INTEGRATING CVE DATA INTO SOFTWARE DEVELOPMENT LIFECYCLES FOR PROACTIVE VULNERABILITY MANAGEMENT

A project report submitted in partial
fulfillment of the requirements for the degree of
Master of Science in
Computer Science

By

PARGAT SINGH
M.S., University of Colorado Denver, 2023

December 2023
University of Colorado Denver

This project report for the Master of Science degree by

Pargat Singh

to be approved for the

Computer Science Program

by

Haadi Jafarian

Date 3 December, 2023

Singh, Pargat (Master of Science, Computer Science)

INTEGRATING CVE DATA INTO SOFTWARE DEVELOPMENT LIFECYCLES FOR PROACTIVE VULNERABILITY MANAGEMENT

Thesis directed by Haadi Jafarian

ABSTRACT

There are numerous security flaws in contemporary software infrastructures that could lead to an attack. Researchers hope to create a secure software design process that starts at the architectural level. They want to make sure that security requirements are strong enough to support the application of appropriate security controls, which will help to mitigate known threats and vulnerabilities. In order to meet the difficulty of guaranteeing the sufficiency of security requirements, this study focuses on the use of external web data sources that contain vulnerability information, including CVE data, attack patterns, threat intelligence, and other security data. They use Natural Language Processing (NLP) to provide a report that helps designers verify if security needs are enough. This validation procedure includes determining which needs match existing risks that have been gathered from external data, locating any requirements that may be unnecessary, and flagging dangers that need to be examined more closely in order to reveal new requirements. The first section of the article describes the availability and attributes of external data. This is followed by a thorough discussion of how natural language processing (NLP) is used to process the data and produce the validation report. They give an example from real life to demonstrate the methodology.

This project report is approved for recommendation to the Graduate committee.

Project Advisor:

Haadi Jafarian

ACKNOWLEDGEMENTS

I extend my heartfelt gratitude to Professor Haadi Jafarian for his unwavering support and invaluable guidance throughout the entirety of this research endeavor. His constant encouragement, insightful advice, and profound knowledge in the field have been instrumental in shaping the trajectory of this paper. Professor Jafarian's commitment to fostering academic growth and providing constructive feedback have been pivotal in navigating the complexities of this study. His dedication to the pursuit of knowledge and his willingness to share expertise have significantly contributed to the completion of this paper. I am profoundly grateful for the time and effort Professor Jafarian invested in reviewing and providing feedback on my work. His mentorship has been a source of inspiration, and I feel fortunate to have had the opportunity to benefit from his wisdom and expertise. In addition, I would like to express my appreciation for the supportive academic environment that Professor Jafarian cultivated, which allowed me to explore new ideas and approaches in my research. This acknowledgement is a token of my sincere appreciation for Professor Haadi Jafarian's integral role in the successful completion of this paper.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 The Problem

The vulnerabilities present in today's software infrastructures provide room for bad actors to take advantage of them. With online apps accounting for 43% of 3,950 global data breaches, the Verizon 2020 Data Breach Investigations Report [2] emphasises how common web application intrusions are. Strong security measures are essential since, as evidenced by the retail industry breach data, 45% of hacking techniques involve the use of credentials that have been stolen. Notwithstanding this, there are still obstacles in the way of creating safe apps, such as limited time, a deficiency of knowledge and resources within development teams, and small rewards for giving security first priority. With an emphasis on thorough security requirements that successfully counter known threats and weaknesses, this study improves the designing process in order to address the issue of a shortage of security expertise and tools.

1.2 Motivation and Challenges

The motivation behind this study lies in the need to overcome obstacles in developing secure software. Pressures on commercial development teams, a lack of expertise, and insufficient incentives for prioritizing security create challenges. Current security-related failures in software development don't incur legal penalties, unlike other industries. Therefore, there is a clear need to enhance the design process to ensure security requirements are comprehensive, leading to secure and functional software.

1.3 Problem Statement

Development teams struggle to provide secure software because they lack security knowledge and resources. Furthermore, a major factor in the frequency of security vulnerabilities in modern software is the lack of incentives to prioritise security. By emphasising improving the process of designing and making sure security needs are adequate, this study seeks to address these issues.

1.4 Research Aims (or Objectives)

The primary aims of this research include:

1. Finding outside data sources that offer insightful threat and vulnerability data.

2. Elucidating the characteristics inherent in these data sources.

3. Explaining how data sourced from these entities can be effectively employed to establish robust security requirements.

These objectives help to address the problems found in the process of developing software that is safe.

1.5 External Data Sources

In this segment, they explore various outside sources of security-related data that are essential for supporting efforts to design secure systems. Although it is not a comprehensive list, their choice of sources of data are based on how widely they are used and how easily accessible they are to the general public.

1.5.1 A. Vulnerability Databases

Several organisations curate databases containing security threat and vulnerability data, with the Common Vulnerabilities and Exposures (CVE) list prominently featured, which catalogues publicly acknowledged security vulnerabilities. CVE entries are unique in that they include a detailed description as well as at least one public reference. The Common Weakness Enumeration (CWE), a community-driven inventory of software and hardware flaws, is another important data source. The National Vulnerability Database (NVD) [3] of the United States augments CVE information by providing standardised and analytical attributes, such as severity scores via the Common Vulnerability Scoring System (CVSS). Other databases, such as the WhiteSource Vulnerability Database [4], specialise in open-source vulnerabilities and aggregate data from multiple sources.

1.5.2 B. Adversarial Behavior Databases & Attack Patterns

Attack patterns define the characteristics and strategies that adversaries use to exploit vulnerabilities. The Common Attack Pattern Enumeration and Classification (CAPEC) and Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) frameworks from MITRE are critical resources [5] CAPEC provides detailed description of attack patterns, related attack patterns, execution flows, prerequisites, and mitigations. ATT&CK is a network defence book that goes over adversary operational phases and techniques. Both databases provide useful insights into adversarial behaviour, assisting system designers in understanding attack patterns and designing systems that are resistant to such exploits.

1.5.3 C. Sources of Threat Intelligence Data

The analysis and compilation of threat data results in threat intelligence, which is essential for cybersecurity initiatives. A commonly used standard called Structured Threat Information eXpression

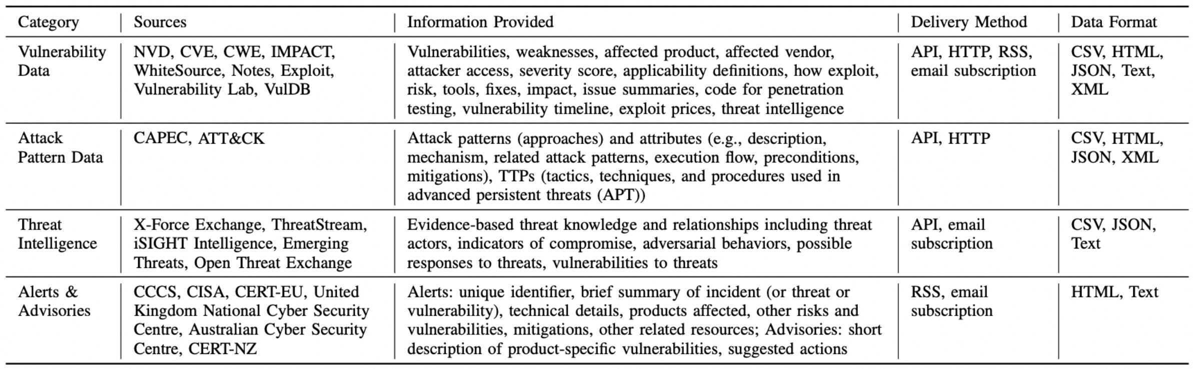
(STIX) [1] makes machine-readable cyber threat intelligence possible. This ecosystem includes open-source feeds like ProofPoint's Emerging Threats and commercial vendors like IBM X-Force Exchange. Threat intelligence helps define security requirements and helps identify critical assets. It includes threat actors, indicators of compromise, and attack patterns.

1.5.4 D. Government Agency Alerts and Advisories

Government departments around the world, such as the Canadian Centre for Cybersecurity and the United States Cybersecurity & Infrastructure Security Agency [6] [7], issue alerts and advisories about emerging threats. These communications include technical information, a list of affected products, and mitigation recommendations. While alerts provide detailed information, advisories provide concise information; however, both serve as valuable resources for infrastructure designers in identifying potential major assets and gathering data for information-driven design solutions.

1.5.5 Summary

A multitude of data is available from external internet sources that are essential for secure system design. Database Information summarises these sources in a categorised manner, describing the types of data they support, how they deliver data, and what kinds of information they offer.



Category	Sources	Information Provided	Delivery Method	Data Format
Vulnerability Data	NVD, CVE, CWE, IMPACT, WhiteSource, Notes, Exploit, Vulnerability Lab, VulDB	Vulnerabilities, weaknesses, affected product, affected vendor, attacker access, severity score, applicability definitions, how exploit, risk, tools, fixes, impact, issue summaries, code for penetration testing, vulnerability timeline, exploit prices, threat intelligence	API, HTTP, RSS, email subscription	CSV, HTML, JSON, Text, XML
Attack Pattern Data	CAPEC, ATT&CK	Attack patterns (approaches) and attributes (e.g., description, mechanism, related attack patterns, execution flow, preconditions, mitigations), TTPs (tactics, techniques, and procedures used in advanced persistent threats (APT))	API, HTTP	CSV, HTML, JSON, XML
Threat Intelligence	X-Force Exchange, ThreatStream, iSIGHT Intelligence, Emerging Threats, Open Threat Exchange	Evidence-based threat knowledge and relationships including threat actors, indicators of compromise, adversarial behaviors, possible responses to threats, vulnerabilities to threats	API, email subscription	CSV, JSON, Text
Alerts & Advisories	CCCS, CISA, CERT-EU, United Kingdom National Cyber Security Centre, Australian Cyber Security Centre, CERT-NZ	Alerts: unique identifier, brief summary of incident (or threat or vulnerability), technical details, products affected, other risks and vulnerabilities, mitigations, other related resources; Advisories: short description of product-specific vulnerabilities, suggested actions	RSS, email subscription	HTML, Text

Figure 1.1: Database Information [1]

1.6 Outline of the report

1.6.1 Introduction

1. Importance of Efficient Decision Support

2. Role of Security Controls

3. Challenges in Threat Modeling
4. Introduction of Data-Driven Approaches
5. Proposed Solution: Leveraging External Data Sources

1.6.2 Literature Review

6. Significance of Robust Security Assurance
7. Role of Threat Modeling
8. Challenges with Reliance on Threat Modeling Alone
9. Introduction of Data-Driven Approaches
10. Proposed Paradigm Shift

1.6.3 Methods

11. Introduction of NLP in External Data and Security Requirements
12. Illustration Using a College Library Web Application
13. UML Diagram of the Web Application
14. Five Sequential Steps of the Methodology
15. JSON Query for Security Requirements and Technologies
16. Extraction of Keywords using NLP
17. Retrieval of Threat Information
18. Examination of Threats and Requirements
19. Report Generation

1.6.4 Results and Discussions

20. Support for System Designers
21. Empowering Designers to Prioritize and Address Unmitigated Threats

22. Challenges Encountered during Method Development

23. Automation of Threat Classification and Categorization

24. Consistency in Data from Various Sources

1.6.5 Conclusions

25. Summary of Findings

26. Contributions of the Approach

27. Future Research Directions

    (a) Machine Learning-Based Named Entity Recognition

    (b) Improved Data Processing for Unmitigated Threats

    (c) Tool Support Enhancement for Automated Extraction

1.6.6 References

27. Citations of Relevant Works

CHAPTER 2
LITERATURE REVIEW

In the complex process of developing secure software-intensive systems, effective decision support is critical. Designers are responsible for embedding robust security controls to protect critical system assets while navigating tradeoffs between competing and sometimes conflicting system qualities. To substantiate security assurance efforts, meticulous documentation and justification are required [8]. To achieve robust security assurance, It is necessary to have a deep understanding of security requirements, threats, and controls, as is their seamless integration across all stages of development [9]. The practise of threat modelling, which is critical in eliciting security requirements and comprehending potential vulnerabilities and attack scenarios, is central to this understanding. Threat modelling facilitates informed security risk decision-making by examining an attacker's profile and potential attack routes methodically, and desired possessions [10], [11]. However, relying solely on threat modelling poses difficulties. The difficulty in determining whether identified threats manifest as real-world limitations or exploits complicates matters further. The effectiveness of threat identification is determined by the security knowledge and experience of those performing the modelling, which introduces subjectivity and bias into decision-making. [12] . Furthermore, the selection process is made more difficult by the multitude of threat modelling techniques available, as there is a lack of precise guidance regarding which approach is best for a given application.It becomes essential to change the paradigm and adopt data-driven strategies in order to overcome these obstacles. Utilising third-party internet data sources, like vulnerability databases, is recognised as an essential tactic. This approach's central question is how data-driven methodologies can promote more objective decision-making, helping in the early detection of weak trends, vulnerabilities, and potential targets by system developers. The suggested approach makes use of outside data to confirm that security requirements are sufficient. This entails a careful analysis of how well-aligned current requirements are with threats found in external data sources. Inconsistencies force requirements to be added to or adjusted based on the data, guaranteeing their sufficiency in the face of possible threats. Interestingly, this creative application of outside data to improve security requirement validation opens up a fresh and uncharted field in the body of current research.

CHAPTER 3
METHODS

This application facilitates book searches and requests for students, faculty, and librarians by providing online services. In addition, librarians have the ability to add books and users. For instance, they illustrate the web application's high-level architecture model using a UML class diagram, as seen in Figure 3.1. The software components in this diagram are represented by classes, the relationships between them by associations, and the initial security requirements (blue boxes) and design requirements (red boxes) are outlined by model annotations. These specifications aim to mitigate a particular group of possible security risks. The methodology consists of five sequential steps, as shown in Figure 3.2.

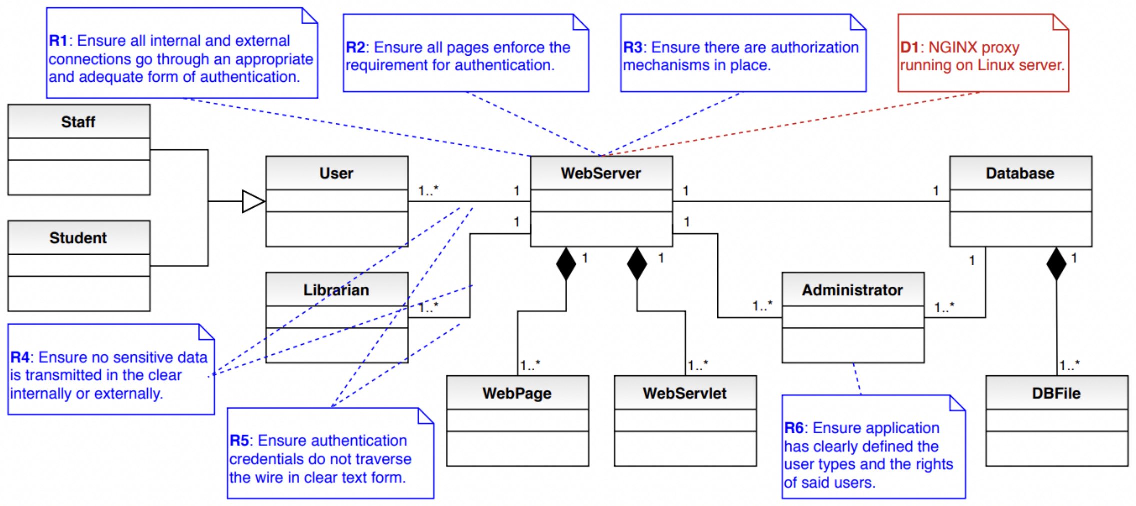

R1: Ensure all internal and external connections go through an appropriate and adequate form of authentication.
R2: Ensure all pages enforce the requirement for authentication.
R3: Ensure there are authorization mechanisms in place.
D1: NGINX proxy running on Linux server.
Staff
Student
User
WebServer
Database
Librarian
Administrator
WebPage
WebServlet
DBFile
1..*
1
R4: Ensure no sensitive data is transmitted in the clear internally or externally.
R5: Ensure authentication credentials do not traverse the wire in clear text form.
R6: Ensure application has clearly defined the user types and the rights of said users.

Figure 3.1: UML Diagram of Web Application [1]

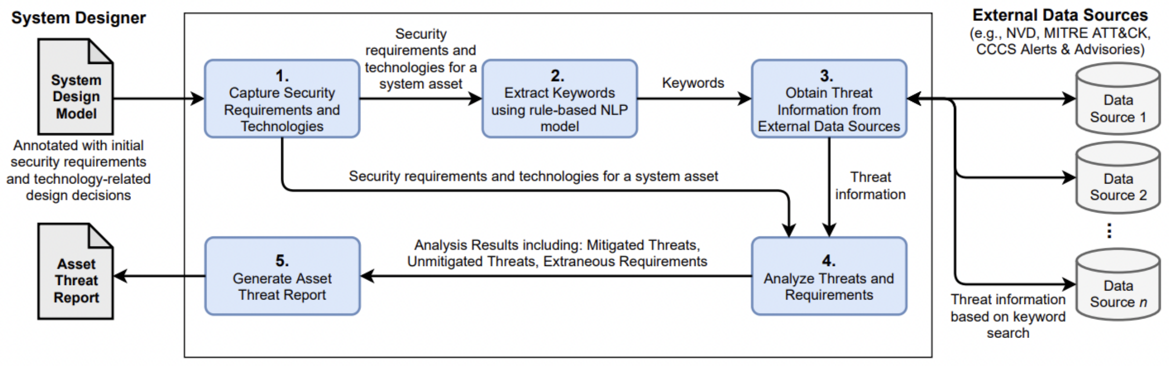
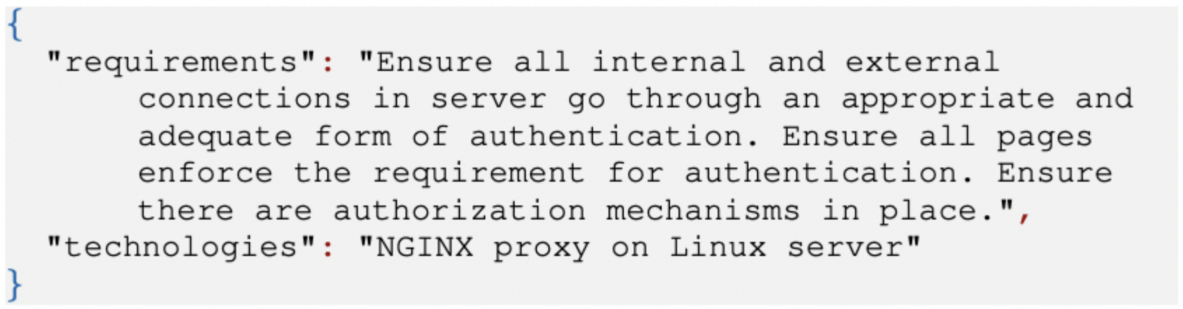

System Designer
System Design Model
Annotated with initial security requirements and technology-related design decisions
Asset Threat Report
1. Capture Security Requirements and Technologies
Security requirements and technologies for a system asset
2. Extract Keywords using rule-based NLP model
Keywords
3. Obtain Threat Information from External Data Sources
Security requirements and technologies for a system asset
Threat information
4. Analyze Threats and Requirements
Analysis Results including: Mitigated Threats, Unmitigated Threats, Extraneous Requirements
5. Generate Asset Threat Report
External Data Sources
(e.g., NVD, MITRE ATT&CK, CCCS Alerts & Advisories)
Data Source 1
Data Source 2
Data Source n
Threat information based on keyword search

Figure 3.2: A review of the recommended analysis technique that makes use of outside data sources to confirm that a set of security requirements is adequate [1]

3.1 Obtain Security Needs and Technologies:

At first, they gather security requirements and design specifications for a specific asset from system documentation, if available. Specifications or design models can be used to create this documentation. The information gathered aids in determining critical asset characteristics such as required security controls and associated technologies. It is assumed that an annotated model is available in a documented format, which makes it easier to formulate requirements into a JSON query for subsequent steps. They formulate the query, as shown in figure 3.3, using the WebServer asset from their ongoing example, recording technology-related design and security requirements (R1, R2, and R3 in Fig. 3.1).



```
{
  "requirements": "Ensure all internal and external
       connections in server go through an appropriate and
       adequate form of authentication. Ensure all pages
       enforce the requirement for authentication. Ensure
       there are authorization mechanisms in place.",
  "technologies": "NGINX proxy on Linux server"
}
```

Figure 3.3: JSON request for the WebServer asset in the college library application [1]

3.2 Extraction of Keywords:

The JSON query from Step 1 is processed through an NLP model in this phase. The requirements text is fed into a pre-trained NLP model capable of recognising lemmas, which group inflectional and derivational forms of a word into its base form, to identify relevant keywords. This process also extracts named entities, syntactic dependencies, and part-of-speech tags (like verbs and nouns). Lemmatization facilitates keyword capture and allows different inflectional forms to be accommodated within the constraints. A rule-based matcher is then used to filter the identified keywords, making sure that only those that are pertinent to the cybersecurity context are kept. The outcome of this process is a compiled list of keywords (D1 in Fig. 3.1) that represent the security and design specifications specified by the designer.

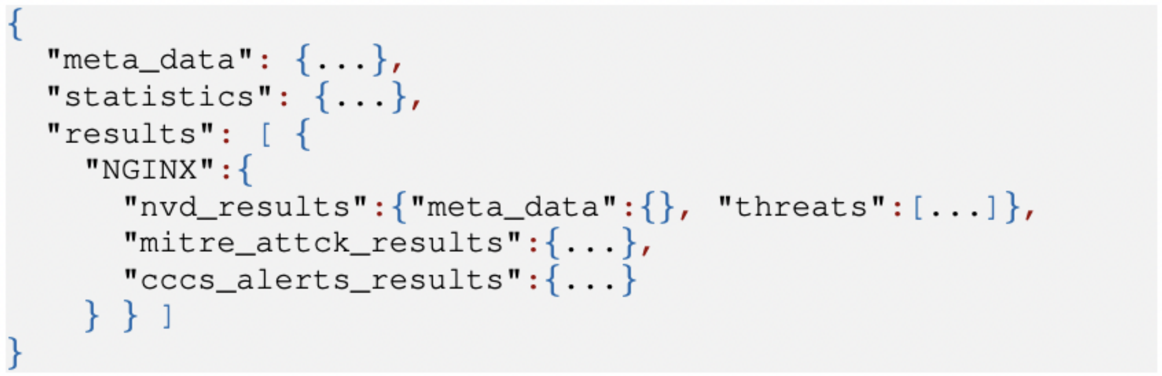
As they continue to illustrate themselves, the keywords that are derived from the technology-related design specifications and security requirements in figure 3.3 include terms like Linux, NGINX, authentication, and authorization. This assembled list of keywords forms the foundation for obtaining relevant threats from multiple external data sources, making it easier to discern connections between the threats that have been identified, security prerequisites, and design specifications. The precision of their subsequent search and analysis for asset assessment is directly influenced by the accuracy of the keyword extraction process. Although a generic NLP model off the shelf produced satisfactory results, it also extracted irrelevant data for their specific analysis. To address this, they developed a rule-based technique to improve extraction, paying special attention to technology-and security-related keywords relevant to the designer's data.

3.3 Gather Threat Data:

They delve into the intricate relationships between the identified security, threats requirements, and design specifications during this phase. Their methodology involves enhancing the data linked to every received threat by integrating insights obtained from the threat's description regarding the threat's influence on security goals and suggested security measures.

NLP extractors are used to search the threat description for cues that indicate a threat's impact on one or more security objectives in order to assess how a threat affects those objectives. If available, explicit data is used, such as the CVSS specification of a threat. Though other goals like authenticity and accountability might also be taken into consideration, their main focus is on the objectives of confidentiality, integrity, and availability.

The structure of the query response for the technology term "NGINX" linked to the WebServer asset in their hypothetical scenario is shown in figure 3.4.



```
{
  "meta_data": {...},
  "statistics": {...},
  "results": [ {
    "NGINX":{
      "nvd_results":{"meta_data":{}, "threats":[...]},
      "mitre_attck_results":{...},
      "cccs_alerts_results":{...}
    } } ]
}
```

Figure 3.4: JSON search result for the "NGINX" keyword associated with the WebServer asset [1]

3.4 Examine Threats and Requirements:

They delve into the intricate relationships between the identified threats, security requirements, and design specifications during this phase. Their approach entails improving the information associated with each received threat by incorporating knowledge derived from the threat's description about the threat's effect on suggested security measures and security objectives.

NLP extractors are used to scan the threat description for cues indicating the threat's effect on one or more security objectives in order to determine the impact of a threat on security objectives. If available, explicit information, such as a threat's CVSS specification, is used. Their consideration is primarily focused on confidentiality, integrity, and availability goals, though other goals such as authenticity and accountability may also be considered.

Utilising a special NLP extractor variant set up to identify technical controls described in ISO/IEC 27001, the security controls of the threats are extracted [13]. Three results are obtained when the initial security requirements are compared with this list of controls:

(1) Threats that are countered or dealt with by the specified security requirements, meaning they are effectively managed or mitigated,

(2) Threats that are deemed unmitigated because they do not adhere to the required security standards, and

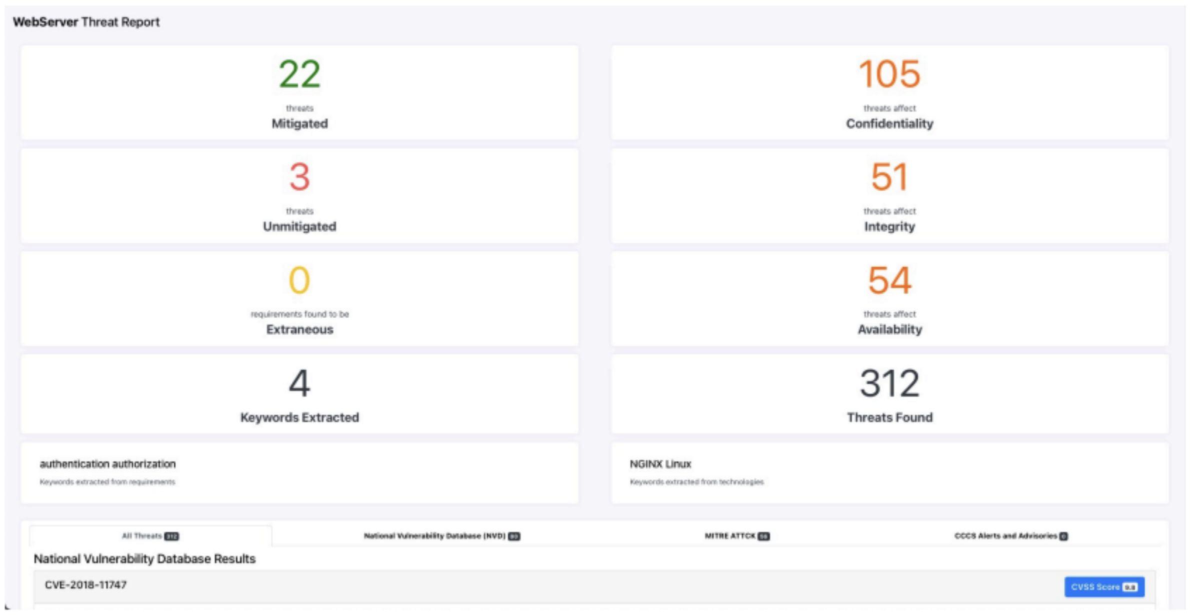
(3) Given security specifications free of related risks (i.e., potentially superfluous requirements).

For example, they compute the set difference between the security requirements supplied by the designer and the set of requirements derived from external threat data related to the design specifications in order to identify unmitigated threats. This result reflects threats related to design, for which a new or altered security requirement might be required.

3.5 Report Generation:

They generate an interactive report for their college library WebServer asset using Step 4 of their methodology, as shown in Fig. 3.5. The primary goal of this report is to assist designers in more effectively navigating a vast amount of information. Based on the initial security and design requirements, the threat analysis report identifies 312 threats, categorising them as 105 affecting confidentiality, 51 affecting integrity, and 54 influencing WebServer availability. While not stand-alone solutions, these counters assist the designer during the system design phase in assessing the impact of known threats on various security objectives.

It's important to note that the 312 total threats represent those with enough information for actionable insights. The report goes over the keywords extracted from design and security requirements in greater detail and summarises the ties between requirements and threats from Step 4. Regarding design requirements technologies, it finds 22 mitigated threats, 3 unmitigated threats, and no superfluous requirements. Threats that have been mitigated correspond to the security controls "authentication" and "authorization," whereas the unmitigated threats point to additional security controls, specifically logging and two-factor authentication.



WebServer Threat Report

22
threats
Mitigated

105
threats affect
Confidentiality

3
threats
Unmitigated

51
threats affect
Integrity

0
requirements found to be
Extraneous

54
threats affect
Availability

4
Keywords Extracted

312
Threats Found

authentication authorization
Keywords extracted from requirements

NGINX Linux
Keywords extracted from technologies

All Threats 312 | National Vulnerability Database (NVD) | MITRE ATTCK | CCCS Alerts and Advisories

National Vulnerability Database Results

CVE-2018-11747

Figure 3.5: Report on threats discovered in the WebServer asset of the college library [1]

CHAPTER 4
RESULTS AND DISCUSSIONS

Their technique helps designers of systems identify known threats from outside data sources that may require new or modified requirements to improve system security. Furthermore, it assists in identifying requirements that may be extraneous for the system and do not appear to address or mitigate any known threats. When faced with financial and scheduling limitations, this kind of information enables designers to prioritise addressing unmitigated threats, thereby enhancing the effectiveness of development. Even though some of the threats they found might not even be relevant to the system, their method condenses a lot of data into a way that designers can easily use. For example, the amount of information returned by external data sources when a keyword search is conducted, like NGINX, can often be too large for designers to sort through for each component of their system. Their method selectively presents threats that affect the asset for which no security controls have been specified in the security requirements in order to reduce this volume of data. As a result, their method offers designers in these fields vital support.While formulating the suggested strategy, they ran into a number of obstacles. One significant obstacle was handling the previously mentioned massive data volume. In order to minimise response time for results, we came up with a script to automatically classify and categorise threats from external sources into a database. One more difficulty was inconsistent data coming from different sources.In order to address this, they developed a unified, simple schema, as shown in figure 3.3, to collect sufficient data in an adaptable format for additional analysis and applications. The calibre and precision of the given security and design specifications determine how effective the strategy will be. If the input is poor, the method might not produce results that are up to par. Lack of information in search queries to external data sources can make it difficult to extract keywords and perform threat and requirements analysis, which can lead to unsatisfactory results.

CHAPTER 5

CONCLUSIONS

5.1 Summary

In order to support system designers in verifying the adequacy of their needs for security and supporting them in making knowledgeable design choices, they looked into the use of outside data sources, such as vulnerability databases, threat intelligence data feeds, attack patterns, and security alerts and advisories. This facilitates the development of systems with suitable controls that can mitigate known threats and vulnerabilities that impact system components. They found and described external data sources that are currently in use and offer helpful information for this purpose. Furthermore, they presented an approach of NLP that drew on existing design documentation to address initial design and security requirements. The method generates a report that summarises which threats to the system design which is identified from external data are addressed by the current security specifications, which threats are unaffected by the current security specifications, and which requirements may be considered extraneous. With the use of this data, system designers can decide how best to address risks and vulnerabilities, leading to the creation of new or modified requirements.

5.2 Contributions

In order to support secure software design activities and address security concerns early in the Software Development Life Cycle (SDLC), especially during the stage of architecture design when changes are less expensive, their approach emphasises the significance of external threat, vulnerability, and attack data. System designers can use an iterative approach to allow for gradual progress and refinement of the system design as they make changes and add security requirements that are new. This supports security evaluation and assurance activities by increasing confidence that the most recent set of security requirements is sufficient.

5.3 Future Research

Their ultimate objective is to develop a machine learning-powered domain-specific Named Entity Recognition (NRE) that is capable of identifying a larger variety of objects and security guidelines. This project offers a more context-specific cybersecurity natural language processing model with the goal of enhancing the threat analysis and keyword extraction processes. To better manage the volume of processed data, they intend to enhance their ability to identify unmitigated threats and look into better mechanisms to produce an actionable set of results that only present the most pertinent threats to the system designer. By automating the extraction of system design model annotations and including more information on the dependencies and relationships among system assets, they also aim to enhance the approach's tool support. This enhancement will enable a more thorough analysis of the system architecture, assisting system designers in building safe systems that can withstand a variety of risks and assaults.

REFERENCES

[1] J. Samuel, J. Jaskolka, and G. O. Yee, "Leveraging external data sources to enhance secure system design," in *2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS).* IEEE, 2021, pp. 1–8.

[2] B. Verizon, "Data breach investigations report," *http://www. verizonbusiness. com/resources/security/databreachreport. pdf*, 2008.

[3] A. Regenscheid and K. Scarfone, "Recommendations of the national institute of standards and technology," *NIST special publication*, vol. 800, p. 155, 2011.

[4] W. Software. (2020) Whitesource vulnerability database. Accessed: Dec 2020. [Online]. Available: https://www.whitesourcesoftware.com/vulnerability-database/

[5] S. Barnum, "Common attack pattern enumeration and classification (capec) schema," *Department of Homeland Security*, 2008.

[6] B. J. Arnold, "Cyber security in canada: Structure and challenges," *Governing Cyber Security in Canada, Australia and the United States*, p. 5, 2018.

[7] Z. Lanz, "Cybersecurity risk in us critical infrastructure: An analysis of publicly available us government alerts and advisories," *International Journal of Cybersecurity Intelligence & Cybercrime*, vol. 5, no. 1, pp. 43–70, 2022.

[8] C. B. Weinstock and H. F. Lipson, "Evidence of assurance: laying the foundation for a credible security case," *Software Engineering Institute Report*, 2013.

[9] J. Jaskolka, "Recommendations for effective security assurance of software-dependent systems," in *Intelligent Computing: Proceedings of the 2020 Computing Conference, Volume 3.* Springer, 2020, pp. 511–531.

[10] A. Shostack, *Threat modeling: Designing for security.* John Wiley & Sons, 2014.

[11] T. UcedaVelez and M. M. Morana, *Risk Centric Threat Modeling: process for attack simulation and threat analysis.* John Wiley & Sons, 2015.

[12] J. Samuel, K. Aalab, and J. Jaskolka, "Evaluating the soundness of security metrics from vulnerability scoring frameworks," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom).* IEEE, 2020, pp. 442–449.

[13] "Iso/iec 27000:2018 information technology – security techniques – information security management systems – overview and vocabulary," https://www.iso.org/standard/62342.html, February 2018.