Configuration Management Policy

This policy, and associated procedures for configuration management, has been sub-contracted in part to a third-party, HIPAA compliant, Platform as a Service Vendor (PaaS Subcontractor). We have verified that their policies and procedures meet or exceed our standards and those of HIPAA and the HITRUST Common Security Framework. As such, we have assurances that configuration management especially as it relates to the hosted server environment and disaster recovery are available on all systems hosted by our sub-contractor.

Proof of such due diligence is kept by the Security Officer.

Applicable Standards from the HITRUST Common Security Framework

- 06 - Configuration Management

Applicable Standards from the HIPAA Security Rule

- 164.310(a)(2)(iii) Access Control & Validation Procedures

Configuration Management

We are responsible for:

1. Using unit tests, end-to-end, and integration tests on all software and systems prior to being moved to production.
2. Reviewing all committed code using pull requests (on Github) to assure software code quality and proactively detect potential security issues in development.
3. Utilizing development and staging environments that mirror production to assure proper function.

PaaS Subcontractor is responsible for:

1. Using Salt to standardize and automate configuration management.
2. Using OSSEC to scan systems every 2 hours and on reboot. These scans capture file system changes and also unauthorized or malicious software.
3. Verifying so system is deployed into an environment without approval of the PaaS Subcontractor's CTO.

4. All changes to production systems, network devices, and firewalls are approved by the PaaS Subcontractor's CTO before they are implemented. Additionally, all changes are tested before they are implemented in production.
5. Clocks are synchronized across all systems using NTP. Modifying time data on systems is restricted.
6. All front end functionality (developer dashboards and portals) is separated from backend (database and app servers) systems by being deployed on separate servers.
7. All software and systems are tested using unit tests and end to end tests.
8. All committed code is reviewed using pull requests (on Github) to assure software code quality and proactively detect potential security issues in development.
9. Utilize development and staging environments that mirror production to assure proper function.
10. Changes to Salt and configuration management tools are documented and approved using appropriate forms and documentation.
11. Changes to production inventory is documented and approved using appropriate forms and documentation.