Product Lifecycle Security Policy

1.1 Make security an inherent part of information systems

1.1.1 Consider security when changing or acquiring systems

Policy

Xyris will require that any development of services that communicate with other services will require at a minimum, the use of TLS 1.2 encryption.
The use of any new Information Systems or Application, will require that the version identified requires communications using a minimum of TLS 1.2
Xyris will conduct a security check before any purchases of new systems that identifies any current vulnerabilities for that version.

Guidelines

- Derive system requirements for information systems security
- Derive from policies and regulations security requirements
- Requirements for incident reviews
- Requirements for monitoring processes
- Consider the needs to authorize access for privileged users
- Derive requirements for security awareness process
- Considerations are taken into account for what potential problems poor security could cause and the negative business impact
- Review security information security requirements

1.1.2 Protect application services on all public networks

Policy

Xyris applications will be accessing the resources via the internet over public networks. Xyris will protect its applications from fraudulent activities, unauthorised disclosure and modifications.

Guidelines

- control the use of application services on public networks
- consider the need to safeguard all application servers
- consider the need to protect network interconnections

1.2 Protect and control system development activities

1.2.1 Establish rules to control internal software development

Policy

- established rules to control how software and systems are developed
- apply internal software development rules

Guidelines

- Establishment of secure development practices
- Maintain a secure software development environment
- Establish development practices to find and fix vulnerabilities
- Establish secure software development methodology
- Establish secure repositories for development projects
- Continual secure application knowledge
- Use secure coding standards for all developments

1.2.2 Use formal procedures to control changes to systems

Policy

All proposed changes to Xyris' network devices, systems, application software deployment and configurations must follow a formal change control process. The change control process shall include:

- Identification, review, and documentation of changes.
- Planning and testing requirements.
- An assessment process for potential operational and security impacts.
- Procedures to validate that changes yield the anticipated results.
- Procedures to roll-back or update the change if the change yield un-anticipated results.
- Formal approval and sign-off requirements for proposed changes.
- Notifications requirements to ensure all affected parties are promptly notified.
- Post-mortem review requirements to ensure lessons-learned from un-expected and un-successful changes are recorded.

Guidelines

- This shall involve the separation of development, UAT and production environments for test and development of change deployment, version control, test plans and peer reviews, back out plans.
- The change process will also include the mechanism to coordinate and communicate with end-user clients where required
- Any changes to Xyris Assets or infrastructure must be authorised and recorded within Linear (current issue tracking system). As a minimum the change control documentation shall capture:-
    - All change requests
    - Requirements
    - Approver
    - Person(s) implementing the change
    - Security Impacts, weaknesses or flaws if applicable
    - Fall back (rollback) or fix if applicable
    - Review of impact if applicable

1.2.3 Review applications after operating platform changes

Policy

- Review Business critical application when operating platforms are changed.
- Business critical applications tested whenever a change is made to the operating platform
- Ensuring changes to the operating platforms do not impact Xyris operations, or clients
- Ensuring that changes to operating platforms do not compromise security

Guidelines

- Xyris will plan out all aspects of the changes as required by the change control process
- Provided plenty of notice to testers and reviewers before the changes are made
- Review application procedures after an operating platform change

1.2.4 Restrict and control changes to software packages

Policy

- Any introduced modification to software packages to be assessed
- Unnecessary changes to be controlled

Guidelines

- Approval process to restrict unnecessary changes
- Verify that no internal controls could be compromised with the implementation of the software package
- Analyse any impact on Xyris with the implementation of the software package
- Establish a software and deployment process for approved updates to the software package
- Ensure software packages are kept up to date

1.2.5 Establish and protect secure development environments

Policy

- establish a secure development environment to protect system development and integration efforts
- secure development environment cover the entire system development lifecycle
- protect your secure development environment

Guidelines

- Consideration of the sensitivity of the data that the system will handle
- establish a secure development environment for each project
- consider the need to segregate development environments
- consider the need to control access to each environment
- consider the data moving from and to each environment
- consider the need to store backups at secure offsite locations

1.2.6 Test security functionality during development cycle

Policy

- test security functionality during system development

Guidelines

- Establish a system testing and verification process
- Engage a third party to carry out independent tests prior to major releases

1.2.7 Use acceptance criteria to test information systems

Policy

- establish acceptance testing
- use acceptance testing to test upgrades and new versions of systems before they are accepted for use

Guidelines

- plan system acceptance testing and verification activities
- make sure that tests are performed in a realistic test setting

- test new systems and new versions/upgrades of old systems
- test whether system security requirements have been met
- verify that all security related defects have been remediated
- test adherence to secure system development practices

1.2.8 Use formal procedures to control system releases

Policy
All system changes must go through the Release Management process and must have completed request for change with appropriate approvals.

Guidelines
- Releases should be frequent and incremental where possible
- A software developer must be identified as the internal contact for every Project. The contact will be responsible for the successful coordination and execution of releases in relation to the Project
- The UAT tester must ensure all required documentation related to the release exists.
- Proof that controls (initiation, testing, and approval) have been followed for all auditable Releases shall be stored with the ability to be reproduced.
- Each Release shall be initiated through a standardized and approved process (service request, incident management or projects)
- Each Release should be well tested and verified prior to implementation.
- All implementation work on the Release should be completed by a scheduled date and time.
- Validation that the Release has been completed successfully should be confirmed through post-Release testing (such as production verification testing).