1. INTRODUCTION

Date	**10 March 2025**
Team ID	**PNT2025TMID02686**
Project Name	**Project – Exploring Cyber Security Understanding Threats and Solutions in the Digital Age**
Maximum Marks	**8 Marks**

List of teammates :–

S.no	name	college	contact
1	Prathamesh Waghvade	DYP-ATU	luffy2op07@gmail.com
2	Shrinivas Kagwade	DYP-ATU	parthbargale16@gmail.com
3	Parth Bargale	DYP-ATU	shrinivaskagwade@gmail.com
4	Shubham Singh	DYP-ATU	Me.shubhamsingh8998@gmail.com

1.1 PURPOSE

Abstract:

The purpose of this project is to explore cybersecurity threats and solutions in the digital age. It aims to identify cyber risks like malware and phishing, assess security measures such as encryption and firewalls, and promote best practices for digital safety. Additionally, it examines emerging technologies and provides recommendations to enhance cybersecurity.

Scope of the Project :

The scope of this project includes identifying common cyber threats, analysing cybersecurity measures, and exploring emerging technologies like AI and blockchain in security. It covers best practices for individuals and organizations, risk mitigation strategies, and real-world case studies. The project aims to provide insights into strengthening digital security in an increasingly connected world.

Objectives of the Project :

1. **Identify Cyber Threats** – Analyse various cyber risks such as malware, phishing, and ransomware.
2. **Explore Security Solutions** – Study encryption, firewalls, multi-factor authentication, and other protective measures.
3. **Promote Cyber Awareness** – Educate individuals and organizations on best practices for digital safety.
4. **Assess Emerging Technologies** – Examine the role of AI, blockchain, and other innovations in cybersecurity.

2. IDEATION PHASE

2.1 The Thought Behind the Project :

Various Ideas of team members :

Step 1: Various Ideas from Each Group Member

Prathamesh Waghvade

- Analyzing different types of cyber threats (malware, phishing, DDoS, ransomware).
- Studying real-world cyberattacks and their impact.
- Emerging threats in AI and IoT security.

Shrinivas Kagwade

- Role of firewalls, IDS/IPS, and network security tools.
- Implementing multi-factor authentication for enhanced security.
- Encryption techniques for secure data transmission.

Parth Bargale

- Using AI for threat detection and response.
- Developing an AI-based phishing detection system.
- Analyzing the role of behavioral analytics in cybersecurity.

Shubham Singh

- Understanding the role of firewalls in cybersecurity.
- Best practices for secure coding and software development.
- Exploring AI-based solutions for cyber threat detection.

2.2 Problem Statement:

As cyber threats grow more sophisticated, there is a critical gap in understanding and addressing these risks. This project aims to explore emerging threats, identify system vulnerabilities, and propose solutions to enhance cybersecurity for better data and privacy protection.

3. REQUIREMENT ANALYSIS :

Target website - http://www.itsecgames.com/

3.1 List of Vulnerability Table

S.no	Vulnerability Name	CWE - No
1	SQL Injection (SQLi)	89
2	Cross-Site Scripting (XSS)	79
3	Broken Authentication	287
4	Insecure Direct Object References (IDOR)	639
5	Security Misconfiguration	16

REPORT:-

1) Vulnerability Name :- SQL Injection (SQLi)

CWE :- CWE-89 (Improper Neutralization of Special Elements in SQL Commands)

OWASP/SANS Category :- OWASP Top 10 (A03:2021 - Injection) / SANS 25 (#1 - SQL Injection)

Description :- bWAPP contains vulnerable input fields that do not properly sanitize user input, allowing attackers to inject malicious SQL queries. Attackers can retrieve or modify sensitive database information by manipulating SQL statements.

Business Impact :-

- Unauthorized access to user data (passwords, payment details).
- Potential data breach and compliance violations (GDPR, CCPA).
- Database corruption or deletion, leading to service disruption.

2) Vulnerability Name :- Cross-Site Scripting (XSS)

CWE :- CWE-79 (Improper Neutralization of Input During Web Page Generation)

OWASP/SANS Category :- OWASP Top 10 (A07:2021 - Identification and Authentication Failures) / SANS 25 (#2 - XSS)

Description :- bWAPP does not properly sanitize or escape user input, allowing attackers to inject JavaScript into web pages. This enables session hijacking, phishing attacks, and defacement of web pages.

Business Impact :-

- Session hijacking leading to account takeovers.
- Data theft (stealing cookies, personal information).
- Reputation damage if malicious scripts alter website content.

3) Vulnerability Name :- Broken Authentication

CWE :- CWE-287 (Improper Authentication)

OWASP/SANS Category :- OWASP Top 10 (A07:2021 - Identification and Authentication Failures) / SANS 25 (#3 - Broken Authentication)

Description :- Weak authentication mechanisms allow brute-force attacks, credential stuffing, and session hijacking. bWAPP lacks multi-factor authentication (MFA) and uses weak session management, making it vulnerable.

Business Impact :-

- User account compromise, leading to identity theft.
- Privilege escalation, allowing attackers to gain admin access.
- Financial and reputational loss due to unauthorized transactions.

4) Vulnerability Name :- Insecure Direct Object References (IDOR)

CWE :- CWE-639 (Authorization Bypass Through User-Controlled Key)

OWASP/SANS Category :- OWASP A06:2021 (Vulnerable and Outdated Components)

Description :- bWAPP exposes object references in URLs, allowing attackers to manipulate IDs to access unauthorized data. Attackers can view or modify records belonging to other users.

Business Impact :-

- Unauthorized access to confidential user data.
- Data tampering (attackers modifying other users' information).
- Regulatory non-compliance, leading to legal consequences.

5) Vulnerability Name :- CWE-16 (Configuration)

CWE :- OWASP Top 10 (A05:2021 - Security Misconfiguration) / SANS 25 (#6 - Security Misconfiguration)

OWASP/SANS Category :- OWASP A05:2021 (Security Misconfiguration)

Description :- bWAPP has default credentials, exposed admin panels, and unnecessary services running, making it easier for attackers to exploit. Unpatched vulnerabilities and overly verbose error messages further increase risk.

Business Impact :-

- Attackers can gain admin access through exposed admin panels.
- Sensitive information leaks through detailed error messages.
- Increased attack surface, leading to a higher likelihood of successful exploits.

3.2 Technology Stack:

Technology Stack & Tools Explored for the Project:

Technology Stack

Programming Languages:

Python: Used for all backend logic, including vulnerability scanning, web application firewall (WAF), intrusion detection system (IDS), and logging.

Web Framework:

Flask: A lightweight Python web framework used to implement the Web Application Firewall (WAF) that blocks malicious requests (like SQL injection or XSS).

Vulnerability Scanning Tools:

Nessus: A vulnerability scanner used for scanning and identifying security vulnerabilities in network systems and applications. The integration is achieved via the Nessus API.

OpenVAS (Optional): Another vulnerability scanning tool that could be used in place of Nessus.

Intrusion Detection System (IDS):

Snort: A widely-used open-source IDS that is capable of real-time traffic analysis and packet logging. In the provided code, Snort is simulated using Python but can be replaced by a real-world integration.

Python: For simulating IDS functionality in the absence of a full deployment.

Logging and Monitoring:

Elasticsearch: A distributed search and analytics engine used for logging and monitoring the events generated by the application, such as suspicious activity.

Kibana (implicitly): A visualization tool for Elasticsearch that can be used to analyze and visualize logs (not explicitly shown in the code, but part of the ELK stack).

Testing Frameworks:

pytest: A testing framework used for unit and integration testing. It is used to test components like the scanner, WAF, IDS, and logging functionality.

Flask Testing: For testing Flask routes in the WAF application.

Selenium (optionally, if front-end testing is needed): A browser automation tool (not explicitly used in the code but could be integrated for front-end testing if needed).

Version Control:

Git (assumed): For version control, which is generally a standard tool used in most software development projects.

Containerization (optional):

Docker (optional, not in code): You can containerize the application using Docker for easier deployment and isolation of components. This is a common practice in cybersecurity projects.

Web Application Security:

Web Application Firewall (WAF) (implemented in Flask): A layer of security between the client and the web application, checking for malicious requests (like SQL injection, XSS) and blocking them before they reach the app.

Virtualization and Infrastructure (Optional):

Virtual Machines or Cloud services: If deploying the project in a production environment, the components could be hosted on cloud platforms like AWS, Azure, or on-premise virtual machines.

Tools Used in the Project

Nessus API: For vulnerability scanning and getting results of security assessments.

OpenVAS (optional): Another tool for vulnerability scanning.

Flask: Web framework for building the WAF (Web Application Firewall).

Snort: IDS used for monitoring network traffic and detecting malicious behavior.

Elasticsearch: For storing logs and security-related event data.

pytest: For unit and integration testing of the different components (vulnerability scanner, WAF, IDS, logging).

Flask-Testing: For testing the Flask-based WAF application.

requests (Python library): For making HTTP requests, specifically for interacting with the Nessus API and other services.

Python: Primary language used for the entire implementation, including scripting for IDS, scanning, WAF, and logging.

Deployment and Environment Setup

Python Environment:

Python 3.x is used as the runtime environment for this project.

Dependencies are installed via pip from requirements.txt.

Containerization (Optional):

Docker can be used to package each component of the project (Flask WAF, Nessus scanner, IDS, etc.) into containers for consistent and isolated deployments.

Testing:

Tests can be run using the pytest testing framework to validate the functionality of each module (e.g., scanner, WAF, IDS).

Additional Notes:

While this project uses Python-based components for each tool, real-world implementation would involve a proper deployment of each tool (Nessus, Snort, Elasticsearch, etc.) on separate machines or containers.

Logging in Elasticsearch is a key part of the project, which is vital for tracking security events and providing insights.

IDS (Snort) would generally run on a network and monitor real-time traffic for malicious activity. In the current setup, it is simulated through Python for simplicity.

This stack provides a basic cybersecurity infrastructure that can be expanded and enhanced with more advanced features like real-time alerts, continuous vulnerability scanning, machine learning for anomaly detection, and so on.

4. PROJECT DESIGN

4.1 Overview of Nessus :

Nessus is a widely used vulnerability scanner developed by Tenable, designed to identify security weaknesses in systems, networks, and applications. It helps cybersecurity professionals detect vulnerabilities, misconfigurations, and compliance issues before attackers can exploit them.

How Nessus Works :

Target Selection : The user specifies the IP addresses, domains, or systems to scan.

Scanning Process : Nessus runs automated tests to find vulnerabilities, such as:

- Outdated software ○ Open ports
- Misconfigured security settings ○ Weak passwords

Report Generation : After scanning, Nessus provides a detailed report with:

- Vulnerability descriptions
- Severity levels (Critical, High, Medium, Low) ○ Recommended fixes and patches

4.2 Proposed Solution :

Testing and Findings

1. Testing Approach :

To identify vulnerabilities and assess security risks, a Nessus vulnerability scan was conducted on selected systems. The testing process included:

- Defining Scope: Selecting target systems for vulnerability assessment.
- Running Nessus Scan: Conducting network, application, and compliance scans.

- Analysing Results: Reviewing detected vulnerabilities and their severity levels.

2. Findings from Nessus Scan :

The results from the scan highlighted several security weaknesses:

Critical Vulnerabilities :

- Unpatched software with remote code execution risks.
- Weak authentication mechanisms allowing unauthorized access.

High-Risk Vulnerabilities :

- Misconfigured firewall rules exposing unnecessary ports.
- Outdated SSL/TLS protocols leading to encryption weaknesses.

Medium to Low-Risk Issues :

- Open services that could be exploited (e.g., FTP, Telnet).
- Default or weak passwords increasing brute-force attack risks.

3. Proposed Solutions :

- Based on the findings, security measures were recommended: Patching and Updates: Regular software and OS updates to mitigate exploits.
- Firewall and Access Control: Restricting open ports and applying least privilege access.
- Encryption & Authentication: Enforcing multi-factor authentication (MFA) and strong encryption standard.
- Continuous Monitoring: Implementing an Intrusion Detection System (IDS) for real-time threat monitoring.

4.3 Understanding of Exploring Cyber Security :

Cybersecurity has become a critical concern in the digital era, where cyber threats such as malware, ransomware, phishing, and advanced persistent threats (APTs) continue to evolve. To protect digital assets, organizations implement robust security solutions like Security Operations Centres (SOC) and Security Information and Event Management (SIEM) tools to detect, analyse, and respond to threats in real time.

1. Security Operations Center (SOC) :

A Security Operations Center (SOC) is a centralized unit responsible for monitoring, detecting, investigating, and responding to cybersecurity incidents. It functions as the frontline defence against cyber threats by continuously analysing network activity and responding to potential attacks.

2. Security Information and Event Management (SIEM)

SIEM (Security Information and Event Management) is a cybersecurity solution that collects, analyses, and correlates security data from different sources to detect potential threats and ensure compliance.

3. Related Cybersecurity Tools

In addition to SOC and SIEM, various cybersecurity tools enhance digital security:

Intrusion Detection & Prevention Systems (IDS/IPS):

- Snort (open-source IDS/IPS)
- Suricata (real-time threat detection) **Vulnerability Scanners:**
- Nessus (detects security vulnerabilities)
- OpenVAS (open-source vulnerability assessment) **Endpoint**

Detection & Response (EDR):

- CrowdStrike Falcon (AI-driven endpoint security)
- Microsoft Defender ATP (integrated threat protection)

5. PROJECT PLANNING & SCHEDULING :

Objectives :

- Define the scope and timeline of the project.
- Establish a structured approach for execution.
- Ensure timely completion of research, testing, and documentation.

Key Considerations for Execution :

- Resource Allocation – Ensure access to necessary cybersecurity tools and datasets.
- Regular Progress Reviews – Conduct weekly reviews to track milestones.
- Risk Mitigation – Address potential delays in setup and testing through contingency plans.
- Final Evaluation – Verify the effectiveness of security measures before final reporting.

Final Thoughts on Project Planning & Scheduling :

- This structured plan ensures a smooth workflow from research to implementation and reporting.
- Regular monitoring and timely adjustments will help in successful project completion.

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Vulnerability Report (Vulnerability Assessment and Impact) :

1. Introduction to Vulnerability Assessment :

Vulnerability assessment is a critical security process used to identify, evaluate, and prioritize security weaknesses in an organization's IT infrastructure. This assessment helps in mitigating risks before cybercriminals exploit them.

Assessment Type : Network Security, Web Application Security, Compliance Audit

2. Impact Analysis :

- **Business Impact:** Data breaches, financial losses, reputational damage.
- **Technical Impact:** System downtime, unauthorized data access, malware infections.
- **Compliance Risk:** Non-compliance with security standards like **ISO 27001, PCI-DSS, GDPR**.

7. RESULTS

7.1 Findings and Reports (Nessus & SOC Analysis) :

1. Nessus Vulnerability Scan Findings :

The Nessus vulnerability scan revealed multiple security risks across network devices, web applications, and endpoints. Critical vulnerabilities included unpatched OS and software, which could allow remote code execution and malware infections. Weak authentication methods, such as missing multifactor authentication (MFA), posed a high risk of unauthorized access. Open ports and services increased exposure to potential exploitation, while SQL injection (SQLi) vulnerabilities threatened data security. Medium-severity issues included outdated SSL/TLS encryption, cross-site scripting (XSS), and misconfigured security policies, which expanded the attack surface.

2. SOC Analysis Findings :

The Security Operations Center (SOC) detected several security incidents through continuous monitoring. Critical threats included multiple unauthorized login attempts, indicating possible brute-force attacks, and suspicious data transfers, suggesting potential data exfiltration. A phishing email campaign targeted employees, attempting credential theft. Ransomware-like activity was observed on endpoint devices, with unusual file encryption patterns detected. Medium-level incidents included a distributed denial-of-service (DDoS) attack causing temporary downtime and abnormal user behavior indicating potential insider threats or compromised accounts.

3. Security Impact Analysis :

- The findings from Nessus and SOC analysis highlighted major risks, including:
- Business disruptions due to ransomware attacks and service downtime.
- Data breaches resulting from unpatched vulnerabilities and weak authentication mechanisms.
- Regulatory compliance risks, with potential violations of security standards such as ISO 27001, GDPR, and PCI-DSS.

4. Recommendations :

To mitigate these risks, the following security measures are recommended:

- Regular system patching and updates to eliminate critical vulnerabilities.
- Implementing multi-factor authentication (MFA) to strengthen access controls.
- Deploying advanced threat detection tools, such as intrusion detection and prevention systems (IDS/IPS).
- Enhancing security awareness training to educate employees on phishing and cyber threats.
- Strengthening the incident response plan (IRP) to improve SOC capabilities in detecting and mitigating security incidents.

8. ADVANTAGES & DISADVANTAGES

Advantages (Pros) :

✓ Proactive Threat Detection: Identifies risks before hackers exploit them.

✓ Automated Monitoring: SOC and SIEM provide real-time alerts.

✓ Comprehensive Security Assessment: Nessus scans a wide range of security threats.

Disadvantages (Cons):

✕ High Costs: Setting up SOC, SIEM, and security tools requires investment.

✕ Complexity: Requires trained cybersecurity professionals.

✕ Performance Impact: Scanning can slow down systems.

9. CONCLUSION

This project, "Exploring Cyber Security: This project studied cybersecurity threats and solutions using tools like Nessus and SOC monitoring. It highlighted the importance of:

1. Understanding Cyber Threats & Security Measures :

Malware, phishing, ransomware, and hacking. Testing for Vulnerabilities: Nessus scan found weak authentication and unpatched software.

2. Vulnerability Assessment & Nessus Findings :

- The Nessus scan detected critical vulnerabilities such as unpatched software, weak authentication, misconfigured firewalls, and outdated SSL/TLS protocols.

3. SOC & SIEM Security Monitoring Analysis :

- SOC detected phishing attacks and unauthorized logins.
- Preventive Measures: Implementing IDS, MFA, and regular system updates.

10. FUTURE SCOPE

Future Scope for Testing and Deployment :

1) **AI-Based Threat Detection:** Using AI to detect cyber threats more efficiently.

2) **Automated Penetration Testing**: Simulating real-world cyberattacks with Metasploit.

3) Improving Security Systems

4) **Self-Healing Systems:** AI-driven patch management for automatic updates.

5) **DevSecOps Integration:** Embedding security into software development.
Research & Innovation

6) **Quantum-Safe Cryptography:** Encryption methods to resist quantum computing attacks.

7) **Blockchain for Security:** Using blockchain for safer identity management.