SECURITY ASSESSMENT FOR INDIAN WEBSITES

SIVA KUMAR MUTHU

UNIVERSITI KEBANGSAAN MALAYSIA

SECURITY ASSESSMENT FOR INDIAN WEBSITES

SIVA KUMAR MUTHU

PROJECT SUBMITTED IN PARTIAL FULFILMENT FOR THE DEGREE OF
MASTER OF CYBER SECURITY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2023

PENILAIAN KESELAMATAN DI LAMAN WEB INDIA

SIVA KUMAR MUPTHU

PROJEK YANG DIKEMUKAKAN UNTUK MEMENUHI SEBAHAGIAN
DARIPADA SYARAT MEMPEROLEH IJAZAH
SARJANA KESELAMATAN SIBER

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2023

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

06 September 2023

SIVA KUMAR MUTHU
P111940

ACKNOWLEDGEMENT

I want to begin by expressing my profound gratitude to the divine presence of Shivan almighty Om Namah Shivaya for providing guidance and inspiration throughout my academic journey.

I am deeply indebted to my supervisor, Dr Azana Hafizah Mohd Aman, for their exceptional guidance, unwavering support, and valuable insights that have significantly contributed to the success of my research. I am also immensely grateful to the UKM Power Research group's postgraduate students for their assistance and for creating a harmonious and motivating environment during my years at UKM. Their collective efforts have played a pivotal role in shaping my research experience and personal growth.

Additionally, I would like to express my heartfelt appreciation to my beloved wife, Saroja, whose unconditional love, understanding, and encouragement have been the bedrock of my perseverance and achievements throughout my master's program in Cybersecurity. Her unwavering support and sacrifices have been a constant source of strength and motivation for me. I am truly blessed to have such a remarkable partner by my side.

To all those mentioned above and those who have contributed in ways seen and unseen, I offer my deepest gratitude for their invaluable support, which has made this academic endeavour possible.

ABSTRAK

Laman web untuk masyarakat India terdedah kepada serangan siber yang serius. Oleh itu, langkah berjaga-jaga mesti diambil. Matlamat utama kajian ini adalah untuk menilai keselamatan laman web untuk masyarakat India bagi mengenal pasti sebarang kelemahan, memberikan cadangan mengenai pengurangan, dan memastikan keselamatan dan keselamatan komuniti yang menggunakan laman web ini. Teknologi sumber terbuka seperti Sublister, SQLmap, Nmap, Online OpenVAS, dan Qualys akan digunakan dalam penilaian untuk mengkaji laman web dan mencari sebarang kelemahan yang berpotensi secara komprehensif. Penjelasan berkenaan keselamatan laman web dan cadangan untuk meningkatkan keselamatannya dan mengurangkan risiko. Penemuan kajian akan membantu pentadbir laman web, masyarakat India, dan pakar keselamatan siber yang menemui kelemahan. Operasi dalam talian sangat bergantung pada keselamatan laman web, terutamanya ketika mengendalikan data pengguna sensitif. Penyelidikan ini menggunakan alat pengimbasan keselamatan sumber terbuka untuk mengesan potensi kelemahan keselamatan. Statistik deskriptif digunakan untuk menganalisis data dari imbasan untuk menentukan kelemahan dan ancaman keselamatan yang paling lazim dalam kumpulan laman web India yang terhad. Penemuan penilaian ini menawarkan maklumat mengenai tahap keselamatan laman web semasa dalam masyarakat India dan menunjukkan bidang yang perlu diperkukuhkan untuk meningkatkan keselamatan keseluruhan laman web ini. Ia dapat membantu perniagaan melalui pengendalian sumber yang lebih baik, mengurangkan risiko serangan siber dan memberi keutamaan aktiviti keselamatan siber. Hasilnya menunjukkan bahawa laman web yang dipilih untuk masyarakat India berada pada tahap yang membimbangkan, di mana 172 kelemahan ditemui, dan kelemahannya berkaitan dengan suntikan SQL, CSRF, kerentanan XSS, dan terdapat satu pertukaran kunci yang lemah. Pelan mitigasi telah digariskan untuk setiap kelemahan.

ABSTRACT

Indian websites are seriously in danger from cyberattacks. Thus, precautions must be taken. This study's primary goal is to thoroughly assess the website's security for the Indian websites to identify any vulnerabilities, provide recommendations on mitigation, and ensure the safety and security of users of the site. Open-source technologies like Sublister, SQLmap, Nmap, Online OpenVAS, and Qualys will be used in the evaluation to study the website and find any potential vulnerabilities comprehensively. By illuminating the website's security posture and offering suggestions to improve its security and lessen risks, the study's findings will be helpful to website administrators, Indian communities, and cybersecurity experts who discovered weaknesses. Online operations depend heavily on website security, especially when handling sensitive user data. This research used open-source security scanning tools to detect potential security vulnerabilities. Descriptive statistics were used to analyse the data from the scans to determine the most prevalent vulnerabilities and security threats in a limited group of Indian websites. The assessment's findings offer insightful information on the current level of website security in Indian communities and point out areas that should be strengthened to improve the overall security posture of these websites. It can assist businesses in better resource allocation, reducing the risk of cyberattacks and prioritizing their cybersecurity activities. The result shows that the selected website for Indian communities is at an alarming stage, whereby 172 vulnerabilities were found, and this vulnerability is related to SQL injection, CSRF, XSS Vulnerabilities, and one weak key exchange. Mitigation plans were outlined for each vulnerability.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
B2B	Business to Business
CIA	Confidentiality, Integrity, and Availability
CORS	Cross-Origin Resource Sharing
CORS	Cross-Origin Resource Sharing
CRLF	Carriage Return Line Feed
CSP	Content Security Policy
CSRF	Cross-Site Request Forgery
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DDoS	Distributed denial-of-service
DoS	Denial-of-Service
EDR	Endpoint Detection and Response
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Communications Technology
MitM	Man-in-the-middle
NSE	Nmap Scripting Engine
NVD	National Vulnerability Database
RC4	Rivest Cipher 4
SQL	Structured query language
SSL	Secure Sockets Layer
SSRF	Server-Side Request Forgery
STS	Strict-Transport-Security
TLS	Transport Layer Security
UDS	User Data Services
URL	Uniform Resource Locators

XSLT	Extensible Stylesheet Language Transformations
XSS	Cross-site scripting

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

In ancient times, battles were waged on physical battlegrounds with armored warriors wielding weapons. In the digital era, wars are fought in the cyber world, which we physically cannot see or touch. In the cyber world, attackers and hackers launch cyber-attacks without physically being present at the crime site. These attacks can cause devastating damage to countries, organizations, and individuals; therefore, cybersecurity has become a critical factor in the cyber world. To counter cybersecurity threats, the vendor is inventing new technology while businesses invest extensively in security. However, the number of events is not declining despite these measures. Many attacks are related to exploiting vulnerabilities in products or configurations (Kumi et al. 2021).

It is also worth noting that cyberattacks are becoming more sophisticated, and attackers are constantly evolving their tactics to overcome the latest security measures. Therefore, organizations must stay current with the latest cybersecurity trends and implement the most advanced security solutions to protect against emerging threats which includes working with vendors who provide:

1. Endpoint solutions such as EDR or network protection such as firewalls.
2. Regularly testing their systems for vulnerabilities.
3. Taking immediate action to address any identified issues.

Every organization is investing heavily and focusing on security by enhancing its current security posture to prevent cyberattack against their organization and preventing devastating damage to their CIA, and product vendor such as EDR vendor or firewall vendor is developing new technologies to prevent or stop cybersecurity attacks. However, the number of incidents is not declining.

Most of the attacks are related to exploiting product vulnerability or configuration vulnerabilities. For 2022, a total of 9009 vulnerabilities were detected, and for 2021, there were 20150 vulnerabilities detected (Serkan Özkan 2022). A total of 25539 website vulnerabilities were detected, some rated critical and could directly impact the organization CIA (Serkan Özkan 2022). The number of web servers globally is enormous, as every organization runs at least one web server for their business needs, such as an E-commerce site. Figure 1.1 shows the most common web server (Q-Success 2023).

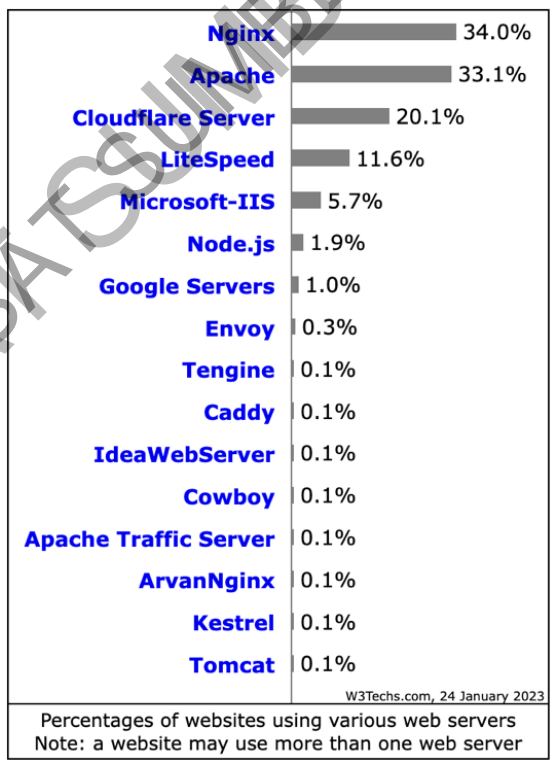


Figure 1.1 Percentages of websites using various web servers

Source: Q-Success 2023

In conclusion, moving to the digital era has led to general challenges for companies and individuals, especially regarding cybersecurity. Security must be given top priority as cyberattacks get more sophisticated and frequent. A constantly changing threat landscape requires organizations to stay cautious and invest in advanced security solutions.

There are still many vulnerabilities being exploited today. Every website design must adhere to strict guidelines and best practices to prevent cyber-attacks. The most common cyber-attacks on websites (Ahmed & Al Dabbagh 2023):

1. Cross-Site Scripting (XSS)
2. Injection Attacks
3. Fuzzing (or Fuzz Testing)
4. Zero-Day Attack
5. Path (or Directory) Traversal
6. Distributed Denial-of-Service (DDoS)
7. Man-In-The-Middle Attack
8. Brute Force Attack

Penetration testing can identify potential security vulnerabilities before attackers exploit them. Penetration testing can help organizations proactively secure their web servers, website and prevent costly security breaches, data leaks, and other security incidents. In addition to penetration testing, organizations can implement other security measures to secure their web servers. Firewalls, intrusion detection systems, and certain coding practices are some of the ways they can monitor and block malicious traffic. Regular security assessments and audits can help organizations stay on top of emerging threats and ensure their security measures remain effective. (Altulaihan et al. 2023).

Finally, securing web servers is a critical task that requires ongoing effort and attention. Organizations can protect their digital assets and maintain their data and

systems' confidentiality, integrity, and availability by implementing a comprehensive security strategy that includes penetration testing, secure coding practices, and other security measures.

1.2 RESEARCH BACKGROUND

We have seen an increase in cyberattacks in India; therefore, cybersecurity has become a hot topic in securing websites, especially those websites that deal with sensitive data, such as B2B, e-commerce, banking, and government websites. Using tools like open-source vulnerability scanners, a website's security posture is tested, analysed, and reported on in a website security assessment. Website security is frequently evaluated with vulnerability scanners like nmap, SQLmap, and online OpenVAS. Despite these initiatives, more cyber-related offenses have been registered in India than in all of 2018 (about 212,485 offenses have been reported in only the first two months of 2022 alone). India recorded 394,499 cybercrimes in 2019 before the COVID-19 outbreak, a sharp rise (Najar & Naik S 2022).

In 2021, many Indian organizations faced ransomware attacks, with 80% of these attacks resulting in data loss. Additionally, website hacking incidents have increased recently, with 26,121 sites being hacked in 2020 alone (Singal & Chhillar 2017). Therefore, the high likelihood of Indian websites being attacked underscores the need for continuous security assessments and improvements to protect against cyber threats.

1.3 PROBLEM STATEMENT

Website security safeguards websites from unauthorized access, use, modification, destruction, or disruption. Security assessment enables organizations to proactively identify potential threats to their software and applications. It is crucial for several reasons: identifying and addressing vulnerabilities and weaknesses in applications and development processes, complying with cybersecurity laws and regulations, evaluating, and enhancing the overall security posture, and preventing security defects and vulnerabilities. Web security testing, particularly at the application layer, focuses on discovering security vulnerabilities and stimulates unexpected behaviours through

diverse input types. The Indian websites is a crucial platform for the Indian population, providing various services such as e-commerce, education, and healthcare. Unfortunately, recent cyberattacks have targeted the website, compromised sensitive data, and exposed vulnerabilities that put the Indian websites at risk. The website's current security posture is unknown, and there is a lack of research on Indian websites' security vulnerabilities. Therefore, there is a pressing need to conduct a security assessment of the Indian websites website to identify and mitigate any existing vulnerabilities and ensure the safety and security of Indian websites members who use the website.

1.4 RESEARCH QUESTION

1. What comprehensive scanning techniques can be used to identify vulnerabilities on an Indian website?
2. What specific vulnerabilities exist on Indian websites, and how can they be identified?
3. What are the best practices for addressing the vulnerabilities identified on an Indian website?
4. What are the appropriate remedial actions that can be taken to address the identified vulnerabilities on an Indian website?

1.5 OBJECTIVES OF RESEARCH

This study aims to conduct a thorough security assessment of the website catering to the Indian websites. The research objectives are:

1. To identify any existing vulnerabilities on the selected Indian website.
2. To analyse the result and suggest appropriate actions for the vulnerabilities.

1.6 RESEARCH SCOPE

The research scope will focus on performing a security assessment on selected Indian websites. A security assessment will be conducted using open-source tools Sublister, SQLmap, Nmap, Online OpenVAS, and Qualys.

Table 1.1 Selected Indian websites

No	Web Site	Sector	Nature of the business	Impact analysis for web site down
1	www.sprink.online	Food	Food Delivery	Reputation and revenue impact. High chance the customer will discontinue their service.
2	www.sifytechnologies.com	IT Service	IT Integration	Reputation and revenue impact. There is a high chance that customer data is stolen, and the patient moves to another healthcare provider.
3	www.sunriseuniversity.in	Education	Higher Education	Reputation and revenue impact. There is a high chance that customer data will be stolen, and the customer will move to another online boutique.
4	www.amity.edu	Education	Higher Education	Reputation and revenue impact. There is a high chance that customer data is stolen, and the customer moves to another online grocery store.
5	www.medtravels.in	Medical	Healthcare	Reputation and revenue impact. There is a high chance that customer data is stolen, and the customer moves to another online grocery store.
6	www.i2ifunding.com	Finance	Loan Provider	Reputation and revenue impact. There is a high chance that customer data is stolen, and the customer moves to another loan provider.
7	www.busindia.com	Ticketing	Online bus ticket reservation	Reputation and revenue impact. There is a high chance that customer data is stolen, and the customer will move to another online ticket.
8	www.manipalhospitalsglobal.com	Health	Healthcare service provider	Reputation and revenue impact. There is a high chance that customer data is stolen, and the customer will move to healthcare.

1.7 THESIS ORGANIZATION

The thesis consists of five chapters which start with an introduction chapter. First chapter emphasizes the necessity for proactive efforts to minimize cyberattacks and focuses on cybersecurity problems in the digital world. Cyber incidents are increasing despite significant investments and technical breakthroughs. The chapter emphasizes how vulnerable web servers are and how crucial it is to secure them. To safeguard digital assets, it offers recommendations, including penetration testing, secure coding techniques, data encryption, and the usage of firewalls and intrusion detection systems. The chapter also discusses the rise in cybercrimes in India and highlights the value of website security evaluations.

Chapter two presents a literature review and discusses the evolution of cyber threats. It evaluates many research papers to comprehend the present status of cybersecurity, including trends, issues, and solutions. It emphasizes the adverse effects of cyberattacks on people, businesses, and national security, underscoring the necessity of effective cybersecurity measures. Insecure communication protocols, cross-site scripting, and SQL injection were identified with scanning tools.

The data collected from the scans were analysed using descriptive statistics to identify the most common vulnerabilities and security risks. The vulnerabilities were prioritized based on the risk level, and recommendations were provided for remediation. The research design and approach for this research using open-source tools provided a reliable and efficient means of identifying potential website security risks in the Indian websites. A diverse sample of websites was selected to provide a comprehensive understanding of website security practices across different sectors within the Indian websites.

In Chapter three, the focus is on the method for assessing the security of Indian websites websites using open-source security scanning technology is discussed. The research will employ automated scanning techniques to identify prevalent vulnerabilities and threats. Eight websites from various sectors, including food delivery, IT services, education, healthcare, finance, online travel ticketing, and healthcare

services, are selected as examples. Open-source tools like Nmap, online OpenVAS, SQLmap, Qualys, and Sublister check for vulnerabilities like SQL injection and cross-site scripting. It is crucial to remember that a thorough examination must involve manual testing. Overall, this technique provides a practical and affordable way to evaluate websites.

Chapter four focuses on assessing the security of Indian websites using open-source technologies. These results provide further information about the security risks and vulnerabilities found during scanning. Each website received a vulnerability rating and focused remedial efforts. The severity of the vulnerabilities and their possible impact on the security of the websites were used to determine the scores.

Based on their severity and possible effects, vulnerabilities were rated as high, medium, or low using a scoring system. Medium-rated vulnerabilities signify significant risks that should be handled quickly, while high-rated vulnerabilities show serious hazards that demand immediate attention. Low-rated vulnerabilities still need to be mitigated to stop potential exploitation.

Chapter five emphasizes the significant contribution made by this research in assessing the website security of the Indian websites while acknowledging the limitations and presenting future directions for further exploration.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

As the Indian websites multiplies online, cybersecurity threats and vulnerabilities are rising in India, making it imperative for businesses, government institutions, and individuals to take proactive measures to safeguard against cyberattacks.

The review concentrates on the most recent academic papers, studies, and research in India's website security evaluation and cybersecurity field. The review also highlights the necessity of implementing strong cybersecurity measures, such as routine security audits and vulnerability assessments, investing in security solutions, providing staff with cybersecurity best practices, and identifying challenges and gaps in the existing literature.

Overall, this literature analysis provides the Indian websites with helpful information on the state of website security assessment and cybersecurity risks and vulnerabilities. The evaluation assists organizations, governmental bodies, and private citizens in better comprehending the dangers posed by cyber-attacks and developing preventative strategies to lessen those dangers. Overall, this review provides valuable insights into the current website security assessment and cybersecurity threats and vulnerabilities for the Indian websites thus helping businesses, government institutions, and individuals better understand the risks associated with cyber threats and take proactive measures to mitigate these risks.

2.1 WEBSITE THREATS AND VULNERABILITIES

Websites confront numerous threats and vulnerabilities that, if exploited by hackers or attackers, can have significant consequences. Understanding these prospective threats

is essential for sustaining strong security measures. Among the most prevalent hazards and vulnerabilities are Cross-site scripting (XSS) attacks, which pose a significant threat to web applications (Dora & Nemoga 2021; Nagarjun & Ahamad 2020). As interconnected devices that utilize various web applications for distinct purposes increase, the susceptibility to XSS attacks increases dramatically.

1. Cross-site scripting (XSS) attacks have persisted since the 1990s. Over the years, XSS has continued to be identified as a significant threat to web applications, maintaining its relevance in the field of cybersecurity even in the present day. Malicious actors inject malicious JavaScript code into vulnerable web applications during XSS attacks.

   Consequently, unauthorized actions initiate when an unsuspecting user executes this injected code within their browser, transmitting sensitive information to the perpetrator or redirecting the user to a malicious website. An XSS attack aims to exploit the web application's vulnerability and maliciously manipulate user interactions.

2. DDoS attacks are the second danger to websites (Naveen Sharma 2020). Malicious attacks, primarily Distributed Denial of Service (DDoS) attacks, have increased significantly in frequency in recent years. These assaults can produce traffic levels reaching Gbps (gigabits per second), with attack capabilities of up to 1000Gbps have also emerged. Additionally, on-demand DDoS attacks between firms have grown in popularity.

   The fact that DDoS attacks are now affecting household electronic equipment, intelligent devices, IoT-connected machines, and more makes the issue more worrying than it already is. These attacks might be volumetric, which floods the target's bandwidth with malicious traffic, or semantic, preventing access to even the smallest bandwidth.

   The ever-changing nature of DDoS attacks underscores the critical need for all connected devices to implement robust cybersecurity measures. DDoS attacks can be carried out in various ways, with assailants frequently employing a technique known as "streaming" malicious traffic toward the target. This traffic stream targets critical resources such as IP addresses, routers, servers, and

machines, rendering them inaccessible to the victim's legitimate consumers and clients.

Compromising devices within a network is yet another plausible strategy. Attackers frequently launch DDoS attacks by exploiting device vulnerabilities. Once the attacker's system has gained control, it becomes the DDoS orchestrator, identifying other vulnerable systems and launching a massive attack against them. These attacks aim to overwhelm servers with malicious traffic, exhausting their bandwidth, RAM, and CPU. In modern scenarios, cybercriminals employ powerful DDoS attacks as phantom or disguised attacks, using them as a diversion while launching crucial data breaches, security breaches, or ransomware attacks. This attack demonstrates the increasing sophistication and destructive potential of DDoS attacks in the hands of criminals.

3. SQL Injection (Ahmed et al. 2021). Data-centric attacks pose a significant and grave security risk. SQL and NoSQL injection is among the most significant cyberattacks against popular database-driven web applications. These attacks involve injecting malicious code into a database, which could expose sensitive information to hackers. SQL injection poses a dire threat to databases, mainly when user input validation is insufficient, as it can result in the disclosure of confidential information without authorization. These hazards are susceptible to both relational and non-relational databases.

   NoSQL databases offer performance, storage, and data retrieval advantages but are not immune to injection assaults. Even though NoSQL injection differs from traditional SQL injection due to using a distinct query language, both attacks depend on exploiting suspicious input execution on the server.

   Consequently, injection assaults are also a significant concern for non-relational databases. A comprehensive examination of various injection assaults was conducted, including SQL and NoSQL injection detection techniques and countermeasures. Web application developers can mitigate injection attacks by understanding and implementing appropriate security measures.

   Inclusion attacks and directory traversal are the most dangerous. Recent web applications are vulnerable to directory traversal vulnerabilities, which enable

intruders to access sensitive files. Vulnerabilities and misconfigurations can allow adversaries to gain unauthorized access. The dynamic file inclusion mechanism in programming frameworks exploits directory traversal flaws. This vulnerability arises if unconstrained user input is used to construct files containing paths, such as form values or headers.

Attackers can use these inputs to traverse directories and access files. Attackers can exploit a web server's directory traversal vulnerability. They can upload a shell onto the server using code injection techniques, granting them unauthorized access, and facilitating defacement attacks (Chawda et al. 2021b).

Path-traversal attacks exploit vulnerable website parameters by including URLs referencing malicious code hosted remotely, enabling remote code execution and privilege escalation. A web browser's directory traversal flaw allows attackers to inject files into a server. When a web application fails to sanitize input properly, an attacker can use input and introduce path traversal characters. Additional web server files can be included by exploiting this vulnerability. Attackers can access and manipulate server files via directory traversal vulnerabilities. Attackers can use input and inject path traversal characters when incoming data is not processed adequately. By traversing the directory structure, they can include additional files (Chawda et al. 2021a).

The successful exploitation of this vulnerability confers malevolent actors the capability to peruse and amend capricious data, including information deemed sensitive. They may gain entry to files that house confidential data, jeopardize user accounts, or tamper with crucial system files. The implications of a directory traversal assault may extend far beyond the immediate impact, compromising the integrity, confidentiality, and availability of the adversely impacted system.

2.1.1 Types of Website Attacks, Impact and Risk

In the contemporary digital landscape, website attacks present a substantial peril, with various techniques capitalizing on the vulnerabilities inherent in web applications. These assaults can have catastrophic repercussions for entities and individuals alike.

Among the notable website attacks are SQL injection, DDoS attacks, and Cross-Site Scripting (XSS). Each attack possesses its distinct impact and level of risk.

Between 2017 and 2021, the global economic cost of cyberattacks was estimated to exceed one trillion dollars (Alharbi et al. 2021). In 2019, roughly two-thirds of organizations surveyed experienced cybersecurity threats. Data breaches and reputational damage are indirect damages caused by cybersecurity incidents and direct monetary losses. Businesses of all sizes have implemented technical and non-technical solutions to combat cybersecurity threats.

These measures include employee training programs, incident response plans, and ongoing risk assessments. Protecting the most valuable assets, ensuring customer trust, and mitigating the potential repercussions of cybersecurity incidents are the goals of these preventative and reactive measures. There may be legal action as a result.

Apache Struts Java framework, commonly used to develop Java web applications, has been vulnerable (Equifax Data Breach - Seven Pillars Institute 2021). Chinese cybersecurity researcher Nike Zheng discovered this critical software vulnerability. Apache Struts framework executes malicious code injected into the "content-type header" of HTTP requests due to the exposure. Numerous state and municipal governments have filed lawsuits against the well-known credit reporting company Equifax. The city of San Francisco filed a lawsuit against Equifax for allegedly engaging in unethical, dishonest, or unlawful business practices. Chicago also filed a lawsuit against Equifax for violating the Illinois Consumer Fraud and Deceptive Business Practices Act and the Chicago Consumer Fraud Ordinance (Alharbi et al. 2021).

2.2 RELATED WORK

This section comprehensively reviews the existing research on cyberattacks in India and emerging cyber security concerns and threats. By performing a literature review, we may understand the current state of knowledge in these areas and provide us with a deeper understanding of the challenges and risks associated with cybersecurity in India.

2.2.1 A Review Paper on Cyberattacks in India

Cybersecurity has emerged as a pivotal aspect of national security in today's interlinked world. Protecting the internet and the systems reliant on it entails a challenging and time-intensive undertaking. To establish a robust cybersecurity stance, meticulous planning, proactive measures, and frequent system evaluations are imperative (Cremer et al. 2022; Kaur et al. 2023).

Cybercriminals are constantly devising new, more covert, adaptable, and quicker strategies than their predecessors, enabling them to inflict more damage while evading detection. The range and variety of cyber threats continue to expand across multiple cyber spectrums. The scholars have collected information about the diverse categories of cyber hazards prevalent in Indian cyberspace. The information might have been sourced from several quarters, including cybersecurity reports, case studies, governmental organizations, and university studies. The identified types of perils have been scrutinized, encompassing phishing, social engineering, malware attacks, SQL injection, and man-in-the-middle attacks (Cremer et al. 2022; Kaur et al. 2023).

To comprehend these hazards' nature, pervasiveness, and repercussions, the author could have evaluated real-life scenarios, case studies, and statistical data. Both qualitative and quantitative methodologies have been employed to analyse the procured data and information. Statistical tools might have been utilized to unearth trends, patterns, and correlations between cyber menaces.

Traditional methods of preventing cyberattacks encounter difficulties in effectively countering the adaptable and intricate attack strategies utilized on the internet. Establishing regulatory mandates that uphold the security and integrity of systems is of utmost importance. Additionally, the ability of the system to achieve true resilience is impeded by the absence of both machine learning algorithms and artificial intelligence, as it primarily relies on pre-established rules for the classification and management of cybersecurity issues (Cremer et al. 2022; Kaur et al. 2023).

Traditional methods are frequently employed in India to mitigate cyber-attacks, which may not be fully equipped to handle flexible and complex attack strategies,

leaving systems vulnerable. The Internet of Things (IoT) has been identified as a weak point in Indian cyberspace, revealing vulnerabilities in IoT equipment that attackers can exploit. Neglecting to prioritize patching vulnerabilities and conducting rigorous application testing can expose systems to cyber threats. The text does not mention any validation techniques to verify the accuracy and reliability of the research findings. The author does not provide information about the specific scope and limitations of the study, which may affect the comprehensiveness and depth of the analysis (Tang et al. 2020).

2.2.2 Emerging Cyber Security India's Concern and Threats

Cybersecurity has evolved substantially and is recognized as a continuously expanding security challenge within current information, communication, and technology-oriented societies. Information and communications technology (ICT) has become more widely used, increasing reliance on it and the risk of cyberattacks. This tendency is especially relevant to India, which is dealing with increasing vulnerabilities as its digital infrastructure develops and its reliance on ICT for societal and economic advancement increases (Shairgojri & Dar 2022).

The investigation has uncovered multiple vulnerabilities in India's cybersecurity ecosystem, with finance, telecommunications, energy, and defense among the crucial sectors (Shairgojri & Dar 2022). These industries' susceptibility is attributed to significant factors such as infrastructure shortcomings, outdated security systems, a shortage of skilled cybersecurity personnel, and a lack of user knowledge. India's current cybersecurity regulatory framework has implemented several laws and regulations to enhance its cybersecurity posture. The Information Technology Act of 2000 and the recently passed Personal Data Protection Bill (Government Bill 2022) aim to increase data protection and privacy. However, further reforms and stricter implementation of existing standards are necessary.

Numerous hindrances to India's cybersecurity initiatives (Rising up to Cyber Security Challenges 2023) comprise a dearth of skilled cybersecurity personnel, the rapid proliferation of cyber threats, insufficient funding, and low user consciousness. The study report proffered remedies to these predicaments, for instance, investing in

cybersecurity education. In general, the findings and outcomes of the study paper accentuated India's pressing need to enhance its cybersecurity capabilities. To effectively counteract cyber perils and safeguard significant sectors of the country's economic and national security, preemptive measures and continual advancements in infrastructure, legislation, and human resources are imperative. Specific gaps and constraints surfaced while studying India's cybersecurity landscape. It is crucial to acknowledge these constraints since they present a prospect for further study and improvements in future research. The ensuing gaps and constraints have been identified as follows:

1. Data Availability and Reliability: Obtaining accurate and up-to-date data on cyber events, vulnerabilities, and security measures in India proved difficult. Access to sensitive information is restricted, cyber events are underreported, and cybersecurity threats are continually developing, making it impossible to collect a comprehensive and trustworthy dataset. The study was based on publicly accessible data and reports, which may limit quality and completeness (Shairgojri & Dar 2022).

2. Methodological Limitations: The study's research approach may have inherent limitations. Secondary sources, such as reports, research, and publicly accessible data, may create bias or restrictions in data comprehensiveness and accuracy. Furthermore, the research might have been hampered by a lack of resources, time, and access to important players for interviews or primary data gathering (Shairgojri & Dar 2022).

3. The report under consideration has delved into the current state of cybersecurity in India and has provided recommendations based on the findings. However, these suggestions' impact and long-term consequences remain to be ascertained. Subsequent research endeavors may scrutinize the proposed measures' implementation and results to establish their effectiveness and identify any potential loopholes that may arise (Kyle Chin 2023; Tiwari 2000). A study on the security of Indian police websites has been conducted, focusing on vulnerability assessment (Shairgojri & Dar 2022).

In the contemporary digital era, ensuring the security of websites, specifically traffic challan websites, is of utmost importance. Despite implementing password protection, encryption, authentication, and integrity, vulnerabilities may persist, making the websites susceptible to attacks such as SQL injection, cross-site scripting, header manipulation, and clickjacking. Addressing these concerns necessitates scanning URLs for embedded links and conducting vulnerability assessments. In India, researchers have concentrated their efforts on securing traffic challan websites, while others have adopted a penetration testing methodology to evaluate the security of Indian government websites (Shairgojri & Dar 2022).

Positive Technologies has released data on the top ten OWASP vulnerabilities in online applications, underscoring the importance of perpetual development in traffic challenges website security. Traffic police challan websites must analyse hacking techniques, fortify their defenses against attackers, and mitigate risks such as DoS attacks and SQL injection. Compliance with established security policies, fostering a culture of information security, and implementing sound management practices are paramount in preventing cyber-attacks and protecting sensitive data. This literary analysis delves into extant research and provides valuable perspectives and recommendations for enhancing the security of traffic challan websites (Shairgojri & Dar 2022).

This study identifies and analyses website security weaknesses. Three phases were involved in the vulnerability assessment: Information Collection, Vulnerability Scanning, and Report Generation. Information gathering is crucial to hacking or penetration testing (Sarker et al. 2023). System or application information was collected, such as network topology, IP addresses, server types, and software versions. As a result, attackers could detect flaws and vulnerabilities that could be exploited. Data was collected through port scanning, network mapping, OS fingerprinting, and banner grabbing.

Identifying vulnerabilities and addressing them before attackers exploit them requires proactive information-gathering activities. Hacking or penetration testing relies on scanning and assessing vulnerabilities. Malevolent entities could exploit the gaps

identified above in each system or application by identifying potential security weaknesses. SQL injection, cross-site scripting, and other typical exploits were detected using scanning tools and techniques. The vulnerabilities were meticulously analysed and prioritized based on their potential impact on the system. An effective remediation and risk mitigation strategy was formulated based on the data. Various vulnerabilities were found in the scrutinized hosts during the study. (Saba Kiran 2023). In summary, the results are as follows:

1. Missing HTTP Cookie Flags: This vulnerability was discovered on Host 1. It is critical to properly secure HTTP cookies using flags such as HTTP Only and Secure to prevent attackers from hijacking user sessions and accessing sensitive information (Nidecki 2020).
2. Clickjacking: The host used in this research is vulnerable to clickjacking. Using transparent or hidden frames, clickjacking tricks users into clicking on unintended links or buttons (Sahani & Randhawa 2021).
3. X-Frame-Options: The host used in this research lacks the security header; it was determined to be vulnerable to clickjacking assaults (Pohan et al. 2020).
4. Strict-Transport-Security: Hosts were susceptible to man-in-the-middle attacks because the Strict-Transport-Security header was missing (Siewert et al. 2022).
5. XSS Protection: Due to the lack of the XSS Protection header, most used in this research were determined to be vulnerable to XSS assaults (Pohan et al. 2020).

The research conducted on vulnerability assessment provided valuable information on the security flaws of websites. The study's conclusions are based on a narrow group of hosts and may not represent the entire spectrum of website vulnerabilities. The study's sample size and focus may restrict the results' generalizability to a larger environment. The vulnerability evaluation primarily focused on missing HTTP cookie flags, clickjacking, captcha, robots.txt, and mail server misconfiguration.

Interestingly, the study did not focus specifically on other vulnerabilities, such as SQL injection, remote file inclusion, or server misconfigurations. Therefore, it is

crucial to be aware that the findings could not completely encompass the vulnerabilities that websites may experience without investigating the contextual factors that may have contributed to such vulnerabilities. A thorough investigation of organizational policies, security practices, and external dangers was not conducted (Shairgojri & Dar 2022).

It is critical to comprehend the larger context of vulnerabilities to adopt adequate security solutions. The research did not explicitly mention external elements that may impact website security, such as the expanding threat environment, developments in hacking tactics, or new security standards. External factors such as these may significantly affect the success of risk assessment and mitigation measures. No comparisons were made by studying the susceptibility levels of the tested hosts to industry benchmarks or best practices. Without such comparisons, determining the severity or impact of the detected vulnerabilities concerning the hosts' overall security posture is difficult (Shairgojri & Dar 2022).

2.2.3 SQLi and Indian Websites: Unmasking the Truth

In today's linked world, cybersecurity is essential for protecting online data and services. However, cybercrime and assaults have significantly escalated due to the rapid development of technology and the widespread use of online platforms. It is essential to comprehend how these dangers impact both individuals and society. This literature review analyses the vulnerabilities found in Indian websites to give insight into the nation's cyber security condition.

This study's main goal is to study Indian websites that is vulnerable to SQL injection. The research also considers various defenses and measures to shield the Indian internet from the destructive activities of dishonest hackers and criminal actors. While securing websites is the focus, a look at the possibility of Android hacking is also included, highlighting the importance of awareness of this evolving online threat. By studying the nuances of cyber security and the vulnerabilities specific to Indian websites, this literature review aims to increase understanding of the evolving threat landscape in the digital domain.

By combining recent research and analysis, it aims to give readers insights that will help them develop effective strategies and policies that safeguard online platforms in India. Researchers conducted an in-depth review of the amount of knowledge and research on SQL injection, hacking, and cyber security. To create a theoretical foundation for the study, acquire pertinent academic books, reports, and articles from reliable sources (Leo & Bijimol 2019).

Define and explain the main terms concerning hacking, SQL injection, and cyber security, which involves defining ethical, black, and grey hat hackers and giving a comprehensive grasp of cyber security and its significance. Gather information on the common vulnerabilities found in Indian websites, paying particular attention to them.

SQL injection - To do this, it is necessary to locate and analyses relevant case studies, reports, and examples that show the prevalence and effect of SQL injection attacks in the Indian environment. Examine the gathered information to find prevalent flaws and trends in Indian websites vulnerable to SQL injection attacks. This report intends to shed light on the nation's cyber security status and highlight SQLi s dangers. Based on the vulnerabilities found, suggest potential fixes and defenses against SQL injection attacks. This step can involve advising developers to use better coding practices, putting security policies in place, and educating website owners and designers about the issue (Leo & Bijimol 2019).

In order to learn more about the security flaws in Indian websites, A poll was conducted among hacker and the result shows that SQL injection was shown to be the most frequent vulnerability reported in Indian websites out of the 35 responses. Approximately 64% of hackers claimed to have learned SQL injection at the beginning of their careers. According to 44% of the hackers who responded to the poll, there is a growing trend in the number of vulnerabilities on Indian websites. For the study, a sample of 50 susceptible websites from India was chosen.90% of the websites were discovered to be susceptible to SQL injection using the popular vulnerability dork 'php?id='. Thirty vulnerable websites lacked a login page, five had IP limitation protection, and ten allowed common hackers access (Leo & Bijimol 2019).

With only 35 responses, the poll of hackers to determine the security flaws in Indian websites had a limited sample size. This small sampling may not well represent the vulnerability landscape. The most common vulnerability identified in Indian websites, SQL injection, was the study's main area of interest. Cross-site scripting vulnerabilities (XSS), cross-site request forgeries (CSRF), and incorrect server configuration were not addressed. As a result, the results might not fully capture all available vulnerabilities (Leo & Bijimol 2019).

The detected vulnerabilities in Indian websites were not thoroughly examined in the report. It did not detail the repercussions or effects of these vulnerabilities, such as data breaches or economic losses. More study is required to understand further these vulnerabilities' severity and consequences (Leo & Bijimol 2019). Table 2.1 summarizes the methods, tools, and scope of related works.

2.3 SUMMARY

The literature review chapter provides an overview of cybersecurity in India. It brings attention to the growing number of cyberattacks and their impact on the nation's economy, public safety, and security. The emphasis is on taking preventative steps to deal with these risks, weaknesses in cyberspace, and how anybody may launch an attack, including nation-states, non-state entities, and people.

It emphasizes the importance of effective cybersecurity strategy and implementation and the function of employee accountability and awareness, examines how machine learning techniques could improve defense tactics, and discusses the many cyber threats common in India. The chapter also highlights the difficulties encountered in Indian cyberspace and the necessity of using machine learning and artificial intelligence. Overall, it offers insightful information on the state of cybersecurity today and recommends how to strengthen security measures.

Table 2.1 Cybersecurity approach: methods, tools, and scope

Author	Year	Article/Thesis Title	Method	Tools	Scope	Conclusion
Henil Vedant	2020	A Review Paper on Cyberattacks in India	Qualitative and quantitative	Machine learning algorithms	Protection of critical infrastructure	The author conducted a survey on common cyberattacks rather than performing a pen test to determine the current security posture.
Aadil Ahmed Shairgojri & Showkat Ahmad Dar	2022	Emerging Cyber Security India's Concern and Threats	Data Collection	Cybersecurity reports, case studies, government organizations, university studies	Gathering information on various cyber risks in Indian cyberspace	The author focuses on cyberattacks and prevention methods. Without knowing what vulnerability affect the device/application is hard to determine the fix
Sahil Durjar & Sushma Desai	2023	A Survey on Vulnerability Assessment of Indian Police Websites Security	Threat Analysis	Analysis of phishing, social engineering, malware assaults, SQL injection, man-in-the-middle attacks, etc.	Understanding the nature, prevalence, and impact of cyber threats in India.	The author focuses on a survey of Indian Police Websites which does not reflect the overall security posture of another sector
Bijimol T.K & Leo Joy	2019	SQLi and Indian Websites: Unmasking the Truth	Data Analysis	Qualitative and quantitative methodologies, statistical tools	Analysing acquired data to uncover trends, patterns, and correlations among different types of cyber threats.	The author focuses on SQL injection attack. However, there is many attacks toward website which not discussed

CHAPTER III

METHODOLOGY

3.1 INTRODUCTION

The main objective of this research is to identify vulnerabilities and risks associated with selected websites for the Indian websites and provide recommendations for enhancing website security. We will perform a series of security assessment tests to achieve this objective. Based on the result, we will determine whether the selected website for the Indian websites has any vulnerabilities. The research will conclude that the website is vulnerable and prone to cyberattacks based on security assessment results.

Figure 3.1 Standard PTES

3.2 RESEARCH DESIGN

This research will utilize open-source security scanning tools to conduct a security assessment for Indian websites. Automated scanning techniques were used in this study to identify common website vulnerabilities. The research design and approach for this research involved passive scanning techniques using open-source security scanning tools such as Nmap, online OpenVAS, SQLmap, and Qualys. The scans were conducted on web site for the Indian websites to identify potential security vulnerabilities and risks.

Scanning involves identifying open ports, services, and web applications on target systems. Open-source tools were used to identify vulnerabilities such as SQL injection, cross-site scripting, and insecure communication protocols. The scans' results were analysed to identify the most common vulnerabilities and security risks. The vulnerabilities were prioritised based on the risk level. Based on risk level, remediation steps should be carried out accordingly. Using open-source tools for website security assessment provided a cost-effective and efficient way to identify potential security risks in the Indian websites. This approach enabled us to assess many websites in a relatively short time. Automated tools should, however, be complemented by manual analysis and testing to ensure a comprehensive website security assessment.

The research design and approach for this research are focused on using open-source tools that provide a reliable and efficient means of identifying potential website security risks in the selected Indian websites.

3.2.1 Data Collection Methods

Data collection is critical to any research as it provides the necessary information to analyse and interpret research findings. In this research, we will assess the security posture of Indian websites We selected eight websites for this research based on a random sampling approach. As the Indian websites is diverse and vast, it was not feasible to include all Indian websites in this research. Therefore, a random sample of eight Indian websites was selected for the security assessment. Table 3.1 shows randomly selected eight Indian websites.

This sample covers the critical business sector. This size of sample enables a more in-depth analysis of each website, allowing for a more comprehensive understanding of website security practices within the Indian websites. Selected websites were chosen to represent a range of Indian websites, including e-commerce, healthcare, and food supplier websites. This diverse sample was selected to provide a comprehensive understanding of website security practices across different sectors within the Indian websites.

Our investigation encompasses seven distinct categories of websites: Food, IT services, Education, Medical, Finance, Online Travel Ticketing, and Health. The rationale behind selecting these specific types of websites remains varied and may be attributed to many factors. For instance, Online travel ticketing often deals with sensitive information and may be vulnerable to attacks such as data breaches, which could affect user privacy. On the other hand, health websites may contain confidential medical information that must be kept secure. Finance websites may handle financial transactions and personal information, while IT services websites may be vulnerable to attacks such as phishing or account takeovers.

Table 3.1 Sector for the selected Indian websites

No.	Web Site	Sector	Nature of the business
1	www.sprink.online	Food	Food Delivery
2	www.sifytechnologies.com	IT Service	IT Integration
3	www.sunriseuniversity.in	Education	Higher Education
4	www.amity.edu	Education	Higher Education
5	www.medtravels.in	Medical	Healthcare
6	www.i2ifunding.com	Finance	Loan Provider
7	www.busindia.com	Ticketing	Online Travel Ticketing
8	www.manipalhospitalsglobal.com	Health	Healthcare service provider

Finally, the goal is to assess the prevalence of specific security vulnerabilities across different sectors or industries. Therefore, a sample of eight Indian websites was randomly selected to provide a representative sample of Indian websites and enable a thorough assessment of website security practices within the Indian websites.

3.2.2 Tool Selection

Open-source tools were used to conduct a comprehensive website security assessment for the selected website for the Indian websites. These tools include Nmap Sublister, Online OpenVAS, Qualys and SQLmap, which are widely used for data collection and security auditing. Open-source tools offer several advantages, including customizing to meet specific research needs, free availability, and transparency through open-source code.

The Indian websites were scanned using Nmap, which provides network mapping and security auditing capabilities. Furthermore, SQLmap will detect SQL injection vulnerabilities in web applications. The website security assessment is further enhanced by using online OpenVAS and Qualys. Online OpenVAS can detect security weaknesses and misconfigurations in network infrastructure and web applications. Qualys scanner that can detect and identify security protocol flaws in web applications configuration and network infrastructure. These open-source tools can be used to provide a comprehensive and detailed report on the security posture of the Indian websites.

In summary, using open-source tools such as Nmap, SQLmap, online OpenVAS, and Qualys provides several benefits for website security assessment, including transparency, cost-effectiveness, and customization. These tools will provide a comprehensive and detailed report on the security posture of Indian websites.

3.3 CONFIGURATION AND SETUP

The host laptop for this research has an i7 processor and 256 GB of memory, with a 500GB SSD. The laptop runs on Windows 11 and has VMware Workstation Player installed. Kali Linux version 2022.4 is installed on VMware as a virtual machine to assess website security. A high-performance laptop with sufficient memory and storage is crucial for conducting an efficient website security assessment. Installing VMware Workstation Player and Kali Linux as virtual machines on the host laptop allows for a secure and isolated environment to perform website security testing. Kali Linux is a popular operating system for cybersecurity professionals with numerous built-in tools for conducting security testing and assessment.

The configuration and setup of the laptop with the appropriate tools and software is a critical first step in conducting a website security assessment. Several security assessment tools are installed on Kali Linux, including SQLmap, Sublister, and Nmap. Using these tools in combination allows for a comprehensive website security assessment, with the ability to identify and exploit potential vulnerabilities. Using Kali Linux as a virtual machine, the security assessment is conducted in a secure and isolated environment, minimizing the risk of damage or exposure to the host laptop. Installing

SQLmap, Sublister, and Nmap on Kali Linux provides a powerful toolkit for conducting a website security assessment, allowing for the detection of various vulnerabilities and potential risks.

Table 3.2 Lab hardware and software specification

Hardware / Software	Specification
Host Machine OS	Windows 11
Processor Specification	I7 12 Gen processor
Memory	256 GB
Virtual Machine	VMware Workstation Player
Guest Machine	Kali Linux 2022.4.
SSD Storage capacity	500 GB

The main reason for using the mentioned specification for this project is because it provides a flexible and secure environment for penetration testing and vulnerability assessment.

3.4 RECONNAISSANCE

The reconnaissance phase of the penetration testing process involves gathering information about the selected Indian website. The information is then used to identify attack vectors and vulnerabilities. Reconnaissance gives an insight into the website (Odun-Ayo et al. 2022). This phase gathers websites information based on Table 3.3.

Table 3.3 Indian websites

No	Website
1	www.sprink.online
2	www.sifytechnologies.com
3	www.sunriseuniversity.in
4	www.amity.edu
5	www.medtravels.in
6	www.i2ifunding.com
7	www.busindia.com
8	www.manipalhospitalsglobal.com

3.4.1 Sublister

Sublister is an open-source tool used for subdomain enumeration. Search engines like Google, Yahoo, and Bing are used to enumerate subdomains of websites Sublister is written in Python and uses libraries like Requests, BeautifulSoup, and argparse. Sublister is a valuable tool for reconnaissance and is commonly used by security researchers and penetration testers for discovering subdomains that can be targeted for further analysis and testing.

The output of Sublister is a list of subdomains associated with each selected Indian website. This information is helpful for this research to gather information about a target domain and identify potential vulnerabilities. It is possible to select a single Indian website or multiple Indian websites when running Sublister. The tool then searches through various public sources, including search engines, DNS records, and other tools, to discover subdomains associated with the selected Indian website.

The output of Sublister is typically a list of subdomains, along with information about their IP addresses and other relevant details. Information like this can be used for identifying potentially vulnerable subdomains or misconfigured on selected websites for Indian communities.

For this research, we will execute the command below. This command is to list all subdomains of the selected website for Indian communities. Once every subdomain has been located, we may utilise this information to conduct reconnaissance or enumeration, such as finding open ports or services connected to the subdomains. This data can contain potential vulnerabilities (Arunima Santhosh & Rinimol Kurian 2021).

sublister -d < Selected Indian website URL>

3.4.2 Nmap

Nmap, also called Network Mapper, is a popular reconnaissance tool. It can identify the port status and detect vulnerabilities by using the NSE script. Nmap scans can detect the type of OS the website is running and its version. It is an automated command-based

tool. This tool can also perform TCP/IP fingerprinting and DNS analysis. For our research, we will be executing the command as follows:

nmap -sV

Nmap will scan the specified hosts and try to identify the services using the access ports. It will then compare the fingerprints of the services against a database of known software and version information to determine what software and version is being used on the target system (Shah et al. 2019).

3.5 ENUMERATION AND SCANNING

Enumeration and scanning are crucial stages in website security assessment. They identify vulnerabilities, weaknesses, and potential exploits in the website's network infrastructure. A comprehensive scan may assist in discovering potential entry points, and these stages are crucial to the website security assessment procedure. We need to perform enumeration to understand better how the target system works and how it can be attacked. This activity involves collecting information about the network, and infrastructure, including IP addresses, device hostnames, and open ports.

Several methods, including DNS searches, network mapping, and banner capturing, can be used to enumerate. DNS queries find IP addresses associated with domain names. The IP address identifies the web server and other services. Network mapping is another technique to gather information about the website's infrastructure. It involves sending probes to different IP addresses to see which ones respond and identifying the operating system running on those IP addresses.

Banner grabbing is a technique that involves connecting to a service running on an IP address and retrieving its banner message. An analysis of the banner message can identify potential vulnerabilities, such as the software version. Enumeration is followed by scanning, which probes the target system for vulnerabilities. There are two primary types of scanning: port scanning and vulnerability scanning.

Port scanning scans the target system for open ports. Services running on the system can be identified as vulnerable to attack by open ports. In contrast, vulnerability scanning examines a system's software and operating system for known vulnerabilities. Vulnerabilities scanners detect SQL injection, cross-site scripting, and buffer overflows (Faircloth 2011).

3.5.1 SQLmap

This open-source tool detects and exploits SQL injection vulnerabilities in web applications. Attackers insert malicious SQL code into web form inputs or URL parameters, allowing them to gain unauthorized access, manipulate data, or deface websites. It automates the process of identifying and exploiting SQL injection vulnerabilities.

SQLmap analyses the HTTP responses from a target website to detect potential SQL injection vulnerabilities. Once a vulnerability is identified, SQLmap can utilize techniques such as blind, time, and error-based injection to exploit the vulnerability and retrieve sensitive data. SQLmap can perform other tasks such as database fingerprinting, password cracking, and privilege escalation.

SQLmap is crucial for website security assessment as SQL injection is a common and dangerous vulnerability in web applications. Detecting and remediating SQL injection vulnerabilities is critical for maintaining the security and integrity of web applications and preventing data breaches. SQLmap automates identifying and exploiting SQL injection vulnerabilities, saving time and effort in manual testing. SQL map detects and exploits SQL injection vulnerabilities in web applications. Website security assessments benefit from their ability to automate vulnerability detection and exploitation (Alanda et al. 2021). This research executed the command as follows:

sqlmap -u <website URL> --crawl 2

The command was set to scan the website up to two levels deep, meaning it will follow links within the target page to other pages within the same domain, but it will not go beyond this level. By running this scan, we identified potential vulnerabilities.

3.5.2 Online OpenVAS

For the convenience of this research, we used Online OpenVAS. It is a popular cloud-based open-source tool that provides flexibility to conduct website security assessments. Online OpenVAS allows performing vulnerability scans on websites and networks without installing and configuring the tool on their system. It is advised that before using the Online OpenVAS for our research projects, Registration is required prior to using this tool. Below are steps for performing the scanning.

1. To initiate the scanning process, click on the "New Scan" button and proceed to input the website's URL that requires scanning.
2. Scan settings should be configured according to the scanning needs, including the type and intensity of scans.
3. Start the scan and wait for it to complete (Aksu et al. 2019).

3.5.3 Qualys Crawls

Security of the website is critical to online transactions or activity. Therefore, it is essential to ensure that Indian websites are secured against potential cyber threats. To assess the security of Indian websites, one tool that can be utilized is the SSL Server Test by Qualys. This tool analyses a website's SSL implementation and identifies potential vulnerabilities or weaknesses that could compromise its security.

A scan of Indian websites was conducted using the SSL Server Test by Qualys. The results of this assessment provide valuable insights into the current state of website security and identify areas for improvement. This evaluation provided an overall rating and detailed feedback on the SSL encryption implementation. The rating system ranges from A and A+ for secure encryption to B, C, D, E, and F, indicating the need for updates or improvements. The rating T signifies that the website is untrusted due to certificate expiration. If SSL implementation is absent, the evaluation report will indicate the same. Each website takes about 15 minutes to evaluate SSL implementation (Thomchick & Nicolas-Rocca 2018).

3.5.4 Nmap

Nmap is used again with a vulnerability script to identify any existing vulnerabilities. When the command below was executed, Nmap scanned the selected Indian website for any vulnerability.

nmap --script vuln <website URL>

To check whether the selected Indian website is vulnerable to the csrf command below is executed used the followings:

nmap --script http-csrf <website URL>

3.6 VULNERABILITY ASSESSMENT

Vulnerability assessment is a critical component of website security assessment. It is a process that helps to identify and analyse the vulnerabilities on selected Indian websites. The objective of vulnerability assessment is to assess the risk associated with each identified vulnerability, determine its potential impact on the selected Indian website, and prioritise remediation action accordingly. Identifying vulnerabilities in a selected Indian website and assessing their severity and impact is part of a vulnerability assessment. Vulnerability assessments help organisations prioritise cybersecurity efforts, allocate resources more effectively, and reduce cyberattack risks.

The vulnerability assessment activity should be carried out periodically. The vulnerability assessments can be conducted using automated scans. Vulnerability assessments help to identify vulnerabilities and provide recommendations to improve website security (Kritikos et al. 2019).

In this research, the identified vulnerability was mapped based on the severity in Table 3.4. Usually, the severity rating runs from 0 to 10, with ten being the most serious. Organisations may prioritise vulnerabilities depending on their severity and take the necessary action to reduce the risks caused by such vulnerabilities. This severity

rating will used during our vulnerability assessment phase whereby identified vulnerabilities on Indian websites will be categorized based on Table 3.4.

Table 3.4 Severity rating based on CVSS

Score	Severity
0.0 - 3.9	Low severity
4.0 - 6.9	Medium severity
7.0 - 8.9	High severity
9.0 - 10.0	Critical severity

3.7 MITIGATION PLAN FOR IDENTIFIED VULNERABILITIES

Once the severity score was finalized, the next step was prioritizing remediation efforts based on the severity score. Each vulnerability was matched against the rating provided by cve.com to develop the score. A mitigation plan ideally should be carried out based on the severity of the vulnerability, whereby the highest severity rating should be addressed first, followed by those with lower severity ratings. Sometimes critical assets might take high priority even though the rating is below critical or below high. The mitigation activity involves various activities, such as patching vulnerable systems, reconfiguring network settings, or updating software or hardware components. Security policies and procedures should align with industry best practices (Nowak et al. 2023). Organizations should also conduct regular vulnerability assessments to promptly identify and address new vulnerabilities. Below is the process of vulnerability assessment. Figure 3.2 summarizes the vulnerability assessment process.

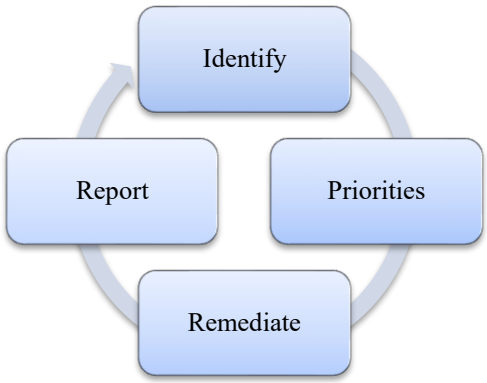


Figure 3.2 Vulnerability Assessment Process

3.7.1 Parameter

A crucial aspect of the website security assessment process is using parameters to extract information from the selected Indian website. Every tool used in this research required a unique set of parameters to obtain the necessary information. These parameters vary depending on the specific tool used.

Each parameter serves a critical role in the security assessment process. For example, tools such as SQLmap, Nmap, Qualys, and online OpenVAS all require specific parameters to identify potential vulnerabilities and assess the security of a website. We can obtain accurate and relevant information about their website's security posture by correctly identifying the relevant parameters. Failure to correctly identify and utilize these parameters may lead to incomplete or inaccurate assessments, potentially leaving the website vulnerable to attacks.

In summary, the parameters used in the website security assessment process are critical to obtaining accurate and relevant information about a website's parameters. Correctly identifying and utilizing these parameters are necessary to perform comprehensive assessments and ensure the website's security. The list of parameters used is as followings:

sublister -d < Selected Indian website URL>

By specifying the -d parameter and providing the selected Indian website URL, the tool will search for subdomains associated with the website. This information can help identify potential attack vectors and vulnerabilities related to the website's parameters.

nmap -sV

The -sV parameter instructs Nmap to probe for service versions using various methods, including sending version-specific probes and comparing service banners against a database of known version numbers. By identifying the service versions

running on the website, website owners can determine if any outdated software versions are running, which could potentially contain known vulnerabilities.

nmap --script vuln <website URL>

The --script vuln parameter tells Nmap to execute its vulnerability-related scripts against the target website. These scripts use techniques to identify known vulnerabilities and potential exploits associated with the website's parameters.

nmap --script http-csrf <website URL>

The --script http-csrf parameter instructs Nmap to execute its HTTP CSRF detection script against the target website. This script sends a sequence of HTTP requests to the website to determine if it is vulnerable to CSRF attacks. A CSRF vulnerability occurs when an attacker can execute unauthorized actions on a website on behalf of an authenticated user.

sqlmap -u <website URL> --crawl 2

The -u parameter specifies the URL of the target website, while the --crawl parameter specifies the number of links from the initial URL that SQLmap should follow. For example, if --crawl two is specified, SQLmap will follow links on the initial URL and the first level of linked pages to identify potential SQL injection vulnerabilities. SQLmap sends malicious SQL queries to the target website to identify vulnerabilities. If a vulnerability is found, SQLmap can extract sensitive data, such as usernames and passwords, or execute arbitrary SQL commands.

online openvas < WebsiteURL>

By navigating to the online OpenVAS website and inputting the selected Indian website URL, we can perform a comprehensive vulnerability scan. By inputting the selected Indian website URL into Qualys, we can perform a comprehensive security assessment to identify potential vulnerabilities in their website's parameters. Once a

security assessment is initiated, Qualys will scan the selected Indian website for potential security weaknesses.

3.8 SUMMARY

This research aims to identify vulnerabilities and risks associated with Indian websites and provide recommendations for enhancing website security. The research utilized a four-step web security assessment methodology: Reconnaissance, Enumeration and Scanning, Vulnerability Assessment, and Reporting. The research used open-source security scanning tools, such as Nmap, OpenVAS, SQLmap, and Qualys, to perform passive scans on eight randomly selected Indian websites representing different sectors.

The vulnerabilities were prioritized based on the risk level, and recommendations were provided for remediation. The research concludes that automated tools efficiently identify potential security risks, but manual testing and analysis are necessary for a comprehensive website security assessment. The sample size of eight websites was chosen to provide a representative sample across different sectors within the Indian websites.

CHAPTER IV

RESULTS AND DISCUSSION

4.1 INTRODUCTION

This chapter presents the results and discussion of the selected website for the Indian websites. The focus of this chapter is to reveal the result and discuss the result and propose a mitigation plan.

The assessment was conducted using open-source tools. The findings of the assessment were categorized based on their severity, ranging from low to critical. The assessment results revealed several vulnerabilities in the selected Indian website, including obsolete software and insecure settings. Malicious actors could exploit these vulnerabilities to compromise the website's security and steal sensitive information.

Updating software and plugins, creating more substantial password restrictions, and carrying out routine security audits are advised. These measures can help improve selected Indian website security posture and protect it from cyber-attacks.

4.2 WEBSITES SECURITY ASSESSMENT

In the initial phase of our website security assessment, we conducted a reconnaissance activity to gather information about the selected Indian website. We used sublister and Nmap to search for subdomains and perform port scanning to obtain information on the service and port running on selected Indian web site. The reconnaissance activity on the selected Indian website was summarized in Table 4.1.

For the second phase of website security assessment, we conducted an enumeration and scanning activity to determine whether the selected Indian website has

a potential vulnerability. We used Nmap, SQLmap, Online OpenVAS, and Qualys to perform a crawl to determine whether the selected Indian website is vulnerable. The result of enumeration and scanning activity on selected Indian websites is compiled in Table 4.2.

4.3 DISCUSSION OF WEBSITE SECURITY ASSESSMENT AND VULNERABILITY MITIGATION

The website security assessment was conducted for Indian websites to identify the current state of website security and suggest improvements. The assessment was conducted using a combination of manual testing and automated tools.

Many Indian websites are vulnerable to vulnerabilities compromising their security and user information. We found vulnerabilities for cross-site scripting (XSS), SQL injection, and cross-site request forgery. The attacker can alter website content or steal sensitive data due to vulnerability (Sahren et al. 2019). The outcome showed that Indian websites were vulnerable to phishing and password-guessing attacks because they lacked basic security precautions, including SSL/TLS encryption and multi-factor authentication.

Overall, the assessment results indicated a pressing need for Indian websites to improve their security measures to protect their users' personal information and ensure their websites' integrity. The following section discusses in detail the result and the recommendations suggested to improve website security measures based on Tables 4.1 and 4.2 of the first and second phases.

Table 4.1 Reconnaissance activity

Website	Number of Subdomain	Name of Subdomain	State	Services	Version
www.manipalhospitalsglobal.com	2	stage.manipalhospitalsglobal.com	80 TCP Open	HTTP	None
		test.manipalhospitalsglobal.com	443 TCP Open	HTTP	None
www.medtravels.in	8	cpcontacts.medtravels.in	80 TCP Open	HTTP	None
		www.medtravels.in	443 TCP Open	HTTP	None
		cpanel.medtravels.in			
		cpcalendars.medtravels.in			
		webdisk.medtravels.in			
		ns2.medtravels.in			
		ns1.medtravels.in			
		mail.medtravels.in			
www.i2ifunding.com	8	demo.i2ifunding.com	80 TCP Open	HTTP	Apache httpd 2.4.18
		www.i2ifunding.com	443 TCP Open	HTTP	Apache httpd 2.4.18
		api.i2ifunding.com			
		www.demo.i2ifunding.com			
		apiv1.i2ifunding.com			
		analytics.i2ifunding.com			
		i2ifunding.com			
		www.api.i2ifunding.com			

to be continued…

…continuation

https://www.amity.edu	16	admissions2022.amity.edu amity.edu addoe.amity.edu pun.amity.edu aitd.amity.edu admissions2018.amity.edu africa.amity.edu alumni.amity.edu aisg46.amity.edu aismv.amity.edu www.addoe.amity.edu mail.amity.edu london.amity.edu aup.amity.edu aiss.amity.edu aie.amity.edu	80 TCP Open 443 TCP Open	HTTP HTTP	Microsoft IIS httpd 10.0 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
www.sunriseuniversity.in	12	www.sunriseuniversity.in webdisk.sunriseuniversity.in hostmaster.sunriseuniversity.in cpanel.sunriseuniversity.in	80 TCP Open 443 TCP Open	HTTP HTTP	Nginx Nginx

to be continued…

…continuation

		newweb.sunriseuniversity.in autodiscover.sunriseuniversity.in cpcalendars.sunriseuniversity.in cpcontacts.sunriseuniversity.in erp.sunriseuniversity.in www.erp.sunriseuniversity.in mail.sunriseuniversity.in webmail.sunriseuniversity.in			
www.busindia.com	4	m.busindia.com media.busindia.com mail.busindia.com hotel.busindia.com	80 TCP Open 443 TCP Open	HTTP HTTP	Apache httpd 2.4.37 (centos) Apache httpd 2.4.37 (centos)
www.sifytechnologies.com	9	learning.sifytechnologies.com elearning.sifytechnologies.com europe.sifytechnologies.com elearningeu.sifytechnologies.com corp.sifytechnologies.com beta.sifytechnologies.com careers.sifytechnologies.com m.sifytechnologies.com	80 TCP Open 443TCP Open	HTTP HTTP	HTTP Apache httpd 2.4.37 (centos) HTTP Apache httpd 2.4.37 (centos)

to be continued…

…continuation

		stage.sifytechnologies.comhotel.busindia.com			
www. www.sprink.online	3	epicenter.sprink.online	80 TCP Open	HTTP	Apache httpd 2.4.37 ((centos)
		token.sprink.online	443 TCP Open	HTTP	Apache httpd 2.4.37 ((centos)
		special.sprink.online			

Table 4.2 Result of enumeration and scanning

Website	SQLmap Result	Nmap Result	Qualys Ratings	Online OpenVAS Result
www.sprink.online	No SQL Injection Vulnerability Found	Apache/2.4.54 (Debian)] PHP 7.4.33 Default Credentials www.sprink.online, | root:&lh;empty> - Valid credentials Http server 2.4.5 vulnerable to CVE-2023-27522 CVE-2022-37436 CVE-2022-36760	B	Cross-Domain Misconfiguration Absence of Anti-CSRF Tokens Content Security Policy (CSP) Header Not Set Missing Anti-clickjacking Header Vulnerable JS Library Cross-Site Request Forgery (CSRF) Missing 'Secure' Cookie Attribute (HTTP) Cookie Without Secure Flag Cross-Domain JavaScript Source File Inclusion

to be continued…

…continuation

				Strict-Transport-Security Header Not Set Server Leaks Version Information via "Server" HTTP Response Header Field Server Leaks Information via "X-Powered-By" HTTP Response Cookie with SameSite Attribute None
www.sifytechnologies.com	No SQL Injection Vulnerability Found	CVE-2022-31813 CVE-2022-23943 CVE-2022-22720 CVE-2021-39275 CVE-2021-26691 CVE-2021-33193 CVE-2021-40438 CVE-2020-35452 CVE-2022-28615 CVE-2022-22721	A+	SQL Injection - MySQL SQL Injection - Hypersonic SQL - Time Based SQL Injection - Oracle - Time Based SQL Injection - PostgreSQL - Time Based

to be continued…

…continuation

CVE-2020-1927	SQL Injection - SQLite
CVE-2022-30556	Secure Pages Include Mixed Content (Including Scripts)
CVE-2022-29404	HTTPS to HTTP Insecure Transition in Form Post
CVE-2022-28614	Cross-Domain Misconfiguration
CVE-2022-26377	Absence of Anti-CSRF Tokens
CVE-2022-22719	Missing Anti-clickjacking Header
CVE-2021-34798	Vulnerable JS Library
CVE-2021-26690	CSP: Wildcard Directive
CVE-2020-1934	Multiple X-Frame-Options Header Entries
CVE-2020-11985	
CVE-2023-25690	
CVE-2022-37436	
CVE-2022-36760	

to be continued…

…continuation

www.sunriseuniversity.in	Found SQL Injection vulnerability. Parameter: fn (GET) Type: boolean-based blind Title: AND boolean-based blind - WHERE or HAVING clause Payload: fn=admission-procedure' AND 7346=7346 AND 'kZpL'='kZpL Type: time-based blind Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP) Payload: fn=admission-procedure' AND (SELECT 2563 FROM (SELECT(SLEEP(5)))Hali) AND 'VtWo'='VtWo	No Vulnerability Found	B	Missing 'Secure' Cookie Attribute (HTTP) Cross-Domain Misconfiguration Absence of Anti-CSRF Tokens Content Security Policy (CSP) Header Not Set Missing Anti-clickjacking Header Vulnerable JS Library Missing 'Secure' Cookie Attribute (HTTP) Cookie Without Secure Flag Cross-Domain JavaScript Source File Inclusion

to be continued…

…continuation

				Server Leaks Version Information via "Server" HTTP Response Header Field
				Server Leaks Information via "X-Powered-By" HTTP Response
				Cookies with SameSite Attribute None
www.amity.edu	No SQL Injection Vulnerability Found	Web server Microsoft-IIS/10.0 is vulnerable to Found Cross-site request forgery	B	SSL/TLS: Report Vulnerable Cipher Suites for HTTPS
				SSL/TLS: Report Weak Cipher Suites
				SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection

to be continued…

…continuation

www.medtravels.in	No SQL Injection Vulnerability Found	Apache 2.2.15 is vulnerable to CVE-2023-28625	A	HTTP to HTTPS Insecure Transition in Form Post Cross-Domain Misconfiguration Absence of Anti-CSRF Tokens Vulnerable JS Library Content Security Policy (CSP) Header Not Set Missing Anti-clickjacking Header Cookie No HttpOnly Flag Cookie Without Secure Flag Cross-Domain JavaScript Source File Inclusion Strict-Transport-Security Header Not Set

to be continued…

…continuation

				X-Content-Type-Options Header Missing
				Server Leaks Version Information via “Server" HTTP Response Header Field
				Cookie without SameSite Attribute
www.i2ifunding.com	No SQL Injection Vulnerability Found	Http Server 2.4.18 is vulnerable to CVE-2022-31813 CVE-2022-23943 CVE-2021-34798 CVE-2021-44790 CVE-2021-39275 CVE-2021-26691 CVE-2021-40438 CVE-2020-35452 CVE-2022-28615 CVE-2021-44224 CVE-2022-22721 CVE-2020-1927 CVE-2022-30556 CVE-2022-29404 CVE-2022-28614	B	Cross-Domain Misconfiguration Absence of Anti-CSRF Tokens Vulnerable JS Library Content Security Policy (CSP) Header Not Set Missing Anti-clickjacking Header Cross-Domain JavaScript Source File Inclusion

to be continued…

…continuation

		CVE-2022-26377 CVE-2022-22719 CVE-2021-34798 CVE-2021-33193 CVE-2021-26690 CVE-2020-1934 CVE-2022-37436 CVE-2022-36760		Strict-Transport-Security Header Not Set X-Content-Type-Options Header Missing Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) Server Leaks Version Information via "Server" HTTP Response Header Field
www.busindia.com	No SQL Injection Vulnerability Found	Website is vulnerable to Cross-site request forgery. Found the following possible CSRF vulnerabilities: Path: http://busindia.com:443/ Form id: searchform Form action: /home Path: http://busindia.com:443/Morning-Star-Travels-online-bus-booking	B	osTicket < 1.14.3 Multiple Vulnerabilities osTicket < 1.14.8, 1.15.x < 1.15.4 Multiple Vulnerabilities

to be continued…

…continuation

	Form id: searchform Form action: /home Path: http://busindia.com:443/Roadlink-India-online-bus-booking Form id: searchform Form action: /home Path: http://busindia.com:443/National-Travels1nts1-online-bus-booking Form id: searchform Form action: /home Apache Server:2.4.37 is vulnerable to CVE-2022-31813 CVE-2022-23943 CVE-2022-22720 CVE-2021-44790 CVE-2021-39275 CVE-2021-26691 CVE-2020-11985 CVE-2021-40438 CVE-2020-35452 CVE-2022-28615 CVE-2021-44224	osTicket < 1.16.6, 1.17.x < 1.17.3 Multiple XSS Vulnerabilities Cross-Domain Misconfiguration Absence of Anti-CSRF Tokens Vulnerable JS Library Content Security Policy (CSP) Header Not Set Missing Anti-clickjacking Header Cross-Domain JavaScript Source File Inclusion Strict-Transport-Security Header Not Set

to be continued…

…continuation

	CVE-2022-22721 CVE-2020-1927 CVE-2022-30556 CVE-2022-29404 CVE-2022-28614 CVE-2022-26377 CVE-2022-22719 CVE-2021-36160 CVE-2021-34798 CVE-2021-33193 CVE-2021-26690 CVE-2020-1934 CVE-2023-25690 CVE-2022-37436		Server Leaks Version Information via "Server" HTTP Response Header Field Server Leaks Information via "X-Powered-By" HTTP Response
www.manipalhospitalsglobal.com	Apache httpd Website is vulnerable to Cross-site request forgery. Found the following possible CSRF vulnerabilities: https://www.manipalhospitals.com:443/ Form id: search_result Form action: https://www.manipalhospitals.com:443/search Path: https://www.manipalhospitals.com:443/ Form id: hospital_id Form action: https://www.manipalhospitals.com:443/search_doctors\ Path: https://www.manipalhospitals.com:443/ Form id: write_to_coo Form action: https://www.manipalhospitals.com:443/save_write_to_coo	A	Directory Browsing - Apache 2 Cross-Domain Misconfiguration Vulnerable JS Library Absence of Anti-CSRF Tokens Content Security Policy (CSP) Header Not Set

to be continued…

…continuation

	Path: https://www.manipalhospitals.com:443/written-testimonial/mr-k-eswar-sai-ganesh Form id: search_result Form action: https://www.manipalhospitals.com:443/search Path: https://www.manipalhospitals.com:443/written-testimonial/mr-k-eswar-sai-ganesh Form id: write_to_coo Form action: https://www.manipalhospitals.com:443/save_write_to_coo Path: https://www.manipalhospitals.com:443/vijayawada/ Form id: search_result Form action: https://www.manipalhospitals.com:443/vijayawada/search Path: https://www.manipalhospitals.com:443/vijayawada/ Form id: speciality_id Form action: https://www.manipalhospitals.com:443/vijayawada/search_doctors Path: https://www.manipalhospitals.com:443/vijayawada/ Form id: write_to_coo Form action: https://www.manipalhospitals.com:443/vijayawada/save_write_to_coo Found the existence of a potential admin folder on a website /admin/	Missing Anti-clickjacking Header Cookie Without Secure Flag Cross-Domain JavaScript Source File Inclusion X-Content-Type-Options Header Missing Strict-Transport-Security Header Not Set Server Leaks Version Information via "Server" HTTP Response Header Field

Table 4.3 Discussion and mitigation website

Website	No of vulnerability found	Risk	Impact	Mitigation
www.manipalhospitalsglobal.com	12	High	High	Enable Anti-CSRF tokens. implement strong authentication mechanisms for accessing the admin folder. Upgrading Apache HTTP Server. Disable directory browsing. Implementing cross-domain policies to restrict access. Enable use anti-CSRF tokens. Upgrade JavaScript libraries. Enable Content Protection Policy (CSP) Implement anti-clickjacking header. Setting the SameSite attribute to Strict or Lax is advisable. Add a CSP header to their HTTP response. Enable the STS header. Deleting or changing the "Server" header in the server configuration file. Set to “no sniff” option for X-Content-Type-Options header
www.medtravels.in	27	High	Medium	Upgrading Apache HTTP Server to version 2.4.10 or later. Upgrade mod_auth_openidc to version 2.4.13.2. Making configuration changes to enforce HTTPS protocol for all form submissions and redirect HTTP requests to HTTPS. Enforce HTTPS for all form submissions and redirect HTTP requests to HTTPS. Enable Anti-CSRF tokens. Upgrade JavaScript libraries Add a CSP header to their HTTP response. Disable TLS 1.0 and TLS 1.2 and enable TLS 1.2 and TLS 1.3 Implementing cross-domain policies to restrict access. Implement anti-clickjacking header. Enable secure Cookie Attribute (HTTP)

to be continued…

…continuation

				Enable the STS header. Deleting or changing the "Server" header in the server configuration file Restrict the information by removing the "X-Powered-By" header in Apache. Set to “no sniff” option for X-Content-Type-Options header. Setting the SameSite attribute to Strict or Lax is advisable
www.i2ifunding.com	35	High	High	Upgrading Apache HTTP Server. Enable use anti-CSRF tokens. Add a CSP header to their HTTP response. Upgrade JavaScript libraries Implement anti-clickjacking header. Enable the STS header. Deleting or changing the "Server" header in the server configuration file Implementing cross-domain policies to restrict access. Enable secure Cookie Attribute (HTTP) Restrict the information by removing the "X-Powered-By" header in Apache. Set to “no sniff” option for X-Content-Type-Options header
www.sifytechnologies.com	52	High	High	Upgrading Apache HTTP Server. Easy Smooth Scroll Links WordPress plugin. Ensure sanitized input data and use prepared statements to prevent SQL injection attacks. Use parameterized queries and avoid using dynamic SQL queries. Ensure all content loaded on secure pages uses HTTPS. We ensure all forms submitted from secure pages use HTTPS. Implement cross domain policy to prevent unauthorized access. Enable use anti-CSRF tokens. Implement anti-clickjacking header. implement CSP to prevent unsafe styles from being executed

to be continued…

…continuation

www.sunriseuniversity.com	15	High	High	Ensure sanitized input data and use prepared statements to prevent SQL injection attacks. Disable TLS 1.0 and TLS 1.2 and enable TLS 1.2 and TLS 1.3 Enable use anti-CSRF tokens. Implementing cross-domain policies to restrict access. Add a CSP header to their HTTP response. Implement anti-clickjacking header. Upgrade JavaScript libraries Enable secure Cookie Attribute (HTTP) Set to “no sniff” option for X-Content-Type-Options header. Deleting or changing the "Server" header in the server configuration file Setting the SameSite attribute to Strict or Lax is advisable
www.amity.edu	4	Medium	Low	Enable use anti-CSRF tokens. Ensure only to use only strong and secure encryption protocols. Disable TLS 1.0 and TLS 1.2 and enable TLS 1.2 and TLS 1.3
www.busindia.com	42	High	High	Enable use anti-CSRF tokens. Upgrading Apache HTTP Server. Reference browsers must be permitted to adopt forward secrecy measures. Upgrade osTicket. Restrict access or remove access to sensitive file. Implementing cross-domain policies to restrict access. Enable use anti-CSRF tokens. Upgrade JavaScript libraries. Implement CSP to prevent unsafe styles from being executed. Implement anti-clickjacking header. Add a CSP header to their HTTP response. Enable the STS header. Deleting or changing the "Server" header in the server configuration file

to be continued…

…continuation

www.sprink.online	17	High	High	Root user account should have complex password. Upgrade Apache HTTP Server version 2.4.55. Disable TLS 1.0 and TLS 1.2 and enable TLS 1.2 and TLS 1.3. Enable reference browsers permit to enable forward secrecy. Implementing cross-domain policies to restrict access. Enable Anti-CSRF tokens. Enable use anti-CSRF tokens. Add a CSP header to their HTTP response. Implement anti-clickjacking header. Upgrade JavaScript libraries. Enable secure Cookie Attribute (HTTP) Set to “no sniff” option for X-Content-Type-Options header. Enable the STS header. Deleting or changing the "Server" header in the server configuration file. Restrict the information by removing the "X-Powered-By" header in Apache. Setting the SameSite attribute to Strict or Lax is advisable

4.3.1 Discussion of Sublister Result

Based on the results from the sublister tool. It appeared that the website www.sprink.online has three subdomains, www.sifytechnologies.com has ten subdomains, www.sunriseuniversity.com has twelve subdomains, www.amity.edu has sixteen subdomains, www.medtravel.in has eight subdomains, www.i2ifunding.com has eight subdomains, www.busindia.com has five subdomains and www.manipalhospitalsglobal.com has two subdomains. Subdomains may differ from the primary domain in configurations and security settings.

4.3.2 Discussion of Nmap with NSE Script Result

For www.sprink.online,the Nmap scan showed that Apache httpd version 2.4.54 is operating on ports 80 (HTTP)and 443 (HTTPS). The results indicated that the root user account on the website has an empty password, indicated by the "&lh;empty-" notation. An empty root password may make the website vulnerable to attack. Strong passwords must be used for all user accounts, especially for powerful accounts like root. It is advised that the website owner immediately updates the root password to one that is robust and safe and evaluates the security settings on the whole website to ensure that all potential vulnerabilities are covered.

This phase might involve implementing extra security safeguards like two-factor authentication, intrusion detection systems, and regular security audits to prevent unauthorised access and safeguard user data. Nmap tool result show that www.sifytechnologies.com and www.sunriseuniversity.com uses HTTPS (port 443), for encrypts communication between the server and the client and keeps hackers from gathering and reading private information. operating on ports 80 (HTTP) and 443 (HTTPS).

For www.amity.edu, The scan result revealed that there are two open ports on the server which is Port 80 is used for HTTP traffic, while port 443 is for HTTPS traffic. The web server type running on port 80 is Microsoft IIS httpd 10.0, a web server developed by Microsoft for use on Windows, indicating that the website is likely

running on a Windows server. The web server type running on port 443 is Microsoft HTTPAPI httpd 2.0, a Windows operating system component that allows applications to communicate over HTTP and HTTPS. SSDP/UPnP protocols often use this service for network discovery and communication.

One vulnerability has been identified that could potentially compromise the web server if exploited by an attacker. The web server is vulnerable to CSRF vulnerabilities. Various form IDs and form actions are being flagged as potentially vulnerable. These are the three pages vulnerable to CSRF vulnerability: http://amity.edu:443/, http://amity.edu:443/infra-fine-arts.aspx, and http://amity.edu:443/about-stalwarts.aspx. The first page, the one at http://amity.edu:443/, contains a form identified as "form1" and transfers data to the website's root directory ("/"). The second page, found at http://amity.edu:443/infra-fine-arts.aspx, contains a form labelled "aspnetform" that redirects to "/infra-fine-arts.aspx".

Finally, the third page, http://amity.edu:443/about-stalwarts.aspx, had a form with the ID "aspnetform" and an action of "/about-stalwarts.aspx". The vulnerabilities were identified by examining the form IDs and form actions of the web pages, which could allow attackers to craft malicious requests that appear to be coming from the victim's browser.

For www, medtravels.in, The scan revealed that the service is running on ports 80 and 443. Port 80 is typically used for HTTP traffic, while port 443 is for HTTPS traffic. The web server type is running on Apache 2.2.15. Using absolute technologies is known to have vulnerability; therefore, upgrading to a newer version of the Apache server is better.

For www.121funding.com,The Nmap scan revealed that there are two open ports on the server. The web server is running on Apache httpd version 2.4.18.

For www.busindia.com, revealed that there are two open ports on the server, which are ports 80 and 443. The web server is running on Apache httpd 2.4.37.

For www.manipalhospitalsglobal.com revealed that Port 80 is used for HTTP traffic, while port 443 is for HTTPS traffic. The web server is running on Apache httpd.

For www.sprink.online,we found four vulnerabilities that will impact the web server if an attacker exploits it, www.sifytechnologies.com with twenty-three vulnerabilities have been found, www.amity.edu found one vulnerability identified that could potentially compromise the web server if exploited by an attacker. The web server is vulnerable to CSRF vulnerabilities. Various form IDs and form actions are being flagged as potentially vulnerable. These are the three pages vulnerable to CSRF vulnerability: http://amity.edu:443/, http://amity.edu:443/infra-fine-arts.aspx, and http://amity.edu:443/about-stalwarts.aspx.

The first page, the one at http://amity.edu:443/, contains a form identified as "form1" and transfers data to the website's root directory ("/"). The second page, found at http://amity.edu:443/infra-fine-arts.aspx, contains a form labelled "aspnetform" that redirects to "/infra-fine-arts.aspx." Finally, the third page, http://amity.edu:443/about-stalwarts.aspx, had a form with the ID "aspnetform" and an action of "/about-stalwarts.aspx". The vulnerabilities were identified by examining the form IDs and form actions of the web pages, which could allow attackers to craft malicious requests that appear to be coming from the victim's browser.

For www.medtravels.in was discovered that Apache 2.2.15 is susceptible to one vulnerability which can significantly impact the web server. For www, 121funding.com Using Nmap nse, we found Apache 2.4.18 is vulnerable to twenty-three vulnerabilities that will impact the web server if an attacker exploits it. For www.busindia.com.we found Apache 2.4.18 is vulnerable to twenty-six vulnerabilities that will impact the web server if the attacker exploits it. This webpage, located at http://busindia.com:443/, is vulnerable to CSRF attacks. This flaw can be used to carry out unauthorised actions for a victim user who has logged in to the website. The vulnerability is present in the website's search form, with form ID "searchform" and form action "/home." The same vulnerability is also present in the search forms of other pages on the website, such as "Morning-Star-Travels-online-bus-booking," "Roadlink-India-online-bus-booking," and "National-Travels1nts1-online-bus-booking".

An attacker can create a specially crafted web page or email that contains a malicious form that targets the vulnerable search form on the busindia website. When a victim user accesses this page or clicks on the email link, the form is submitted to the busindia website on behalf of the victim user without their knowledge or consent, thus can lead to unauthorised actions like booking tickets or changing the user's account information. To mitigate this vulnerability, it is recommended to implement CSRF protection measures such as using unique, unpredictable tokens for each form submission and validating the token on the server side to ensure the request is from an authorised user.

For www.manipalhospitalsglobal.com, Using nmap nse, we found Apache is vulnerable to two vulnerabilities that will impact the web server if the attacker exploits it. Manipal Hospital's website is vulnerable to cross-site request forgery in several forms on the website. CSRF vulnerabilities can allow an attacker to manipulate a user's session and perform actions on their behalf without their consent. To mitigate CSRF vulnerabilities, website developers can implement the following measures. Use CSRF Tokens to generate a unique token for each user session and include it as a hidden field in forms. Verification that the token matches the one associated with the user's session upon form submission to prevent attackers from forging requests because they will not have the correct token.

Interestingly, we found a potential admin folder on a website. "admin" folder or directory is commonly used to store files or resources related to website administration or management. It may contain sensitive files, administrative tools, or restricted areas that are not accessible to regular website visitors. An admin folder's exact content and functionality can vary depending on the website or application it belongs to. To mitigate this, implement robust authentication mechanisms for accessing the admin folder, including secure username/password combinations, multi-factor authentication (MFA), or IP-based restrictions. Ensure that only authorized individuals or roles have access to the admin folder.

Next discussion will focus on the CVE found on these websites. The CVE-2022-27522 was found on www.sprinkle.online effect Mod_proxy_uwsgi module of Apache

HTTP Server versions 2.4.30 through 2.4.55. A vulnerability in HTTP Response Smuggling makes it possible to manipulate response headers and potentially include harmful code. This vulnerability could lead to data theft, code injection, and cross-site scripting attacks. Upgrade Apache HTTP Server or turn off mod_proxy_uwsgi if not in use to reduce risk. This configuration may be accomplished by deleting or editing the "LoadModule proxy_uwsgi_module" line in the Apache configuration file (Huang et al. 2022).

The CVE-2022-37436 vulnerability was found on www.sprink.online, www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions prior to 2.4.55. Due to this flaw, a malicious backend can prematurely truncate response headers, including portions of the response body. The client may not understand necessary security-related headers appropriately, which might lead to vulnerabilities. It is advised to update Apache HTTP Server to version 2.4.55 to address this issue because it has a fix for CVE-2022-37436. Utilizing a reverse proxy or load balancer in front of the Apache HTTP Server is another precaution that can be taken to sanitize headers and stop any malicious headers from making it to the server (Huang et al. 2022).

The CVE-2022-36760 vulnerability was found on www.sprink.online, www.sifytechnologies.com, www.i2ifunding.com which effects Apache HTTP Server's mod_proxy_ajp contains a vulnerability called "Inconsistent Interpretation of HTTP Requests" or "HTTP Request Smuggling." An attacker can sneakily send requests to the AJP server that the Apache HTTP Server relays. It happens because the two servers' interpretations of HTTP requests differ, which makes it possible for unauthorized requests to be executed. The most recent version of Apache HTTP Server (2.4.55), which includes a patch for CVE-2022-36760, is advised for mitigation of this vulnerability. If mod_proxy_ajp is not utilized in the Apache setup, the probability of exploitation can be minimized by either retaining the "LoadModule proxy_ajp_module" line in the configuration file without commenting it out or by remove it (Huang et al. 2022).

The CVE-2021-44790 vulnerability was found on www.i2ifunding.com and www.busindia.com. which related to Apache HTTP Server contained a buffer overflow vulnerability. Lua scripts can cause a buffer overflow in mod_lua's multipart parser. It is crucial to fix this issue despite no known exploits. Apache HTTP Server can be mitigated against CVE-2021-44790 by following these steps. Upgrade to version 2.4.52: A patch is included in that version. Update to this version or the following secure release to resolve the problem. Mod_lua can be turned off to prevent the module from being exploited. To prevent buffer overflow, use input validation techniques to sanitize request bodies before processing (Butt et al. 2022).

The CVE-2022-31813 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.53 and earlier. In the client-side Connection header hop-by-hop method, the server fails to deliver the X-Forwarded-* headers to the origin server. This vulnerability can be exploited to avoid IP-based authentication. This vulnerability can be mitigated by taking the following steps. The problem has been fixed in Apache HTTP Server version 2.4.54; therefore, upgrade the server. By placing the server behind a reverse proxy and making sure the X-Forwarded-* headers are kept up to date attack can be prevented (Kaur et al. 2023).

The CVE-2022-23943 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects the mod_sed module of the Apache HTTP Server due to an out-of-bounds writing flaw. An attacker can corrupt heap memory with potentially harmful data. Apache HTTP Server versions 2.4, 2.4.52, and prior are affected. Mod_sed modifies the server's response using preset patterns and replacement rules. Its implementation might allow an attacker to write past the allocated memory buffer. Mod_sed can be exploited by an attacker by creating carefully constructed requests. The attacker may overwrite heap memory with arbitrary data. A server may crash if this flaw is exploited, or the attacker may be able to run arbitrary code. This issue can be countered by updating Apache HTTP Server to version 2.4.53 or a later, patched version. Modifying mod_sed will fix the out-of-bounds write problem in the module, ensuring server security (Rinard et al. 2004).

The CVE-2022-22720 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.52 and earlier are vulnerable to CVE-2022-22720.HTTP Request Smuggling attacks are possible because the server does not close inbound connections when errors occur. Mitigate Apache HTTP Server's CVE-2022-22720 vulnerability by following these steps. Version 2.4.53 of Apache HTTP Server fixes the vulnerability. The vulnerability can be mitigated by upgrading Apache HTTP Server. The "strict" mode in Apache HTTP Server can be enabled using the HttpProtocolOptions directive to prevent HTTP request smuggling. This directive will reject HTTP protocol violations (Huang et al. 2022).

The CVE-2021-39275 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.48 and earlier are vulnerable to CVE-2021-39275. The ap_escape_quotes() function may write beyond the end of a buffer when given malicious input. Third-party and external modules may pass untrusted data to these functions. Consider implementing the following mitigations to mitigate CVE-2021-39275. Version 2.4.49 of Apache HTTP Server fixed the vulnerability. This vulnerability can be mitigated by upgrading Apache HTTP Server. Reduce the attack surface by turning off external or third-party modules (Rinard et al. 2004).

The CVE-2021-26691 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.0 to 2.4.46 are vulnerable to CVE-2021-26691. A specially crafted SessionHeader can cause a heap overflow in Apache HTTP Server. Implement the following mitigation to mitigate CVE-2021-26691 in Apache HTTP Server. Apache HTTP Server Version 2.4.48 fixes the vulnerability. Mitigating the vulnerability requires upgrading Apache HTTP Server. Turn off the affected module if mod_session is not used to reduce the attack surface (Huang et al. 2019).

The CVE-2021-33193 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects the HTTP/2 message processing process does not validate input correctly. As a result, CVE-2021-33193

allows HTTP request-splitting attacks. An attacker can exploit this vulnerability by injecting arbitrary HTTP requests, resulting in web cache poisoning or cross-site scripting attacks can occur. Upgrading Apache HTTP Server to version 2.4.49 or later can mitigate this vulnerability. It turns off HTTP/2 support in their web server configuration as a temporary workaround (Chatzoglou et al. 2023).

The CVE-2021-40438 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server version 2.4.48 and earlier. When a specially crafted request URI path is sent, mod_proxy forwards the request to the remote user's selected origin server, enabling attackers to circumvent access restrictions. Upgrade Apache HTTP Server to version 2.4.49 or later to mitigate this vulnerability. Users can also implement access controls to restrict sensitive access (Rinard et al. 2004).

The CVE-2020-35452 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.0 to 2.4.46. A specially crafted Digest nonce can cause a stack overflow in mod_auth_digest. Due to the overflow's size and value, some compilers and compilation options might make it exploitable. Upgrade to the latest version of Apache HTTP Server, which fixes this vulnerability. Alternatively, users can turn off mod_auth_digest until the upgrade is complete (Butt et al. 2022).

The CVE-2022-28615 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which can crash the server. Apache HTTP Server version 2.4.53 is vulnerable to this vulnerability. This vulnerability occurs when a large input buffer is provided to the ap_strcmp_match() function, causing a read-beyond-bounds error. Despite being unlikely, third-party modules and Lua scripts that use ap_strcmp_match() may be vulnerable. Patches and updates should be applied to Apache HTTP Server versions 2.4.53 and earlier as soon as possible. Also, ap_strcmp_match() may be used by unnecessary third-party modules and Lua scripts. Reducing the attack surface reduces the likelihood of an adversary exploiting this vulnerability (Butt et al. 2022).

The CVE-2022-22721 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.52 and earlier are vulnerable to this vulnerability. An integer overflow occurs when the server allows request bodies more significant than 350MB on 32-bit systems. This vulnerability allows denial-of-service attacks or remote code execution; update the Apache HTTP Server to avoid exploitation. It is better to have the requested body size limited.LimitXMLRequestBody can be used to restrict the size of the request body. Access controls should also be implemented to restrict server access (Butt et al. 2022).

The CVE-2020-1927 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.0 to 2.4.41. An attacker can use this vulnerability to redirect requests to unexpected URLs. This vulnerability is caused by redirects configured with mod_rewrite intended to be self-referential but can be tricked by encoded newlines. Upgrade Apache HTTP Server to version 2.4.42 or later to mitigate this vulnerability. If mod_rewrite is unnecessary for the web server, it can be turned off (Chen & Freire 2021).

The CVE-2022-30556 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server version 2.4.53 and earlier. Application calling r:wsread() may receive lengths past the buffer's end due to the vulnerability. This vulnerability allows for denial-of-service attacks or remote code execution. Update the Apache HTTP Server to avoid exploitation. Upgrade to Apache HTTP Server version 2.4.54 or later to mitigate this vulnerability. Ensure that the web server restricts access to sensitive data and limits the impact of an attack (Rinard et al. 2004).

The CVE-2022-29404 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.53 and earlier. A lua script that calls r:parsebody(0) has no default limit on the input size, resulting in a denial of service (DoS) attack. A malicious request could cause the server to consume significant resources and crash by calling r:parsebody(0)

in a lua script. To mitigate this vulnerability, upgrade to Apache HTTP Server version 2.4.54 or later. In addition, we can limit the input size in the server configuration (Rinard et al. 2004).

The CVE-2022-28614 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.53 and earlier. A flaw in the ap_rwrite() function may allow an attacker to read unintended memory if the server reflects considerable input using ap_rwrite() or ap_rputs(). This vulnerability could be exploited to read sensitive information from the server's memory or cause a denial-of-service attack. To mitigate this vulnerability upgrade Apache HTTP Server version 2.4.54 or later. Compile all modules that use 'ap_rputs' and may pass a very large (INT_MAX or more significant) string to current headers compiled separately from Apache HTTP Server (Rinard et al. 2004).

The CVE-2022-26377 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.53 and earlier. 'HTTP Request Smuggling' is caused by an inconsistent interpretation of HTTP requests in the mod_proxy_ajp module. This vulnerability allows an attacker to smuggle requests to the AJP server it sends queries to by sending a carefully constructed HTTP request. Data exfiltration, unauthorized access, or other malicious actions may occur. To mitigate this vulnerability, upgrade to Apache HTTP Server version 2.4.54 or later. Mod_proxy_ajp should also be configured correctly, and only trusted sources should access the server (Huang et al. 2022).

The CVE-2022-22719 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.52 and earlier. It is caused by a flaw in how the server handles request bodies. An attacker could manipulate this weakness and force the server to read from a random memory region, possibly causing a process crash. Denial of service (DoS) attacks could impact the server's availability. To mitigate this vulnerability, upgrade to Apache HTTP Server version 2.4.53 or later (Butt et al. 2022).

The CVE-2021-34798 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which effect Apache HTTP Server versions 2.4.48 and earlier. The server's handling of malformed requests has a vulnerability. A malicious attacker could exploit this vulnerability by sending a request that may force the server to dereference a NULL pointer causing denial of service (DoS) attacks that can impact server availability. To mitigate this vulnerability, upgrade to Apache HTTP Server version 2.4.49 or later (Huang et al. 2022).

The CVE-2021-26690 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.0 to 2.4.46. A flaw in mod_session handles a specially crafted Cookie header. An attacker could exploit this vulnerability by sending a specially crafted Cookie header, which would cause a NULL pointer to dereference and crash the server. To mitigate this vulnerability, upgrade to Apache HTTP Server version 2.4.47 or later (Butt et al. 2022).

The CVE-2020-1934 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.0 to 2.4.41, specifically the mod_proxy_ftp module. When proxying to a malicious FTP server, uninitialized memory causes the vulnerability. A malicious FTP server can exploit this vulnerability by sending specially crafted responses to a mod_proxy_ftp-enabled Apache HTTP Server. Other users can acquire sensitive information from uninitialized memory. A fix for this vulnerability is included in Apache HTTP Server version 2.4.42 or later (Lu et al. 2017).

The CVE-2020-11985 vulnerability was found on www.sifytechnologies.com, and www.busindia.com which effect mod_remoteip and mod_rewrite for proxying, an attacker could spoof their IP address using CVE-2020-11985. The vulnerability affects Apache HTTP Server 2.4.0 to 2.4.41. Apache HTTP Server cannot handle requests with fake IP addresses when using mod_remoteip and mod_rewrite. As a result, the attacker's IP address could be recorded or used in PHP scripts, resulting in additional attacks. To mitigate this vulnerability upgrade Apache HTTP Server to version 2.4.43 or later (Rinard et al. 2004).

The CVE-2023-25690 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects when mod_proxy is enabled with a RewriteRule or ProxyPassMatch that matches a portion of the user-supplied request-target data and inserts it into the promised request-target using variable substitution, CVE-2023-25690 affects Apache HTTP Server versions 2.4.0 through 2.4.55. HTTP Request Smuggling attacks can exploit this vulnerability to bypass proxy server access controls, unintended proxy URLs to existing origin servers, and poison caches. Update Apache HTTP Server to 2.4.56 to mitigate this vulnerability. RewriteRule and ProxyPassMatch patterns matching user-supplied request-target data should not be reinserted into the proxied request-target using variable substitution. Users should instead use a specific way that matches only the desired URL and not include user-supplied data. A web application firewall (WAF) is also recommended to detect and block malicious HTTP Request Smuggling attacks. Users should also follow web security best practices, such as input validation and sanitization, to prevent harmful data from being injected (Huang et al. 2022).

The CVE-2022-37436 vulnerability was found on www.sifytechnologies.com, www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions before 2.4.55. Malicious backends can prematurely truncate response headers and bodies. Clients may not understand security-related titles appropriately, leading to vulnerabilities. CVE-2022-37436 has been fixed in Apache HTTP Server version 2.4.55. Another precaution is using a reverse proxy or load balancer before the Apache HTTP Server to sanitize headers (Huang et al. 2022).

The "Inconsistent Interpretation of HTTP Requests" or "HTTP Request Smuggling" in mod_proxy_ajp of the Apache HTTP Server allows an attacker to send covert requests to the AJP server. Unauthorized requests can be executed because the two servers interpret HTTP requests differently. This vulnerability can be mitigated by updating Apache HTTP Server to version 2.4.55, incorporating the CVE-2022-36760 patch. We can reduce the risk of exploitation by leaving the "LoadModule proxy_ajp_module" line uncommented or removing it from the configuration file (Huang et al. 2022).

The CVE-2023-28625 vulnerability was found on www.medtravels.in which impacts the mod_auth_openidc module used in Apache 2.x for authentication and authorisation. Versions ranging from 2.0.0 to 2.4.13.1 are impacted. The vulnerability arises when OIDCStripCookies is enabled, and a specifically constructed cookie is given. NULL pointer dereferences cause segmentation faults. This vulnerability allows Denial-of-Service (DoS) attacks. To prevent upgrade mod_auth_openidc to version 2.4.13.2. The patch in this version fixes the problem and avoids NULL pointer dereferences. Until the upgrade is performed, avoid using the OIDCStripCookies directive. The vulnerability can be mitigated by turning off this directive (Butt et al. 2022).

The CVE-2021-44224 vulnerability was found on www.i2ifunding.com and www.busindia.com which affects Apache HTTP Server versions 2.4.7 through 2.4.51 whereby a formulated URI sent to httpd setup as a forward proxy (ProxyRequests on) can cause a crash (NULL pointer dereference) or permit requests to be redirected to a declared Unix Domain Socket endpoint (Server-Side Request Forgery) in configurations that mix forward and reverse proxy declarations. It is advised to update Apache HTTP Server to version 2.4.52 or later to mitigate this vulnerability, as this release offers a remedy for the problem. Turning off the "ProxyRequests" directive is also advised if it is not required for the server's functionality (Huang et al. 2022).

The CVE-2021-36160 vulnerability was found on www.busindia.com which affects Apache HTTP Server versions 2.4.30 to 2.4.48, inclusive A carefully crafted request URI-path can trigger a buffer overread in the mod_proxy_uwsgi module of the Apache HTTP Server. Buffer overread can cause the server reading memory beyond its allocated boundaries, potentially resulting in a crash or denial-of-service (DoS) condition. To migtigate this issue, it's recommended to upgrade your Apache HTTP Server installation to a version that includes a fix for CVE-2021-36160 (Butt et al. 2022).

4.3.3 Discussion of Qualys Result

Qualys scan was carried out on selected Indian websites that provide very alarming result. Most of the web site is using obsolete technologies that prone to cyber-attack.

Qualys scan of sunrise university revealed two problems with key exchange and product support. The first problem is that Sunrise University still uses TLS 1.0 and TLS 1.1, two outdated and unsafe versions of the TLS protocol. Turning off these versions and switching to TLS 1.2 or later is advised for greater security. If any intermediate certificate in the chain is missing, the chain of trust is broken, and the SSL/TLS certificate cannot be verified, resulting in a security warning or error message displayed to the user, indicating that the connection may not be secure. To mitigate we must contact the organisation responsible for managing the SSL/TLS certificate and request that they provide the certificate(s) lacking from the certificate chain. Install the missing intermediate certificates on the server once they have been obtained to conclude the certificate chain (Moriarty & Farrell 2021).

Based on Qualys scanning for www.amity.com, we found two issues related to product support; using the RC4 cipher is known to be weak and vulnerable to cryptographic attacks. Therefore, turning off its usage on web servers is generally recommended to ensure secure communications. However, in this case, it was found that the server does accept RC4 cipher, but only with older protocols.RC4 is an encryption algorithm to secure network communications by encrypting data transmitted across the network. However, it has known vulnerabilities that attackers can exploit to intercept and decrypt network traffic.

As a result, modern networks rejected using the RC4 cipher and stopped supporting it altogether. While servers can still accept RC4 ciphers, it is only an older protocol that may not be supported or outdated by modern web browsers, which means that if an attacker can bypass network traffic, he may be able to use known vulnerabilities in RC4 ciphers in used to decrypt encrypted data to access sensitive information. Ensure only to use solid and secure encryption protocols to prevent attackers from exploiting known vulnerabilities in older encryption algorithms that do not use and do not get important information. In addition, it is recommended that regularly monitor server configuration and update with the latest security patches and updates to eliminate potential security vulnerabilities Concerned that the www.amity.edu server supports older protocols such as TLS 1.0 and TLS 1.1 only, like these protocols are now considered insecure due to known vulnerabilities with this old

protocol, RC4 accepts ciphers, which are again considered insecure (Moriarty & Farrell 2021).

A grade cap of B indicates that the server lacks certain security features and does not follow current best practices. It is recommended to update the server to the most recent version of the TLS protocol, such as TLS 1.3, and not use stronger ciphers that are more resistant to attack, such as AES (Moriarty & Farrell 2021).

Based on Qualys scanning for i2i funding, we found two issues related to key exchange and product support: key exchange for i2i funding support supports TLS 1.0 and TLS 1.1. TTLS 1.0 and TLS 1.1 contain known flaws and are no longer considered secure. TLS 1.2 or subsequent versions are suggested to improve security, and TLS 1.0 and TLS 1.1 is disabled. For product support, we found that i2i funding does not support forward secrecy with the reference browsers. Forward secrecy is a property of specific cryptographic protocols that ensures that past communications remain secure even if a secret key is compromised in the future. It is an important security feature for protecting the confidentiality of data in transit. We must allow reference browsers to support forward secrecy to mitigate this risk. Due to these two risks, the rating capped to a B indicates the possibility that the attack could exploit these weaknesses (Moriarty & Farrell 2021).

Based on the Qualys scanning conducted for www.busindia.com, it was observed that there exists a particular issue with product support. Specifically, it was revealed that Bus India does not provide support for forward secrecy in reference browsers. Forward secrecy is a property of specific cryptographic protocols that ensures that past communications remain secure even if a secret key is compromised in the future. It is an important security feature for protecting the confidentiality of data in transit. To address this risk, reference browsers must be permitted to adopt forward secrecy measures (Ge et al. 2022). Due to these risks, the rating capped to a B indicates that the attack could exploit these weaknesses.

The rest of Indian websites rated as “A”. Having an A+ rating from Qualys SSL Labs is a positive indicator of the security of www.sifytechnologies.com. The website

owner has taken sufficient precautions to safeguard communications and protect the data of its users from interception, manipulation, and other attacks. However, it is essential to realise that SSL/TLS setup is only one part of website security; other variables, such as safe coding practises, access limits, and monitoring and response, may all contribute to a website's overall security posture.

Having an A rating from Qualys SSL Labs is a positive indicator of the security of www.medtravels.com. The website has taken appropriate measures to safeguard communications and protect the data of its users from interception, manipulation, and other attacks. However, it is essential to realise that SSL/TLS setup is only one part of website security; other variables, such as safe coding practises, access limits, and monitoring and response, may all contribute to a website's overall security posture.

Having an A rating from Qualys SSL Labs is a positive indicator of the security of www.manipalhospitalsgobal.com The website has taken adequate steps to secure communications and protect its users' data from interception, tampering, and other attacks.

4.3.4 Discussion of SQLmap Result

Based on SQLmap result out 8 selected Indian websites only one website is vulnerable to SQL injection which is www.sunriseuniversity.in whereby the injection point is in the 'fn' parameter passed through a GET request. Two types of injection points have been identified: boolean-based blind and time-based blind. The Boolean-based blind injection point can infer information about the database by constructing SQL statements that return true or false based on the injected payload. In this case, the payload is 'admission-procedure' AND 7346=7346 AND 'kZpL'='kZpL'. This payload will always return true, but an attacker can modify the payload to extract sensitive information from the database. The time-based blind injection point can delay the database response and infer information about the database based on the delay. The payload, in this case, is 'admission-procedure' AND (SELECT 2563 FROM (SELECT(SLEEP(5)))HaIi) AND 'VtWo'='VtWo'. This payload will cause the database to sleep for 5 seconds before responding, indicating that the injected payload is valid. To prevent SQL injection attacks, we should validate and sanitise all user input to mitigate these vulnerabilities.

In addition to using prepared statements and parameterised queries, we can ensure that user input is correctly sanitised prior to being sent to the database by sanitising it before sending it. Limiting database privileges and monitoring the system for suspicious activities is also recommended.

4.3.5 Discussion of Online OpenVAS Result

Using the Online OpenVas tool to assess the security of the selected Indian websites and we found many vulnerabilities and most of the vulnerability is the same across all websites this raised concerned potential cyber-attack against these websites. here is breakdown of number of vulnerabilities for each website. www.sprink.online and we discovered fourteen vulnerabilities, www.sifytechnologies.com and we discovered fourteen vulnerabilities, www.sunriseuniversity.com we discovered twelve vulnerabilities, www.amity.edu we discovered three vulnerabilities, www.medtravels.in we discovered fourteen vulnerabilities, www.i2ifunding.com we discovered ten vulnerabilities, www.busindia.com we discovered twelve vulnerabilities, www.manipalhospitalsgobal.com we discovered twelve vulnerabilities.

Here is list of vulnerabilities discover in detail. Cross-Domain Misconfiguration vulnerabilities was found on www.sprink.online, www.sifytechnologies.com, www.sunriseuniversity.in, www.medtravels.in, www.i2ifunding.com, www.busindia.com www.manipalhospitalsglobal.com which permits an attacker to access data from a different domain, potentially exposing sensitive data. The impact and risk depend on the specific misconfiguration, which is generally considered high. The risk can be mitigated by adequately configuring cross-domain policies to prevent unauthorized access (Helmiawan et al. 2020).

Absence of Anti-CSRF Tokens vulnerabilities was found on www.sprink.online,www.sifytechnologies.com,www.sunriseuniversity.in,www.medtravels.in,www.i2ifunding.com, www.manipalhospitalsglobal.com whereby Web applications are vulnerable to CSRF attacks due to unavailable anti-CSRF tokens to prevent the use of anti-CSRF tokens. Tokens associated with form submissions are unique to each user session. The server validates the token to ensure the request's validity. Without Anti-CSRF tokens, attackers can send fake forms the server accepts,

allowing them to operate without the user's permission. Web developers should utilize Anti-CSRF tokens to address this issue by creating distinct tokens for each user session and server-side verifying them—regular security audits, secure coding procedures, and applying security updates (Wardana et al. 2022).

Content Security Policy (CSP) Header Not Set vulnerability was found on www.sprink.online,www.sunriseuniversity.in,www.medtravels.in,www.i2ifunding.com,www.busindia.com,www.manipalhospitalsglobal.com whereby websites are vulnerable to Cross-Site Scripting (XSS) attacks in the absence of a Content Security Policy (CSP) header. Website owners can define trustworthy content sources and prohibit material from untrusted sources using the security feature known as CSP. Attackers can insert harmful code into a website without a CSP header, and the browser will run that code without any limitations resulting in session modification and unauthorized access to user data. Website owners should add a CSP header to their HTTP response that specifies trustworthy content sources and criteria for content evaluation to close this issue. It is crucial to thoroughly evaluate the CSP policy to prevent legal content from being blocked. Website operators should also follow other security best practices, such as input validation, output escaping, and updating software with security patches. Website owners may guard against XSS attacks and guarantee security by installing CSP and following best practices (Roth et al. 2020).

Missing Anti-clickjacking Header vulnerability was found on www.sprink.online,www.sifytechnologies.com,www.sunriseuniversity.in,www.medtravels.in,www.i2ifunding.com,www.busindia.com whereby website visitors are persuaded to take activities without their knowledge. Website owners can include an anti-clickjacking header, X-Frame-Options, in their HTTP response to stop clickjacking attacks. This header indicates whether an iframe or frame should be used to display the webpage. To successfully stop clickjacking attacks, change the header value to "DENY" so the browser will not allow the website to be shown within a frame or iframe (Mu’min et al. 2022).

Vulnerable JS Library vulnerability was found on www.sprink.online, www.sifytechnologies.com,www.sunriseuniversity.in,www.medtravels.in,www.i2ifun

ding.com,www.manipalhospitalsglobal.com whereby using vulnerable JavaScript libraries can put websites at risk for security breaches, giving attackers access to user data or the ability to run harmful code. Website owners should often upgrade their JavaScript libraries to the most recent versions since updates frequently include security patches to minimize this issue. Additionally, they should keep up with any known vulnerabilities in their libraries and take the necessary precautions to mitigate them. Another practical step is to implement a material Security Policy (CSP), which forbids the execution of material from unauthorized sources, preventing the execution of malicious code inserted. Website owners may improve the security of their websites and shield visitors from potential threats by upgrading JavaScript libraries and putting a CSP in place (Mu'min et al. 2022).

Cross-Site Request Forgery (CSRF) vulnerability was found on www.sprink.online whereby there are no anti-CSRF tokens, web apps are susceptible to CSRF attacks. To prevent CSRF attacks, in which a user can be deceived into sending a dangerous request without realizing it, prevent the use of anti-CSRF tokens. Each user session is specific to these tokens, a component of form submissions. The server validates the token to ensure the legitimacy of the request. Without Anti-CSRF tokens, attackers can send fake forms the server accepts, allowing them to operate without the user's permission. Web developers should utilize Anti-CSRF tokens to address this issue by creating distinct tokens for each user session and server-side verifying themregular security audits, secure coding procedures, and applying security updates (Rawat et al. 2020).

Missing 'Secure' Cookie Attribute (HTTP) vulnerability was found on www.sprink.online, www.sunriseuniversity.in. An essential indication for cookies is the 'Secure' property, which guarantees that they may only be transferred through encrypted HTTPS connections. Cookies can be transferred across unencrypted HTTP connections when the 'Secure' property is absent, rendering them susceptible to interception by attackers. Attackers might utilize intercepted cookies to obtain unauthorized access to user accounts, and this is especially risky for cookies storing sensitive data like login passwords or session tokens. To reduce this issue, website owners should ensure that cookies containing sensitive information have the 'Secure'

property set, enabling transfer only via HTTPS connections. Cross-site scripting (XSS) attacks that steal cookies can be prevented by making cookies 'HttpOnly,' which makes cookies inaccessible to client-side scripts. The 'Strict-Transport-Security' (HSTS) header can further improve security by mandating HTTPS connections and forbidding HTTP downgrades (Mu'min et al. 2022).

Cookie Without Secure Flag vulnerability was found on www.sprink.online,www.sunriseuniversity.in,www.medtravels.in,www.manipalhospitalsglobal.com whereby HTTP cookies are short data packets a website sends to a user's browser and stores there. They could include private data like login passwords or session tokens. The ' Secure ' property is essential for cookies because it specifies that cookies should only be transferred via encrypted HTTPS connections. Attackers can intercept and read cookies that do not have the "Secure" tag and are sent via unencrypted HTTP connections, which might result in session hijacking and data theft. Website owners must ensure that cookies containing sensitive information have the 'Secure' attribute set to reduce the danger of cookie hijacking, ensuring they are only sent via encrypted HTTPS connections, making it far more difficult for hackers to intercept them. Another suggested security step is using the 'HttpOnly' property, which prohibits client-side scripts from accessing cookies and lowers the risk of cross-site scripting (XSS) attacks (Mu'min et al. 2022).

Cross-Domain JavaScript Source File Inclusion vulnerability was found on www.sprink.online,www.sunriseuniversity.in,www.medtravels.in,www.i2ifunding.com,www.busindia.com,www.manipalhospitalsglobal.com whereby allows an attacker to insert malicious JavaScript code into a website by adding a script file from another domain. An attacker can host a malicious file on a separate domain and fool the victim website into including it when a website fails to verify the source of the JavaScript code it includes. The malicious file can run arbitrary code on the user's browser once included, which might result in data theft or other harmful actions. Website owners should apply many security precautions to prevent this issue. They should ensure to only include JavaScript files from trustworthy domains by checking the provenance of every file they include on their website. It is also crucial to use CSP headers (Alfarizi & Ashari 2022).

Website owners can prevent scripts from loading from unreliable sources and mitigate XSS attacks by specifying which content sources can load on their websites using CSP headers. Also, consider adding Subresource Integrity (SRI) as a security precaution. By embedding a cryptographic hash of the script file in the HTML source code, SRI enables the validation of the consistency of included script files. The browser will refuse to load the file if the hash does not match the anticipated value, providing security against tampering (Alfarizi & Ashari 2022).

Strict-Transport-Security Header Not Set vulnerability was found on www.sprink.online,www.medtravels.in,www.i2ifunding.com,www.manipalhospitalsglobal.com whereby browsers can only view a website through a secure HTTPS connection due to the Strict-Transport-Security (STS) header. Without this header, attackers can force users onto an unsafe HTTP connection and intercept their online activity. Website owners should set the STS header in their web server's configuration file to shield visitors against such assaults. This configuration must be included in the header: *Strict-Transport-Security: max-age=31536000; includeSubDomains.*

This directive extends the policy to all subdomains and directs browsers to only visit the website through HTTPS for one year. Website owners must set up their web servers correctly and offer HTTPS for all resources and web pages. Security is improved since users or attackers cannot change the STS header after it has been established (Siewert et al. 2022).

Server Leaks Version Information via "Server" HTTP Response Header Field vulnerability was found on www.sprink.online, www.sunriseuniversity.in, www.medtravels.in,www.i2ifunding.com,www.busindia.com,www.manipalhospitalsglobal.com whereby when a response is provided from the server to a client, the "Server" HTTP response header field contains details about the server software and its version. However, sharing this information runs the danger of assisting attackers in locating software weaknesses in the server. It is advised to conceal the server version information in the "Server" header field to reduce this vulnerability. Web server administrators can fix this problem by deleting or changing the "Server" header in the

server configuration file. Setting the "ServerTokens" directive in the configuration file to a value like "Prod" is one method.

The server will no longer send the "Server" header field's exact version information; instead, a generic value, such as "Apache" or "Microsoft-IIS," will be sent in its place. Administrators can also set their web application firewall (WAF) to prohibit requests that contain server-version data. By taking this extra precaution, attackers are deterred from using the server version information as a weapon against the server (Gadient et al. 2021).

Server Leaks Information via "X-Powered-By" HTTP Response vulnerability was found on www.sprink.online, www.sunriseuniversity.in, www.i2ifunding.com, www.busindia.com whereby an HTTP response header called "X-Powered-By" frequently identifies the platform that powers a website or web application. However, revealing this information might be dangerous since it gives potential attackers vital information, such as the precise server-side language or framework used and version specifics. Attackers might use this information to find weaknesses and conduct focused strikes. To lower the risk, we restrict the information by removing the "X-Powered-By" header in Apache requires adding the following line to the configuration file: *Unset header X-Powered-By.* These actions improve security by reducing the information attackers may access through the "X-Powered-By" header (Gadient et al. 2021).

Cookie with SameSite Attribute None vulnerability was found on www.sprink.online, www.sunriseuniversity.in, www.medtravels.in. Cookies are small pieces of data stored on a user's computer by websites; they aid in keeping user sessions active and saving preferences. However, cookies can be vulnerable to attacks like CSRF and XSS if not adequately safeguarded. The SameSite property can be used as a means of enhancing cookie security. The options are Strict, Lax, or None. Strict guarantees that cookies are only transmitted to the website that initially placed them and only in a first-party context. Like Strict, Lax permits cookies to be transmitted when a user clicks on a link from an outside website. None make cookies susceptible to CSRF attacks by not allowing them to be transmitted in all situations, including cross-site requests. In CSRF attacks, the attacker takes advantage of the cookie that the browser, by default,

includes sending requests to a vulnerable website. By doing so, the assailant can act in the victim's place and engage in unauthorized activity. To safeguard against CSRF attacks, setting the SameSite attribute to Strict or Lax is advisable. Setting the SameSite property to Strict or Lax is advised to protect against CSRF attacks. The maximum level of security is offered by strict. However, it may disrupt some cross-site request-dependent website functionality. Lax is suited for most websites since it balances security and functionality. Selecting the right SameSite setting improves cookie security and reduces CSRF threats (Compagna et al. 2021).

For www.sifytechnologies.com, SQLmap and online OpenVAS was executed, and we had divergent results from these tools can be attributed to a combination of factors inherent to their methodologies and detection techniques. SQLmap, primarily designed to detect SQL injection vulnerabilities, operates by automating the injection of malicious SQL queries into input fields. SQLmap is proficient in identifying standard SQL injection vulnerabilities, it might overlook more complex or unconventional injection vectors.

Online OpenVAS adopts a broader approach as a comprehensive vulnerability scanner. Its algorithms encompass a spectrum of detection techniques, such as known vulnerability databases, heuristic analysis, and anomaly detection. OpenVAS's capacity to recognize various vulnerability types, including but not limited to SQL injection, contributes to its potential to uncover a wider range of security flaws. The presence of SQL injection vulnerabilities detected exclusively by online OpenVAS, such as time-based attacks on different database management systems, highlights the tool's ability to probe deeper into potential weaknesses. Time-based SQL injections, by nature, introduce subtler and more intricate attack vectors that may evade conventional automated scanners like SQLmap. Below is SQL injection vulnerabilities found.

1. SQL Injection – MySQL; this vulnerability allows attackers to execute arbitrary SQL queries against a MySQL database. To mitigate this, we ensure sanitized input data and use prepared statements to prevent SQL injection attacks (Yunus et al. 2018).

2. SQL Injection - Hypersonic SQL - Time-Based; This type of SQL injection relies on time delays to extract information from a database. To mitigate we use parameterized queries and avoid using dynamic SQL queries (Yunus et al. 2018).

3. SQL Injection - Oracle - Time-Based; this vulnerability can be exploited to execute arbitrary SQL queries against an Oracle database using time delays. To mitigate we use prepared statements and avoid using dynamic SQL queries (Yunus et al. 2018).

4. SQL Injection - PostgreSQL - Time-Based; this is another type of SQL injection vulnerability that exploits time delays to extract information from a PostgreSQL database. To mitigate we use parameterized queries and avoid using dynamic SQL queries (Yunus et al. 2018).

5. SQL Injection – SQLite is a type of SQL injection vulnerability that allows attackers to execute arbitrary SQL queries against an SQLite database. To mitigate we use prepared statements and avoid using dynamic SQL queries (Yunus et al. 2018).

Secure Pages Include Mixed Content (Including Scripts) vulnerability was found on www.sifytechnologies.com whereby this vulnerability occurs when secure pages (using HTTPS) include insecure content, such as scripts loaded via HTTP. To mitigate we ensure all content loaded on secure pages uses HTTPS (Darwis et al. 2022).

HTTP to HTTPS Insecure Transition in Form Post vulnerability was found on www.sifytechnologies.com and www.medtravels.in whereby is a security vulnerability that occurs when a web application fails to enforce a secure transition from HTTP to HTTPS during form submission. This vulnerability refers to the interception of sensitive data by attackers and tampering with data sent from the user's browser to the server.

When the website configuration is not correctly set to use, a secure connection such as HTTPS will lead to data being transmitted over an insecure connection. Without this configuration on the web server, an attacker can intercept the user's HTTP request and modify it to use an insecure connection instead of HTTPS. To prevent ensure that

the entire form submission process occurs over a secure HTTPS. Therefore, a web server must enforce HTTPS for all form submissions and redirect HTTP requests to HTTPS.

Web developers should ensure that the form action URLs explicitly use HTTPS to prevent any insecure transition. Implement HTTP Strict Transport Security (HSTS) which instructs the user's browser to automatically upgrade all HTTP requests to HTTPS, providing an additional layer of protection against insecure transition (Darwis et al. 2022).

CSP Wildcard Directive vulnerability was found on www.sifytechnologies.com whereby this vulnerability occurs when a web application allows any domain to load content from its pages, which an attacker can use to execute unauthorized actions. We implement CSP to restrict the domains from which content can be loaded to mitigate the risk. CSP: style-src unsafe-inline; this vulnerability occurs when a web application allows unsafe inline styles to be executed, which can be used by an attacker to execute arbitrary code. To mitigate we implement CSP to prevent unsafe styles from being executed (Mu'min et al. 2022).

Multiple X-Frame-Options Header Entries vulnerability was found on www.sifytechnologies.com whereby this vulnerability occurs when multiple X-Frame-Options headers are included in a response, leading to unpredictable behaviour. To mitigate this, we only include a single X.

This vulnerability was found on www.amity.edu which consist of SSL/TLS Vulnerable Cipher Suites for HTTPS, SSL/TLS custom Weak Cipher Suites and SSL/TLS - Deprecated TLSv1.0 and TLSv1.1 Protocol purpose of cryptographic algorithms is to provide a secure connection to the Internet. Some of these cipher suites have known vulnerabilities that attackers can exploit.

SSL/TLS configuration of the web server at amity.edu uses a vulnerable cipher suite for HTTPS communication. Weak cipher suites can open the server and users to potential attacks and data breaches. It is crucial to ensure that only strong and secure

ciphers are used to prevent attackers from exploiting known vulnerabilities in older encryption algorithms to obtain sensitive information. Using more robust cipher sets can prevent possible attacks.

A better option is to use TLSv1.2 or TLSv1.3 for secure communication. If a web server still supports this earlier TLS, it may be subject to attacks like POODLE (Padding Oracle on Downgraded Legacy Encryption), which may decode SSL/TLS communication. It is recommended to disable TLSv1.0 and TLSv1.1 support on the server (Rawat et al. 2020).

Cookie No HttpOnly Flag vulnerability was found on www.medtravels.in whereby when a web application fails to set the HttpOnly flag on cookies. A cookie has been set without the HttpOnly flag, which indicates that JavaScript may access it. If a malicious script is executed on this website, the cookie becomes accessible and can be sent to another site. If this is a session cookie, session hijacking is a possibility. Make sure all cookies have the HttpOnly flag set (Mu'min et al. 2022).

X-Content-Type-Options Header Missing vulnerability was found on www.medtravels.in, www.i2ifunding.com, www.manipalhospitalsglobal.com whereby the X-Content-Type-Options header is a security measure that helps stop web browsers from mistakenly classifying files as various MIME types. Attackers may trick browsers into opening files with a different MIME type, allowing them to run malicious malware or launch other assaults. Attackers may utilise its absence to trick browsers into reading files with a different MIME type, enabling them to run malicious code or conduct other assaults.

For example, when an attacker compiles a file that appears to be a harmless image but this image is embedded with malicious code, and if the file is identified as a JavaScript file, it will be executed in the absence of the X-Content-Type-Options header is not set, The browser may read the file as a JavaScript file as the X-Content-Type-Options header is not set. To prevent this attack, all websites should be set to "no sniff" for the X-Content-Type-Options header. After this option is set, the web server will direct the browser to only interpret files following the MIME type supplied by the

server.The X-Content-Type-Options header can be modified by editing the configuration file in the web server, using a security plugin, or using middleware. The ability to specify this header is also included in several web application frameworks. By implementing the X-Content-Type-Options header with the appropriate value, website owners may improve the security of their online applications and protect against this attack (Mu'min et al. 2022).

osTicket < 1.14.3, osTicket < 1.14.8, 1.15.x < 1.15.4 and osTicket < 1.16.6, 1.17.x < 1.17.3. Multiple vulnerabilities was found on www.busindia.com related to osTicket. This vulnerability is related to cross-site scripting (XSS) vulnerability that was found in the "include/class.sla.php" file of osTicket versions prior to 1.14.2. This vulnerability enables attackers to exploit the SLA Name field, leading to XSS attacks. XSS vulnerabilities emerge when an application fails to effectively sanitize or validate user input, allowing malicious code injection into web pages. In the case of osTicket, the lack of proper input sanitization within the SLA Name field is the root cause of this vulnerability. To mitigate this vulnerability and ensure the security of osTicket installations, upgrading to version 1.17.5 or later is recommended. The developers of osTicket have released a patch that addresses this issue by implementing improved input sanitization measures (Kaur et al. 2023).

Directory Browsing - Apache 2 vulnerability was found on www.manipalhospitalsglobal.com whereby server is configured to allow directory listings. This step can expose sensitive information to attackers, such as directory structures, file names, and contents. To prevent the directory browsing vulnerability on an Apache 2 server, turn off directory browsing and review and modify the server configuration to delete the directory listing. Locate configuration files (httpd.conf or apache2.conf) and remove or comment out any lines that include "Options Indexes" or "Options +Indexes." By doing so, the server will no longer display the contents of directories when no specific file is requested. Ensure that the server is configured to serve a default document, such as index.html or index.php, when a specific file is not specified in the URL, thus preventing directory listings by presenting a default page instead (Riadi et al. 2020).

4.4 VULNERABILITY ASSESSMENT

Vulnerability assessments were done based on the outcomes of phases one and two. Each vulnerability was assessed and graded based on the Common Vulnerabilities and Exposures (CVE). This approach ensures a standardized and objective assessment of the identified vulnerabilities. By utilizing the disclosed Common Vulnerabilities and Exposures (CVEs), we can precisely classify and rank the vulnerability according to risk, consequence, and exploitability. This rating system provides a clear understanding of the potential risks posed by each vulnerability and helps guide mitigation efforts. Integrating Common Vulnerability and Exposure (CVE) ratings into the vulnerability assessment process ensures a thorough and uniform appraisal, as all known risks and potential impacts of each vulnerability are considered. This approach allows for prioritized remediation efforts and a proactive approach to managing the website's security (Nowak et al. 2023).

Table 4.4 Vulnerability assessment rating

Website	Critical	High	Medium	Low
www.sprink online	1	2	9	7
www.sifytechnologies.com	13	18	6	-
www.sunriseuniversity.com	-	1	7	5
www.amity.edu	-	4	-	-
www.medtravel.in	-	3	10	2
www.i2ifunding.com	10	10	9	4
www.busindia.com	11	15	9	3
www.manipalhospital.com	2	-	7	5

The National Vulnerability Database (NVD) and the Common Vulnerabilities and Exposures (CVE) system are two trustworthy sources from which the rating is generated. These tools offer a thorough grasp of the significance and effect of discovered vulnerabilities. By utilizing the information procured from NVD and CVE, it is imperative to guarantee the precision and reliability of the vulnerability evaluation attributed to the recognized vulnerabilities (Nowak et al. 2023). After conducting a comprehensive security assessment on selected Indian websites, we found 172 vulnerabilities existed within this website. These 172 vulnerabilities can be breakdown

further based on severity: 37 critical vulnerabilities, 53 high vulnerabilities, 57 medium vulnerabilities, and 26 low vulnerabilities.

The website www.sunriseuniversity.com has the highest number of critical severity vulnerabilities with thirteen vulnerabilities and www.busindia.com has highest amount of vulnerability which is thirty-eight. If we breakdown by sector education section has most vulnerability whereby website www.sunriseuniversity.in with thirteen vulnerability and ww.amity.edu with four vulnerabilities. For highest number of high severity vulnerabilities found on www.sifytechnologies.com which is eighteen followed by www.busindia.com with fifteen vulnerabilities. For highest number of medium severity vulnerabilities found on www.medtravel.in with ten followed by www.sprink online, www.i2ifunding.com and www.busindia.com with nine vulnerabilities each.

For highest number of low severity vulnerabilities found on www.sprink online followed by www.sunriseuniversity.com with five and www.manipalhospital.com with five. The lowest number of vulnerabilities found on www.amity.edu with four and www.sunriseuniversity.com with thirteen vulnerabilities.

Some of these websites have common vulnerabilities, which raises a significant concern as it indicates that if one website is targeted and exploited using a particular method, the same approach can also be used on other websites. This discovery emphasizes the need for immediate action to address these vulnerabilities and enhance the overall security posture of the affected websites.

An effective mitigation plan must be implemented promptly to address the critical and high vulnerabilities identified. Given the seriousness of these weaknesses, it is essential to act immediately to reduce the danger of exploitation and potential harm. By prioritising these vulnerabilities, allocating appropriate resources, and implementing efficient security measures, one can substantially diminish the potential impact on the chosen Indian website and guarantee its comprehensive resilience. The mitigation plan should include comprehensive remediation strategies, such as patching vulnerabilities, implementing secure coding practices, conducting regular security audits, and

enhancing network defences. By taking proactive measures to address critical or high vulnerabilities, it is possible to enhance the website's security posture and effectively safeguard it against any potential attacks.

In summary 172 vulnerabilities have been discovered on selected Indian websites and it is concluded that some of these vulnerabilities related to the use of outdated web servers and unpatched web servers. These outdated and unpatched systems pose a significant risk to the security and stability of the selected Indian website attacks.

4.5 SUMMARY

Using an open-source tool in this study has yielded promising results in identifying vulnerabilities in selected Indian websites. The analysis revealed that these websites are susceptible to attacks primarily caused by outdated web servers, misconfigurations, and unpatched systems. The steps for mitigating each vulnerability have been identified. Given the seriousness of these vulnerabilities, it is essential to create and implement a thorough mitigation strategy to address the risks outlined adequately.

CHAPTER V

CONCLUSION AND FUTURE WORKS

5.1 SUMMARY

This research evaluated the website security measures implemented within the Indian websites. The main aim was to perform scanning to identify vulnerabilities on an Indian website and recommend measures to mitigate the associated risks using various techniques and tools to evaluate the security posture of the websites. This research has significantly contributed to the website security assessment field.

By performing a security assessment of selected Indian websites, we found many vulnerabilities that could gravely jeopardize the security of both the websites themselves and the information they contain. In this research, we found 172 vulnerabilities. These vulnerabilities can be categorized as follows: 41 critical vulnerabilities, 58 high vulnerabilities, 68 medium vulnerabilities, and 39 low vulnerabilities. Common vulnerabilities found is SSL/TLS report Vulnerable Cipher Suites for HTTPS and SSL/TLS report Weak Cipher Suites, Cross-Domain Misconfiguration, Absence of Anti-CSRF Token, Missing Anti-clickjacking Header, Vulnerable JS Library, Cross-Domain JavaScript Source File Inclusion, HTTP to HTTPS Insecure Transition in Form Post. All this vulnerability was identified by using online OpenVAS. Based on vulnerability assessment, we have identified best way forward to address the vulnerability is my upgrading current webserver to latest version and upgrading cipher suite to use latest which is TLS 1.3 Apart from them they need make configuration changes on webserver such enable use anti-CSRF tokens, enable Anti-CSRF tokens, upgrade JavaScript libraries. and implementing cross-domain policies to restrict access.

5.2 LIMITATIONS

This research has some limitations. One of the primary limitations is that a small number of Indian websites were evaluated, which raises questions regarding the generalizability of the results to other websites. Another additional restriction is that solely automated tools were utilized for assessing the security of websites, while manual testing was not carried out, thereby limiting the number of vulnerabilities found. Finally, this security assessment was conducted without understanding how the website is set up and the technologies that are running behind the website.

5.3 FUTURE WORKS

In the future, we can build on our findings and conduct more extensive studies to assess the website security of different communities and regions. Incorporating manual testing techniques to complement the automated tools for a more comprehensive website security assessment would also be helpful. Additionally, future studies can explore emerging threats and vulnerabilities that could affect the security of websites and devise appropriate countermeasures to address them. Staying current on the latest trends and emerging threats is essential as cyber threats evolve. Future research could explore machine learning and artificial intelligence techniques to identify and mitigate emerging cyber threats in real-time.

This research provides valuable insights into the current website security posture within the Indian websites. Furthermore, it highlights the need for increased attention to cybersecurity for this website. By proactively enhancing their security posture, websites for the Indian websites can better protect their users' data and prevent cyber-attacks.

In conclusion, our research has provided valuable insights into the website security of the Indian websites. A substantial contribution has been made towards assessing website security by identifying prospective vulnerabilities and suggesting practical measures to rectify them. Nonetheless, our research is not exempt from certain constraints, providing an opportunity for future studies to enhance and combat emerging menaces to website safety.

REFERENCES

Ahmed, A.A. & Al Dabbagh, N.B. 2023. Web attacks and defenses: review paper. *Journal of Education and Science* 32(2): 114-127.

Ahmed, A.S.S., Brishty, A.A., Shachi, M., Shourav, N.S. & Sakib, N. 2021. An approach to detect cyber attack on server-side application by using data mining techniques and evolutionary algorithms. *International Journal of Applied Information Systems* 12(37): 1-9.

Aksu, M.U., Altuncu, E. & Bicakci, K. 2019. A first look at the usability of openvas vulnerability scanner. *Workshop on Usable Security (USEC)* 1-11.

Alanda, A., Satria, D., Ardhana, M.I., Dahlan, A.A. & Mooduto, H.A. 2021. Web application penetration testing using SQL Injection attack. *JOIV: International Journal on Informatics Visualization* 5(3): 320-326.

Alfarizi, M. & Ashari, I.F. 2022. Vulnerability analysis and proven on the neonime.co website using OWASP ZAP 4 and XSpear. *Jurnal Teknologi Komputer dan Sistem Informasi*, 5(2): 75-81.

Alharbi, F., Alsulami, M., Al-Solami, A., Al-Otaibi, Y., Al-Osimi, M., Al-Qanor, F. & Al-Otaibi, K. 2021. The impact of cybersecurity practices on cyberattack damage: The perspective of small enterprises in Saudi Arabia. *Sensors* 21(20): 6901.

Allodi, L. & Massacci, F. 2014. Comparing vulnerability severity and exploits using case-control studies. *ACM Transactions on Information and System Security (TISSEC)* 17(1): 1-20.

Altulaihan, E.A., Alismail, A. & Frikha, M. 2023. A Survey on Web Application Penetration Testing. *Electronics* 12(5): 1229.

Butt, M.A., Ajmal, Z., Khan, Z.I., Idrees, M. & Javed, Y. 2022. An in-depth survey of bypassing buffer overflow mitigation techniques. *Applied Sciences* 12(13): 6702.

Chatzoglou, E., Kouliaridis, V., Kambourakis, G., Karopoulos, G. & Gritzalis, S. 2023. A hands-on gaze on HTTP/3 security through the lens of HTTP/2 and a public dataset. *Computers & Security* 125: 103051.

Chawda, M. & Sharma, D.P. 2021. Deep dive into directory traversal and file inclusion attacks leads to privilege escalation. *International Journal of Scientific Research in Science, Engineering and Technology* 8(3): 115–120.

Chen, Z. & Freire, J. 2021. Discovering and measuring malicious URL redirection campaigns from fake news domains. *2021 IEEE Security and Privacy Workshops (SPW)*, hlm. 1-6.

Compagna, L., Jonker, H., Krochewski, J., Krumnow, B. & Sahin, M. 2021. A preliminary study on the adoption and effectiveness of SameSite cookies as a CSRF defence. *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, hlm. 49-59.

Cremer, F., Sheehan, B., Fortmann, M., Kia, A.N., Mullins, M., Murphy, F. & Materne, S. 2022. Cyber risk and cybersecurity: a systematic review of data availability. *The Geneva Papers on Risk and Insurance-Issues and Practice* 47(3): 698-736.

Darwis, E. & Musdar, I.A. 2022. Analisis kerentanan website renovaction menggunakan rangkaian security tools project berdasarkan framework OWASP. *Kharisma Tech* 17(1): 1-15.

Dora, J.R. & Nemoga, K. 2021. Ontology for cross-site-scripting (XSS) attack in cybersecurity. *Journal of Cybersecurity and Privacy* 1(2): 319-339.

Faircloth, J. 2016. *Penetration Tester's Open Source Toolkit*. 4th Ed. Cambridge: Elsevier.

Friedman, J. 2019. Vulnerability scoring systems, remediation strategies and taxonomies. Ph.D. Thesis, University of Pennsylvania.

Gadient, P., Nierstrasz, O. & Ghafari, M. 2021. Security header fields in http clients. *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, hlm. 93-101.

Ge, M., Kumari, S. & Chen, C.M. 2022. AuthPFS: a method to verify perfect forward secrecy in authentication protocols. Journal of Network Intelligence 7(3): 734-750.

Helmiawan, M.A., Firmansyah, E., Fadil, I., Sofivan, Y., Mahardika, F. & Guntara, A. 2020. Analysis of web security using open web application security project 10. *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, hlm. 1-5.

Huang, N., Huang, S. & Chang, C. 2019. Analysis to heap overflow exploit in Linux with symbolic execution. *IOP Conference Series: Earth and Environmental Science*, hlm. 042100.

Huang, Q.X., Chiu, M.Y., Chen, Y.F. & Sun, H.M. 2022. Attacking websites: detecting and preventing HTTP request smuggling attacks. *Security and Communication Networks* 2022: 1-14.

Jaisinghani, G. 2022. Vulnerability management in the age of containers–a review. *International Journal of Information Security* 1(01): 1-5.

Kaur, J., Garg, U. & Bathla, G. 2023. Detection of cross-site scripting (XSS) attacks using machine learning techniques: a review. *Artificial Intelligence Review* 1-45.

Kaur, R., Gabrijelčič, D. & Klobučar, T. 2023. Artificial intelligence for cybersecurity: literature review and future research directions. *Information Fusion* 97: 101804.

Kritikos, K., Magoutis, K., Papoutsakis, M. & Ioannidis, S. 2019. A survey on vulnerability assessment tools and databases for cloud-based web applications. *Array* 3: 100011.

Kumi, S., Lim, C., Lee, S.G., Oktian, Y.O. & Witanto, E.N. 2021. Automatic detection of security misconfigurations in web applications. *Proceedings of International Conference on Smart Computing and Cyber Security: Strategic Foresight, Security Challenges and Innovation (SMARTCYBER 2020)*, hlm. 91-99.

Lu, K., Walter, M.T., Pfaff, D., Nümberger, S., Lee, W. & Backes, M. 2017. Unleashing use-before-initialization vulnerabilities in the Linux kernel using targeted stack spraying. *Network and Distributed System Security (NDSS) Symposium*, hlm. 1-15.

Mambetov, S., Begimbayeva, Y., Joldasbayev, S. & Kazbekova, G. 2023. Internet threats and ways to protect against them: a brief review. *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, hlm. 195-198.

Moriarty, K. & Farrell, S. 2021. RFC 8996 deprecating TLS 1.0 and TLS 1.1. *Internet Engineering Task Force (IETF)* 1-18.

Mu'min, M.A., Fadlil, A. & Riadi, I. 2022. Analisis keamanan sistem informasi akademik menggunakan open web application security project framework. *Jurnal Media Informatika Budidarma* 6(3): 1468-1475.

Nagarjun, P.M.D. & Shaik, S.A. 2020. Cross-site scripting research: a review. *International Journal of Advanced Computer Science and Applications* 11(4): 626-632.

Najar, A.A. 2022. Covid-19 impact on cyber crimes in India: a systematic study. *2022 IEEE India Council International Subsections Conference (INDISCON)*, hlm. 1-8.

Nowak, M.R., Walkowski, M. & Sujecki, S. 2023. Support for the vulnerability management process using conversion CVSS base score 2.0 to 3.x. *Sensors* 23(4): 1802.

Odun-Ayo, I., Owoka, E., Okuoyo, O., Ogunsola, O., Ikoh, O., Adeosun, O., Etukudo, D., Robert, V. & Oyeyemi, G. 2022. Evaluating common reconnaissance tools and techniques for information gathering. *Journal of Computer Science* 18(2): 103-115.

Pohan, Y.A., Yuhandri, Y. & Sumijan, S. 2021. Meningkatkan keamanan webserver aplikasi pelaporan pajak daerah menggunakan metode penetration testing execution standar. *Jurnal Sistim Informasi dan Teknologi* 1-6.

Q-Success. 2023. Usage statistics and market share of web servers. https://w3techs.com/technologies/overview/web_server [24 January 2023].

Rawat, S., Bhatia, T. & Chopra, E. 2020. Web application vulnerability exploitation using penetration testing scripts. *International Journal of Scientific Research & Engineering Trends* 6(1): 311-317.

Riadi, I., Umar, R. & Lestari, T. 2020. Analisis kerentanan serangan cross site scripting (XSS) pada aplikasi smart payment menggunakan framework OWASP. *Jurnal Informatika Sunan Kalijaga* 5(3): 146-152.

Rinard, M., Cadar, C., Dumitran, D., Roy, D.M. & Leu, T. 2004. A dynamic technique for eliminating buffer overflow vulnerabilities (and other memory errors). *20th Annual Computer Security Applications Conference*, hlm. 82-90.

Roth, S., Barron, T., Calzavara, S., Nikiforakis, N. & Stock, B. 2020. Complex security policy? a longitudinal analysis of deployed content security policies. *Proceedings of the 27th Network and Distributed System Security Symposium*, hlm. 1-18.

Sahani, R. & Randhawa, S. 2021. Clickjacking: beware of clicking. *Wireless Personal Communications* 121(4): 2845-2855.

Sahren, S., Dalimuthe, R.A. & Amin, M. 2019. Penetration testing untuk deteksi vulnerability sistem informasi kampus. *Prosiding Seminar Nasional Riset Information Science (SENARIS)*, hlm. 994-1001.

Sarker, K.U., Yunus, F. & Deraman, A. 2023. Penetration taxonomy: a systematic review on the penetration process, framework, standards, tools, and scoring methods. *Sustainability* 15(13): 10471.

Shah, M., Ahmed, S., Saeed, K., Junaid, M. & Khan, H. 2019. Penetration testing active reconnaissance phase–optimized port scanning with nmap tool. *2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) 2019*, hlm. 1-6.

Shahid, J., Hameed, M.K., Javed, I.T., Qureshi, K.N., Ali, M. & Crespi, N. 2022. A comparative study of web application security parameters: current trends and future directions. *Applied Sciences* 12(8): 4077.

Shahriar, H. & Devendran, V.K. 2014. Classification of clickjacking attacks and detection techniques. *Information Security Journal: A Global Perspective* 23(4-6): 137-147.

Sharma, N. & Gupta, K.D. 2020. Everything on DDoS attacks, DDoS incidents & DDoS defense mechanisms!. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 11(1): 608-616.

Siewert, H., Kretschmer, M., Niemietz, M. & Somorovsky, J. 2022. On the security of parsing security-relevant HTTP headers in modern browsers. *2022 IEEE Security and Privacy Workshops (SPW)*, hlm. 342-352.

Singal, K. & Chhillar, P. 2017. Ransomware-worldwide cyber attacker. *International Journal of Advanced Research and Development* 2(5): 324-329.

Tang, P., Qiu, W., Huang, Z., Lian, H. & Liu, G. 2020. Detection of SQL injection based on artificial neural network. *Knowledge-Based Systems* 190: 105528.

Thomchick, R. & San Nicolas-Rocca, T. 2018. Application level security in a public library: a case study. *Information Technology and Libraries* 37(4): 107-118.

Wang, W., Li, Y., Wang, C., Yan, Y., Li, J. & Gu, D. 2021. Re-check your certificates! experiences and lessons learnt from real-world https certificate deployments. In Yang, M., Chen, C. & Liu, Y. (ed.). *Network and System Security*, pp. 17-37. Berlin: Springer.

Wardana, W., Almaarif, A. & Widjajarto, A. 2022. Vulnerability assessment and penetration testing on the xyz website using NIST 800-115 standard. *Jurnal Ilmiah Indonesia* 7(1): 520-529.

Wilkinson, S. 2022. UK data protection and digital information bill explained. *Journal of Data Protection & Privacy* 5(3): 242-253.

Yunus, M.A.M., Brohan, M.Z., Nawi, N.M., Surin, E.S.M., Najib, N.A.M. & Liang, C.W. 2018. Review of SQL injection: problems and prevention. *JOIV: International Journal on Informatics Visualization* 2(3-2): 215-219.

APPENDIX A

ONLINE OPENVAS RESULT

Figure A.1 Result of www.businidia.com

Figure A.2 Result of www.sunriseuniversity.in

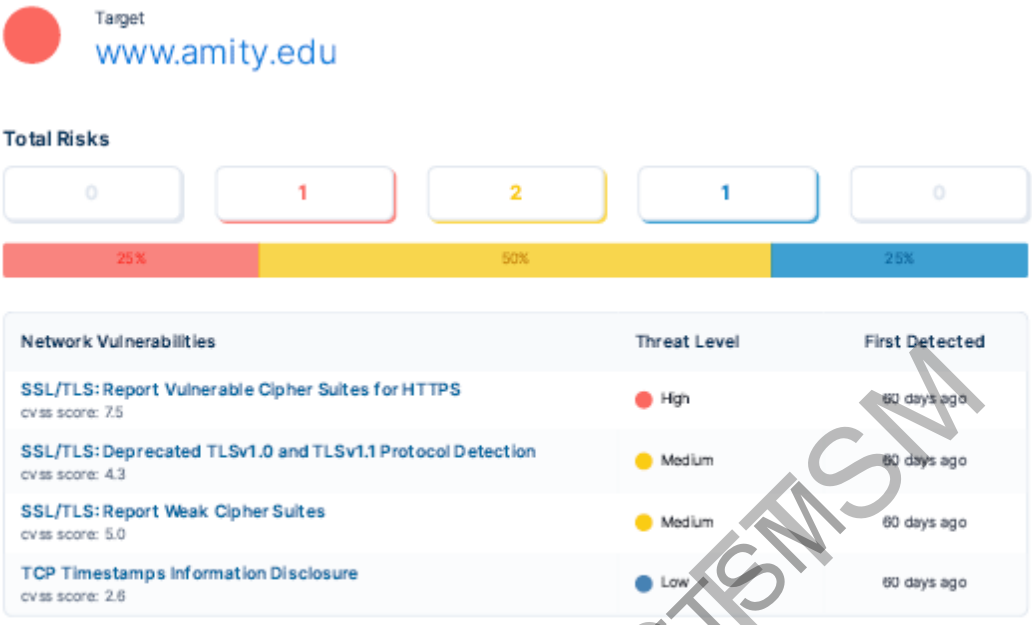


Figure A.3 Result of www.amity.edu

Figure A.4 Result of www.manipalhospitalsglobal.com

Figure A.5 Result of www.i2ifunding.com

Target
www.sifytechnologies.com

Total Risks

0	11	9	0

55% 45%

Passive Web Application Vulnerabilities	Threat Level	First Detected
Absence of Anti-CSRF Tokens	Medium	0 days ago
Cross-Domain Misconfiguration	Medium	0 days ago
Secure Pages Include Mixed Content (Including Scripts)	Medium	0 days ago
Multiple X-Frame-Options Header Entries	Medium	0 days ago
Vulnerable JS Library	Medium	0 days ago
Vulnerable JS Library	Medium	0 days ago
Missing Anti-clickjacking Header	Medium	0 days ago
CSP: Wildcard Directive	Medium	0 days ago
CSP: script-src unsafe-inline	Medium	0 days ago
CSP: style-src unsafe-inline	Medium	0 days ago
HTTPS to HTTP Insecure Transition in Form Post	Medium	0 days ago
X-Content-Type-Options Header Missing	Low	0 days ago
Cross-Domain JavaScript Source File Inclusion	Low	0 days ago
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	0 days ago
Cookie Without Secure Flag	Low	0 days ago
Cookie No HttpOnly Flag	Low	0 days ago
Cookie without SameSite Attribute	Low	0 days ago
Strict-Transport-Security Header Not Set	Low	0 days ago

Figure A.6 Result of www.sifytechnologies.com

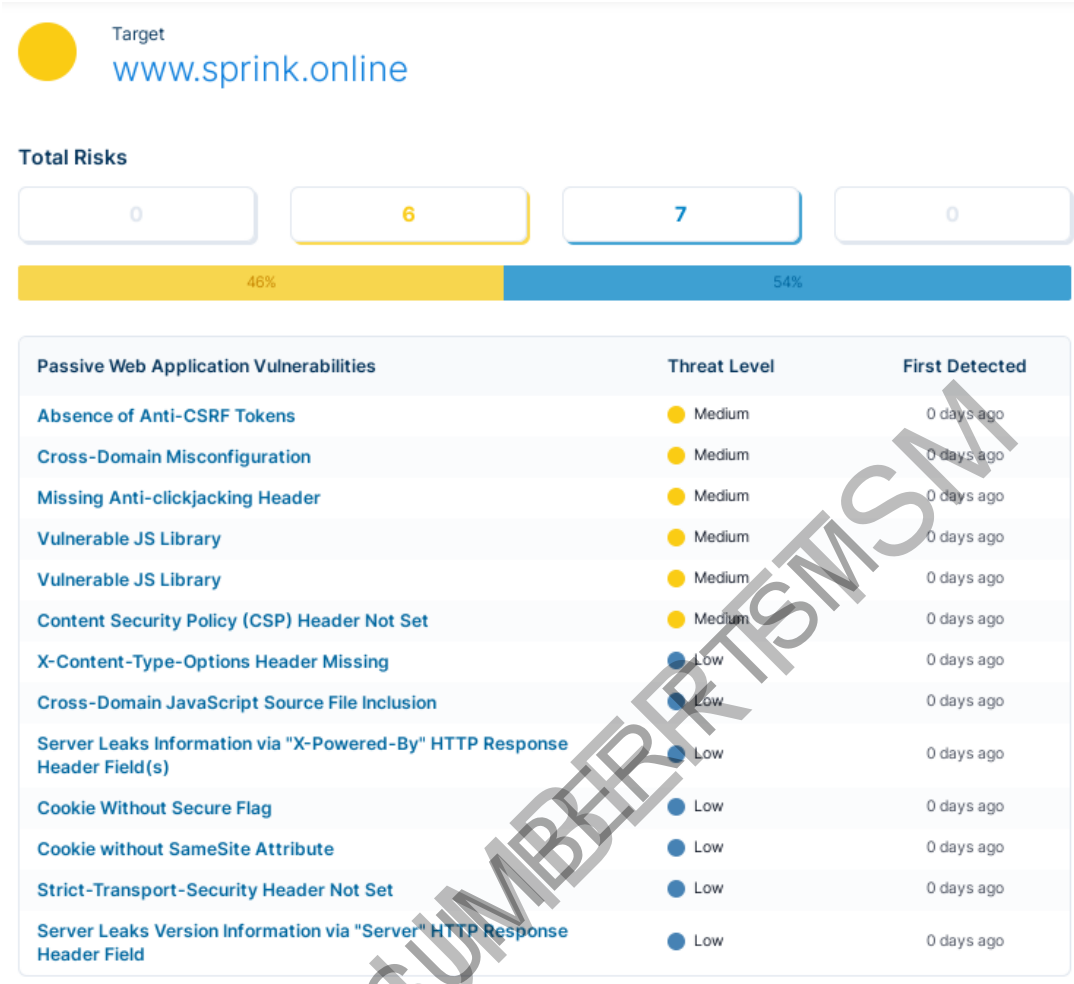


Figure A.7 Result of www.sprink.online

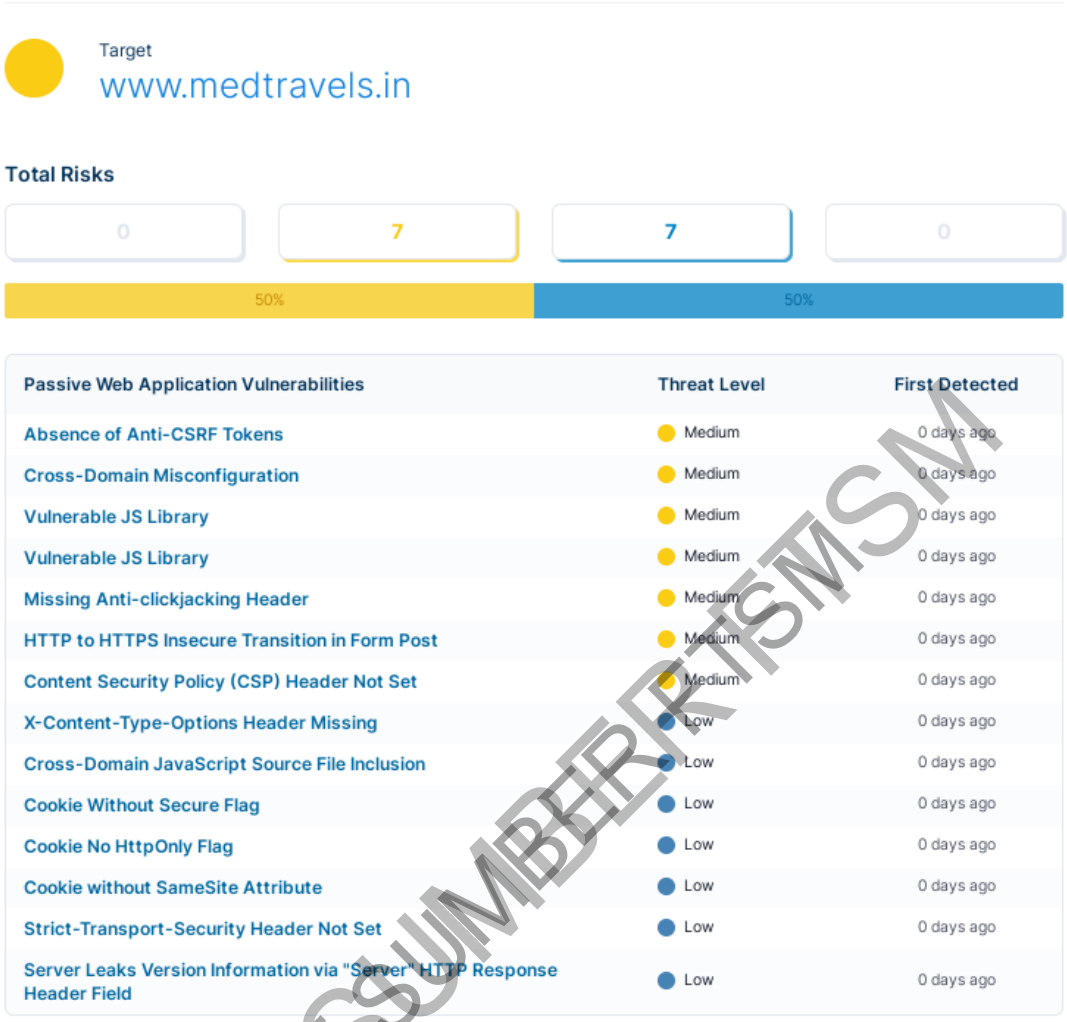


Figure A.8 Result of www.medtravels.in

APPENDIX B

NMAP RESULT

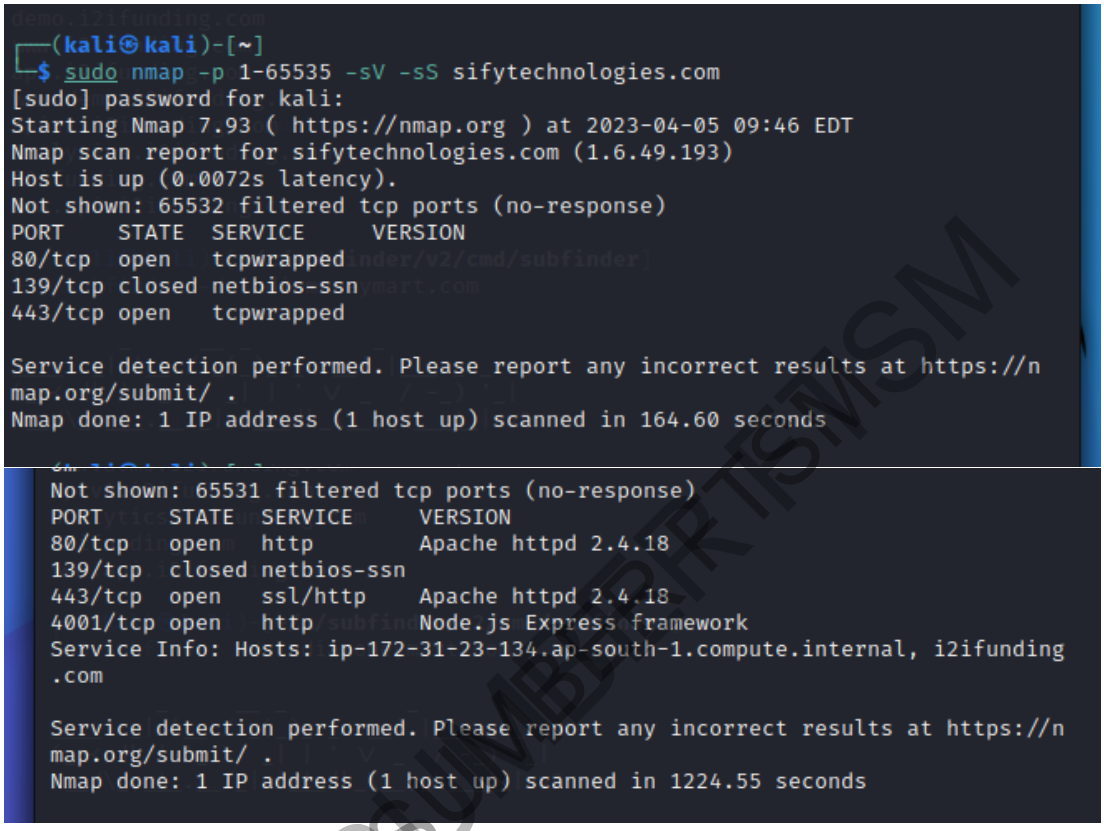


Figure B.1 Result of www.sifytechnologies.com

```
Nmap scan report for sunriseuniversity.in (103.127.31.203)
Host is up (0.00029s latency).
rDNS record for 103.127.31.203: cloud.dpmc.in
Not shown: 65377 filtered tcp ports (no-response), 134 closed tcp ports (res
t)
PORT     STATE SERVICE      VERSION
21/tcp   open  ftp          Pure-FTPd
53/tcp   open  domain       PowerDNS Authoritative Server 4.4.3
80/tcp   open  http         nginx
110/tcp  open  pop3         Dovecot pop3d
111/tcp  open  rpcbind      2-4 (RPC #100000)
143/tcp  open  imap         Dovecot imapd
443/tcp  open  ssl/http     nginx
465/tcp  open  ssl/smtp     Exim smtpd 4.96
587/tcp  open  smtp         Exim smtpd 4.96
993/tcp  open  imaps?
995/tcp  open  pop3s?
2077/tcp open  tsrmagt?
2078/tcp open  ssl/http     cPanel httpd (unauthorized)
2082/tcp open  infowave?
2083/tcp open  ssl/radsec?
2086/tcp open  gnunet?
2087/tcp open  ssl/eli?
2091/tcp open  ssl/http     cPanel httpd (unauthorized)
2095/tcp open  nbx-ser?
2096/tcp open  ssl/nbx-dir?
```

Figure B2: Result of www.sunriseuniversity.in

```
443/tcp open   ssl/http    Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1g PHP/5
.6.40 mod_perl/2.0.11 Perl/v5.26.3)
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=busindia.com
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://busindia.com:443/
|     Form id: searchform
|     Form action: /home
|
|     Path: http://busindia.com:443/Morning-Star-Travels-online-bus-booking
|     Form id: searchform
|     Form action: /home
|
|     Path: http://busindia.com:443/Roadlink-India-online-bus-booking
|     Form id: searchform
|     Form action: /home
|
|     Path: http://busindia.com:443/National-Travels1nts1-online-bus-booking
|     Form id: searchform
|_    Form action: /home
| http-server-header:
|   Apache-Coyote/1.1
```

```
009     *EXPLOIT*
|       CVE-2018-15919  5.0     https://vulners.com/cve/CVE-2018-15919
|       CVE-2021-41617  4.4     https://vulners.com/cve/CVE-2021-41617
|       CVE-2019-16905  4.4     https://vulners.com/cve/CVE-2019-16905
|       CVE-2020-14145  4.3     https://vulners.com/cve/CVE-2020-14145
|       CVE-2019-6110   4.0     https://vulners.com/cve/CVE-2019-6110
|       CVE-2019-6109   4.0     https://vulners.com/cve/CVE-2019-6109
|       CVE-2018-20685  2.6     https://vulners.com/cve/CVE-2018-20685
|_      PACKETSTORM:151227      0.0     https://vulners.com/packetstorm/PACKE
TSTORM:151227   *EXPLOIT*
80/tcp  open   http        Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1g PHP/5
.6.40 mod_perl/2.0.11 Perl/v5.26.3)
| vulners:
|   cpe:/a:apache:http_server:2.4.37:
|       CVE-2019-9517   7.8     https://vulners.com/cve/CVE-2019-9517
|       PACKETSTORM:171631      7.5     https://vulners.com/packetstorm/PACKE
TSTORM:171631   *EXPLOIT*
|       EDB-ID:51193    7.5     https://vulners.com/exploitdb/EDB-ID:51193  *
EXPLOIT*
|       CVE-2022-31813  7.5     https://vulners.com/cve/CVE-2022-31813
|       CVE-2022-23943  7.5     https://vulners.com/cve/CVE-2022-23943
|       CVE-2022-22720  7.5     https://vulners.com/cve/CVE-2022-22720
|       CVE-2021-44790  7.5     https://vulners.com/cve/CVE-2021-44790
|       CVE-2021-39275  7.5     https://vulners.com/cve/CVE-2021-39275
|       CVE-2021-26691  7.5     https://vulners.com/cve/CVE-2021-26691
```


to be continued…

…continuation

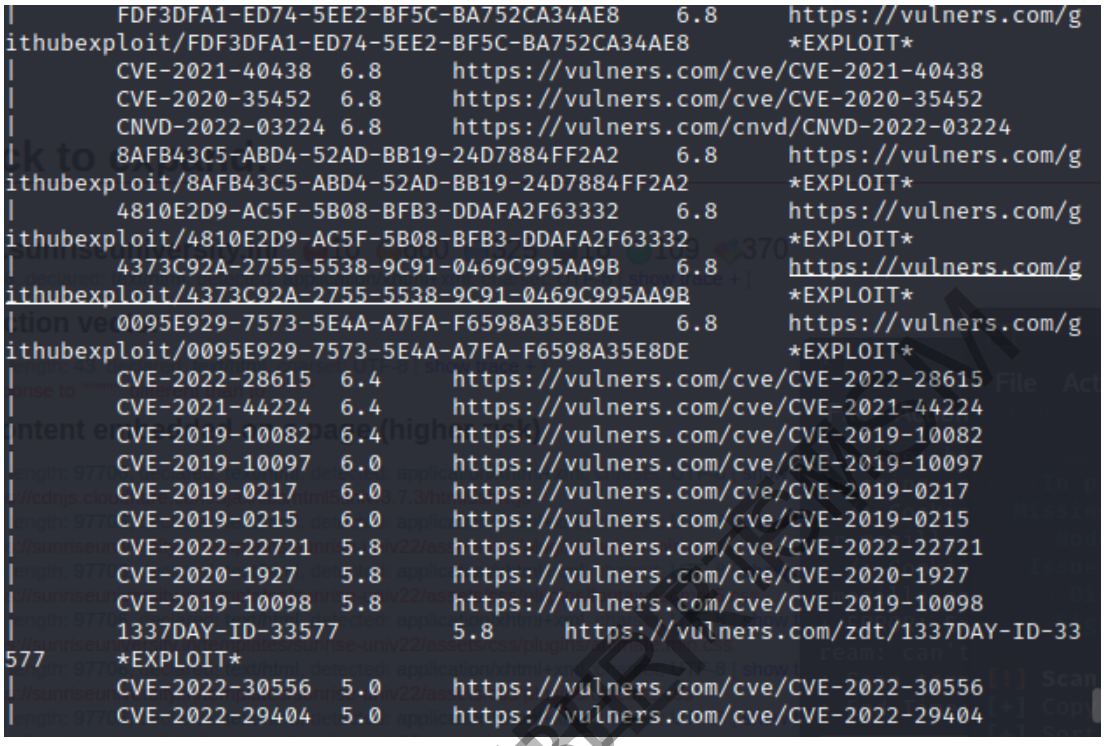


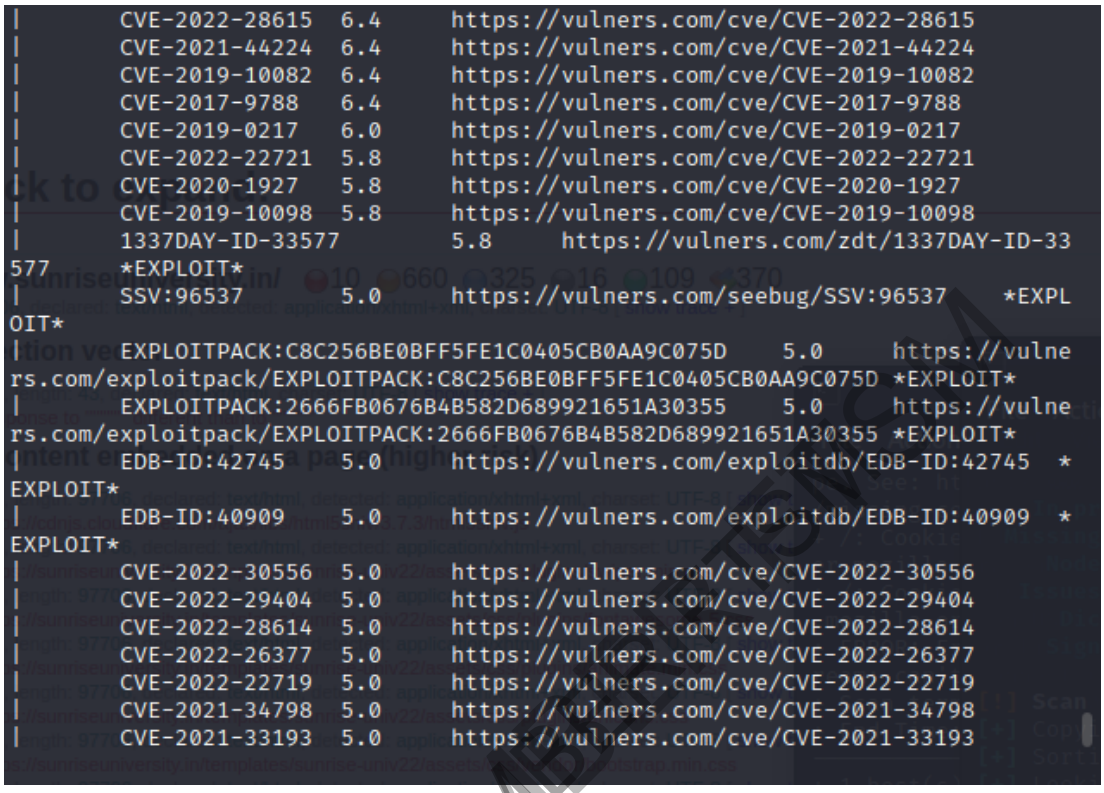
Figure B.3 Result of www.busindia.com

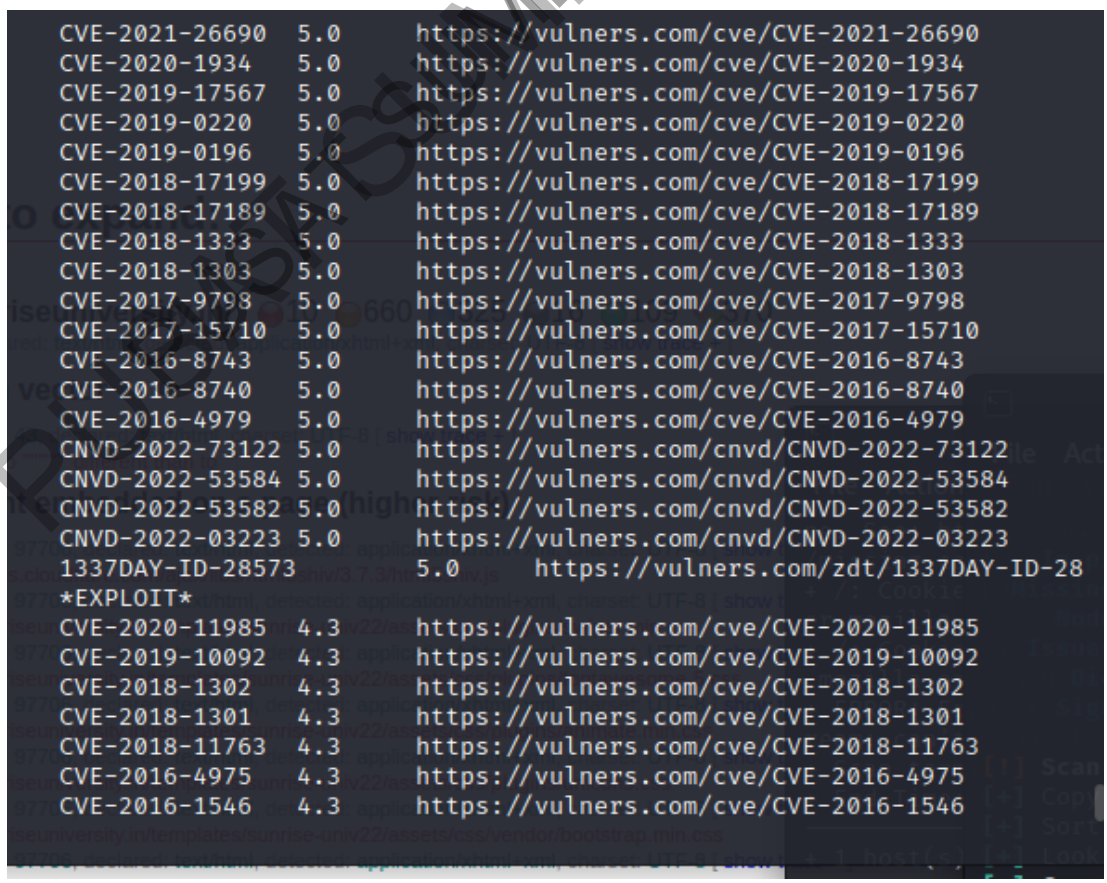
```
Nmap scan report for i2ifunding.com (52.66.84.230)
Host is up (0.33s latency).
rDNS record for 52.66.84.230: ec2-52-66-84-230.ap-south-1.compute.amazonaws.c
om
Not shown: 991 filtered tcp ports (no-response), 5 filtered tcp ports (host-u
nreach)
PORT     STATE  SERVICE     VERSION
80/tcp   open   http        Apache httpd 2.4.18
|_http-server-header: Apache/2.4.18 (Ubuntu)
| vulners:
|   cpe:/a:apache:http_server:2.4.18:
|       PACKETSTORM:171631      7.5     https://vulners.com/packetstorm/PACKE
TSTORM:171631   *EXPLOIT*
|       EDB-ID:51193    7.5     https://vulners.com/exploitdb/EDB-ID:51193  *
EXPLOIT*
|       CVE-2022-31813  7.5     https://vulners.com/cve/CVE-2022-31813
|       CVE-2022-23943  7.5     https://vulners.com/cve/CVE-2022-23943
|       CVE-2022-22720  7.5     https://vulners.com/cve/CVE-2022-22720
|       CVE-2021-44790  7.5     https://vulners.com/cve/CVE-2021-44790
|       CVE-2021-39275  7.5     https://vulners.com/cve/CVE-2021-39275
|       CVE-2021-26691  7.5     https://vulners.com/cve/CVE-2021-26691
|       CVE-2017-7679   7.5     https://vulners.com/cve/CVE-2017-7679
|       CVE-2017-3169   7.5     https://vulners.com/cve/CVE-2017-3169
|       CVE-2017-3167   7.5     https://vulners.com/cve/CVE-2017-3167
|       CNVD-2022-73123 7.5     https://vulners.com/cnvd/CNVD-2022-73123
|       CNVD-2022-03225 7.5     https://vulners.com/cnvd/CNVD-2022-03225
```

```
|       CNVD-2021-102386        7.5     https://vulners.com/cnvd/CNVD-2021-10
2386
|       1337DAY-ID-38427        7.5     https://vulners.com/zdt/1337DAY-ID-38
427     *EXPLOIT*
|       EXPLOITPACK:44C5118F831D55FAF4259C41D8BDA0AB    7.2     https://vulne
rs.com/exploitpack/EXPLOITPACK:44C5118F831D55FAF4259C41D8BDA0AB *EXPLOIT*
|       EDB-ID:46676    7.2     https://vulners.com/exploitdb/EDB-ID:46676  *
EXPLOIT*
|       CVE-2019-0211   7.2     https://vulners.com/cve/CVE-2019-0211
|       1337DAY-ID-32502        7.2     https://vulners.com/zdt/1337DAY-ID-32
502     *EXPLOIT*
|       FDF3DFA1-ED74-5EE2-BF5C-BA752CA34AE8    6.8     https://vulners.com/g
ithubexploit/FDF3DFA1-ED74-5EE2-BF5C-BA752CA34AE8       *EXPLOIT*
|       CVE-2021-40438  6.8     https://vulners.com/cve/CVE-2021-40438
|       CVE-2020-35452  6.8     https://vulners.com/cve/CVE-2020-35452
|       CVE-2018-1312   6.8     https://vulners.com/cve/CVE-2018-1312
|       CVE-2017-15715  6.8     https://vulners.com/cve/CVE-2017-15715
|       CVE-2016-5387   6.8     https://vulners.com/cve/CVE-2016-5387
|       CNVD-2022-03224 6.8     https://vulners.com/cnvd/CNVD-2022-03224
|       8AFB43C5-ABD4-52AD-BB19-24D7884FF2A2    6.8     https://vulners.com/g
ithubexploit/8AFB43C5-ABD4-52AD-BB19-24D7884FF2A2       *EXPLOIT*
|       4810E2D9-AC5F-5B08-BFB3-DDAFA2F63332    6.8     https://vulners.com/g
ithubexploit/4810E2D9-AC5F-5B08-BFB3-DDAFA2F63332       *EXPLOIT*
|       4373C92A-2755-5538-9C91-0469C995AA9B    6.8     https://vulners.com/g
ithubexploit/4373C92A-2755-5538-9C91-0469C995AA9B       *EXPLOIT*
|       0095E929-7573-5E4A-A7FA-F6598A35E8DE    6.8     https://vulners.com/g
ithubexploit/0095E929-7573-5E4A-A7FA-F6598A35E8DE       *EXPLOIT*
```

to be continued…

…continuation





to be continued…

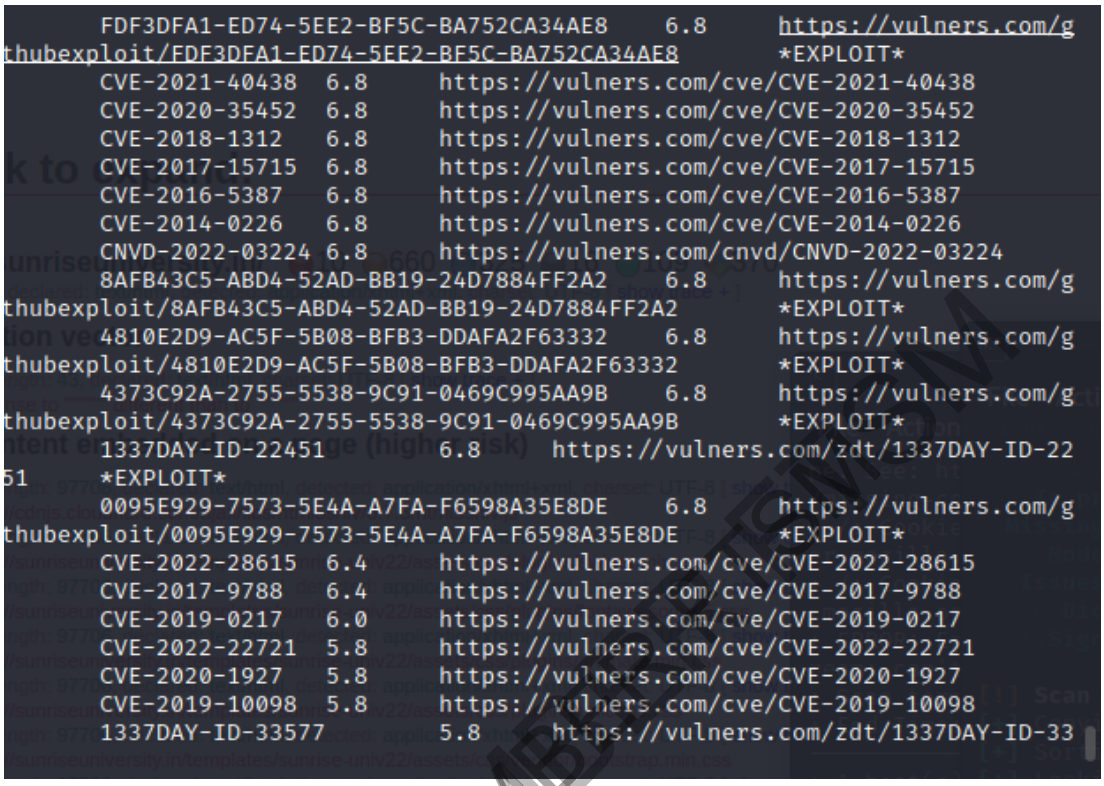
…continuation

```
|       4013EC74-B3C1-5D95-938A-54197A58586D    4.3     https://vulners.com/g
ithubexploit/4013EC74-B3C1-5D95-938A-54197A58586D       *EXPLOIT*
|       1337DAY-ID-33575        4.3     https://vulners.com/zdt/1337DAY-ID-33
575     *EXPLOIT*
|       CVE-2018-1283   3.5     https://vulners.com/cve/CVE-2018-1283
|       CVE-2016-8612   3.3     https://vulners.com/cve/CVE-2016-8612
|       PACKETSTORM:152441      0.0     https://vulners.com/packetstorm/PACKE
TSTORM:152441   *EXPLOIT*
|       CVE-2023-25690  0.0     https://vulners.com/cve/CVE-2023-25690
|       CVE-2022-37436  0.0     https://vulners.com/cve/CVE-2022-37436
|       CVE-2022-36760  0.0     https://vulners.com/cve/CVE-2022-36760
|_      CVE-2006-20001  0.0     https://vulners.com/cve/CVE-2006-20001
139/tcp  closed netbios-ssn
443/tcp  open   ssl/http    Apache httpd 2.4.18
|_http-server-header: Apache/2.4.18 (Ubuntu)
| vulners:
|   cpe:/a:apache:http_server:2.4.18:
|       PACKETSTORM:171631      7.5     https://vulners.com/packetstorm/PACKE
TSTORM:171631   *EXPLOIT*
|       EDB-ID:51193    7.5     https://vulners.com/exploitdb/EDB-ID:51193  *
EXPLOIT*
|       CVE-2022-31813  7.5     https://vulners.com/cve/CVE-2022-31813
|       CVE-2022-23943  7.5     https://vulners.com/cve/CVE-2022-23943
|       CVE-2022-22720  7.5     https://vulners.com/cve/CVE-2022-22720
|       CVE-2021-44790  7.5     https://vulners.com/cve/CVE-2021-44790
|       CVE-2021-39275  7.5     https://vulners.com/cve/CVE-2021-39275
|       CVE-2021-26691  7.5     https://vulners.com/cve/CVE-2021-26691
```

```
Nmap scan report for sifytechnologies.com (1.6.49.193)
Host is up (0.10s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE     VERSION
80/tcp  open   http        Apache httpd 2.4.6 ((Red Hat Enterprise Linux) Ope
nSSL/1.0.2k-fips PHP/7.4.10)
| vulners:
|   cpe:/a:apache:http_server:2.4.6:
|       PACKETSTORM:171631      7.5     https://vulners.com/packetstorm/PACKE
TSTORM:171631   *EXPLOIT*
|       EDB-ID:51193    7.5     https://vulners.com/exploitdb/EDB-ID:51193  *
EXPLOIT*
|       CVE-2022-31813  7.5     https://vulners.com/cve/CVE-2022-31813
|       CVE-2022-23943  7.5     https://vulners.com/cve/CVE-2022-23943
|       CVE-2022-22720  7.5     https://vulners.com/cve/CVE-2022-22720
|       CVE-2021-44790  7.5     https://vulners.com/cve/CVE-2021-44790
|       CVE-2021-39275  7.5     https://vulners.com/cve/CVE-2021-39275
|       CVE-2021-26691  7.5     https://vulners.com/cve/CVE-2021-26691
|       CVE-2017-7679   7.5     https://vulners.com/cve/CVE-2017-7679
|       CVE-2017-3167   7.5     https://vulners.com/cve/CVE-2017-3167
|       CNVD-2022-73123 7.5     https://vulners.com/cnvd/CNVD-2022-73123
|       CNVD-2022-03225 7.5     https://vulners.com/cnvd/CNVD-2022-03225
|       CNVD-2021-102386        7.5     https://vulners.com/cnvd/CNVD-2021-10
2386
|       1337DAY-ID-38427        7.5     https://vulners.com/zdt/1337DAY-ID-38
427     *EXPLOIT*
|       PACKETSTORM:127546      6.8     https://vulners.com/packetstorm/PACKE
```

to be continued…

…continuation



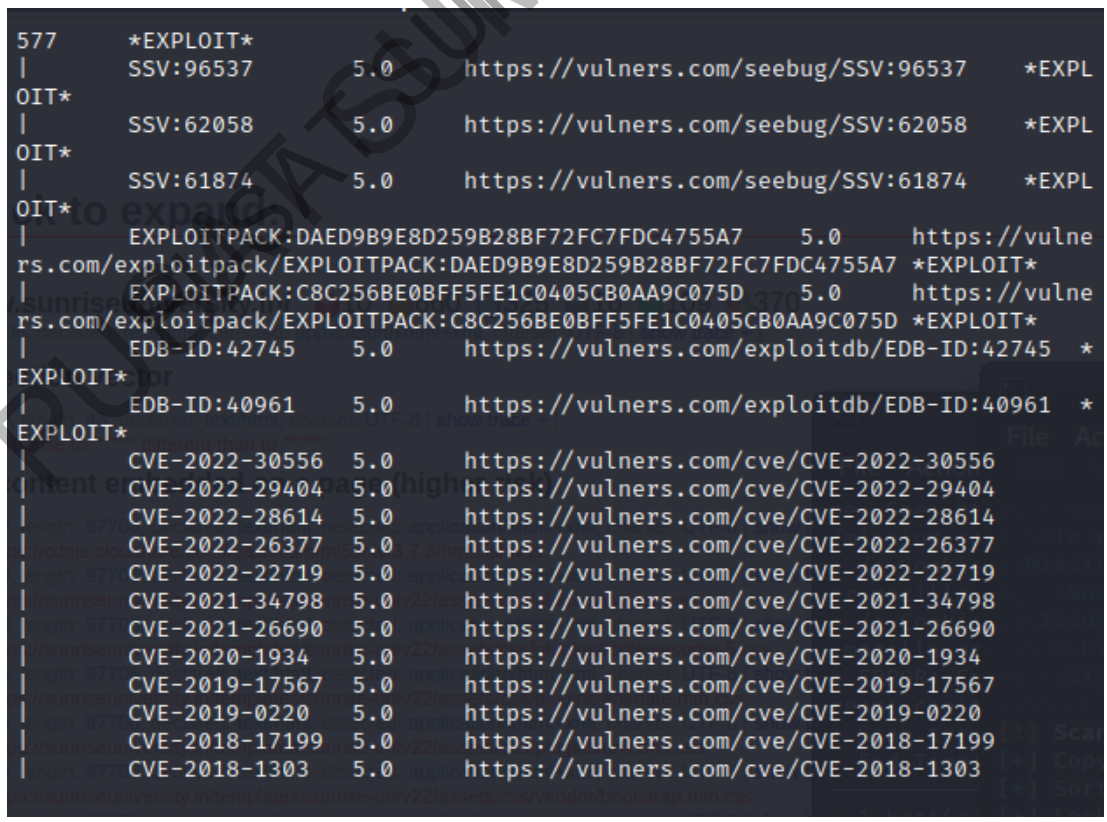
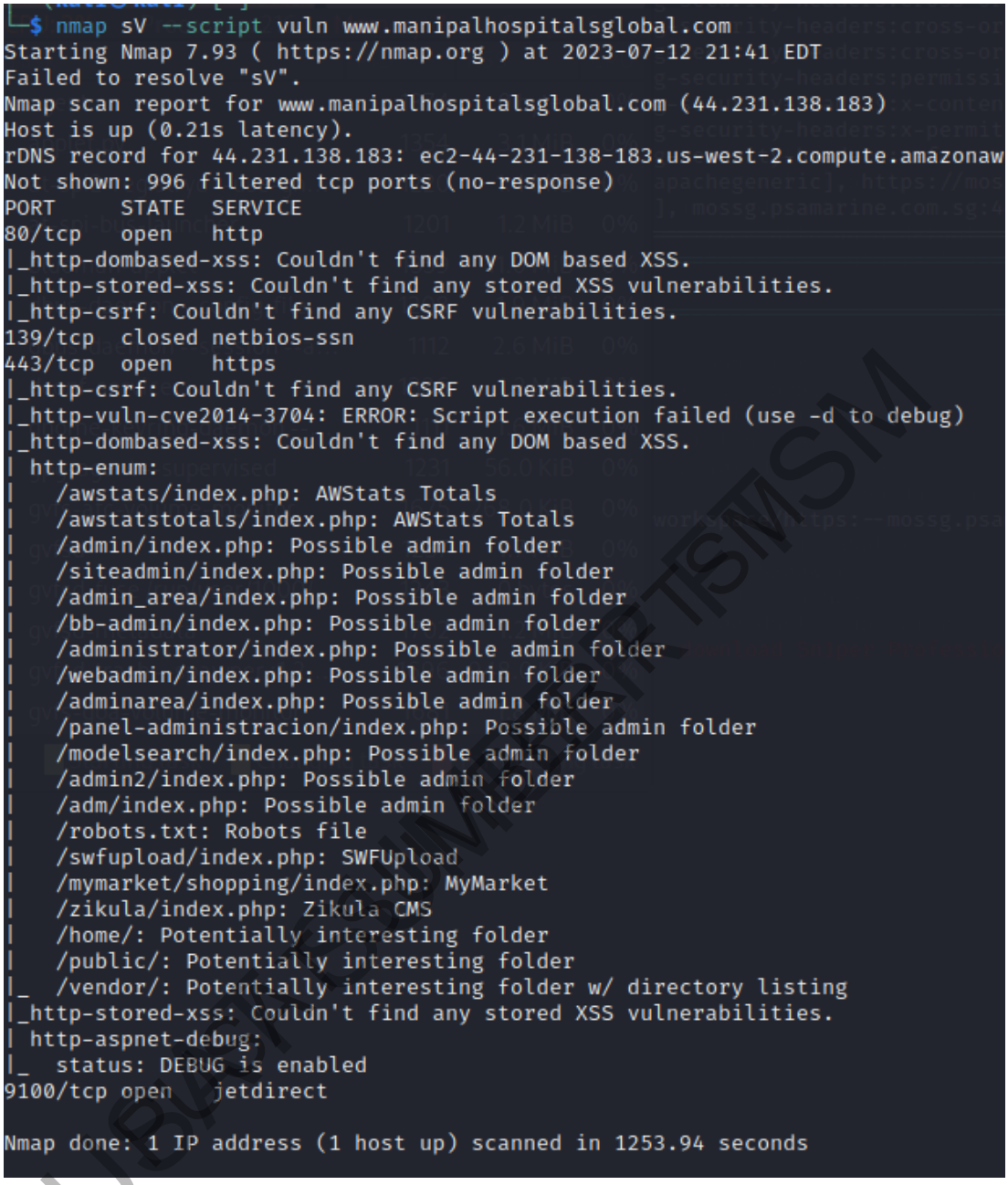


Figure B.4 Result of www.i2ifunding.com

```
└─$ nmap sV --script vuln www.amity.edu
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-12 21:33 EDT
Failed to resolve "sV".
Nmap scan report for www.amity.edu (52.66.187.102)
Host is up (0.24s latency).
rDNS record for 52.66.187.102: ec2-52-66-187-102.ap-south-1.compute.amazonaws.com
Not shown: 993 filtered tcp ports (no-response), 4 filtered tcp ports (host-unreach)
PORT    STATE  SERVICE
80/tcp  open   http
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-vuln-cve2013-7091: ERROR: Script execution failed (use -d to debug)
139/tcp closed netbios-ssn
443/tcp open   https
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=www.amity.edu
|   Found the following possible CSRF vulnerabilities:
|
|     Path: https://www.amity.edu:443/
|     Form id: form1
|     Form action: /
|
|     Path: https://www.amity.edu:443/infra-play-initiatives.aspx
|     Form id: aspnetform
|     Form action: /infra-play-initiatives.aspx
|
|     Path: https://www.amity.edu:443/infra-play-facilities.aspx
|     Form id: aspnetform
|     Form action: /infra-play-facilities.aspx
|
|     Path: https://www.amity.edu:443/student-section.aspx
|     Form id: aspnetform
|_    Form action: /student-section.aspx
| http-aspnet-debug:
|_  status: DEBUG is enabled
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.

Nmap done: 1 IP address (1 host up) scanned in 287.35 seconds
```

Figure B.5 Result of www.amity.edu



```
└─$ nmap sV --script vuln www.manipalhospitalsglobal.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-12 21:41 EDT
Failed to resolve "sV".
Nmap scan report for www.manipalhospitalsglobal.com (44.231.138.183)
Host is up (0.21s latency).
rDNS record for 44.231.138.183: ec2-44-231-138-183.us-west-2.compute.amazonaw
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE  SERVICE
80/tcp   open   http
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
139/tcp  closed netbios-ssn
443/tcp  open   https
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|   /awstats/index.php: AWStats Totals
|   /awstatstotals/index.php: AWStats Totals
|   /admin/index.php: Possible admin folder
|   /siteadmin/index.php: Possible admin folder
|   /admin_area/index.php: Possible admin folder
|   /bb-admin/index.php: Possible admin folder
|   /administrator/index.php: Possible admin folder
|   /webadmin/index.php: Possible admin folder
|   /adminarea/index.php: Possible admin folder
|   /panel-administracion/index.php: Possible admin folder
|   /modelsearch/index.php: Possible admin folder
|   /admin2/index.php: Possible admin folder
|   /adm/index.php: Possible admin folder
|   /robots.txt: Robots file
|   /swfupload/index.php: SWFUpload
|   /mymarket/shopping/index.php: MyMarket
|   /zikula/index.php: Zikula CMS
|   /home/: Potentially interesting folder
|   /public/: Potentially interesting folder
|_  /vendor/: Potentially interesting folder w/ directory listing
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-aspnet-debug:
|_  status: DEBUG is enabled
9100/tcp open   jetdirect

Nmap done: 1 IP address (1 host up) scanned in 1253.94 seconds
```

Figure B.6 Result of www.manipalhospitalsglobal.com

```
┌──(kali㉿kali)-[~]
└─$ nmap sV --script vuln www.sprink.online
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-12 22:13 EDT
Failed to resolve "sV".
Nmap scan report for www.sprink.online (52.221.214.232)
Host is up (0.10s latency).
rDNS record for 52.221.214.232: ec2-52-221-214-232.ap-southeast-1.compute.amazonaws.com
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE  SERVICE
22/tcp   open   ssh
80/tcp   open   http
|_http-passwd: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-vuln-cve2013-7091: ERROR: Script execution failed (use -d to debug)
139/tcp  closed netbios-ssn
3306/tcp open   mysql

Nmap done: 1 IP address (1 host up) scanned in 623.90 seconds
```

Figure B.7 Result of www.sprink.online

```
└─$ nmap sV --script vuln www.medtravels.in
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-13 01:39 EDT
Failed to resolve "sV".
Nmap scan report for www.medtravels.in (162.214.156.4)
Host is up (0.21s latency).
rDNS record for 162.214.156.4: cloud.servers800.com
Not shown: 984 filtered tcp ports (no-response), 3 filtered tcp ports (host-unreach)
PORT     STATE  SERVICE
21/tcp   open   ftp
26/tcp   open   rsftp
53/tcp   open   domain
80/tcp   open   http
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|   //system.html: CMNC-200 IP Camera
|   /admin/: Possible admin folder
|   /webmail/: Mail folder
|   /robots.txt: Robots file
|   /info.php: Possible information file
|   /health/: Spring Boot Actuator endpoint
|   /webmail/images/sm_logo.png: SquirrelMail
|   /application/: Potentially interesting folder
|   /beta/: Potentially interesting folder
|   /controlpanel/: Potentially interesting folder
|   /partner/: Potentially interesting folder
|   /public/: Potentially interesting folder
|_  /register/: Potentially interesting folder
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_http-fileupload-exploiter: ERROR: Script execution failed (use -d to debug)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=www.medtravels.in
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://www.medtravels.in:80/
|     Form id: query_from
|     Form action: https://www.medtravels.in/enquiry
|
|     Path: http://www.medtravels.in:80/
|     Form id: editdoctor
```

to be continued…

…continuation

```
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=www.medtravels.in
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://www.medtravels.in:80/
|     Form id: query_from
|     Form action: https://www.medtravels.in/enquiry
|
|     Path: http://www.medtravels.in:80/
|     Form id: editdoctor
|     Form action: https://www.medtravels.in/medical-treatment
|
|     Path: http://www.medtravels.in:80/
|     Form id: comment
|     Form action: https://www.medtravels.in/enquiry
|
|     Path: http://www.medtravels.in:80/
|     Form id: form1
|     Form action:
|
|     Path: http://www.medtravels.in:80/
|     Form id: form1
|_    Form action:
110/tcp  open   pop3
139/tcp  closed netbios-ssn
143/tcp  open   imap
443/tcp  open   https
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-fileupload-exploiter: ERROR: Script execution failed (use -d to debug)
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=www.medtravels.in
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://www.medtravels.in:443/
|     Form id: query_from
|     Form action: https://www.medtravels.in/enquiry
|
|     Path: http://www.medtravels.in:443/
|     Form id: editdoctor
|     Form action: https://www.medtravels.in/medical-treatment
|
|     Path: http://www.medtravels.in:443/
|     Form id: comment
|     Form action: https://www.medtravels.in/enquiry
|
|     Path: http://www.medtravels.in:443/
|     Form id: form1
|     Form action:
|
```


to be continued…

…continuation

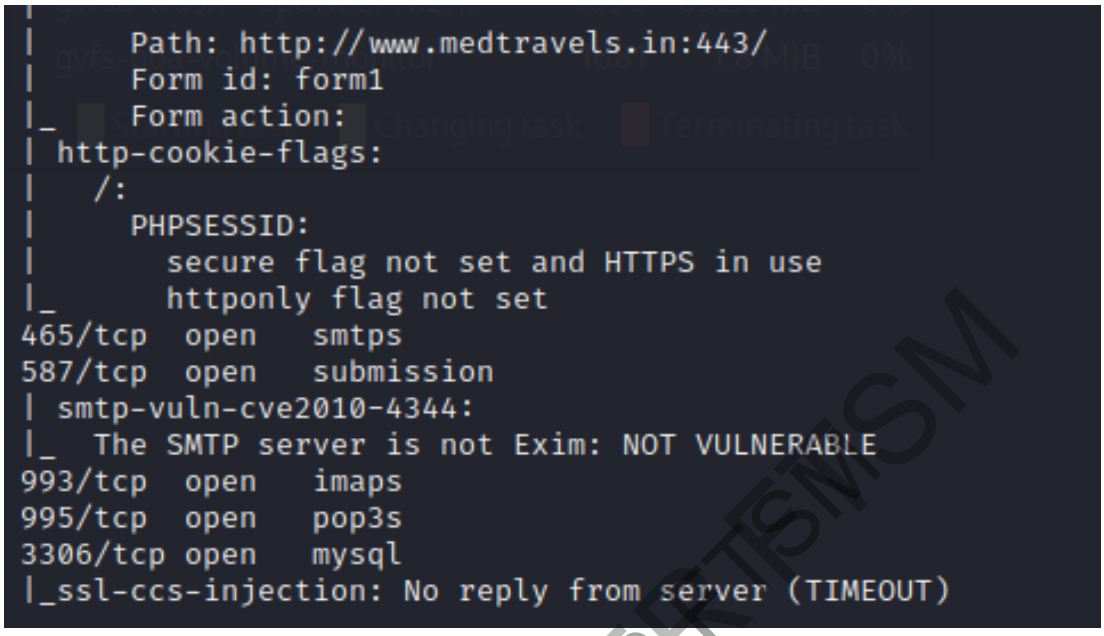


Figure B.8 Result of www.medtravels.in

APPENDIX C

SUBLISTER RESULT

Figure C.1 Result of www.i2ifunding.com

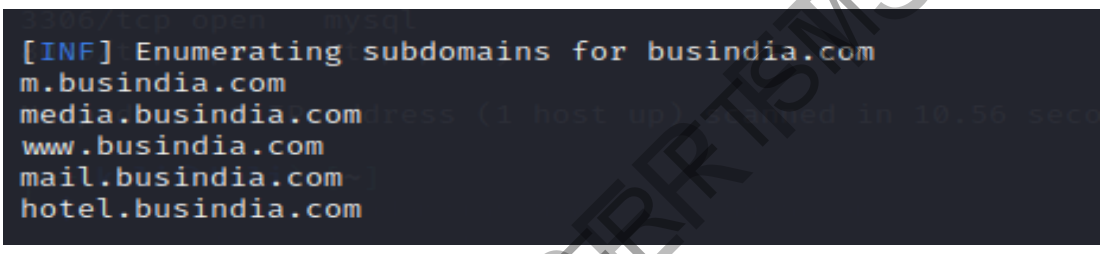


Figure C.2 Result of www.busindia.com

Figure C.3 Result of www.sifytechnologies.com

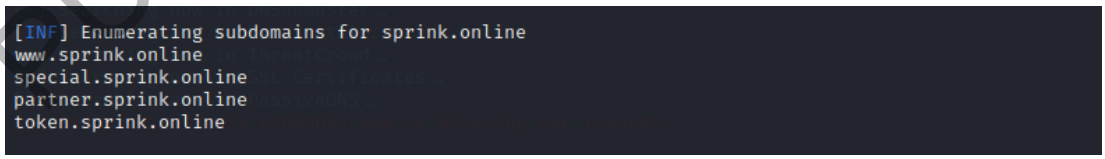


Figure C.4 Result of www.sprink.online

```
www.sunriseuniversity.in
webdisk.sunriseuniversity.in
cpcalendars.sunriseuniversity.in
cpanel.sunriseuniversity.in
hostmaster.sunriseuniversity.in
newweb.sunriseuniversity.in
autodiscover.sunriseuniversity.in
cpcontacts.sunriseuniversity.in
www.erp.sunriseuniversity.in
erp.sunriseuniversity.in
mail.sunriseuniversity.in
webmail.sunriseuniversity.in
```

Figure C.5 Result of www. sunriseuniversity.in

```
[INF] Enumerating subdomains for amity.edu
mail.amity.edu
www.amity.edu
portal.amity.edu
aitd.amity.edu
auup.amity.edu
aice.amity.edu
spinkg.amity.edu
new.amity.edu
mail.auup.amity.edu
alumni.amity.edu
addoe.amity.edu
abs.amity.edu
virtualconvocation.amity.edu
amity.edu
webmail.amity.edu
odl.addoe.amity.edu
admissions2020.amity.edu
www.aitd.amity.edu
_sipfederationtls._tcp.pun.amity.edu
www.addoe.amity.edu
aiss.amity.edu
asoe.amity.edu
ags.amity.edu
aismv.amity.edu
aisn.amity.edu
alsonline.amity.edu
london.amity.edu
ww1.amity.edu
mysbeindia.amity.edu
singapore.amity.edu
ajhp.amity.edu
ais.amity.edu
mauritius.amity.edu
ainst.amity.edu
aic.amity.edu
aib.amity.edu
aittm.amity.edu
jpr.amity.edu
pb.amity.edu
pg-communication-courseswww.amity.edu
20www.amity.edu
mail.abs.amity.edu
admissions2022.amity.edu
admissions2023.amity.edu
```

Figure C.6 Result of www.amity.edu

```
[INF] Enumerating subdomains for medtravels.in
www.medtravels.in
cpcalendars.medtravels.in
cpcontacts.medtravels.in
webdisk.medtravels.in
cpanel.medtravels.in
mail.medtravels.in
webmail.medtravels.in
```

Figure C.7 Result of www.medtravels.in

```
[INF] Enumerating subdomains for manipalhospitalsglobal.com
www.manipalhospitalsglobal.com
stage.manipalhospitalsglobal.com
test.manipalhospitalsglobal.com
```

Figure C.8 Result of www.manipalhospitalsglobal.com

APPENDIX D

QUALYS REPORT

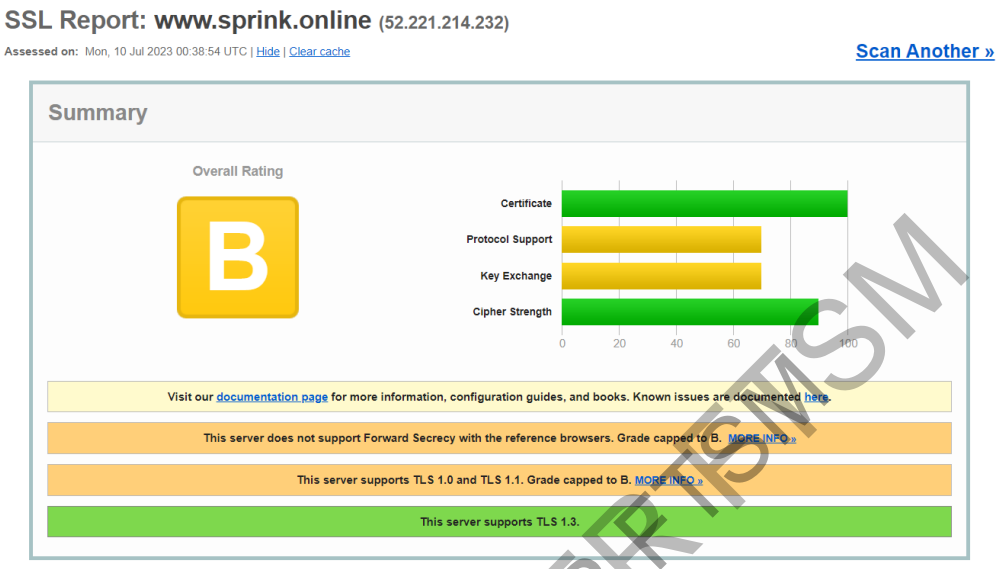


Figure D.1 Result of www.sprink.online

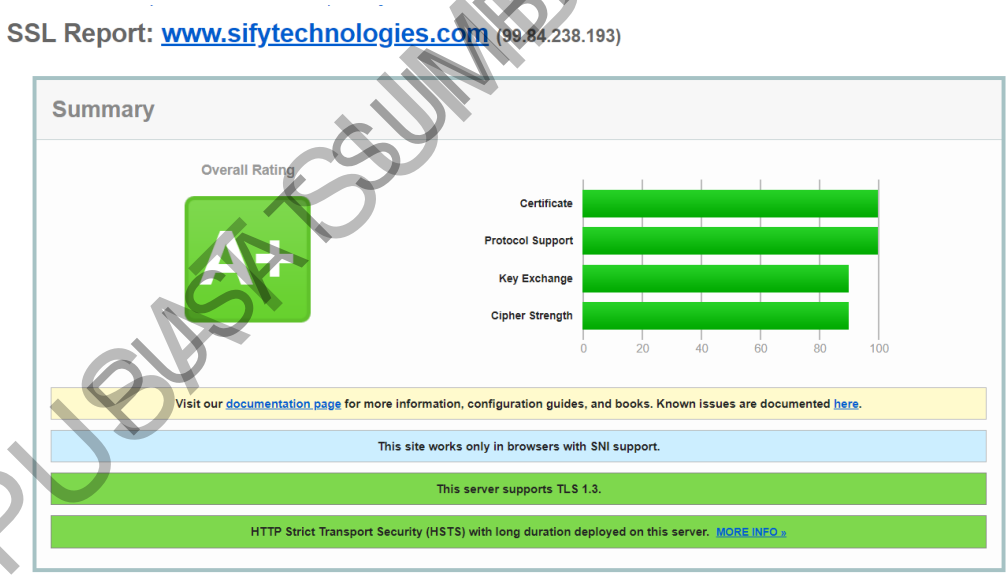


Figure D.2 Result of www.sifytechnologies.com

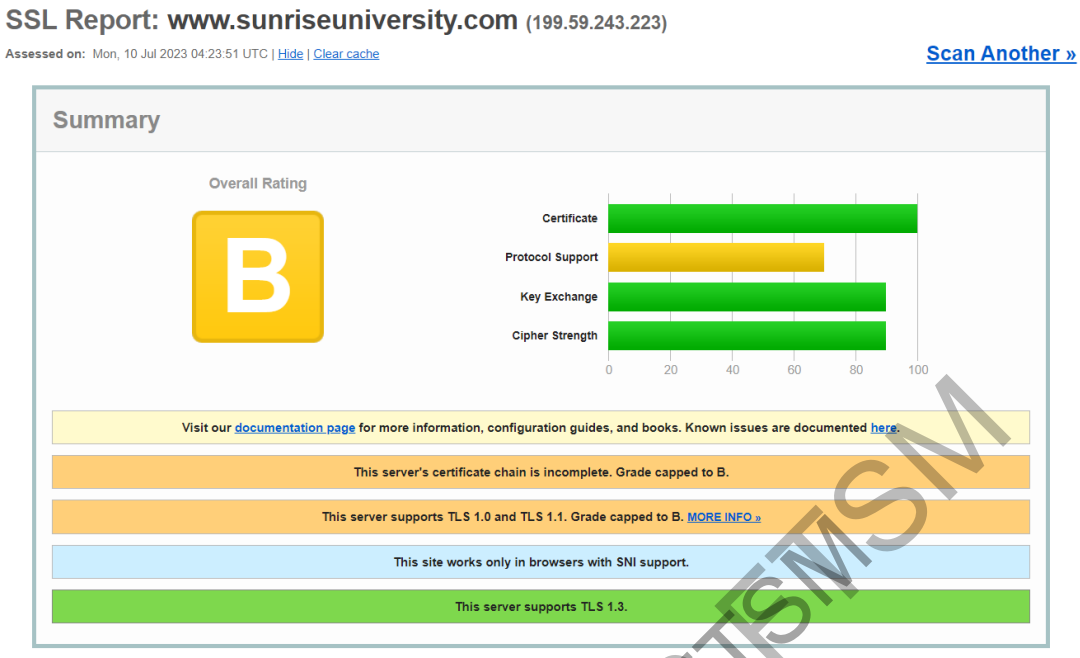


Figure D.3 Result www.sunriseuniversity.in

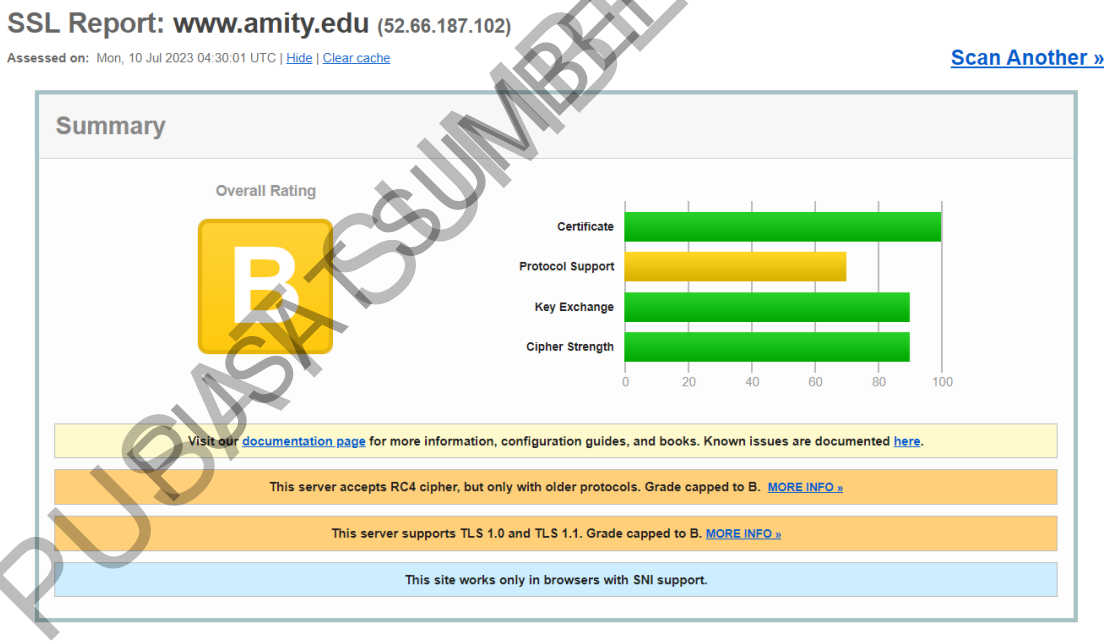


Figure D.4 Result of www.amity.edu

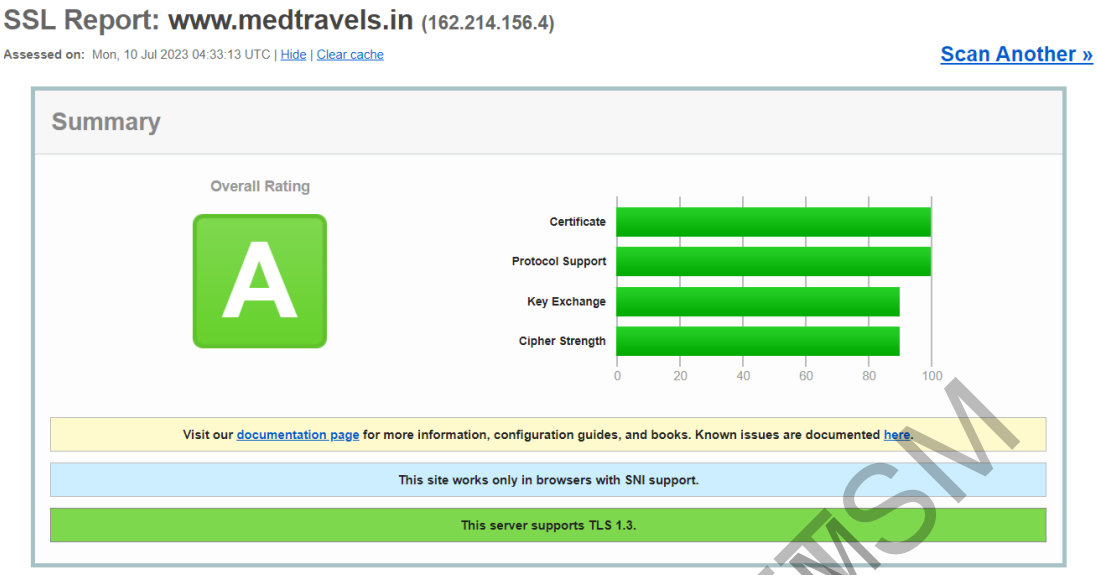


Figure D.5 Result of www.medtravels.in

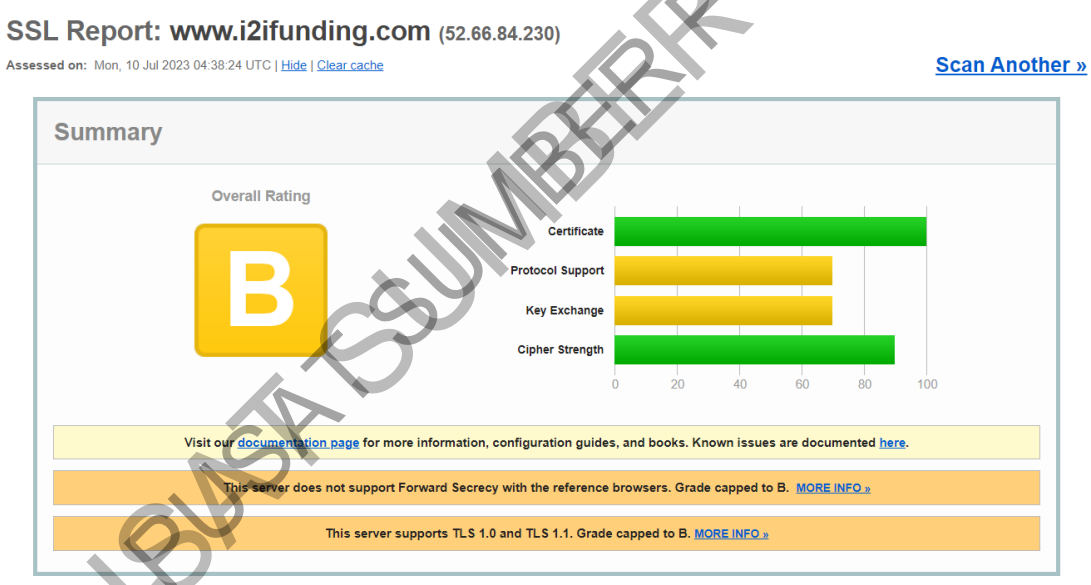


Figure D.6 Result of www.i2ifunding.com

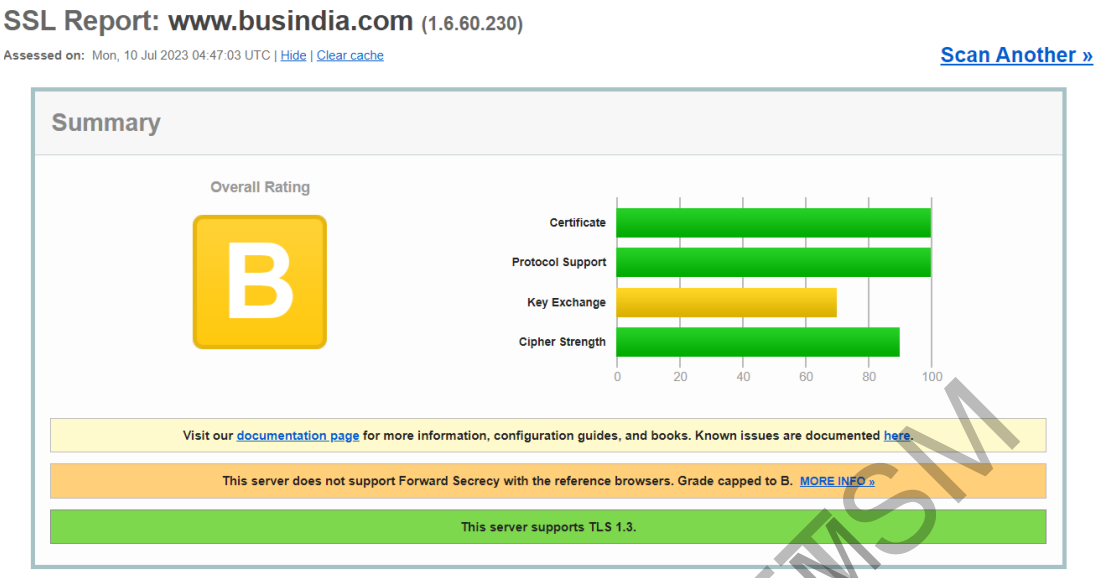


Figure D.7 Result of www.businda.com

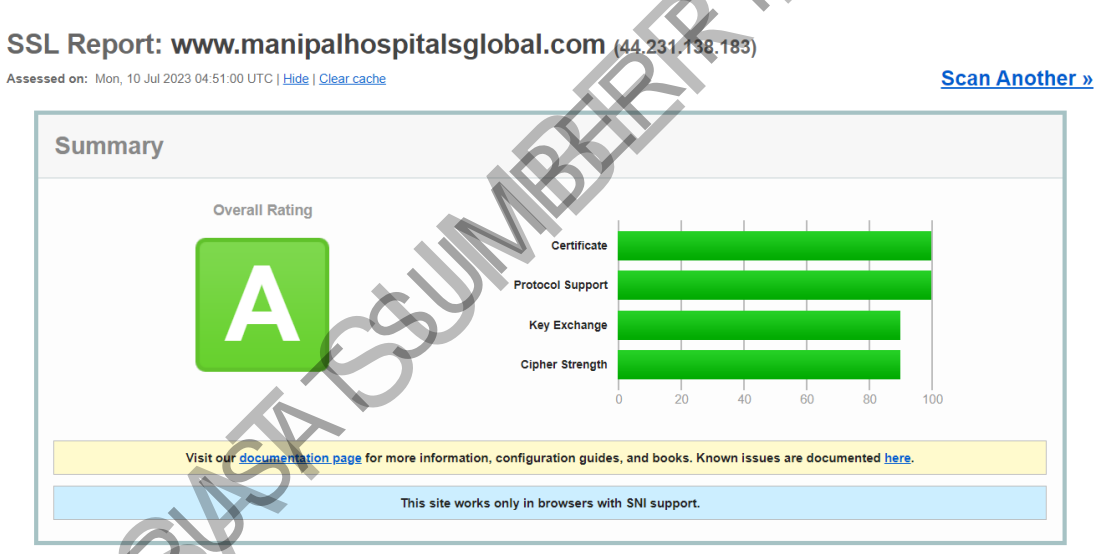


Figure D.8 Result of www.manipalhospitalsglobal.com

APPENDIX E

SQLMAP REPORT

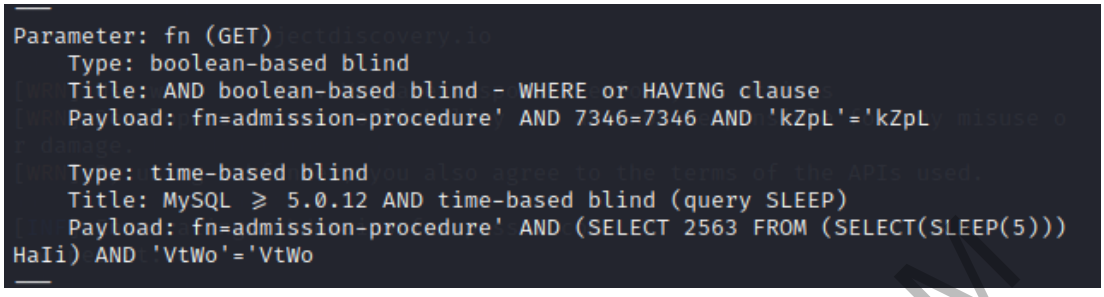


Figure E1 Result of www.sifytechnologies.com