International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 2, Mar-Apr 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Real-Time Vulnerability Intelligence and Mitigation System for IT Infrastructure

Mrs.Sowmyashree C S[1], Mr. S Guruprasad[2], Mr. Sreejith T S[3], Mr. Sujay R[4], Mr. Ravi[5]
[1](Assistant professor, Department of Information Science and Engineering, Sambhram Institute of Technology, Bangalore, Email: sowmyashree2724@gmail.com)
[2](Department of Information Science and Engineering, Sambhram Institute of Technology, Bangalore, Email: guruprasad2903@gmail.com)
[3](Department of Information Science and Engineering, Sambhram Institute of Technology, Bangalore, Email: sreejithts260403@gmail.com)
[4](Department of Information Science and Engineering, Sambhram Institute of Technology, Bangalore, Email: sujayr799@gmail.com)
[5](Department of Information Science and Engineering, Sambhram Institute of Technology, Bangalore, Email: sravim2002@gmail.com)

ABSTRACT-- The proposed web scraping tool enhances cybersecurity by continuously monitoring and reporting critical IT and OT vulnerabilities in a critical sector organization. It automates the extraction of vulnerability details from OEM websites and relevant platforms, reducing reliance on conventional databases with potential reporting delays. Key data, including product name, OEM, severity, description, mitigation strategies, and publication date, is systematically collected and structured for analysis. Upon detecting new vulnerabilities, the tool promptly notifies recipients via email, facilitating rapid response. By minimizing manual effort and enabling real-time updates, this automation-driven approach strengthens IT and OT security resilience.

Keywords-- Cybersecurity, Web Scraping, IT and OT Security, Vulnerability Monitoring, OEM Websites, Automation, Real-time Threat Intelligence, Critical Sector Security, Vulnerability Detection, Incident Response.

I. INTRODUCTION

Critical sector organizations, including healthcare, finance, transportation, and energy, rely on IT and OT systems for essential operations. However evolving technologies introduce new vulnerabilities, increasing risks of data breaches, operational disruptions, and financial losses. Timely access to vulnerability information is crucial to mitigate these risks.

Traditionally, vulnerability data is aggregated Vulnerability Database (NVD), which, despite its standardized reporting, suffers from update delays. Such delays hinder rapid response in critical sectors where immediate action is necessary to safeguard

To address this challenge, the proposed web scraping tool automates the collection of vulnerability data directly from Original Equipment Manufacturer (OEM) websites and other sources. By continuously monitoring critical and high-severity vulnerabilities in ICT components, the tool extracts key details such as

product names, severity levels, mitigation strategies, and publication dates. It then delivers real-time alerts to designated recipients, enabling proactive threat mitigation.

This solution enhances the cybersecurity posture of critical infrastructure by minimizing reporting delays and facilitating swift responses to emerging threats. Through automation, the tool ensures security teams remain informed, strengthening overall resilience against potential cyber risks.

II. OBJECTIVES

The purpose of this research is to design an automated web scraping tool that will provide critical sector organizations with real-time, actionable insights into vulnerabilities affecting their IT and OT infrastructure. This is done by overcoming the delays inherent in traditional vulnerability reporting methods to enhance cybersecurity resilience through continuous monitoring, direct data extraction, and automated alerts. This establishes a real-time monitoring system to scan OEM sites and associated channels. That way, this tool reduces delays in the report of new emerging threats, with direct extraction from OEM sources of vulnerability data to minimize the conventional databases that normally rely on them, like NVD. In addition, an automated alerting system will alert stakeholders via email or SMS, providing critical details such as affected products, severity ratings, and mitigation recommendations.

The tool will further support proactive risk management by enabling cybersecurity teams to prioritize vulnerabilities based on risk analysis, ensuring efficient threat response. Historically based trend analysis and reporting increase the robustness of long-term planning of cybersecurity for organizations.

III. PROBLEM STATEMENT

In today's rapidly changing technological landscape, organizations face significant risks stemming from vulnerabilities in both Information Technology (IT) and Operational Technology (OT) environments. As cyber threats become increasingly sophisticated, the timely identification and reporting of critical and high-severity vulnerabilities are essential for effective risk management and mitigation. However, many organizations encounter several challenges that hinder their ability to effectively monitor and respond to these vulnerabilities. Firstly, most vulnerability management activities are based on manual monitoring, which mainly involves laborious checks of several OEM websites. It is very time-consuming and subject to human errors, hence delay in discovering newly identified vulnerabilities, as well as increasing chances for malicious use by malicious people. Secondly, most OEM websites include dynamic web content, making the extraction of data difficult.

More importantly, due to the availability of modern web technologies like JavaScript, traditional scrapping methods would no longer work in many scenarios. This situation leads to partially missing or stale information about the vulnerability, resulting in a flawed security posture from organizations. Further, anti-scraping practices undertaken by a site make consistent data gathering significantly harder. Many sites deploy mechanisms designed to block automated tools, making it challenging for organizations to collect the necessary data without being detected or blocked. This creates a critical need for solutions that can navigate these challenges effectively. Additionally, organizations often struggle with data overload due to the vast amount of information available online. This large volume overwhelms security teams, making it challenging to focus on the vulnerabilities that are critical and relevant to the organization.

IV. Related Work

The CVE system is the globally accepted framework for naming and classifying software and hardware vulnerabilities. This centralized repository facilitates research, supports vulnerability management, and tracks compliance.

The CVSS categorizes vulnerabilities by their severity, assisting organizations in determining where to first prioritize mitigation. However, a major drawback to the CVE system is that its data updates rely on vendor submissions and verification processes, which result in delays when updating vulnerability information and prevent an organization from being alerted to a critical vulnerability at the right time.

The National Vulnerability Database, which is managed by the National Institute of Standards and Technology, enhances the CVE system by adding metrics such as impact scores and exploitability ratings to vulnerability data. The NVD offers advanced search capabilities to assist organizations in finding vulnerabilities specific to their environment. However, similar to CVE, it relies on vendor submissions for updates, thus experiencing similar delays, and its generalized approach may not meet the needs of every organization.

The proposed web scraping tool is aimed at overcoming the mentioned limitations through real-time monitoring and reporting of vulnerabilities. It will collect data directly from OEM sites and other sources, avoiding the delay in CVE and NVD updates. The tool will structure key vulnerability details, such as product name, OEM, CVE ID, severity level, description, and mitigation strategies, and provide organizations with timely and actionable intelligence.

V. LITERATURE REVIEW

The first paper by R.N. R. S and V. M., titled "Web Scraping Tools and Techniques: A Brief Survey" [1], provides a comprehensive overview of the methodologies involved in web scraping. It discusses various tools available for data extraction, their specific applications, and evaluates the challenges faced when extracting data from modern websites. The authors emphasize the importance of selecting appropriate tools based on the specific requirements of the scraping task, considering factors such as the complexity of the target website and the nature of the data being extracted.

Building on this foundation, A. S. Bale et al., in their paper "Web Scraping Approaches and Their Performance on Modern Websites" [2], delve deeper into different web scraping approaches. This research evaluates the performance of various techniques specifically on contemporary websites that often employ dynamic content and complex structures. The authors compare these techniques in terms of efficiency, accuracy, and adaptability to changes in website layouts. The findings underscore practical aspects of web scraping in real-world applications, providing valuable insights for practitioners looking to optimize their data extraction processes.

Ahluwalia, A., and Wani, S. (2024). "Leveraging Large Language Models for Web Scraping"
This paper attempts to investigate the relevance of using Retrieval-Augmented Generation (RAG) models in enhancing data extraction from the web. [3]. The authors stressed the limitations of conventional web scraping techniques with dynamic content and issues in directly applied data extraction applications. The authors leverage large language models to propose a solution that enhances data collection by generating contextual information from web pages, thus overcoming traditional scraping bottlenecks. This study explores the intersection of natural language processing and web scraping, focusing on the benefits of RAG models in generating meaningful, structured data from complex, dynamic web content, making it a significant contribution to real-time web data extraction for cybersecurity applications.

Xu, Z., Liu, Z., Yan, Y., Liu, Z., Xiong, C., and Yu, G. 2024. "Cleaner Pretraining Corpus Curation with Neural Web Scraping"
The paper [4] presents the neural web scraper tool called NeuScraper to effectively scrape out clean text contents from complex web pages for the purpose of language model pretraining. The authors discuss the challenges posed by noisy data and the importance of preprocessing high-quality,

structured text for use in training deep learning models. The NeuScraper, therefore, uses advanced techniques that are able to not only extract the data but filter out irrelevant or noisy content to ensure that the corpus from this step is clean enough for model training. This tool has practical implications in enhancing the quality of training datasets for AI applications, especially in NLP tasks.

Xu, Z., Liu, Z., Yan, Y., Liu, Z., Xiong, C., & Yu, G. (2024). "Anywhere: A Web Crawler Automation Management Interface"
This paper [5] introduces the "Anywhere" web crawler automation management interface, which is meant to simplify the management of web scraping tasks. The tool provides an easy-to-use interface for structuring and automating web crawling as well as data extraction processes, allowing users to schedule, monitor, and control web crawlers easily. The integration of this interface with the Neu Scraper framework is discussed in the paper on how the process of large-scale extraction from the web can be simplified and automated. The "Anywhere" interface ensures web scraping operations are efficient, customizable, and scalable. It can, therefore, be a great resource for one seeking to collect and manage large datasets from various web sources for applications in several domains.

Dalbeattie, M., Dobberstein, C., Breiding, A., & Akbik, A. (2024). "Fundus: A Simple-to-Use News Scraper Optimized for High-Quality Extractions"
This paper [6] introduces Fundus, a web scraping tool optimized for extracting high-quality data from news websites. Fundus is designed to be simple to use while delivering accurate and reliable extraction of articles and news data. The authors focus on improving the efficiency of data extraction by integrating advanced techniques that filter out irrelevant content, ensuring that the data collected is of high quality. The paper demonstrates how Fundus can be applied to scrape news websites in real time, and thus it becomes a useful tool for applications like sentiment analysis, trend monitoring, and news aggregation. The simplicity of the design of the tool makes it accessible to both non-experts and experts in the field of web scraping.

In addition to these studies [7], understanding the context of vulnerabilities is crucial for this project. The National Vulnerability Database (NVD) serves as a foundational resource for comprehending data formats related to vulnerabilities, particularly Common Vulnerabilities and Exposures (CVE) identifiers. However [8], it is essential to note that while NVD provides standardized vulnerability data, it can experience delays in updating information. This delay can hinder timely responses to emerging threats, making it imperative for organizations to seek alternative methods.

VI. METHODOLOGY

The proposed web scraping tool is designed to make the monitoring and reporting of vulnerabilities easy by utilizing open-source APIs and Large Language Models (LLMs). Multi-phase in its design, the system ensures the accurate and timely delivery of vulnerability data. The first phase, Data Collection and Monitoring, systematically scrapes data from OEM sites and other relevant platforms that publish vulnerability information. This will ensure continuous monitoring of high-severity vulnerabilities that would affect the IT and OT infrastructures. Advanced scraping techniques, such as rendering, proxy rotation, and handling anti-scraping mechanisms to really extract dynamic content, will be utilized in this tool. The tool will structure key vulnerability details such as product name, OEM name, CVE ID, vulnerability description, mitigation strategies, and publication date, ensuring that security teams can easily interpret and prioritize threats.

The Automated Reporting and Notifications module will focus on the development of a real-time reporting system based on the Python smtplib library for sending out email notifications. These will contain detailed vulnerability reports with all necessary information so that stakeholders are

always notified about a potential threat as soon as it arises. In addition, it will allow email addresses to be customized as well as triggers of severity levels for notification.

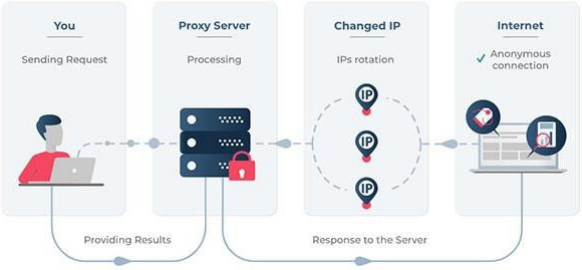

Fig.1.1 How Proxy Servers Enable Anonymous Internet Browsing with IP Rotation

Data Collection and Monitoring, systematically scrapes data from OEM sites and other relevant platforms that publish vulnerability information. This will ensure continuous monitoring of high-severity vulnerabilities that would affect the IT and OT infrastructures.
Advanced scraping techniques, such as rendering, proxy rotation, and handling anti-scraping mechanisms to really extract dynamic content, will be utilized in this tool.

Data Processing and Filtering phase, the collected data will be processed and filtered using LLMs and API support to categorize vulnerabilities based on severity levels, prioritizing critical and high-severity threats. The tool will structure key vulnerability details such as product name, OEM name, CVE ID, vulnerability description, mitigation strategies, and publication date, ensuring that security teams can easily interpret and prioritize threats.

Automated Reporting and Notifications module will focus on the development of a real-time reporting system based on the Python smtplib library for sending out email notifications. These will contain detailed vulnerability reports with all necessary information so that stakeholders are always notified about a potential threat as soon as it arises. In addition, it will allow email addresses to be customized as well as triggers of severity levels for notification.

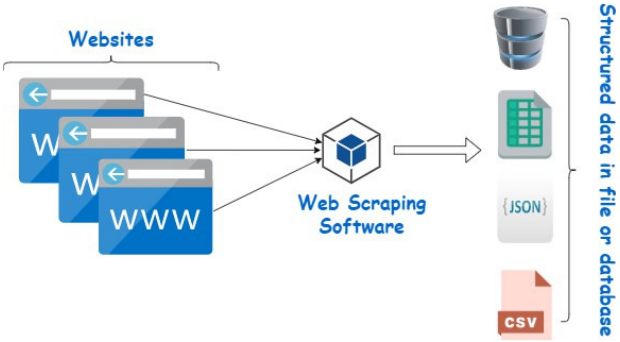

Fig.1.2 Web scrapping "Extracting, processing, and storing web data."

Automation and Maintenance, the Windows Task Scheduler will be used to automate the execution of the scraping process at predefined intervals, such as daily or weekly. This ensures continuous data collection without requiring manual intervention, reducing downtime and improving efficiency. By integrating automation into the workflow, the tool enhances operational reliability while maintaining up-to-date vulnerability intelligence.

VII. SYSTEM ARCHITECTURE

Task Scheduling and Training Process: The system first starts with a task scheduling mechanism that periodically initiates a training process referred to as "Topper-fits." This would mean that the system is supposed to handle recurrent tasks, perhaps for model training or data processing, to keep the system up-to-date with the latest data or models. Build Phase and Web Creation After scheduling, the system goes into a build phase where web creation and web spidering activities are performed. The Fire crawl API is probably used to perform web spidering, which collects data from several web sources. This phase is important for gathering raw data that will be processed in subsequent stages.

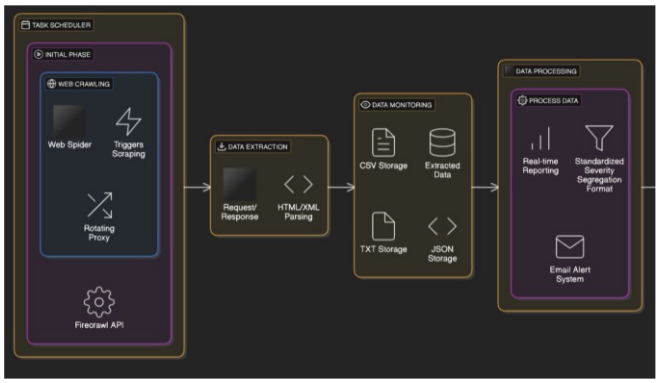

Fig.1.4 End-to-End Data Monitoring System with Web Scraping and Real-Time Reporting

OEM and Web Data Handling: The system interacts with an OEM (Original Equipment Manufacturer) component labeled "webpage." This component might be responsible for managing web pages or handling web-related data. The system processes requests and responses, indicating a client-server interaction model where data is requested from web sources and responses are received for further processing.

Data Extraction and Processing: The extracted data is passed through a transformation process where HTML/XML data is parsed and processed using an auto-generated tool or library as called "LLLR". In such a process, raw web data is transformed into structured formats to allow easy analysis and storage.

Data Storage: The processed data is then stored in a data storage system labeled "CSWASON1X7D." This storage solution is probably a database or a data warehouse designed to handle large volumes.

Proper data storage ensures that the data is accessible for future analysis and reporting.

Severity Level and Base Score Segregation: The system contains a module that segregates data based on severity levels and base scores. This is a very important step because it helps the system to prioritize data or tasks based on their importance or impact, thereby handling critical data with higher priority.

It applies standardized data format and regex during the process of registration and data extraction for a smooth and proper processing system and also maintains consistency. This consistency allows it to ensure integrity with data and, further, provides easier query handling and analysis operations with the data.

Email Alert System and Real-Time Reporting: The system finally includes an email alert system that supports TLS and SSL for secure communication. This system provides real-time email reporting, ensuring that stakeholders are promptly informed about important events or updates. Real-time reporting is essential for timely decision-making and maintaining operational efficiency.

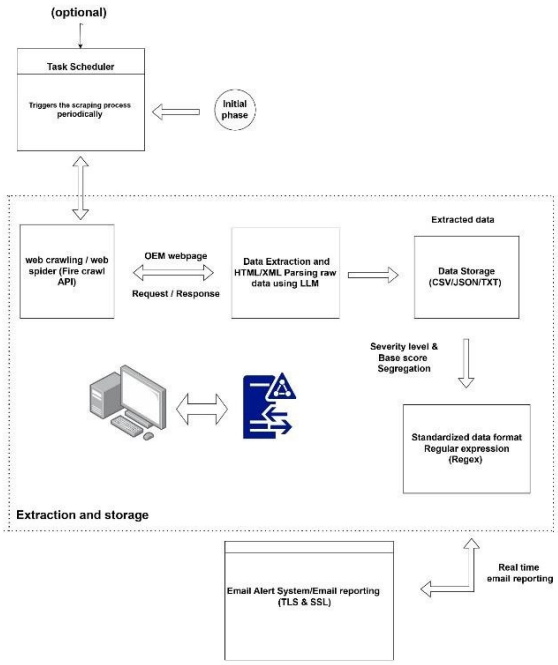

Fig.1.5 Workflow for Automated Web Scraping and Vulnerability Reporting System

VIII. RESULT AND ANALYSIS

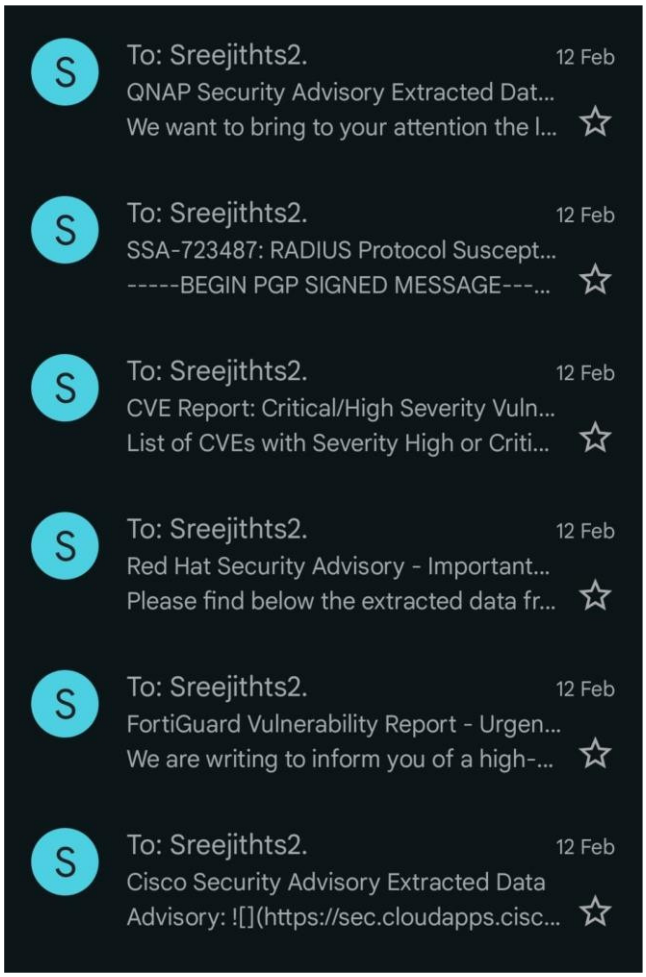

Fig.1.6: Screenshot showing a centralized inbox receiving high and critical severity vulnerability alerts from multiple OEM sources such as QNAP, Red Hat, Cisco, and FortiGuard, highlighting automated data extraction and monitoring.

The **fig.1.7** presents a FortiGuard vulnerability email alert reporting two high-severity CVEs affecting FortiADC systems. The vulnerabilities, CVE-2023-36554 and CVE-2023-42788, impact Forti Manager and Forti Analyzer products respectively, across multiple versions. Each alert includes CVE ID, product details, severity level, and the update date. Such notifications play a critical role in proactive security measures within organizational infrastructure.

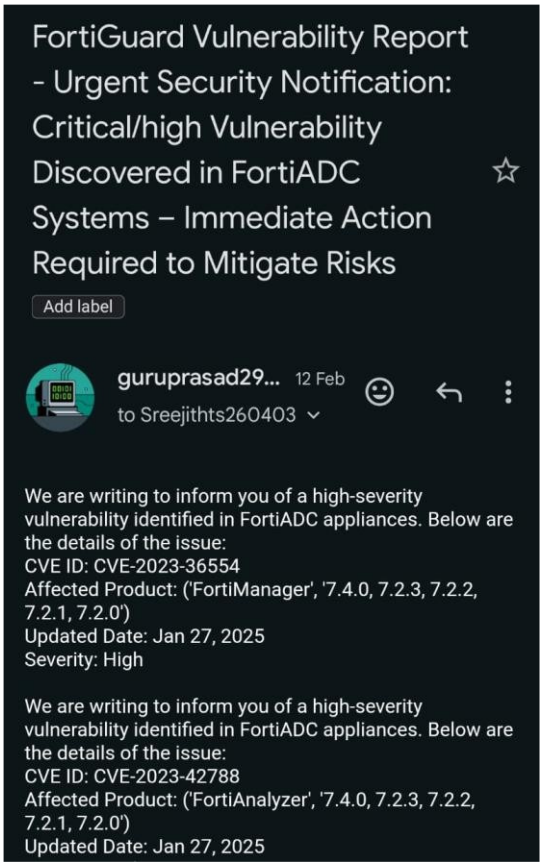

Fig.1.7: Email notification from FortiGuard disclosing two high-severity vulnerabilities (CVE-2023-36554 and CVE-2023-42788) found in FortiADC systems, advising immediate action to mitigate potential risks

The **fig.1.8** displays a QNAP security advisory email that highlights a high-impact vulnerability identified in Clam AV by OSS-Fuzz. The advisory, tagged as CVE-2025-20128 and internal ID QSA-25-04, urges immediate attention to mitigate risks. It warns users about multiple vulnerabilities in QNAP systems, some classified as Important and Moderate. The last update for this CVE was recorded on January 28, 2025.

It alerts users to multiple vulnerabilities affecting QNAP systems, with varying severity from Important to Moderate.The vulnerability could potentially allow attackers to exploit services using Clam AV for malware scanning.

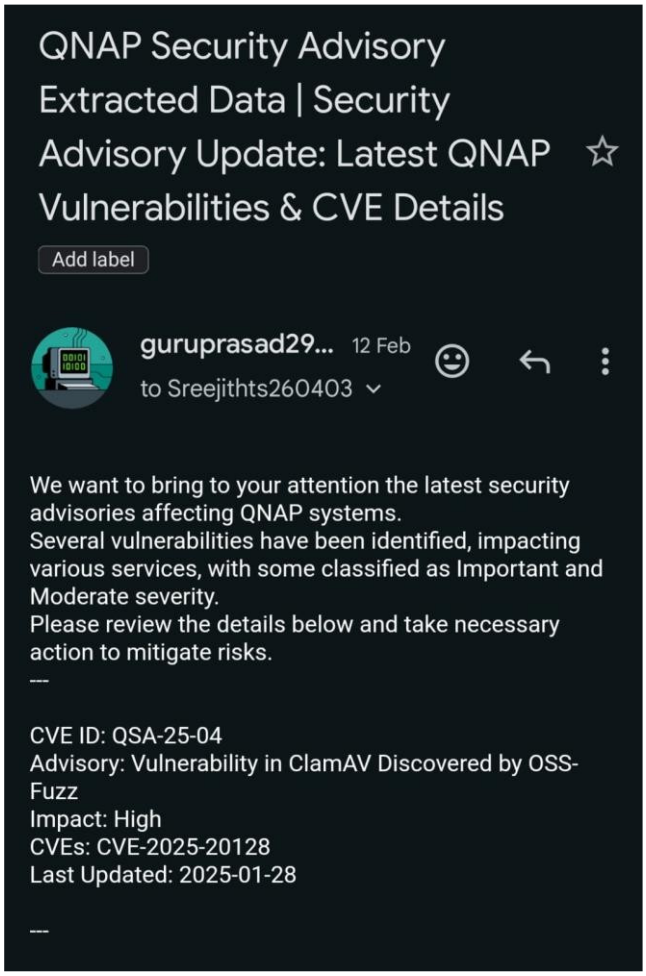

Fig 1.8: Email alert from QNAP Security Advisory outlining a high-impact ClamAV vulnerability (CVE-2025-20128) discovered by OSS-Fuzz, urging users to assess the advisory (QSA-25-04) and take prompt action to prevent system exploitation.

The Cisco security advisory outlines critical vulnerabilities in Cisco Identity Services Engine This advisory, referencing CVE-2025-20124 and CVE-2025-20125, has a **high impact rating**. The vulnerabilities include issues related to access control (CWE-285) and serialization (CWE-502). Cisco released this advisory on February 5, 2025, with the latest update on February 10, 2025. The flaws could allow unauthorized access or manipulation of sensitive data within ISE environments.
Cisco has published remediation steps and CSAF (Common Security Advisory Framework) doc

Users are advised to apply the free software updates and monitor future security alerts.

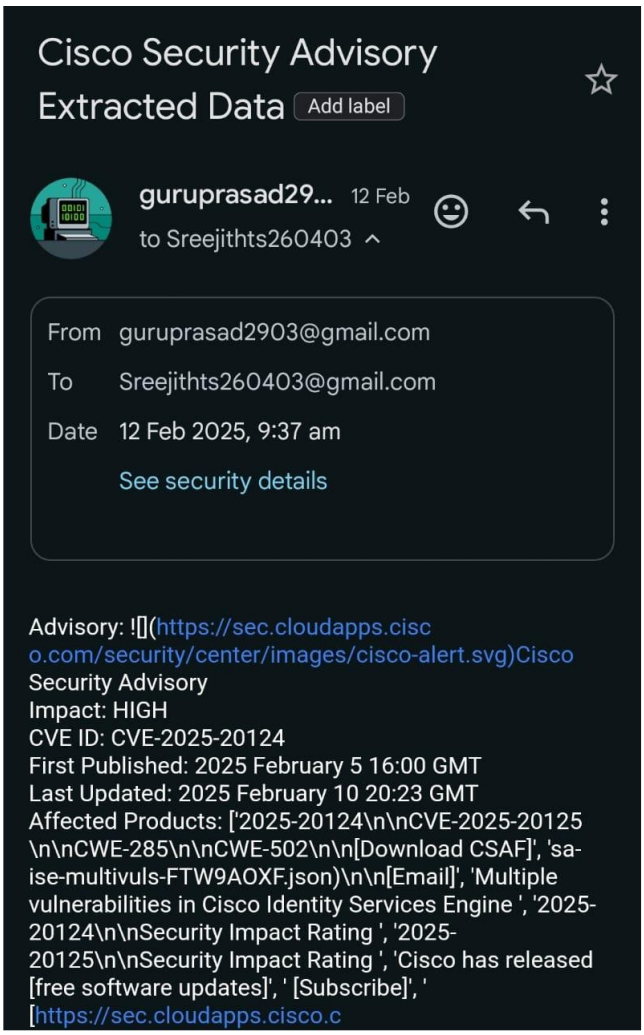

Fig 1.9: It highlights a **critical vulnerability** (CVE-2025-20124) found in Cisco Identity Services Engine, marked with **High Impact**.

The email highlights a **critical vulnerability** in Oracle Solaris, identified as **CVE-2024-53899**, with a **base CVSS score of 9.8**. The flaw is related to the third-party component virtual env and uses the HTTP protocol as its attack vector. It carries **High** impact ratings across confidentiality and integrity, and can be exploited remotely over a network

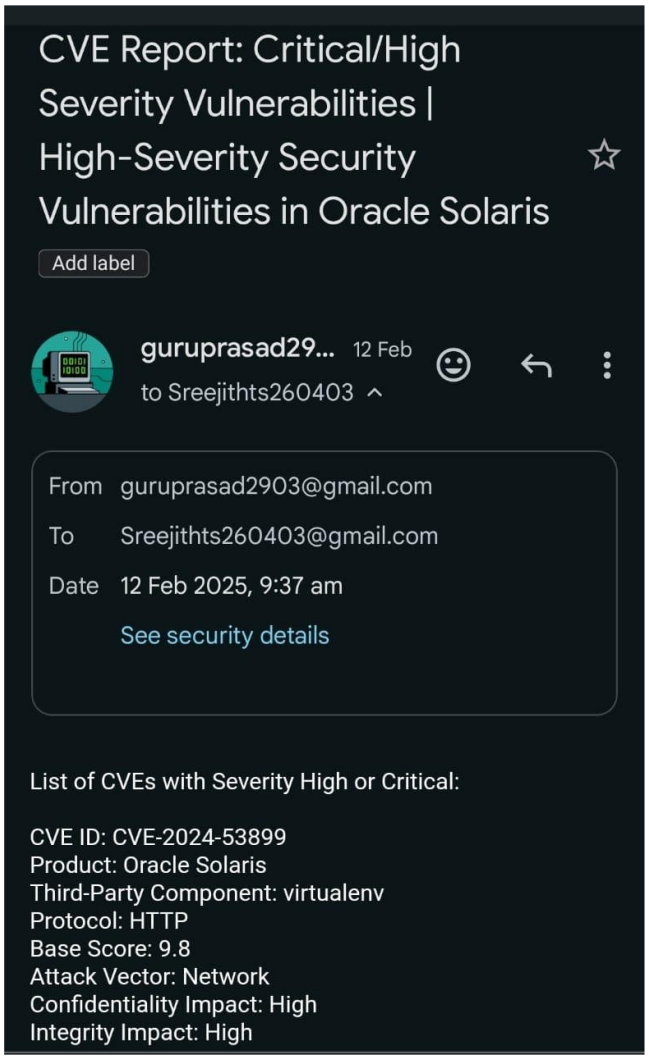

Fig.1.10: Screenshot of a security advisory email reporting CVE-2024-53899 in Oracle Solaris, with a CVSS score of 9.8. The vulnerability impacts the virtual env component and is exploitable via network using the HTTP protocol.

This email contains a **vulnerability advisory (SSA-723487)** concerning the **RADIUS protocol**, specifically **CVE-2024-3596**, which is susceptible to forgery attacks. The issue affects **FortiGate NGFW devices (version < V7.4.3)** used on **Siemens RUGGEDCOM APE1808** systems. The CVSS v3.1 base score is **9.8**, indicating a **critical severity**, while the CVSS v4.0 base score is **8.7**. The vulnerability impacts authentication and communication mechanisms in FortiOS-based firewalls.

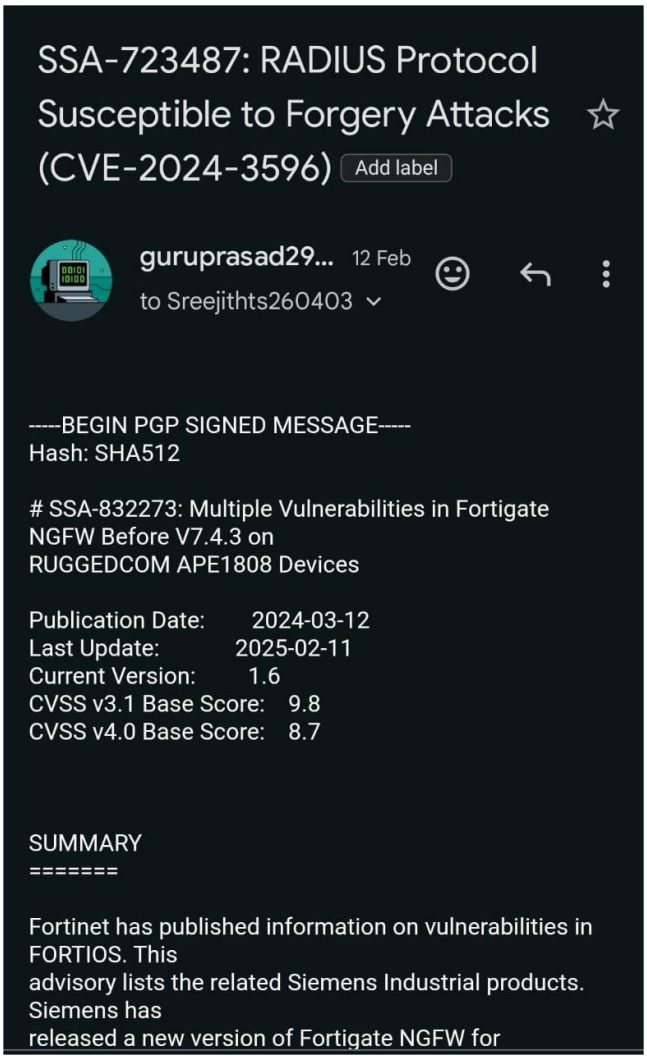

Fig.1.11: Screenshot of an email advisory discussing CVE-2024-3596, a critical forgery vulnerability in the RADIUS protocol affecting FortiGate NGFW.

Fig.1.12: Scheduled Task - Spider Firewall Scheduler

The fig.1.12 shows a Windows Task Scheduler entry named "Spider Firewall Scheduler" on system LAPTOP-08RI2UCQ\ S Guruprasad. It runs Python scripts to scrape vulnerability advisories from sources like Cisco, FortiGuard,

JPCERT, REDHAT, and others. The task activates a virtual environment, collects and processes data, and stores it for analysis and reporting.

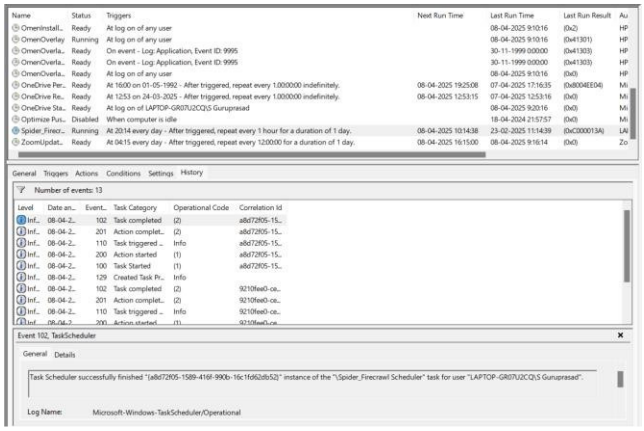

Fig.1.13 Security Considerations for Task Scheduler: Spider Firecrawl – Action Execution Analysis

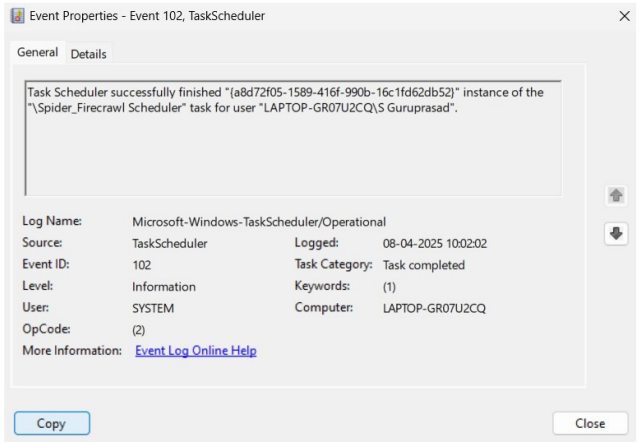

Fig.1.14 Security Considerations for Task Scheduler: Spider Firecrawl - Triggered Execution Analysis

IX. CONCLUSION AND FUTURE ENHANCEMENT

Conclusively, the creation of this web scraping tool represents a new tier in cybersecurity as it can immediately and report on key vulnerabilities in IT and OT equipment. With technologies such as the Fire crawl API, rotating proxies, and anti-bot mechanisms, this tool automates vulnerability detection, improves decision-making, and increases the efficacy of security teams. AI integrates to further fortify threat intelligence, enabling proactive defence against the ever-evolving cyber threats. This tool will play an important role in safeguarding digital assets while being ethically and legally compliant as cybersecurity threats evolve. Integrating AI and machine learning will significantly enhance the web scraping tool for vulnerability monitoring. AI can analyze large datasets for threat patterns, accelerate vulnerability detection, and perform behavioral analysis to identify anomalous activity. Implementing a Zero Trust Architecture will further bolster security by enforcing thorough verification for every access request, regardless of origin, minimizing internal breach risks. These developments will keep the tool automatically discovering and remedying sophisticated attacks, such as zero-day flaws, with help from real-time security analytics, as well as multi-layer defenses.

The enhancement includes further superior threat detection coupled with prevention due to the exploitation of machine learning and behavioral-based analysis. Due to this aspect, integrating natural language processing enables better intelligence gathered from text feeds of phishing from different sources by identifying maliciously related activity. A user feedback mechanism will ensure continuous improvement and adaptation to evolving threats. These cutting-edge technologies will make the tool a much more powerful asset for organizations that are seeking to strengthen their cybersecurity defenses against critical IT and OT vulnerabilities, with agility and in accordance with ethical and legal standards.

X. REFERENCES

[1]. R. R. N. R, N. R. S and V. M., "Web Scrapping Tools and Techniques a Brief Survey," 2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT), Kottayam, India, 2023, pp. 1-4, DOI: 10.1109/ICITIIT57246.2023.10068666.

[2]. A. S. Bale, N. Ghorpade, R. S, S. Kamalesh, R. R and R. B. S, "Web Scraping Approaches and their Performance on Modern Websites [2]," 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2022, pp. 956-959, DOI 10.1109/ICESC54411.2022.9885689.

[3] Ahluwalia, A., and Wani, S. (2024). "**Leveraging Large Language**

Models for Web Scraping". DOI: 10.48550/arXiv.2406.08246.
[4] Xu, Z., Liu, Z., Yan, Y., Liu, Z., Xiong, C., and Yu, G. (2024). "**Cleaner Pretraining Corpus Curation with Neural Web Scraping"**. DOI: 10.48550/arXiv.2402.14652.
[5] Xu, Z., Liu, Z., Yan, Y., Liu, Z., Xiong, C., & Yu, G. (2024). **"Anywhere: A Web Crawler Automation Management Interface":** *Cleaner Pretraining Corpus Curation with Neural Web Scraping*. arXiv preprint arXiv:2402.14652.
[6] Dallabetta, M., Dobberstein, C., Breiding, A., & Akbik, A. (2024). **"Fundus: A Simple-to-Use News Scraper Optimized for High-Quality Extractions":** arXiv preprint arXiv:2403.15279.
[7]. National Vulnerability Database (NVD). (n.d.). About the National Vulnerability Database
[8]. National Institute of Standards and Technology (NIST). (2020). Framework for Improving Critical Infrastructure Cybersecurity. Retrieved from (https://www.nist.gov/cyberframework)
[9]. Common Vulnerability and Exposures (CVE). (n.d.). Common Vulnerability Enumeration. Retrieved from [CVE Website] (https://cve.mitre.org/)
[10]. Verizon. (2023). 2023 Data Breach Investigations Report. Retrieved from [Verizon BIR]

