https://doi.org/10.2352/EI.2024.36.3.MOBMU-326

# Vulnerability Management Using Open-Source Tools

***Navaneeth Shivananjappa, Reiner Creutzburg***

***SRH Berlin University of Applied Sciences, Berlin School of Technology, Ernst-Reuter-Platz 10, D-10587 Berlin, Germany***
***Email: navaneeth.shivananjappa999@gmail.com, reiner.creutzburg@srh.de***

***Technische Hochschule Brandenburg, Department of Computing and Media, Magdeburger Str. 50, D-14770 Brandenburg, Germany***
***Email: creutzburg@th-brandenburg.de***

***Keywords: Vulnerability Management, Cyber Security, Open-Source Tools, Penetration Testing, Vulnerability Scanning, Vulnerability Remediation, Pen Testing.***

## Abstract

*In today's cybersecurity landscape, protecting information systems is crucial due to the rising threat of cyber-attacks. This research focuses on vulnerability management using open-source tools for domain and subdomain enumeration, vulnerability scanning, and remediation. Open-source software offers cost-effective and collaborative security solutions. Domain and subdomain enumeration tools play a vital role in mapping an organization's attack surface, providing insight into security posture. The analysis of vulnerability scanning tools highlights their effectiveness in identifying critical flaws in web applications and databases. Vulnerability remediation through patching, hardening, and exposure management processes closes security gaps. The research provides an empirical insight into using open-source tools for vulnerability management, listing their benefits and limitations empowering organizations to enhance their security posture. Recommendations for integrating these tools into existing security frameworks help combat cyber threats and protect valuable assets.*

## Problem Description

In cases where companies do hire qualified cyber security professionals, the cost of hiring and retaining them can be prohibitive [1], particularly for small businesses that may not have the same level of resources as larger enterprises. This can lead to inadequate protection of information systems, leaving them vulnerable to cyber-attacks. Based on the problem description provided, here are some of the key questions that can be asked:

1. What is the extent of the shortage of cyber security professionals, and how does this affect organizations of different sizes and industries? [2]
2. What are the costs associated with hiring and retaining cyber security professionals [1], and how do these costs vary depending on the size and complexity of the organization's information systems?
3. How can open-source software be used to help organizations address the shortage of cyber security professionals and improve their security posture?
4. What are the benefits and drawbacks of using open-source software for vulnerability management, and how can organizations effectively implement these technologies?

## Objectives

To address the problem statement in the previous section the following objectives for empirical evaluation are derived.

1. To identify a variety open-source tools for asset discovery, to map out the attack surface of target organization by enumerating domains, subdomains and ASN's through passive means, their scope and limitations and a comparative analysis the result of their enumerations.
2. To identify and test a range of open-source vulnerability scanners and conduct a comparative analysis of the scanners by testing them against test website http://testphp.vulnweb.com/, http://php.testsparker.com/ and comparing the types of issues found and how we can use them to identify vulnerabilities in our assets.
3. To discuss vulnerability remediation approach through patching, hardening, and exposure management.

## Methodology

The methodology employed in this study adopts a mixed-methods approach to achieve the research objectives. Initially, a comprehensive literature review is conducted to establish the existing knowledge base on the subject matter. This literature review serves as a foundation for subsequent empirical analysis. Data for the empirical phase is collected from multiple sources, including open-source tools, official documentation, user manuals, community forums, and datasets comprising various web applications and databases with known vulnerabilities. The empirical analysis is divided into three phases: domain and subdomain enumeration, vulnerability scanning, and remediation techniques. Quantitative and qualitative data analysis methods are used to interpret the collected data. Finally, the findings are summarized, implications for future research are discussed, and recommendations are provided based on the results.

## Literature Review

The literature review conducted in this study explores the existing body of knowledge on vulnerability management, focusing on open-source tools and related techniques. Research papers, open-source tools on GitHub, the Bug Hunters Methodology [3][4], the Open Web Application Security Project (OWASP) [5], YouTube videos, and cybersecurity blogs and forums were examined to gather insights into vulnerability management practices. Key themes identified include vulnerability management in modern business environments [6][7], domain and subdomain enumeration techniques [8], vulnerability scanning for web applications and databases [9][10], and remediation techniques such as patching, hardening, and exposure management [11][12]. The literature review provides a comprehensive overview of vulnerability management in cybersecurity, highlighting the significance of addressing security challenges and the potential of open-source tools in enhancing information security posture.

## Tools Considered

The following open-source or free tools were considered for our 3 research objectives.

### *Domain Enumeration*

- bgp.he.net
- Reverse WHOIS - https://www.whoxy.com
- builtwith.com
- Google Dorking
- Censys

### *Subdomain Enumeration*

- Subscraper
- Subfinder
- Amass

### *Scanners*

***Open-Source Website and Application Vulnerability Scanners***

- OWASP Zap
- Sqlmap
- OSV scanner

***Open-Source Infrastructure Vulnerability Scanners***

- Nikto2
- Nuclei
- Nmap

## Findings

### *Comparative Analysis of Domain Enumeration tools*

A comparative analysis based on the data obtained from the empirical evaluation for the tools in scope for domain enumeration for target srh-berlin.de through passive enumeration was done and here are the findings.

***Reverse WhoIs***

**Accuracy:** Less accurate than other sources mentioned in this research. Searches based on keyword on a huge database and hence all domains matching the keyboard might not be accurate.

**No. of discovered domains:** The number of discovered domains is high because of keyword query

**No. of ASN's discovered:** Does not discover ASN’s

**No. Cloud hosted IPs discovered:** The ability to discover cloud IP addresses is weak i.e. it can get relevant domains which might have the Cloud IP but nothing explicit to the cloud

***builtwith.com***

**Accuracy:** Accurate - because it identifies domains based on AD tags which are unique to organizations

**No. of discovered domains:** Many but limited to the domains having the same AD tags

**No. of ASN's discovered:** No ASN's

**No. of Cloud hosted IPs discovered:** Can get cloud hosted domains

***Google Dorking***

**Accuracy:** Accurate - but limited to the google search engine DB

**No. of discovered domains:** Limited to the domains indexed by Google spider

**No. of ASN's discovered:** No ASN's

**No. of Cloud hosted IPs discovered:** Can get cloud hosted domains

***bgp.he.net***

**Accuracy:** Accurate because Hurricane Electric's DB for ASN's is a valid and reliable source

**No. of discovered domains:** High number of IP's and ASN ranges

**No. of ASN's discovered:** Great source for ASN's

**No. of Cloud hosted IPs discovered:** Gets IP's and subnets which are hosted outside cloud environments

***Censys***

**Accuracy:** Very accurate because of continuously scanning the internet and updating the hosts and services discovered. Excellent at discovering live hosts but limited to 250 queries/month in its non-commercial version [13]

**No. of discovered domains:** The live hosts discovered are accurate and valid

**No. of ASN's discovered:** Great source for ASN's

**No. of Cloud hosted IPs discovered:** Very good for getting cloud hosted IPs as the returned infomation logs the name of cloud provider

### *Comparative Analysis of Sub-domain Enumeration tools*

A comparative analysis based on the data obtained from the empirical evaluation for the tools in scope for subdomain enumeration of target srh-berlin.de was done and here are the findings.

***Subscraper***

**Sources:** archiveorg, certsh, dnsbrute, threatcrowd, dnsdumpster, bufferoverrun, searchengine scraping

**Type of Enumeration:** Active +Passive, can be run only in passive mode when there is no authorization

**Application:** exclusively for subdomain enumeration

**Number of discovered subdomains:** 16 Subdomains discovered for srh-berlin.de from passive sources and lists the sources of the discovered sub domains

**Quality of discovered subdomains:** Good and accurate

***Subfinder***

**Sources:** Only passive sources such as BeVigil, BinaryEdge, BufferOver, C99, Censys, CertSpotter, Chaos, Chinaz, DnsDB, Fofa, FullHunt, GitHub, Intelx, PassiveTotal, quake, Robtex, SecurityTrails, Shodan, ThreatBook, VirusTotal, WhoisXML, API, ZoomEye, ZoomEye API, dnsrepo, Hunter

**Type of Enumeration:** Passive

**Application:** exclusively for subdomain enumeration

**Number of discovered subdomains:** 8 subdomains discovered for srh-berlin.de from passive sources and lists the sources of the discovered sub domains

**Quality of discovered subdomains:** Accurate but a smaller number of domains discovered

***Amass***

**Sources:** APIs, Certificates, DNS, Routing, Scraping, Web Archives, WHOIS

**Type of Enumeration:** Active + Passive, can be run only in passive mode when there is no

**Application:** Has additional modules such as

1. amass intel - Discover targets for enumerations
2. amass enum - Perform enumerations and network mapping
3. amass viz - Visualize enumeration results
4. amass track - Track differences between enumerations
5. amass db - Manipulate the Amass graph database

**Number of discovered subdomains:** 14 Subdomains discovered for srh-berlin.de from passive sources, in addition to discovering subdomains, it also provides information on ASN no, IP and Subnet ranges

**Quality of discovered subdomains:** Accurate

***Scope and Limitations of Open-Source Tools for Asset Discovery***

**Scope*:***

- Have diverse sources, including search engines, DNS databases, web archives, and active reconnaissance, ensuring comprehensive scanning of targets.
- Enable users to integrate new sources and techniques, enhancing flexibility and adaptability.
- Produces relevant results.

**Limitations**:

- Limited access to specific commercial databases, potentially impacting the accuracy and completeness of enumeration results.
- Passive enumeration techniques might not capture all subdomains, particularly in organizations with stringent security measures.

## *Comparative Analysis of scanners*

A comparative analysis based on the vulnerabilities discovered from the empirical evaluation for the in-scope open-source scanners was conducted and here are the findings.

***Nmap***: The Nmap scan summary highlights critical vulnerabilities such as CSRF in many pages such as http://testphp.vulnweb.com:80/artist.php, SQL Injection in one among many queries such as http://testphp.vulnweb.com:80/artists.php?artist=2%27%20OR%20sqlspider, and Cross-domain client access issues at clientaccesspolicy.xml: Microsoft Silverlight crossdomain policy, and possible admin folders admin/: that can potentially compromise a website. This underscores the effectiveness of Nmap's vuln scripts in detecting critical flaws, making it a valuable tool for improving website security.

***Nuclei***: While Nuclei didn't discover any critical vulnerabilities on http://testphp.vulnweb.com/, it successfully identified a medium severity vulnerability on php.testsparker.com regarding exposed-svn at http://php.testsparker.com/.svn/entries. To maximize its effectiveness, combining Nuclei with additional scanners like OWASP ZAP and Nikto is recommended for a comprehensive web application penetration test (WAPT). Nuclei's strength lies in detecting missing security headers and exposures, which remain potential entry points for skilled attackers.

***Google OSV Code Scanner***: As a continuously supported and developed project by Google, the OSV scanner demonstrates a strong commitment to ongoing improvements and the integration of new features. Its active development ensures its adaptability to evolving security challenges, solidifying its position as a reliable and innovative tool for code scanning. Scanning code repositories with OSV can aid in white box vulnerability scanning, offering valuable insights into application codebase security before deployment or during later penetration testing [14].

***Nikto***: Nikto stands out as a comprehensive web server scanning tool, combining extensive file and program checks, version verification, server configuration assessments, software detection, and credential guessing capabilities. In the scan report of http://testphp.vulnweb.com/ it was able to detect that the anti-clickjacking X-Frame-Options header was not present, the X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS, /clientaccesspolicy.xml contains a full wildcard entry, Server leaks inodes via ETags, header found with file /clientaccesspolicy.xml, fields: 0x5049b03d 0x133. Its accurate vulnerability detection and flexible output formats make it an asset in ensuring the security of web servers, emphasizing the importance of including it in the scanning arsenal.

***SQLmap***: The scan summary showcases SQLmap's ability to identify and exploit database vulnerabilities, SQLmap was successful in retrieving the DB schema for http://php.testsparker.com/. Although the remaining scanners in this research can detect possible SQL Injection queries SQLmap goes one step further to exploit the query and retrieve data from the database thereby evaluating it to be a true positive. This tool is vital for testing database security against SQL injection attacks, as successful exploits could lead to data breaches with severe financial and legal consequences. Having SQLmap as part of the scanning arsenal becomes paramount to safeguarding database security.

***OWASP ZAP*:** ZAP was successful in discovering high critical vulnerabilities such as path traversal, 2 Remote OS Command Injection and 7 possible SQL injection queries on http://testphp.vulnweb.com/. It also discovered medium criticality vulnerabilities such as .htaccess information leak, absence of Anti-CSRF Tokens and Content Security Policy header not set.

Zap's reporting feature also gives you the respective CWE ID's for detected vulnerabilities and does risk prioritization automatically. ZAP's spidering capabilities comprehensively test web applications for potential vulnerabilities, making it an indispensable tool in the scanning arsenal. It excels at detecting critical and high-risk vulnerabilities, providing an automated solution for risk prioritization and remediation. However, caution should be exercised due to potential false positives, warranting validation of detected vulnerabilities for a more accurate assessment.

***Scope and Limitations of Open-Source Tools for Vulnerability Scanning:***

**Scope**:

- Capability to assess web applications: front and backend, networks, and databases to identify potential vulnerabilities.
- Leveraging a wide range of vulnerability databases, plugins, and scripts, these tools detect known vulnerabilities across different software and configurations.
- Customizable scanning options allow users to tailor scans based on specific needs and compliance requirements.
- Regular updates and active community involvement ensure these tools stay up to date with the latest vulnerabilities.

**Limitations**:

- Access limitations to proprietary vulnerability databases may restrict coverage of vulnerabilities.
- High number of false positives can lead to wasted time or overlooked vulnerabilities.
- Slower scanning.

## *Vulnerability Remediation Approaches*

### *Hardening*

An unauthenticated CouchDB database detected by SQL map in http://php.testsparker.com/ will be a vulnerability of Critical Severity, where an attacker can dump tables. This may include sensitive information such as customer data. By doing this, the attacker can compromise data and cause a breach.

To fix the issue a quick remediation exercise must be carried out. Which in this case would be to restrict and limit access to the database should to people who require the access by setting up authentication and authorization [15][16]. Always, set-up authentication for the user groups who would be interacting with Couch DB within the organization there by limiting accessibility only to required users.

The steps in the official documentation [16] were able to fix the vulnerability of an unauthorized Couch DB through hardening by setting up admin and member user groups with passwords, thereby preventing unauthorized access to the Couch DB. Similar to this, many other types of vulnerabilities require to harden the configuration of the system and software to fix the vulnerabilities and a similar approach of referring to the official documentation of these systems or software can be followed to understand the configuration options available and choose the one that will fix the vulnerability.

### *Patching*

Patching refers to the process of applying updates, fixes, or patches to software, operating systems, or other components of an information system. These updates typically address known vulnerabilities, bugs, or security issues that have been identified by the software vendor or the security community [17].

### Effective guide to enterprise patch management

An effective guide to enterprise patch management should encompass the following key components and considerations [18]:

1. Inventory and Asset Management: Maintain a comprehensive inventory of all hardware and software assets within the enterprise. This includes servers, workstations, networking devices, and applications. Regularly update and maintain accurate records to ensure visibility into the systems that require patching.
2. Vulnerability Assessment: Conduct regular vulnerability assessments to identify and prioritize vulnerabilities within the environment. This involves scanning systems and applications to detect known vulnerabilities, assessing their severity, and determining the necessary patches.
3. Patch Prioritization and Risk Assessment: Develop a risk-based approach to prioritize patches based on their severity, exploitability, and potential impact on the organization's systems and data. Consider factors such as the criticality of the affected systems, the level of exposure to threats, and any available threat intelligence.
4. Patch Testing and Validation: Establish a testing environment to evaluate the impact of patches on system stability, functionality, and compatibility with existing software or configurations. Test patches on representative systems or in a controlled testing environment to identify any potential conflicts or issues before deploying them in production.
5. Patch Deployment Strategy: Define a well-defined and structured process for patch deployment. This should include a schedule or maintenance window for deploying patches, ensuring minimal disruption to business operations. Utilize automation tools or patch management solutions to streamline and expedite the deployment process.
6. Patch Management Tools: Implement patch management tools or solutions that provide centralized control and automation for patch deployment, tracking, and reporting. These tools can help streamline the patch management process, facilitate patch deployment across multiple systems, and provide visibility into patch status and compliance.
7. Change Management and Rollback Procedures: Incorporate proper change management practices to track and document patch deployments. Establish rollback procedures in case a patch causes unexpected issues or conflicts, ensuring the ability to revert changes and restore system functionality if necessary.
8. Ongoing Monitoring and Compliance: Continuously monitor the patch status of systems, validate patch deployment, and ensure ongoing compliance with patch management policies. Implement a monitoring mechanism to identify any unpatched systems or missed patches, and promptly address any gaps.
9. User Awareness and Education: Educate employees and system users about the importance of patching, the role they play in maintaining security, and the potential risks associated with unpatched systems. Encourage users to promptly install updates on their workstations and devices.
10. Regular Review and Improvement: Continuously evaluate and refine the patch management process based on feedback, lessons learned, and emerging threats. Stay up to date with the latest vulnerabilities, patches, and security best practices to adapt the patch management strategy accordingly.

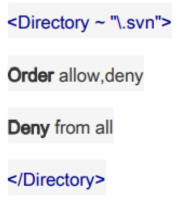
By following these guidelines, organizations can establish a robust and effective enterprise patch management program that helps maintain a secure and resilient IT infrastructure, reduces the risk of vulnerabilities, and enhances overall cybersecurity posture.

### *Exposure Management*

The nuclei scan results on http://testphp.vulnweb.com/ showed an exposed .svn repository which was validated as a true positive and is a high severity vulnerability. Revealing the contents of SVN repository files can potentially divulge critical details, including SVN addresses, associated usernames, and timestamps [19]. Although this type of disclosure might not present immediate opportunities for direct attacks, it can prove highly valuable to malicious actors when leveraged in conjunction with other vulnerabilities or during the exploitation of additional weak points within a system. Consequently, disclosures of this nature could significantly contribute to the success of an attacker's overall strategy, emphasizing the importance of promptly addressing such vulnerabilities to bolster the overall security posture of the system.

### Remediation through Exposure Management

To enhance the security of production systems, it is essential to take proactive measures by either removing these files or imposing access restrictions on the .svn directory. To prevent unauthorized access to all .svn folders, specific lines should be added in the appropriate context. This can be achieved through various means, such as adjusting the global configuration, configuring vhost or directory settings, or employing directives within the .htaccess file. Implementing these preventive measures ensures that sensitive data contained in .svn folders remains safeguarded from potential unauthorized access, thereby fortifying the overall security posture of the system. To remediate the following lines were added in the global config, or vhost/directory, or from .htaccess in the vulnerable webserver.



```
<Directory ~ "\.svn">

Order allow,deny

Deny from all

</Directory>
```

***Figure 1.*** *Lines added in the global config file to remediate .svn repository exposure*

## Peer Interviews

This research scope also considered interviewing Jonathan Kennard who is a Security Engineer at Autobahn Security for qualitative analysis, Jonathan has 3 years of penetration testing experience for different clients, mostly banking apps and boasts the following certifications: OSCP, OSWE, CRTO.

Question 1: How do you go about your enumeration approach in a bug bounty?
Answer: In order:

- Directory brute-force
- Get all endpoints (POST is prioritized)
- Find all subdomains
- Check for code leaks from GitHub or other sources.

Question 2: What are the different scanners you use to scan for potential vulnerabilities during bug bounty?
Answer:

- Dirsearch
- Nmap
- Nikto
- Burp Suite

Question 3: What is your approach to false positive validation?
Answer: Every possible vulnerability found by tools is validated (by manual pentest) to make sure it's not a false positive.

Question 4: What is your advice or approach to reduce and remediate vulnerabilities in your hardware and software systems?
Answer:

- If it's built by hand, fix the problematic code that causes the bug to appear (E.g. SQL Injection)
- Firewall is only as a last resort and is not a reliable means to remediate vulnerabilities.

## Summary and Recommendation

## *Domain enumeration tools*

The comparative analysis of open-source tools for domain enumeration highlights their respective strengths and limitations in accuracy, number of domains discovered, and their ability to identify ASN and cloud-hosted IPs by logging cloud service provider information.

- Reverse WhoIs: This tool is less accurate compared to other sources, as it searches based on keywords on a large database, potentially missing relevant domains. It does not provide ASN information and only indirectly discovers cloud-hosted IPs.
- builtwith.com: Builtwith.com is accurate in identifying domains based on AD TAGS which will be unique to an organization hence accuracy is higher. However, its scope is limited to domains with the same AD tags, and it does not offer ASN discovery. It can identify domains hosted on the cloud, but no information of the cloud provider is logged.
- Google Dorking: While accurate, Google Dorking is restricted to the Google search engine's database and only lists whitelisted domains allowed to be indexed. It does not provide ASN information. Could service provider information is not available.
- bgp.he.net: Highly accurate due to reliable data from Hurricane Electric's ASN database. It can discover many ASN's and is a great source for ASN information. It can identify IPs and subnets hosted outside cloud environments.
- Censys: Very accurate with daily scanning for live hosts, providing valid information about target domains. It is excellent for discovering live hosts and identifying ASN and cloud-hosted IPs and cloud service provider information, making it a valuable tool in domain enumeration. However, there is a query limitation of 250 queries/month.
- Peer Suggestions: Jonathan suggests additionally querying GitHub for code leaks and missing endpoints if not already discovered using the tools in the scope of this research.

In summary, each tool offers advantages and limitations in domain enumeration. Builtwith.com and Google Dorking are useful for specific domains based on AD TAGS and Google's indexed whitelist, respectively ensuring success in finding relevant domains while not logging cloud service provider information. Reverse WhoIs can get many relevant domains because it searches based on keyword but can be less accurate due to keyword-based searches can also fetch unrelated domains which might have the same keyword. ASN with bgp.he.net excels in providing accurate ASN information but not cloud hosted IPs. However, Censys stands out for its very accurate and extensive scanning capabilities, making it a powerful tool for discovering live hosts and obtaining ASN.

In recommendation, organizations should consider logging the results of the domains enumerated by all the tools (Reverse WhoIs, builtwith.com, Google Dorking, bgp.he.net, Censys, in order to ensure coverage and then pass the discovered domains through a validation algorithm which will make a request to the domain and matches key identifiable information in the returned DOM content against a keyword set to ensure validity of the discovered domain. Key identifiable information can be organization name, organizations copyright text, terms of service text, privacy policy text.

### Subdomain Enumeration tools

The comparative analysis of subdomain enumeration tools highlights their respective strengths, limitations and features relevant to each of these tools.

- Subscraper: Subscraper is a versatile tool that supports both active and passive subdomain enumeration. In passive mode. It utilizes multiple sources, including archiveorg, certsh, dnsbrute, threatcrowd, dnsdumpster, bufferoverrun, and search engine scraping. From passive sources alone, Subscraper discovered 16 subdomains for srh-berlin.de, demonstrating good accuracy in its results which was validated by accessing the discovered subdomain.

- Subfinder: Subfinder specializes in passive subdomain enumeration and relies on a wide range of passive sources, including BeVigil, BinaryEdge, BufferOver, C99, Censys, and others. From these sources, it discovered 8 subdomains for srh-berlin.de. Although the number of subdomains discovered is fewer than Subscraper, the tool still provided accurate results which was validated by accessing the discovered subdomain.

- Amass: Amass is a comprehensive subdomain enumeration tool that supports both active and passive modes, making it adaptable to various scenarios. It offers additional modules for target discovery, network mapping, visualization of results, and database manipulation. From passive sources, Amass discovered 14 subdomains for srh-berlin.de, which were found to be accurate.
- Peer Suggestions: Jonathan suggests using Subscraper and Amass because of their Directory brute forcing capabilities for subdomain enumerations.

In summary, all three subdomain enumeration tools—Subscraper, Subfinder, and Amass—offer valuable capabilities for discovering subdomains. Subscraper excels in versatility, supporting both active and passive enumeration and providing the highest number of subdomains. Subfinder is a reliable passive tool with accurate results, while Amass stands out as a powerful and feature-rich option that not only discovers subdomains but also provides additional network mapping and data manipulation functionalities.

In recommendation, organizations should consider logging the results of the domains enumerated by all the tools (Subscraper, Subfinder, Amass) in order to ensure coverage then create a uniqe lsit to remove duplicates and then pass the discovered domains through a validation algorithm which will make a request to the subdomain and matches key identifiable information in the returned DOM content against a keyword set to ensure validity of the discovered subdomain. Key identifiable information can be organization name, organizations copyright text, terms of service text, privacy policy text and relavant keywords unique to the organization.

### Scanners

The comparative analysis of the scanning tools reveals the strengths and areas of expertise for each tool in the vulnerability scanning process. Nmap emerges as a powerful tool for detecting critical vulnerabilities like CSRF, SQL Injection, and Cross-domain client access issues, making it highly effective for improving website security. Nuclei, while not finding critical vulnerabilities, showcases its prowess in identifying medium severity issues and exposing potential entry points for skilled attackers through missing security headers and exposures. The Google OSV Code Scanner's active development and continuous support by Google make it a reliable and innovative choice for code scanning, its focus on white box vulnerability scanning and insights into codebase security before deployment or penetration testing adds significant value. Nikto, stands out as a comprehensive web server scanning tool, offering extensive checks and assessments for server security, highlighting its importance in securing web servers. SQLmap's ability to identify and exploit database vulnerabilities, including retrieving DB schemas, underlines its critical role in safeguarding database security against SQL injection attacks. Lastly, OWASP ZAP impresses with its comprehensive spidering capabilities, providing an invaluable tool for scanning web applications for potential vulnerabilities and offering automated risk prioritization and remediation. In the peer interview Jonathan goes on to recommend Nmap and Nikto which are part of the research scope. And in addition, he also recommended two more scanners, Dirsearch and Burp Suite which can increase the chances of finding out more vulnerabilities which may not be detected using the above-mentioned tools. In the case of False Positive validation Jonathan recommends that every possible finding found by tools should be validated (by manual pentest) to make sure it’s not a false positive. And in the case of Open-Source Scanners false positive rates are observed to be high and hence False Positive validation is crucial.

In recommendation, each scanner offers unique features and strengths that cater to specific aspects of web application security. Organizations should consider their specific security requirements and objectives when selecting scanning tools. Integrating a combination of these scanners, depending on their respective expertise, will lead to a more robust and thorough web application security assessment which should be followed by false positive validation for all the identified vulnerabilities. This can help organizations prevent common vulnerabilities when there is a constraint to use licensed scanners and pen testers.

### Vulnerability Remediation

Vulnerability remediation through patching helps organizations stay ahead of potential threats by keeping their systems up to date with the latest security fixes. It is essential to promptly apply patches released by software vendors to prevent exploitation of known vulnerabilities. Automated patch management tools can streamline this process and ensure timely updates across the infrastructure.

Hardening helps securing systems by configuring them according to industry standards and security guidelines. This includes disabling unnecessary services, implementing strong authentication mechanisms, and restricting access to sensitive resources. By hardening their systems, organizations can reduce the likelihood of successful attacks and limit the impact of any potential breaches.

Exposure management identifies and mitigates potential risks posed by sensitive information exposure such as Web server version, type web framework used and its version, exposed internal services.

And as per recommendations in the peer interview with Jonathan the best way to stay secure is by updating application if applicable through regular patching. This will help keep safe against zero-day vulnerabilities as the companies keep releasing regular patches to safeguard against zero-day vulnerabilities. And Jonathan goes on to recommend that best security practices should be considered during the development process to be protected against attacks such as SQL injection. And lastly, having a firewall is a last resort to safeguard against vulnerabilities and should not be the only remediation approach.

In recommendation, vulnerability remediation through patching, hardening, and exposure management is a multifaceted approach by following which organizations can improve their security posture. Organizations must consider promptly applying software patches, implementing hardened security configurations and continuous monitoring to mitigate potential risks. A comprehensive and proactive vulnerability remediation strategy is essential to safeguard critical assets, protect sensitive data, and maintain a strong defense against evolving cybersecurity threats.

### *Future Scope*

The future scope of the research could involve the following areas of exploration and investigation:

1. In-Depth Analysis of Open-Source Tools: Conducting a more detailed analysis of the identified open-source tools for asset discovery, endpoint mapping, and vulnerability scanning. This could include examining their source code, exploring their capabilities in different scenarios, and assessing their performance under various conditions.
2. Integration and Automation: Exploring ways to integrate the identified tools into a comprehensive vulnerability management framework. The research could focus on creating a unified and automated approach for asset discovery, vulnerability scanning, and remediation, thereby streamlining the entire vulnerability management process.
3. Assessing Tool Accuracy: Conducting further evaluations of the identified vulnerability scanners to measure their accuracy in identifying and classifying vulnerabilities. This would involve comparing their results against known vulnerabilities and assessing false positive rates to improve their reliability.
4. Vulnerability Remediation Strategies: Researching and proposing effective vulnerability remediation strategies beyond patching and hardening. This may include implementing proactive security measures, continuous monitoring, and feedback loops to ensure a more robust and resilient security posture.
5. Real-World Application: Validating the effectiveness of the research findings and proposed methodologies by applying them to real-world scenarios and organizations. This would involve conducting case studies and practical implementations to measure the impact of using open-source tools in a real-world context.
6. Emerging Technologies and Threats: Considering the impact of emerging technologies, such as Internet of Things (IoT), cloud computing, and artificial intelligence, on vulnerability management. The research could also explore new threats and attack vectors that may arise due to evolving technologies.
7. Comparative Analysis of Commercial Tools: Expanding the comparative analysis to include commercial vulnerability management tools. This would provide a broader perspective on the advantages and limitations of open-source solutions in comparison to commercial offerings.

By addressing these future scope areas, the research can contribute to advancing the field of vulnerability management, providing valuable insights, and guiding organizations in making informed decisions to enhance their cybersecurity practices.

## References

[1] Libicki, Martin C., David Senty, and Julia Pollak. Hackers wanted: An examination of the cybersecurity labor market. Rand Corporation, 2014.

[2] Pierce, Adam O. "Exploring the cybersecurity hiring gap." PhD diss., Walden University, 2016

[3] TBHM Git Hub Page, Author: jhaddix, https://github.com/jhaddix/tbhm

[4] TBHM Hacktivitycon2020 Author: https://www.hacker101.com/conferences/hacktivitycon2020/tbhm

[5] Marchand-Melsom, Alexander, and Duong Bao Nguyen Mai. "Automatic repair of OWASP Top 10 security vulnerabilities: A survey." In Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, pp. 23-30. 2020

[6] Nyanchama, Matunda. "Enterprise Vulnerability Management and Its Role in Information Security Management." Inf. Secur. J. A Glob. Perspect. 14, no. 3 (2005): 29-56.

[7] Sotiropoulos, Panagiotis, Christos-Minas Mathas, Costas Vassilakis, and Nicholas Kolokotronis. "A Software Vulnerability Management Framework for the Minimization of System Attack Surface and Risk." Electronics 12, no. 10 (2023): 2278

[8] Kathrine, G. Jaspher, Ronnie T. Baby, and V. Ebenzer. "COMPARATIVE ANALYSIS OF SUBDOMAIN ENUMERATION TOOLS AND STATIC CODE ANALYSIS." ISSN (Online): 2454-7190.

[9] Fonseca, Jose, Marco Vieira, and Henrique Madeira. "Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks." In 13th Pacific Rim international symposium on dependable computing (PRDC 2007), pp. 365-372. IEEE, 2007.

[10] Daud, Nor Izyani, Khairul Azmi Abu Bakar, and Mohd Shafeq Md Hasan. "A case study on web application vulnerability scanning tools." In 2014 Science and Information Conference, pp. 595-600. IEEE, 2014.

[11] Brykczynski, Bill, and Robert A. Small. "Reducing internet-based intrusions: Effective security patch management." IEEE software 20, no. 1 (2003): 50-57.

[12] Dissanayake, Nesara, Asangi Jayatilaka, Mansooreh Zahedi, and Muhammad Ali Babar. "An Empirical Study of Automation in Software Security Patch Management." In 37th IEEE/ACM International Conference on Automated Software Engineering, pp. 1-13. 2022

[13] Censys Search Engine Intro, Author: Chad Warner, https://warnerchad.medium.com/censys-search-engine-intro-d502d9839c1c

[14] Google Security Blog, Author: Rex Pan, software engineer, Google Open-Source Security Team,

https://security.googleblog.com/2022/12/announcing-osv-scanner-vulnerability.html

[15] Couch DB Documentation Page, Author: CouchDB, https://docs.couchdb.org/en/

[16] Couch DB Documentation Security Page, Author: CouchDB, https://docs.couchdb.org/en/stable/intro/security.html

[17] Mell, Peter, Tiffany Bergeron, and David Henning. "Creating a patch and vulnerability management program." NIST Special Publication 800 (2005): 40

[18] Souppaya, Murugiah, and Karen Scarfone. "Guide to enterprise patch management technologies." NIST Special Publication 800 (2013): 40

[19] SVN Detected page, Author: Acunetix, https://www.acunetix.com/vulnerabilities/web/svn-detected/

## Biography

Navaneeth Shivananjappa received his M.Sc.~in Computer Science focus Cyber Security from SRH Berlin University of Applied Sciences, Berlin School of Technology in 2023, he received his Bachelors in Mechanical Enginnering in 2011 from Visvesvaraya Technological University (VTU), Karnataka, India. Since 2023 he is working at the SRH Berlin University of Applied Sciences as lecturer teaching Penetration Testing and Cyber Security.
Until 2022 he worked in IT industry for 12 years in Cyber Security and Non-functional testing. His research work is focused on Cybersecurity Vulnerability Management, Vulnerability Remediation, Secure Software Development Practices and Performance Testing and Engineering.

Reiner Creutzburg is a Retired Professor for Applied Computer Science at the Technische Hochschule Brandenburg in Brandenburg, Germany. Since 2019, he has been a Professor of IT Security at the SRH Berlin University of Applied Sciences, Berlin School of Technology.
He has been a member of the IEEE and SPIE and chairman of the Multimedia on Mobile Devices (MOBMU) Conference at the Electronic Imaging conferences since 2005.
In 2019, he was elected a member of the Leibniz Society of Sciences to Berlin e.V. His research interest is focused on Cybersecurity, Digital Forensics, Open Source Intelligence (OSINT), Multimedia Signal Processing, eLearning, Parallel Memory Architectures, and Modern Digital Media and Imaging Applications.

## Acknowledgements

This work was supported partially by the European Union in the framework of ERASMUS MUNDUS, Project CyberMACS (Project # 101082683) (https://cybermacs.eu)