CASE STUDY

RADAPEX
TECH & DATA SOLUTIONS

RADAPEX Securing REST APIs infrastructure using APEX & ORDS

RADAPEXs cyber security team analysed a platform used by enterprises that rapidly deploy Oracle APEX applications and REST APIs. The system supports mission-critical apps for multiple clients, making data integrity, availability, and security top priorities.

Over time, the environment had accumulated technical debt, particularly around its REST API stack, which relied on Oracle APEX, Oracle REST Data Services (ORDS), and Apache Tomcat. Security scans and penetration tests revealed that some components were running outdated versions, and several APIs were accessible only via HTTP, exposing vulnerabilities.

The Challenge

Outdated Software Versions

- APEX, ORDS, and Tomcat versions were several releases behind, lacking recent security patches.
- Legacy configurations prevented quick upgrades, risking additional downtime.

Unencrypted API Endpoints

- Some REST endpoints were only accessible over HTTP, risking interception of sensitive data.

Operational Risks

- Production downtime had to be minimized since RADAPEX supported real-time client operations.
- Dependencies between APEX, ORDS, Tomcat, and the underlying Oracle Database meant upgrades had to be carefully sequenced.

Compliance Pressure

- Industry security standards (e.g. OWASP, NCSC) required encrypted communications and up-to-date software versions.

Objectives

- Upgrade APEX, ORDS, and Tomcat to the latest stable, supported versions.
- Ensure all REST APIs are accessible only via HTTPS.

CYBER SECURITY

“RADAPEX identified that we had failed on 8 of the top 10 OWASP vulnerabilities, RADAPEX not only identified these issues, but derived a clear implementation plan to ensure we remain safe and secure”

ORACLE | Partner

- Apply secure configuration best practices to mitigate potential vulnerabilities.
- Minimise downtime and ensure a smooth upgrade path for the future with an agreed patching schedule.

RADAPEX Approach

1. Assessment and Planning

- Conducted a full inventory of the current APEX, ORDS, and Tomcat versions.
- Mapped application dependencies and identified high-risk APIs.
- Scheduled maintenance windows with clients to avoid service disruptions.
- Verified Oracle Database compatibility with target APEX/ORDS versions.

2. Environment Preparation

- Set up a staging environment mirroring production.
- Backed up APEX applications, ORDS configuration, and Tomcat settings.
- Downloaded and verified checksum of the latest versions:
  - **APEX**: Latest release from Oracle's support portal.
  - **ORDS**: Updated release supporting new APEX features and enhanced REST security.
  - **Tomcat**: Latest stable Apache Tomcat version with TLS support.

3. Upgrade Execution

- **APEX Upgrade**:
  - Deployed the new APEX version to the Oracle database schema.
  - Migrated and validated APEX applications in staging before production cutover.
- **ORDS Upgrade**:
  - Updated ORDS to match the new APEX version compatibility matrix.
  - Migrated ORDS configuration, enabling force-HTTPS and disabling unsecured HTTP ports.
- **Tomcat Upgrade**:
  - Installed the latest Tomcat version.
  - Applied secure server.xml configuration, enforcing latest TLS.
  - Disabled weak ciphers and enabled HTTP Strict Transport Security (HSTS).

4. Security Hardening

Implemented firewall rules to block direct HTTP traffic.

- Configured ORDS to listen only on HTTPS.
- Added Encrypt SSL certificates.

- Verified compliance with OWASP Top 10 API Security recommendations.

5. Testing and Validation

- Conducted end-to-end regression tests in staging.
- Performed vulnerability scans to confirm no open HTTP endpoints.
- Validated all API calls worked with HTTPS and correct authentication.

6. Production Deployment

- Scheduled cutover during low-traffic hours.
- Performed database and configuration backups.
- Deployed upgrades in sequence: **Tomcat - ORDS - APEX**.
- Monitored logs and API response metrics in real time.

Results

- **Security**: 100% of REST APIs now enforce HTTPS with modern TLS protocols.
- **Patching**: All components (APEX, ORDS, Tomcat) are running the latest supported versions with critical security fixes applied.
- **Performance**: Response times improved by 20% due to Tomcat optimisations and ORDS enhancements.
- **Compliance**: Achieved audit readiness for upcoming security compliance checks.
- **Downtime**: Production downtime was limited to 25 minutes, well below the 1-hour maintenance window.

Key Takeaways

- **Planned Sequencing**: Coordinating the upgrade order of Tomcat ensured compatibility and minimised rollback risk.
- **Staging First**: Mirroring production in a test environment avoided surprises during the live upgrade.
- **HTTPS Enforcement**: Security is not just about encryption, forcing HTTPS, removing HTTP, and disabling weak ciphers closed attack vectors.
- **Continuous Maintenance**: Regular patch cycles prevent the accumulation of risky technical debt.

About RADAPEX

RADAPEX simplifies processes, strengthens evidence based decision making and delivers measurable efficiencies. With over a decade of experience, we integrate advanced technology with the unique priorities of the sector, enabling more effective, accountable, and resilient outcomes.

Visit radapex.com