Date	12 February 2025
Team ID	1.15
Project Name	Exploring Cyber Security Understanding Threats and Solution in the Digital Age
Maximum Marks	8 Marks

List of teammates–

S.no	name	collage	contact
1	Priti Shivaji Chavan-Patil	DYP Agriculture and Technical University Talsande	7276029080
2	Pratibha Mahadev Chavan	DYP Agriculture and Technical University Talsande	8767651712
3	Rajani Jayraj Salunkhe	DYP Agriculture and Technical University Talsande	9322461040
4	Anuja Bhikaji Chavan	DYP Agriculture and Technical University Talsande	8767348641

3.Requirement Analysis:

1. Scope:

- Identify and analyze cyber threats (phishing, malware, ransomware, DDoS, etc.).
- Implement detection mechanisms (AI-based, anomaly detection, IDS).
- Propose mitigation strategies (firewalls, encryption, authentication).
- Provide real-time monitoring and user awareness.

2. Functional Requirements:

- Secure login & role-based access control.
- Threat database & real-time analysis.
- Security alerts for suspicious activities.
- Reporting & dashboard for insights.

3.Enhancements (Optional):

- Blockchain for secure logging. • Honeypots to analyze attacks.
- Ethical hacking tools for penetration testing.

3.1 Customer Journey Map:

For your cybersecurity exploration, integrating a Customer Journey Map (CJM) can help identify vulnerabilities at different touchpoints in a user's interaction with a system. Here's how you can approach it:

1. Define the User Persona

- Who are the users? (e.g., general users, IT admins, cyber professionals)
- What are their goals related to security? (e.g., secure login, data protection)

2. Identify Key Stages of Interaction

- Awareness: Users become aware of cybersecurity risks.
- Consideration: They explore security tools or best practices.
- Onboarding: Setting up security measures (e.g., MFA, encryption).
- Usage: Daily interactions with secure systems.
- Incident Handling: Responding to threats like phishing or malware.

- Feedback & Improvement: Learning from security breaches or updates.

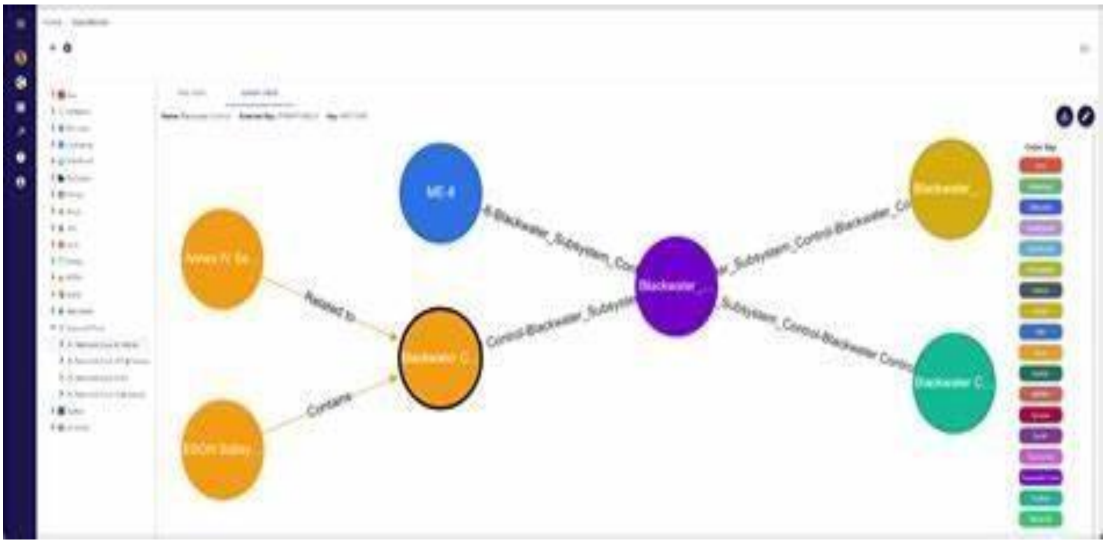
3. Identify Threats at Each Stage

- Awareness: Misinformation about cybersecurity.
- Consideration: Fake security solutions, scams.
- Onboarding: Weak password setup, insecure configurations.
- Usage: Phishing, malware, unauthorized access.
- Incident Handling: Delayed responses, lack of training.
- Feedback: Neglecting security patches or user concerns.

4. Define Solutions

- Education: Security awareness training.
- Best Practices: Strong authentication, encryption.
- Detection & Response: Threat monitoring, incident response plans.
- Continuous Improvement: Regular security audits.

3.2 Data Flow Diagram :


ME-4
Annex IV Se
Related to
EDOH Safety
Contains
Blackwater C
Control-Blackwater_Subsystem
Blackwater_Subsystem_Control
Blackwater_
Blackwater_Subsystem_Control-Blackwater_C
Subsystem_Control-Blackwater Contro
Blackwater_
Blackwater C

3.3 Solution Requirement:

When analyzing solution requirements in cybersecurity, you need to define how a system should address threats effectively. Here’s a structured approach:

1**. Functional Requirements (What the System Should Do)**

User Authentication & Access Control

- Implement multi-factor authentication (MFA)
- Role-based access control (RBAC)

Threat Detection & Prevention

- Real-time intrusion detection system (IDS)
- AI-based anomaly detection

Data Protection

- End-to-end encryption for sensitive data
- Secure backup & recovery mechanisms

Incident Response & Logging

- Automated threat alerts & reporting
- Detailed security logs & audit trails

2. Non-Functional Requirements (Performance, Security, Compliance)

Security & Compliance

- Adhere to GDPR, ISO 27001, NIST frameworks
- Regular security updates & patches

Performance & Scalability

- Low-latency real-time monitoring
- Scalable architecture for handling large traffic

Usability & Accessibility

- User-friendly security dashboards
- Support for different devices (mobile, web)

3.4 Technology Stack:

When analyzing the technology stack for a cybersecurity-focused system, you need to consider tools and frameworks for security, monitoring, and response. Here's a structured breakdown:

1. Frontend (User Interface & Security Features)

- Frameworks: React.js, Angular, Vue.js (with security best practices)

- Security Enhancements: Content Security Policy (CSP), HTTPS enforcement, Input validation
- Authentication: OAuth 2.0, OpenID Connect, JWT

2. Backend (Processing & Security Layers)

- Languages: Python (Django, Flask), Node.js, Java (Spring Boot), Golang
- Security Measures: Secure API Gateway, Encryption (AES, RSA), Secure Coding Practices
- Authentication & Authorization: OAuth, LDAP, SAML, MFA

3. Database (Secure Data Storage & Encryption)

- SQL: PostgreSQL, MySQL (with encryption at rest)
- NoSQL: MongoDB, Firebase (for unstructured data)
- Security: Database encryption, Access control, Regular backups

4. Security & Threat Monitoring

- Intrusion Detection Systems (IDS/IPS): Snort, Suricata
- SIEM (Security Information & Event Management): Splunk, ELK Stack • Vulnerability Scanning: Nessus, OpenVAS

5. DevSecOps & Cloud Security

- CI/CD Security Tools: Snyk, SonarQube
- Cloud Security: AWS Security Hub, Azure Defender, Google Cloud Security Command Center
- Container Security: Docker Bench for Security, Kubernetes RBAC

6. Incident Response & Logging

- Log Management: Graylog, ELK Stack
- Incident Response Platforms: TheHive, MISP