**Client Overview**

Artemis Financial is a financial advisory firm specializing in personalized investment strategies and wealth management solutions. Their web application processes sensitive client data, including financial records and personal identification details. The company engaged me to conduct a comprehensive security assessment of their software to identify vulnerabilities and recommend mitigation strategies to protect against potential cyber threats.

**Strengths in Security Assessment**

During the vulnerability assessment, I excelled in identifying critical security flaws using both automated tools (such as OWASP Dependency-Check) and manual code review. Secure coding is essential because financial applications are prime targets for cyberattacks—exploits like SQL injection or insecure dependencies could lead to data breaches, regulatory penalties, and loss of customer trust. Implementing strong security measures enhances a company’s reliability, safeguards sensitive data, and ensures compliance with industry standards like PCI-DSS and GDPR.

**Challenges & Key Takeaways**

The most challenging aspect was distinguishing between theoretical vulnerabilities and those that posed real-world risks to Artemis Financial’s specific application. For example, some flagged dependencies were not actually used in production, while others required immediate patching. This experience helped me refine my ability to prioritize risks based on exploitability and business impact.

**Enhancing Security Layers**

To strengthen security, I implemented multiple defenses, including:

* **HTTPS encryption** to secure data in transit
* **Input validation** to prevent injection attacks
* **Dependency updates** to eliminate known vulnerabilities
* **Secure API authentication** using OAuth 2.0

In future assessments, I would integrate **threat modeling** (using frameworks like STRIDE) and **penetration testing** to evaluate vulnerabilities more holistically before deciding on mitigation techniques.

**Ensuring Functionality & Security**

After refactoring the code, I:

1. Ran **unit and integration tests** to confirm core functionality.
2. Re-executed **dependency scans** to check for new vulnerabilities.
3. Conducted **manual penetration tests** on critical user flows.
4. Verified security headers and encryption protocols.

This multi-layered verification ensured that fixes didn’t introduce new weaknesses.

**Valuable Tools & Practices**

Key resources that aided this project:

* **OWASP ZAP** for dynamic security testing
* **SonarQube** for static code analysis
* **CVE databases** to research vulnerabilities
* **Secure coding checklists** from CERT

These tools and methodologies will be instrumental in future security-focused projects.

**Showcasing to Employers**

From this assignment, I would highlight:

* The **Vulnerability Assessment Report**, demonstrating my ability to analyze risks systematically.
* **Code snippets** showing before/after security improvements.
* **Testing documentation** proving secure development practices.

This work exemplifies my skills in cybersecurity, risk assessment, and secure software development—qualities highly sought after in tech roles.