Implementing and Explaining Advanced Cybersecurity Defense Strategies

In this project, I designed and implemented a secure architecture for a web-based document management system used by a mid-sized financial services firm. The system was responsible for handling sensitive financial records, contracts, and internal memos, requiring a robust security approach across multiple layers.

1. Zero Trust Architecture (ZTA)

To ensure strict access control and minimize trust assumptions, I applied **Zero Trust Architecture (ZTA)** at both the **network layer** and the **application layer**.

Network Layer – Identity-Based Microsegmentation

I implemented **network microsegmentation** using software-defined networking. This approach isolated the network into multiple zones based on role-specific needs: **Finance**, **Legal**, **HR**, and **IT Ops**. Access to each zone was strictly controlled by identity-based policies enforced through an **Identity Provider (IdP)** using multi-factor authentication (MFA). For example, an HR employee connected to the corporate VPN would be restricted from accessing Finance servers unless explicitly granted permission. **Lateral movement** across zones was entirely blocked by default.

Application Layer – Role-Based Access Control (RBAC) with Just-In-Time Provisioning

At the **application level**, I implemented **OAuth 2.0** with **Role-Based Access Control (RBAC)**. Different roles (e.g., "Finance Analyst," "Legal Reviewer," "Admin") were assigned minimal permissions necessary for their work. I also enforced **just-in-time (JIT)** access provisioning using **Azure AD Privileged Identity Management (PIM)**. For instance, an Admin needing elevated access to approve sensitive documents would request temporary rights that expired automatically after 15 minutes, ensuring minimal access duration. By combining **identity-first segmentation** at the network level and **role-based access** at the application layer, I ensured that access was continually verified and never assumed.

2. Defense in Depth

To protect the system from various attack vectors, I implemented a **Defense in Depth** strategy, incorporating layered security controls:

Layer 1: Endpoint Security

I enrolled all user devices in **Microsoft Intune**, enforcing strict compliance policies before allowing network connections. Each device was equipped with:

- **Real-time threat protection**
- **Application whitelisting**
- **Full-disk encryption (BitLocker)**
- **USB access restrictions**

Daily monitoring ensured that operating system patches were up-to-date, and

devices without compliance were blocked from connecting to the network.

Layer 2: Identity and Access Management (IAM)

I integrated a **centralized IAM solution** with MFA, **conditional access** policies, and **behavioral risk scoring**. Any login attempt from an **unusual IP address** or **unrecognized device** would trigger an additional authentication challenge or block access entirely, reducing the risk of unauthorized access.

Layer 3: Application and Data Security

To secure data within the document system, I implemented **Data Loss Prevention (DLP)** policies. For example, any attempt to email a document labeled as "Confidential" externally would be automatically blocked and reported to the **InfoSec team**. Additionally, all data was encrypted in both **transit (using TLS 1.3)** and **at rest (using AES-256)**, ensuring that sensitive information remained protected at all stages.

This multi-layered defense approach offered comprehensive protection, ensuring that even if one control failed, others would mitigate the risk.

3. Supply Chain Security: Risk Identification and Mitigation

During the design phase, I conducted a **Software Composition Analysis (SCA)** to assess third-party components used within the system. One of the **backend services** relied on a package (**pdfgen-core v2.8**) that had a known **Remote Code Execution (RCE)** vulnerability (**CVE-2024-0112**).

To mitigate this risk, I took the following actions:

- Replaced the vulnerable library with a **forked, patched version**.
- Documented the issue and resolution in the **Software Bill of Materials (SBOM)**.
- Set up **continuous dependency scanning** in our **CI/CD pipeline** using **GitHub Dependabot**, which alerted us to any new vulnerabilities in third-party components.

This proactive approach ensured that our system was not exposed to known threats from the supply chain.

4. Application of Advanced Security Model: Bell-LaPadula (BLP) Model

To maintain **data confidentiality**, I applied the **Bell-LaPadula (BLP)** security model to control access to documents within the system. I created three **clearance levels**: **Top Secret**, **Confidential**, and **Public**, which were assigned to both users and documents.

Using BLP's core principles:

- **Simple Security Property (No Read Up):** Users could not access documents at a higher classification than their own clearance. For example, a Legal Reviewer with **Confidential** clearance could not access **Top Secret** documents.
- ***-Property (No Write Down):** Users with higher clearance could not

upload documents to lower-level classified folders. For instance, a Finance Admin with **Top Secret** clearance could not accidentally upload sensitive documents to the **Public** folder.

This enforced strict confidentiality rules, ensuring that users only had access to data based on their classification level and preventing accidental data leaks.

Conclusion

This project successfully integrated **Zero Trust Architecture** across multiple security layers, adopted a **Defense in Depth** approach to safeguard systems, mitigated third-party risks via proactive supply chain security practices, and applied the **Bell-LaPadula** model for strong data confidentiality controls. Through these measures, I ensured that the web-based document management system was secure and capable of handling sensitive financial data with robust protection from a wide range of threats.