POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Automatic Cybersecurity Risk Analysis

Supervisors:

Prof. Cataldo BASILE

Dott. Ing Gabriele GATTI

Candidate:

Alessandro BIANCO

Academic Year 2023/2024
Torino

Abstract

Since the management of technology has become central to the operations of organizations worldwide, digital assets have also emerged as critical weak points, making it essential for companies to assess cyber threat levels to prevent security breaches and meet requirements from insurers and regulatory bodies. The evaluation of cyber risks, however, is time-consuming, resource-intensive, and requires skilled personnel, comprehensive data gathering and rigorous analysis. Building upon RiskMan, an expert system for automatic assessment of organization cyber-risk, based on open-source tools and publicly available data, this thesis aims to expand the framework through the evaluation and subsequent integration of additional tools, enhancing the process through the use of AI.

The research involves a comprehensive analysis of open-source cybersecurity tools, with particular attention paid to the inputs required and the resulting outputs, followed by the selection of the tool that best suits integration in the existing framework. Additionally, large language models are investigated and incorporated into the workflow, aiming to remove the necessity for human expertise while achieving equally good risk estimations. Finally, the enhanced expert system is evaluated in comparison to the previous version to analyze differences in the produced risk scores.

The resulting framework minimizes the need for expert intervention and provides a more adaptable, intelligent approach to cyber risk management, accessible to organizations with limited budgets and personnel. Through these developments, this research presents a robust, automated solution for cyber risk assessment that brings sophisticated risk management capabilities within reach for a wider range of organizations.

Acknowledgements

I would like to sincerely thank all my friends for their support and for making this journey more enjoyable. To my family, thank you for the constant encouragement you gave me, I couldn't have done this without you.

A special mention goes to my supervisor, Gabriele Gatti, for his constant support, patience, and the time he dedicated to me during the thesis period. I would also like to thank Prof. Basile, who was always available to provide feedback and suggestions whenever I needed.

Thank you to everyone who has been part of this journey.

Table of Contents

List of Figures

Acronyms

AI

Artificial Intelligence

APT

Advanced Persistent Threat

CLIPS

C Language Integrated Production System

CSRF

Cross-Site Request Forgery

DBMS

Database Management System

FISMA

Federal Information Security Modernization Act

GDPR

General Data Protection Regulation

GPL

General Public License

GVM

Greenbone Vulnerability Manager

HIPAA

Health Insurance Portability and Accountability Act

IDS

Intrusion Detection System

ISMS

Information Security Management Systems

LHS

Left Hand Side

LLM

Large Language Model

NIST

National Institute of Standards and Technology

NVD

National Vulnerability Database

OSINT

Open-Source Intelligence

OWASP

Open Web Application Security Project

PDCA

Plan-Do-Check-Act

PII

Personally Identifiable Information

RHS

Right Hand Side

RMF

Risk Management Framework

SDLC

System Development Life Cycle

SME

Medium-Sized Enterprise

SSP

Software Security Project

WAF

Web Application Firewall

xSS

Cross-Site Scripting

ZAP

Zed Attack Proxy

Chapter 1

Introduction

In today's increasingly connected world, cybersecurity has become a core concern for every organization, whether big or small. With growing businesses and individuals getting heavily dependent on digital systems, the number and sophistication of cyber threats grow from affecting large corporations that are well-equipped to handle such issues to smaller and medium-sized enterprises (SMEs) that often lack the resources to deal with these situations effectively. The problem is, while large businesses can spend a lot on cybersecurity, for SMEs protection is often beyond their budget and lack of specialist knowledge. The European Union Agency for Cybersecurity, ENISA[1], has pointed out that recent cyber-attacks involved state-sponsored actors and organized crime, posing a serious threat to critical infrastructure, government agencies and businesses. The growing number of threats make cybersecurity solutions more important than ever, especially for smaller organizations that, unlike larger companies, cannot afford to ignore the issue.

Cyber risk management is at the very heart of this challenge. It denotes the process of identification, prioritization and mitigation of risks to information systems, in order to reduce the exposure the the growing number of threats. However, this is a complex pocess, that requires time and expertise, and for many businesses, especially SMEs, to fully engage with it is rather difficult. Traditional frameworks, like the NIST Risk Management Framework[2], remain a good basic guide for managing these risks. However, these do not go far enough in providing actionable and automated tools that are simple enough for organizations without specialized cybersecurity knowledge to implement.

Because of such challenges, the development of automated solutions for cyber risk assessment needs to be accelerated. This thesis, therefore, focuses on enhancing and adapting an existing expert system, RiskMan[1], to automate the risk assessment process. RiskMan serves as a foundation, combining tools and platforms for information gathering, vulnerability assessment, public databases and even dark web intelligence. These components are further enriched by AI-driven techniques to calculate risk scores. The solution

[1] https://www.enisa.europa.eu/

[2] https://www.nist.gov/

being developed aims to reduce the need for human intervention, enabling businesses, especially those without cybersecurity experts, to manage their risks in a highly reliable, automated way.

Risk assessment is at the core of every cybersecurity effort, which involves identifying, evaluating and prioritizing potential threats to an organization's digital assets. However, most of the current ways to achieve this are either too labour-intensive or require such a level of technical expertise that most SMEs lack. While different tools exist for activities related to vulnerability scanning or information gathering, most of these tools are limited to only one point within a much larger security landscape. That forces organizations to use several tools just to get a somewhat complete view of their cybersecurity risks. Even then, interpretation of the results often requires expertise, further complicating situations for companies without dedicated IT staff. That leaves many SMEs piecing together what appears at best a partial understanding of their vulnerabilities, placing them at greater risk than they may know.

AI can transform the way an organization analyzes and contextualizes the data regarding a given set of vulnerabilities, thus vastly improving the capability for alert correlation and proper determinations regarding the validity of real-world threats. Instead of relying on generic scoring systems, AI could allow organizations to prioritize vulnerabilities according to actual threat levels. In turn, this focused approach would allow organizations to address the most critical issues first and optimize resources in ways that are simply impossible with traditional means.

As threats gain momentum in their complexity and as organizations struggle to manage their resources, which are often not enough to manage security properly, the need for a more integrated solution that unites various cybersecurity tools and amplifies their output through AI is growing. That would, in turn, enable organizations to take the convoluted data and provide clear, actionable insights from it, thus automating the interpretation of vulnerability data and streamlining the entire process of risk assessment. It will even empower companies that are not deeply technical with an easy way to handle their cybersecurity risks and to do it with much less effort.

This thesis is dedicated to the design of a unified expert system that will unify available cybersecurity tools and make use of AI in order to apply a regular, automated process of risk scoring. This system will automatically quantify risks with regard to various integrated tool results, it will give an organization a fuller view of its security situation and thereby enable it to comprehend its risk exposure quickly and efficiently. The system will be validated by comparing the risk assessments provided before and after introducing the AI improvements to demonstrate how the enhancements would improve the overall accuracy and usefulness of the results.

Key contributions in this research have been related to the development of an integrated expert system which automated vulnerability scanning and risk assessment, improvement of vulnerability evaluation by means of AI - in particular large language models - and formalization of a risk scoring calculation method that simplifies how organizations understand their cybersecurity risks. It follows that this thesis, in the form of a comparative study, will aim to prove how the integration of AI can yield better and more accurate assessments that enhance the reliability of risk scores and thus make cybersecurity accessible to SMEs.

This thesis is structured in such a way as to take the reader through the steps of development and validation of the proposed system. The second chapter introduces the reader to the necessary background, including a cybersecurity framework overview, methodologies and existing tools.

In the third chapter, the starting point of this thesis, the RiskMan expert system, is presented in detail, as it has been used as a base for all further works. Along with the expert system, also general information about CLIPS[2] is presented.

Chapter 4 describes the selected tools for evaluation in detail by explaining their functionalities, strong points and weak points. Much attention has been given to the preovided outputs in order to be able to pick the best solution to integrate, with the objective of scoring the output of the tool.

The fifth chapter describes the implementation process of the selected tool and its integration into the existing expert system. This chapter provides details about the implementation choices that have been made to automate and make the execution efficient, along with the integration steps that have been performed.

The sixth chapter focuses on the validation process and is divided into two main sections. The first one describes the AI validation process, in which the AI scoring outputs are validated against known vulnerabilities and scores, presenting the result. The second section presents the comparison between the system's output before and after tool integration and AI-based scoring.

Finally, the last chapter summarizes the contributions of the thesis, discusses limitations, and suggests areas for future research and development in AI-driven cyber risk assessment.

In addressing these objectives, this research hopes to offer a valued solution by enabling organizations, especially those with limited means, to manage cybersecurity risks more effectively and to make cybersecurity risk management more accessible and practical for businesses in general of all scales.

Chapter 2

Background

Understanding the importance and scope of automated cyber risk assessment requires a proper understanding of the general environment in which modern cybersecurity works. This chapter attempts to present a detailed review of the basic cybersecurity frameworks, methodologies and tools, with particular emphasis on their evolving role in managing cyber risks. While doing so, this chapter looks into these fundamental building blocks. It demonstrates how they become a backbone in developing strategies and technologies for mitigating the increasing complexity of cyber threats.

Cybersecurity encompasses a wide range of activities that protect the information system from unauthorised access, disruption, theft, or other forms of damage while ensuring the confidentiality, integrity and availability of sensitive information. In the last two decades, the nature of cyber threats has changed dramatically, tending to increase in their volume and sophistication. Different organisations, together with government agencies, have put forward a variety of frameworks and guidelines that would contribute to battling this growing fight and reducing risks associated with digital infrastructure. These frameworks ensure the provision of consistent methods for identifying, evaluating, and controlling any particular weakness in the cybersecurity posture within broad guidelines.

2.1 Information Security Standards and Guidelines

Information security standards and guidelines[3] are partiularly useful in providing procedures for minimizing cybersecurity risks. They offer organizations frameworks for protecting data and processes, building trust and meeting compliance requirements. These frameworks help put into practice high-level policies and match the security measures towards the organisation's overall goals. According to many standards, it is possible to solve emerging threats in the sphere of cybersecurity as well as ensure interoperability and effective functioning of security measures.

2.1.1 NIST Risk Management Framework

The NIST Risk Management Framework (RMF), developed by the National Institute of Standards and Technology (NIST), provides a comprehensive, systematic process to identify, assess and reduce risks related to information systems. Being one of the most widely recognised frameworks, it makes a structured, step-by-step guide available for any organisation to keep its information systems secure against an evolving spectrum of cyber threats. Unlike many regulatory systems, which tend to be overly detailed and strict, the NIST Risk Management Framework offers a more practical approach by providing flexible guidance that organisations of all sizes, from small businesses to large enterprises, can use to manage risks effectively, even if they have limited specialised cybersecurity staff, helping them break down and address risks in a clear and straightforward way. The goal is to incorporate risk management into the System Development Life Cycle (SDLC) to ensure that risks are continually identified and disposed of throughout a system's design in the beginning, operational phases and eventual decommissioning.

At its core, the basic RMF has six steps: categorise, select, implement, assess, authorise and monitor. However, in practice, most discussions tend to collapse these into four more significant phases of activity: risk framing, risk assessment, risk mitigation and continuous monitoring. Taken together, these phases guide organisations through the multi-layered process of securing their systems in a way that is intended to make risk management continuous and ongoing.

The first step, Risk Framing, sets the stage for the overall risk management process. During this step, an organisation defines its risk profile and needs to consider the boundaries of the system under study, its architecture, constraints and goals, plus the assets that need to be protected. Assets might include hardware, software, data and more intangible assets such as intellectual property and reputational equity. By establishing the risk context, organisations can make informed decisions about the parameters of the risk management process, such as which elements of the system are critical and, therefore, require protection. The output of this phase is a well-defined decision-making context, describing the approach to carry out risk assessments and detailing the techniques that will be applied to manage the risks identified. Risk framing involves establishing the organisation's risk appetite and defining acceptable levels of risk. This process is required to ensure the risk management efforts are aligned with general business objectives and regulatory requirements.

With the framing phase complete, Risk Assessment is arguably one of the most critical steps the organisation will undertake in risk management. This stage involves the organisations' comprehensive process of threat, vulnerability and risk assessment of the system. Risk assessment in cybersecurity refers to the process of analysing unwanted events concerning their likelihood, failure modes and the consequences resulting from such happenings. To conduct this analysis, many organisations first begin with a vulnerability assessment to identify their systems' weaknesses, which malicious actors may leverage. A vulnerability refers to a weakness or flaw that may exist in a security system. It could be leveraged to gain unauthorised access to it, disrupt operational continuity, or destroy information integrity. While risk assessment indeed consists of listing all the identified vulnerabilities, it also includes a deeper analysis of each of them in terms of the likelihood that it would be exploited and the impact of such exploitation. Most of the time, the approach requires identifying various types of threats, ranging from external ones, including malicious hackers, to insider threats, affecting system availability and integrity.

Risk assessment involves reviewing the possible impacts of successful attacks, including financial loss, reputational damage, and legal and operational disruptions. Organisations must evaluate a range of potential scenarios, assess the likelihood of their occurrence, and analyse the severity of their consequences. Through this process, risk assessment identifies risks and assists organisations in prioritising them by discerning which vulnerabilities present the most significant threats, thus necessitating prompt intervention. The culmination of the risk assessment stage is a comprehensive understanding of the organisation's risk posture, which lays the foundation for informed decision-making in later stages.

After identifying and evaluating risks, the next step is Risk Mitigation, where the focus shall fall on the control and minimisation of the recognised risks. This step includes the implementation of security countermeasures and plans designed to eliminate or weaken the impact of possible vulnerabilities. The strategies adopted for risk mitigation may vary widely depending on the nature of the risk and the organisation's available resources and priorities. Standard strategies to mitigate risk include patching software vulnerabilities, hardening the network with firewalls or intrusion detection systems, or encryption of information to avoid access to sensitive data. However, not all risks can be eliminated entirely at all times, organisations must also decide to accept, transfer, or mitigate risks based on their appetite for risk and available resources. The acceptable level could be a low-level risk with very minimal consequences, while a high level of risk in a vulnerability may require immediate action or the distribution of risk through cyber insurance. Risk mitigation requires careful planning and coordination to ensure effective resource allocation and focus on mitigating the highest-value risks. Assigning priority to actions according to the seriousness and likelihood of the risks identified allows an organisation to take specific measures that significantly reduce its overall risk profile.

The final major step, Continuous Monitoring, ensures that risk management is treated as an ongoing, proactive discipline instead of a one-time activity. Because technology continuously evolves, with new threats developed all the time, an organisation's risk landscape can shift very quickly. Ongoing monitoring is essential to maintain an up-to-date understanding of the system's security posture and to determine whether or not security controls implemented in the past remain effective. It includes regularly carrying out vulnerability testing, reviewing risk assessments and the effectiveness of risk mitigation measures. This also involves updating policies, procedures, and security controls to reflect changes to the system's design, legal environment, or the threat landscape. Through the system's ongoing surveillance, organisations can recognise emerging vulnerabilities promptly and mitigate them prior to exploitation, thus sustaining a strong and adaptive defense against cyber threats.

The organisation should be encouraged to document findings, actions, and decisions made during the different phases. For instance, based on the vulnerability assessment done, a report should be well elaborated to describe the nature of the identified vulnerabilities, the level of risk involved in each and suggested mitigation plans. This will serve several purposes: it will provide a clear record for audit and compliance purposes, it will promote communication between technical and non-technical stakeholders, and it will ensure that all measures taken are transparent and open to reevaluation for change, if necessary.

Proper documentation also enhances accountability by making risk management activities transparent and clearly stating roles and responsibilities.

The NIST RMF is, in the end, more than a prescription in one size, but a very dynamic and flexible framework that makes adaptation to needs possible. In fact, it incorporates risk management into the system development lifecycle so that risks are continuously identified, assessed, mitigated, and monitored. It provides a risk-aware culture that includes proactive management, which requires organisations to be vigilant due to continuously evolving threats. RMF can be applied to make informed risk decisions, effective resource allocation, as well as enhance both the security and resilience of information systems.

2.1.2 NIST 800 Series

NIST 800 Series is the collection of documents that set out standards, guidelines and recommended practices concerning cybersecurity in the using information technologies by federal agencies and other civil and commercial entities. Organised by the National Institute of Standards and Technology (NIST), the series includes numerous cybersecurity publications covering everything from risk management and privacy to controls and cryptography, as well as incident response. The purpose of the NIST 800 series is to provide organisations with actionable, well-defined methods for protecting information systems, especially those that are vital to the national security, economic well-being, and operational continuity of critical infrastructure.

The NIST 800 series is one of few that are considered as the most valuable information resources in the field of cybersecurity. The series does not merely serve as a static guideline but is updated regularly to reflect the ever-changing nature of cyber threats and technological advancements. Organisations that apply the NIST 800 series can build robust risk management frameworks, establish adequate security controls and benchmark their cybersecurity programmes against global standards. This makes the NIST 800 series indispensable for entities ranging from government agencies to commercial enterprises aiming to enhance their cybersecurity posture.

Among the numerous documents in the NIST 800 series, three key publications stand out for their particular focus on risk management and security controls: NIST 800-39, NIST 800-30, and NIST 800-53. These publications complement each other to give an extensive, systematic approach to cybersecurity risk management at different organisational tiers and technology tiers.

NIST SP 800-39

NIST SP 800-39 [4] is an excellent strategic guide on carrying out information security risk management across an organisation's structure, which has earned this document a place within most of today's popular cybersecurity frameworks. Its main focus is to help organisations manage risks in a way that makes risk management an integral and recurrent process in the organisational systems that fulfills the purpose of the organisational goals and objectives. The publication is based on the Risk Management Framework (RMF), which has been discussed above. However, NIST 800-39 contains more information, which is outside the sphere of the RMF's activity. It is in fact based on a multi-tiered risk management model, which is aimed at ensuring that risks are managed at various levels of the organisation, as well as four key principles of governance, trust, and trustworthiness.

Most importantly, NIST 800-39 acknowledges the fact that risk management should not be a standalone process that is done in isolation and owned at any one department, but is a business-wide process. The separation into different layers of risk management guarantees that risk will be judged and handled in more than one way by several people depending on their standing in the organisational hierarchy. This hierarchical structure enables organisations to consider risk in its different fashions simultaneously, and achieve consistency in the organisation's strategic objectives set at the organisational level and practice at the system level.

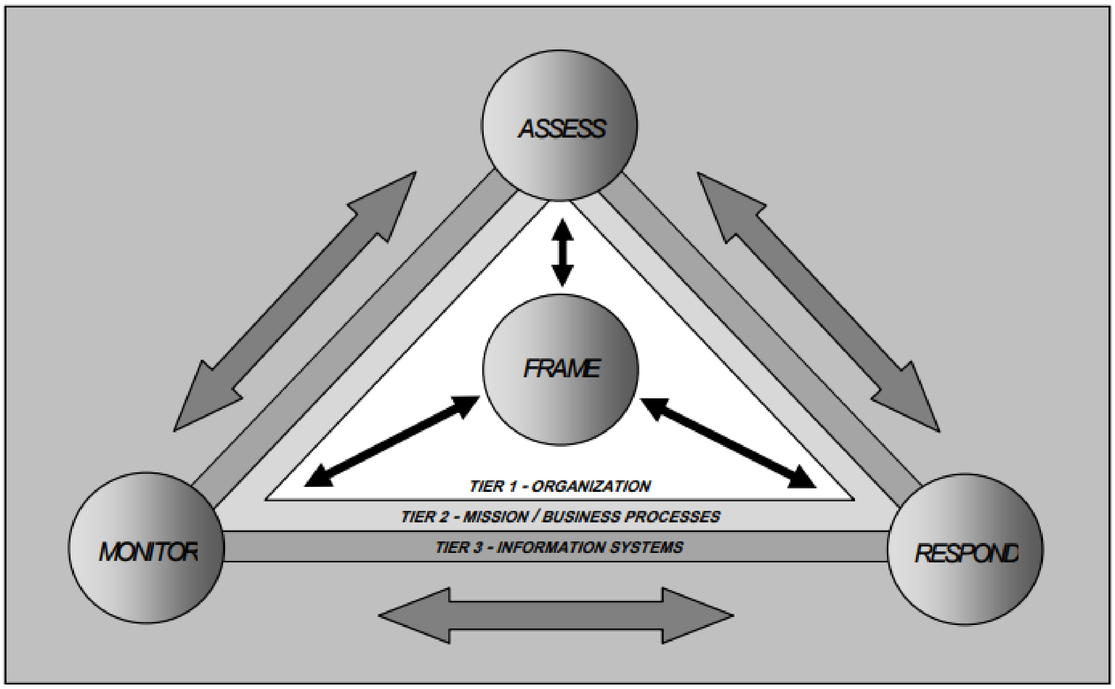
ASSESS
FRAME
TIER 1 - ORGANIZATION
TIER 2 - MISSION / BUSINESS PROCESSES
TIER 3 - INFORMATION SYSTEMS
MONITOR
RESPOND

Figure 2.1: Risk management process applied across the tiers

The first level is known as Tier 1 and addresses risk at the organisational level. Here, risk is handled through setting up governance structures that in turn guide risk management and direction of risk. Many of the duties are vested in Tier 1, such as defining the creation of the Risk Executive function, which is charged with supervising that risk management processes are aligned to the strategic map of the organisation. Another component of the Tier 1 framework is the establishment of the risk management policy, through which an organisation outlines its risk management direction. This strategy is informed by key factors such as the organisation's risk appetite and risk tolerance, as well as the levels of risk the organisation is willing to accept in pursuit of its business objectives. For instance, the amount spent on security resources within an organisation is determined by what has prioritisation at the Tier 1 level to provide the necessary security standards in accordance with the criticality of security assets for an organisation. This top-level framing creates the foundation for the more granular risk management activities that occur at lower tiers.

Tier 2, Business Process View, addresses the business activities and makes certain that these activities are developed focusing on risks. Here, risk management is embedded within enterprise architecture and business processes are designed to comply with strategic goals, security considerations, and risk management goals. This is where the concept of risk-aware business processes is introduced, since the processes themselves are built with certain knowledge of risks they may bring. Implementation tier 2 responsibilities involve infusing information security architecture into the organisation's processes as a way of aspiring to implement enterprise architecture. This framework component ensures that security considerations are an inherent part of the organisation's business processes from the outset, rather than an afterthought. All security risks are addressed at this level: the distribution of security controls and the concrete measures that are used to minimise threats. These controls are based on the strategies that are set at Tier 1 and are meant to offer the right level of protection to the organisational key systems and data resources.

The third and final refined tier is Tier 3, which addresses risk management from an information system perspective. In this case, risk is addressed by detailed prescriptive specifications and continuous monitoring of the security controls on both system and subsystem levels. The activities at this level are highly technical since they involve classifying information systems in terms of their relevance to the organisation and choosing and applying security controls that mitigate the prevailing risks. Tier 3 also involves the assessment and authorisation of these controls to guarantee that the controls work and offer the needed degree of security. Crucially, this tier is informed by Tiers 1 and 2, thus creating alignment between the strategic level and the risk management jobs in the organisation. Third, it consists of ongoing evaluation of the controls' effectiveness over time and their possible modification due to new threats or changes in the organisational conditions. This continuous feedback loop is significant because if, during the operational phase, some lower tiers' vulnerabilities or deficiencies are identified, changes can be made in the upper tiers.

In addition to the multi-tiered risk management approach, this document also emphasises the use of trust and trustworthiness in information systems. In the context of NIST 800-39, trust is defined as confidence in the actions of an individual or an entity whenever that entity is expected to deliver a certain quality of performance in a given context. In nature, trust is a relative benchmark, yet it can be supported by facts, like historical records of the performance of an entity or prior studies, such as security reviews. Trust influences many choices in an organisation but specifically it influences the choices made as to which external systems or partners are to be relied upon and in what way. For example, trust in a third-party service provider might decide on outsourced functions and their key roles in the organisation or integration of third-party technologies into organisational infrastructure.

Trustworthiness, on the other hand, is a more objective measure, referring to the proven ability of an entity to fulfill its responsibilities and perform reliably under specific conditions. Trustworthiness can be demonstrated through compliance with security standards, certifications, and the entity's historical performance record. In NIST 800-39, trustworthiness is not limited to individuals but extends to systems, processes, and even organisations. Ensuring that an entity's trustworthiness aligns with the organisation's acceptable level of risk is critical, in fact deploying systems that are not sufficiently trustworthy can expose the organisation to risks that exceed its tolerance, potentially leading to security breaches or operational failures. Therefore, NIST 800-39 emphasises the importance of evaluating and maintaining the trustworthiness of information systems and the entities that manage or interact with them.

NIST SP 800-30

The NIST Special Publication 800-30 [5], entitled Guide for Conducting Risk Assessments, is a detailed document which presents a methodology for performing risk assessments in organisations, with emphasis on Information Technology systems, as well as Information security risks. Being one of the key modern streams of risk management practices development, most notably in the context of the public sector, it is not limited to the influence on public organisations, in fact private areas are also affected by its impacts as well as international standards for risk assessment and management.

At the outset, the document establishes the importance of risk assessments as a critical element of any organisation's overall risk management strategy. It explains that risk assessments are proactive measures designed to identify, quantify, and prioritise risks that could potentially affect an organisation's operations, assets, individuals and other connected entities. Within the broader Risk Management Framework (RMF) laid out by NIST, risk assessments serve as an integral component in maintaining organisational security. The document clarifies that its guidance is intended for a wide audience, ranging from risk managers, system administrators and security professionals to executives responsible for making decisions concerning information system security.

In order to provide a foundation for understanding risk assessments, NIST SP 800-30 introduces several essential concepts and terminologies, each playing a vital role in the formulation of risk assessments within organisations. To begin with, the guide identifies risk as the possibility of an undesirable result where threats exploit the identified vulnerabilities.

In this context, risk is the intersection of several key components: the presence of vulnerabilities, the likelihood of a threat exploiting those vulnerabilities and the potential impact that such an event might have on the organisation's operations, assets or individuals. By understanding these factors, organisations can evaluate their overall risk profile and develop strategies for addressing the most critical areas of concern. The guide meticulously defines these components (threat, vulnerability, impact and likelihood) and explains how each contributes to the broader understanding of organisational risk.

Threats are identified as circumstances or events with the potential to cause harm, and the document classifies threats into three primary categories: natural, human and environmental risks. Natural threats include factors beyond man's control, including floods, hurricanes, earthquakes or any other natural calamities that may disrupt operations or damage business assets or even endanger human life. On the other hand, human threats are either intentional or unintentional actions performed by a person that may cause damage. Malicious human threats are people who have a negative intent that may lead them to unleashing attacks on an organisations' IT systems, examples include hackers, cybercriminals, employees with a grudge and hostile nations among others since they benefit from the destruction that they cause. Such threats are especially topical as concerns advanced persistent threats (APTs) and complex malware that can heavily impact an organisation's information systems. Non-malicious human threats relate to situations where an employee makes a wrong decision or performs a task incorrectly or makes an error, accidentally deletes a file or configures a system improperly and is equally capable of compromising organisational security or causing data loss or system failure. Last but not least, environmental threats include factors such as power failure, toxic material leaks, fire disasters or any other catastrophes that, while not necessarily intentional, may greatly affect structures, facilities and information technology networks. Analysing each type of threat is critical because threats leverage weaknesses or vulnerabilities in the overall design of the organisation, which may incorporate Information technology and communications systems, security measures, control structures or other business enablers.

In this context, vulnerabilities mean weaknesses that threats can exploit to cause harm. In information system-related threats, exploitation occurs as a result of usage of old software, non-standardised controls, incorrect settings or systems that have not been updated. We see that even while the real situations change, with the development of technologies on one hand and the appearance of new threats on the other hand, the vulnerabilities can also change, which is why improving, preserving and updating the effective safeguard measures is necessary. These weaknesses are not just technical, but organisational weaknesses that open vulnerabilities in the organisation, such as lack of sufficient risk management or generating different decision-making criteria. For example, an organisation may have poor or no security policies that can easily make the organisation vulnerable to security risks that could otherwise be reasonably addressed by standard measures or practices.

Vulnerabilities can also emerge from external relationships, such as dependencies on third-party vendors or suppliers. These relationships may bring vulnerabilities to the organisation because it becomes dependent on some technologies, resources, or services that, if threatened, will further expose the organisation to risk. Furthermore, a lack of well-implemented business processes may not enhance risk awareness components, which may increase exposure to threats or lack effective ways of managing these risks. The other important factor is the organisation's enterprise architecture design so as to contain the likely threats that may occur. If an organisation has a robust approach to information system protection, a weakness at an architectural level can compromise the robustness of deployed systems, leaving them prone to a cascade failure or a focused attack. Thus, the problem of maintaining an enterprise architecture that is flexible and built in accordance with certain standards is one of the ways of minimising the level of said risk.

Regarding threats, organisations also recognise predisposing conditions necessary when discussing vulnerabilities. Predisposing conditions are factors which make an event either more or less likely to be attacked by a threat by virtue of its vulnerability. These conditions are not the vulnerabilities, because on their own they do not dictate the likelihood of vendors being targeted, however they do define under what conditions vulnerabilities may become more or less likely targets. For example, a predisposing condition could be location that may expose an organisation to specific natural disasters or the use of old hardware which has fewer protection mechanisms than current ones. The States that are experiencing these types of conditions may have existing vulnerabilities that these problems amplify or new vulnerabilities that the organisation has to weigh in its risk profile.

Impact is another critical concept outlined in NIST SP 800-30. Impact refers to the degree of harm that would result from the successful exploitation of a vulnerability by a threat. The guide emphasises that impacts can be multi-dimensional, affecting not only the confidentiality, integrity and availability of information systems but also the organisation's reputation, legal standing and operational continuity. Confidentiality impact arises when the data is in the wrong hands, which can lead to identity theft, loss of intellectual property or breaches of privacy regulations. In the case of data, integrity concerns are realised through staleness or manipulations in a way that produces unadaptable or erroneous information. Availability threats occur when service or information is unavailable to the permitted consumers and which hampers organisational operations or services.

In addition to these technical consequences, the guide also reviews financial, regulatory and reputation loss associated with the risk events. A data breach, for instance, would lead to massive financial losses, fines, legal costs or the costs of remedial action that would be required in the event of the incident. Moreover, the regulatory bodies may bring fines if the organisation cuts compliance with legal requirements like the General Data Protection Regulation (GDPR) or the Health Insurance Portability and Accountability Act (HIPAA). Finally, reputational risk may arise when the customer or other stakeholders withdraw their support in respect of business, stock or belief due to the failure of the organisation to protect their data.

Likelihood refers to the probability of a threat exploiting a vulnerability within the organisation. NIST SP 800-30 defines likelihood as a function of two primary factors: the presence of threat actors and the organisation's existing safeguards. Estimating likelihood involves analysing how frequently threat actors may attempt to exploit vulnerabilities based on historical data, threat intelligence and predictive models. In some cases, the likelihood of a threat can be high due to the presence of active adversaries or well-known vulnerabilities, while in other cases, it may be lower because of robust security controls and a lack of adversarial interest. Likelihood can also fluctuate based on the effectiveness of the organisation's defences, such as firewalls, intrusion detection systems and employee training programs. For example, an organisation that has implemented multi-factor authentication may reduce the likelihood of certain attacks, such as phishing, by making it more difficult for attackers to gain unauthorised access. For the same reason, organisations with inadequate defenses or untrained staff may face a higher likelihood of successful exploitation.

In order to assess the level of risk that an organisation needs to face, it is necessary to understand the balance between likelihood and impact. Risk that has likelihood and impact at a high level is a high risk and needs the most attention in prevention, while the risk with low likelihood and impact is classified as low risk and does not require much attention in prevention even if it requires much resources.

Finally, NIST SP 800-30 expands the notion of an acceptable level of risk that an organisation is willing to accept while achieving its goals, the risk tolerance. The tolerance to risk is not identical in all organisations due to the differences between the field of business, the regulating laws, the available capital and the vision of operations. For example, a financial institution handling sensitive customer data may have a very low-risk tolerance, requiring stringent security measures to protect that data. In contrast, a startup company with limited resources may have a higher risk tolerance, accepting certain risks in order to focus on growth and innovation. Determining the organisation's risk tolerance is a critical step in risk management because it guides decision-making regarding which risks to mitigate, which to transfer (e.g., through insurance) and which to accept.

With the concepts regarded as the basis of risk assessment identified, it is now possible to describe the process as seen through the prism of the standard. According to NIST SP 800-30, the risk assessment process is structured into four distinct phases: preparation, execution, communication and upkeep.

The preparation phase sets the stage for the entire risk assessment process. During this phase, the organisation defines the risk assessment's purpose, scope and parameters. This involves identifying the specific systems, processes or assets to be evaluated, as well as determining the assessment's objectives and the risk criteria to be used. In this stage, one also needs to ensure that the main stakeholders in the assessment process are identified before the assessment takes place and that enough resources are provided. Further, during the preparation of the risk management plan, organisations have to decide on the following about the assessment method and tools to be used, whether qualitative, quantitative or a hybrid approach. The preparation phase defines what is to be achieved in the next stages of the assessment and acquaints all the stakeholders with their part to play in the whole process.

The execution phase comes right after preparation is done. This phase is all about gathering and analysing data to generate risk factors. This concept refers to an in-depth review of the organisation's structures, its weaknesses, potential threats and the existing form of control measures. This data collection is usually done through the following techniques: interviews, documents review, system scan and vulnerability assessment. In the execution stage, the gathered data is employed to measure the probability of a threat event occurrence and the potential loss that a certain event will bring to the organisation. This estimation is carried out by rating the risks and also considering the effectiveness of controls in the organisation in protecting it from attacks. The deliverable of the execution phase is, therefore, a list of developed risks and a categorisation of the identified risks based on their likelihood and impact. With the help of this analysis, it becomes possible to determine priorities of risks in order to pay most attention to dangerous threats.

The communication phase is crucial since it enables dissemination of the risk assessment findings to all the organisation's pertinent stakeholders. The clarity in the message or recommendation given by a technologist in a simple manner reminds the decision-maker of the information given. In this phase, information gathered during the risk identification and assessment stages is put down in writing in the form of a report that contains results of the assessment, suggestions for management or reduction of risk as well as an explanation of the rationale for the ranking of the risks. Some of the outputs for the phase may include presentations to top management, discussions with other technical departments, and consultations with outside agencies such as regulatory bodies or partners. It is thus important that all the stakeholders are kept informed and in tune with the assessment results for the required change to be effected in the required risk management culture of the organisation.

The final phase of the risk assessment process is upkeep. Risk assessments are not one-time events, they must be revisited and updated regularly to account for changes in the organisation's environment, technology, and threat landscape. The upkeep phase involves continuous monitoring of the risk environment, ensuring that new threats, vulnerabilities or changes in business processes are identified and integrated into the risk management strategy. The update also occurs when major changes are noticed in the organisation like acceptance of new technologies, a merged company or even a new regulation implemented. Therefore, each organisation should update its risk assessment to ensure that the management of risks is not only reactive, but proactive as well.

NIST SP 800-53

NIST Special Publication 800-53 [6], known as "Secure and Privacy Controls for Federal Information Systems and Organizations", is one of the most popular publications within the NIST 800 series because of its extensive and systematic concepts of security and privacy control. While this publication was primarily designed to support federal agencies in response to the Federal Information Security Modernization Act (FISMA)[1], it has evolved beyond that simple purpose to become the universal standard for any organisation that wants to protect its information systems from numerous types of cyber threats. Since this framework is an initial document, it has an essential function not merely in informing federal entities but also organisations of the private sector, both commercial and state, including those in heavily secured sectors such as healthcare, finance and energy, that would like to conform to federal norms or implement exemplary cybersecurity management systems.

The controls presented in NIST 800-53 are arranged in a set of families, each of which relates to a particular domain of information security and privacy. These families span a very wide array of security areas and are divided between the technical and procedural aspects of the domain. For instance, the Access Control family deals with the aspects of who can access what information and when while the System and Communications Protection deals with how to protect the data's confidentiality, integrity and availability as it passes through systems and communications networks. Other families include Incident Response and Contingency Planning, as the first of them emphasises the procedural aspect of security, that is the necessity of clear and consistent reaction to security incidents. The second, the strategic one, focuses on preparing the organisation to continue its operations after encountering some disruption. In each family, NIST 800-53 provides a definition

[1] https://www.cisa.gov/topics/cyber-threats-and-advisories/federal-information-security-modernization-act

and narrative of individual controls that can be enacted by an organisation, explaining not only what each individual control is but also the role or purpose of the control in relation to meeting security objectives. These controls can be implemented in different organisational environments and aligned with other models, thus creating a clear risk management structure.

One of the key features that sets NIST 800-53 apart from other cybersecurity frameworks is the concept of control baselines, which are predefined sets of controls categorised into three levels: low, moderate and high. These basic categories are grounded on the extent to which a security threat can affect the functioning of an organisation, its assets and individuals. As the overall level of the baseline is higher, the number and the extent of the necessary controls increases. For instance, an organisation that processes and stores its data at high levels of sensitivity or is located in a higher risk environment, for example, a government contractor processing data classified as sensitive, would be implementing the high baseline of controls, that ranges from multifactor authentication to data encryption, both at rest and in transit, to continuous monitoring and detailed log keeping and analysis. On the other hand, organisations that process less sensitive information or operate in relatively risk-free environments are likely to choose the low or moderate reference points that give much simpler, less invasive control implementations.

Another strength is the fact that the majority of the NIST 800-53 framework is highly adaptive and can be tailored to fit specific risk profiles, operational needs and regulatory environments. The flexibility of the situational analysis is very important because not all organisations are exposed to the same threats and not all organisations have the same means to protect themselves. Hence, even though the publication offers a clear and comprehensive list of controls, it prompts organisations to consider their categorical risk elements, the value of the data they process, the nature of the IT environments they operate under as well as the threats that they actually face, and apply these controls as necessary. This customisation also applies to the privacy controls inherent in NIST 800-53, wherein policies and procedures on how to address special personal data in relation to lawful and proper requirements are set. The proposed framework has been developed in such a way that security controls are matched with privacy objectives so that the organisations can introduce privacy measures into the general concept of security with more emphasis given to sectors such as healthcare, finance and telecommunications, whereby handling of personally identifiable information (PII) is a critical issue.

In addition to the initial identification and deployment of controls, NIST 800-53 also insists on a continuous monitoring approach since cybersecurity and privacy controls should be proactive and responsive to organisational and technical changes. To this end, the publication describes procedures for ongoing observation and evaluation of controls that have been put in place. It is constantly advised that organisations should conduct periodic assessments of their security needs and modify their controls correspondingly. This proactive, lifecycle-based approach to cybersecurity management is crucial to make certain that security controls don't become useless or vulnerable to new threats. Therefore, NIST 800-53 encourages organisations to look at cybersecurity as a continuous project rather than a singular fix of a certain vulnerability and not have to worry about anything else again.

A final key feature of NIST 800-53 is its focus on newly identified security risks with a focus on privacy. The integration of privacy controls as a sub-set of the overall security concept is explained by the understanding that overall information security cannot be achieved without the corresponding protection of the privacy of the individuals whose data is being processed. NIST 800-53 provides a clear list of specific privacy controls to assist organisations in meeting various countries' privacy laws, such as HIPAA in health care or GDPR in Europe, and also to address what is considered acceptable and ethical use of people's information. These controls help organisations set up prompt and proper privacy policies and privacy Data Impact Assessments and ensure that the organisation's data is collected, used and shared without infringing the basic fair, accountable and transparent way. The fact that NIST 800-53 has both a security and privacy slant means that it is a holistic solution to the protection of information and the privacy of people.

2.1.3 ISO/IEC 27000 Series

The ISO/IEC 27000 series[2] is a family of standards that has grown to be a reference guide for organisations on information security management. The goal of the standard is to help organisations protect their stored and processed information by providing a systematic approach to establishing, implementing, maintaining, and further developing an information security management system (ISMS). These standards offer internationally accepted best practices for the protection of several classes of information, such as financial data, Intellectual property and employee records, third-party information and others. As the number and density of threats in digital environments continue to increase rapidly, the ISO/IEC 27000 series gives a structured, risk management-based method of protecting information assets in various industries.

The most notable standards within the ISO/IEC 27000 family include ISO/IEC 27001, ISO/IEC 27002 and ISO/IEC 27005. These each make up one of the standards that individually addresses a specific aspect in the overall system of information security management and is intended to fill a specific role in an organisation's risk based security management system. ISO/IEC 27001 prescribes the requirements for an ISMS, ISO/IEC 27002 supports recommendations for the implementation of security controls in a context of an ISMS and ISO/IEC 27005 provides detailed methods for managing information security risk.

ISO/IEC 27001: Information Security Management System (ISMS)

ISO/IEC 27001 is the first of a group of standards and guides put forth in the ISO/IEC 27000 series, and consists of the framework for implementing and maintaining an Information Security Management System (ISMS). Like all management system standards, ISO/IEC 27001 specifies the measures and procedures to be implemented in order to protect information and determine the risks associated with information security. ISO/IEC 27001

[2] https://www.iso.org/standard/iso-iec-27000-family

is intended to provide organisations with the framework for implementing a risk management approach to maintain confidentiality, integrity and availability of their information.

Fundamentally, ISO/IEC 27001 demands that an organisation establish and implement an information security risk assessment process that seeks to identify all risks and potential threats to the organisation's information assets. This risk-based approach ensures that security measures are prioritised and aligned with the organisation's specific needs and risk tolerance. After the risk assessment has been done, organisations should put in place measures and precautions to tackle the identified risks. The standard does not prescribe specific controls, but instead provides a flexible framework that can be tailored to each organisation's unique context and requirements.

Continual improvement as a concept is one of ISO/IEC 27001's primary pillars. Organisations are required to develop an ISMS, but they also have to continuously establish, implement, review, and monitor the security processes. This process is also known as the Plan-Do-Check-Act (PDCA) cycle, which ensures that the ISMS remains effective and responsive to emerging threats and vulnerabilities. ISO/IEC 27001, like other ISO standards on management systems, is also a certifiable standard to which organisations can undergo formal audits in order to demonstrate compliance with the standard's requirements.

ISO/IEC 27002: Code of Practice for Information Security Controls

ISO/IEC 27002 is an information security code of practice, which is complementary to the ISO/IEC 27001 standard providing detailed guidance regarding the selection and application of certain types of information security controls to support the ISMS created under ISO/IEC 27001. Whereas ISO/IEC 27001 provides guidelines for a management system for information security, ISO/IEC 27002 provides implementation guides through security controls to address the risk evaluation findings. Of course, ISO/IEC 27002 is indeed not the certifiable standard, it acts more like the code of practice that assists the organisations in creating and maintaining their security policies and strategies.

ISO/IEC 27002's advice is divided into fourteen domains representing important parts of information security. Among these, it is possible to find Asset management, Access control, Cryptography and Incident management. In these domains, ISO/IEC 27002 offers guidelines on the most effective means of applying security measures from a technical and organisational perspective. For each control, the standard points out its purpose, and it gives recommendations on how it should be applied. This makes ISO/IEC 27002 an effective guideline for organisations which want to align their security efforts with the existing international standards.

The most significant part of ISO/IEC 27002 is its affiliation with ISO/IEC 27001. ISO/IEC 27001 mandates organisations to set up an ISMS enabled through the identification and management of risks but it does not propose which control should be applied. It only points to ISO/IEC 27002 as the guide to choose the right controls to implement in an organisation. This is why, by specifying the technical and operational controls, ISO/IEC 27002 completes the framework offered by ISO/IEC 27001 on managing information security risks. Therefore, ISO/IEC 27002 is a logical and useful companion to ISO/IEC 27001 because it provides implementable interpretations of the organisational requirements set out in the management system standard.

ISO/IEC 27005: Information Security Risk Management

ISO/IEC 27005 is available as a standalone document on the risk management aspect of information security and provides very practical information on how organisations can systematically approach information security risks. Although ISO/IEC 27001 mandates organisations to carry out risk assessments as part of their ISMS, ISO/IEC 27005 essentially describes how best to perform those assessments systematically. With help of risk management, considered in ISO/IEC 27005, an organisation can make the right decisions about which security controls have to be used and how the resources can be distributed effectively.

The overall concept of risk management, as described in ISO/IEC 27005, is based on several activities, such as risk identification, analysis, evaluation and treatment. An organisation must define its information assets, analyse the risks that can affect those assets and estimate the probability and consequences of a security breach. Consequently, organisations can order their risks and determine suitable risk treatment measures that may involve risk reduction, transfer, elimination or acceptance. It is intended to ensure security issues respond sensibly to its risk management and organisational goals.

ISO/IEC 27005 is directly associated with and built to support ISO/IEC 27001 and ISO/IEC 27002. It complements ISO/IEC 27001 by offering extensive guidelines on how to perform some of the risk assessments that form part of the ISMS. At the very same time, it aligns with the ISO/IEC 27002 and assists organisations in determining which controls from the catalogue set out in the 27002 should be applied to counteract particular risk situations. Thus, ISO/IEC 27005 has a significant function of establishing linkage between the risk management process and overall ISMS, so that the security controls that are selected are not random but are grounded on risk identification and assessment process.

2.2 Risk Management Frameworks

Risk management frameworks (RMFs) are essential tools that organizations use to comprehensively address more complex and frequent risks in the modern environment. These frameworks offer structured methodologies for identifying threats, appraising risks and deploying measures for managing threats in relation to achievable safety goals. Through clear and straightforward guidelines touching on risk evaluation and response, RMFs assist organizations in channeling their efforts where it would most matter, on the various assets most vulnerable to certain risks.

A strong RMF incorporates risk management into organisational culture, keeps risk assessment continuous, and continually modifies the strategies that are approached regarding numerous emerging risks. This approach is crucial, especially in today's ever-growing technology, which is squared by increased advanced cyber threats. Finally, RMFs also improve the degree of compliance with regulations and standards and contribute to the decision-making process due to the usage of the frameworks to determine the effect of risks on business activities. In this regard, organizations can achieve both operational functionality and security objectives that strengthen protection and create confidence.

In addition, RMFs are not just static models since they undergo changes after some time to adapt to new threats and technological changes. Dynamic frameworks also allow for ongoing assessment so that changes can be made to an organization's plan and it can remain vigilant against possible exposures. This helps make RMFs useful in managing risks from innovations such as cloud computing, IoT and artificial intelligence, which complicate the risk environment[7][8].

2.2.1 OCTAVE Risk Management Framework

The OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) framework has been designed by the CERT Division of the Software Engineering Institute at Carnegie Mellon University as a facility to encourage an asset-focused, self-contained risk analysis method that assists certain organisations in identifying, evaluating and subsequently managing risks that are related to information assets and supporting infrastructure. OCTAVE stands apart from other frameworks by emphasising a collaborative, organisational focus in which stakeholders from across departments work together to identify security priorities that align closely with the organisation's broader operational objectives. This approach makes it possible to adapt OCTAVE for use in different organisations of different sizes and cybersecurity maturity levels, especially for those with a limited budget. By integrating risk management into operational planning, OCTAVE ensures that security efforts are closely aligned with mission-critical functions.

The OCTAVE framework is divided into three primary phases: Build Enterprise-Wide Security Requirements, Identify Infrastructure Vulnerabilities and Determine Security Risk Management Strategy. These phases altogether define an ongoing and integrated process aimed at continuous changes as the organisational risks emerge.

The first phase, Build Enterprise-Wide Security Requirements, directs organisations to identify the organisation's assets and determine the type of threat each asset may face. This phase centres mainly on evaluating assets which will support or add worth to the organisation's objectives. This may include physical commodities, data or network systems, intellectual properties, customers' trust and organisational reputation. Once these assets have been identified, stakeholders follow the process of constructing a range of threat profiles. This is necessary because information and other assets in organisations have different security needs in terms of confidentiality, integrity and availability. By clearly identifying and documenting these needs, security can be better designed and implemented with operational objectives in mind. The result of this phase is an identification of organisational assets and the risks that surround them in an organisational context, which supports building the subsequent focus on systematic risk management in the later phases.

Building on the insights gained from asset identification, the second phase of the framework, Identify Infrastructure Vulnerabilities, changes the perspective from the organisation's technical and operational infrastructure to explore their vulnerabilities. This phase involves mapping the critical assets identified in the previous phase to the supporting infrastructure components, including hardware, networks, software systems and operational processes. After this mapping is done, OCTAVE suggests carrying out a detailed vulnerability analysis in order to establish specific areas of weakness regarding these components and thus expose the valuable assets to the threats that may be present out there. Software and administrative weaknesses, such as uninstalled software updates, improper configurations, or incorrectly set access controls, are prioritised according to the level of danger they pose to organisational operations. By focusing on the technical infrastructure supporting key assets, OCTAVE provides a nuanced view of the organisation's security posture, highlighting where targeted improvements are needed to protect operational integrity and continuity.

The third phase, Determine Security Risk Management Strategy, is dedicated to formulating actionable risk mitigation strategies. In this phase, the organisation calculates the likelihood of being exploited for each of the risk factors identified and the potential effect on the assets that the organisation regards as crucial. This assessment process results in a list organised by importance in terms of risk, which means that an organisation can invest resources to manage the most threatening issues. Based on the highest risk rating, OCTAVE asks an organisation to create an exclusive set of mitigation measures that might be technical, such as patches and system hardening, or procedural, such as revisions in the policies of access control, or even organisational, such as the intensification of training for employees. The framework recognises that not all risks can be eliminated totally and helps organisations make informed choices about where to tolerate, transfer, or reduce risks, depending on their risk tolerance level and organisational goals and objectives. These strategies, combined with documentation for every process, result in a systematic security plan that outlines the roles, responsibilities and time frame for undertaking each measure that would otherwise be considered a risk to the organisation's operations.

The three phases presented here offer risk managers a systematic way of identifying, ranking and responding to risks in a cyclic fashion. To help organisations stay vigilant and strong, OCTAVE focuses on continuous risk assessment and reconsideration of their risk and guard measures as necessary. This includes vulnerability assessment, threat review and assessment of security measures, so that organisational security can proactively adapt to changes in the organisation environment or threat data. This continuous review process also helps the organisation maintain a flexible and sustainable security architecture that is capable of countering new and emerging threats.

Documentation is a valuable part of the OCTAVE methodology, which implies documenting every step, from asset identification to the assessment of vulnerability and choosing mitigation measures to apply. This documentation serves multiple purposes: it provides a clear record for auditing and compliance, enhances communication among technical and non-technical stakeholders and promotes a culture of accountability within the organisation. Furthermore, consistent documentation can be disclosed periodically to review the identified risks and the previously applied solutions, which can be enhanced if needed. In this way, OCTAVE provides accurate documentation of risk management practices that are accurate, logical, and consistent with both operational and security goals.

OCTAVE focuses on assets and prioritising qualitative risk assessment, so it is most suitable for organisations that aim to have a highly customisable approach that can be maintained within an organisation while engaging all departments. At the same time, OCTAVE has some certain drawbacks, one of which is that it is not quantitative and can hardly provide detailed calculations of financial risks for an organisation, although it can be a good basis for organisations that decide to incorporate security planning into their further operational strategies flexibly. Finally, OCTAVE helps organisations build a risk-aware culture to incorporate cybersecurity into an organisation's everyday operations and effectively risk manage as a core organisational activity consistent with business strategy. Because OCTAVE mainly aims to identify critical assets, engage stakeholders and continuously refine risk management solutions, it offers a structured and effective approach to risk management which helps organisations sustain a flexible and robust security regime.

2.2.2 FAIR Risk Management Framework

The FAIR (Factor Analysis of Information Risk) framework[3], developed by the FAIR Institute, offers a structured and quantitative approach to understanding and evaluating information risk. The major difference between the FAIR model and many other conventional models of qualitative risk assessment is that, unlike other approaches, it focuses on delivering specific financial measurements of risk by quantifying the probable frequency and impact of potential loss events. This makes it possible for organisations to prioritise risk mitigation regarding financial loss, making FAIR more appropriate for organisations that want to incorporate risk management into their business planning processes.

The FAIR framework consists of four main stages, that allow risk professionals to quantify risk effectively: Identify Risk Scenario Components, Evaluate Loss Event Frequency, Assess Probable Loss Magnitude and Derive and Interpret Risk Analysis Results. Together, these stages produce a structured and repeatable process that helps align cybersecurity and risk management efforts with business objectives.

The first stage, known as Identify Risk Scenario Components, involves identifying the assets at risk coupled with the possible threat sources and organisational weaknesses pertaining to every asset. In FAIR, a risk scenario includes an identified service, plus the type of occurrence that would cause a loss. FAIR also prescribes a manner by which probable threat agents, such as external people or events, internal people or processes and environmental factors, are matched to the organisation's valuable assets to fit the scope in a real-world context of each identified asset. It lays down the risk scenarios for all the subsequent phases and sets the frequency and impact assessment parameters.

In the second stage, Evaluate Loss Event Frequency, FAIR examines how often a given threat might realistically affect an asset, or, in other words, the frequency of potential loss events. This step breaks down frequency into two distinct components: Threat Event Frequency and Vulnerability. Threat Event Frequency measures how often a threat actor might attempt to compromise an asset, while Vulnerability assesses the probability of

[3] https://www.fairinstitute.org/fair-risk-management

the threat actor's success in exploiting any weaknesses. For instance, while a phishing attack might occur frequently, the likelihood of it successfully impacting a well-trained employee base would be lower. By calculating both components, FAIR allows organisations to understand the likelihood of materialising each scenario and prioritise resources towards high-frequency and high-risk areas.

The third stage, Assess Probable Loss Magnitude, shifts the focus in understanding the impact of successful threats on an organisation's assets. FAIR divides loss magnitude into Primary Loss and Secondary Loss factors. Primary Loss includes direct effects, such as operational disruptions, regulatory penalties or immediate financial losses. Secondary Loss covers indirect impacts, such as reputational damage, customer loss and legal repercussions. By estimating both direct and indirect losses, FAIR equips organisations with a comprehensive view of the potential economic damage associated with different risk scenarios, thereby supporting informed decision-making on risk mitigation.

Finally, the fourth stage, Derive and Interpret Risk Analysis Results, uses the findings from the previous stages to quantify risk in financial terms. This stage includes producing a final risk estimate for each scenario, typically using a risk exposure range or confidence intervals. Additionally, FAIR encourages sensitivity analysis to understand how changes in certain variables, such as the frequency of threat events or the estimated costs of loss, could affect overall risk estimates. The results from this stage provide clear, actionable insights, enabling risk managers to compare risks based on their financial implications, prioritise risks with higher potential losses and optimise cybersecurity investments based on their potential return in reducing organisational risk.

A key advantage of FAIR is its emphasis on documentation and model transparency. By clearly documenting each step, from scenario identification to loss magnitude assessment, FAIR ensures that the risk analysis process remains consistent, repeatable and transparent. This transparency allows organisations to audit their risk management practices, track the effectiveness of implemented controls and foster an organisational culture that values proactive risk management.

Although FAIR provides a powerful quantitative risk assessment model, it requires access to reliable data and may involve complex calculations. This can make FAIR more resource-intensive compared to qualitative models, particularly for smaller organisations with limited data or analytical capabilities. However, FAIR offers a robust and financially-focused approach to evaluating and mitigating risks for organisations with the necessary data infrastructure and a desire to align risk management with business objectives.

Overall, the FAIR framework offers a structured approach to information risk management that helps organisations identify, quantify, and prioritise risks based on their potential economic impact. This approach makes it easier for decision-makers to justify security investments, allocate resources effectively, and ensure that risk management aligns with broader business goals and financial priorities.

2.3 Vulnerability Management Tools

Vulnerability management tools are cornerstones of contemporary IT security solutions, helping organizations conduct structured detection, evaluation and remediation of threats in their information systems. These tools work by performing a reconnaissance of systems, applications, and networks, then analysing results while presenting a broad database of security risks for classification and ranking purposes. Their work is not limited to identification, they help organizations assess the likelihood of risks stemming from vulnerabilities, helps prioritize the order of patching and helps address these problems effectively. As a result, such tools assist organizations in minimizing the extent of vulnerability and improving on general security.

The value of vulnerability management lies in its ability to align technical scanning with broader organizational objectives. There are also other tools such as Nessus, Netsparker and Greenbone Vulnerability Manager that come with reports with results of vulnerability assessment and even a hint on how to manage them. These tools allow constant supervision as new weaknesses, which can be added through new releases or changing threat vectors, are quickly detected. Further, by automating such features as identification and reporting vulnerabilities, operations efficiencies are dramatically raised, thereby allowing security personnel to concentrate on critical threats.

2.3.1 Greenbone Vulnerability Manager

Greenbone Vulnerability Manager (GVM)[4] is one of the most known open-source solutions for vulnerability scanning and management, containing a range of tools intended for vulnerability assessment and reporting of an organisation's network. Being a key process of risk assessment, vulnerability management defines a pivotal aspect of the steps necessary to find the weaknesses in the infrastructure susceptible to being used by a threat agent and, therefore, is an important part of the reduction procedures used to protect information systems. GVM is a further development of the OpenVAS project, an Open Vulnerability Assessment Scanner that became a powerful and flexible tool to detect security weaknesses in various platforms and environments for organisations wanting to automatise the process.

In the context of automated risk assessment, systems like GVM are highly beneficial, disrupting the manual effectuation of vulnerability discoveries to a significant extent. GVM uses several scanning techniques and is able to detect weaknesses in operating systems, applications and networks. This idea can be closely associated with the objectives of automatic risk assessment, where the primary goal is to collect as much relevant security data as possible without necessarily constant human supervision. Through integration with the Greenbone Community Feed, a vulnerability database that the GVM updates frequently, GVM is able to identify the said range of known vulnerabilities, meaning that its scans remain ever-optimised and relevant against the current security threats.

One of the organisational strengths of this solution is that it not only automates the

[4] https://www.greenbone.net/en/

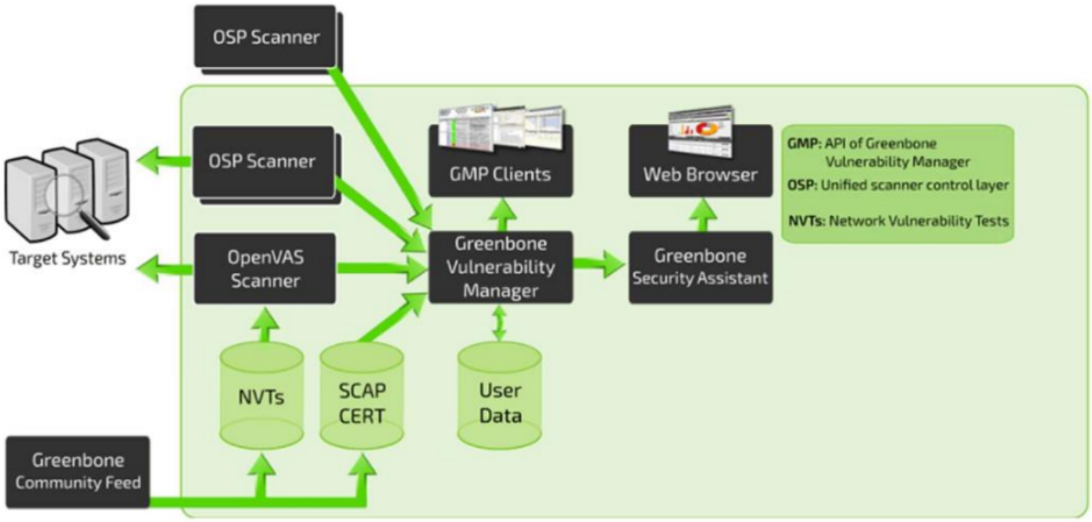
OSP Scanner
OSP Scanner
Target Systems
OpenVAS Scanner
GMP Clients
Greenbone Vulnerability Manager
Web Browser
Greenbone Security Assistant
GMP: API of Greenbone Vulnerability Manager
OSP: Unified scanner control layer
NVTs: Network Vulnerability Tests
NVTs
SCAP CERT
User Data
Greenbone Community Feed

Figure 2.2: GVM schema

vulnerability scanning process but also reports a ranking of the vulnerabilities. After threats have been identified, GVM produces comprehensive reports, managed and customisable to meet the needs of various audiences of an organisation. These reports offer durations with IT-related specifics and other brief executive summaries for decision-makers, thus enriching a choice of the coming action regarding remediation. This ability to automatically report these findings is especially useful for entities who may not have the capital or trained personnel to analyse these intricate vulnerability details individually, as it eliminates the need for an organisation to interpret a difficult scheme of vulnerabilities and enhance the security weaknesses' resolution speed.

Globally, the role of GVM is not only to serve as an automatic risk assessment tool, but it is broader than just being a vulnerability scanner. The components of GVM's architecture provide seamless integration with other security tools and the formation of a great security framework. This capability to interact with intrusion detection systems, Firewalls, and other security infrastructure components makes GVM a preferred option for incorporation into larger, automated Frameworks for cyber risk management. In this regard, GVM is not only used for the identification of vulnerabilities but also as a tool in a larger framework for constant scans and security management. Its modularity also complements the support for external plugins, making the tool flexible and able to be adjusted by organisations depending on the specific risk levels and organisational operations.

In addition, GVM complies with the current established cybersecurity standards, including the NIST SP 800-53 and the international ISO/IEC 27001, which address the effectiveness of vulnerability scanning and risk handling in an organisation's security. So, with GVM, organisations can prove compliance with such universally adopted standards, because GVM helps perform vulnerability identification and assessment on a regular basis, which is one of the critical steps of risk assessment in compliance with the described frameworks. The integration of vulnerability management with these conventional cybersecurity frameworks underscores the fact that the process of vulnerability management is unending, especially due to new forms of threats that are being created in different parts of the world.

Regarding the development of an automatic risk assessment tool, the automation part of GVM has shown the possibility of integrating vulnerability detection in a comprehensive risk management environment. A holistic risk evaluation environment implies the ability to perform vulnerability assessment, risk rating and prioritisation of the remediation efforts, and as such, GVM provides several features to address these needs. For example, its reporting capabilities allow risk scores, that correspond to the severity of vulnerability, to be generated and then used in a grand risk management framework that informs other organisational responses. Such integration of vulnerability detection into risk assessment goes hand in hand with the enhancement of risk evaluations and guarantees the most recent risk scores.

The importance of GVM in an automated risk assessment process could further be highlighted based on its support for regular updates and continuous scans, which is vital for the best security posture. The smooth scheduling of the scans guarantees that the vulnerabilities that are present in an organisation's IT systems are well identified and addressed appropriately, unlike a situation where the number of unresolved security threats keeps piling up and increasing the vulnerability of an organisation to a cyber threat. Using GVM as an example, this continuous scanning approach coupled with the automated reporting feature of GVM make it an influential tool in the organisations automation and streamlined compliance to cybersecurity risk management.

2.3.2 Netsparker Vulnerability Manager

Netsparker Vulnerability Manager[5] is a popular solution for automated web application security that provides powerful features for effective vulnerability scan and management. Built with the intention of vulnerability identification and remediation in web environments, Netsparker fully automates the checking, evaluating, and reporting of various web-based threats, services, and API endpoints. This functionality is highly useful for organisations that hope to maintain a strategic approach to risk management as it not only helps foreshadow threats that have a high potential of being leveraged by malicious actors, but also avoid them. Netsparker's advanced scanning system, which utilises both dynamic and interactive analysis techniques, is extremely useful in identifying various complex vulnerabilities, facilitating extensive risk analysis on which organisations depend on in enhancing their security positions.

In the context of automated risk management, specific Netsparker attributes facilitate vulnerability identification, requiring minimal human interaction and shortening the time taken for risk assessment. With heuristic scanning complemented by selective vulnerability checks, Netsparker quickly detects security threats while avoiding false positives that often

[5] https://www.invicti.com

overwhelm the security team, reducing their effectiveness. Netsparker's advanced heuristic engine targets new vulnerabilities and routine updates to catch known vulnerabilities at a particular time. This hybrid approach provides comprehensive security that adapts to evolving threats and helps organisations maintain strong protection for their web applications.

One of the most interesting features of Netsparker is its reporting and integration feature. It provides rich reports that may be presented in specific forms for different audiences in an organisation, including a technical view for IT departments and a summary view for strategic decision-makers. It strengthens the interconnection of teams and facilitates evidence-based remedial action decisions, enabling organisations without extensive security experience to understand their security gaps for remedial action. The automated reporting feature of Netsparker is especially beneficial for organisations looking to streamline their vulnerability management process without dedicating substantial resources to manual analysis.

In addition to stimulating vulnerability detection, Netsparker links to numerous security and development tools to allow vulnerability management in broader risk and compliance systems. When integrated with industry standards as OWASP, ISO/IEC 27001 and NIST SP 800-53, Netsparker assists organisations to achieve compliance with these recognised frameworks and re-emphasises the criticality of vulnerability management in the overall cybersecurity plan. As such it is ideal for organisations that require a flexible security tool that can be adjusted to current security policies ad risk management strategies.

Netsparker's ability to automatically scan for vulnerabilities and its continuous scanning support is particularly important in keeping an organisation's map of vulnerabilities up to date. By scheduling regular scans and automating both vulnerability detection and reporting, Netsparker helps prevent the accumulation of unaddressed security risks, enabling a proactive approach to managing potential threats. These capabilities demonstrate that the tool is important in today's cybersecurity world, allowing organisations to maintain robust protection against emerging threats in an efficient and cost-effective manner.

2.3.3 Nessus

Nessus[6] is a powerful vulnerability scanner developed by Tenable that addresses the need of organisations to detect vulnerabilities and risks in their information systems more effectively. In other words, the primary application of Nessus is to provide comprehensive scans that identify vulnerabilities with operating systems, applications and network devices to help organisations assess and control risks before they potentially lead to system breaches and compliance with industry best practices for Information Technology.

Nessus has numerous advantages, and one of the key ones is that it comprises a large and constantly updated vulnerability database through which Nessus will be able to identify a huge number of known vulnerabilities, including newly uncovered ones. Such dynamic update ensures that the scans are up-to-date and effectively protect the organisation

[6] https://www.tenable.com/products/nessus

against emerging cyber threats, thereby improving their protection. Thirdly, Nessus offers various types of scans and credentialed scans, which are getting deeper into the possible risks, thus providing highly accurate and actionable assessments.

Another strength of Nessus is that it carries out reporting and analysis by providing reports of discovered risks, categorising them according to their importance and the extent of their impact at the company. The fix is especially valuable to decision-makers, providing clear, concise summaries alongside technical details and facilitating informed remediation strategies tailored to the organisation's risk profile. The capability of scheduling and filtering reports to suit various stakeholders significantly increases the usefulness of Nessus as a technical tool in the security context within a conceptual framework of security management.

Nessus also states compatibility with other security products, making including them in a current security framework easy. This modular approach allows the construction of complex security structures, including firewalls, IDS and numerous other solutions. That integration is critical for automating processes, overseeing, and constantly improving the security profile.

Moreover, Nessus aligns with industry standards such as the NIST Cybersecurity Framework and the ISO/IEC 27001, demonstrating its capability to assist organisations in achieving compliance with regulatory requirements. Nessus provides the opportunity for timely evaluation, proposing tools to manage risks, thus becoming a valuable asset for an organisation that wants to meet its cybersecurity obligations. This results in the fact that vulnerability management is cyclical since organisations are constantly threatened by the emergence of new threats and vulnerabilities in their environments.

Nessus has enhanced the process flow of vulnerability management through advanced automated features, allowing organisations to arrange their automated scans to periodically scan for security vulnerabilities and remedies. It not only helps to address the risk in the case of organisations that have not been diligent in fixing existing and emerging vulnerabilities but also compliments the overall cybersecurity posture of organisations. Thus, the synergy of detailed vulnerability identification with efficient risk evaluation measures makes Nessus indispensable in the context of present global lines of automated security defence systems and renders it possible to manage modern tendencies of cyber threats with reasonable certainty.

Chapter 3

RiskMan

As cybersecurity threats continue to increase in frequency and sophistication, the demand for efficient, scalable systems capable of assessing and managing cyber risks has grown exponentially. Cyber risk assessment, an essential aspect of cybersecurity, mainly demands specialized resources, technical knowledge and human resources, which SMEs cannot afford due to limited budget. To begin to solve these challenges, Gatti, Basile and Perboli suggested RiskMan, an expert system for cyber risk assessment that should be relatively autonomous, requiring very little intervention from human operators and limited cyber security training, which served as a basis for this thesis work. RiskMan is thus designed to work independently with limited input from human operators, thus making it suitable for organizations with limited cybersecurity personnel.

RiskMan employs a combination of rule-based expert system principles alongside AI enhancements to assess cyber risks by computing a risk score derived from a broad range of public and open-source information-gathering tools. This design makes automated risk assessment feasible even for organizations constrained by limited resources. Building upon RiskMan's framework, this thesis investigates and extends its methodologies to provide a scalable, accessible model for cyber risk evaluation.

In addition to presenting RiskMan as a substantial advancement in automated cybersecurity management, this chapter delves into the theoretical foundations and technical implementation of RiskMan's architecture. With a particular emphasis on the expert system logic achieved through the C Language Integrated Production System (CLIPS) framework, this chapter examines the system's structured approach to data gathering, risk scoring and the application of AI techniques to manage incomplete data. Following an introduction to the core functionalities of the CLIPS framework and its forward-chaining capabilities, the chapter explores the detailed architecture and workflow of RiskMan, discussing how these elements have influenced the design and development of this thesis work.

3.1 CLIPS

CLIPS stands for C Language Integrated Production System and is a sophisticated and very flexible expert system shell designed by NASA to help implement rule-based expert systems. CLIPS was built with a forward-chaining inference engine to let developers define and organize intricate decision-making processes through a rule-based protocol capable of self-executing as will based upon changing conditions within the knowledge base. These controls offer great flexibility in terms of logical progression. For this reason, it enjoys excellent acceptance in fields such as robotics, space technology and cybersecurity that require far-reaching automation and adaptive reasoning. The fact that CLIPS is highly modular, provides comprehensive coverage of most common expert system requirements, and is equipped with a very powerful and efficient internal pattern matcher makes the RiskMan system very suitable for CLIPS's workings.

In the context of CLIPS, the knowledge base streamlines and stores all data invocations that CLIPS will use to infer rules. Every fact in CLIPS is one piece of information that the system contains in its knowledge base and every rule defines specific conditions for when certain actions should be triggered. In CLIPS, facts may be asserted into the knowledge base, which adds new data to the base, or may be retracted from the knowledge base, which deletes data from the base to be updated when new information is in or during changes of conditions. Moreover, this dynamic updating of the knowledge base enables the system to update its responses for assessment of the cyber risk, which seeks information that may be relevant or may have changes in newly stated threats or vulnerabilities.

One central feature that defines CLIPS is its forward-chaining inference model, which evaluates rules in real time as new facts are asserted into the knowledge base. The forward-chaining model based on IF-THEN statements fires the rule when the condition of the IF part of the statement is true in the knowledge base. Once the rule is fired, the THEN part (action) takes place. Although action can be as simple as creating a new fact, changing an existing one, invoking an external API or firing other rules, this subsequent triggering of rules helps ensure that CLIPS can string together various rule activations into a comprehensive sequence, which in turn helps design far more elaborate logical sequences from even quite subordinate rule designations. In RiskMan, using this forward-chaining model, the system can determine from new data points which inference rules to fire continually and keep the system updated on the current state of the cybersecurity environment.

In CLIPS, rules are presented in a format that describes the circumstances under which a rule will be triggered, as follows. Rules contain conditions, also known as the LHS (Left Hand Side), and actions, also referred to as the RHS (Right Hand Side). The LHS of a rule belongs to the concept of pattern-matching statements and may contain, for example, logical operators, conditional tests and references to specific facts. These conditions can be combined using boolean operations to set up complex conditions for the rule to get activated; in this way, precise control over the time when some actions should be performed is achieved. The RHS of the rule also defines what action is going to be performed when the LHS conditions become true. These actions may be to assert or retract facts, change the knowledge base, call of external procedures and even call other rules. This format of structuring the system provides a fine-grained level of control over how and what data is processed within RiskMan, a system for making cyber risk assessments; as such, the criteria used to filter data sources and the analyses performed depend on the precise stage of the process.

The RETE algorithm is a basic feature that upgrades CLIPS' capacity to deal with large rules and data. This efficient pattern-matching algorithm is the essence of CLIPS, regulating the rule evaluation process, and minimizing the need for extra computations that could slow it, even with large massive rule sets. This is accomplished by caching selected results of rule evaluations, avoiding duplicated calculations, and thus allowing high rule throughput despite an ever-growing knowledge base. In particular, RETE constructs a network of nodes indicating conditions within rules and connecting them in a way that makes it possible to assess shared conditions multiple times. This makes it unnecessary to continuously check the same conditions for different rules, an enhancement that highly benefits rule-based systems like RiskMan that operate on real-time information gathering. As in any working environment, the CLIPS' RETE algorithm enables RiskMan to effectively respond to situations in which data is processed and updated continuously while maintaining the ability to deliver cyber risk assessments in real time.

CLIPS is highly modular, this goes beyond the internal rule structure of CLIPS as the language comes with built-in facilities to interact with other tools and libraries, something employed heavily within the RiskMan architecture. Due to CLIPS' flexibility, it can interact with other APIs, command-line utilities, and databases used by RiskMan to gather cybersecurity data. For example, whenever RiskMan scrutinizes an unfamiliar point of data ingress in a network as, for instance, an IP address or domain name, the application of CLIPS gives it the ability to initiate routine data harvesting from vulnerability mapping utilities, threat intelligence knowledge bases or net analysis software. All these and the subsequent interactions take place through the use of rule-triggered commands that call up external routines, query the knowledge base for data pertinent to the interaction, and update the knowledge base to conform to the result of the interaction. This means that integrating with external sources is perhaps one of the essential aspects for RiskMan because it helps the system gather countless data points regarding cybersecurity without intervention from the employees.

Additionally, CLIPS provides a templating system that enhances data management by allowing developers to define specific data types (or templates) that structure how information is stored within the knowledge base. They favourably describe the properties of data types, thus enabling easy retrieval of structured data and uniformity of data all through the inference processes. As in many other similar information management systems, templates are used within RiskMan to define data categories for different types of cyber threat information, including IPs, vulnerabilities, and services. In this way, CLIPS allows RiskMan to use particular inference rules for different forms of presented data, putting them into predefined templates that increase both the precision and time needed for the assessment.

Furthermore, CLIPS support hierarchical activation where the rules can be put under a preferred activation level to be triggered. This prioritization is administered by rule salience, which is a function that provides distinct priority to the rules, thereby regulating the order of activation of the rules. In RiskMan, salience makes sure that the rules containing information on prioritizing particular class of actions, e.g., those related to the collection of crucial data or identified high-risk vulnerabilities, will be completed before the rules with the lower priority actions, enhancing in this way the effective collection and analysis of the data in accordance to the present threats to cybersecurity. This capability is relevant when applying rules in a multistage process as it is in RiskMan where subsequent stages may depend on the data calculated at earlier stages.

According to the previous descriptions of functionalities, CLIPS provides a powerful and adaptable framework for implementing the rule-based logic and structured data management required by RiskMan's cyber risk assessment architecture. By leveraging its forward-chaining inference engine, RETE algorithm for optimized rule processing, modular API integration and flexible templating system, CLIPS enables RiskMan to conduct continuous and comprehensive cybersecurity evaluations autonomously. The high degree of automation supported by CLIPS reduces reliance on human expertise. It ensures that RiskMan can operate effectively across diverse network environments and data contexts, making it a robust and practical solution for cyber risk assessment in resource-constrained organizations.

3.2 RiskMan Tool Integration and Data Sources

While traditional risk assessment involves manual processing of potentially inconsistent and sometimes inconclusive sources, RiskMan relies on cybersecurity tools and public data sources to identify, evaluate, and position the risk factors. These tools range from well-known vulnerability scanners to specific open-source intelligence (OSINT) platforms, which enter highly valuable information for a complete risk assessment. The integration of the RiskMan framework with these tools is designed so that it can make context-based and precise assessments by updating the knowledge base containing the details of contemporary vulnerabilities, network exposure, and past data breaches. In what follows, the reader is presented with a detailed outline of the key instruments and databases that underlie RiskMan's fully integrated assessment procedure.

RiskMan can authenticate and extract data from the target index including sources such as Shodan and LeakIX, indexed data from the National Vulnerability Database (NVD), and IntelX through API integration. Simply, Shodan will search through all addresses that are connected to the internet in a certain percentage and what kind of devices or services are available from it, along with the geographic location of exposed and disclosed assets. RiskMan can also ask Shodan what is potentially vulnerable within the organization's network because these devices are publicly exposed.

LeakIX[1] is one of the helpful services for receiving information regarding the services that are available in open access related to the specified IP addresses. RiskMan uses the LeakIX API to determine running services on exposed IPs, such as open ports, used

[1] https://leakix.net/

protocol, and detected service/software product and service version. This information helps RiskMan decide which network services are available for outside connections, which improves the system's knowledge of the dangers of exposing objects through publicly connected infrastructure.

The National Vulnerability Database (NVD)[2] is another essential source for RiskMan. It provides a comprehensive repository of known vulnerabilities, complete with detailed descriptions and severity ratings. By integrating NVD data, RiskMan can enrich its vulnerability assessments, aligning discovered vulnerabilities with recognized security threats and their potential impact on the organization's infrastructure.

RiskMan uses IntelX[3], an OSINT tool working with information gathered from various sources such as the dark web, data leaks, and locked forums. This integration enables RiskMan to determine its extent of exposure and classify areas and where the target appears, in leaked or darknet data or in sensitive or high-risk categories. Based on the amounts of these exposures, RiskMan was able to incorporate a risk amplifier that can include high-risk, unconventional sources to their assessment for a given organisation, thereby providing an extra layer of insight on the overall exposure of the organisation in limited or prohibited platforms.

RiskMan utilizes Nmap[4], an open-source network scanning application that is used for specific identification of services and host fingerprinting. Nmap allows the system to scan the architecture of the network and find out the working services and the ports on the organization's open IP addresses. This capability is very important in determining the prospective threats of attack and discovering areas of misconfiguration or vulnerability within a network.

Also, RiskMan contains TheHarvester[5], a tool that is specifically used for subdomains. TheHarvester is helpful in collecting information concerning subdomains connected with the target organization to reveal unperceived resources that may be indispensable in cyberattacks. This data is more important when it comes to portraying the organization's internet presence and risk profile.

The second part of the assessment within RiskMan checks whether the system assigns the correct individual scores to all the added data included in the knowledge base. This includes scoring elements such as IP addresses, domains, services and all identified vulnerabilities. RiskMan has experimented with the ability of the author's implementation to produce aggregate scores regarding services (IP and port-related) and machines (IP-related) to produce a risk index for the target organization. The scoring mechanism ensures that risk assessments are not only accurate but also reflective of the organization's specific context and exposure factors.

By integrating these tools and leveraging their capabilities, RiskMan offers a powerful

[2] https://nvd.nist.gov/

[3] https://intelx.io/

[4] https://nmap.org/

[5] https://github.com/laramies/theHarvester

and fully automated solution to the problem of quantitative cyber risk assessment, which synthesizes numerous data sources relevant to cybersecurity to compute a precise risk score. This approach of structuring RiskMan provides elasticity in handling levels of data measures incompleteness while providing value in failing scenarios.

Chapter 4

Tool Selection

This chapter focuses on selected tools, where a functional description of each of them is given, during which capabilities of every single tool are evaluated to determine how appropriate and necessary it might be in the system's overall architecture. The following sections present each evaluated tool individually, providing a comprehensive overview of its functionalities.

4.1 SQLMap

The main objective of SQLMap[1] is to detect and exploit SQL injection vulnerabilities within web applications. Once SQL injection points are identified on a target host, users have the ability to perform extensive back-end database management operations. These include retrieving session user details and database names, enumerating users and their privileges, dumping entire databases or specific tables, executing SQL queries, and reading selected files from the file system. This level of access makes SQLMap a powerful tool for understanding the vulnerabilities of a database system.

Effectiveness, Autonomy, Aggressiveness, and Configurability

Since SQLMap is programmed to detect and fully leverage the technique of SQL injection in an application's front end, it is extremely reliable as a tool for penetration testing. While this tool can often manage detection and exploitation independently, it may occasionally prompt the user to enter data manually when making specific requests. However, besides these prompts, SQLMap has a batch option which lets the tool work with default parameters without requesting user input.

As for the aggressiveness of SQLMap, it works most often in the safe mode so as not to harm the target system. However, it can be set to attack mode to provide a more

[1] https://sqlmap.org/

powerful SQL injection test. Several options allow for controlling test aggressiveness to tune the testing process to achieve defined goals. Moreover, SQLMap is highly configurable, and users can adjust nearly everything, from connection retrieval to detailed detection or amplification of particular SQL injection methods. For instance, with this tool, testers can alert the tool to crawl URLs to the specified depth, scan and exploit HTML forms, define the number of concurrent requests for fast testing, and set the risk and overall testing depth levels preferred by testers. Moreover, there is another option in SQLMap to check for an active Web Application Firewall (WAF), which makes this tool even more powerful.

Techniques

SQLMap supports the detection and exploitation of five primary types of SQL injection techniques, each providing a unique approach to extracting data from the target database system.

In the Boolean-based blind technique, SQLMap can infer output by appending syntactically valid SQL statements to the affected parameter in an HTTP request. By comparing HTTP response headers and body to that of the HTTP request, the tool is able to guess the content of the injected statement on a character-by-character basis. The bisection algorithm included in the tool optimizes the rate of operations by acquiring every instance of a character in fewer than seven HTTP requests. An adaptation of the algorithm applies to non-plain-text types if needed.

In the time-based blind technique, SQLmap inserts a delay function in a real SQL statement that is included in the targeted parameter. By observing the delay in the response time, SQLMap infers the output of the injected SQL statement. This method also applies the bisection algorithm for character-by-character extraction to obtain data that other means may not yield.

In the error-based technique, SQLMap injects a complete SQL statement in the parameter, which produces an error message depending on the type of DBMS. By analyzing the context of the HTTP response of DBMS error messages, which contains the result of the SQL statement, SQLMap can extract data if the web application shows detailed error messages.

The UNION query-based technique is employed where, from the web application, the output of the SELECT statement is returned. SQLMap adds the UNION ALL SELECT statement to a targeted affected parameter, and if the application structure loops through each line of results, data can be printed in the page content. SQLMap also supports single-entry union-based injection for those applications where the target application displays only one record.

The stacked queries, or piggybacking technique, check if the web application accepts multiple SQL statements in one query. SQLMap can also append another SQL statement after a semicolon, making it possible to run more SQL statements, depending on the ability of the target application to handle them. It enables the testers to execute either data definition or manipulation commands which, in most situations, would grant the request of file system access or even operating system command execution depending on the privileges of a particular DBMS.

By supporting these five techniques, SQLMap provides penetration testers with a versatile and powerful toolset to comprehensively evaluate SQL injection vulnerabilities, ensuring a robust database and system security testing approach.

4.2 OWASP ZAP

The Open Web Application Security Project (OWASP) Zed Attack Proxy (ZAP)[2] is a free and open-source penetration testing tool that is currently supported under the Software Security Project (SSP). ZAP is aimed at finding vulnerabilities in web applications, it is both versatile and modular, making it suitable for users of varying skill levels. ZAP operates as a man-in-the-middle proxy, enabling it to monitor, inspect, modify, and forward messages from a client browser to a web application in a way that is capable of testing and analyzing traffic between the two points. ZAP is equally useful as a stand-alone application as it is as a daemon process and can be configured to integrate with existing network proxies, which is advantageous given the use of proxies in many companies today.

Furthermore, ZAP's cross-platform availability across major operating systems, including a Docker version, enhances its usability in various setups. In addition, the tool can also be extended by third parties in the form of marketplace add-ons that are available for direct access through a ZAP client with specialized functionalities. Being an open-source platform, the ZAP's codebase is available for people to read and modify, which means that anyone can contribute towards its enhancement, fixing bugs, modifications or additions of new features and even coming up with new add-ons, all of which contribute to its ongoing improvement and adaptability to emerging security needs.

Discovering Resources in Target Applications

ZAP's detection tool functionality greatly relies upon exploring the target application. The more comprehensively ZAP can explore an application, the more accurate and complete its vulnerability detection results will be. To do this, ZAP offers many ways of getting started: spidering using a traditional web crawler, AJAX, Proxy using regression tests or unit tests, or importing OpenAPI/SOAP if defined. Whereas the traditional spider is designed to map from one page to another based on HTML hyperlinks, the AJAX spider is specifically invaluable in providing value in dynamic applications in which AJAX makes asynchronous requests in loading content. Further, proxy-based exploration is suggested with scenarios where regression or unit tests are already in place since ZAP can intercept the tests and develop an application map from them. If an OpenAPI definition is available, ZAP can leverage this and directly identify API endpoints, making vulnerability detection in RESTful APIs much more effective.

The spider component of ZAP is extremely important as it crawls through a set of seed

[2] https://www.zaproxy.org/

URLs and then follows the links to map out an entire site. Depending on its configuration, the spider handles various responses appropriately. The spidering process is normally carried out before an active or passive scan by mapping all the structures within the application so that ZAP can then analyze them more deeply.

Effectiveness, Autonomy, Aggressiveness, and Configurability

ZAP is very efficient for discovering numerous types of common web application weaknesses, including cross-site scripting (XSS), SQL injection, remote file inclusion, and CSRF (Cross-Site Request Forgery). This effectiveness is derived from its combination of active and passive scanning, supported by spidering tools that systematically assess application security. ZAP provides automation features and is compatible with CI/CD tools, it remains good to sometimes have a human touch, especially where the testing may entail specific configurations that automated tools may not be able to capture presenting it with a medium level of self-sufficiency. Despite this, ZAp's scans can also be performed autonomously without requiring user intervention.

In terms of aggressiveness, within ZAP, users can set the degree of intrusiveness in testing to achieve the best result without being invasive. Through the scan policies that are provided for use by users, the depth, detail, and number of effects that may be realized from the scan of the defined target application can be regulated. This level of configurability is high; ZAP supports nearly all types of customization options, including scan policies, authentication settings, proxy configurations, and other scan-specific parameters. This makes ZAP flexible enough to fit into various testing scenarios because, depending on the architecture of the target application or its security level, appropriate features will be available for use.

Testing Approaches

ZAP incorporates various testing approaches to identify vulnerabilities in web applications, including passive scanning, active scanning, spidering, fuzzing, and forced browsing, each serving specific security assessment functions.

Passive scanning performs an analysis over HTTP or HTTPS traffic without any form of intervention on the application. The scanner can evaluate traffic patterns and data to identify possible vulnerabilities. Such an approach is useful for discovering security problems such as absent security headers and unprotected paths and serves as a safe assessment that does not modify the application state.

On the contrary, with the active scan, the scanning engine sends a direct assault on the application; it tries to invade the application by feeding it with a certain kind of payload and observing the application's reactions to these inputs. This method detects problems like SQL injections, XSS, and path traversal by actively engaging the application with simulated attacks. In traditional and hybrid AJAX modes, the spider captures all parts of the application by following links, form tags, and sitemap elements to reveal pages that are often ensconced.

Additionally, ZAP has a fuzzing feature that enables testers to introduce payloads into any input field to check how the application behaves toward invalid data. On the other hand, forced browsing is used to try to get to other potential resources through files and directories not shielded by user authentication.

Active and Passive Scanning Details

In ZAP, passive scanning analyses HTTP requests and responses without modifying their contents, ensuring a non-intrusive form of security assessment. This approach allows security testers to obtain insights into potential application weaknesses without actively altering the application's state, data, or behaviour. Passive scanning is automatically triggered in ZAP whenever HTTP traffic is proxied, capturing and assessing each transaction without affecting the application's integrity. Consequently, passive scanning is well-suited for initial assessments, as it allows security professionals to gather essential information on structural vulnerabilities with minimal risk.

Following the initial spidering scan, which specializes in mapping the application's structure and determining which resources are available to the attacker, passive scanning can analyze each recognised resource for typical security vulnerabilities. This mode is good at pointing out issues like HTTP security headers, including Content-Security-Policy, X-Frame-Options and Strict-Transport-Security, which protect against threats such as click-jacking and injection. Also, passive scanning verifies anti-CSRF tokens, which are vital in combating cross-site request forgeries meant to authenticate user operations as intentional. Further low-risk vulnerabilities which may be detected during the passive scan are information disclosure related to details of the HTTP server or application used, HTTP methods which might lead to undesired interactions with the application, such as TRACE and OPTIONS, and possibly misconfigurations which allow the scanning tool to access sensitive information without further aggressive probing. Hence, the passive detection mechanism of the ZAP comprises an important starting point for evaluating the system's security health, and a key element is to put into perspective the system configuration problems that do not involve active modification of requests.

On the other hand, active scanning in ZAP is a direct and sometimes aggressive approach that simulates different types of attacks to identify several vulnerabilities. This scan proactively tries to look for different great impact insecurity problems, such as SQL injection, cross-site scripting, command injection, remote file inclusion and bypass of authentication. To start an active scan, users have to set a target URL or context in ZAP, as well as a scan policy which defines the range and intensity of the scan and what kinds of attacks are to be performed. Active scanning then utilises crafted requests containing attack payloads designed to exploit known security flaws. By observing how the application processes these potentially malicious inputs, ZAP identifies vulnerabilities and highlights exploitable weaknesses.

Active scanning builds upon ZAP's extensive library of pre-configured payloads and attack patterns, which complies with OWASP best practices in security testing. This library consists of SQL injection payloads like union, time and boolean injections, XSS payloads to check poor data handling and encoding of the user input fields and command injection strings to perform system commands on the server side. ZAP places these payloads within form fields, URLs, cookies and HTTP headers, and then watches for signs of vulnerabilities in the application's response. For instance, an SQL injection attack may reveal specific database error messages or access to unauthorized information whereby a vulnerable XSS endpoint will display the following unescaped JS in the client browser.

However, active scanning should be applied only after caution and informed consent because the simulated attacks mentioned above change the application's data, produce logs or errors, and interfere with a normal application's functioning. For example, an SQL injection scan carried out forcefully can change records in a database or create disparities in data, or an XSS payload may trigger several unwanted dialogues or change the application's layout. Thus, active scanning is best suited to an already controlled environment, or it can be used for testing purposes only because its approach based on vulnerability scanning can lead to numerous unfavourable consequences in the network it scans.

In addition to its predefined attack patterns, ZAP's active scanning module is highly configurable, allowing testers to refine the scan scope according to application requirements. Customization allows for selecting specific HTTP methods (e.g., GET, POST) and providing custom postData values to simulate various input types and formats relevant to the application. Advanced users may also define custom scan policies, tailoring scan intensity, attack patterns, and payload types based on the application's unique architecture and risk profile under test.

Although extremely powerful, ZAP's active scanning process has to be carefully managed and configured, balancing efficacy with potential operational disruptions of target applications. Configuring it properly, the active scan can reveal critical vulnerabilities that would be undetected with a passive scan only, such as SQL injection attacks.

Post-Scan Information and Reporting

ZAP sorts findings after a scan on priority levels, categorizing alerts as high, medium, low, and informational according to the detected vulnerabilities' severity. It also provides mapping with CWE codes, which is particularly useful in risk estimation. ZAP provides a composite report consolidating findings from active and passive scans in a human-readable format so administrators can examine the vulnerabilities and risks encountered. ZAP provides reporting capabilities that can output reports in different formats like HTML and XML, which comply with dissimilar standards and maintain a thorough list of discovered vulnerabilities.

4.3 Nikto

Nikto[3] is an advanced, fast, interactive open-source web server scanner specialising in comprehensive security testing. Its primary use is for scanning the web servers for identified

[3] https://github.com/sullo/nikto

weaknesses and faults along with outdated server applications. By scanning web servers for over 6700 possible vulnerable files, over 1250 outdated web servers, and 270 server-specific version problems, Nikto demonstrates high efficiency in identifying a number of basic server-side vulnerabilities. It checks several configuration elements, including HTTP server options and the presence of multiple index files, and attempts to identify the specific web server and software in use. As it has a number of plugins that are updated frequently, Nikto has the advantage of the periodic update of the scan items, which is beneficial in a network that needs constant server checks.

Nikto is fast and effective, focusing on speed for finding problems when scanning rather than stealth. While not a strictly covert tool, in fact it is well known to many Internet users, Nikto is useful because of its fast analysis and a large database of signatures that can help detect threats on web servers. Its primary use is in penetration testing and security auditing, where it offers a fast view of a server's vulnerability.

Effectiveness, Autonomy, Aggressiveness, and Configurability

Nikto is only effective when used as a vulnerability assessment tool to scan and reveal various web server vulnerabilities, such as outdated software versions, misconfigurations, and common vulnerabilities documented online. Although the latter might be useful for specific environments, Nikto fails to connect these conclusions to global vulnerability signs, such as CVE or CWE. However, it is a very helpful tool in checking the server level and figuring out what areas may require more scrutiny with refined tools.

The tool's automation level is considerably high because it was designed to scan multiple Web servers and applications with minimal human intervention, in fact the main feature of Nikto is that it is largely automated. Users have limited control over it, although there are options for selecting individual tests or plugins to run automatically by default, making Nikto suitable for large and/or frequent scan tasks. In complex and more unique test cases, the results may be further improved manually to suit the testing option, and even though Nikto can be scripted into automatic test runners, it may be necessary in certain instances to tweak the settings manually.

Nikto's design is oriented toward the aggressive scan, which focuses on inspecting web servers quickly. Unlike other generic vulnerability assessment tools in nature, which work in a stealth mode to avoid being detected, Nikto scans the targets quickly and makes its requests easily visible to the Intrusion Detection System (IDS). It is equipped with configuration options for tuning scan speed and intrusiveness, yet it stays somewhat invasive, so its use is best for controlled environments where the testing speed is more valued than its stealth.

As for configurability, Nikto guarantees a moderate level, enabling users to change certain scanning parameters and output results. It provides fundamental settings for customized scan and plugin usage but has fewer possibilities than enhanced scanning tools. However, this capability is sufficient to meet a broad spectrum of testing necessities to afford critical configuration standards for effective connection with numerous settings.

Scanning Techniques and Assessment Methods

Nikto uses a set of specific scanning techniques designed to quickly and effectively identify server-side vulnerabilities, known security flaws and insecure configurations. The most important underlying principle that the scanner depends on is the use of signatures combined with a large, often updated, database of known issues. By comparing the signature to the server responses, Nikto can effectively and rapidly determine potential vulnerabilities, thus making it a suitable tool for a large preliminary evaluation.

Among those techniques, Nikto's old software identification targets older server software versions that may be penetrated. Using a comprehensive database of software version information, Nikto cross-references detected server versions against a repository of known security issues to flag versions that could be susceptible to exploits. This capability is available to multiple types of servers, such as Apache, IIS, Nginx, and various web server plugins, which could potentially contain version-specific security concerns. Since Nikto searches for information that is unique to a given version, the tool does not have to actively exploit the vulnerability, thereby reducing the effects it might have on the target server and, at the same time, providing meaningful insights.

Furthermore, Nikto operates another check over the misconfigurations and insecure server options triggered if the server is fully open to exploitation. For instance, Nikto analyzes headers of HTTP requests, such as security headers like X-Frame-Options, Content-Security-Policy, and Strict-Transport-Security, which prevent us from click-jacking, injection, and protocol downgrade attacks. The tool also checks HTTP methods supported by the server optionally to include dangerous methods like TRACE, PUT, or DELETE that, if activated, can be used by attackers. This is a misconfiguration of the first kind, and any such mistakes are dangerous as they can lead to data leakage or unauthorized access and should be found during the first-stage security checks.

As another important activity of Nikto's scanning procedure, directory and file enumeration are worth mentioning. Nikto looks for a rather extensive list of files and directories that might be harmful or contain sensitive information by using paths that may unveil administrative interfaces, testing pages, backups, and documented settings which may contain potentially sensitive information. For instance, it seeks paths that are related to certain web app frameworks, default installation files, and directories that do not require authentication since they carry data on the layout of the server. This technique corresponds to the search for directories that remain initially open or unprotected by a password, thus allowing an attacker to gain access to a directory containing important files, such as '/admin', '/backup', or '/config'.

Besides the default directory scans, Nikto performs so-called plugin scans, using modules to enhance the scanner functionality for specific scans. These plugins attack application, component or framework-level exposures like PHP or Java environments containing set-up problems or some other inherent exposures. For example, the Nikto scanners have plugins for detecting issues with content management systems such as WordPress and Joomla, where outdated or wrongly configured plugins may permit unauthorized access or code implementation. With the help of these plugins, Nikto can perform more in-depth scans of the areas where some specific web applications or server configurations might be most at risk.

Although Nikto does not conduct deep application-level vulnerability testing, it is very efficient at quickly pointing out top-level server security issues and possible misconfigurations. This rapid systematic scan technique enables Nikto to draw a server's network security perimeter precisely, making it a perfect starting point in a vulnerability assessment. It enables security testers to quickly assess server security and prioritize areas for further investigation, particularly in large environments where multiple servers must be assessed regularly.

Reporting and Results Interpretation

Despite the fact that Nikto successfully scans for a number of possible risks on web servers and offers fully configurable report generation, it has one major drawback: it is not compatible with CVE and CWE systems. This lack of standard severity mapping entails security practitioners having to browse the identified vulnerabilities and look for the corresponding CVE or CWE entry to determine the likely consequences and risks. The absence of specific CVE or CWE numbers also creates challenges for automated reporting since it narrows the communication with other tools in the vulnerability management processes, which might result in slower or less efficient remediation.

Additionally, Nikto's output focuses more on high-level indicators rather than detailed threat analysis, which, while useful for initial assessments, requires further in-depth analysis to understand vulnerability specifics. This characteristic can limit its utility in environments where detailed vulnerability categorization is essential for compliance or risk management.

4.4 Arachni

Arachni[4] is an advanced open-source web application security scanning tool suitable for monitoring modern web applications. Developed for fast and accurate web application security testing, Arachni follows a modular pattern, allowing for flexibility in scanning profiles. This tool efficiently identifies most of the known vulnerabilities, such as SQL injection, cross-site scripting, remote file inclusion and many more. Passive and active scanning used in Arachni guarantees its efficient operation in terms of vulnerability detection.

Another important advantage of Arachni is its interface, which is, in fact, available in both web-based and command-line, making it very user-friendly. This dual approach makes it feasible for the novices and, at the same time, useful to professional security personnel. Arachni is extensible through plugins, allowing the user to specify certain vulnerability checks that need to be performed or allowing the tool to fit into the existing security framework. Quantitative data reports produced by Arachni contain actionable information,

[4] https://github.com/Arachni/arachni

which makes it possible for the security testers to rank the discovered weaknesses according to the risk level and remedial actions.

Effectiveness, Autonomy, Aggressiveness, and Configurability

Arachni demonstrates high effectiveness in web application security scanning. It employs asynchronous HTTP requests to optimise bandwidth utilisation, ensuring that scans are executed quickly and efficiently, limited only by the physical resources of the scanning machine or the targetted server. This capability is further supported by activities encompassing several approaches to address the variability of web application contexts. While not claiming any originality in concept, Arachni integrates common techniques such as taint analysis, fuzzing, differential analysis, timing/delay attacks in addition to the rDiff analysis and the meta-analysis that are exclusive to the framework. This makes it much more advanced than other programs and allows Arachni to make decisions based on multiple different inputs all at once.

A major feature of Arachni is its Trainer module, which allows the system to learn from the scans it performs. It can tell which requests are more likely to disclose new elements or attack vectors, allowing it to improve detection outcomes in real time. Additionally, individual components of Arachni can force the framework to learn from the HTTP responses they generate, further increasing the likelihood of uncovering hidden vulnerabilities.

Concerning the level of autonomy, Arachni is a fully automated tool which came as the “fire and forget” principle. When a scan starts, no interaction is needed until the scan process completes. After terminating a scan, the result can be stored in a file or, if desired, easily converted into various formats, such as HTML, plain text, or XML, facilitating straightforward reporting and analysis.

However, Arachni also has quite a high level of aggressiveness, which can definitely lead to significant stress on the scanned web and database servers. This possibility of disruption requires a careful moderation of the intensity of scans. Arachni employs two primary methods to mitigate this risk, these include the manual limiting and the auto-throttling. This default configuration allows 20 concurrent requests, but this number can be changed in any way using the ‘–http-req-limit’ parameter. Auto-throttling, on the other hand, depends on the response time where, if the response time is low, the concurrent request limit is safe to increase, and if it is high, the concurrent request limit is reduced to reduce server pressure.

Another advantage of Arachni is configurability, which reaches high levels thanks to its modular architecture. The framework can be further extended by adding components such as path extractors, modules, plugins and even user interfaces. Arachni, in addition to security scanning features, can also offer various forms of black-box testing and data scraping. Additionally, Arachni supports multiple deployment options, ranging from a simple command-line interface for single-user scans to a multi-user, parallel scan environment utilizing server pools.

Tool Name	Effectiveness	Autonomy	Aggressiveness	Configurability
Sqlmap	High	Medium	Medium	High
ZAP	High	Medium	Medium	High
Arachni	High	High	High	High
Nikto	Medium	High	High	Medium

Table 4.1: Summary of Security Tools for Penetration Testing and Vulnerability Scanning

Scanning Techniques and Assessment Methods

Arachni provides a variety of scanning techniques designed for comprehensive vulnerability detection. Among these, active scanning is one of the core methodologies, through which the tool sends well-crafted requests to the target application to get responses that may include vulnerabilities. By simulating real-world attack scenarios, Arachni tries to uncover such security flaws.

In addition to active scanning, the tool provides a passive scan, allowing traffic monitoring and identifying security misconfigurations without altering the application state. This approach is particularly useful for detecting issues in HTTP headers or unprotected endpoints.

Arachni also provides an advanced fingerprinting capability that effectively identifies web application frameworks, libraries and components. By detecting the employed technologies accurately, the tool can provide context for the found vulnerabilities, allowing more informed prioritization and remediation efforts.

Overall, Arachni's combination of active and passive scanning techniques, along with its modular architecture, makes it a versatile tool for security professionals looking to evaluate and enhance the security of their web applications.

4.5 Tool evaluation summary

The following table summarises the four security tools for penetration testing and web application security scanning that have been previously presented. Each tool offers varying levels of effectiveness, autonomy, aggressiveness, and configurability, making them suitable for different security testing scenarios.

In conclusion, evaluating the four security tools, Sqlmap, ZAP, Arachni, and Nikto, highlighted their unique strengths and capabilities across various attributes. While each tool demonstrates specific advantages, the ZAP tool is selected as the most suitable option for integration into the expert system.

ZAP's high levels of effectiveness and configurability, combined with its moderate level of aggressiveness and sufficient level of autonomy, make it well-suited for a balanced approach to web application security testing. Therefore, based on the evaluation criteria, ZAP is the recommended choice for incorporation into the expert system for effective and adaptable security assessments.

Chapter 5

Improving the Expert System

The step that followed the tool evaluation process was selecting and integrating one of the previously described tools. Due to its high level of effectiveness and configurability, OWASP ZAP (Zed Attack Proxy) was chosen as the tool to be integrated into the RiskMan expert system. This decision was driven by several factors identified during the evaluation phase, including ZAP's robust feature set, ease of integration, and adaptability to various testing and automation requirements.

ZAP provides seamless integration with Python through a well-documented API that exposes most of its core features. By leveraging ZAP's APIs, RiskMan can efficiently perform security testing operations in both desktop and daemon modes, enhancing the flexibility and scalability of the testing framework. The daemon mode, in particular, allows ZAP to run in a background process, making it possible to conduct continuous scans without manual intervention. This headless mode is ideal for automated testing workflows, where RiskMan can invoke ZAP's scanning functions directly through API calls, analyze results, and proceed with risk scoring seamlessly.

One of the strengths that factored into the decision to adopt ZAP was its comprehensive alert reporting mechanism. ZAP's alert reports are particularly useful as they detail specific vulnerabilities and include mappings to Common Weakness Enumeration (CWE) identifiers. This mapping provides valuable context for each identified risk, aligning it with a standardized framework widely used in the cybersecurity industry. The availability of CWE mappings is critical in the risk-scoring phase of RiskMan, as it enables a more structured assessment of the vulnerabilities based on their severity and impact.

Moreover, ZAP supports a range of plugins and extensions that can be utilized to further customize its capabilities within the RiskMan system. These extensions allow for targeted testing approaches, such as simulating specific types of attacks or tailoring scan parameters to fit the organization's unique security policies. For example, specific plugins can be enabled to test for SQL injections, cross-site scripting (XSS), and other vulnerabilities that may pose significant risks.

5.1 Python integration through ZAP's APIs

Integrating ZAP within a Python environment provides an automated and efficient security assessment of web applications. The Python function presented below is aimed to help launch ZAP in daemon mode, which prepares the tool for further interaction through API calls, to start a spider scan of the target URL, to manage and supervise the processes of the scan operation, as well as to organize the extraction of differentiated security alerts obtained during the scanning activity. This implementation considers many operational characteristics, such as introducing the ZAP process in a non-blocking manner, verifying ZAP's readiness and ensuring resilience through error handling.

The function starts by initializing ZAP in daemon mode, which is done by launching the ZAP process through a subprocess. The function then waits for ZAP to start by checking its status through periodic API calls to confirm the availability of the ZAP instance, with a maximum timeout of 120 seconds.

```
    # Start ZAP in daemon mode
    zap_process = subprocess.Popen(['java', '-jar', zap_path, '-daemon', '-port', '8080', '-host', '127.0.0.1','-config', 'api.disablekey=true'], stdout=subprocess.DEVNULL, stderr=subprocess.DEVNULL)

    while True:
        try:
            zap = ZAPv2(apikey=apikey)
            zap_version = zap.core.version  # This will throw an exception if ZAP isn't reachable
            break
        except Exception as e:
            seconds_counter = seconds_counter + 1
            if seconds_counter == 120: #if zap does not start in 2 minutes, go on
                break
            time.sleep(1)
```

Once the ZAP instance is ready, the function constructs the target URL to ensure it is in the correct format (i.e., including `http://` or `https://`), this target URL will be subjected to security scanning. By default, the ZAP API client connects to localhost on port 8080, which is consistent with the port configuration defined when starting the ZAP process in daemon mode.

```
    if not (target.startswith('http://') or target.startswith('https://')):
        target = f'http://{target}/'

    # By default, ZAP API client will connect to port 8080
    zap = ZAPv2(apikey=apikey)

    try:
        zap.urlopen(target)
```

The main part of the function is structured into two core stages: the spidering stage and the passive scanning stage. In the spidering phase, the function starts a spider scan of the target URL, primarily responsible for discovering all links within the specified domain, creating a comprehensive map of the web application's structure and ensuring comprehensive coverage of all accessible endpoints. The function tracks the progress of the spider scan and employs several control mechanisms to ensure it completes efficiently and does not stall indefinitely. If the scan progress becomes 0% for a longer period, or the percentage of progress does not rise within certain time intervals, the function stops the scanning process to avoid a very long time for the scan. This responsiveness to scan progress contributes to the function's resilience and efficiency, making it adaptable to various network conditions or application response behaviours. In order to always provide some data, the discovered alerts are saved periodically so that if an exception occurs, it is still possible to analyse the results.

```
        # SPIDERING TARGET
        scanid = zap.spider.scan(target)

        # Track the previous progress percentage
        previous_progress = -1
        same_progress_start_time = time.time()
        scan_start_time = time.time()

        while (int(zap.spider.status(scanid)) < 100):
            progress = int(zap.spider.status(scanid))

            # Check if maximum scan time is exceeded
            if time.time() - scan_start_time > max_scan_time:
                zap.spider.stop(scanid)
                break

            #If scan remains 0 for the same time
            if progress == previous_progress == 0:
                if time.time() - same_progress_start_time >=
    max_same_zero_progress:
                    zap.spider.stop(scanid)
                    break

            # If progress remains the same
            if progress == previous_progress:
                if time.time() - same_progress_start_time >=
    max_same_progress_interval:
                    zap.spider.stop(scanid)
                    break
            else:
                previous_progress = progress
                same_progress_start_time = time.time()

            # Periodically retrieve alerts found so far
            alerts = zap.core.alerts()

            # Short sleep instead of busy-waiting
            time.sleep(polling_interval)
```

During the next stage, the passive scanning phase, the function remains idle, waiting for the ZAP's passive scan component to process the collected HTTP request and response. This kind of scan starts right after the spidering phase and is of paramount importance because it is essential for identifying potential vulnerabilities without stressing heavily the target application. It keeps polling ZAP's status until all records passively scan through and review the identified application resources.

```
# PASSIVE SCAN
while (int(zap.pscan.records_to_scan) > 0):
    time.sleep(polling_interval)
```

Once the scan is complete, the function collects the security alerts created by ZAP during the process. An example of the information provided with every alert includes CWE ID, the type of weakness involved, a description of the weakness, and the URL of the web page in which the alert has been found. To avoid redundant information, the function utilizes a set to filter out duplicate alerts within the same URL, ensuring that only unique findings are presented in the output.

```
# Get all alerts (even after an exception or if the scan is incomplete)
try:
    alerts = zap.core.alerts()
except Exception as e:
    pass

# List to store unique alerts as tuples
seen_alerts = set()  # A set to track unique (cweid, alert, description, url) combinations
out = set()  # The final output set

for alert in alerts:
    alert_tuple = (alert['cweid'], alert['alert'], alert['description'], alert['url'])

    # Check if this combination is already in the set
    if alert_tuple not in seen_alerts:
        seen_alerts.add(alert_tuple)
        out.add(f"{alert['cweid']}\\{alert['alert']}\\{alert['description']}\\{alert['url']}")
```

The function includes comprehensive exception handling to manage potential issues, such as network errors or unexpected interruptions during scanning. In case of any such error, it attempts to terminate any active scan and ensures that all gathered alerts, regardless of scan completion, are returned to the user.

Finally, the function stops the ZAP process to ensure no orphan process remains active, which is good for the system's health. The function outputs a list of unique alerts in a structured format, making it ideal for further processing them in a format that could easily facilitate data analysis. This demonstrates a strong approach toward automating security testing, as shown in using the integration of ZAP's API via a Python function to improve vulnerability checks in web applications.

```
    # Terminate the ZAP process when done
    zap_process.terminate()

    return list(out)
```

5.2 ZAP function integration in RiskMan expert system

The next step is to incorporate the previously described Python function that calls ZAP's API in the RiskMan expert system. This integration is achieved with the help of the creation of a structured alert template, namely ‘ALERTEMPLATE‘, which defines the structure of all types of alerts, and ‘ZAP_SCAN_RULE‘, which controls the processes of ZAP scanning for the domains inserted in the knowledge base. Combined, these components offer an automated, knowledge-based approach to trigger security assessments and to prescriptively document the findings of the subsequent scans. These results will be available to the expert system in a structured format that is ready for processing.

The ‘ALERTEMPLATE’ template is a foundational element for organizing and processing the results of the ZAP scan. This template describes the format of each alert, which includes several fields that hold the important characteristics of vulnerabilities that ZAP might spot. By using this schema, the expert system can store vulnerability-related data to perform then the evaluation in the next steps.

The structure of ‘ALERTEMPLATE’ is as follows:

```
(deftemplate ALERTEMPLATE
    (slot cwe (type INTEGER))
    (slot alert (type STRING))
    (slot description (type STRING))
    (slot url (type STRING))
    (slot cvss (type STRING))
    (slot score (type FLOAT))
    (slot impact_score (type FLOAT))
    (slot expl_score (type FLOAT))
    (slot risk_amplifier (default -1.0) (type FLOAT))
    (slot weight (default -1.0) (type FLOAT))
    (slot parent_weight (default 1.0) (type FLOAT))
)
```

The fields within ‘ALERTEMPLATE’ provide comprehensive details about each alert. The ‘cwe’ slot preserves the identification number from the Common Weakness Enumeration (CWE), which is the universal model for weakness representation and can be applied to almost every problem we address. The ‘alert’ slot holds the name of each vulnerability, which can be useful to categorise the discovered security problem, and the ‘description’ and ‘url’ slots secure detailed descriptions and specific locations of the subject vulnerability.

Additional fields, such as ‘cvss‘, ‘score‘, ‘impact_score’ and ‘expl_score‘, record the Common Vulnerability Scoring System (CVSS) classification and overall score associated with the vulnerability. These slots will be further analysed later during the alert scoring section of this chapter.

Finally, the ‘risk_amplifier‘, ‘weight‘, and ‘parent_weight’ fields are designed to capture the broader risk context of each alert and ensure coherence within the expert system. In fact, these fields are consistently included in every fact stored in the knowledge base, allowing for a comprehensive calculation of the target’s overall risk at the end of processing.

After defining the template responsible for storing the alerts resulting from the ZAP scans, the CLIPS rule that calls the previously defined Python function is inserted in the expert system.

The structure of ‘ZAP_SCAN_RULE’ is defined as follows:

```
(defrule ZAP_SCAN_RULE
   (declare (salience 99))
   (DOMAINTEMPLATE (name ?name) (weight ?wgh) (risk_amplifier ?risk_amplifier))
   (test (eq ?risk_amplifier -1.0))
   =>
   (log_rule "ZAP_SCAN_RULE" "Starting execution")
   (bind $?alerts (call_zap_on_target ?name))
   (foreach ?alert ?alerts
      (bind ?rslt (split_on_slash ?alert))
      (bind ?cwe (nth$ 1 ?rslt))
      (bind ?alert_name (nth$ 2 ?rslt))
      (bind ?description (nth$ 3 ?rslt))
      (bind ?url (nth$ 4 ?rslt))
      (assert (ALERTEMPLATE
         (cwe ?cwe)
         (alert ?alert_name)
         (description ?description)
         (url ?url)
         (parent_weight ?wgh)
      ))
      (log_rule "ZAP_SCAN_RULE" "CWE {} - Alert: {} Description: {}"
         ?cwe
         ?alert_name
         ?description)
   )
)
```

The ‘ZAP_SCAN_RULE’ starts by scanning domains out of the expert system’s knowledge base, where the rule looks specifically for domains with risk_amplifier equal to -1, avoiding double scans on the same target. For each domain, the rule calls the ZAP scan function named ‘call_zap_on_target(?name)’ in the Python code previously analysed in detail, using the domain name as the target.

After the ZAP scan is over, the rule analyses each alert and presents it in the format of ‘ALERTEMPLATE‘. This step promotes organizational and analytic vector conversion of the raw scan data into a format that fits into a template that is processed by the expert system. The rule pulls knowledge or input as the CWE ID of the alert (‘?cwe‘), the name of the alert (‘?alert_name‘), an explanation of the sort of vulnerability that the alert is for (‘?description‘), and the Web page location where the vulnerability is located (‘?url‘). It can then assert each structured alert into the expert system’s knowledge base for later rule-based processing.

Moreover, ‘ZAP_SCAN_RULE’ also contains logs for every initiated scan and ended scan, increasing the visibility and also leading to the availability of records that are useful for tracking the flow of operations of the expert system as well as for historical assessment of scan data.

The concept of ‘ALERTEMPLATE’ and ‘ZAP_SCAN_RULE’ implemented in the RiskMan expert system allows for the integration of vulnerability scanning in an efficient, fully automated manner. A formal and systematic model for vulnerability assessment is provided by establishing a standard that specifies how an alert must be constructed and by creating a rule for the expert system to automatically set up and process ZAP scans. Therefore, integrating ZAP’s API with rule-based reasoning makes it possible for the expert system to provide comprehensive and context-sensitive consideration of security threats and, in this manner, facilitate improved risk estimation of the target.

5.3 Alert scoring in the expert system

Once the expert system has successfully called ZAP and retrieved the alerts as described above, it adds the results to the knowledge base in the form of ‘ALERTTEMPLATE’ facts, containing attributes such as CWE ID, alert name, description and placeholders for CVSS vector string (‘cvss‘), score (‘score‘), impact score (‘impact_score‘) and exploitability score (‘expl_score‘). The step that RiskMan performs right after that is the evaluation of the retrieved alerts one by one. This is achieved through the ‘ALERT_SCORING_RULE’ rule, which applies the ‘score_alert’ function to each alert. It is defined as follows:

```
(defrule ALERT_SCORING_RULE
  (declare (salience 99))
  ?alrt <- (ALERTEMPLATE (cwe ?cwe) (alert ?alert) (description ?description)
            (cvss "") (score 0.0) (impact_score 0.0) (expl_score 0.0))
  (test (neq ?description ""))
  =>
  (bind ?scores (score_alert ?cwe ?alert ?description))
  (bind ?cvss (nth$ 1 ?scores))
  (bind ?score (nth$ 2 ?scores))
  (bind ?impact (nth$ 3 ?scores))
  (bind ?expl (nth$ 4 ?scores))

  (log_rule "ALERT_SCORING_RULE" "Resolved: {} to cvss:{} with score: {}, impact: {}, exploitability: {}"
            ?cwe ?cvss ?score ?impact ?expl)
  (modify ?alrt (cvss ?cvss) (score ?score) (impact_score ?impact) (expl_score ?expl))
)
```

Whenever an ‘ALERTEMPLATE’ fact with missing ‘cvss’, ‘score’, ‘impact_score’ and ‘expl_score’ values is encountered, this rule triggers, calling the ‘score_alert’ function to score the alert based on its CWE ID, alert name and description. This function performs the scoring operation using the chatGPT[1] APIs to get the evaluation in the CVSS version 3.1 vector string format. It then returns the CVSS vector, Base Score, Impact Score, and Exploitability Score, which are then populated back into the ‘ALERTEMPLATE’ fact in the expert system’s knowledge base.

The function begins by initializing the OpenAI client, which will be used to communicate with the ChatGPT API. This requires a valid API key to authenticate requests.

```
    client = OpenAI(
        api_key='APIKey',
    )
```

The vulnerability data includes the data retrieved by ZAP: CWE ID, alert name and description. This information is formatted into a prompt, instructing ChatGPT to interpret it and return a CVSS vector string for scoring the alert. The prompt is divided in three different sections, the first one provides a detailed explaination of the CVSS v3.1, with particular focus on the different components and their possible values. In the second section, the alert’s data are provided to chatGPT to perform the evaluation. Finally, in the last section, details about the response format are introduced since the response has to include the CVSS vector string only, without any explanation that is not required and would otherwise create problems with the response’s interpretation.

[1] https://openai.com/

The function maintains a cache to avoid redundant API requests for already scored alerts (‘alerts_cache.json‘). If this file exists, the function loads it; otherwise, a new dictionary is initialized to store results for future use. Every time an alert is evaluated, the response is retrieved by the cache if present, using the CWE ID, alert name and description as the index to get the CVSS provided by ChatGPT, or if not found in the file, the ChatGPT request is sent and the alert is evaluated, updating the cache. If the same alert is requested in future, the same response will be provided, avoiding investing time and resources in evaluating the same alert again, even in subsequent scans. The file can be periodically deleted to get updated responses if needed.

```
    file_path = 'temp/alerts_cache.json'

    if os.path.exists(file_path):
        with open(file_path, 'r') as f:
            alerts_cache = json.load(f)
    else:
        alerts_cache = {}

    # Create a unique key for the alert using CWE, alert, and description
    alert_key = f"{cwe}_{alert}_{description}"

    # Check if the alert is already scored in the cache
    if alert_key in alerts_cache:
        return alerts_cache[alert_key]
    else:
        # Request CVSS vector from ChatGPT API
        response = client.chat.completions.create(
            messages=[
                {"role": "user", "content": prompt}
            ],
            model="gpt-4o",
            max_tokens=100,
            temperature=0.0
        )

        # Extract the CVSS vector from the response
        cvss = response.choices[0].message.content.strip()
```

The function interprets then the CVSS vector using the ‘calculate_score’ function, which calculates the vulnerability’s Base, Impact, and Exploitability scores using the set of formulas provided in the CVSSv3.1 documentation, allowing to retrieve the last information needed by the expert system to perform the overall evaluation.

```
        # Calculate the CVSS score, impact, and exploitability from the vector
        score, impact, expl = calculate_score(cvss)

        # Update the cache with the new alert score
        alerts_cache[alert_key] = (cvss, score, impact, expl)

        # Save the cache to file
        with open(file_path, 'w') as f:
            json.dump(alerts_cache, f)

        return (cvss, score, impact, expl)
```

Finally, the function returns the scores to the calling rule, which modifies the alert in the expert system's knowledge base with the updated information. The ‘score_alert’ function, combined with the ‘ALERT_SCORING_RULE‘, enables the expert system to process the result provided by the ZAP's scans and to obtain additional information useful to perform the evaluation of the target system.

5.4 RiskMan risk amplifier and weight updartes

To calculate the target system risk score, the RiskMan expert system needs to calculate two values for each of the facts contained in the knowledge base: weight and risk amplifier. To respect this constraint, the last step of the implementation stage of this research consisted in the implementation of 2 additional CLIPS rules and related Python functions.

The first one is related to the weight, which in the context of the RiskMan expert system is a multiplicative factor that is used to report the relevance of the data for the target system. A higher weight is assigned to facts with a higher relevance that would lead to higher impact threats. The rule is defined as follows:

```
(defrule ALRTWEIGHT
(declare (salience 1000))
  ?data <- (ALERTEMPLATE (weight -1.0) (parent_weight ?parent_weight))
  =>
  (modify ?data (weight (* (alrt_weight ?*CATEGORY*) ?parent_weight)))
)
```

This rule is triggered every time an ALERTEMPLATE fact with a weight equal to -1.0 is inserted in the knowledge base and is used to update the fact itself with the proper weight, taking into account also the parent_weight field, which represents the weight of the fact that generated the one taken into consideration.

The second rule introduced is related to the risk amplifier parameter, a value contained in the range [0-100] that represents the maximum risk the data may introduce to the target. The rule is specified as:

```
(defrule ALRTRISKAMPLIFIER
(declare (salience -1000))
  ?data <- (ALERTEMPLATE (score ?score) (risk_amplifier -1.0))
  =>
  (bind ?ra (alert_risk_amplifier ?score))
  (if (neq ?ra -1.0)
      then
      (modify ?data (risk_amplifier ?ra))
  )
)
```

The condition that an alert has to satisfy to trigger this rule is to have risk_amplifier set to -1.0. If the condition is met, the rule will call the Python function that will return the new value, if the returned value is not equal to -1.0 the fact is than updated in the knowledge base.

Chapter 6

Validation of the results

This chapter presents the validation of the research conducted in this thesis, focusing on two core areas: the assessment of the expanded expert system results and the performance evaluation of the newly integrated AI-based alert scoring. These validation activities are intended to confirm the reliability and effectiveness of the proposed enhancements, ensuring they meet the system's objectives. With a structured validation process, we can confirm that these enhancements support the operation and align meaningfully with the goals of this research.

6.1 AI Validation Process for Alert Scoring

After having successfully integrated ZAP into the expert system, there is now the need to evaluate and quantify the risk associated with each alert produced by the different scans performed during the target system analysis.

6.1.1 Process description

This section presents the broad validation strategy for assessing the effectiveness of four AI models in scoring alerts based on vulnerability descriptions to determine which models are the most suitable to be integrated into the expert system. The comparison covers each model's capacity, starting from textual descriptions of vulnerabilities, getting the right Common Vulnerability Scoring System (CVSS) version 3.1 vector string returned as a result. In fact, scoring the alerts with good precision is essential to assign an accurate risk estimation within the proposed RiskMan expert system, making this validation process a critical step.

A set of 5,000 Common Vulnerabilities and Exposures (CVEs) was selected for evaluation to ensure consistency and comparability across models. This dataset, containing a representative range of vulnerability descriptions, was uniformly applied across all four AI models under review: Mistral, Gemma, GPT-4o and Llama. Each model received the same 5,000 CVE descriptions and was prompted to analyze each one to assign a CVSS 3.1

vector string that reflects the vulnerability's severity based on criteria such as exploitability, potential impact and scope of impact, as outlined in the CVSS framework.

The prompt provided to each AI model required it to interpret and evaluate the key aspects of each CVE description, effectively simulating a real-world scenario in which automated systems assess vulnerabilities to inform cybersecurity prioritization and response strategies. The outputs that were produced from each model in the objective form of a CVSS score were then compared with the official CVSS score available in CVE databases. This step allowed for the accurate measurement and comparison of the relationship between the output of each model and the scores validated by the experts regarding each case.

When making this comparison, special emphasis was placed on how each model produces scores that closely match the official CVSS ratings. This alignment is a critical indicator of a model's proficiency in understanding and evaluating the severity of vulnerabilities across different types and categories. The evaluation also highlights any systematic biases or tendencies found in each model that may affect the scoring accuracy.

The validation process employed quantitative performance measurements to compare the models' accuracy, facilitating a standardized assessment of their respective alert-scoring capabilities. Metrics such as mean absolute delta for each score (Base, Impact and Exploitability) and for each component of the CVSS, both raw and normalized values, were used to assess each model's performance in relation to the official CVSS scores. These metrics made it possible to define shifts in the model reliability and, therefore, form a basis for choosing the most accurate, consistent and reliable scoring model.

Initially, the data resulting from 5000 CVE entries evaluated by each model were compared systematically to select which of the four AI models had higher accuracy and consideration for integration. The relative comparison included comparing the models' overall scores of the scoring accuracy at the data set level and per each of the CVSS attributes such as attack vector, required privilege and user interaction. This overall approach is important in our evaluation since it will not only give us a broad view of the performance of all the models but will also tell us where each model stands in terms of its scoring accuracy in specific areas.

The findings from this analysis will be presented in detail in the following section, where each model's strengths, limitations and overall suitability are evaluated for integration based on their alert-scoring capabilities. Through this structured and rigorous evaluation, the research aims to identify the AI model that best meets the needs of automated vulnerability scoring in a cybersecurity context, thereby enhancing the reliability and efficiency of our alert response system.

6.1.2 AI performances presentation

This section provides a focused analysis of each AI model's performance in assigning CVSS 3.1 vector strings based on vulnerability descriptions. The results are presented and compared for each evaluated model (Mistral, Gemma, GPT-4o, and Llama).

The following visualizations depict each model's precision compared to CVSS scores and reveal the patterns of scoring, the overall strong points and potential important tendencies or biases. This paper seeks to identify which of these models is most accurate and consistent in its scoring for different types of vulnerability so that the best one could be used in the expert system.

With these quantitative findings, the study establishes the effectiveness of the models on automated vulnerability scoring so as to form the basis for a sound decision on which model is most appropriate.

GPT-4o

The GPT-4o model, developed as a highly adaptive large language model, enables the interpretation of difficult texts and much deeper analysis in comparison with many other similar models; that is why it can be rather efficient if the tasks that have to be solved demand a high level of language understanding and awareness of the context, such as the vulnerability assessment required for this research. As a follow-up to previous models, GPT-4o is pre-trained on a large data set and fine-tuned on numerous sources while being designed to address subtle characteristics in natural language.

In the CVSS 3.1 vector scoring context, GPT-4o language capabilities make it a model suitable for scoring vulnerabilities in various categories, from configuration problems to deep exploitable paths. Its architecture enables description text analysis to derive the potential impacts, including aspects like confidentiality, integrity, and availability, representing important aspects of CVSS. Though processing these elements, the GPT-4o is expected to generate vector scores, which will be very close to human expert rating when assessing the level of severity and consequence of vulnerability impacts.

The following results and visualizations summarise GPT-4o's scoring patterns and highlight its comparative accuracy. This analysis sheds light on GPT-4o's strengths in automated vulnerability scoring and helps to evaluate its reliability as a candidate for integration into an expert system focused on cybersecurity risk management.

Figure 6.6 presents the performances of the AI model in evaluating and providing CVSS vector string for a total of 5,000 CVEs. The diagonal line corresponds to the situation in which the vector string predicted by the model actually replicates the official CVSS scores. Beliefs based upon the assessment of the light blue area reveal predictions that the deviation from the expected score is actually lower or equal to 2.0, revealing fairly accurate scores by ChatGPT. Over 3,500 records are in this blue zone, meaning ChatGPT can quite accurately determine CVSS scores most of the time. The red colour in the graph is used to indicate that the provided evaluation wasn't that close.

Of the 5,000 records used in the study, 1,028 (20.6%) were correctly scored, meaning that the CVSSs and all the scores were exactly the expected ones. On average, the score delta for a given input sent to ChatGPT sums up to 1.39, where the score delta refers to the approximated difference from the basic scoring system. The standard deviation of this score delta is 1.32, indicating moderate variability around the average error. The impact and exploitability components also convey moderate deltas, with impact scores having an average delta of 0.89 and those with exploitability scores of 0.61, with corresponding standard deviations of 1.05 and 0.76. This suggests that while GPT-4o generally provides

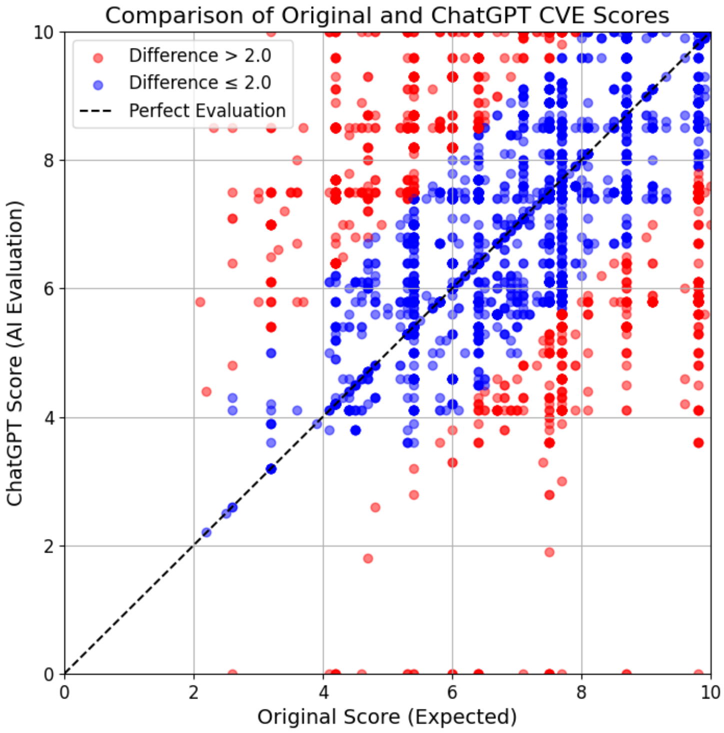

Figure 6.1: GPT-4o performancies

consistent evaluations, variability is higher for some components, particularly those with complex interpretations. When translated to a normalized scale from 0-10, these deltas give an average score delta of 1.48 for impacts and 1.52 for exploitability scores. The normalized standard deviations for these components rise to 1.75 and 1.91, respectively, reflecting that the deviations in GPT-4o's predictions are influenced by the component-specific scoring range and sensitivity.

The clustering of points near the diagonal line and comparatively small deltas, coupled with their standard deviations, confirm that, in most cases, ChatGPT performs reasonably well in scoring the original results. However, there are particular instances where ChatGPT can either underestimate or overestimate the severity of some vulnerabilities. However, the distribution of the evaluations is symmetric, meaning that, even when errors occur, they tend to balance out. This symmetry is good because it shows that the model doesn't tend to give a consistently higher or lower score to the evaluated vulnerabilities. Therefore, in cases of many alerts, which can be typical for the RiskMan expert system, the average evaluation remains relatively accurate, and the model is suitable for integration.

The bar graph in Figure 6.2 displays the CVSS total number of scores for each component provided by the GPT-4o model compared with official CVSS scores upon normalized

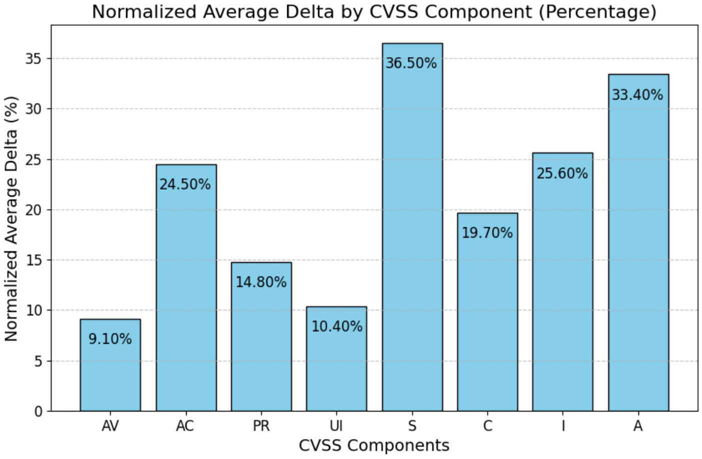

Figure 6.2: GPT-4o avarage delta by CVSS component

average delta (percentage difference) from CVE databases. This delta quantifies the average deviation of GPT-4o for each component, aligning and differing from the original predictions.

The chart highlights that the model behaves most differently in the Scope (S) component, where the delta reaches 36.5%. This implies that the model could not properly determine whether a vulnerability expands the range of the area (The impact beyond the security authority).

The next component, Availability, has a delta of 33.4%, meaning there is a great deal of uncertainty about how a given vulnerability could affect the availability of affected systems. It is also possible to observe that Attack Complexity (AC) has a delta of 24.5%, which indicates that, there could be issues in the outputs of the model for identifying low and high complexities of an attack.

In contrast, Attack Vector (AV) and User Interaction (UI) show lower deltas of 9.1% and 10.4%, respectively, which suggest that GPT-4o is more consistent with official scores in how a vulnerability is accessed (e.g., network, physical) and if the application requires a user to interact with resources as a means of exploitation. For Confidentiality (C) and Privileges Required (PR), deltas are at 19.7% and 14.8%, respectively, which are relatively close, indicating that, although there is some fluctuation, these components are rather stable in their performance.

This analysis suggests that GPT-4o performs well in certain areas, particularly Attack Vector and User Interaction. However, further tuning may be required for scope, availability impact, and attack complexity to enhance accuracy in automated CVSS scoring across all components.

Mistral

Mistral[1] is a state-of-the-art large language model (LLM) that marks a significant step forward in the evolution of artificial intelligence. Based on the current and enhanced transformer architecture, Mistral has been developed to enhance natural language understanding, generation, and contextual reasoning capabilities. It also employs new approaches, including dynamic attention mechanisms and advanced model optimization strategies, to ensure high-quality results with low computational requirements. These optimizations allow for higher efficiency of Mistral, making it perfect for high-performance computing environments and resource-constrained devices. Mistral embodies one of the industry's most welcoming, efficient and capable revolutions in Language-based AI by seamlessly blending innovative research with real-world applicability.

Regarding Mistral performance, the chart presented below displays the evaluations performed over the same 5,000 CVEs evaluated by GPT-4o that have been made by this model. As it is possible to see, with Mistral, the chart is not symmetric; this model tends to underestimate the vulnerability score. In addition to this, only 2600 of the evaluated CVEs have a delta that is less than 2.0, highlighting a less precise evaluation.

Of the 5000 records evaluated, only 571 records (11.42%) achieved accurate CVSS, matching the expected score. On average, the difference between the scores given by the Mistral model and the initial scoring system increased to call it a score delta of 2.02. The model also showed that there existed differences in specific components, with an average delta of 1.71 in the impact scores and a delta of 0.65 in the exploitability scores. After data normalization on a scale of 0-10, the averages of the deltas were 2.02 for the overall score, 2.86 for the impact score and 1.63 for the exploitability score. The standard deviations for these scores are as follows: 1.62 for the overall score, 1.37 for the impact score, and 0.66 for the exploitability score.

The chart in Figure 6.3 gives the percentage of the normalized average delta between the Mistral model scores and the expected CVSS 3.1 baseline scores across components. The data used also proves the presence of several fluctuations in Mistral performance for different components.

The User Interaction (UI) component has the highest delta, 59.8%, meaning that Mistral frequently deviates from the expected value when evaluating vulnerabilities that depend on user interaction. The deltas for Availability (A) and Integrity (I) are also high: 43.4% and 34.9% for the former, respectively, suggesting that the model performs poorly in estimating possible impacts on system availability or data integrity. In the meantime, such components as Attack Vector (AV) and Confidentiality (C) represent lower deltas, which makes sense, indicating the proximity of their scores to the expected level in these aspects.

The lowest delta is associated with the Attack Complexity (AC) index, with only 5.8%, indicating that Mistral is most accurate in assessing vulnerabilities based on this criterion. Thus, Figure 6.4 shows where Mistral is congruent with expected scores and where large

[1] https://mistral.ai/

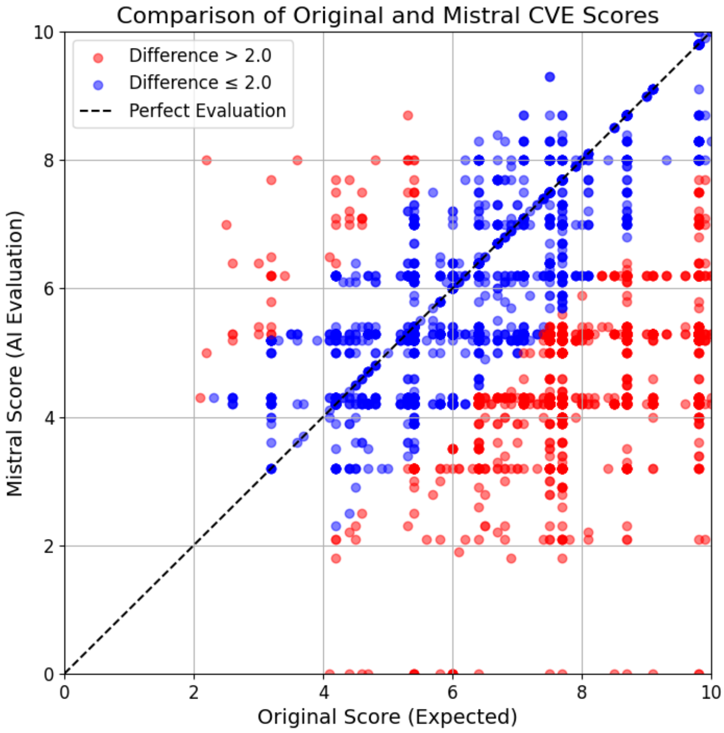

Figure 6.3: Mistral performance

discrepancies appear.

LLaMa

The Large Language Model Meta AI, which is abbreviated as LLaMA[2], is an accessible natural language processing system created by Meta to provide reliable natural language creation and comprehension. LLaMA is not another language model on the transformer basis: it has improved architecture and efficiency and is designed for accessibility. So, under open science, LLaMA has ambitions to democratize AI as a tool to experiment in areas like machine learning, conversational AI and content generation for researchers and developers. However, its real strength is its high performance combined with more extensive applicability, which serves as a foundation for the development of AI.

In terms of performance, the below chart presents the result of the evaluation of the CVEs. As it is possible to see, the distribution is symmetric, and more than 3,200 of the

[2] https://www.llama.com/

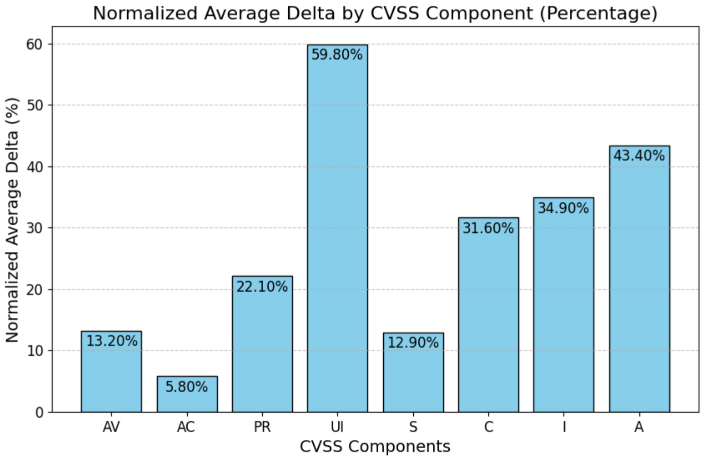

Figure 6.4: Mistral average delta by CVSS component

evaluated vulnerabilities have an absolute difference of base score lower than 2.0, which is good. However, only 886 (17.7%) of the total alerts have been correctly evaluated.

Regarding the deltas of the different scores, the average delta of the base score is 1.47, and impact and exploitability deltas are 1.75 and 1.88, respectively, after normalization on a scale of 0-10. The standard deviations for these scores further illustrate the model's performance: the base score has a standard deviation of 2.53, indicating a high level of variability between the evaluated scores and the expected CVSS values. This suggests that the model's overall vulnerability assessments may sometimes be inconsistent. The impact score, with a standard deviation of 1.84, shows moderate variability, meaning that while LLaMA performs reasonably well in assessing the potential impact of vulnerabilities, there are some fluctuations in the results. The exploitability score, with a lower standard deviation of 0.86, demonstrates greater consistency, suggesting that LLaMA is more stable in evaluating how easily vulnerabilities can be exploited.

While LLaMA's overall performance is impressive, the relatively high standard deviations for the base and impact scores imply that there may be inconsistencies in its overall vulnerability severity assessments.

On a detailed look at all the constituent parts of the CVSSv3.1 vector string, it can be clearly seen that the model is relatively accurate in the general average sense, especially for all the individual metrics with the exception of the scope (S) factor, which has a delta of 70.4%. On the other hand, the Attack Complexity (AC) component is nearly perfectly identifiable with a delta of only 4.2%. For the other components, the deltas range between 12% and 35%; a little higher deltas are shown in the Integrity (I), Confidentiality (C), and Availability (A) components of 33.5%, 24.5%, and 35.0% respectively. The User Interaction (UI) metric also deviates, with a delta of 19.8% and the Privileges Required (PR), which also differs with a delta of 23.1%. When the results are viewed at an overall level, they show us where the model is the most accurate and the least stable.

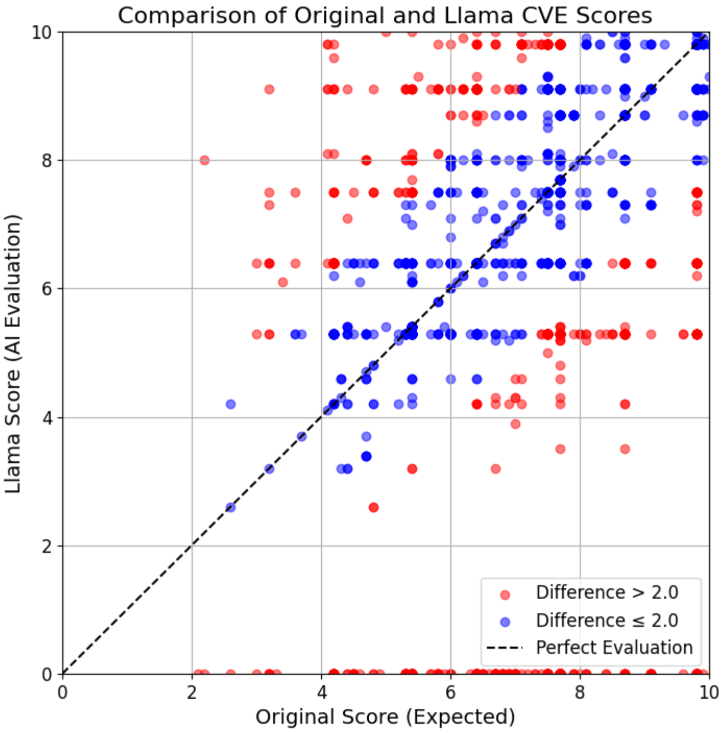

Figure 6.5: Mistral performancies

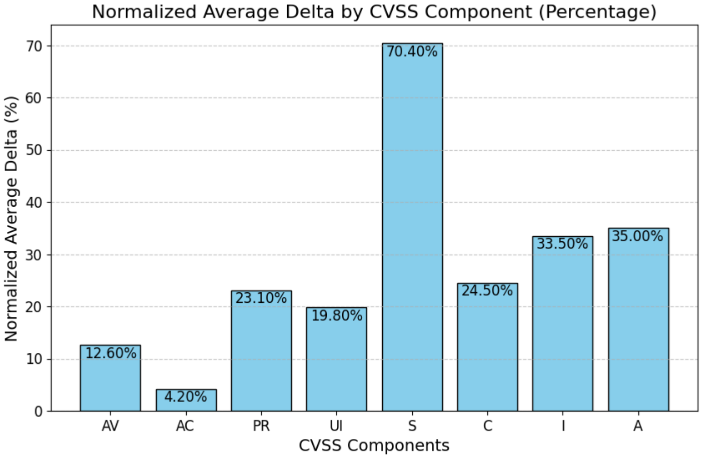

Figure 6.6: LLaMa average delta by CVSS component

Gemma

Gemma[3] is a family of light, open-source language models developed by Google DeepMind and other teams in Google. Stemming from more sophisticated Gemini models, Gemma harnesses the latest research in AI technology in development to provide high performance without requiring a large architecture. These models, Gemma 2B and Gemma 7B, are compatible with any natural language processing tasks, such as text generation, summaries generation, and answering questions. Nonetheless, they are optimal in many main benchmarks with comparable efficiency to the greater models despite the size.

Gemma models are designed to be flexibly deployable across a wide range of devices, from personal laptops and desktops to cloud infrastructure. They must be safe, using methods such as reinforcement learning from human feedback and auto-filtering training data. In order to ensure the safe development of AI, Google's Responsible Generative AI Toolkit is now in place to ensure the secure and ethical development of AI.

In addition to safety, Gemma models are compatible with major machine learning frameworks, allowing for easy integration into various development environments. They can be deployed on platforms such as Google Cloud's Vertex AI and NVIDIA GPUs, offering scalable and efficient solutions for both research and commercial use. With a strong emphasis on performance, safety and accessibility, Gemma represents a significant advancement in open-source AI models.

Using this model (see Figure 6.7), as displayed in the above chart, the evaluations tends to be an over-estimation with respect to the expected score, in fact the symmetry is not respected and the majority of the CVEs evaluations have been palced above the diagonal (where the expected value is placed). The correctly evaluated CVEs are 674, representing 13.50% of the total, with a base score delta of 1.66 and normalized impact and exploitability scores of 1.95 and 2.49.

However, over the 5,000 evaluations performed, over 3,000 have been placed in the blue area of the graph. In addition to these, from the chart, it is possible to notice that the model always tends to assign the same values. In fact, the distribution of the points shows a clustering in rows, meaning that, despite the evaluated CVEs having increasing expected scores (and so different CVSS vector strings), the model tends to always assign the same CVSS resulting from a pool of around 10 CVSSs.

Also, for this model (see Figure 6.8), the delta of the single components is provided, highlighting a more balanced evaluation of each of them. In fact, except for Attack Complexity (AC), which has a delt of 8.20%, and for Availability (A) with 41%, all the other components' deltas range from around 15% to around 30%.

In terms of the standard deviations, Gemma's performance can be better understood by examining the following details: The base score has a standard deviation of 2.56, indicating a high level of variability across evaluations. This suggests that while the model may often predict similar scores, there is considerable fluctuation in its base score assessments. The impact score has a standard deviation of 1.85, pointing to moderate variability, while

[3] https://ai.google.dev/gemma?hl=en

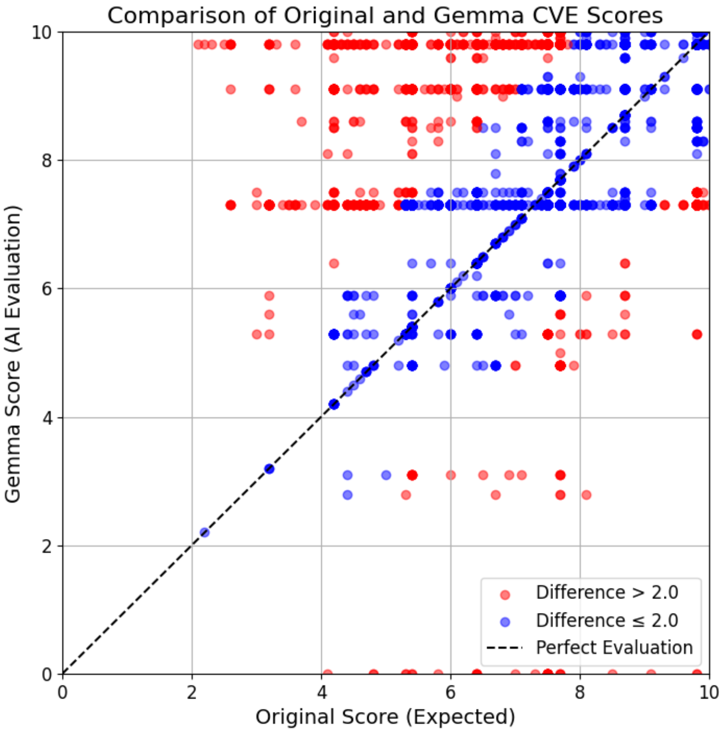

Figure 6.7: Gemma performance

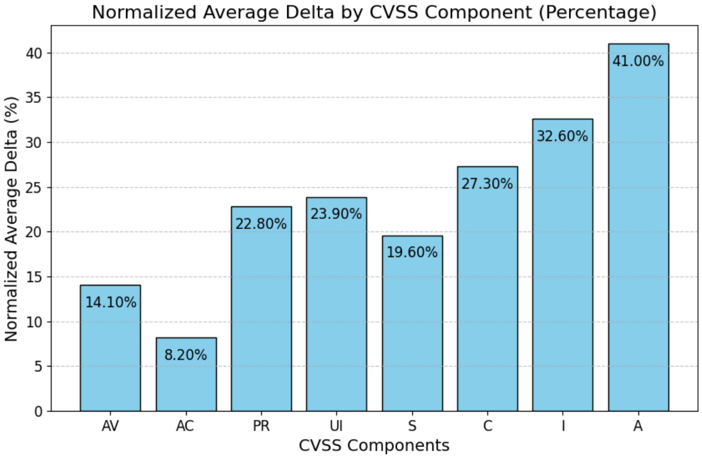

Figure 6.8: Gemma average delta by CVSS component

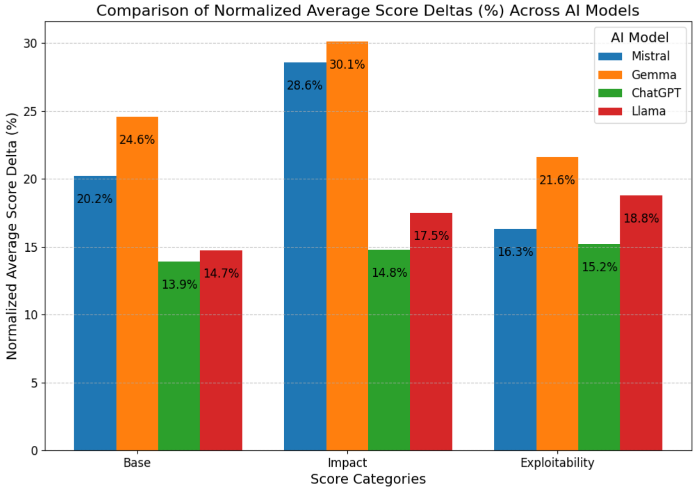

Figure 6.9: Models' deltas in scores

the exploitability score has a standard deviation of 0.86, indicating more stability in its evaluations.

For normalized scores on a 0-10 scale, the standard deviations increase, with the base score having a standard deviation of 2.56, the impact score at 3.08, and the exploitability score at 2.15. These values suggest that while the base and impact scores show moderate instability, the exploitability score remains more consistent, even if not so encouraging.

Comparison of the models

This section compares the previously described results, highlighting the strength of the chosen model for integration into the RiskMan expert system. The main points taken into consideration in the decision are the correct scoring rate and the deltas of the scores. In addition to these, the performance over the various fields of the CVSS vector string have also been considered.

The comparison across the main score categories (Base, Impact and Exploitability) reveals significant performance differences among the four evaluated AI models (see Figure 6.9): Mistral, Gemma, GPT-4o and Llama. Similarly, in the case of the base score category, GPT-4o shows a tolerable accuracy improvement with a normalized delta of 13.9%. Though this value is less than that of the other models, it gives a picture of GPT-4o's reliably accurate scoring directions without being too much off the mark. Looking at the performance obtained in the Impact score, two models seem to be more stable, in fact GPT-4o and Llama provide way better results with respect to the other two models. Almost all models provided similar results for the Exploitability score category, yet GPT-4o performed better than the others. These results indicate that the GPT-4o model could be the most appropriate one to integrate, bearing in mind that, in terms of

CVSS Component	Mistral (%)	Gemma (%)	ChatGPT (%)	Llama (%)
AV	13.2	16.3	**9.1**	12.6
AC	5.8	5.6	24.5	**4.2**
PR	22.1	19.2	**14.8**	23.1
UI	59.8	16.3	**10.4**	19.8
S	12.9	**12.8**	36.5	70.4
C	31.6	33.9	**19.7**	24.5
I	34.9	29.9	**25.6**	33.5
A	43.4	35.8	**33.4**	35.0

Table 6.1: Comparison of Normalized Average Delta by CVSS Components for Different AI Models.

correct scores, in the CVEs evaluations, the chatGPT's model has obtained the highest results with 20.5% of correct score rate.

These results in Table 6.1 confirm that GPT-4 can be used for RiskMan, as the detailed examination of the differences between the CVSS components supports this. In the case of the Access Vector (AV), GPT-4o brings a normalized delta of 9.1%, which is lower than every other model's performance on the same component. Regarding Access Complexity (AC), GPT-4o has the highest delta of 24.5%, with all the other three models under 6.0%, meaning that in this field, chatGPT lacks a bit in the understanding of the vulnerability. Another weak point in the GPT-4o evaluation is the Scope (S) component, for which GPT-4o provides the second worst result with 36.5%. Except for these two, it provided better results, so the lowest deltas from the expected ones, in every other field.

Although Gemma shows similar and more consistent deltas in some of these categories, GPT-4os balanced scores are better aligned with the RiskMan system's requirement for adaptability and consistent accuracy across all components. It must also be considered that, in case of errors during evaluations, GPT-4o tends to provide balanced errors, compensating these for the overall evaluation of a target system with RiskMan. In contrast, Gemma tends to overestimate, providing an unbalanced result.

Therefore, the GPT-4o model was chosen for integration into the expert system to conclude the evaluation. While the results indicate that there remains significant room for improvement, which may be addressed by newer and more powerful models in the future, the current GPT-4o already exhibits the potential to be useful in generating insights within the confines of the expert system and therefore is fit to be incorporated now.

6.2 Expanded expert system results

An important step after the implementation is the assessment of the effectiveness of the updated RiskMan expert system by comparing the results of the different versions. The comparison pursued in this study is designed to answer the question of whether the changes made in the new version of the expert system yield tangible improvements in the risk

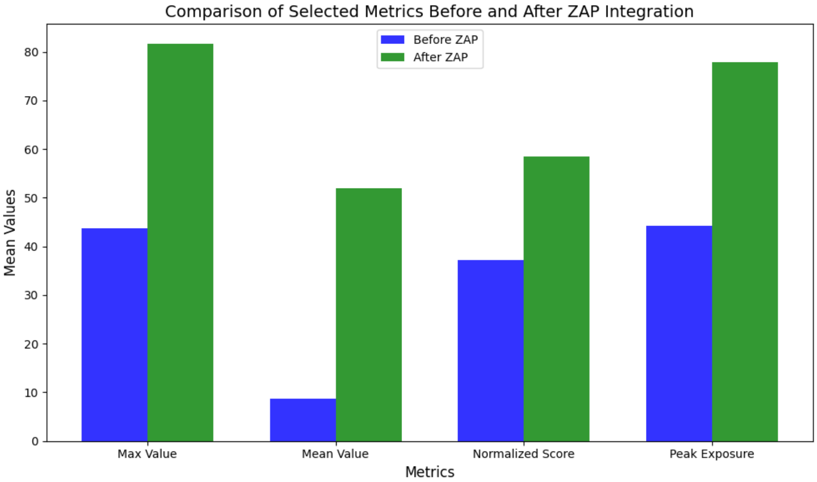

Figure 6.10: Significant metrics to compare RiskMan results before and after integration

assessment process. To attain this goal, broad experiments were carried out in more than 30 distinct scans on different target systems. In each target, risk assessment was conducted using both the initial RiskMan and the developed new RiskMan and all differences and improvements were carefully compared.

This comparison analysis has given a clear indication of the difference in time taken for the two versions. With the new advanced model of the RiskMan system, it is evident that there is an increase in its execution time due to the addition of the OWASP ZAP tool in the assessment process. Although this extension is helpful in expanding the breadth of the system's analytical capacity and analysis of risks and threats that exist in a given organization, its implementation does not mean that it takes longer to analyze the target system. The degree of this time increase is significantly influenced by two factors: the number of nodes which are included in the assessment process and the complexity or size of the target system.

One of the most significant findings in the analysis is the massive enhancement in the average number of assessed facts from 33.58 in the basic RiskMan system to 19,704.77 with the incorporation of OWASP ZAP (see Figure 6.10). This dramatic growth, however, does not indicate any form of constraint in the original system's ability to analyze. However, the primary reason stems from the massively increased number of alerts produced by the ZAP tool, which increased the amount of information to be evaluated. Thus, working with ZAP, the updated system captures more spectra of vulnerability and potential risk and makes the dataset and analysis more comprehensive. This increase should not be seen as a decadent feature of the original RiskMan system, but rather the integration of a tool which is intended to search a larger amount of data. This results in the improved RiskMan being capable of yielding more elaborate and reflective results concerning the target systems, thus introducing a work that increases the probability of no main risk being left undetected.

Besides the augmentation of the averaged amount of the evaluated facts, it is also possible to state the growing possibility of solving the critical problems connected with evaluating applied values and their extremes. The mean maximum value, which reflects the maximum degree of risk identified throughout the expert system before the integration of ZAP, was 43.64. This value was higher after integration, showing a value of 81.64, which indicates that the post-integration system can easily capture higher critical thresholds in risk analysis. Notably, again, the mean minimum value that represents baseline threats or risks did not alter from being equal to zero in both versions of the system since the addition of the ZAP tool doesn't affect the previously retrieved facts, keeping also the ones with value set to 0.

Another evaluation parameter that shows the enhancement of the system involves the mean normalized score. This score is an important instrument that may help assess overall risk levels in the target systems. However, in the initial version of the RiskMan system, the mean normalized score recorded was 37.13. When integrated with OWASP ZAP, this score equals 58.46, which is increased as expected since every alert provided by ZAP represents a possible vulnerability in the target system. Adding to the above is the average of the mean peak exposure, which increased from 44.23 in the first version to 77.95 in the second version. Combined, they show that the new RiskMan system provides a richer picture of risks where appropriate risks are not overlooked.

However, it should be mentioned that in four different cases, the integration of ZAP caused a decline in the normalized score. These cases imply that, despite significant improvement in the results derived from the updated methodology, there are specific conditions under which the implementation of the new strategy may result in an actual assessment score which is lower than the previous one. This is particularly interesting since although the scan performed by ZAP provides, as a result, a list of alerts representing possible vulnerabilities, this might lower the overall risk, providing a possibly more precise risk score as a result.

In total, 34 target systems were considered when comparing the initial and updated versions of RiskMan systems. This consistency in the number of targets ensures that the comparative analysis is based on equivalent datasets, thereby providing a fair and accurate basis for evaluating the impact of the changes introduced in the updated version.

In conclusion, the comparative analysis of the initial and updated versions of the RiskMan expert system demonstrates significant improvements in the system's ability to evaluate risks and vulnerabilities. Integrating OWASP ZAP enriches the system's knowledge base by providing additional facts, enabling a more comprehensive view of risks. Notably, the occasional decrease in normalized scores is a positive outcome, as it reflects the system's ability to identify and appropriately lower risk levels when justified, rather than systematically increasing them, highlighting a more nuanced and precise analytical approach.

Chapter 7

Conclusions

Automating cybersecurity risk assessment currently remains an issue, and consequently, it creates a vastly diverse and complex research domain. The great number of components and the specificity of the integration process of various tools and methods also influenced the research process. However, some meaningful progress has been achieved with this thesis, representing a good starting base from which further improvements in this important area can be made.

One of the crucial findings of this research was the evaluation and selection of cybersecurity tools to be implemented into the RiskMan expert system. To that purpose, a critical analysis of some tools assessed their strengths and weaknesses. Finally, the OWASP ZAP was implemented into the system, showing how an open-source vulnerability scanner can be used to make automatic risk evaluations. This improvement was also useful in mapping out the potential of optimizing existing tools for information gathering or vulnerability detection through an intelligent application of AI.

An important topic that emerged from the research is the application of AI as a complementary tool to strengthen the role of vulnerability scoring. Large language models were used to assess outputs from ZAP and enhance and enrich risk estimations. This approach helped the system prioritize vulnerabilities more efficiently, which helped to fill a gap in providing well-targeted and efficient information to resource-restricted organizations. The integration of these technologies, which was briefly depicted above, demonstrated the potential of the feature and the problems associated with data availability, as well as the necessity for further validation of the model outputs.

This thesis underscores the potential of automation and AI to revolutionize cybersecurity risk assessment processes. By addressing the foundational elements of risk management and integrating them into an automated framework, the research contributes to developing more accessible, efficient and reliable cybersecurity solutions for various organizations. Although challenges remain, the work presented here provides a clear path for further innovation and improvement.

7.1 Future Work

Based on the results of this research, it is possible to outline a few directions for further studies. Firstly, diverse vulnerability assessment tools can be considered for improved ranges and system flexibility. Tools capable of addressing cloud-specific risks and advanced persistent threats should be prioritized to ensure comprehensive coverage of emerging cyber risks.

Secondly, the usage of AI-driven components of the system can increase as well as the overall performance of the system. Adding explainable AI methods will improve the credibility and reliability of the risk assessment results and increase the probability of adoption by organizations with different technical experience levels.

Another sensible direction is to make the expert system change its behaviour in real time. Continuous learning mechanisms can be incorporated into the AI models to progressively update their knowledge, which will help the system change its approaches depending on alterations in the threats. Integration with real-time threat intelligence feeds can also give organizations timely information to enhance threat modelling and mitigate risks.

By pursuing these future directions, the RiskMan framework can evolve into a more robust, adaptive, and universally applicable tool, further democratizing access to effective cybersecurity risk management.

Bibliography

[1] Guido Perboli Gabriele Gatti Cataldo Basile. «An Expert System for Automatic Cyber Risk Assessment and Management». In: *IEEE Xplore* (June 2023). URL: `https://ieeexplore.ieee.org/abstract/document/10196997`.

[2] Gary L. R. et al. *CLIPS User's Guide: Building Expert Systems*. 6.4.1. CLIPS Project, 2023. URL: `https://www.clipsrules.net/documentation/v641/bpg641.pdf`.

[3] Hamed Taherdoost. «Understanding Cybersecurity Frameworks and Information Security Standards—A Review and Comprehensive Overview». In: *Electronics* 11.14 (2022). ISSN: 2079-9292. DOI: `10.3390/electronics11142181`. URL: `https://www.mdpi.com/2079-9292/11/14/2181`.

[4] National Institute of Standards and Technology. *Managing Information Security Risk: Organization, Mission, and Information System View*. Tech. rep. SP 800-39. Accessed: 2024-11-25. National Institute of Standards and Technology, Mar. 2011. URL: `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-39.pdf`.

[5] National Institute of Standards and Technology. *Guide for Conducting Risk Assessments*. Tech. rep. SP 800-30. Accessed: 2024-11-25. National Institute of Standards and Technology, Sept. 2012. URL: `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf`.

[6] National Institute of Standards and Technology. *Security and Privacy Controls for Information Systems and Organizations*. Special Publication SP 800-53 Revision 5. Accessed: 2024-11-25. National Institute of Standards and Technology, Sept. 2020. URL: `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf`.

[7] Henock Mulugeta Melaku. «Context-Based and Adaptive Cybersecurity Risk Management Framework». In: *Risks* 11.6 (2023). ISSN: 2227-9091. DOI: `10.3390/risks11060101`. URL: `https://www.mdpi.com/2227-9091/11/6/101`.

[8] Khalifa AL-Dosari and Noora Fetais. «Risk-Management Framework and Information-Security Systems for Small and Medium Enterprises (SMEs): A Meta-Analysis Approach». In: *Electronics* 12.17 (2023). ISSN: 2079-9292. DOI: `10.3390/electronics12173629`. URL: `https://www.mdpi.com/2079-9292/12/17/3629`.