say that? ", as well as inquiries to test understanding (such as, "What does the term X mean to you? How did you recall that?"), retrieval (e.g., "What did you comprehend by X? "How well do you recall this?"), length of time (e.g., "Did you have a certain time in mind? "How confident are you in your response?"), reaction (e.g., "How did you feel about providing this response? Did the response choice present help you find your first response to the questions? The survey's internal validity was increased by reducing measurement errors thanks to the respondent debriefing and cognitive testing, which also helped to confirm the survey's face and content validity.

Because the expert believed several questions did not make sense for the answers we were seeking, they requested us to eliminate them. So they suggested making questions more specific. For instance, "It is possible to upload files to our web application and we look for malicious material in the submitted files." was changed to "The program loads active material from third-party servers." They also commented that the response choice presented for some questions did not help them, so we had to change and also remove some of our non-relevant questionnaires. Some questions were too long and complex to comprehend the meaning, so we made the questions simpler to understand wherever possible. For instance, "We have access to the system's log, as well as the ability to add, alter, and delete data, and have a sensitive data log handling policy." was simplified to "We have a sensitive data log handling policy."

When we asked them what they comprehended by some questions, they misunderstood what we were trying to ask as they felt the questions were vague. Depending on the above expert's answers, some questions were filtered and edited, and some were not presented to avoid asking unnecessary questions. We removed those unnecessary questions to reduce survey respondent overload and improve answer quality (Galesic & Bosnjak, 2009).

4.4.2.1.2 Phase two: Pilot Study

Snowball sampling is a sampling approach where the researcher selects a small number of subjects relevant to the research questions. These participants then suggested additional participants sharing the experience or traits the research needed. Then others will be suggested by these participants, and so on (Bell et al., 2019). In order to get input on the survey questions and projected completion time, we performed a pilot survey. Pilot participants were chosen through a snowball sampling of connections that the recruiter knew personally. Twenty respondents were invited, and thirteen of them finished the pilot survey. The survey took an average of 15 minutes to complete. Then, questionnaire responses were analyzed using the response bias category known as "content non-responsivity," which defines responses given without considering the questionnaire (Meade & Craig, 2012). This analysis revealed response patterns or instances when respondents repeatedly selected the same response, such as "not- applicable."

In our questionnaires, some statements were phrased positively, while others were phrased negatively. Participants were likely not replying with the appropriate care or attention if they gave the same uniform response to each statement. We looked for instances where participants' responses to these statements were consistent and for any indication of uniformity. Two suspected cases were found because of this. One participant, for instance, chose the same response without giving it any thought, selecting "strongly agree" to all statements. These two individuals were eliminated since it was determined that they were not being truthful with their responses. That left 11 viable answers.

The pilot study also revealed some statements that were subsequently altered. It was decided that some of these questions may be made simpler to avoid responder confusion since they were regarded to be overly complex. For instance, "Our Organization has a management system that allows us to assign different responsibilities to different user accounts in our application." was changed to "Our Organization has a user management system." Furthermore, the note regarding what "user management system" was provided in a note section to make it simpler.

As a result of the above-mentioned post-pilot study, it was possible to pinpoint frequent problems that participants had and rewrite survey items to their current structure. The findings of a pilot study were then analyzed to find any problematic items that remained and to determine the validity of the survey's core elements. The implementation of the main study was justified by the pilot study, cognitive testing, and respondent debriefing results, which offer preliminary support for the validity and reliability of the questionnaires. In this way, the results of the main study's implementation were supported by the respondent debriefing, cognitive testing, and the pilot study's preliminary evidence of the validity and reliability of the questionnaires.

4.4.3 Qualitative Approach

Qualitative research is a scientific investigation that systematically uses a specific set of methods to answer a question (Ahmad, Wasim, Irfan, Gogoi, Srivastava & Farheen, 2019).

Instead of a predetermined set and order of questions, semi-structured interviews employ an interview guide with generic groups of themes and questions (Bacchelli & Bird, 2013; Lindlof & Taylor, 2002). The study used the qualitative approach to understand the experience of security personnel of different Organizations in Nepal. According to Cohen, Manion & Morrison (2007), interviews can be used in research to follow up on unexpected results and investigate some questions raised. The research technique is effective when using interviews since they provide the researcher with more detailed information. An Organization from each sector was chosen using the convenience sampling approach. The interview may be combined with additional techniques like questionnaire measures or observation, according to Brewerton & Millward (2001). For the interview, we reached out with a few of the security specialists from the various Organizations that we surveyed. By setting up an interview with a Web Application Security specialist from Nepal with an open-ended question, the primary data for the qualitative portion of the study was gathered. After reaching out to 11 Organizations who participated in our survey, 6 Organizations agreed to send their security expert to be interviewed. See Table 6 for participants' details. The invitation included a brief explanation of the study's objectives.

A questionnaire was developed for the interview, including questions that the authors thought would be subjective and would help understand the context through the experiences of the experts. Each expert was interviewed for 60 minutes through an online media called Zoom, as we were geographically distributed. Also, it is a less costly and convenient medium to understand the security aspects of the web applications developed or used in Organizations. We surveyed participants to gather demographic data and potentially skewing characteristics like years of professional experience and highest level of education (all questions are available in Appendix B). This information is required to be provided since gathering it enables us to determine which subset of the population our response represents. We asked for the participant's feedback on the interview questionnaires. Additionally, we asked if they would be interested in receiving the study results and sharing their anonymous data in a public research dataset (Braz & Bacchelli, 2022). To prevent asking pointless questions at a subsequent interview, some questions were filtered and not provided based on participant input.

The main goal is to hear from them about what they believe to be significant regarding the current issue in their own words. The author obtained consent to record the interview based on the assumption to avoid losing any information on what they should say. Additionally, it was said that the audio would be transcripted and deleted after completion (Santos & Santos, 2019). Informants received guarantees that their privacy would be respected and that no published study would divulge their names or the identities of their Organizations. So the name is not displayed in the thesis as agreed with the terms and conditions of the interviewee. Additionally, it was stated that they were free to answer any question based on their preferences, technical expertise, or experience. The interview was handled so that the questions were less scripted and naturally present. The interview method followed (Gill, Stewart, Treasure & Chadwick, 2008)'s recommendation that interaction with the interviewee be kept to a minimum while paying close attention to what was being said to allow participants to explain their experiences as thoroughly as possible without interruptions. After every interview, the authors thanked them for taking the time to contribute to this thesis. Also, the authors asked if they had any further information they would like to share, which frequently resulted in discovering previously unknown facts (Gill et. al, 2008). The authors organized their ideas into a mind map using the Mindmup tools. A mindmup is a diagram that is used to graphically organize information with descriptive details, supports in seeing the "bigger picture," and helps to build significant connections (Bhattacharya & Mohalik, 2020).

The researcher asked follow-up questions to clarify or obtain further information related to the research question. After the sixth interview, the same concepts from many participants began to reoccur, signalling a theoretical saturation point. Six interviews were deemed adequate for this round of data collection because of this.

5. Data Analysis

In this section, we summarize the results of our investigation by analyzing our data.

5.1 Quantitative Data Analysis

Appendix A contains the survey questionnaire. Survey questionnaires were used to collect information for this study. The 87 complete and accurate replies from security working in Nepal's Organizations obtained for the survey formed the basis for the results. To address the quantitative research questions, we used a variety of techniques. We used descriptive statistics to answer the research questions and to produce tables and charts. The responses collected from the distribution of questionnaires via nettskjema were analyzed using IBM SPSS V28.0.0.0 (190). We began by presenting descriptive data that outlined the participant's demographics. After that, we analyzed the data concerning our study topics.

5.1.1 Descriptive Statistics of Demographic Sample

The descriptive statistics regarding who answered the questionnaire are contained in this section.

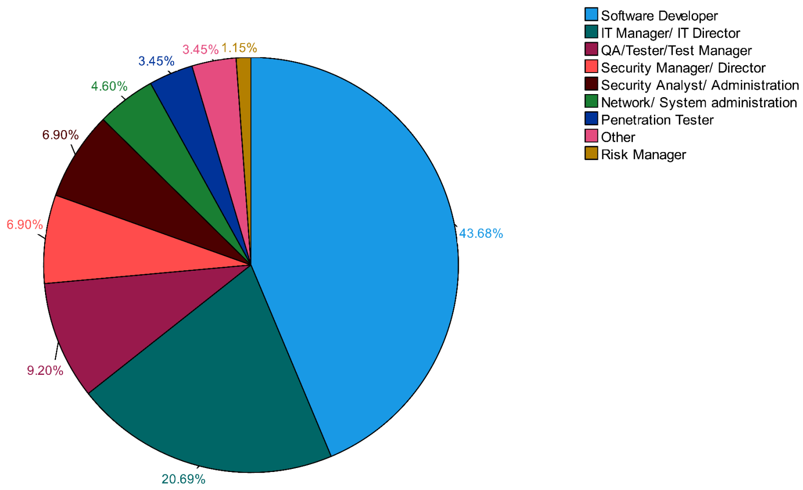


Figure 7: Pie Chart Representing the Primary Role of Respondents

The majority (43.68%) of survey respondents were software developers. The IT Manager/IT Director placed second with 20.69%. Similarly, QA Tester/Test Manager ranked third with

9.20%. The risk manager's percentage of respondents who answered the questionnaire was the lowest.

Table 3: Primary Roles of Respondents in Organization

Roles	Frequency	Percent	Valid Percent	Cumulative Percent
Software Developer	38	43.68	43.7	43.7
IT Manager/ IT Director	18	20.69	20.7	64.4
QA/Tester/Test Manager	8	9.20	9.2	73.6
Network/System Administration	6	6.90	6.9	80.5
Other	6	6.90	6.9	87.4
Security Manager/ Director	4	4.60	4.6	92.0
Security Analyst/ Administration	3	3.45	3.4	95.4
Penetration Tester	3	3.45	3.4	98.9
Risk Manager	1	1.15	1.1	100.0
Total	**87**	**100.0**	**100.0**	

The first and second highest response frequencies are thirty eight for software developers and eighteen for IT managers/IT directors, respectively. Security Managers/Directors made up four respondents, while Penetration Testers and Security Analyst/Administration each gave three. With regards to the fewest amount of responses, the Risk Manager role is filled with just one.

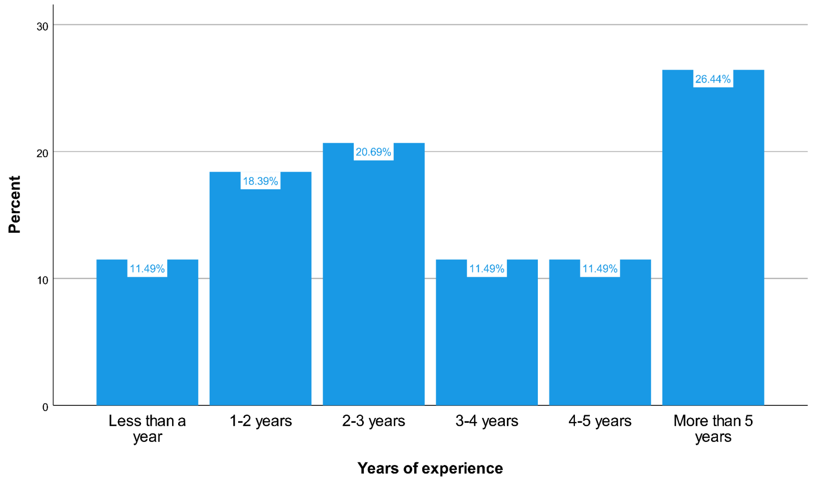


Figure 8: Bar Chart representing the number of years of work experience

With 26.44% of the total respondents, the highest proportion had more than five years of work experience. With 20.69%, those with 2-3 years of experience came in second, followed closely by those with 1-2 years of experience (18.39%). 11.49% of all respondents were at the same level as the others.

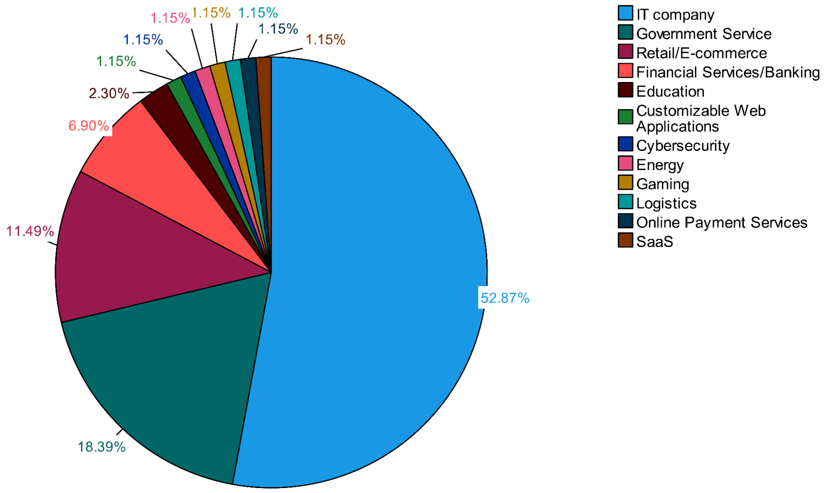


Figure 9: Pie Chart Representing Organization's primary industry

A little over half of the respondents (52.87% of the total) indicated IT companies as their Organization's primary industry. Additionally, it can be observed from the data that IT professionals are more likely than other people to answer the survey. With 18.39% of the total respondents, Government Service came in second. With 11.49% of respondents, retail/e-commerce came in third place.

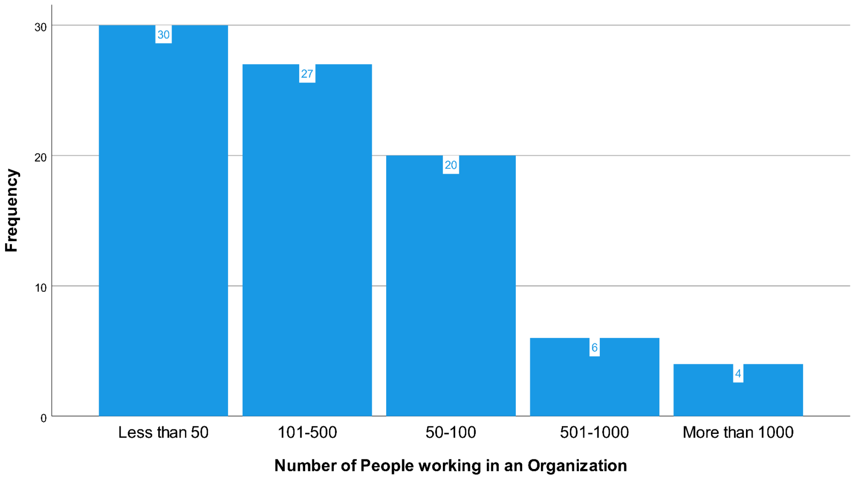


Figure 10: Bar Chart representing the number of people working in an Organization, either as employees or consultants

Thirty of the total respondents claimed they worked for a business with fewer than fifty employees. Second place went to twenty-seven respondents with between 101 and 500 workers. Companies with more than 1000 employees included four respondents.

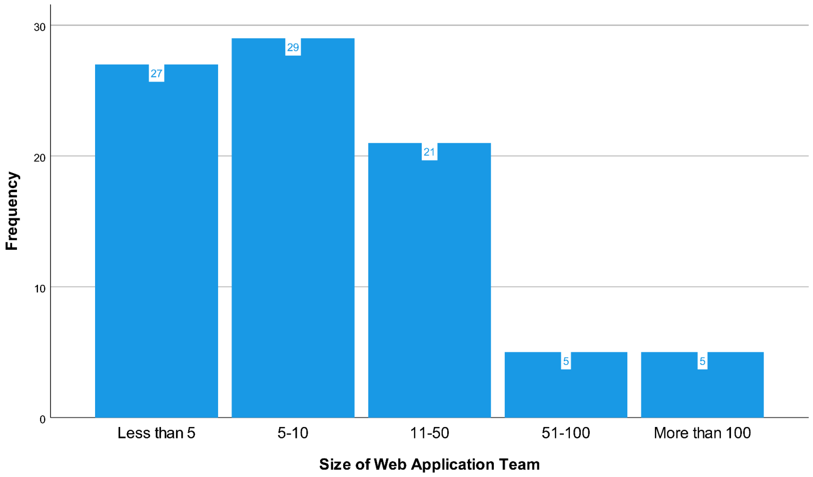


Figure 11: Size of respondents' web application development team in the Organization

The web application development team of 29 respondents consisted of 5–10 individuals. The second largest respondents came from companies whose web application development teams had fewer than five members. Five respondents for the employee categories of 51–100 and 100+ on their web application development team.

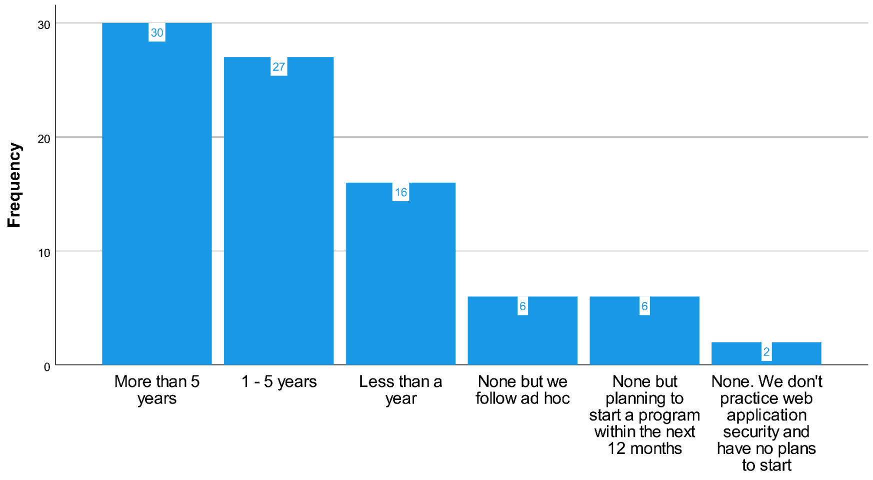


Figure 12: Bar Chart representing the number of years respondents' Organizations have been practicing web application security

Thirty respondents' Organizations, or the majority, had been using web application security. Twenty-seven respondents worked for companies using web application security for 1 to 5 years. Fourteen respondents either had no plans to start or were following web application security ad hoc but had not yet started applying it demonstrates that not all businesses actively commit time and resources to web application security.

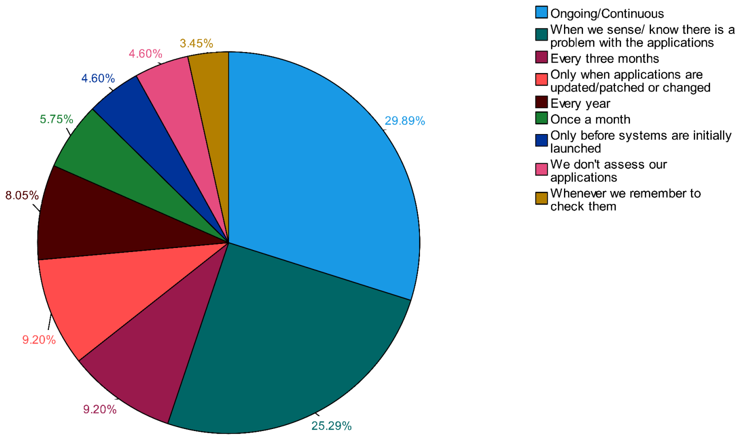


Figure 13: Pie chart showing how frequently participants' Organizations assess the security of their web applications

29.89% of those surveyed claimed to regularly or continuously evaluate web application security. 25.29% of the total respondents who stated that they only evaluate it when they detect or are aware of a problem came in second. 9.2% of respondents indicated they evaluate it every three months. 4.6% of respondents admitted they do not evaluate the security of their web applications.

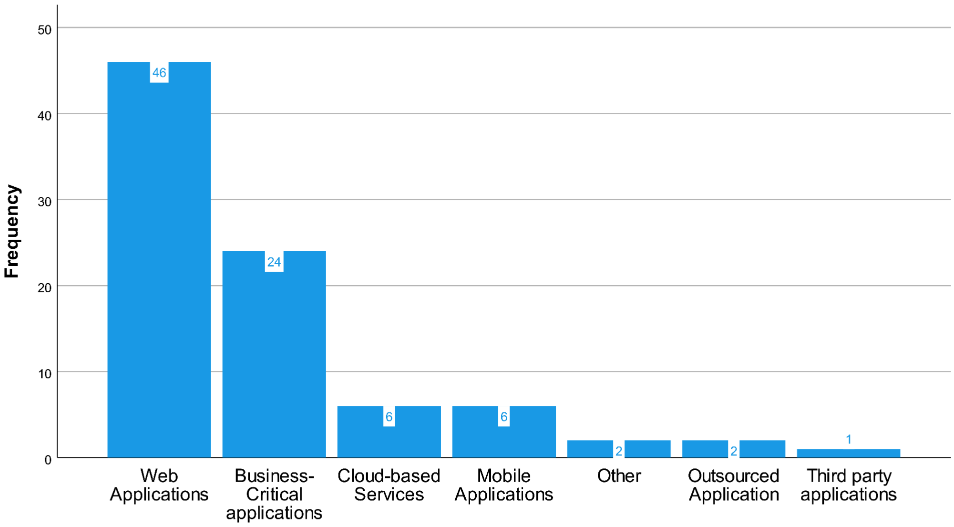


Figure 14: Bar Chart representing Respondents' Organization Spending

Given that 46 respondents chose web applications, it is evident that most Organizations focus the majority of their software development resources on them. In contrast, 24 respondents claimed that business-critical programs account for most of their resource usage. Only one person claimed that third-party applications consume the majority of their resources.

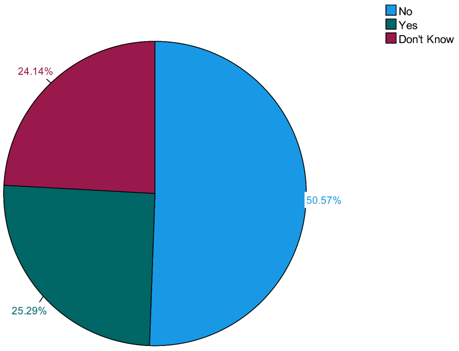


Figure 15: Pie Chart representing the proportion of Organizations where outside researchers can report security flaws in their Web Applications

The majority of respondents (50.57%) claimed that no outside researchers were able to report security flaws in their web applications. This analysis demonstrates that businesses are hesitant to request that researchers notify them of web application security issues. According to 25.29% of respondents, they welcome reports of security weaknesses from outside experts. 24.14% of respondents admitted they are unsure if their company permits this.

5.1.2 Descriptive Analysis of the Survey Answers

These questionnaires aim to investigate respondents' perceptions of security in web application development in the context of Nepal and to highlight current trends and practices. The options are rated using the following scale: Not Applicable, Strongly Agree, Agree, Neutral, Disagree, Strongly Disagree.

General Web Application Security Practices

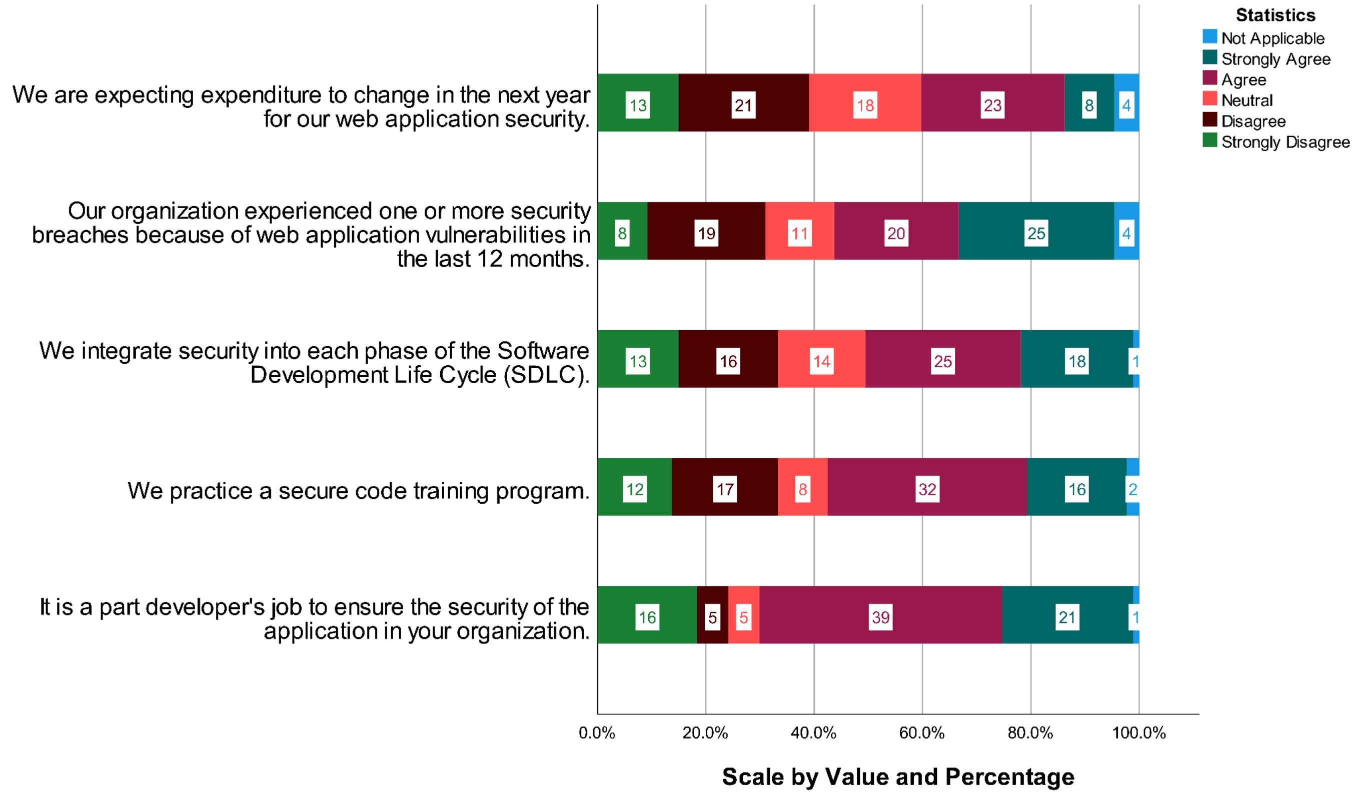


Figure 16: Analysis of General Practices

According to 45 respondents (51.7%), despite experiencing web attacks in the previous 12 months, only 34 respondents (39%) believe their budgets will change over the next few years. The majority of developers, 60 respondents (68.9% of the total), hold the view that it is their responsibility to maintain the security of web applications; however, 21 respondents (24.1%) disagree with this view. There are still 29 respondents (33.3%) who do not have a secure code training program, even though 48 respondents (55.4%) agreed they should. Similarly, only 33.3% of the 43 participating Organizations (or Organizations) included security throughout the whole software development life cycle (SDLC).

Participants' familiarity with OWASP top 10 Web Application Vulnerabilities

The options are rated using the following scale: Not Applicable, Strongly Agree, Agree, Neutral, Disagree, Strongly Disagree. The Likert scale's scoring range for the survey is shown in the table below:

Table 4: Scoring range of Likert scale of the survey

Level	Scale	Interval Length	Lower Limit	Upper Limit	Interval
Not Applicable	0	0	0	0	0
Strongly Agree	1	1	1.80	0.80	[1 : 1.80)
Agree	2	1.80	2.60	0.80	[1.80 : 2.60)
Neutral	3	2.60	3.40	0.80	[2.60 : 3.40)
Disagree	4	3.40	4.20	0.80	[3.40 : 4.20)
Strongly Disagree	5	4.20	5.00	0.80	[4.20 : 5]

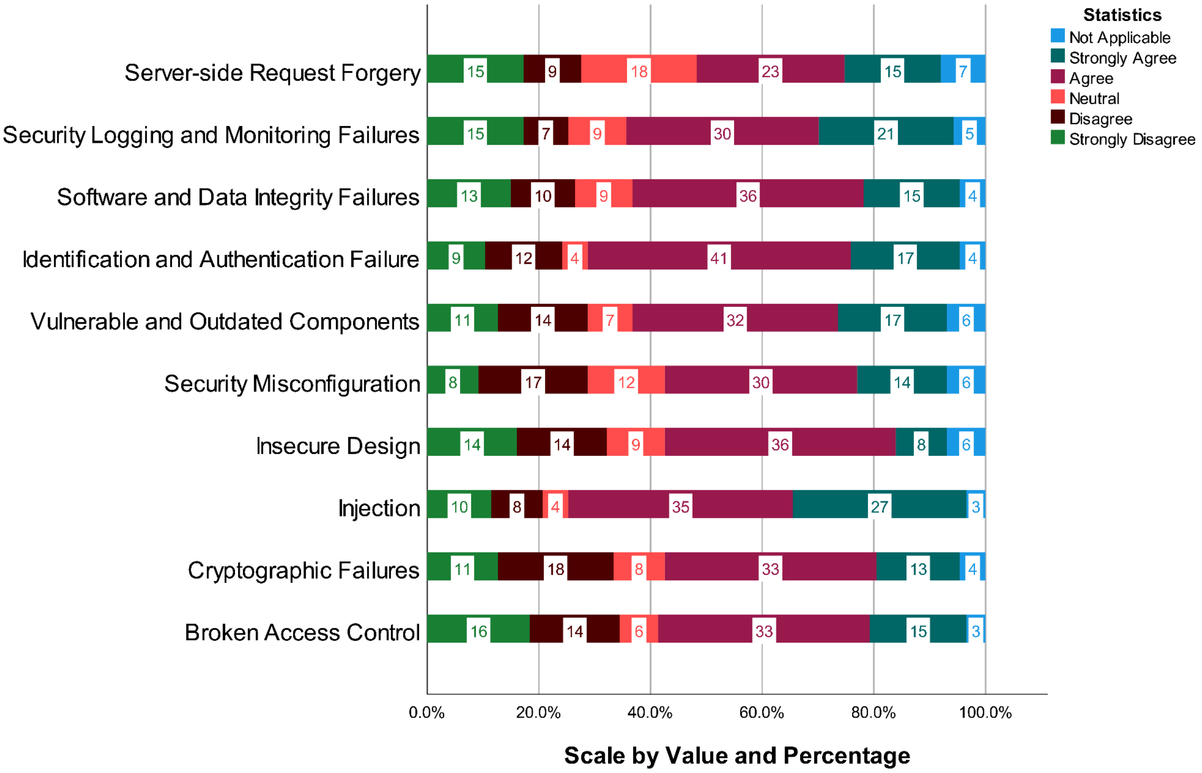


Figure 17: Participants' familiarity with OWASP top 10 Vulnerabilities

Broken Access Control Vulnerability: By agreeing or strongly agreeing, 48 respondents (55.1%) indicated that they are at least somewhat familiar with the Broken Access control vulnerability, as opposed to 30 respondents (34.5%) who disagreed. The scores were evenly distributed, with a standard deviation of 1.487 and an overall average score (mean) of 2.7 which lies in the neutral range of likert scale.

Cryptographic Failures Vulnerability: Based on our observation, we can conclude that, of the total respondents, 46 respondents (52.8%) were aware of the overall concept of cryptographic failures vulnerability. Similarly, 29 respondents (33.3%) did not know cryptography weaknesses. They scored, on average, 2.64 indicating average response falls under Neutral range of likert scale, with a standard deviation of 1.414.

Injection Vulnerability: 72.2% of the 62 respondents claimed they knew the risk injection vulnerabilities available to their web applications. Compared to the other OWASP top 10 vulnerabilities, the Injection Vulnerability has the most significant percentage. Eighteen respondents (20.7%) said they were unaware of this issue. The mean score is 2.2 indicating average response falls under Agree range of likert scale. The low mean of these vulnerabilities suggests that fewer respondents firmly agreed that they were aware of them while having the highest percentage of awareness compared to other weaknesses. The standard deviation of the mean was 1.37 , meaning that the scores were not deviating way too much.

Insecure Design Vulnerability: 44 respondents, or 50.6% of those questioned, claimed they were aware of the web application security flaw, Insecure Design. A total of 28 respondents, or 32.2%, said they are unaware of this type of vulnerability, showing that web application developers are unfamiliar with this type of issue. The average score for Insecure Design was 2.68 indicating average response falls under Agree range of likert scale, and the standard deviation was 1.459.

Security Misconfiguration vulnerability: Another vulnerability that 44 respondents (50.6%) were familiar with was the security misconfiguration issue. Regarding the aspects of this vulnerability that we attempted to address in our questionnaire, 25 respondents (28.7%) indicated that they are unaware of it and have not taken any steps to reduce any potential risks. The mean is 2.51 indicating average response falls under Agree range of likert scale, and the standard deviation is 1.397.

Vulnerable and Outdated Components Vulnerability: According to the findings, 49 respondents (56.3%) were aware of the web application vulnerability known as Vulnerable and Outdated Components. In contrast, 25 respondents (28.7%) claimed they lacked web application security safeguards for this vulnerability. They deny that they or the Organization have considered this vulnerability while developing web applications. The remaining respondents claimed that they were unsure whether their Organization took the risk into account while developing web applications. The standard deviation is 1.469, while the mean score is 2.45 indicating average response falls under Agree range of likert scale.

Identification And Authentication Failure Vulnerability: Even though 58 survey respondents (66.6%) said they are somewhat or very familiar with Identification And Authentication Failure Vulnerability and have procedures and policies in place to make their web applications secure with regards to it, 21 respondents (24.1%) do not agree that they are aware of this kind of vulnerability. Their mean score is 2.34 lindicating average response falls under Agree range of likert scale, and the standard deviation is 1.345.

Software And Data Integrity Failures Vulnerability: More than 23 respondents (26%) said that they were not familiar with one of OWASP's Top 10 vulnerabilities (Software and Data Integrity Failures vulnerability), whereas 51 respondents (58.6%) said that they either agree or strongly agree with the fact that they implement web application security to mitigate this vulnerability. The mean of all the responses is 2.54 indicating average response falls under Agree range of likert scale, and the standard deviation of the mean is 1.421.

Security Logging And Monitoring Failures Vulnerability: In response to the question about if their team or Organization is familiar with Security Logging and Monitoring Failures as a vulnerability in the context of their web applications' security, 51 respondents (58.6%) agreed. 22 respondents (25.2%) indicated they either disagree or strongly disagree that they are unfamiliar with the vulnerability. Their standard deviation is 1.515, and their mean score is 2.43 indicating average response falls under Agree range of likert scale.

Server-Side Request Forgery Vulnerability: A significant portion of the respondents, 38 respondents (31% of the total), seem to be not familiar with the Server-Side Request Forgery (SSRF) vulnerability. They disagree that their Organization had been implementing any measures to tackle this kind of vulnerability on their web application security. They had the lowest percentage of respondents who said they were familiar with the vulnerability, at 49.4% (24 in number). The overall average score is 2.60 indicating average response falls under Neutral range of likert scale, with a standard deviation of 1.528.

As per the overall familiarity with the OWASP top 10 web application security vulnerability, the mean score is 2.50 (Agree) out of 5, which is half the total score indicating average response falls under Agree range of likert scale. The standard deviation of the mean is 1.216, which shows that the individual responses, on average, were a little over 1 point away from the mean. It shows that many respondents were familiar with OWASP top 10 vulnerabilities.

Table 5: Overall Participants' familiarity with OWASP top 10

N	**Valid**	87
	Missing	0
Mean		2.5057
Std. Deviation		1.21659

Analysis of OWASP top 10 Vulnerabilities in Nepals' Organizations

1. BROKEN ACCESS CONTROL

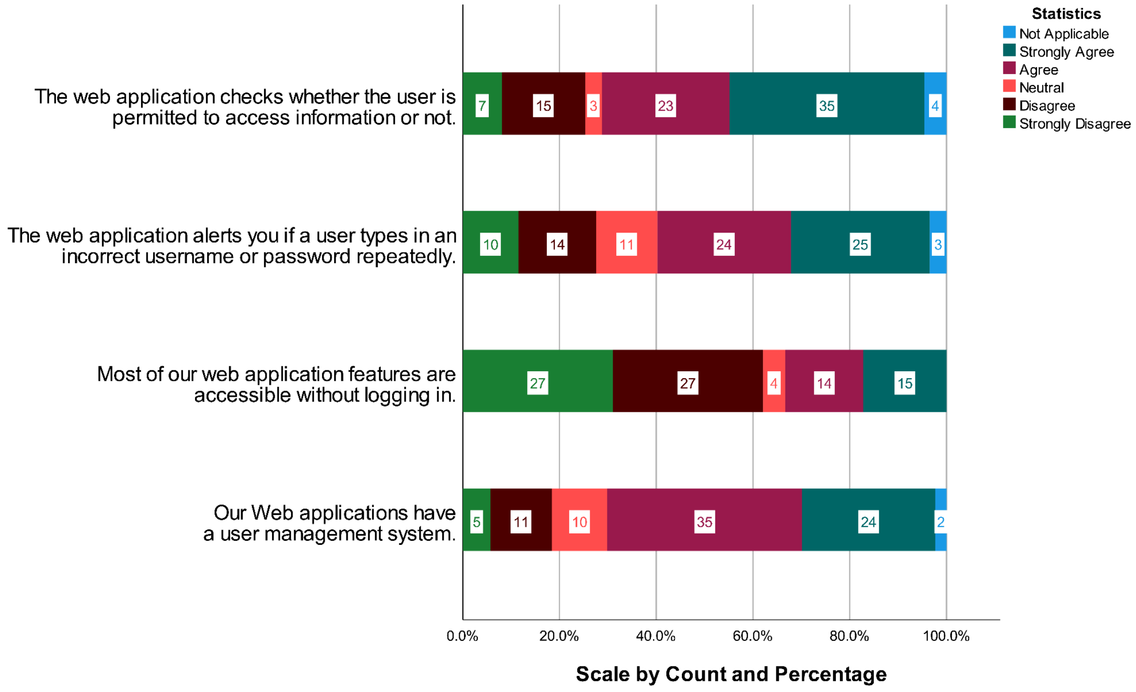


Figure 18: Analysis of Broken Access Control Vulnerability

Fifty-nine respondents (67.8%) said user management systems allow them to offer various user accounts with varying roles in the web applications. The lack of a user management system was present in 16 respondents (18.3%). The majority of respondents, 54, or about 62%, disagreed that most features of their web applications could be accessed without signing in. Most of them agreed that when a user repeatedly enters an incorrect username or password, their web applications notify the user. Twenty-four respondents (27.6%) claimed that their web applications did not provide this feature. Therefore, the analysis indicates that not all Organizations follow secure practices (OWASP, 2021) to prevent Broken Access Control.

2. CRYPTOGRAPHIC FAILURES

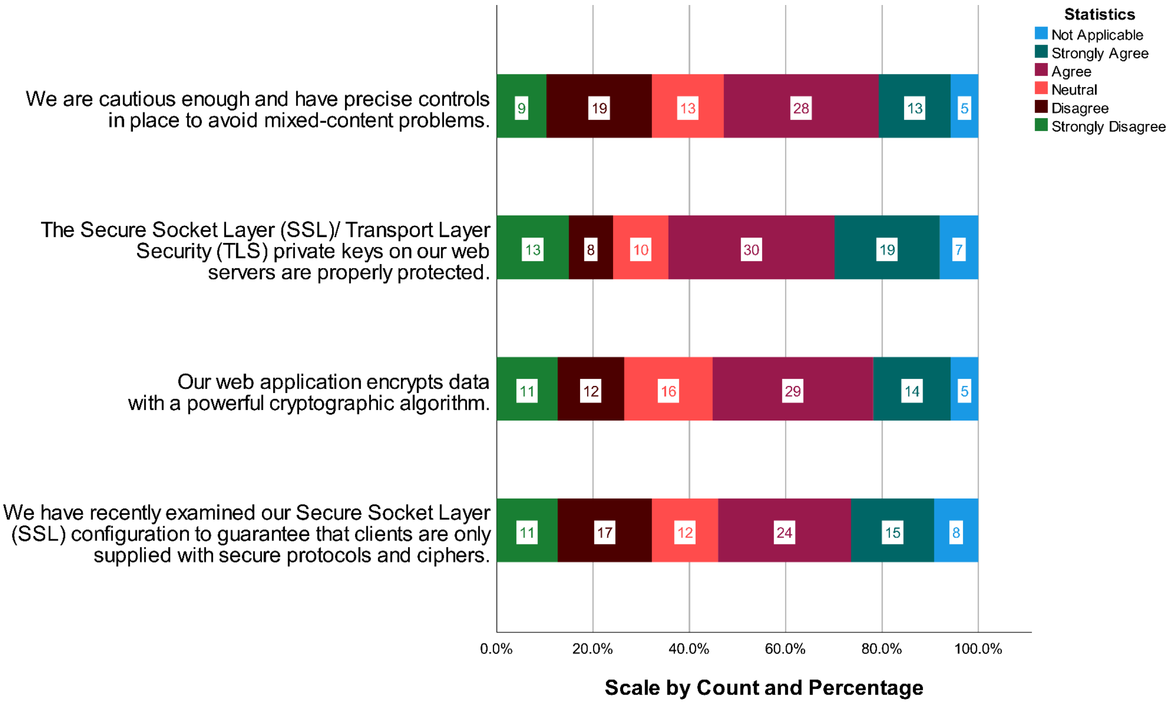


Figure 19: Analysis of Cryptographic Failures Vulnerability

The majority of 39 respondents (44.8%) concurred that they recently looked at their Secure Socket Layer (SSL) configuration to ensure that customers are only given secure protocols and ciphers. Twenty-eight respondents (32.1%) of them disagreed with this assertion, nevertheless. About 43 respondents (49.4%) were confident that their web application used a powerful cryptographic technique to encrypt data. Twenty-three respondents (26.4%), however, disagreed. Twenty-one respondents (24.1%) disagreed with the statement that their Secure Socket Layer (SSL) / Transport Layer Security (TLS) private keys are adequately safeguarded on their web servers. In comparison, 49 respondents (56.3%) agreed with it. Additionally, approximately half of them did not exercise sufficient caution or put in place specific controls to prevent issues with mixed-content problems. Consequently, the analysis indicates that not all Organizations follow secure practices (OWASP, 2021) to prevent Cryptographic Failures Vulnerability.

3. INJECTION

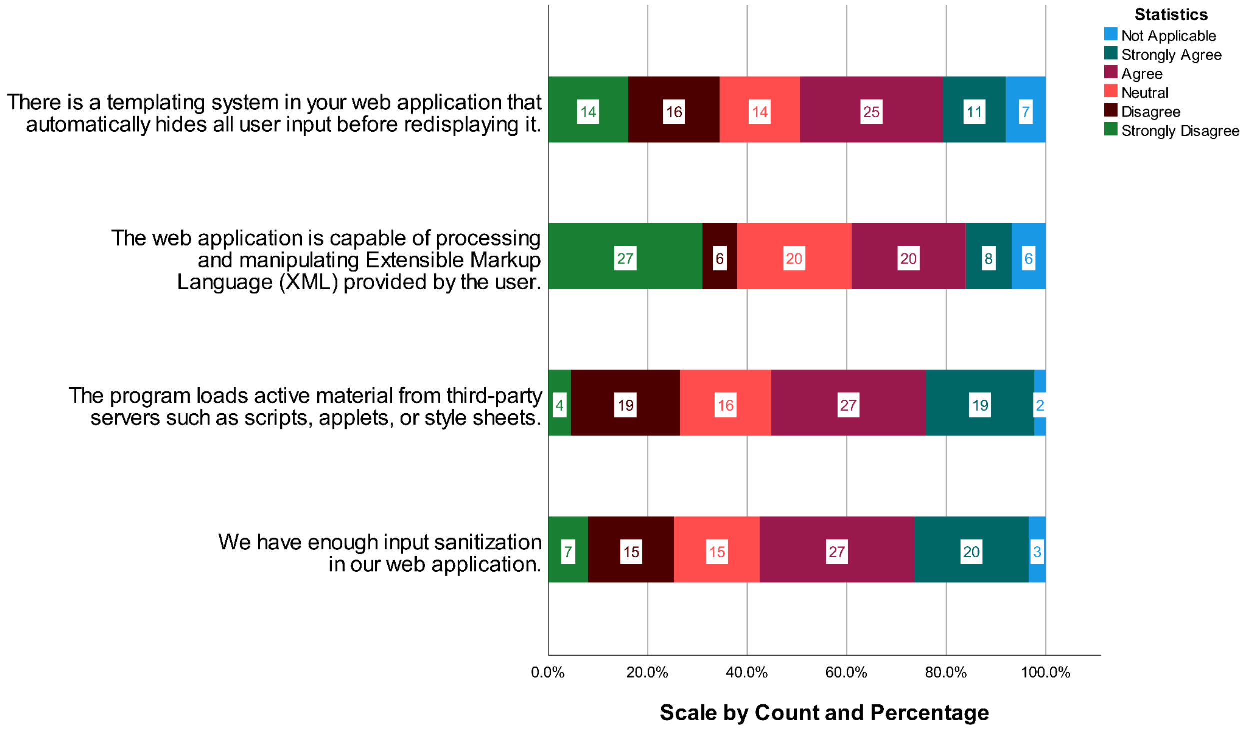


Figure 20: Analysis of Injection Vulnerability

Forty-seven respondents (54%) agreed that their web application has enough input sanitization, compared to 22 respondents (25.2%) who disagreed. Forty-six respondents (52.8%) stated that the software loads dynamic content, such as scripts, applets, or style sheets, from third-party servers (i.e., any servers not directly under their control). Most of them lacked confidence in their web application's capacity to handle and manipulate user-provided Extensible Markup Language (XML). Using a templating system to escape user input is a great strategy to prevent XSS. However, thirty respondents (34.5%) said their web application automatically hides all user input before redisplaying it. Consequently, the analysis indicates that not all Organizations follow secure practices (OWASP, 2021) to prevent Injection Vulnerability.

4. INSECURE DESIGN

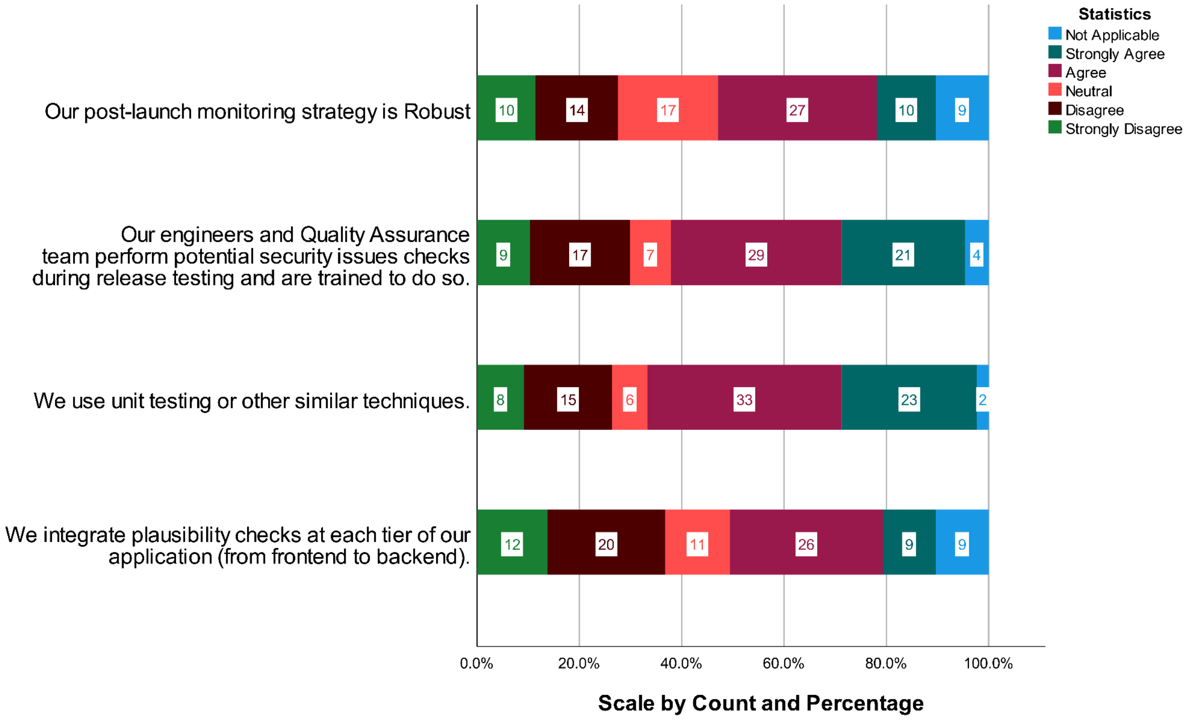


Figure 21: Analysis of Insecure Design Vulnerability

Thirty-five respondents (40.2%) at each web application level implemented plausibility checks (from frontend to backend). However, a sizable portion of responders, thirty-two respondents (36.8%), disagreed with this claim. Unit testing and other comparable approaches are used by fifty-six respondents (57.4%), while just twenty-three respondents (26.4%) did not utilize any of them. Even though fifty respondents (57.4%) said, their engineers and quality assurance team undertake possible security problem checks during release testing, twenty-six respondents (29.8%) disagreed and said they are not equipped to do so. Moreover, twenty-four respondents (27.6%) of those surveyed think their post-launch monitoring approach is weak. Thus, the analysis indicates that not all Organizations follow secure practices to prevent Insecure Design Vulnerability (OWASP, 2021).

5. SECURITY MISCONFIGURATION

While forty-eight respondents (55.1%) said they had disabled unused ports, services, sites, accounts, or privileges, twenty-three respondents (26.4%) still had not done so. Regarding having an automated method to check the efficacy of the configurations and settings in all contexts, thirty-six respondents (41.3%) disagreed. The majority of respondents, 40 respondents (46%), agreed that their default accounts and passwords were still active and unaltered. Additionally, 31 respondents (35.6%) concurred that updated systems lack the most recent security features and are not set securely. Thus, the analysis indicates that not all Organizations follow secure practices to prevent Security Misconfiguration Vulnerability (OWASP, 2021).

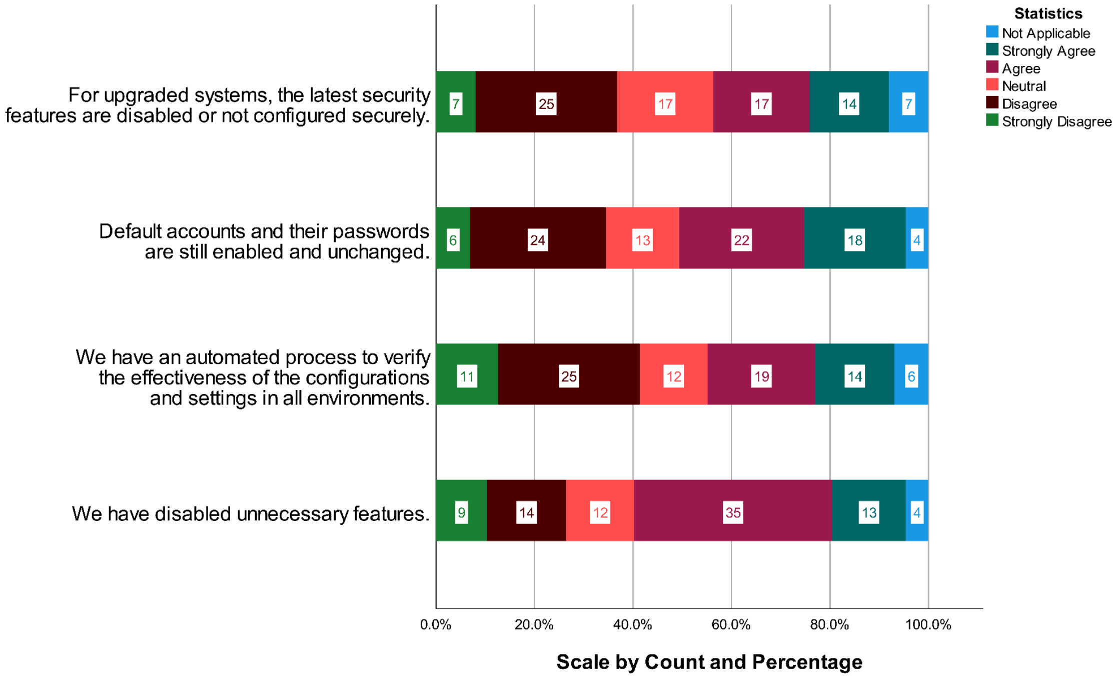


Figure 22: Analysis of Security Misconfiguration Vulnerability

6. VULNERABLE AND OUTDATED COMPONENTS

A sizable portion of respondents, thirty-one respondents (about 35.6%), disagreed, even though forty-six respondents (52.8%) of those surveyed said they knew the versions of all the client- and server-side software they use. Thirty-seven respondents (34.5%) of those surveyed disagreed that they constantly check for vulnerabilities and subscribe to security bulletins for their components. Moreover, thirty-three respondents (39.1%) disagreed that the administrator of their firm deletes old and backup data before launching a web application. Thus, the analysis exhibits that not all Organizations follow secure practices to prevent Vulnerable and Outdated Components Vulnerability (OWASP, 2021).

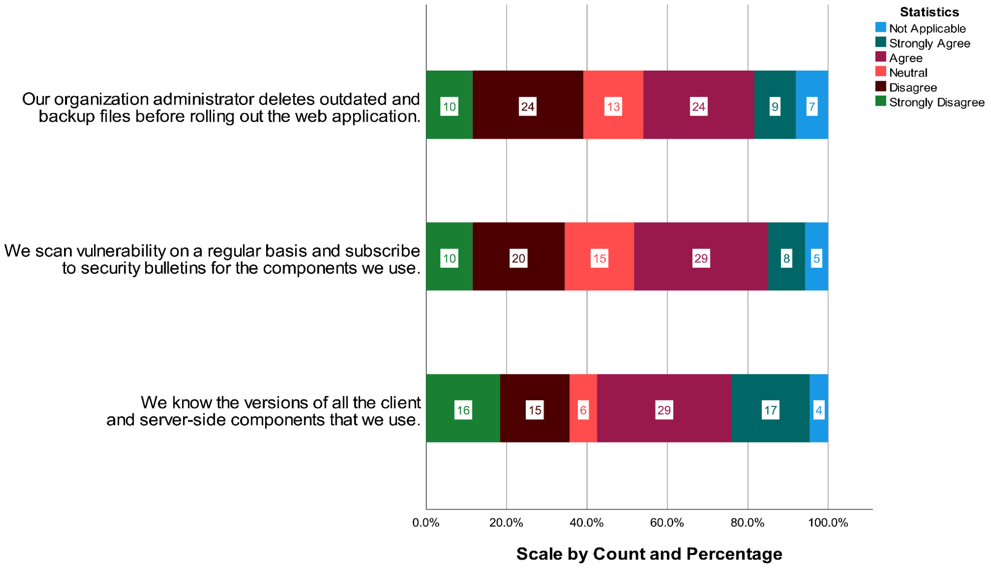


Figure 23: Analysis of Vulnerable and Outdated Components Vulnerability

7. IDENTIFICATION AND AUTHENTICATION FAILURE

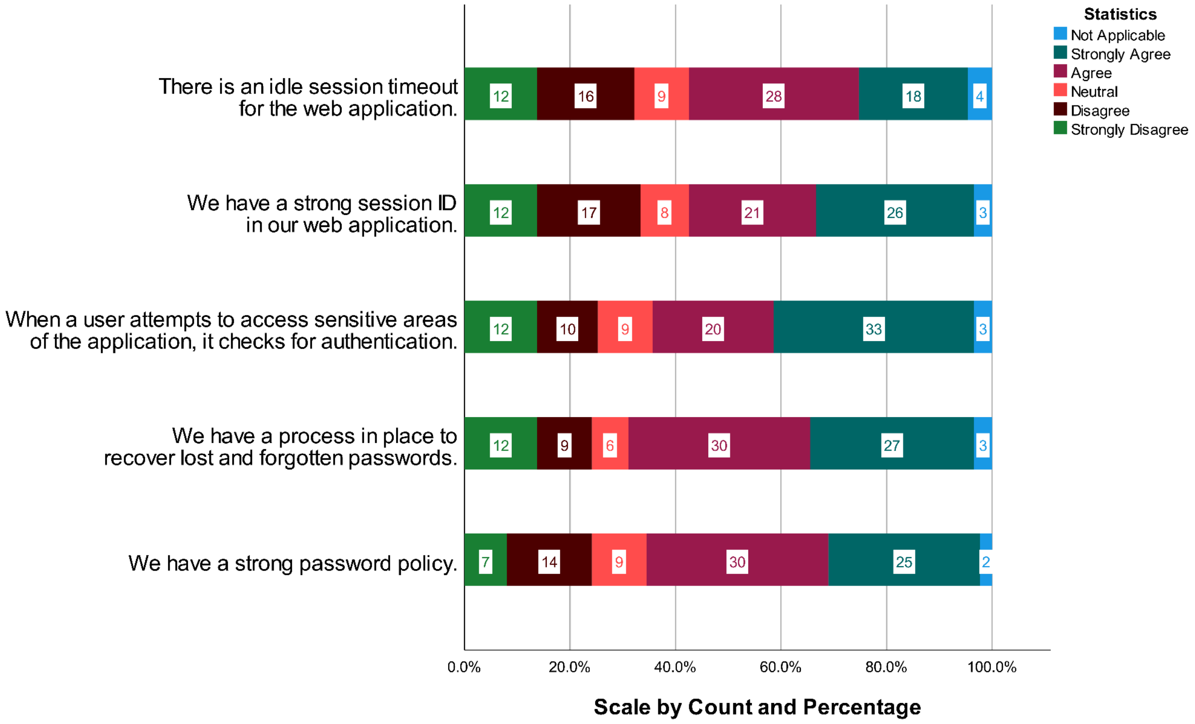


Figure 24: Analysis of Identification and Authentication Failure Vulnerability

The majority of 55 respondents (63.2%) concur that they have a solid password policy. However, 21 respondents (24.1%) are still uncertain. A mechanism for recovering lost or forgotten passwords are in place, according to 57 respondents (65.5%). In addition, 53 respondents (60.9%) agreed that their web application checks for authentication whenever a user tries to access critical portions of the program, whereas 22 respondents (25.3%) disagreed. A quarter of 28 respondents (32%) continue to contradict the existence of an idle session timeout and the existence of a strong session ID in their web application. Therefore, the analysis conveys that not all Organizations follow secure practices (OWASP, 2021) to prevent Identification and Authentication Failure Vulnerability.

8. SOFTWARE AND DATA INTEGRITY FAILURES VULNERABILITY

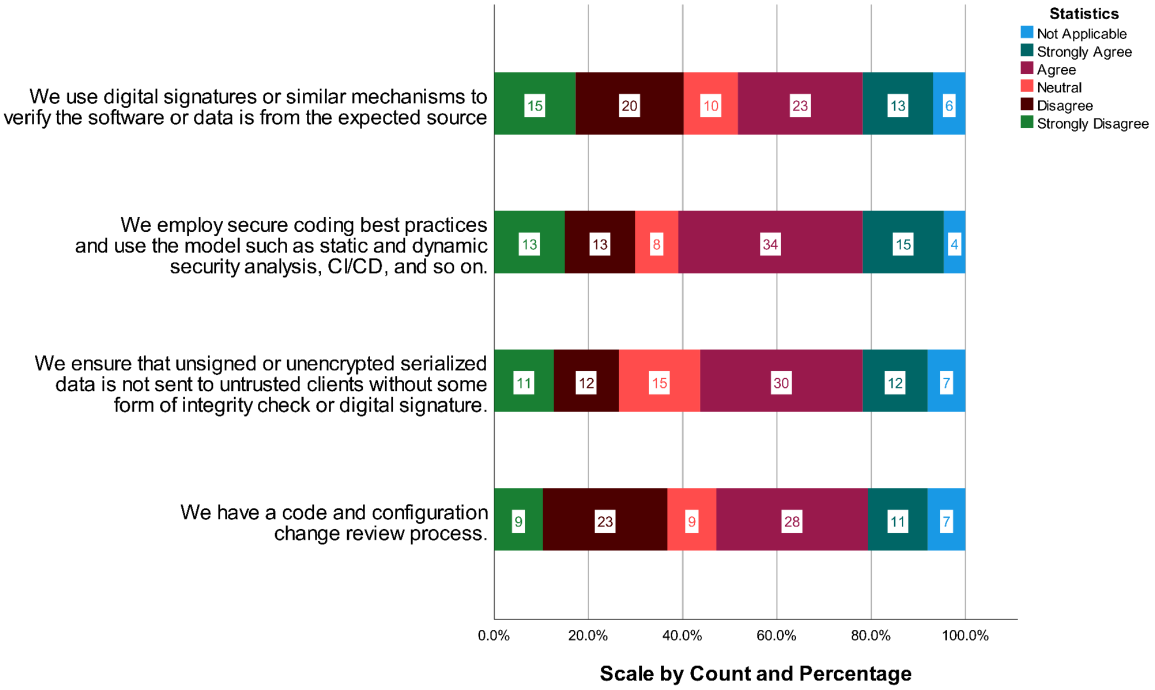


Figure 25: Analysis of Software and Data Integrity Failures Vulnerability

The majority of respondents, thirty-nine (44.8%), agreed that they have a mechanism in place for reviewing changes to code and configuration, while almost thirty-two (36.8%) disagreed. Unsigned or unencrypted serialized data is not delivered to untrusted clients without any integrity check or digital signature, according to forty-two respondents (48.3%), while twenty-three respondents (26.4%) disagreed. Compared to twenty-six respondents (29.8%), more than half - forty-nine respondents (56.3%) said they used models like CI/CD, static and dynamic security analysis, and other safe coding best practices. Thirty-six respondents (41.2%) said they employ digital signatures or similar procedures to confirm that the program or data is from the anticipated source. However, the majority, thirty-four respondents (40%), said they do not. Therefore, the analysis shows that not all Organizations follow secure practices to prevent Software and Data Integrity Failures Vulnerability (OWASP, 2021).

9. SECURITY LOGGING AND MONITORING FAILURES

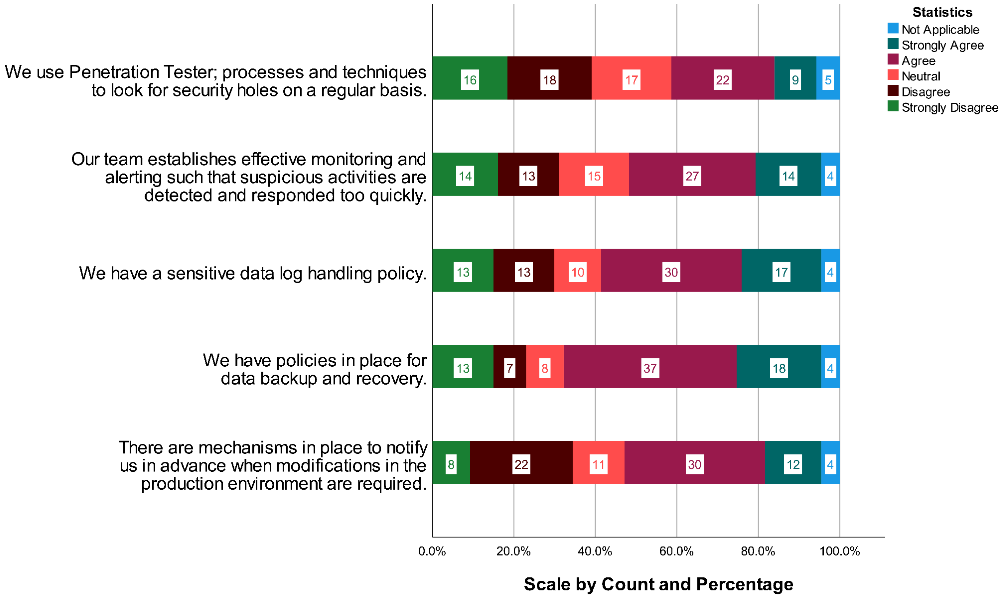


Figure 26: Analysis of Security Logging and Monitoring Failures Vulnerability

While thirty respondents (34.5%) disagreed, the majority, forty-two respondents (58.6%), said there were procedures in place to alert them when changes to the production environment when necessary. While twenty respondents (22.9%) said they lacked data backup and recovery procedures, fifty-five respondents (63.2%) said they did. A sensitive data log handling policy is also in place for forty-seven respondents (54.4%) but not for twenty-six respondents (29.8%) of them. While twenty-seven respondents (31%) were still dubious, forty-one respondents (47.1%) were convinced that their team established adequate monitoring and alerting to identify and deal with suspicious behaviors immediately. Only thirty-one respondents (35.6%) claimed to use "Penetration Tester" methodologies and procedures to look for security flaws regularly, while thirty-four respondents (39.1%) did not. Thus, the analysis shows that not all Organizations follow secure practices to prevent Security Logging and Monitoring Failures Vulnerability (OWASP, 2021).

10. SERVER-SIDE REQUEST FORGERY (SSRF)

In order to restrict undesired traffic from the network layer, thirty-seven respondents (42.5%) agreed to use "deny by default" firewall policies and network access control rules; however, twenty-seven respondents (31%) did not. In addition, forty-three respondents (49.4%) agreed that they check and sanitize all client-supplied input data, whereas twenty-five respondents

(28.7%) disagreed. Hence, the analysis indicates that not all Organizations follow secure practices to prevent Server-side Request Forgery Vulnerability (OWASP, 2021).

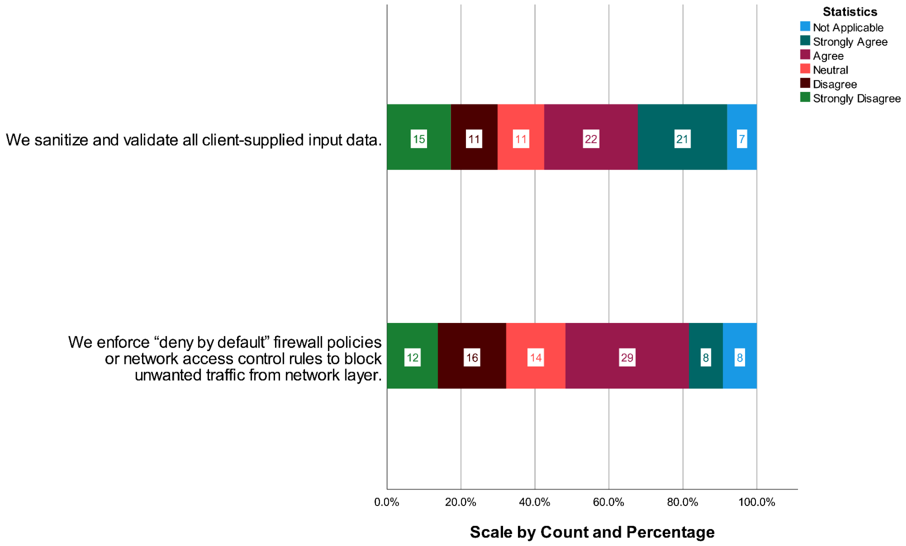


Figure 27: Analysis of Server-side Request Forgery Vulnerability

5.2 Qualitative Data Analysis

From the quantitative analysis, we looked at Nepalese Organizations' familiarity with the top 10 OWASP vulnerabilities and found loopholes in their systems. We also discovered that few Nepalese Organizations emphasized deploying security in each software development life cycle (SDLC) stage. Hence, interviews were conducted to acquire a more in-depth understanding of respondents' viewpoints based on their experience. The interview aims to triangulate quantitative data obtained and contextualize them. From the interview, we seek to learn more about how extensively the Organization in Nepal practices web application security and perceives security as a critical part of web application development. Furthermore, it was evident from quantitative research that Organizations in Nepal were not adhering to the best security procedures; therefore, we sought to understand the underlying causes.

Appendix B contains the interview protocol. The interview took place through a Zoom video call. The interview audios were analyzed using the Thematic Analysis approach (Cruzes & Dyba, 2011) to compare and contrast the insights and knowledge discovered from the information received from each group of experts. Thematic analysis offers a purely qualitative, in-depth, and comprehensive analysis of the data (Braun & Clarke, 2012). The interviews' audio recordings were utilized to transcribe them. Transcribing for each recorded response began 24 hours after the interview. After the recording files had been wholly transcribed, they were read aloud several times to gain a sense of how the participants felt about specific topics (Agbo-ola, 2022). Recording and transcribing obtained data is widely viewed as crucial for maintaining the study's quality because it can boost its reliability and accuracy (Sarker et al., 2013). Any private or Organization information was removed during this procedure to comply with the confidentiality agreement.

Table 6: Demographics of the Interviewed Experts

Expert No.	Job Profession	Years of experience	Point of contact with Security
Expert 1	Security Analyst	8 years	Security Analyser
Expert 2	Senior Software Engineer	8 years	Web Application Manager
Expert 3	Backend Developer	3 years	Web Application Developer
Expert 4	Software Engineer (Both Frontend and Backend Developer)	4 years	Web Application Developer
Expert 5	Frontend Developer	5 years	Web Application Developer
Expert 6	Information Security Officer	15 years	Security Planner

5.2.1 Descriptive Analysis of the Interview Answers

Web Application Security Management: We discovered through quantitative analysis that Nepalese Organizations had not managed web application security very well. We thus questioned their perception of web application security management to understand it better. Three specialists covered various aspects of web application security, such as secure code, adhering to security guidelines, API security, and session management. One expert said their Organization has a separate team that deals with security aspects like maintaining CIA, database injections, banner grabbing, and brute force. Two experts stated that security management does not fall under their job description, and they have less idea about web application security management. Expert 4 said, "*My work description does not include* *software security; hence I am not responsible for it. In the [software] lifecycle, [someone else] is in charge of software security.*" Similarly, expert 5 also believed the same. In general, it was found that in the context of Nepal, security aspects were not well managed at the Organizational level. Most interviewees were aware of how important it is to add security in each phase of the lifecycle and understood how crucial it is to protect data. However, most participants did not consider human management a big concern until we specifically questioned them about it.

Observation 1: *Baby steps in web application security management. It looked like organizations in Nepal are just slowly having that conversation about web application security but not even close to the level it should. It seems to be not well managed by organizations. Also, the human aspects of security look to be not even mentioned anywhere. It was observed that experts only considered technical aspects of security while the non-technical aspects were ignored fully or unmentioned.*

Security Policies: From the quantitative analysis, we discovered that most respondents agreed that they have a strong password policy, sensitive data log handling policy, data backup, recovery, and firewall policy, and agreed to many more such statements. However, when we asked experts regarding the security policies in the Organizations, surprisingly, Expert 1 said they do not have proper planning or policies since they do not have big applications. Expert 1, for instance, said, *"Most applications have an issue with weak password policies. Applications frequently provide quick, easy passwords that never expire, " implying* they had a weak password policy. Whereas expert 2 reported that the Organization used services like Azure or other cloud service providers. They had a security team that would guide them about various security vulnerabilities." Experts 4 & 5 said, "we do not have a well-defined security policy for web application vulnerability and did not elaborate on it." Expert 6, however, shared a brief list of policies the Organization implemented to maintain web application security, including a data backup policy, access control policy, sensitive data log handling policy, network security policy, and system security policy. Expert 5 said, "*we have certain policies in place when building new web applications."*

Observation 2. *There seems to be a lack of proper processes and policies that would govern and guide security experts and developers to implement security measures in their web applications. Some organizations seem to be doing it, but they do not cover all security aspects.*

Secure Software Development Life Cycle (SDLC): Even though about half of the respondents understood the Secure Software Development Life Cycle practice and agreed that they implement security in each phase of the Software development life cycle (SDLC), they were still being attacked. So we wanted to understand the underlying cause and questioned participants if they integrated security in each phase of SDLC. The participants discussed the importance of the need to keep security in mind throughout the whole development process. Expert 2 believes it is crucial to consider security in each phase of the SDLC to ensure that the applications are not subject to attacks from hackers. He claims, "*It is needed thing to include in each phase of the SDLC […] Anyone can potentially attain credentials to your source code, you must ensure that you are coding with potential vulnerabilities [...] having a robust and secure SDLC process is critical to ensuring your application is not subject to attacks by hackers.*" Expert 1, however, claims that software developers do not address web application security consistently in each step of the web development life cycle. Expert 1 adds that security is mainly considered when the application is fully developed. He stated, "*Some companies have a DevOps team that does look at the security measures during all the phases [...], but it is most common that they do it once everything is built.*" Expert 3 also claimed that security is not consistent in every phase of the cycle. He stated, "*The software developers do not consistently address the inclusion of security in software development from the initial design phase.*" Similarly, Experts 4, 5, and 6 responded that they were not implementing web application security in most of the SDLC phase because they did not have the time or resources to implement it.

Observation 3: *Security is not considered in every stage of the Software Development Life Cycle. While some organizations do not have security measures in place, some do it after the completion of the project. For others, implement it in some phases of the SDLC but not in all phases.*

Motivation and Support: Several security flaws were discovered within Organizations, even though most survey respondents agreed and strongly agreed they were aware of OWASP vulnerabilities. Thus, we interviewed several experts to determine this cause and discovered numerous explanations. For instance, during the interview, it seemed like most Organizations were motivated to invest in web application security only when there was some attack or exploit of web applications vulnerability. Expert 4 said*, "Because of a security compromise on software similar to the one I work on, management is suddenly concerned with protecting* *our apps."* However, expert 1 stated that, sometimes, even after being attacked and even if the Organization support it, some employee does not care and make mistakes again. He said, *"Even sometimes when we get attacked, people in our Organizations do not have a habit of learning from the past. In some cases, Organizations are supportive, but the developers neglect it."* It implies that there is a people or human issue with it. However, one of the experts was motivated by his work. He said he likes to make informal inquiries by casually asking a colleague and seeking assistance from others who have worked on related problems when working on a software security issue. One of the experts (expert 5) expressed that they do not receive enough compensation from the Organizations to be dedicated. He said, "*Dedicated security professionals are not economically viable for most smaller-scale companies and projects."* Another expert (expert 1) said that the primary motivation for some Organizations was the legislation that required some Organizations to maintain a certain level of security in their applications. When asked further, the expert shared examples of banks and insurance companies. He said, *"In my six years of experience, banking sectors invest the most in web application security in Nepal. The main reason is not solely because they are concerned with security, but they have to do it because of the regulations of Nepal Rastra Bank (the banking governing body of Nepal). They do it to fulfill the requirements. Some health care services are also interested in this."* Likewise, expert 3 stated that his Organization must adhere to security requirements sometimes; as a result, his team conducts security testing whenever they anticipate an external audit *"to ensure the auditors cannot uncover any vulnerabilities during the penetration test."* Most of the experts mentioned that they did not get much support from their Organizations, which is very common in Nepal. Expert 2 claimed that companies might be willing to accept the security assessment as long as the process will not impact the product delivery time keeping in mind the additional budget required.

Observation 4: *Employees were less motivated by their work. Also, Motivation starts when there is an attack. Organizations are motivated or concerned about security only when there is an attack on their system or if the local regulations demand them to conduct a security audit. Proper support is also provided when there is an attack but not necessarily before that.*

Attitude, Behaviour, and Involvement: Several security flaws were discovered within Organizations, even though most survey respondents agreed and strongly agreed they were aware of secure code practices. Thus, we interviewed several experts to determine this cause and discovered numerous explanations. For instance, the interviewees were asked some questions to analyze their attitudes and behavior toward web application security. The experts had completely different answers to this. Expert 4 said they had a mental checklist of security aspects developed by their own set of recommended procedures to check security vulnerabilities. He stated, *"I have a mental checklist of the software security problems I need to take into account when writing my code. I've developed my own set of recommended procedures for software security."* Another expert (expert 5) said that code reviewers do not necessarily know security vulnerabilities and want others to cover the security aspects. Expert 3 said that security was not their responsibility and assumed that security would be taken care of by other components or people. He said, "*I do not care much about security. I assume that [another component] of the system or [another person] has already addressed it, so at this time I am secure and shouldn't worry too much."* Similarly, when asked how he deals with accessibility vulnerabilities, expert 3 responded that they would look at the security related to the back end but were unsure about the front end. He said, "*I am a backend engineer. Verifying accessibility vulnerabilities, for instance, is something I could not readily perform because I am not a front-end specialist.*" Expert 5 had a similar attitude to Expert 3 when he was asked about handling security during code reviews. He said, *"We deal with security issues less frequently; I am more often assigned a front-end programmer these days [...] I believe that security considerations during code reviews should focus more on the back end."* The statement shows developers were considering security as someone else's job. Expert 6 believed that the developers are involved in maintaining the web application security. Moreover, based on the impressions from the interviews, the developers seemed to be relying upon the security team to handle the security components. They would like to be less involved.

Observation 5: *The perception of web application security differs from person to person. Mostly, security is perceived not to be in their job roles. The common understanding is that the security team is the one that should be handling security aspects of the application because of a lack of knowledge or other resources on their end.*

Organization's Perspective in evaluating Web Application Security: Analyzing the responses revealed that experts were not entirely comfortable handling the web application's security components. They also felt they were less concerned because they believed that their applications were secure and not so vulnerable. They had not been attacked in the past and used automated vulnerability tools. Expert 4 said, *"I don't believe we need to make any* *changes regarding software security since we're doing okay. Our program is not vulnerable to security breaches. There have been no security lapses; everything is perfect as it is. We also have automated vulnerability scanning tools [...] it's unlikely that attackers will attack us."* Also, since Nepal is a small market, experts believe that attackers would be less likely to be interested in it. Expert 3 said that the developers maintain a minimum security level per the Organization's compliance requirements. Moreover, the Organizations were wary about the security handling of third-party components. Expert 5 stated, *"When creating new applications, we tend to employ lots of third-party components of trusted ones."* However, these third-party components might also be subject to several security vulnerabilities.

Observation 6: *Developers had assumptions that they were not entirely comfortable handling their web application's security aspects. Moreover, they believed that their applications were secured enough and would not affect by attacks.*

Awareness and Training: Not all respondents agreed that they received the appropriate training and awareness. Qualitative evidence was used to back up these quantitative findings—one of the experts mentioned how their companies steadily raised their security awareness levels. Expert 1, for instance, described how their company began as a small and how its expansion impacted the web application's security. He said, *"When I joined, the application had a very low scale engagement, so it was not such a big concern. However, now it is used by over 10,000 people... we started to have a certain kind of care. We strengthened the web application's security."* However, another expert claims that fewer developers have proper security knowledge and training, which brings an issue of security measures being applied at every level of the Organization. Expert 4 highlighted that to implement security measures in their web applications; all the developers must be trained and aware of those vulnerabilities. However, there is no process or resources which would help them achieve it. Another expert (expert 3) expressed that the developers lack the time and effort to invest in security training, although they know the importance of web application security. Expert 2 believed that even the training is insufficient for all developers to become sufficiently skilled. According to expert 2, *"Even while some training could be beneficial, it is definitely unreasonable to expect all developers to become sufficiently skilled in IT security. Maybe [...] better tool support and (enhanced) frameworks, libraries, API [...] are needed to address this issue."* Expert 1 demonstrated how important awareness is, but the Organization lacks it. He said, *"[Security assessment] requires specialist expertise (exploit* *tactics), and if they do not have the relevant skills, it might be difficult to persuade coworkers that there is a security [vulnerability] in their code without conducting a demonstration."* Furthermore, expert 6 said, *"Even while I am aware of several major security issues, I am not sufficiently briefed. As a result, sometimes I fail to notice [little things] because I am clueless that they might result in a flaw."* The statement shows the lacking of an awareness program inside an Organization which was also agreed upon by experts 4 and 5.

Observation 7: *Employees are not trained enough even though they know security is essential. Experts somewhat believed that security could be a big issue in the coming days, but their major problem was that they did lack proper knowledge on the subject. They believe that training would help them learn how to make their applications more secure. The good thing is that people in organizations have become aware that security is essential, but they still lack proper knowledge and training.*

Organizational Communication: Even while some Organizations had defined regulations, we discovered that many employees were unaware of them, and the leading cause was a lack of communication. Many experts claimed that their Organization still does not have an internal communication plan or strategy, which hinders them from being aware of the security rules. For instance, Expert 3 stated, "*Many workers disregard established rules because they are unaware of them and which makes us more vulnerable sometimes."* Also, Expert 4 said*, "Because of our poor communication, our Organizations are frequently stressful, discouraging individuals from working hard and discouraging employees from collaborating."* Similarly, expert 5 stated they were unaware because *"employees feel informed about the important company updates and are not engaged in [daily] Organizational conversations."*

Expert 1 also noted that inadequate communication was to blame for bad policies in a company. He discussed the value of communication in a company. He stated, "*Teams in charge of security are in charge of monitoring an Organization's security. In order to ensure that rules, processes, and pertinent breaking news are widely and frequently conveyed, security must have an internal voice in the form of communications representation. Not only that, there should be a feedback policy such as 360-degree feedback so that everyone can be aware of the strength and weaknesses of each other to improve professional and personal life."*

Observation 8: *There is no proper organizational communication strategy to update employees on new or existing policies, secure practices, as well as to give and receive feedback from and to employees.*

Priority: The quantitative data analysis discovered that Nepal's web application's security was not adequately managed. We discovered that security was not given as much priority after the interview. For instance, Expert 4 mentioned that usability and features were their main concerns when creating web applications. He stated, "*Our main focus while developing web applications is always the usability.*" They seemed to care less about the security aspect of it. Another expert (expert 3) during the interview added that security is generally ignored because there are many other things to do, and they have a deadline to deliver the application. He said, *"Delivering products on schedule is more crucial to my team than addressing software security. Even though I recognize how crucial it is to handle security, I won't waste my time on it because nobody else does."* Similarly, most experts agree that there are occasions when there is a rush and security goes ignored. One of Nepal's primary problems with developing web applications was that usability is frequently prioritized over security.

Observation 9: *Security rarely comes as a priority. People are so focused on developing application features that they never get the time to think about their security.*

Resources Constraint: During the conversations with experts, one of the primary reasons why security was not prioritized was the lack of resources (time, budget, etc.). One expert stressed that the financial aspect is the biggest challenge in Nepal in this context. He said, "*The only challenge we have in Nepal is the financial aspect. Though we have the resources to implement security, it just won't fit in the time and budget that we get."* It infers that Organizations do not have enough money to spend on security. However, one expert stated that their business has some budget for security, and they employ third-party security specialists to conduct further application security testing. However, they sometimes defer this stage until right before the application is released because the testing procedure costs them a lot of money. Another reason is that Nepal does not have enough security experts (workforce) who can do security checks on the applications, as highlighted by one interviewee. He added, "*Unfortunately, I do not believe we have enough individuals who are security specialists [...] who are able to conduct a rigorous security verification in all of the code modifications we have. Moreover, those who have the skills go out of the country to work because of the lack of* *opportunity in Nepal."* Similarly, another expert came up with the same issue about lacking a workforce. He said, *"We create applications that require high levels of security, yet we do not have many security professionals who can properly evaluate our code."* Another expert stated that even though their applications require a high level of security, they do not do it because of a lack of budget.

> ***Observation 10:*** *The resource is a significant constraint. One of the biggest reasons organizations are not investing in security is that they lack the financial or human resources to do so, especially in the least developing country like Nepal.*

5.3 Validity and Reliability

The discourse criterion (the strength of the research's statements and arguments), heuristic value (the result should be new thinking), empirical base (the result should be based on reality), texture (all parts should be a part of a whole), and the pragmatic criterion (the result must add value) are some criteria for good validity mentioned by Starrin and Svensson (1994, p. 177-186). However, Qualitative research's validity is difficult to evaluate (Golafshani, 2003; Onwuegbuzie & Leech, 2007). Despite our most significant efforts, there are certain restrictions. In this section, we explain how we attempted to reduce them. We could have influenced interviewees to respond better during the interviews (Hildum & Brown, 1956). We questioned our conclusions and used the survey results in a triangulation process to help reduce this problem. Our research participants might have socially acceptable responses to appear favorably. In order to lessen this social desirability bias (Donaldson & Grant-Vallone, 2002; Furnham, 1986), we let participants know that their answers would be kept anonymous.

Furthermore, Acceptable reliability necessitates a thorough explanation of how the study was carried out, which the researcher believes the methodology section provides. According to Ryen (2004, p. 137), some qualitative researchers claim that reliability, and even validity, only apply to quantitative research, where the difference between natural science and social science is irrelevant. It is hard for research to provide information if there is no reality on which all researchers agree. Instead, the researcher must demonstrate that he or she is aware of the ramifications of the approach adopted (Wigmo & Wikström, 2010).

6. Discussion

This section provides an additional in-depth analysis of the findings, discussing the findings in light of the theoretical background described in Chapter 2, their practical applications, and a discussion of the study's limitations and future directions.

6.1 Summary of Findings

The thesis development direction was based on and helped by the OWASP source, a highly reliable and current source in the field of cybersecurity. Every day, thousands of specialists and enthusiasts add to the realm of secure web applications (OWASP, 2021). The OWASP community published a new Top Ten (2021) list following four years of work on the OWASP Top Ten (2017) categories. Based on the quantitative research result, we understood that the web applications in Nepal were vulnerable; many were still unfamiliar with OWASP's top 10 vulnerabilities, most Organizations did not practice secure coding, and security was not implemented in each phase of SDLC. Hence, we conducted a semi-structured interview with some security experts to be more confident in the results, get more information about why this was happening, and get deep insights into respondents' opinions. We found various underlying causes of why Organizations in Nepal lag in managing web application security. This sub-section compares security practices from our analysis to standard practices discussed in the theoretical background and presents factors impacting such practices.

6.1.1 Best Practices versus Current Practices

An explanation of the top 10 web application vulnerabilities that are there, how they can be exploited, and how to address and mitigate them has been given in the theoretical background. This sub-section compares the current security practices in Nepal with the best standard practices.

Familiarity of top 10 OWASP vulnerabilities within Nepalese Organization

Surprisingly, we found that Nepalese Organizations are most familiar with Injection vulnerability which holds third position in OWASP top 10 vulnerability. In Contrast, we found that Organizations are least familiar with the Broken Access Control vulnerability on average which is strange because it holds the top one position in the OWASP top 10 vulnerability. Similarly, the second most vulnerabilities that Organizations were familiar with was Identification and Authentication Failure vulnerability which holds seventh position in the OWASP top 10 vulnerability. Whereas, the familiarity position of Security Misconfiguration vulnerability was in the same place as it holds in the top 10 OWASP vulnerability. The familiarity of top 10 OWASP vulnerability within Nepalese Organizations is mapped with OWASP top 10 vulnerability as shown below in the figure 28.

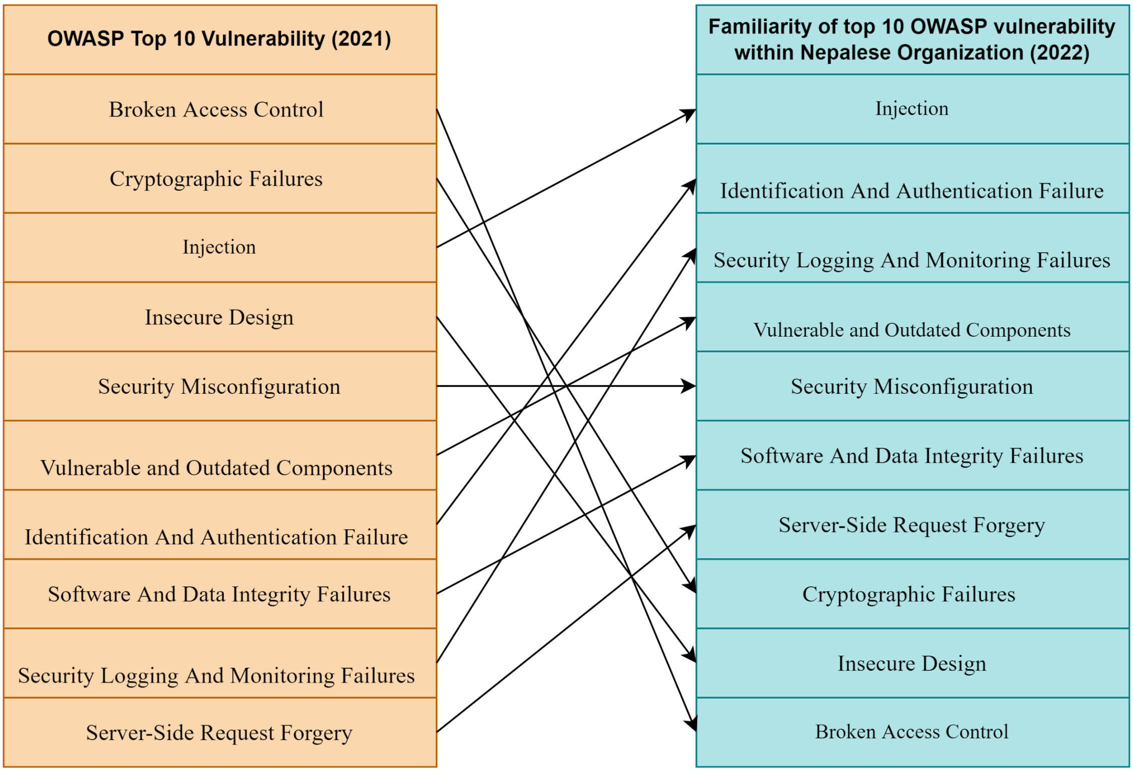

OWASP Top 10 Vulnerability (2021)
Broken Access Control
Cryptographic Failures
Injection
Insecure Design
Security Misconfiguration
Vulnerable and Outdated Components
Identification And Authentication Failure
Software And Data Integrity Failures
Security Logging And Monitoring Failures
Server-Side Request Forgery
Familiarity of top 10 OWASP vulnerability within Nepalese Organization (2022)
Injection
Identification And Authentication Failure
Security Logging And Monitoring Failures
Vulnerable and Outdated Components
Security Misconfiguration
Software And Data Integrity Failures
Server-Side Request Forgery
Cryptographic Failures
Insecure Design
Broken Access Control

Figure 28: Familiarity of OWASP top ten vulnerabilities in Nepal

The Data Analysis section analyzes the top 10 vulnerabilities identified by OWASP for Nepal. Through the survey, we discovered that Nepal's web applications are probably vulnerable. Security for web applications is not always effectively managed. The system has several flaws. A few of the issues we encountered frequently are listed below. We also mapped the solution, which we discussed in the theoretical background.

From the data analysis, it is seen that even though most of the Organizations agreed they were managing Broken Access Control, they were not confident enough to make their web application send an alert even if a user typed in an incorrect username or password repeatedly, which provides a great deal of possible information to an attacker. An attacker may create a list of all legitimate users of that specific application using an automated script. This medium poses a significant danger, particularly for financial and telecom applications. Also, we found that there were issues with access management. Security regulations should not stop anyone from accessing information or using systems necessary to perform their jobs. However, they also should not permit anyone to access anything that is not necessary (OWASP, 2021). Such a strategy's flexibility and practical application necessitate accurate tracking of who needs access to what. It takes a lot of time and resources to keep track of this information matrix and update it to reflect changes in staffing and business procedures. Using groups, each of which has its rights, is the most typical method of simplification (Acronis, 2020).

In terms of cryptographic failure vulnerability, on average, respondents agree that they have recently examined their Secure Socket Layer configuration so that the clients are supplied with secure protocols and ciphers. The average score was very close to neutral, so many disagreed with this, which means there is room for improvement in this area. Out of eighty-seven respondents, only fourteen respondents strongly agreed that they have a robust cryptographic algorithm. If the data is encrypted using a poor cryptographic technique, it may be quickly decrypted to reveal the original plain text (Rahalkar, 2016). It was also seen that some Organizations did not feel it necessary to encrypt sensitive data. The survey respondents agreed that the privates are protected on the web servers. However, none were fully agreeing to it. It means that Organizations in Nepal can do better in protecting the private keys they use.

Nevertheless, they should understand that any device, either a networked computer or a remote device that transmits, receives, or stores data, can protect that data with encryption. Respondents were neutral on average when asked about void mixed content problems which is an area of improvement in the context of web application vulnerability. Every business needs to make the most of this crucial security endeavour, no matter how big or little. Data privacy is crucially dependent on encryption. Encryption successfully scrambles legible text so only the owner of the decryption key can decode it (Acronis, 2020; OWASP, 2021).

Although the average respondents agreed that they have some input sanitization, we found that many Organizations still did not have sufficient input sanitization, which is critical and can lead to various web application attacks. Input sanitization is a cybersecurity solution that involves verifying, cleansing, and filtering user, API, and web service data inputs of undesirable characters and strings to prevent the introduction of hazardous codes into the system (OWASP, 2021). When asked whether the Organizations load active material from third-party servers or not, the respondents said that they agree to it on average. Contrary to that, the average respondents stayed neutral when asked if web applications can mitigate vulnerability related to processing and manipulating Extensible Markup Language (XML) provided by the user. This medium turned out to be one of the most prominent vulnerable aspects concerning web application security in Nepal's Organization. One of the effective ways to mitigate this vulnerability is to disable features making the XML processor weak and the application vulnerable (Singh, 2021).

Another vulnerable aspect that resulted from our analysis was a templating system. Most respondents voted neutral on average and stated that they do not consider it when it comes to their web application system. One of the most significant vulnerable aspects of insecure design is that the Organizations do not perform integrated plausibility checks into the applications in each phase of application development. The respondents seemed to agree that they performed unit testing or release testing to mitigate insecure design vulnerability but did not strongly agree. Likewise, the Organizations implemented a monitoring strategy, but the respondents were not entirely confident about it. Another big vulnerable area, as seen from the data analysis, was security misconfiguration and its various components. From the average respondents' answers, they were neutral about having automated processes to verify the effectiveness of the web application security configuration and settings, which can be an area of improvement for the Organizations of Nepal. The average respondents agreed that they disable unnecessary features in their web applications to make them more secure. One aspect that Organizations can improve is not allowing insecure default settings and asking to change passwords regularly. The average respondents chose neutral on this aspect which can be a severe vulnerability issue because attackers can exploit the default setting and try to enter the system. Also, security experts were neutral when asked about having the latest security features enabled in their web applications.

We discovered that most of the Organizations' administrators in Nepal frequently neglect to delete outdated files and make a backup before publishing an application, leading to configuration management issues concerning improperly configured web servers and the environment. Even if an application is very secure, if it is deployed in an unsafe setting, it remains susceptible to attacks (OWASP 2021; Rahalkar, 2016). Patching frequently guarantees that issues are resolved as soon as feasible, reducing the exposure to potential risks. From the survey, it was clear that keeping track of all the software used by a company was a challenging chore for Nepal, especially when they must match installed software to new releases of patches and versions that arrive at frequently erratic intervals. Patches for significant new vulnerabilities can be issued relatively fast, but new threats can also take advantage of such flaws. Installing such patches as soon as possible protects Organizations from these new hazards (Acronis, 2020; OWASP, 2021). However, to receive early warnings of potential issues, one can sign up for reliable alerts on newly discovered vulnerabilities and updates that affect essential software and tools (OWASP, 2021). On average, most Organizations concur that they are doing a good job defending against Identification and Authentication failures in their applications. However, the majority of the Organizations were not consistent in believing that they had excellent session management. According to OWASP (2021), sessions and connections always need to end after logging out.

Moreover, the same User ID should not be used for multiple logins, and the session inactivity timeout interval should be as brief as possible, considering the risks and business objectives. Furthermore, not all businesses felt secure enough to assert that they had a solid password policy, which is crucial. Organizations should use weak password tests, such as comparing newly created or modified passwords to a list of the top 10,000 worst passwords. Password length, complexity, and rotation rules should conform with the recommendations in section 5.1.1 of NIST 800-63b for memorized secrets or other current, research-based password policies (OWASP, 2021). Following that, most Organizations were not sure they had a procedure to recover lost or forgotten passwords. The Organization should not use flimsy techniques for forgotten passwords and recovering credentials.

Most Organizations did not employ digital signatures or similar tools to confirm that the software or data came from the expected source. In order to guard against web application attacks, they must make careful to confirm that it comes from the expected source. Even while most of the Organizations agreed with the other questions we asked about these risks; they lacked the confidence to concur strongly. For instance, most Organizations still do not have a procedure for reviewing code and configuration changes. Businesses must ensure that there is a review method for modifications to code and configuration to lessen the likelihood of harmful code or configuration entering the development pipeline, claims OWASP (2021).

Similarly, not all Organizations strongly agreed that they were preventing the transmission of serialized data that is not signed or encrypted to clients who are not trusted without some integrity check or digital signature. Additionally, the Organizations were reluctant to say that they used secure coding techniques. They must be concerned about this since adopting safe programming techniques is crucial for protecting against web application attacks.

Most Organizations lack the assurance necessary to ensure that their team has set up efficient monitoring and alerting to identify and deal with suspicious actions quickly. Breach detection is impossible without logging and monitoring (OWASP, 2021). Furthermore, not many Organizations felt secure in their policy for handling sensitive data logs. In order to guard against attacks like injections, they must ensure that log data is encoded securely. Additionally, it was observed that Organizations lacked policies for data backup and recovery. As a result, they should create or implement an incident response and recovery plan, such as NIST 800-61r2 or later, from the National Institute of Standards and Technology (OWASP, 2021).

SSRF issues can arise when a web application gets a remote resource without verifying the user-supplied URL. In order to prohibit undesired traffic from the network layer, it was discovered that many businesses did not apply "deny by default" firewall policies or network access control rules. Because of cloud services and sophisticated infrastructures, SSRF is growing more severe. All client-supplied input data must be sanitized, validated, and HTTP redirections must be turned off (OWASP, 2021).

Consistency of Security in all Stages of Software Development

To our surprise, we found a shift in the developer's security viewpoint. In the past, developers have adopted the Secure Development of Applications (SDA) minimally and exhibited an "it is not my duty" mentality regarding security flaws (Xie, Lipford & Chu, 2011). However, software security appears to be taking up more of a developer's workload. In fact, according to research (Christakis & Bird, 2016), developers are more concerned with Security than other reliability-related issues. Our quantitative research findings imply a shift in developers' attitudes toward this trend, as most respondents claimed that it is their duty to ensure Security, which is very good. According to OWASP (2021), it is the developer's job to ensure. Security is integrated into each phase of the Software Development Life Cycle. However, we found from the interview that Security is not genuinely considered in every stage of the

Software Development Life Cycle. According to OWASP (2021), secure web applications can only be created when a "**Secure SDLC (Software Development Life Cycle)** is followed." Application developers must appreciate the benefits of this security solution, which calls for skill development and investment. Changes in the software development process for incorporating security practices are required due to the rise in web application attacks (Coutinho & Pinheiro, 2021).

Technology is not always a solution

According to the interview, most Organizations in Nepal rely only on automated testing. Automated scanning technologies cannot detect business logic errors (Rahalkar, 2016). The business logic of the application they used supplied the fundamental guidelines for how an application should operate. It is possible, nevertheless, that the business logic itself is flawed and contains specific problems. Attackers may use these issues to compromise the program. They can only be disclosed upon rigorous application architecture and design evaluation. However, using automated testing in conjunction with manual testing is the most precise way to identify vulnerabilities and risks. Monthly automated testing may be performed to deliver data on a regular, timely basis at a reasonable cost. At a more significant expense, manual testing may be carried out regularly to uncover vulnerabilities that the automated tester cannot find (Lepofsky, 2014). Regularly verifying that the defenses are in place and operating as intended is crucial to protecting the Organization's systems and data. Protocols are followed to ensure that an Organization's physical, software, and hardware protections are functioning correctly. No gaps can be detected in their attack surface; testing puts various components of their networks, systems, and even individuals, which entails routinely checking any tools or systems one could very well be operating for unexpected flaws (Acronis, 2021).

From the interview, we also found that most Organizations in Nepal focus on the technical test but ignore the management perspective (people and process). We found from the survey that most Organizations lost their proprietary information due to hackers, insiders, or business partners. Most Organizations in Nepal still thought this issue was being addressed with technology alone, but that is not realistic. The article (Cotenescu, 2016) demonstrated that only focusing holistically on people, processes, and technology can reduce the impact of data loss. Moreover, according to OWASP (2021), testing an application's technical implementation alone will not disclose significant managerial or operational problems unless a comprehensive strategy is adopted. A company can eradicate bugs early and identify the origins of defects by testing the people, policies, and procedures to find issues before they become technological errors. Only testing a small subset of a system's potential technological vulnerabilities will result in an incomplete and inaccurate assessment of the system's security posture. The elements of an effective testing program should ensure that people are informed and educated enough; to ensure there are enough laws and norms and that everyone knows how to follow them (process); utilizing technology, one can determine whether a technique has been followed correctly.

OWASP Secure Code Practices are not well implemented

Overall, it was found that most of the Organizations in Nepal didn't practice secure coding. The guidelines like OWASP Secure Coding Practices can be used which ensures that programmers create secure applications without leaving any potential threats to exploits. It can be used as a security protocol for all software development life cycles and deployment platforms to reduce risks related to poor coding standards (SecureCoding, 2021).

6.1.2 Factors Affecting Current Security Practices

Human Factors: From the study, we found that several human factors affect security practices. The biggest threat to information security comes from employees, who may do so maliciously or negligently, frequently due to ignorance (Niekerk & Solms, 2010). When guaranteeing information security across the organization, a holistic approach to part information security management highlights the significance of accounting for the "human" aspect (Antonsen, & Ekstedt, 2014; Siponen, 2005; Flores). It cannot be presumed that the current workforce has the skills to do their job efficiently. All actions would need to be carried out compatible with secure practices if an Organization attempts to promote a subculture of information security (Niekerk & Solms, 2010). Moreover, to establish a healthy information security culture, Organizations must ensure that technological systems and human behavioral aspects of information security management are combined (Sthapit, Submitted Mandatory Assignment, 2020). Some of the human factors we encountered in our studies are briefly explained below.

i. Security Knowledge and Training: Security knowledge and training were seen to be a factor affecting security practice from the result. Organizations in Nepal do not seem to be investing as much as they should in increasing knowledge sharing and training for their employees. Experts wanted their Organizations to invest more to increase their knowledge and training, but they felt it was hard to convince them. Developers need to be knowledgeable about typical security risks to their code. The Organizations should set aside additional funds for security training and development. Through team workshops that analyze intrusions and risks to their systems, companies should educate developers about the vulnerabilities their code may encounter. Developers should be informed of which components are security-critical by security experts (NCSC, 2019). The Organizations should encourage their developers to adapt their working methods to incorporate security and provide them with training.

ii. Security Awareness: The results showed that many Organizations were still unfamiliar with the top 10 OWASP vulnerabilities and their countermeasures, demonstrating that most Organizations lacked effective awareness programs. Assuring that all workers are aware of the policies and procedures for protecting the information within the firm is the goal of information security awareness. In the context of Nepal, it can be hard to channel awareness campaigns because of the lack of enough professional security community. OWASP Nepal is somewhat doing it, but it will take time to reach a broader mass. Security awareness is necessary to achieve desirable security behavior among a range of audiences inside an Organization, including business users, experts, top management, developers, and IT service providers (SecurityForum, 2020). The Organization must offer a strong awareness campaign. Better Information Security Policy compliance will result from this since the employee will be more assured that they possess the knowledge and abilities needed to carry out essential security activities (Alzahrani, Johnson, & Altamimi, 2018).

iii. Attitudes, Behavior, and Involvement: The qualitative research found that the perception of web security differs from person to person. Also, some do not take security as a part of their job as it was not mentioned in their job description. However, security is the responsibility of everyone in the Organization. Organizations in Nepal do lack not only technical aspects but also non-technical human aspects of web application security. Although technology-based solutions aid in the mitigation of some of the mass issues related to information security, even the finest technology is ineffective in the absence of ethical behavior on the part of employees in businesses (Stanton, Mastrangelo, Stam & Jolton, 2004). For whatever cause, an employee's failure to follow the Information Security Policy of their Organization may endanger data and systems. For instance, external attackers could take advantage of staff weaknesses. Security must be taken into account at every stage of development and must permeate everyone on the delivery team's attitude and behavior (NCSC, 2019). Moreover, Blaming Culture should be stopped. Everyone can make mistakes. So, if there is a bad culture surrounding identifying and correcting security flaws, there is a chance that these concerns will go unnoticed until it is too late (NCSC, 2019).

iv. Motivation: A clear gap can be observed regarding how employees are motivated toward maintaining Security in the applications they are developing or maintaining. Organizations in Nepal do not seem to be fully encouraging their employees to take the security aspects seriously. Few Organizations are doing it because they have legal obligations to do it or if the clients ask them to do it; otherwise, there is a lack of motivation. Employees need to be motivated to maintain high Security in applications, and it can be accomplished via awareness campaigns to promote engagement, conversations, knowledge sharing, and interaction among employees. Employees are more likely to be secure if they cooperate and are more inclined to follow their company's security policy (Alzahrani, Johnson, & Altamimi, 2018).

v. National Culture: A significant portion of the procedures required to safeguard web applications depends on cooperative human behavior. Many security recommendations we discussed previously might be overly general and ignore Nepal's cultural considerations. National culture must, however, be taken into account. OWASP top 10 is not a common topic regarding Nepalese security culture. Companies might build a more effective information security function that matches the cultural contexts of the nation in which the business operates by recognizing the effects of national culture (Antonsen & Ekstedt, 2014).

Policies: The policy must be specified to ensure that web applications uphold an Organization's security infrastructure, compliance, risk management, and change control. However, just as compliance rules are unique to each industry sector, state laws governing cybersecurity obligations differ from state to state. Knowing which of these Organizations impacts the Organization is crucial (Acronis, 2020). The demands and guidelines these regulatory Organizations follow can then be ordered. Though initially intimidating, they are developing an information security infrastructure to safeguard the Organization's assets, services, and personnel while also adhering to all legal requirements can significantly lower operational risk to the network and help ensure the Organization's future-proof against the company's possible threats (Acronis, 2020). Based on our research findings, Organizations do not implement security policies unless there is a breach or if the law wants them to have a policy in place. The Organizations should routinely check in with all pertinent staff to ensure their objectives are being fulfilled and update rules to consider any new legal or business demands. Moreover, Organizations should encourage their employees to input on a particular policy or, at the very least, let them contribute to the subjects and ideas of the awareness campaign. Employees will thus have more incredible support and value, which may favor their behavioral intentions regarding Information security policy compliance (Alzahrani, Johnson, & Altamimi, 2018).

Organizational Communication: From the findings, we discovered many Organizations did not have adequate Organizational communication. However, effective communication is essential to an Organization's growth and development in the current global environment (Neves and Eisenberger, 2012). The finding (Arhin & Wiredu, 2018) showed that both information security preventive and response techniques would surely fall short of producing the expected results without effective communication. It lessens misunderstanding and disinformation, which therefore lessens demotivation and complaints. It guarantees that workers have a voice, and that voice might offer the criticism the Organization needs in order to develop and prosper. Employee purpose and motivation will be enhanced by a better understanding of the company's goal and vision (Guthrie, 2019).

Experiencing a Security Incident: As per findings, we realized that some Organizations and developers only cared about security after they faced the breaches themselves. A business should learn from its mistakes and demonstrate to employees how the applications they are developing were breached. This process may encourage the developers to strengthen the security of the program.

Technology Advancement: According to an interview, recent technological advancements have hampered web security testing in Nepal. The fast-evolving digital world has a more significant influence on information security. The rise in cyberattacks is significant and consistent with the advancement of technology. Thankfully, the capacity to prevent has advanced along with technology. When frequent changes happen, and new security vulnerabilities emerge, it can be challenging for Organizations to re-invest in identifying vulnerabilities in their new and existing systems, especially in Nepal, where there are budget constraints.

Resources Constraint: It was observed that most Organizations' security measures were impacted by funding, time constraints, and workforce. A brief explanation of each is provided below:

i. Budget: Because they did not know how to spend their budgets efficiently, we found that many firms neglected security. While some groups lacked sufficient funding, others planned to spend more in the future years. In order to secure information security throughout their whole company, some Organizations in Nepal are still investing a significant amount of money, yet even with high priority given to these expenditures, Organizing these efforts remains difficult. Therefore, top managers should understand this as the net advantages of an investment must be calculated before choosing an investment in information security. For this, the IT security governing body should ask for data visualizations showing how the price of risk and mitigation are related (Lepofsky, 2014). Both during the budget proposal process to explain the anticipated plan and after the budget cycle to display the actual outcomes, the presentation process should take place. A straw poll should be conducted with financial managers and other interested parties to determine the possible financial consequences of security breaches (Lepofsky, 2014).

ii. Time Pressure: We found many Organizations participating had security problems, such as web application breaches which may result in Security being delayed in favor of functionality and timely delivery. One of the primary problems we found from the in-depth interview was developing web applications in Nepal: usability is frequently prioritized before Security. Thus, security issues must be considered to solve the worries about web application vulnerabilities.

iii. Manpower: It was found from the interview that Organizations sometimes have to rely on a third party due to a shortage of qualified labor or fewer security tools. They examine the application and provide a certificate per standards attesting to its Security before allowing it to be hosted. They must re-run the security audit whenever modifications are made to the application. It is highly challenging when a private vendor is needed to perform a security audit again.

6.2 Limitations

Even though this study followed the proper procedures, there were some restrictions. Grey literature, whose content could not be comprehensive or was likely to contain errors, was also included in the literature search for this thesis. The study's primary focus on web application vulnerabilities was disclosed to the participants, which may have caused them to intentionally avoid it by altering their responses to match it, i.e., demand characteristics (Braz & Bacchelli, 2022). Additionally, even though our sample includes several Organizations and there was strong agreement among them on their perspectives, we cannot assert that it is typical of all Organizations from Nepal. The outcomes may alter if the study is redone with different individuals. Additionally, the questionnaire we created did not go into depth about every vulnerability, so we cannot be sure their systems are safe/unsafe of all vulnerabilities by asking the questions we did. Thus, future researchers may add more questions to uncover more vulnerabilities fully. Furthermore, since we deal with questions that may have sensitive answers and compromise the company they work for, this research did not collect information that could be used to identify the interviewee, such as name and company, to safeguard their identity and make the interviewee more comfortable answering the survey sincerely. Because of this, there was a chance that multiple employees from the same company would respond to the survey (Coutinho & Pinheiro, 2021). In this case, we would not be able to tell whether our analysis had reached a general conclusion or if the responses had been skewed because only one company or a group of similar companies had participated. However, we tried to mitigate this problem by contacting as many employees from different companies as possible to improve the survey's statistical data. Last but not least, it took a substantially long time to collect and analyze the two different data sets (quantitative and qualitative data) since simultaneous data collection is not supported by a sequential explanatory approach, creating a time limitation.

6.3 Implication for Practice

The results of this study have an impact on both the world of academia and businesses. The currently available literature is investigated to comprehend Web Application Security management mentioned in an Organization. We discovered that no studies had been published that analyze different elements of web application security management in the context of Nepal. The trends, limitations, and needs of current web application security management are well shown in this paper. The results will thus give researchers looking into the business in Nepal a foundational understanding of how processes work. Additionally, a complete questionnaire design cycle was used to ensure that the survey given to participants was founded on well-recognized scientific ideas. Numerous significant takeaways that were discovered as a consequence of the examination of the survey data are helpful for future scientific research and industry practitioners. Our effort creates awareness about secure coding rules in the scientific community and among software developers about this challenging subject. In order to aid in a future study on the subject, we also offer the raw survey data.

6.4 Implication for Research

We only collected data through a small sample of surveys and interviews, which might not be sufficient to represent the entire industry population. However, it yields fascinating results that motivate further research into Nepal's web development security sector using a larger data sample. Examining code, design documentation, and other sources can help to alleviate this constraint. Our findings serve as a foundation for future research that triangulates detailed findings with data gathered from other sources. Further research can focus more on developers' perspectives on web application security. Additionally, to understand the cost and benefit aspect of web application security, researchers can study the cost associated with maintaining web application security and present the benefits of having good web application security policies and practices so that Organizations could use the data and findings to implement that in their Organizations. Our findings from the research can be helpful to researchers and Organizations to understand not only the technical aspects of web security but also the human side of it and its impact. Further, future researchers can employ individuals from various nations, Organizations, educational levels, skills, and backgrounds to produce a broad sample of people.

7. Conclusion

We conducted this study to examine how Organizations in Nepal perceive and practice web application security. We defined the research problem and objective before conducting a thorough literature review. Different sources were used to research how Organizations perceive web application security management and practices in Nepal's web development context. Quantitative and quantitative data analysis was conducted to analyze the different aspects of web application security with reference to OWASP's top 10 vulnerabilities. Nepal seems to be overgrown in web application development in the past decade. People are more and more aware of security in web applications. Although Organizations are more or less aware that security is a significant concern, not enough investment seems to be made to tackle web application vulnerabilities. Our results show that Organizations do not have enough resources to focus on security. Security is considered not a priority unless there is an attack on the applications. Moreover, Organizations perceive security as a technical thing and are not fully aware of the people or the process side, which can be very crucial. We also feel that there has not been enough research on this topic, mainly focusing on least developed countries like Nepal, where systems are very vulnerable. For Nepal, it is more applicable because not all IT experts are fully aware of it. This research can serve as the basis for observing overall web application security in the context of various industries of Nepal and understanding how Organizations from not only the IT field but also other sectors perceive it. It can provide data and findings for Organizations to start conversations about the security of the applications they are developing and look at the people, process, and technology side of things to improve security, tackle potential vulnerabilities and develop secure web applications.

References

Acharya, A. S., Prakash, A., Saxena, P., & Nigam, A. (2013). Sampling: Why and how of it. Indian Journal of Medical Specialties, 4(2), 330-333.

Acharya, S., & Dahal, S. (2021). Security Threats and Legalities with Digitalization in Nepal. Research Nepal Journal Of Development Studies, 4(2), 1-15. doi: 10.3126/rnjds.v4i2.42666

Acronis. (2020). Cybersecurity Assessment Questionnaire. Retrieved from Acronis: https://promo.acronis.com/rs/929-HVV-335/images/Acronis_Cybersecurity_Assessment-Q%26A_full_EN-US.pdf

Ahmad, S., Wasim, S., Irfan, S., Gogoi, S., Srivastava, A., & Farheen, Z. (2019). Qualitative v/s. Quantitative Research- A Summarized Review. Journal Of Evidence Based Medicine And Healthcare, 6(43), 2828-2832. doi: 10.18410/jebmh/2019/587

Alloghani, M. A. M. (2019). An Intelligent Citizen-Centric Oriented Model for Egovernance: A Uae Case Study. Liverpool John Moores University (United Kingdom).

Allwood, C. M. (2011). The distinction between qualitative and quantitative research methods is problematic. Quality & Quantity, 46(5), 1417-1429.

Al-Waisi, Z., & Agyeman, M. (2018). On the Challenges and Opportunities of Smart Meters in Smart Homes and Smart Grids. *Proceedings Of The 2Nd International Symposium On Computer Science And Intelligent Control - ISCSIC '18, 2-6.* doi: 10.1145/3284557.3284561

Alzahrani, A., Johnson, C., & Altamimi, S. (2018, May). Information security policy compliance: Investigating the role of intrinsic motivation towards policy compliance in the organisation. In *2018 4th International Conference on Information Management (ICIM)* (pp. 125-132). IEEE.

Andress, A. (2003). How to Integrate People, Process, and Technology (2nd Edition ed.). New York: Auerbach Publications. doi:https://doi.org/10.1201/9780203501405

Antunes, N., & Vieira, M. (2011). Enhancing penetration testing with attack signatures and interface monitoring for the detection of injection vulnerabilities in web services. In 2011 IEEE international conference on services computing (pp. 104-111). IEEE.

Arhin, K., & Wiredu, G. O. (2018). An Organizational communication approach to information security. *The African Journal of Information Systems*, *10*(4), 1.

Assal, H., & Chiasson, S. (2018). Security in the software development lifecycle. In *Fourteenth symposium on usable privacy and security (SOUPS 2018)* (pp. 281-296).

Atashzar, H., Torkaman, A., Bahrololum, M., & Tadayon, M. (2011). A survey on web application vulnerabilities and countermeasures. Retrieved from https://www.researchgate.net/publication/261389106_A_survey_on_web_application_vulnerabilities_and_countermeasures

Bacchelli, A. and Bird, C., 2013. Expectations, outcomes, and challenges of modern code review. 2013 35th International Conference on Software Engineering (ICSE).

Bell, E., Bryman, A., & Harley, B. (2019). Business Research Methods. United Kingdom: Oxford.

Bhattacharya, D., & Mohalik, R. (2020). Digital mind mapping software: A new horizon in the modern teaching-learning strategy. Journal of Advances in Education and Philosophy, 4(10), 400-406.

Bianco, P., Lewis, G., Merson, P., & Simanta, S. (2011). Architecting Service-Oriented Systems. Retrieved 24 August 2022, from http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9829

Bird, J., & Kim, F. (2014). Survey on application security programs and practices. A SANS Analyst Survey; SANS Institute: Bethesda, MD, USA, 1-24.Narayana, S. (2022). Security Analysis of Web Application for Industrial Internet of Things.

Braun, V., & Clarke, V. (2012). Thematic analysis. American Psychological Association.

Braz, L., & Bacchelli, A. (2022). Software Security during Modern Code Review: The Developer's Perspective. arXiv preprint arXiv:2208.04261.

Brewerton, P. M., & Millward, L. J. (2001). Organizational research methods: A guide for students and researchers. Sage.

Christakis, M., & Bird, C. (2016, August). What developers want and need from program analysis: an empirical study. In Proceedings of the 31st IEEE/ACM international conference on automated software engineering (pp. 332-343).

Cohen, L., Manion, L., & Morrison, K. (2007). Research Methods in Education (6th ed.). Routledge.

Cotenescu, V. M. (2016). People, Process, and Technology; A Blend To Increase An Organization Security Posture. *Scientific Bulletin of Naval Academy*, 19(2), 394-396.

Coutinho, A., & Pinheiro, Y. (2021). A Study of OWASP-Related Security Practices Among Developers. Retrieved 30 August 2022, from http://bib.pucminas.br:8080/pergamumweb/vinculos/00009c/00009ccf.pdf

Creswell, J. W., Plano Clark, V. L., Gutmann, M. L., & Hanson, W. E. (2003). Advanced mixed methods research designs. *Handbook of mixed methods in social and behavioral research*, *209*(240), 209-240.

Creswell, J.W. (2013). Steps in Conducting a Scholarly Mixed Methods Study. DBER Speaker Series. 48.

Cropley, A. (2015). Introduction to Qualitative Research Method. doi: 10.13140/RG.2.1.3095.6888/1

Cruzes, D. S., & Dyba, T. (2011, September). Recommended steps for thematic synthesis in software engineering. In *2011 international symposium on empirical software engineering and measurement* (pp. 275-284). IEEE.

CWE List Version 2.9. (2016). Retrieved 1 June 2022, from https://cwe.mitre.org/data/index.html

Dangol, S., & Kautish, S. (2019). IT SECURITY RELATED ISSUES AND CHALLENGES IN ELECTRONIC PAYMENT SYSTEM IN NEPAL: A STUDY FROM CUSTOMER'S PERSPECTIVE. Research Journal of Science, Technology and Management, 1(2), 85-103.

Debbie Collins (2003). Pretesting survey instruments: An overview of cognitive methods. , 12(3), 229–238. doi:10.1023/a:1023254226592

Deepa, G., & Thilagam, P. S. (2016). Securing web applications from injection and logic vulnerabilities: Approaches and challenges. *Information and Software Technology*, *74*, 160-180.

DeMaio, T. J., & Rothgeb, J. M. (1996). Cognitive interviewing techniques: In the lab and in the field. In N. Schwarz & S. Sudman (Eds.), Answering questions: Methodology for determining cognitive and communicative processes in survey research (pp. 177–195). Jossey-Bass/Wiley.

De Win, B., Scandariato, R., Buyens, K., Grégoire, J., & Joosen, W. (2009). On the secure software development process: CLASP, SDL and Touchpoints compared. *Information and software technology*, *51*(7), 1152-1171.

Dhungana, S. (2019). Hackers steal over Rs 47 million from Agriculture Development Bank in the biggest heist yet. Retrieved from The Kathmandu Post:

https://kathmandupost.com/national/2019/09/26/hackers-steal-over-rs-45-million-from-agriculture-development-bank-in-the-biggest-heist-yet

Dissanayake, N., & Dias, K. (2017). Web-based Applications: Extending the General Perspective of the Service of Web. Researchgate. Retrieved from https://www.researchgate.net/publication/319058851_Web-based_Applications_Extending_the_General_Perspective_of_the_Service_of_Web

Donaldson, S. I., & Grant-Vallone, E. J. (2002). Understanding self-report bias in Organizational behavior research. Journal of business and Psychology, 17(2), 245-260.

Dongol, R., & Chatterjee, J. M. (2019). Robust Security Framework for Mitigating Cyber Threats in Banking Payment System: A Study of Nepal.

Draugalis JR, Coons SJ, Plaza CM. Best practices for survey research reports: a synopsis for authors and reviewers. Am J Pharm Educ 2008;72(1). Article 11.

Elahi, G., Yu, E., Li, T., & Liu, L. (2011, July). Security requirements engineering in the wild: A survey of common practices. In *2011 IEEE 35th Annual Computer Software and Applications Conference* (pp. 314-319). IEEE.

Eminağaoğlu, M., Uçar, E., & Eren, Ş. (2009). The positive outcomes of information security awareness training in companies–A case study. information security technical report, 14(4), 223-229.

Eshete, B., Villafiorita, A., & Weldemariam, K. (2011). Early Detection of Security Misconfiguration Vulnerabilities in Web Applications. 2011 Sixth International Conference On Availability, Reliability And Security. doi: 10.1109/ares.2011.31

Eshete, B., Villafiorita, A., & Weldemariam, K. (2011). Early detection of security misconfiguration vulnerabilities in web applications. In 2011 Sixth International Conference on Availability, Reliability and Security (pp. 169-174). IEEE.

F5. (2022). Cryptographic failures (A2) | Secure against the OWASP Top 10 for 2021. Retrieved 30 August 2022, from https://support.f5.com/csp/article/K00174750

Ferrara, P., Mandal, A., Cortesi, A., & Spoto, F. (2020). Static analysis for discovering IoT vulnerabilities. International Journal On Software Tools For Technology Transfer, 23(1), 71-88. doi: 10.1007/s10009-020-00592-x

Flores, W. R., Antonsen, E., & Ekstedt, M. (2014). Information security knowledge sharing in Organizations: Investigating the effect of behavioral information security governance and national culture. *Computers & security*, *43*, 90-110.

Fortinet. (2021). Mitigating Application Security Threats OWASP Top 10. Retrieved 17 August 2022, from https://www.fortinet.com/content/dam/fortinet/assets/white-papers/wp-OWASP-top-10.pdf

Fowler FJ. Improving survey questions: design and evaluationIn Applied social research methods series, vol. 38. Thousand Oaks, CA: Sage Publications, Incorporated; 1995.

Furnham, A. (1986). Response bias, social desirability and dissimulation. Personality and individual differences, 7(3), 385-400.

Furnham, A. (1986). Response bias, social desirability and dissimulation. Personality and individual differences, 7(3), 385-400.

Galesic, M., & Bosnjak, M. (2009). Effects of questionnaire length on participation and indicators of response quality in a web survey. Public opinion quarterly, 73(2), 349-360.

Gasiba, T. E., Lechner, U., Pinto-Albuquerque, M., & Mendez, D. (2021, May). Is secure coding education in the industry needed? An investigation through a large scale survey. In 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET) (pp. 241-252). IEEE.

Geer, David (2010). Are Companies Actually Using Secure Development Life Cycles?. , 43(6), 12–16. doi:10.1109/mc.2010.159

Ghimire, S. (2017, July 24). Nepal vulnerable to cyber attacks. Retrieved from MyRepublica: https://myrepublica.nagariknetwork.com/mycity/news/nepal-vulnerable-to-cyber-attacks

Gill, P., Stewart, K., Treasure, E., & Chadwick, B. (2008). Methods of data collection in qualitative research: interviews and focus groups. British dental journal, 204(6), 291-295.

Giri, S., & Shakya, S. (2020). High Risk of Cybercrime, Threat, Attack and Future Challenges in Nepal. International Journal of Computer Sciences and Engineering, 8(2), 46-51.

Golafshani, N. (2003). Understanding reliability and validity in qualitative research. The qualitative report, 8(4), 597-607.

Gupta, M. K., Govil, M. C., & Singh, G. (2015). Predicting Cross-Site Scripting (XSS) security vulnerabilities in web applications. In 2015 12th International Joint

Conference on Computer Science and Software Engineering (JCSSE) (pp. 162-167). IEEE.

Guthrie, G. (2019, April 25). *Why effective Organizational communication should be your top priority.* Retrieved from nulab: https://nulab.com/learn/collaboration/why-effective-Organizational-communication-should-be-your-top-priority/

Hancock, B., Ockleford, E., & Windridge, K. (2001). An introduction to qualitative research. London: Trent focus group.

Hasa. (2021, June 18). *What is the Difference Between Explanatory and Exploratory Research*. Retrieved from Pediaa: https://pediaa.com/what-is-the-difference-between-explanatory-and-exploratory-research/

Hassan, M., Ali, M., Bhuiyan, T., Sharif, M., & Biswas, S. (2018). Quantitative assessment on broken access control vulnerability in web applications. In International Conference on Cyber Security and Computer Science 2018.

Hhs. (2022). The OWASP Top 10. Retrieved 20 August 2022, from https://www.hhs.gov/sites/default/files/OWASP-top-10.pdf

Hildum, D. C., & Brown, R. W. (1956). Verbal reinforcement and interviewer bias. The Journal of Abnormal and Social Psychology, 53(1), 108.

Hoffman, A. (2020). Web Application Security: Exploitation and Countermeasures for Modern Web Applications. Sebastopol: O'Reilly Media.

Hoy, W. K., & Adams, C. M. (2015). Quantitative research in education: A primer. Sage Publications.

Huang, Y., & Lee, D. (2005). Web Application Security—Past, Present, and Future. Computer Security In The 21St Century, 183-227. doi: 10.1007/0-387-24006-3_12

Ingalls, S. (2021, May 21). Web Application. Retrieved from webopedia: https://www.webopedia.com/definitions/web-application/

ipa. (2011). How to Secure Your Website. Retrieved 19 August 2022, from https://www.ipa.go.jp/files/000017318.pdf

Jabiyev, B., Mirzaei, O., Kharraz, A., & Kirda, E. (2021). Preventing server-side request forgery attacks. In Proceedings of the 36th Annual ACM Symposium on Applied Computing (pp. 1626-1635).

Jackson, R., Drummond, D., & Camara, S. (2007). What Is Qualitative Research?. Qualitative Research Reports In Communication, 8(1), 21-28. doi: 10.1080/17459430701617879

Jazayeri, M. (2007). Some Trends in Web Application Development. Future of Software Engineering (FOSE '07), 199-213.

Jose, C. R. (2020). *Exploring Security Process Improvements for Integrating Security Tools within a Software Application Development Methodology* (Doctoral dissertation, Colorado Technical University).

Jovanovic, N., Kruegel, C., & Kirda, E. (2010). Static analysis for detecting taint-style vulnerabilities in web applications. Journal Of Computer Security, 18(5), 861-907. doi: 10.3233/jcs-2009-0385

Jovanovic, N., Kruegel, C., & Kirda, E. (2010). Static analysis for detecting taint-style vulnerabilities in web applications. Journal Of Computer Security, 18(5), 861-907. doi: 10.3233/jcs-2009-0385

Kabir, S. (2016). METHODS OF DATA COLLECTION. Researchgate. Retrieved from https://www.researchgate.net/publication/325846997_METHODS_OF_DATA_COLLECTION

Karande, C. (2017). Securing node applications. O'Reilly Media, Inc.

Khatiwada, N. (2021, April 7). Nepali state, Organizations remain highly vulnerable to cyberattacks. Retrieved from The Annapurna Express: https://theannapurnaexpress.com/news/nepali-state-Organizations-remain-highly-vulnerable-to-cyberattacks-3131

Kumar, R. (2011). Mitigating the authentication vulnerabilities in Web applications through security requirements. In 2011 World Congress on Information and Communication Technologies (pp. 1294-1298). IEEE.

Lamsal, R. (2019). Recent ATM Hacking Incident Calls For Effective Monitoring Of CCTV At ATM Booths. Retrieved from Nepal24Hours: https://nepal24hours.com/recent-atm-hacking-incident-calls-for-effective-monitoring-of-cctv-at-atm-booths/

learncisco. (2015). learncisco. Retrieved from Building Blocks of Information Security:https://www.learncisco.net/courses/iins/common-security-threats/information-security-and-common-threats.html

Leite, G., & Albuquerque, A. (2018). An Approach for Reduce Vulnerabilities in Web Information Systems (pp. pp. 86–99). in Proceedings of the Computational Methods in Systems and Software: Springer.

Lepofsky, R. (2014). *The manager's guide to web application security: a concise guide to the weaker side of the web*. Apress.

Lindlof, T. and Taylor, B., 2002. Qualitative communication research methods /cThomas R. Lindlof and Bryan C. Taylor. Thousand Oaks, Calif.: Sage Publications.

Magusiak, D. (2019). Research strategy. Retrieved 25 August 2022, from https://ceopedia.org/index.php/Research_strategy

Maher, Z. A., Shah, A., Chan-dio, S., Mohadis, H. M., & Rahim, N. H. B. A. (2020). Challenges and limitations in secure software development adoption-A qualitative analysis in Malaysian software industry prospect. Indian Journal of Science and Technology, 13(26), 2601-2608.

Mahoney, J. (2010). After KKV: The new methodology of qualitative research. World Politics, 62(1), 120-147.

Malhotra, G. (2022). Strategies in Research. Ijarnd, 2(5). Retrieved from https://www.ijarnd.com/manuscripts/v2i5/V2I5-1220.pdf

Mardisalu R.(2019), "14 Most Alarming Cyber Security Statistics in 2019,"2019. [Online]. Available: https://thebestvpn.com/cyber-securitystatistics-2019/.

Meade, A. W., & Craig, S. B. (2012). Identifying careless responses in survey data. Psychological methods, 17(3), 437.

Medeiros, I., Neves, N., & Correia, M. (2016). Detecting and Removing Web Application Vulnerabilities with Static Analysis and Data Mining. IEEE Transactions On Reliability, 65(1), 54-69. doi: 10.1109/tr.2015.2457411

Mitnick, K. D., & Simon, W. L. (2003). The art of deception: Controlling the human element of security. John Wiley & Sons.

Mohamed, S. F. P., Baharom, F., Deraman, A., Yahaya, J., & Mohd, H. (2016). An exploratory study on secure software practices among software practitioners in Malaysia. *Journal of Telecommunication, Electronic and Computer Engineering*, *8*(8), 39-45.

Monette, D. R., Sullivan, T. J., & DeJong, C. R. (2013). Applied social research: A tool for the human services. Cengage Learning.

Nagpure, S., & Kurkure, S. (2017). Vulnerability assessment and penetration testing of web application. In 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA) (pp. 1-6). IEEE.

NCSC. (2019, February 20). *Secure development and deployment guidance*. Retrieved from National Cyber Security Centre: https://www.ncsc.gov.uk/collection/developers-collection/principles/keep-your-security-knowledge-sharp

NCSC. (2019, February 20). *Secure development is everyone's concern*. Retrieved from National Cyber Security Centre: https://www.ncsc.gov.uk/collection/developers-collection/principles/secure-development-is-everyones-concern

Neves, P., & Eisenberger, R. (2012). Management communication and employee performance: The contribution of perceived Organizational support. Human Performance, 25(5), 452-464.

Onlinekhabar. (2021, March 23). 10 things you should know about cybersecurity in Nepal. Retrieved from Onlinekhabar: https://english.onlinekhabar.com/cybersecurity-in-nepal.html

Onwuegbuzie, A. J., & Leech, N. L. (2007). Validity and qualitative research: An oxymoron?. Quality & quantity, 41(2), 233-249.

OWASP. (2010, November). OWASP Secure Coding Practices-Quick Reference Guide. Retrieved from OWASP: https://OWASP.org/www-project-secure-coding-practices-quick-reference-guide/migrated_content

OWASP. (2021). Who is the OWASP® Foundation? Retrieved from OWASP: https://OWASP.org/

OWASP. (2021a). WSTG - v4.2. Retrieved from OWASP: https://OWASP.org/www-project-web-security-testing-guide/v42/

Palmer, S. (2011). Web application vulnerabilities: Detect, exploit, prevent. Elsevier.

Pandya, D., & Patel, N. J. (2016). OWASP top 10 vulnerability analyses in government websites. *International Journal of Enterprise Computing and Business Systems*, *6*(1).

Parsons K, et al., Determining employee awareness using the Human Aspects of Information Security Questionnaire (HAIS-Q), Computers & Security (2014), http://dx.doi.org/10.1016/j.cose.2013.12.003

Parveen, H., & Showkat, N. (2017). Data Collection. Researchgate. Retrieved from https://www.researchgate.net/publication/319128325_Data_Collection

Positive Technologies. (2022, June 14). Threats and vulnerabilities in web applications 2020–2021. Retrieved from Positive Technologies: https://www.ptsecurity.com/ww-en/analytics/web-vulnerabilities-2020-2021/

Preston, M. (2022, January 27). Web Application Security. Retrieved from CloudDefense: https://www.clouddefense.ai/blog/web-application-security

Rahalkar, S. A. (2016). Certified Ethical Hacker (CEH) Foundation Guide. Maharashtra: Apress.report-50-of-all-web-applications-were-vulnerable-to-attacks-in-2021/

Rao, V. R. (2016). Government Web Application Security: Issues and Challenges-A Case of India.

Republica. (2020, October 9). *eSewa data breached! A hacker releases nearly two dozen eSewa users' details including passwords*. Retrieved from MyRepublica : https://myrepublica.nagariknetwork.com/news/esewa-data-breached-a-hacker-releases-nearly-two-dozen-esewa-users-details-including-passwords/

Rexha, B., Halili, A., Rrmoku, K., & Imeraj, D. (2015, November). Impact of secure programming on web application vulnerabilities. In *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)* (pp. 61-66). IEEE.

Ridgesecurity. (2021). OWASP Top 10 Compliance with RidgeBot® 3.8. Retrieved 30 August 2022, from https://www.bitrate.co.za/wp-content/uploads/2022/03/OWASP-Top-10-7-Dec-2021.pdf

Roopa, S., & Rani, M. S. (2012). Questionnaire designing for a survey. Journal of Indian Orthodontic Society, 46(4_suppl1), 273-277.

Ryen, A. (2004). Kvalitativ intervju. (S. E. Svensson, transl.) Malmö: Liber. (Original work published 2004).

Sahin, M., Ünlü, T., Hébert, C., Shepherd, L. A., Coull, N., & Mc Lean, C. (2022, May). Measuring developers' web security awareness from attack and defense perspectives. In *2022 IEEE Security and Privacy Workshops (SPW)* (pp. 31-43). IEEE.

Santos, D. F. M., & Santos, W. B. (2019). Web Application Failures from Specialist's Point of View: A Qualitative Study.

Sapkota, Pratik. (2021). *Information System Research Methods, Submitted Mandatory Assignment* [Unpublished paper]. Department of Economics, Marketing and Law, University of South-Eastern Norway.

Sarker, S., Xiao, X., & Beaulieu, T. (2013). Guest editorial: Qualitative studies in information systems: A critical review and some guiding principles. MIS quarterly, 37(4), iii-xviii.

Saunders, M. N. K., & Lewis, P. (2011). Doing Research in Business and Management: an essential guide to planning your project (1st ed.). Pearson Education Canada.

SecureCoding. (2021, March 22). OWASP Secure Coding Checklist. Retrieved from SecureCoding: https://www.securecoding.com/blog/OWASP-secure-coding-checklist/#Input_Validation

SecurityForum. (2020). *STANDARD OF GOOD PRACTICE FOR INFORMATION SECURITY 2020*. Retrieved from ISF: https://www.securityforum.org/solutions-and-insights/standard-of-good-practice-for-information-security-2020/

Shinde, P. S., & Ardhapurkar, S. B. (2016). Cyber security analysis using vulnerability assessment and penetration testing. In *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)* (pp. 1-5). IEEE

Shklar, L., & Rosen, R. (2003). Web Application Architecture. England: John Wiley & Sons Ltd.

Silhavy, Radek; Silhavy, Petr; Prokopova, Zdenka (2019). [Advances in Intelligent Systems and Computing] Intelligent Systems in Cybernetics and Automation Control Theory Volume 860 || An Approach for Reduce Vulnerabilities in Web Information Systems., (Chapter 9), 86–99. doi:10.1007/978-3-030-00184-1_9

Singh, R. (2021, March 25). How to identify and mitigate XXE vulnerability?: Indusface Blog. Retrieved October 6, 2022, from https://www.indusface.com/blog/how-to-identify-and-mitigate-xxe-vulnerability/

Siponen, M. T. (2005). An analysis of the traditional IS security approaches: implications for research and practice. *European Journal of Information Systems*, *14*(3), 303-315.

Softwaretestinghelp. (2022). OWASP Top 10 Security Vulnerabilities – How To Mitigate Them. Retrieved 30 August 2022, from https://www.softwaretestinghelp.com/OWASP-top-10-security-vulnerabilities/#5_Broken_Access_Control

Stanton, J., Mastrangelo, P., Stam, K., & Jolton, J. (2004). Behavioral information security: Two end user survey studies of motivation and security practices.

Starrin, B., & Svensson, P. G. (1994). Kvalitativ metod och vetenskapsteori. Studentlitteratur.

Sthapit, Anjeela (2020). Information Security Management, Submitted Mandatory Assignment [Unpublished paper]. Department of Economics, Marketing and Law, University of South-Eastern Norway.

Sullivan, B., & Liu, V. (2011). Web Application Security, A Beginner's Guide. Mcgraw-Hill Education Group. Retrieved from https://dl.acm.org/doi/abs/10.5555/2829333

Synopsys. (2022). What Is Web Application Security and How Does It Work? | Synopsys. Retrieved 3 October 2022, from https://www.synopsys.com/glossary/what-is-web-application-security.html

Sϕhoel, H., Jaatun, M. G., & Boyd, C. (2018, June). OWASP Top 10-Do Startups Care?. In *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)* (pp. 1-8). IEEE.

Tashakkori, A., & Teddlie, C. (2008). Introduction to mixed method and mixed model studies in the social and behavioral science. In V.L. Plano-Clark & J. W. Creswell (Eds.), The mixed methods reader, (pp. 7-26).

Techopedia. (2020). What is a Mobile Application? - Definition from Techopedia. Retrieved 17 May 2022, from https://www.techopedia.com/definition/2953/mobile-application-mobile-app

Terrell, S. (2011). Mixed-methods research methodologies. The Qualitative Report, 17(1), 254-280. Retrieved from http://www.nova.edu/ssss/QR/QR17-1/terrell.pdf

The Human Firewall. (2019). Proactive approach to cyber-security. Retrieved from Twitter: https://twitter.com/cyberriskaware/status/1126777827372695552

Tipton, H. F., & Krause, M. (2007). Information security management handbook. CRC press.

Upadhyaya, P., Shakya, S., & Pokharel, M. (2012). Information security framework for e-government implementation in Nepal. Journal of Emerging Trends in Computing and Information Sciences, 3(7), 1074-1078.

Van Niekerk, J. F., & Von Solms, R. (2010). Information security culture: A management perspective. *Computers & security*, *29*(4), 476-486.

Velasco, R. (2019). Unified Application Security | Hdiv Security. Retrieved 30 February 2022, from https://hdivsecurity.com/OWASP-broken-access-control

Venson, E., Alfayez, R., Gomes, M. M., Figueiredo, R. M., & Boehm, B. (2019, September). The impact of software security practices on development effort: An initial survey. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (pp. 1-12). IEEE.

VentureBeat. (2022, February 21). Report: 50% of all web applications were vulnerable to attacks in 2021. Retrieved from VentureBeat: https://venturebeat.com/security/

Vibhandik, R., & Bose, A. K. (2015). Vulnerability assessment of web applications-a testing approach. In 2015 Forth International Conference on e-Technologies and Networks for Development (ICeND) (pp. 1-6). IEEE.

Vieira, T., & Serrão, C. (2016). Web security in the finance sector. In 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST) (pp. 255-259). IEEE.

Vijayan, J. (2020, September 15). More Cyberattacks in the First Half of 2020 Than in All of 2019. Retrieved from Dark Reading: https://www.darkreading.com/attacks-breaches/more-cyberattacks-in-the-first-half-of-2020-than-in-all-of-2019

Webopedia. (2020, August 7). Input Sanitization. Retrieved from Webopedia: https://www.webopedia.com/definitions/input-sanitization/#:~:text=Input%20sanitization%20is%20a%20cybersecurity,harmful%20codes%20into%20the%20system.

Weiss, R., 1995. Learning from strangers. New York: Free Press.

Wigmo, J., & Wikström, E. (2010). Social media marketing: What role can social media play as a marketing tool? Bachelor dissertation, Linnaus University.

Willberg, M. (2019). Web application security testing with OWASP top 10 framework.

Xie, J., Lipford, H. R., & Chu, B. (2011, September). Why do programmers make security errors?. In 2011 IEEE symposium on visual languages and human-centric computing (VL/HCC) (pp. 161-164). IEEE.

Yergaliyev, A. (2022, April 12). Continuous security testing for an existing client-server application. Retrieved from Hochschule Rhein-Waal: https://opus4.kobv.de/opus4-rhein-waal/frontdoor/index/index/docId/1374

Zhang, B., Li, J., Ren, J., & Huang, G. (2021). Efficiency and Effectiveness of Web Application Vulnerability Detection Approaches: A Review. ACM Computing Surveys, 54(9), 1-35. doi: 10.1145/3474553

Appendix A: Survey Questionnaires

Section 1- About the Survey

The use of web applications is increasing. Almost everything is built as a web-based database application. It can be tough to build a web application as the developer team must have excellent expertise considering the most prevalent security risks that web applications can have. These days, different vital infrastructures are compromised by unreliable applications as the complexity of maintaining system confidentiality rises rapidly. Rouge Hackers can steal cookies, and session tokens, or compromise the database, leading to a massive financial and reputational loss. Many websites do not apply any security policy. Most of those who apply still make mistakes and are still vulnerable to many attacks like SQL injection, Cross-site scripting, etc. Most of the time, the vulnerability exists due to a coding mistake. The flaws in web applications are prevalent, usually easy to exploit and easy to fix. If the company does not find its vulnerabilities first, someone else might find the weakness and hack it.

The survey will take approximately 10-15 minutes to complete.

Anonymity

This survey is completely anonymous, and we guarantee that it is not possible to track answers to you.

Section 2 - Demographic Questionnaire

1. What is your primary role in the Organization?

a. Security Analyst/ Administration
b. Security Manager/ Director
c. IT Manager/ IT Director
d. Network/ System administration
e. Software Developer
f. QA/Tester/Test Manager
g. Penetration Tester
h. Risk Manager
i. Other (Please specify)*

2. How long have you been working in your field?

a. Less than a year
b. 1-2 years

c. 2-3 years
d. 3-4 years
e. 4-5 years
f. More than 5 years

3. What is your Organization’s primary industry?

a. Financial Services/Banking
b. IT company
c. Retail/E-commerce
d. Government Service
e. Data Center
f. Other (Please specify)*

4. How many people work at your Organization, either as employees or consultants?

a. Less than 50
b. 50-100
c. 101-500
d. 501-1000
e. More than 1000

5. What is the size of your web application development team in your Organization?

a. Less than 5
b. 5-10
c. 11-50
d. 51-100
e. More than 100

6. How long has your Organization been practicing web application security?

a. Less than a year
b. 1 - 5 years
c. More than 5 years
d. None but we follow ad hoc
e. None but planning to start a program within the next 12 months
f. None. We don't practice web application security and have no plans to start

7. How frequently do you assess the security of your web applications that are in development?

a. Ongoing/Continuous
b. Every year
c. Every three months
d. Only when applications are updated/patched or changed
e. Once a month
f. Only before systems are initially launched
g. When we sense/ know there's a problem with the applications
h. We don't assess our applications
i. Whenever we remember to check them
j. Other (Please specify)*

8. Where are most of your software development/IT resources being spent?

a. Web Applications
b. Business-Critical applications
c. Mobile Applications

d. Cloud-based Services e. Outsourced Application f. Third-party applications g. Other (Please specify)*
9. Can outside researchers report security flaws in your web application? a. Yes b. No c. Don’t Know

Section 3- General Secure Software Practices

Please rate your level of agreement with the following statements.	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	**Not Applicable**
It is a part developer's job to ensure the security of the application in your Organization						
We practice a secure code training program **Note:** A secure coding training program governs the coding practices, techniques, and decisions developers make while building software. They aim to ensure that developers write code that minimizes security vulnerabilities.						
We integrate security into each phase of the Software Development Life Cycle (SDLC). **Note:** Planning, Requirements, Design, Build, Document, Test, Deploy, and Maintain are the traditional six to eight processes in SDLC.						
Our Organization experienced one or more security breaches because of web application vulnerabilities in the last 12 months **Note**: For instance, Cross Site Scripting, Brute-force attack, Denial Service of attack, Broken access control, etc.						
We are expecting expenditure to change in the next year for our web application security. **Note:** Change in an amount that you are expecting to spend						

Section 4 - OWASP Top 10 Web Application Vulnerabilities

Please rate your level of agreement with the following statements.	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	**Not Applicable**
1. BROKEN ACCESS CONTROL I am familiar with Broken Access Control Vulnerability.						
Our Web applications have a user management system. **Note:** This allows you to assign different responsibilities to different user accounts.						
Most of our web application features are accessible without logging in.						
The web application alerts you if a user types in an incorrect username or password repeatedly.						
The web application checks whether the user is permitted to access information or not. **Note:** To check if a certain user has permission to access the information within the system.						
2. CRYPTOGRAPHIC FAILURES I am familiar with Cryptographic Failures Vulnerability.						
We have recently examined our Secure Socket Layer (SSL) configuration to guarantee that clients are only supplied with secure protocols and ciphers.						
Our web application encrypts data with a powerful cryptographic algorithm.						
The Secure Socket Layer (SSL)/ Transport Layer Security (TLS) private keys on our web servers are properly protected.						
We are cautious enough and have precise controls in place to avoid mixed-content problems. **Note:** TLS (commonly known as SSL) encrypts Internet communication, resulting in a safer surfing experience. TLS-encrypted						

sites are immediately identifiable since their URLs begin with 'https: //' rather than 'http: //'. However, an HTTPS site may contain parts that are loaded via the plaintext HTTP protocol in some cases. This results in a scenario known as mixed content, sometimes known as 'HTTP over HTTPS.'						
3. INJECTION I am familiar with Injection Vulnerability.						
We have enough input sanitization in our web application. **Note:** To avoid the introduction of hazardous codes into the system, input sanitization is a cybersecurity solution that involves verifying, cleaning, and filtering data inputs from users, APIs, and web services of any undesirable characters and strings.						
The program loads active material from third-party servers (i.e., any server that is not under our direct control), such as scripts, applets, or style sheets.						
The web application is capable of processing and manipulating XML provided by the user.						
There is a templating system in your web app that automatically hides all user input before redisplaying it. **Note**: An excellent technique to prevent XSS is to implement a templating system to escape user input.						
4. INSECURE DESIGN I am familiar with Insecure Design Vulnerability.						
We integrate plausibility checks at each tier of our application (from frontend to backend).						
We use unit testing or other similar techniques.						
Our engineers and Quality Assurance team perform potential security issues checks during release testing and are trained to do so.						

Our post-launch monitoring strategy is Robust.						
5. SECURITY MISCONFIGURATION I am familiar with Security Misconfiguration vulnerability.						
We have disabled unnecessary features. **Note**: For instance: unnecessary ports, services, pages, accounts, or privileges						
We have an automated process to verify the effectiveness of the configurations and settings in all environments.						
Default accounts and their passwords are still enabled and unchanged.						
For upgraded systems, the latest security features are disabled or not configured securely.						
6. VULNERABLE AND OUTDATED COMPONENTS I am familiar with Vulnerable and Outdated Components Vulnerability.						
We know the versions of all the client and server-side components that we use.						
We scan vulnerability regularly on a regular basis and subscribe to security bulletins for the components we use. (**Note**: Subscribing to email alerts for security vulnerabilities related to components you use)						
Our Organization administrator deletes outdated and backups files before rolling out the web application.						
7. IDENTIFICATION AND AUTHENTICATION FAILURE I am familiar with Identification And Authentication Failure Vulnerability.						
We have a strong password policy.						

We have a process in place to recover lost and forgotten passwords.						
When a user attempts to access sensitive areas of the application, it checks for authentication.						
We have a strong session ID in our web application. **Note:** A session ID is an identifier that an application uses to store and handle all of the data for a specific user throughout a specific time period in a unique way.						
There is an idle session timeout for the web application.						
8. SOFTWARE AND DATA INTEGRITY FAILURES VULNERABILITY I am familiar with Software And Data Integrity Failures vulnerability.						
We have a code and configuration change review process.						
We ensure that unsigned or unencrypted serialized data is not sent to untrusted clients without some form of integrity check or digital signature.						
We employ secure coding best practices and use the model such as static and dynamic security analysis, CI/CD, and so on.						
We use digital signatures or similar mechanisms to verify the software or data is from the expected source.						
9. SECURITY LOGGING AND MONITORING FAILURES I am familiar with Security Logging And Monitoring Failures vulnerability.						
There are mechanisms in place to notify us in advance when modifications in the production environment are required.						
We have policies in place for data backup and recovery.						

We have a sensitive data log handling policy.						
Our team establishes effective monitoring and alerting such that suspicious activities are detected and responded too quickly.						
We use Penetration Tester processes and techniques to look for security holes on a regular basis.						
10. SERVER-SIDE REQUEST FORGERY (SSRF) I am familiar with "Server-Side Request Forgery (SSRF) Vulnerability".						
We enforce “deny by default” firewall policies or network access control rules to block unwanted traffic from the network layer						
We sanitize and validate all client-supplied input data.						

Section 5 - Feedback

Do you have any feedback to make this survey better?

____

____

____

____

Thank you for your participation!

Appendix B: Interview Questions

About Consent

- Aware about the research project
- Aware about the rights of the participants, including privacy rights
- Aware about the way data is being stored
- Information about the recording

Semi-structured interview questions

Demographic

1. What is your job role? Could you give brief information about your job role?
2. How many years of experience do you have in this scope?
3. Has your Organization experienced one or more security breaches because of web application vulnerabilities in the last 12 months? If yes, what types of attacks have you encountered?

Practice and Perceive

4. Could you briefly introduce what you know about Web Application Security and its management?
5. To what extent does your Organization practice web application security and regard security as an essential factor in web application development?
6. Do you think ensuring security during web application development is challenging for Nepal? If yes, why do you think it is challenging? Which are the challenges you can see for the following years?
7. As threats to applications have increased, developers have begun including security in their software design. Secure development life cycles are methodologies for accomplishing this, but are companies using SDLCs in each phase?

8. Could you briefly explain what should be the key to the testing approaches, techniques, and practices we should follow to ensure the minimum quality and safety in a web application?

9. Do you employ secure coding best practices? If yes, What Secure Code Practices do you apply in the Web application Development Process? How are they applied?

10. Were you instructed by the company you work with or have worked with to adopt secure coding practices? If so, how was it?

11. What is your perspective on ensuring software security during code review? Do you think ensuring security during code reviews is challenging? If yes, why do you think it is challenging?

12. Does a third party develop your web application? If yes, how do you manage Security on Applications written by third parties?

13. Do you use unit testing or other similar techniques? If other, which techniques do you use?

14. What are the policies that you have in your Organization to ensure web application security? For instance, data backup policies.

15. Do you conduct vulnerability scans regularly? If yes, do you use any tools?

16. To what extent do companies support security assessment during web application development?

17. Does your Organization give awareness sessions on web application security vulnerabilities?

Feedback

18. Is there something that you believe is important to add about Web Application Security Management that we haven’t asked you about?

19. Do you have any other feedback/suggestions?