Cyber Security Project Report

Date	10 March 2025
Team ID	PNT2025TMID02564
Project Name	Exploring Cyber Security: Understanding Threats and Solutions in the Digital Age
Maximum Marks	8 Marks

Team Details:--

S.no	name	collage	contact
1	Anuj Kadam	DYP-ATU	anujkadam3554@gmail.com
2	Arin Irache	DYP-ATU	irachearin@gmail.com
3	Akif Panari	DYP-ATU	akifpanari@gmail.com
4	Abhishek Patil	DYP-ATU	abhidpatil27@gmail.com

Institution: DY Patil Agriculture and Technical University, Talsande

1. INTRODUCTION

1.1 Project Name

Exploring Cyber Security: Understanding Threats and Solutions in the Digital Age

Team ID: PNT2025TMID02564
Maximum Marks: 8 Marks

1.2 Purpose

Abstract:
In this project, our team conducted a thorough, hands-on assessment of cybersecurity vulnerabilities by manually scanning and analysing designated web applications and network systems. Using industry-standard tools such as Nessus, OWASP ZAP, and Burp Suite, we identified critical vulnerabilities, evaluated their potential business impact, and proposed effective remediation strategies. Our research also discusses current cybersecurity frameworks, incident response plans, and emerging trends to enhance overall digital security.

Scope of the Project:

- Target Environment: Web applications and network systems of designated test sites.
- Tools & Techniques: A combination of manual testing and automated scanning (e.g., Nessus, OWASP ZAP, Burp Suite, Wireshark).
- Focus Areas:
    - Identification and categorization of vulnerabilities
    - Business impact analysis of each issue
    - Detailed mitigation recommendations
    - Exploration of cybersecurity frameworks and future trends

2. IDEATION PHASE

2.1 Thought Behind the Project

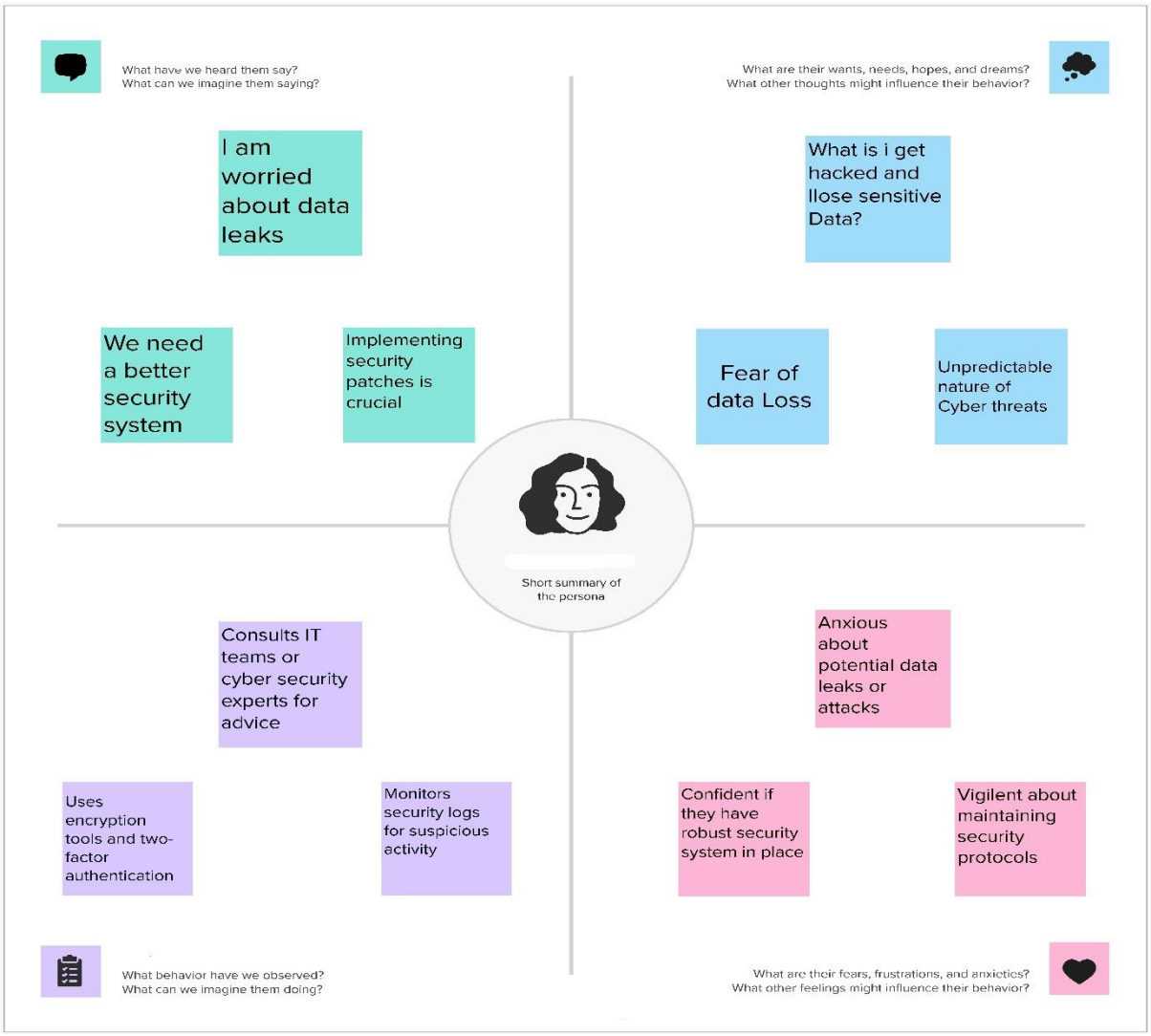
Our team brainstormed multiple ideas based on known attack vectors and our previous hands-on experiences. Initial scans provided an early list of potential vulnerabilities which we refined by reviewing academic research and industry case studies.

2.2 Features

- Data Collection:
    - Manual scanning of target websites using tools like Burp Suite.
    - Automated vulnerability scanning using Nessus.
- Feature Grouping:
    - Categorization of vulnerabilities into web application issues (e.g., SQL Injection, XSS, CSRF) and network-related issues

2.3 Empathy Map

We developed an empathy map to understand the perspectives of end users and stakeholders. This helped us focus on concerns such as data privacy, potential financial loss, and overall system trustworthiness.



3. REQUIREMENT ANALYSIS

3.1 List of Vulnerabilities

Sr .No	Vulnerability Name	CWE Number
1	Insecure Direct Object References (IDOR)	639
2	Cross-Site Request Forgery (CSRF)	352
3	Security Misconfiguration	16
4	Unvalidated Redirects and Forwards	601
5	XML External Entity Injection (XXE)	611

3.2 Solution Requirement

Based on our vulnerability assessments, our solution requirements include:

- Risk Evaluation: Detailed analysis of the severity and business impact of each identified vulnerability.
- Remediation Measures: Propose and implement countermeasures such as strict server-side validation, proper use of CSRF tokens, secure configuration practices, and safe XML parsing methods.
- Documentation: Comprehensive reporting of our findings to support ongoing security improvements.

3.3 Technology Stack

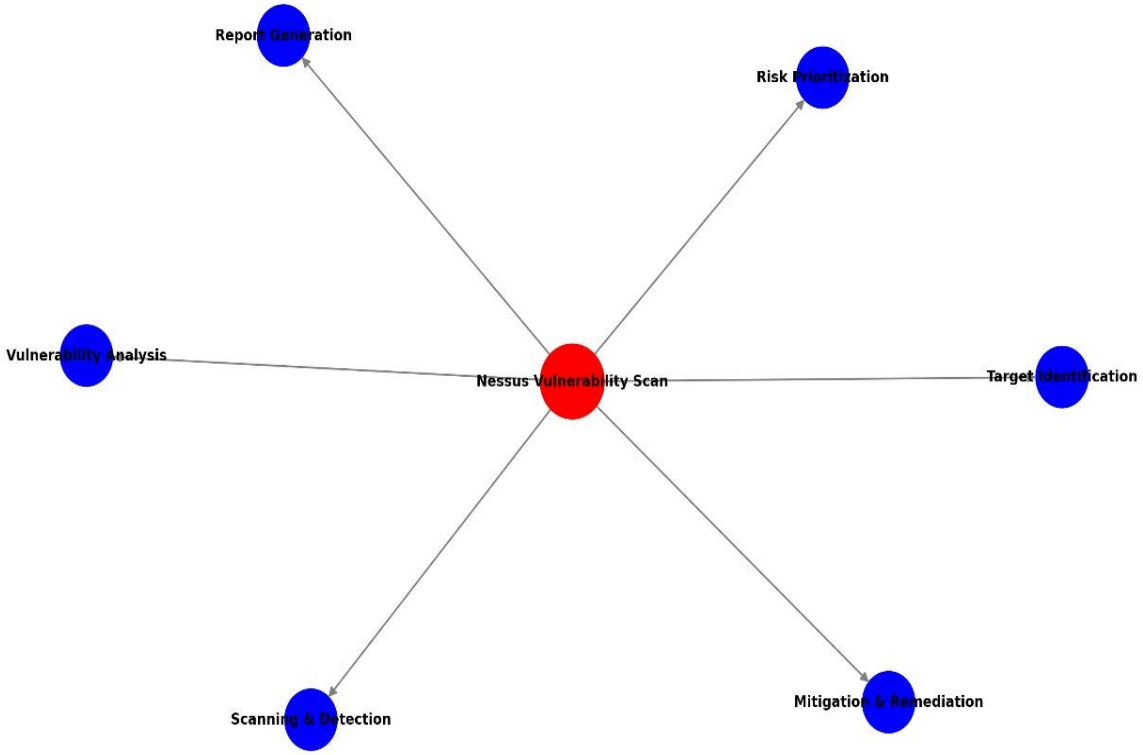
Tools Explored:

- Nessus: For automated vulnerability scanning.
- OWASP ZAP & Burp Suite: For manual penetration testing and vulnerability verification.
- Wireshark: For network traffic analysis.
- Metasploit Framework: For simulating attacks and verifying vulnerabilities.
- Kali Linux: Operating system bundling various ethical hacking tools**.**

4. PROJECT DESIGN

4.1 Overview of Nessus

Nessus is an industry-standard vulnerability scanner that automates the software, and exploitable vulnerabilities. Our use of Nessus provided a detailed risk assessment and helped prioritize remediation efforts.



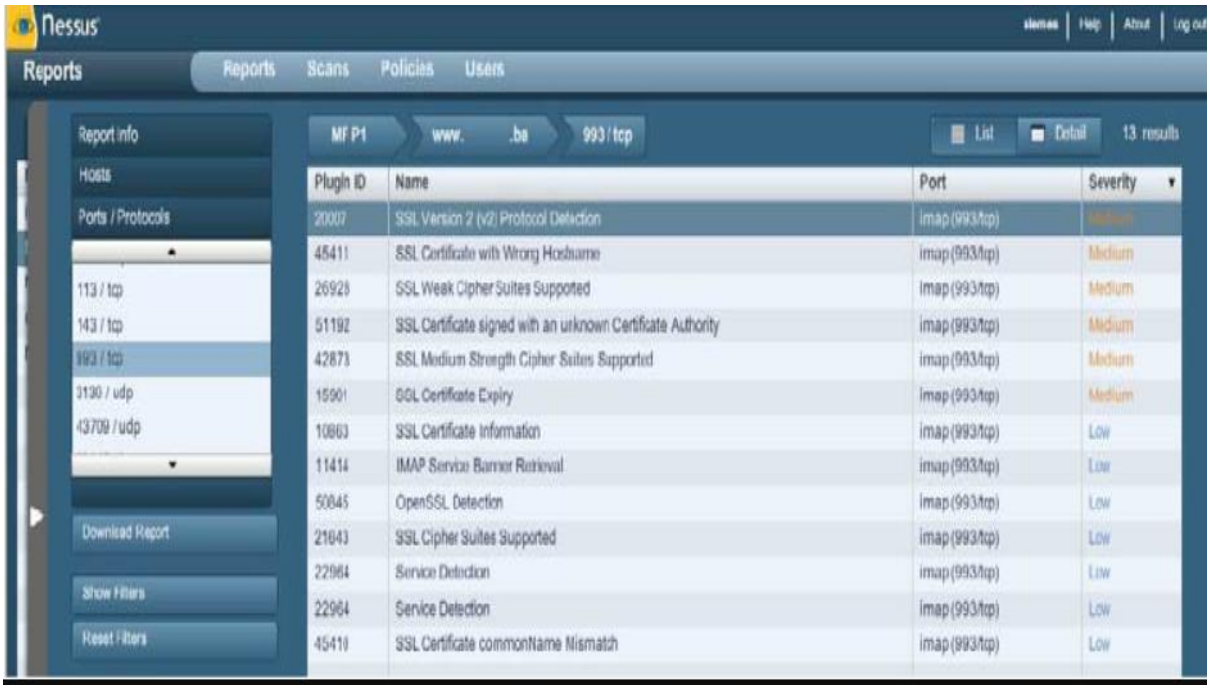
4.2 Proposed Solution (Testing and Findings)

Testing Approach:

- Combined manual testing with automated scanning to validate vulnerabilities.
- Verified findings using multiple tools to ensure accuracy.
- Documented each vulnerability with details on business impact and recommended fixes.

Findings:

- IDOR: Demonstrated through URL parameter manipulation.
- CSRF: Validated via crafted malicious forms without token validation.
- Security Misconfiguration: Identified through use of default credentials and exposed configuration files.
- Unvalidated Redirects: Confirmed by altering URL redirects.
- XXE: Demonstrated through malicious XML payloads extracting sensitive data.



4.3 Understanding of the Main Theme

Our project also explores broader cybersecurity aspects such as SOC and SIEM. By examining systems like IBM Q-Radar, Splunk, and ArcSight, we highlight the role of real-time log analysis and coordinated incident response in mitigating security threats.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Our project planning involved detailed scheduling and task allocation based on our initial reference template. The plan included:

- Phase 1: Vulnerability assessment of web applications.
- Phase 2: Automated scanning with Nessus and complementary manual testing.
- Phase 3: Comprehensive cybersecurity analysis covering frameworks, SOC/SIEM insights, and future trends.

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Vulnerability Report

We conducted in-depth functional and performance testing to document the impact of each vulnerability. The report details:

- IDOR, CSRF, Misconfiguration, Unvalidated Redirects, XXE: Each vulnerability's description, impact, discovery method, and remediation measures.
- Business Impact: Analysis on how each vulnerability affects system integrity and user trust.

S. No.	Vulnerability	Test Method	Outcome
1	Insecure Direct Object References (IDOR)	- Intercepted requests using Burp Suite. - Modified user_id parameter to different values. - Observed if restricted data was accessible.	Confirmed unauthorized data access, proving IDOR vulnerability.
2	Cross-Site Request Forgery (CSRF)	- Created a malicious HTML form mimicking a password change. - Logged in as a victim in a separate session. - Submitted the malicious form to see if changes were applied.	Confirmed no CSRF token or verification; the malicious request was successful.
3	Security Misconfiguration	- Attempted default credentials (bee/bug). - Searched for exposed config files via Dirb/Gobuster.	Successfully logged in with defaults; found / phpinfo.php and .bak files accessible.
4	Unvalidated Redirects and Forwards	- Looked for redirect parameters (e.g., redirect.php?url=). - Modified the URL to point to external malicious domains.	Verified the application redirected users to untrusted sites without any validation.
5	XML External Entity Injection (XXE)	- Submitted malicious XML payloads. - Checked if external entities (e.g., <!DOCTYPE foo [<!ENTITY xxe SYSTEM "file:///etc/passwd">]>) were processed.	Successfully read local files, confirming XXE vulnerability.

S. No.	Vulnerability	Test Method	Outcome
1	Cross-Site Scripting (XSS)	- Injected malicious scripts (<script>alert('XSS')</script>) into search fields. - Observed if script execution occurred in the returned page.	Confirmed script execution in the browser, indicating improper input validation/ encoding.
2	Cross-Site Request Forgery (CSRF)	- Crafted malicious links/forms to perform actions on a logged-in user's behalf (e.g., password change). - Observed successful execution without CSRF tokens.	Verified that user actions could be triggered silently, indicating lack of CSRF protection.
3	Insecure Direct Object References	- Modified resource identifiers (user IDs, order IDs) in URLs and request parameters. - Checked if we could access or alter another user's data.	Confirmed unauthorized data access or modification, demonstrating IDOR vulnerability.
4	SQL Injection	- Inserted SQL payloads (e.g., ' OR 1=1--) in login fields. - Monitored the application's error messages or login bypass.	Bypassed authentication with malicious queries, proving SQL Injection.
5	Broken Authentication	- Exploited weak session tokens or session management flaws. - Tested known default/weak passwords to see if unauthorized access was possible.	Achieved unauthorized access and user impersonation, confirming broken authentication.

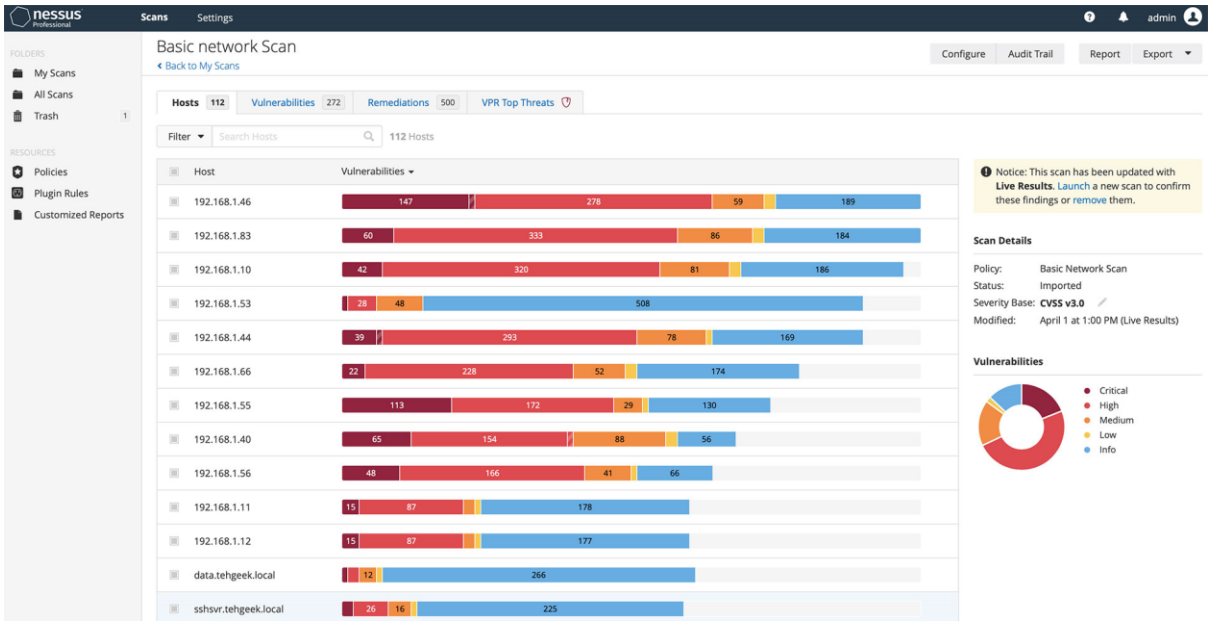
7. RESULTS

7.1 Findings and Reports

Our testing yielded the following key results:

- Nessus Scan Findings: Identified issues such as outdated software, open ports, weak encryption, and zero-day exploit susceptibilities.

- Manual Testing Insights: Confirmed critical vulnerabilities like IDOR, CSRF, and XXE.
- SOC Analysis: Reviewed real-time log analysis and threat detection measures, emphasizing the need for continuous monitoring.



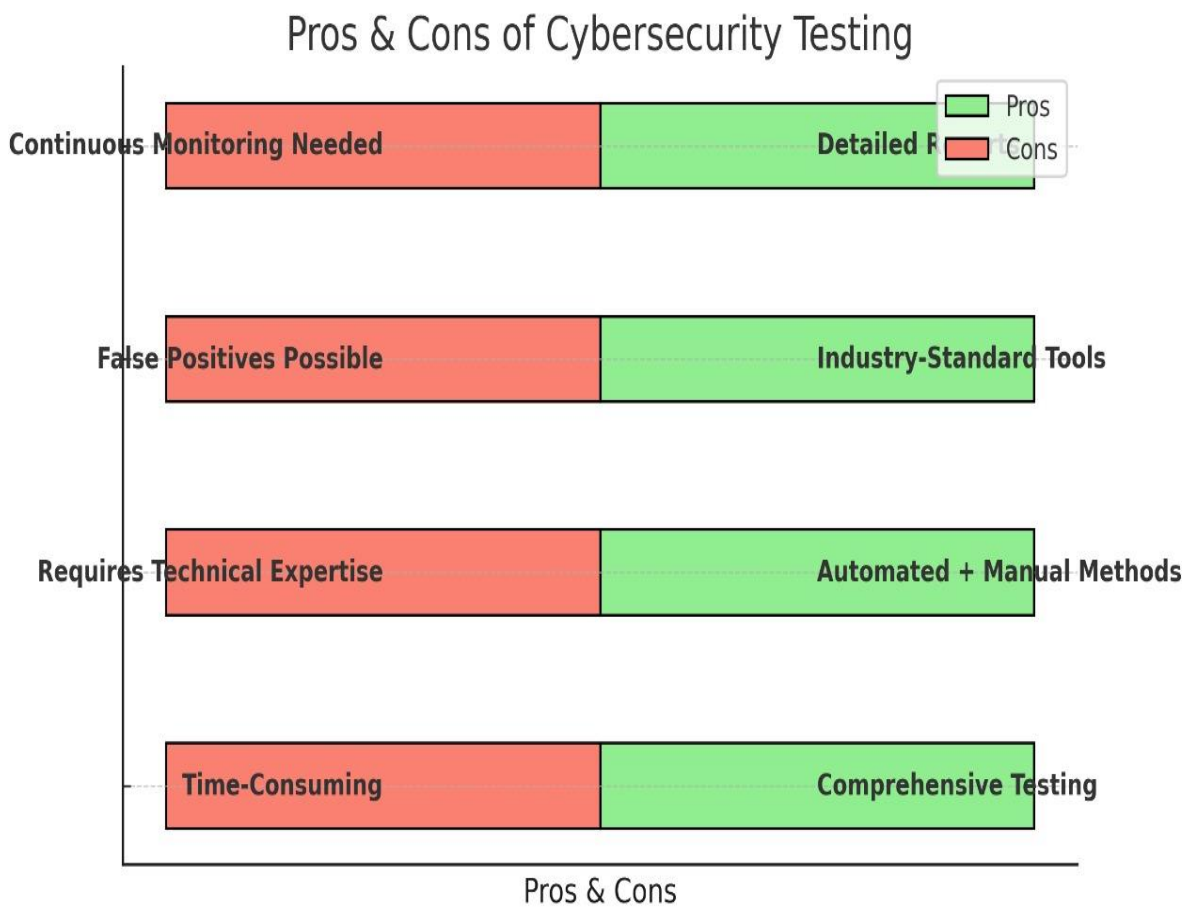
8. ADVANTAGES & DISADVANTAGES

Advantages:

- Comprehensive multi-layered testing combining automated and manual methods.
- Detailed vulnerability analysis with practical remediation recommendations.
- Incorporation of industry-standard tools and cybersecurity frameworks.

Disadvantages:

- Resource-intensive testing process requiring significant manual intervention.
- Some vulnerabilities may require continuous monitoring beyond initial testing.



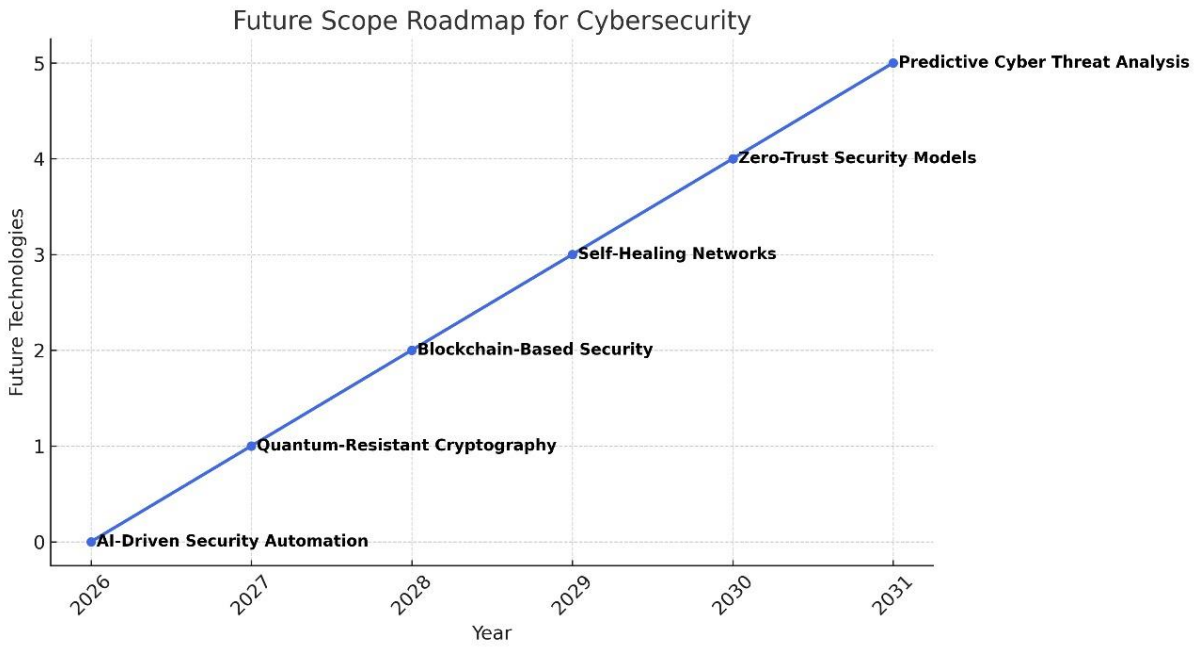
9. CONCLUSION

Our project has successfully demonstrated the importance of a proactive, multi-layered approach to cybersecurity. Through rigorous manual testing and automated scans, we identified and analysed critical vulnerabilities, assessed their business impacts, and recommended robust remediation measures. The findings underscore the necessity for continuous monitoring and improvement in cybersecurity practices to safeguard digital assets.

10. FUTURE SCOPE

- Enhanced Automation:
  Increase the integration of AI-driven tools for continuous vulnerability assessment and real-time threat detection.

- DevOps Integration:
  Embed security measures throughout the Software Development Life Cycle (SDLC) to catch vulnerabilities early.
- Emerging Technologies:
  Investigate quantum-resistant cryptography and blockchain-based security solutions for secure digital identity management.
- Continuous Improvement:
  Develop self-healing networks and dynamic threat intelligence systems to adapt to evolving cyber threats.



11. APPENDIX

GitHub & Project Demo Link:

github link :-- https://github.com/Anujkadam1881/CYBER-SECURITY

video link :--https://github.com/Anujkadam1881/CYBER-SECURITY