7. Configuration Management Policy

Dash Solutions standardizes and automates configuration management through the use of Bash scripts, as well as documentation of all changes to production systems and networks. Bash scripts automatically configure all Dash Solutions systems according to established and tested policies and are used as part of our Disaster Recovery plan and process.

7.1 Applicable Standards

7.1.1 Applicable Standards from the HITRUST Common Security Framework

- 06 - Configuration Management

7.1.2 Applicable Standards from the HIPAA Security Rule

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

7.2 Configuration Management Policies

1. Dash Solutions uses Bash scripts to standardize and automate configuration management.

2. No systems are deployed into Dash Solutions environments without approval of the Dash Solutions CTO.

3. All changes to production systems, network devices, and firewalls are approved by the Dash Solutions CTO before they are implemented to assure they comply with business and security requirements.

4. All changes to production systems are tested before they are implemented in production.

5. Implementation of approved changes are only performed by authorized personnel.

6. Tooling to generate an up-to-date inventory of systems, including corresponding architecture diagrams for related products and services, is hosted on Amazon S3 Storage.

- All systems are categorized as production and utility to differentiate based on criticality.

- The Security Officer maintains scripts to generate inventory lists on demand using APIs provided by each cloud provider.

- Diagrams and asset lists will be created through diagram library and/or manual creation as required by the Risk Assessment phase of Dash Solutions's Risk Management procedures (§2.3.1).

- After every use of these scripts, the Security Officer will verify their accuracy by reconciling their output with recent changes to production systems. The Security Officer will address any discrepancies immediately with changes to the scripts.

7. All frontend functionality (developer dashboards and portals) is separated from backend (database and app servers) systems by being deployed on separate servers or containers.

8. All software and systems are tested using unit tests and integration tests.

9. All committed code is reviewed using pull requests to assure software code quality and proactively detect potential security issues in development.

10. Dash Solutions utilizes development and staging environments that mirror production to assure proper function.

11. Dash Solutions also deploys environments locally using Vagrant to assure functionality before moving to staging or production.

12. All formal change requests require unique ID and authentication.

13. Dash Solutions uses the Security Technical Implementation Guides (STIGs) published by the Defense Information Systems Agency as a baseline for hardening systems.

    - Windows-based systems use a baseline Active Directory group policy configuration in conjunction with the Windows Server 2012 STIG.

    - Linux-based systems use a Red Hat Enterprise Linux STIG which has been adapted for Ubuntu and improved based on the results of subsequent vulnerability scans and risk assessments.

14. Clocks are continuously synchronized to an authoritative source across all systems using NTP or a platform-specific equivalent. Modifying time data on systems is restricted.

7.3 Provisioning Production Systems

1. Before provisioning any systems, ops team members must file a request in the Trello Deployment Ticket (DT) project.

   - GitHub access requires authenticated users.

   - The CTO grants access to the GitHub project following the procedures covered in the Access Establishment and Modification section.

2. The VP Engineering or CTO must approve the provisioning request before any new system can be provisioned.

3. Once provisioning has been approved, the ops team member must configure the new system according to the standard baseline chosen for the system's role.

   - For Linux and Windows systems, this means adding the appropriate roles to the system's configuration settings.

   - If the system will be used to house production data (ePHI), the ops team member must add an encrypted block data volume to the VM during provisioning.

   - For systems on AWS, the ops team member must add an encrypted Elastic Block Storage (EBS) volume.

   - For systems on other cloud providers, the ops team member must add a block data volume and set up OS-level data encryption.

4. Once the system has been provisioned, the ops team member must contact the security team to inspect the new system. A member of the Security team will verify that the secure baseline has been applied to the new system, including (but not limited to) verifying the following items:

   - Removal of default users used during provisioning.

   - Network configuration for system.

    - Proper IAM roles and security groups are implemented.
    - Data volume encryption settings.
    - Intrusion detection and virus scanning software installed.
    - All items listed below in the operating system-specific subsections below.

5. Once the security team member has verified the new system is correctly configured, the team member must add that system to the OpenVAS configuration.

6. The new system may be rotated into production once the security team verifies all the provisioning steps listed above have been correctly followed and has marked the Issue with the Approved state.

7.3.1 Provisioning Linux Systems

1. Linux systems have their baseline security configuration applied via configuration management tools described in §7.2. These baseline configuration states cover:
    - Ensuring that the machine is up-to-date with security patches and is configured to apply patches in accordance with our policies.
    - Stopping and disabling any unnecessary OS services.
    - Installing and configuring the IDS agent described in the IDS Policy.
    - Configuring 15-minute session inactivity timeouts.
    - Installing and configuring the ClamAV virus scanner.
    - Installing and configuring the NTP daemon, including ensuring that modifying system time cannot be performed by unprivileged users.
    - Configuring audit logging as described in the Auditing Policy.
    - Connecting system to proper security roles
    - Configure system logging into centralized logging tool

2. Any additional system configuration applied to the Linux system must be clearly documented by the ops team member in the DT request by specifying the purpose of the new system.

7.3.1 Provisioning Linux Systems

1. Linux systems have their baseline security configuration applied via Bash scripts. These baseline configuration states cover:

    - Ensuring that the machine is up-to-date with security patches and is configured to apply patches in accordance with our policies.
    - Stopping and disabling any unnecessary OS services.
    - Installing and configuring the IDS agent described in the IDS Policy.
    - Configuring 15-minute session inactivity timeouts.
    - Installing and configuring the ClamAV virus scanner.
    - Installing and configuring the NTP daemon, including ensuring that modifying system time cannot be performed by unprivileged users.
    - Configuring audit logging as described in the Auditing Policy.
    - Connecting system to proper security roles
    - Configure system logging into centralized logging tool

2. Any additional system configuration applied to the Linux system must be clearly documented by the ops team member in the DT request by specifying the purpose of the new system.

7.3.3 Provisioning Management Systems

1. Provisioning management systems, LDAP servers, or VPN appliances follows the same procedure as provisioning a production system.
2. The VP Engineering will oversee the provisioning of new configuration management services.
    - Once the configuration management service has been provisioned, the ops team member will apply the baseline configuration to the service.
3. Critical infrastructure services such as logging, monitoring, LDAP servers, or Windows Domain Controllers must be configured with appropriate baseline security settings.

    - These baseline security settings have been approved by the VP Engineering and CTO to be in accordance with all Dash Solutions policies, including setting appropriate:
        - Audit logging requirements.
        - Password size, strength, and expiration requirements.
        - Transmission encryption requirements.
        - Network connectivity timeouts.

4. Critical infrastructure roles applied to new systems must be clearly documented by the ops team member in the DT request.

7.4 Changing Existing Systems

1. Subsequent changes to already-provisioned systems are unconditionally handled by one of the following methods:
    - Configuration changes with Bash scripts
    - For configuration changes that cannot be handled by Bash scripts, a runbook describing exactly what changes will be made and by whom will be created.
2. Configuration changes to Bash scripts must be initiated by creating an issue in the deployment GitHub Repository.
    - The ops team member will create a feature branch and make their changes on that branch.
    - The ops team member must test their configuration change locally when possible, or on a development and/or staging sandbox otherwise.
    - At least one other ops team member must review the configuration management change before merging the change into the main branch.
3. Once the request has been approved by the CTO, the ops team member may roll out the change into production environments.

7.5 Patch Management Procedures

1. Dash Solutions uses automated tooling including Bash scripts, to ensure systems are up-to-date with the latest security patches.

2. On production systems, the unattended-upgrades tool is used to apply security patches in phases.

   - Dash Solutions will install and test new patches through test systems

   - All patches will be added through Bash scripts configuration

   - Newly patched systems will be deployed/launched as new systems via Bash scripts

   - Bash scripts will remove old systems once newly patched systems have been properly configured in the environment.

7.6 Software Development Procedures

1. All development uses feature branches based on the main branch used for the current release. Any changes required for a new feature or defect fix are committed to that feature branch.

   - These changes must be covered under 1) a unit test where possible, or 2) integration tests.

   - Integration tests are *required*

2. Developers are strongly encouraged to follow the commit message conventions suggested by GitHub.

3. Once the feature and corresponding tests are complete, a pull request will be created using the GitHub web interface. The pull request should indicate which feature or defect is being addressed and should provide a high-level description of the changes made.

4. Code reviews are performed as part of the pull request procedure. Once a change is ready for review, the author(s) will notify other engineers using an appropriate mechanism, typically via an @channel message in Slack.

   - Other engineers will review the changes, using the guidelines above.

- Engineers should note all potential issues with the code; it is the responsibility of the author(s) to address those issues or explain why they are not applicable.

5. If the feature or defect interacts with ePHI, or controls access to data potentially containing ePHI, the code changes must be reviewed by the Security Officer before the feature is marked as complete.

    - This review must include a security analysis for potential vulnerabilities such as those listed in the OWASP Top 10.

    - This review must also verify that any actions performed by authenticated users will generate appropriate audit log entries.

6. Once the review process finishes, the reviewer should approve the pull request, at which point the original author(s) may merge their change into the release branch.

7.7 Software Release Procedures

1. Software releases are treated as changes to existing systems and thus follow the procedure described in §7.4.