ORLOX

internship assignment

Automatic M365 Cybersecurity Tenant Assessment

CONTACT

Kristof Laerenbergh

Veldkant 7, 2550 Kontich
info@orlox.be
www.orlox.be

TABLE OF CONTENTS

Description **3**

Deliverables **3**

Possible Extensions **3**

Project Methodology **4**

Description

The rapid proliferation of cloud-based solutions like Microsoft's M365 suite requires robust cybersecurity measures. This internship project focuses on the development of an APP to automatically assess the cybersecurity posture of an M365 tenant with Microsoft. The goal is to highlight potential vulnerabilities and provide actionable insights to bolster security.

In order to deliver this project, you will initially set up an M365 environment, making it vulnerable intentionally. Following this, the APP will be developed and tested against this environment, seeking to identify the vulnerabilities you set.

Deliverables

1. Set up an M365 environment with intentional vulnerabilities.
2. Design and develop the APP, based on Python code to automatically assess the cybersecurity posture of the M365 tenant, based on our current M365 cybersecurity assessment.
3. Implement the assessment APP on the test environment.
4. Analyze results and identify areas for enhancement in the APP.
5. Optimize the APP based on findings.
6. Compare results from the initial assessment and post-optimization assessment to measure APP effectiveness.

Possible Extensions

1. Create a multi-tenant security reporting dashboard. Develop an alert system within the APP to notify administrators of critical vulnerabilities in real-time.

Project Methodology

Secure Application Development using CIS, NIST, and Microsoft Cybersecurity Frameworks

1. **Framework Integration:**
   - Integrate the key principles from the CIS, NIST, and Microsoft Cybersecurity Frameworks to define the foundational guidelines for the project.
   - Prioritize framework recommendations based on the criticality of application functionalities and the risk associated with vulnerabilities.

2. **Requirement Analysis and Threat Modeling:**
   - Use the NIST Cybersecurity Framework to identify, assess, and prioritize cybersecurity requirements tailored to the application's functionality and usage.
   - Leverage Microsoft's threat modeling techniques to identify potential vulnerabilities in the design phase.

3. **Secure Development Environment:**
   - Establish a secure development environment in alignment with CIS best practices, ensuring that all tools, systems, and platforms used in development are regularly updated and free from vulnerabilities.

4. **Secure Coding Practices:**
   - Adopt secure coding practices, leveraging guidelines from all three frameworks to prevent common vulnerabilities like SQL injection, cross-site scripting, etc.
   - Conduct regular code reviews focusing on security, using tools that adhere to the principles of the integrated framework.

5. **Testing & Validation:**
   - Utilize the NIST framework for a rigorous testing regimen to validate application security. Ensure that testing includes penetration testing, vulnerability assessments, and source code analysis.

6. **Feedback & Iteration:**
   - Actively seek feedback during development and testing phases, using insights to iterate and enhance security postures.
   - Ensure any changes to the code are re-tested and reviewed for security implications.

7. **Deployment & Maintenance:**
   - Ensure secure deployment practices, considering guidelines from the CIS framework for cloud or on-premises deployments.
   - Continuously monitor the application in the production environment. Use the Microsoft Cybersecurity Framework to establish alerting mechanisms for potential threats.
   - Periodically review and update the application based on the evolving recommendations from the CIS, NIST, and Microsoft Cybersecurity Frameworks.