**VULNERABILITY MANAGEMENT**

A **Software Vulnerability** is a flaw or weakness in an application's code, design, or configuration that can be exploited by an attacker to gain unauthorized access, steal data, or disrupt service.

|  |  |  |
| --- | --- | --- |
| **Potential Impact** | **Description** | **Mitigation Strategy** |
| **Service Outage** | Exploiting flaws (e.g., Buffer Overflows) to crash a critical application like the Broker Portal during AEP. | Proactive patching, **Monitoring SLOs**, and **Error Budget** enforcement. |
| **Data Breach** | Exploiting Injection Flaws (e.g., SQL Injection) to steal sensitive Protected Health Information (PHI) or customer data. | **Protegrity Tokenization** and mandatory **TLS Re-encryption** via OpenShift Routes. |
| **Supply Chain Attack** | Exploiting vulnerabilities in third-party libraries or base operating system images (like Log4J). | **Golden Image** reliance and **Wiz Scanning** integrated into the CI/CD pipeline. |

**2. Our Mitigation Strategy: The Shift-Left Approach**

We mitigate risk by integrating security checks early (Shift-Left) and managing base infrastructure security centrally.

|  |  |  |
| --- | --- | --- |
| **Mitigation Layer** | **Tool / Process** | **SRE Responsibility** |
| **Code Scanning** | **SonarQube** | Enforces code quality checks on application source code (Maven build phase). |
| **Container Scanning** | **Wiz** | Scans the final application image for known CVEs before deployment to Quay. |
| **Base Image Security** | **Golden Images / AMI Rotation** | We build applications on pre-vetted, hardened base images. **Container Management Team** is responsible for patching and ensuring all base images are rotated every **90 days (3 months)**. |
| **Runtime Protection** | **OpenShift (ROSA)** | Manages Pod health, networking, and enforces security context constraints (SCCs). |

**3. Vulnerability Triage Protocol & Team Ownership**

All vulnerability findings are sourced from the **Wiz Report** and are automatically categorized for assignment based on where the fault lies (code vs. infrastructure) and the asset's function.

**A. Triage Assignment Logic (The "Where")**

The assignment logic relies on filtering columns within the raw Wiz Excel/CSV report (e.g., LocationPath, AssetName).

|  |  |  |
| --- | --- | --- |
| **Category** | **Ownership Logic** | **Responsible Team** |
| **DB Assets (Prod/Non-Prod)** | AssetName contains keywords like db, mongo, or data. | **Database Team** (DBAs) |
| **CI/CD/App - Dev Dependencies** | AssetName does **NOT** contain DB keywords AND LocationPath contains .m2 or xml | **Development Team** (Application Owners) |
| **CI/CD/App - Base Image OS** | AssetName does **NOT** contain DB keywords AND LocationPath is outside the /app directory (e.g., /usr/lib, /var) | **Container Management Team** (Base Image/OS) |
| **CI/CD/App - SRE Tooling** | AssetName contains keywords related to bamboo agents or OpenShift infra, or general tooling (/opt/). | **DevOps/SRE Team** (Platform Owners) |

**B. SLA Matrix: Time to Remediate (TTR)**

The severity column in the Wiz Report dictates the maximum allowed time (TTR) before the vulnerability must be fixed and redeployed to production, as per regulatory and internal policy requirements.

|  |  |  |
| --- | --- | --- |
| **Severity (Column I)** | **Target Time to Remediate (TTR)** | **SRE Compliance Responsibility** |
| **Critical** | **30 Days** | Ensure priority assignment and necessary resources are allocated. |
| **High** | **60 Days** | Monitor progress to prevent escalation to Critical. |
| **Medium** | **189 Days** | Schedule remediation during regular patch cycles. |
| **Low** | **360 Days** | May be accepted, pending official risk review. |