Security fact sheet

The following covers an outline of how the revolutioniseSPORT platform handles security and risk.

32/11-21 Underwood Road
Homebush NSW 2140

www.sportsgrid.com.au

- **Vulnerability Reporting and Management:** We have an easily discoverable way for external researchers to report security vulnerabilities in our systems, and these are reviewed in a timeline and triaged way.
- **HTTPS and mixed content:**
    - The web application is reachable exclusively over HTTPS. Even if the user manually edits the URL to start with http://, it won't work or it will redirect to https://.
    - SSL/TLS private keys are appropriately protected on our web servers.
    - We have specific controls in place to prevent mixed-content issues.
    - Authentication cookies marked with the secure attribute
    - The HttpOnly keyword is set for all our authentication cookies.
- **Authentication, Authorisation and Recovery:**
    - The application enforces minimum password security requirements (e.g., a certain length, character classes, etc.)
    - Passwords are stored using a dedicated password-based key derivation function, such as bcrypt, PBKDF2 or scrypt
    - We deal with users who can no longer access their accounts via a password reset link being sent via email to the user's registered email address.
    - We have both **horizontal** and **vertical** access control
        - Horizontal access control refers to isolation between users of the same role. For example, consider an application that allows users to access their payroll statements. The application must ensure that a user cannot access another user's statements; i.e., if the user's statement for the month of May is found at statement.html?id=837**2**&month=5, it shouldn't be possible to see someone else's pay stub simply by loading statement.html?id=837**3**&month=5
        - Vertical access control refers to when an application supports multiple roles, users should not be able to gain privileges or perform unauthorized actions by loading pages or features that should only be available to users in a different role.

- **Authorization-Related Web Vulnerabilities:**
    - We protect all state-changing actions against cross-site request forgery (XSRF)
    - We protect against Cross-Site Script Inclusion (XSSI)
    - We protect against Clickjacking
    - We use an object-relational mapping (ORM) framework to protect against SQL injection
- **File storage:**
    - File uploads are stored on Amazon Web Services in Sydney
    - These are tokenised to prevent file enumeration
    - File type restrictions are in place
- **Data storage:**
    - Client data is stored on Amazon Web Services in Sydney
    - Data is encrypted at rest
- **Testing, QA, and Monitoring**
    - Security testing is part of standard application tests.
    - **Simple unit tests:** Unit tests are typically used to confirm that the basic building blocks of the application work as expected. Unit tests are easy to repeat — they can run whenever new code is checked into the repository, to confirm that the code still behaves as expected. Unit tests can also check for security features. For example, they can be used to confirm that requests fail without XSRF tokens; that authentication is required to access user data; or that unexpected HTML tags can't get through input filters or escaping routines.
    - **Release testing:** Before a new version of a product is released, human testers typically go through the application, try the new features, and make sure previous features still work correctly (regression testing). Security testing should be included in this process as well. For example, release testing is a great time to verify that user A cannot access the data of user B.
    - **Monitoring:** Once the application is deployed, the focus usually shifts from testing to monitoring. Watch out for unexpected spikes in error rates, sandbox violations, and other flaky or inexplicable behavior (including intermittent test failures) — and before you dismiss an anomaly, check with your security team. Crashes and flakiness can indicate a race condition or a memory corruption bug.

32/11-21 Underwood Road
Homebush NSW 2140

www.sportsgrid.com.au

- **Server hardening**
    - Servers are hardened, including such methods as:
        - Removing or disabling all non-essential services
        - Disabling all default accounts and/or changing the password
        - Review/tightening of permissions for key files and directories
        - Setting up logging
        - Securely configuring remote access and management interfaces
        - Setting up appropriate host-based firewall rules
        - Securely configuring a backup mechanism
    - We have comprehensive logging, including security events, for all relevant services.
    - All devices that support logging write logs to one or more dedicated log systems.
- **Backups**
    - If a disaster occurs at one site, very little data will be lost because almost everything will already have been copied to our backup site.
    - We have a fixed backup cycle
    - We test the entire process of recovery, including restoring entire systems from backup.
- **Physical office**
    - Servers are not located in our physical office and are located in Amazon Web Services data centres located in Sydney
    - That being said, we have an auditable process in place for granting and revoking physical access to office facilities
    - Various security methods are in place (e.g. Security cameras, remote security monitoring, alarms, etc)
    - Only a few IT employees have physical access to networking equipment.
    - Guest wifi is segmented from office wifi
- **PCI DSS**
    - PCI DSS is covered by upstream payment providers, and we do not stored credit card information on our servers.