Document ID: CIST-0056	Title: Application Security Standard	
Parent Documents: CIST-0004		
Effective Date: See Document Information Page	Last Review Date: See Review and Revision History Section	Business Process Owner (BPO): Dir, Information Security, Security Policy
Exhibit(s): Exhibit A: Compliance Effective Date;		
Document Type: Policy and Standard		

PURPOSE

This standard establishes a comprehensive set of Application security requirements derived from industry best practices and from internal CVS Health security assessments. The topics covered in this Document include, but are not limited to design requirements, configuration requirements and required Secure Development Life-Cycle requirements.

SCOPE

This standard applies to all internal, external, and public (internet) facing Applications and any Application that stores, processes, creates, modifies or destroys data owned by CVS Health, its subsidiaries or affiliates.

POLICY

Information Security Policy, CIST-0004

COMPLIANCE EFFECTIVE DATE

Revisions to this standard shall be effective as of the date of publication except as set forth in Appendix A. All other provisions within this policy remain in effect.

STANDARD

1. Roles and Responsibilities
   a. Application architects and developers shall implement and follow the requirements of this standard during the design, development, and maintenance phases of the Software Development Life-Cycle.
   b. Where development is outsourced, change control procedures to address security are included in the contract(s) and specifically require the developer to track security flaws and flaw resolution within the Application, System, component, or service and report findings to CVS Health defined personnel or roles.
2. Corporate and Industry Standards and Government Regulations
   To ensure the confidentiality, integrity, and availability of Confidential Information, Applications shall comply with the following standards and regulations:
   a. Any applicable CVS Health policies, procedures, and standards (e.g., data shall be kept in accordance with CVS Health's then-current records and data Management and retention policies, procedures, and standards. Please refer to the Corporate Records Management Program, CRCMGT 0002).
   b. Any applicable laws, rules, and regulations, including, but not limited to, those pertaining to the security and privacy of information for Systems that contain PHI and CHD.
      i. HIPAA
      ii. PCI-DSS
3. Protection of System Environment and Business Data
   a. Applications shall utilize separate environments for development, testing, and Production. Application Managers shall ensure that all proposed Application

development and System changes are reviewed to check that they do not compromise the security of either the Application or the development, testing, and Production operating environments. Further, project and support environments are strictly controlled by the appropriate Technology or Application Owner.

b. Applications storing, processing, and/or transmitting PCIDSS data shall be physically segmented from other Systems. Network Infrastructure shall ensure the storage of PCI-DSS Data is located outside the publicly accessible environments (e.g., on a storage platform existing on CVS Health's intranet.
c. User access to development, testing, and Production environments shall be granted based on Minimum Necessary requirements to perform a job function. Requests and approvals for access to these environments shall be in accordance with CIST-0050, User Access Management Standard.
d. Users shall only have the ability to change Production Systems or Applications through a Change Control Board (CCB). CVS Health shall develop, a documented control System for implementations and configuration Management plans.
e. Use of operational data containing any Confidential Information including, but not limited to, PII, PHI/ePHI, PCI Data, non-published Financial Statement information, or CVS Health's competitive data shall not be copied into test or development environments. If it is necessary to test or develop with operational data all Sensitive Information shall be removed or masked in accordance with IFGV-045474, Data Masking Policy.
f. Test data, Application Internal User Accounts, System Internal User Accounts, and passwords shall be removed from Productions Systems and Applications before the Systems and Applications become active.
g. Application software shall be designed and implemented to utilize security logging capabilities to ensure appropriate access is maintained to Information Assets, when technically feasible. Access rights to Application functions shall be limited to the minimum necessary in all environments.
h. Separate development/test environments from Production environments, and enforce the separation with access controls.
g. Users or developers shall not hard code any usernames/passwords in scripts or clear text files in any scripts, batch jobs, word processing Documents and/or Applications in accordance with CIST-0010, Password and Credential Standard.
h. CVS Health QA Testers shall execute appropriate risk-based security test plans including but not limited to those designed to identify the following security risks:
    a. Clickjacking
    b. HttpOnly Cookie Attribute Not Set
    c. Secure Cookie Attribute Not Set
    d. AutoComplete HTML Attribute Not Disabled for Sensitive Fields
    e. Memory Allocation Errors (i.e., Buffer Overflows)
    f. Acceptance of Out of Range Values

4. Authentication

Applications shall not be susceptible to authentication circumvention. Multi-factor authentication methods shall be used in accordance with CVS Health's CIST-0010, Password and Credential Standard. In addition, the following specific guidelines apply:

a. All credentials shall be handled securely and shall be encrypted during transmission and in storage on all System components.
b. Applications shall not use SSN to authenticate.
c. The Application shall deny access by default and specifically identify the conditions under which access is allowed. The Application shall not implement "Fail-open" login mechanisms.
d. Web Application Authentication
    i. The entire web application shall be delivered to the User's browser using HTTPS.
    ii. Login Credentials shall be submitted using POST over HTTPS.
    iii. Submit Sensitive Information using POST over HTTPS.
    iv. Application POST data and information shall be encrypted in accordance with CIST-0018, Encryption and Key Management Standard when used for submitting Confidential Information or Sensitive Information.
    v. Basic and Digest Authentication shall be used only with HTTPS.

5. Authorization
    a. Adhere to the Principle of Least Privilege
        i. Every process or transaction shall execute with the least set of privileges necessary to complete the job.
        ii. Applications shall run under Special Account IDs with minimal privileges. CVS Health shall limit authorization to privileged accounts on information Systems to a pre-defined subset of Users.
        iii. Database accounts used by Applications shall not have more access privileges to the Database than is required to implement Application functionality. The Application shall use limited Special Account IDs that do not have schema-modification privileges unless required.
        iv. The Application shall utilize platform capabilities to limit the privileges of the Application code.
        v. Business Internal User Accounts shall not be Privileged Access accounts and Privileged Access accounts shall not be business Internal User Accounts. If a User requires both an Internal User Account and a Privileged Account, separate accounts shall be created.
        vi. Development, test, and staging environments shall be set up to function with the lowest possible privileges so that the associated Production environments shall also work with the lowest possible privileges.
        vii. Database access shall only be provided using a low privilege Database account that does not hold any SQL roles above "User."
    b. Improper Access Control
        i. Demonstrate that improper access control – such as insecure direct object references, failure to restrict URL access, and directory traversal – is addressed by coding techniques that include:
            1) Standard authentication of Users.
            2) Sanitizing input – removing Universal Resource Identifier (URI).
            3) Not exposing internal object references to Users.
            4) User interfaces that do not permit access to unauthorized functions.
            5) Attempted unauthorized remote connections to the information Systems shall be

monitored and reviewed and appropriate action shall be taken if an unauthorized connection is discovered.

c. Avoid Use of Client-side Tokens
Do not trust any client-side authentication or Authorization Tokens in headers, cookies, hidden form fields, or in URL arguments unless they have been cryptographically secured via signing or encryption.

6. Session Management
   a. Maintain Session State on the Server
   The Application shall maintain all User ID session state data (authentication, Authorization, and role data) on the Server only. For User ID authentication and Authorization, the Application shall not rely on any data stored on the client other than the session identifier. Session state shall be tied to a specific client (e.g., browser) session through the use of session cookies. Additionally, hidden fields shall not be used to pass Sensitive Information or Confidential Information between the browser and the Server, unless the Sensitive Information and Confidential Information is encrypted during transit.
   b. Do Not Span Sessions Over Protected and Unprotected Resources
   For protected parts of the Application, use HTTPS to transmit session identifiers and do not reuse session identifiers that were initially transmitted for unprotected resources in clear via HTTP, thereby increasing the Risk of session hijacking. If the session identifier was initially transmitted over clear HTTP, an additional session identifier shall be used for secure parts of the Application.
   c. Destroy Session Tokens on Logout
   d. Broken authentication and session Management
      i. Corroborate that broken authentication and session Management are addressed via coding techniques that commonly include:
         1) Flagging session Tokens (for example Cookies) as “secure”.
         2) Not exposing session IDs in the URL.
         3) Incorporating appropriate time-outs and rotation of session IDs after a successful login.

4) Validate Session Identifiers.
5) Enforce Session Identifier Security.

ii. To prevent session hijacking, the Application shall protect session identifiers in transit. Encryption shall be used to protect session identifiers covered information in on Session cookies mobile/removable media and across communication lines based on pre-determined criteria defined in CIST-0018, Encryption and Key Management Standard. The Application shall generate the session IDs using strong, non-predictable algorithms. Session identifiers shall not be stored in persistent cookies or in hidden fields. Instead, session IDs shall be stored in non-persistent cookies with the secure flag enabled. Session cookies shall be configured to restrict distribution beyond the Application by using the domain and path attributes in the cookie properly. NOTE: While cookies generated by an Application shall use domain and path attributes to restrict distribution beyond the Application, it is permissible for Enterprise level SSO solutions like SiteMinder to share cookies between Applications. An Application's individual SiteMinder cookie configuration shall include the setting of the Secure Flag and the HTTP Flag to avoid the sharing of sessions across unprotected resources.

e. Enforce Session Expiration
Information shall be enabled to automatically timeout the session after a maximum period of time. Re-establishment of the session shall take place only after the User has provided a valid password. The following requirements shall be followed:
i. Server: Twenty (20) Minutes.
ii. Workstation: Thirty (30) Minutes.
iii. Mobile Device: Thirty (30) Minutes.
iv. For terminals or Workstations in high-Risk locations (e.g., public areas) or sensitive Systems (i.e. contains CHD or PCI data), the time-out delay shall be set to a maximum of fifteen (15) minutes. A time-out System (e.g. a screen saver) shall pause the session screen after 15 minutes of inactivity and shall close Network sessions after 30 minutes of inactivity.
v. Active sessions shall expire after a maximum of twenty-four (24) hours.

7. Cryptography and Data Security
a. Use of cryptographic algorithms to ensure the confidentiality and integrity of Confidential Information and Sensitive Information (including passwords) shall conform to the requirements specified in CVS Health's Encryption and Key Management Standard, CIST 0018 and shall utilize a FIPS 140-2 Validated Cryptographic Module. Permitted algorithms, key strengths and cryptographic modules are identified in the Encryption and Key Management Procedure, CIST-0111.
b. Any transmission of Sensitive (inclusive of passwords, PCI, PII, PHI, ACH and EFT), Confidential or Proprietary Information or System/User Credentials shall use a secure mechanism such as Cryptography. It shall be used to protect the confidentiality and integrity of Remote Access sessions to the internal Network and to external Systems, which provides for:
i. Authenticated access.

ii. Encryption of covered data elements using a mechanism meeting the requirements of Section 7a of this Document. The encryption can be within the payload, via the transport mechanism (e.g., TLS) or via a secure channel (e.g., IPsec, MACsec, and VPN). If transport layer security is used, the requirements found in CIST-0107, CVS Health's Secure Network Protocol Use Standard shall be adhered to.
iii. Message integrity.

c. Applications shall avoid frequent misuses of cryptography, such as:
i. Poor source of random numbers for cryptographic algorithms.
ii. Not managing key material safely.
iii. Hiding cryptographic credentials in client software or on client Systems.
iv. Use of homegrown cryptographic algorithms.
v. Use of homegrown implementation of well-known cryptographic algorithms.

8. HTTP Cookies (Web Applications Only)
a. Cookies shall not contain Confidential Information and Sensitive Information unless they have been configured for strong security. Cookies shall not contain plaintext Usernames and passwords.
b. Configuring cookie attributes for strong security:
If it is unavoidable to use cookies for transmitting Confidential Information or Sensitive Information, you shall ensure that the following cookie attributes are properly configured to reduce the Risks associated with passing around Confidential Information and Sensitive Information inside cookies:
1) An expiration date and time shall be included in the cookie.
2) Cookies shall always be sent over HTTPS, rather than plain HTTP. You can do this by enabling the ***secure*** attribute on the cookie.
3) Cookies shall be created as "non-persistent".
4) The ***domain*** and ***path*** attributes shall define the exact URLs to which browsers shall submit your cookie. These two attributes shall be as restrictive as possible.
5) The ***HttpOnly*** attribute shall be used for cookies.
6) Any sensitive data contained in the cookie shall be encrypted in accordance with Encryption and Key Management Standard, CIST 0018.
c. Validating cookie data:
Before using any cookie data, cookie content shall be examined to verify that they are not storing malicious content by performing proper input validation for type, length, syntax, and range.

9. Database Calls
a. Parameterized queries shall be used and the input parameters shall be validated.
b. Use Limited Privileges
i. Database connections shall be created using limited Privilege Accounts.
c. The Database connection strings shall be secured.

10. User IDs and Passwords shall be in accordance with CIST-0010, Password and Credential Standard.
a. Applications shall meet the following key security objectives pertaining to User credential Management:

i. Handling of credentials (both in transit and in storage) in accordance with CIST-0010, Password and Credential Standard. ;
ii. Prevention of information leakage (to prevent User ID enumeration, etc.);
iii. Protection against Brute-Force Attacks (to prevent User ID/password guessing);
iv. Protection against misuse of the password change function;
v. Protection against misuse of the password recovery function; and
ix. Use of proper logging and notification of the activities of all Users, including privileged Users (administrators, operators, etc.) include the success/failure of the event, time the event occurred, the account involved, the processes involved, and additional information about the event in accordance with Logging, Monitoring and Reporting Standard, CIST-0006.

b. User IDs
Applications shall ensure that User IDs are unique. Elevated privileges shall be assigned to a different User ID from those used for normal business use, all Users access privileged services in a single role, and such privileged access shall be minimized.

c. Password Length
i. Password lengths shall conform to CVS Health's Password and Credential Standard, CIST 0010. The Application shall not allow blank passwords.
ii. Minimum password length for the Application shall not be less then what is allowed in Password and Credential Standard, CIST 0010.

d. Password Complexity
Password complexity requirements shall conform to CVS Health's Password and Credential Standard, CIST 0010.
The Application shall enforce password quality controls, consistent with CVS Health's Password and Credential Standard, CIST 0010

e. Password History
Password history requirements shall conform to CVS Health's Password and Credential Standard, CIST 0010. The password history shall consist only of Cryptographic Hashes of previous passwords. See below for password Cryptographic Hashing requirements.

g. Password Expiration
Password expiration requirements shall conform to CVS Health's Password and Credential Standard, CIST 0010 unless mandatory password expiration is not an option as a result of usability concerns or functionality.

h. Temporary Account Lockout After Unsuccessful Login Attempts
i. The account lockout policy shall conform to CVS Health's Password and Credential Standard, CIST 0010 and CVS Health client contracts.
1) The Application shall never indicate that any specific User ID or Special Account ID has been locked out. It shall not reveal the details of the lockout policy.
2) If a User ID or Special Account ID is locked out, then login attempts shall be rejected without even checking the credentials.

i. Displaying and Printing of Passwords
i. A permanent password Cryptographic Hash or clear-text password shall never be sent to any User in any form.
ii. If the User has forgotten his/her password, they shall not be allowed to recover the existing password.

iii. The Application shall not implement any function, page, or report that retrieves passwords from the Server.

j. Protection of Passwords in Transit Over the Network
   i. The Application shall deliver its login form to the User's browser over HTTPS and shall submit the login form back to the Server using the POST method. Protocols used to communicate between all involved parties shall be secured using HTTPS cryptographic techniques (e.g., SSL).
   ii. User ID/Passwords shall not be included in URL query string parameters.

k. Storage of Passwords
   i. Storage of passwords shall conform to CVS Health's Password and Credential Standard, CIST 0010. Additionally, the following best practices guidelines shall be implemented where they do not conflict with CIST-0010.
      1) A password salting mechanism shall be used as part of the Cryptographic Hashing process. The salt values shall be protected.
      2) User ID/passwords used to access other Systems (e.g., Database, directories, etc.) shall be protected in accordance with CVS Health's Password and Credential Standard, CIST 0010, this standard, and any other applicable CVS Health policies, procedures, and standards.

l. Password Change Functionality
   i. The Application shall have a change password feature.
   ii. The functionality shall be accessible only to authenticated User IDs.
   iii. The Application shall ensure that a User can change only his/her password and not another User's.
   iv. The password change form shall include fields to capture the old password, the new password, and confirmation of the new password. For all three passwords HTML fields, AUTOCOMPLETE=OFF shall be used to prevent browsers from caching the passwords locally.
   v. If the User gets the old password wrong five (5) times, the Application shall lock the account and kill the session.

m. Forgotten Password Functionality
   i. If the Application provides an online method for letting Users reset their forgotten password, the method shall follow one of the following two mechanisms:
      1) The Application shall send a unique recovery URL (that is not capable of being guessed) to the Email address the User provided during registration. The User shall visit this URL upon receipt in order to set a new password.
      2) Allow Users to answer their challenge question with a secret answer.
      3) The use of a "Password Hint" mechanism shall not be used.
   ii. If the challenge question approach is used, the possible questions shall be selected from a pre-approved list of questions. Any variation from these questions shall receive approval before use. Answers to questions shall be stored in salted Cryptographic Hash form, and never as plain text.
   iii. Users shall not be allowed to set their secret answer to be the same as their User ID, password, or to the corresponding challenge question.
   iv. Accounts shall be locked out temporarily following five (5) failed attempts to complete the challenge.

v. The Application shall never display the existing, forgotten password to the User after successful completion of the challenge.
vi. Generic messages shall be used to prevent the "Forgot Password" functionality from being vulnerable to User ID Enumeration Attacks.

n. Handling Incorrect Passwords
i. In case of login error, the Application shall not disclose the part of the credentials that were invalid. Instead the Application shall display a generic message to the User, such as, "Unable to login. User ID or Password is incorrect."

o. Logging and Notification
i. The Application shall log all failed and successful authentication-related events, including, but not limited to, login, logout, password change, password reset, account lockout, and account recovery.
ii. The log entries shall contain non-secret details such as IP address and User ID, but never security secrets such as passwords and answers to secret questions.
iii. Logs shall be strongly protected from unauthorized access in accordance with Logging, Monitoring and Reporting Standard, CIST-0006.
iv. The Application shall log additionally any information as required by the Logging, Monitoring and Reporting Standard, CIST-0006.

11. Secure Development Life-Cycle Requirements

This section provides coding requirements which shall help eliminate or reduce the number of coding flaws that can lead to security Vulnerabilities.

a. All Application Vulnerabilities identified through any mandated application security testing activities shall be remediated. Vulnerabilities shall be remediated in accordance with the following Remediation Timeline Requirements (or sooner if required by contractual or regulatory obligations).

- **Critical Severity**: Prior to PROD deployment or immediately (within 30 days **or less**) if identified in PROD
- **High Severity**: Prior to PROD deployment or within 90 days if identified in PROD
- **Medium Severity**: within 180 days
- **Low Severity**: within 365 days

See: ACS-1111 Software Security Control Remediation

b. If the dynamic application security testing/scanning (DAST) tool contains severity ratings on a 5 severity scale, the below severity mapping is used to determine the correct severity (to match section 11a above):

- Severity 1 = Minimal (no policy about fixing minimal)
- Severity 2 = Low Severity (within 365 days)
- Severity 3 = Medium (within 180 days)
- Severity 4 = High (prior to PROD deployment or within 90 days if identified in PROD)

- Severity 5 = Critical (prior to PROD deployment or immediately (within 30 days **or less**) if identified in PROD)

c. Administrative Requirements
   i. See: ATCS-643 Application Inventory
   ii. See: ACS-903 Application Inherent Risk
   iii. See: ACS-1103 Security Mavens
   iv. See: ACS-1104 Secure Code Training
   v. See: ACS-1105 Threat Modeling
   vi. See: DOC-014412 Information Security Risk Management Standard
   vii. See: CIST-0013 Information Security Management Procedure

d. Application Security Testing Requirements:

Application Security Testing shall occur at a minimum, at the frequencies stated within the descriptions (below) for that specific activity. Where contractual or legal obligations dictate a more frequent cadence, that cadence must be followed as well.

See the SSG GitHub for information regarding what tools are available and accepted method for satisfying the following Security Testing requirements.

See: ACS-1101 Static Source Code Analysis

See: ACS-1102: Open Source Component Management

See: ACS-1107 and ACS-1151 Dynamic Scanning

See: ACS-1111 Software Security Control Remediation

ACS-1111 should be referenced to find specifics of the remediation standard for defects discovered during development methodology phases: Dynamic Scanning.

All web and mobile applications shall undergo dynamic application scanning to detect and remediate security vulnerabilities during the application testing process prior to production releases. If an application is released:

- 0 to 1 times within 12 months, it must undergo 1 Dynamic Scan within those 12 months
- 2 times within 12 months, it must undergo 2 Dynamic Scans within those 12 months, *maximum 1 per quarter
- 3 or more times within 12 months, it must undergo 3 Dynamic Scans within those 12 months, *maximum 1 per quarter
- 4 or more times within 12 months, it must undergo 4 Dynamic Scans within those 12 months, *maximum 1 per quarter

* Once per quarter limitation not applicable to automated Dynamic Scans (with SSG approved scanners) in development CI/CD pipeline.

* No scan limits apply to Dynamic Scanning for the purpose of re-testing.

If changes are applied to the application after Dynamic Scanning has occurred and before the relevant release to production, then additional scanning will be required to ensure application is free of vulnerabilities.

Dynamic testing involves executing authenticated security tests on applications at an appropriate time during the QA lifecycle to discover and fix application vulnerabilities prior to production releases. Dynamic testing is a test requirement similar to feature testing, regression testing, etc.

Additionally, if a Pen Test is performed, it may be used in lieu of a Dynamic Scan but it should be noted that an application requiring multiple Dynamic Scans within 12 months can only use the Pen Test to satisfy the requirement for 1 of those Dynamic Scans.

See: ACS-1108 Penetration Testing

Penetration tests shall be conducted on applications selected by the Software Security Group (SSG) as well as on those that fall under contractual or regulatory obligations for Penetration Testing.

Except where required by contractual or regulatory obligations to be conducted in production, Penetration tests will only be conducted in pre-production environments. Penetration Testing performed in PROD environments will only be done under an approved Change Request. Other environments may require approved change requests if deemed appropriate per business and leadership requirements. The SSG will obtain appropriate change request approvals in these situations.

Penetration tests shall be performed on the interface exposed at the CVS Health perimeter regardless of any additional controls in place for public access, e.g. exposed publically via Akamai. The penetration tests may additionally be performed at the publically exposed interface to determine the effectiveness of such additional controls but do not impact the need to remediate identified vulnerabilities exposed at the actual perimeter.

e. All Applications shall be reviewed when changes occur in accordance with CITD-0020 Change Management Standard.
    - Only authorized Agents shall be allowed to implement approved upgrades to software, Applications, and program libraries, based on business requirements and the security implications of the release.

12. Additional Design and Security Requirements

a. Fail Securely

When an error or exception occurs during program execution the Application shall fail securely and shall not disclose Application errors or Sensitive Information.

b. Use of Validation Controls (ATCS-421):

i. Every data entry point shall be validated within the Application. Client validation shall not be solely relied upon. Validation shall be done on all Server entry points. User-controlled files and Database queries shall be properly validated.

ii. The following input validation controls should be implemented:

(1) Automated input checks (error checks) to detect out-of-range values, invalid characters, missing or incomplete data, syntax, length and type using regular expression or otherwise.

(2) Periodic review of key fields and suspicious or unusual data to confirm validity

(3) Procedures for testing the validity of input data (e.g., dual entry)

(4) Procedures for responding to validation errors

iii. All Output data shall be validated to ensure data integrity.

c. Sanitize Data.

When passing string data containing special characters within the Application or other System, the data shall be properly sanitized.

e. Injection flaws

i. Verify that injection flaws, particularly SQL injections (but also considering OS Command Injection, LDAP and XPath injection flaws as well as other injection flaws) are addressed by coding techniques that include:

1) Validating input to verify User data cannot modify meaning of commands and queries.

2) Utilizing parameterized queries.

f. Buffer Overflows

i. Demonstrate that buffer overflows are addressed by coding techniques that include:

1) Validating buffer boundaries.

2) Truncating input strings.

g. Insecure Communications

i. Validate that insecure communications are addressed by coding techniques that properly authenticate and encrypt all sensitive communications in accordance with Encryption and Key Management Standard, CIST 0018.

h. Improper Error Handling

i. Confirm that improper error handling is addressed by coding techniques that do not leak information via error messages (for example, by returning generic rather than specific error details).

i. Web Applications and Application Interfaces (Internal and External)

i. Cross-site scripting (XSS)

1) Verify that cross-site scripting (XSS) is addresses by coding techniques that

include:

(a) Validating all parameters before inclusion.

(b) Utilizing context-sensitive escaping.

ii. Cross-site Request Forgery (CSRF)

1) Establish that cross-site request forgery (CSRF) is addressed by coding techniques that ensure Applications do not rely on Authorization credentials and Tokens automatically submitted by browsers.

j. Internet facing API / SOAP services (ACS-932):

As a method to allow access to common services, the company exposes both REST and SOAP APIs to the Internet in order to address business requirements such as providing services that are consumed by mobile and web applications. Any API therefore shall only be exposed to the Internet via Enterprise Architecture approved standard mechanisms (for e.g., using the IBM APIConnect appliance) and shall not be exposed directly from web servers.

k. Message Integrity

All messages coming into the Server shall be authenticated and checked for integrity'. Stronger levels of authentication shall be implemented to control access from publicly accessible Networks.

l. Usage of HTTP & HTTPS for Web and Mobile Applications (ACS-3763):

Internal and external CVS Health web and mobile applications shall be configured to solely leverage HTTPS, (HyperText Transfer Protocol Secure) and to not leverage HTTP (Hyper Text Transport Protocol). Leveraging HTTPS helps maintain privacy of application interactions for CVS Health consumers, employees, and other stakeholders, and mitigates impact of brand reputation impacts stemming from browser and application feedback about insecure connections.

m. Browser, Middleware and Component Independence (ACS-904):

Applications shall be architected and developed in a manner such that they are not reliant upon functionality of browsers, middleware, and other components in a manner that impedes the ability to update packages and assure currency in the future. Application owners are responsible for assuring that middleware and other integrated components remain current and supported and that these packages are updated regularly to assure currency and compliance with security best practices. Application owners shall maintain an inventory of such components and are responsible for assuring upgrade activity takes place appropriately as new versions are released, and that only supported versions of components are leveraged.

Web applications should be developed in a manner that assures compatibility with all supported versions of major web browsers (Internet Explorer, Google Chrome, Mozilla Firefox, etc.). Web applications should be adequately tested to assure compatibility on all platforms prior to release, and the web application must be routinely updated to assure compatibility with all mainstream supported browsers. Web applications shall not be

architected or developed in a manner that is reliant upon proprietary functionality within a given browser platform, and instead shall be designed and developed in a manner that assures consistent functionality across all mainstream browsers.

n. PHP is prohibited in external facing environments (ACS-902):

   i. PHP based applications are not allowed to be deployed in any of the externally facing environments such as DMZ. For that reason, no PHP runtime environment is also allowed to be deployed on the servers in those environments.

   ii. Additionally, business units should not procure any application that includes functionality that leverages the PHP server-side scripting language.

   iii. Applications that currently include PHP components must be re-architected to leverage other technologies. Vendor applications that currently include PHP components must be updated to leverage other technologies.

   iv. In the case of Cloud Environments, GS and Platform Engineering review for PHP related vulnerabilities must be completed for PHP applications before their deployment to include the following:

      1) If Wordpress, the site must upgrade to the latest WordPress version

      2) An ASM (WAF) policy must be put on the site as soon as possible to mitigate potential risks with the legacy version of WordPress running

      3) Lock down the wp-login page to a subset of IP addresses belonging to the organization only, mitigating brute force campaigns

   v. In addition, periodic review of PHP applications for any known PHP related vulnerabilities need to performed and vulnerabilities found must be remediated on time.

o. Java Runtime Environment (JRE) and Java Development Kit (JDK) Currency (ACS-502):

   i. CVS Health applications and systems shall only leverage instances of JRE and JDK that are under active support from the supplying vendor (i.e. IBM, Oracle, etc.). Note that this requirement extends not only to applications and systems developed by CVS Health, but to applications and systems purchased from third party vendors.

   ii. These requirements apply independently of the hosting location of a system or application (i.e. company internal, company cloud, external cloud, third party hosting provider, etc.)

   iii. Maintaining currency of JRE/JDK platforms is an important step in assuring that company systems maintain the appropriate security stature, currency, and supportability.

   iv. JRE dependencies in vendor software shall be reviewed prior to purchase, and the business owner must work with the proper vendor support team to assure a supported

version is leveraged throughout the entire system lifecycle. CVS Health system owners are responsible for maintaining currency of the software platform and JRE as appropriate.

v. JRE dependencies for company developed applications shall be routinely reviewed. Applications must be migrated to new versions of the JRE/JDK prior to the end of life data for the existing component.

vi. Instances of non-compliance with this standard must be submitted and approved as exceptions.

p. Usage of iFrames Within CVS Health Applications (ACS-4889):

CVS Health web applications leveraging iFrames for integration with third party services (i.e. marketing, tracking, advertisement, etc.), must leverage the 'Sandbox' attribute for such iFrames.

Enabling the iFrame Sandbox attribute provides the following security benefits:

i. Treat the content as being from a unique origin

ii. Block form submission

iii. Block script execution

iv. Disable APIs

v. Prevent links from targeting other browsing contexts

vi. Prevent content from using plugins (through <embed>, <object>, <applet>, or other)

vii. Prevent the content to navigate its top-level browsing context

viii. Block automatically triggered features (such as automatically playing a video or automatically focusing a form control).

ix. The value of the sandbox attribute can either be just sandbox (then all restrictions are applied), or a space-separated list of pre-defined values that will REMOVE the particular restrictions.

Third party integration partners must assure that their iFrame functionality works properly within a sandboxed iFrame prior to integration.

q. Application Signing (Continuous Integration) (ACS-4711)

All applications which need to be deployed and require signing during continuous integration process, must use the enterprise provided API to obtain the signing certificates. The certificate must also be appropriately disposed of after the signing process.

r. Business Developed Applications (BDA's) must follow these controls:

i. Required Security Controls for Business Developed Applications (BDA) (ACS-4741)

ii. Restricted Data Contained Within Business Developed Applications (BDAs) (ACS-4789)

iii. Sarbanes-Oxley (SOX) Data & Business Developed Applications (BDAs) (ACS-4790)

s. Restricted Use of Adobe Flash within Applications (ACS-1113):

Given the presence of several widely publicized security vulnerabilities within the Adobe Flash platform, no internally developed or externally procured applications may leverage Adobe Flash as a component of the application. The platform is considered to be deficient, as exploitation of the vulnerabilities existing within the platform may lead to the compromise of customer and enterprise data, or the compromise of enterprise computing assets. Vulnerabilities manifest themselves in several different ways on the Flash platform, including through tight coupling with JavaScript, which provides a fruitful attack vector.

Usage of Adobe Flash technology in both internally developed and vendor procured applications is disallowed, unless explicitly permitted via approved exception against this control standard.

The control owner shall be responsible for the review the listed exceptions at least once every two years.

t. Usage of Adobe Flash within Internally Developed Applications (ACS-1114):

Company applications are restricted from new implementations of Adobe Flash. Any application team seeking to implement Adobe Flash within new application development activity may only do so following approval an exception launched against this standard. Existing applications leveraging Adobe Flash must create remediation plans to migrate away from Flash and re-develop required functionality using an alternative platform, such as HTML5.

The control owner shall be responsible for the review the listed exceptions at least once every two years.

u. Secure Development Standards and Libraries (ACS-1106):

This requirement is currently only pertain to Aetna's systems. Global Security will review the current libraries used in CVS Health to see if this requirement is applicable to CVS Health systems.

Use of Aetna Enabling Framework for .NET (all versions)

The .NET TKC is no longer supporting the Aetna Enabling Framework for .NET (AEFW). Additionally, AEFW will not be certified for Windows 10 at CVS Health. Because of this, any application using AEFW must remove these references before it can be certified for use with Window 10.

The .NET TKC provides replacement modules for the main functionality of AEFW, this network of modules is known as Aetna Enabling Microframeworks (AEMF). Hence, any applications that are intended to be use AEFW must use AEMF instead.

Applications shall implement secure development libraries or frameworks approved by Global Security to improve the resiliency of applications against software attacks.

All Java and .NET applications must implement a strategy to mitigate common application security attacks by adopting libraries and frameworks approved by Global Security.

Java and .NET applications shall leverage the Aetna Enablement Framework as appropriate, and shall leverage all appropriate security resources within the library.

The AEFW integrates core functionality of ESAPI, including input validation and output encoding. Development on platforms not covered by the AEFW must leverage a similar standardized library (i.e. ESAPI) for common security functionality, such as input validation and output encoding.

Developers shall adhere to all published secure development standards, including both documented control standards, as well as documentation published to the Software Security Group intranet site

v. Third Party Process Assessment (ACS-1110):

Third parties that develop and distribute software used by CVS Health for business purposes shall undergo software security process maturity assessments as part of CVS Health's vendor management program.

Third parties that develop and distribute high risk applications will undergo software security process maturity assessments. Global Security will select vendors that require assessments. The process maturity assessment method will be provided by Global Security. Assessments will be conducted at least one time as part of CVS Health's third party oversight program and thereafter, at a frequency determined by Global Security.

w. Security Testing Credentials (ACS-1130):

All applications (regardless of platform) must have test credentials available for security testing in all environments (including Production). The Company has contractual and regulatory directives that require many applications to undergo security testing.

13. Web Security Requirements

a. A consistent character set shall be defined for website's output HTML.
b. Applications shall not leak Confidential Information or Sensitive Information to Third Parties.
c. The referrer HTTP request header field shall not be used to make security decisions.
d. Disable Autocomplete.

For HTML fields that capture Confidential Information, Sensitive Information, or PCI Data the AUTOCOMPLETE attribute shall be set to OFF to prevent browsers from caching the data locally.

e. Disable Default Accounts.

The Application shall not have any default Service Account IDs active.

f. Disallow "Remember Me" Functionality.
g. Logout Users Securely.

i. Logout functionality shall be implemented.

ii. A logout button shall be included on every view and not just the main page or screen.
iii. Logout shall abandon or close the Server session and clear any cookies left on the client Application (browser, etc.).

h. All Extraneous and Default Content from Web Servers shall be removed.
i. Disable Directory Listing on web Servers.
j. Disable Dangerous HTTP Methods.

i The HTTP TRACE, HTTP HEAD, and HTTP CONNECT shall be disabled.
ii The HTTP PUT, HTTP DELETE, and HTTP PATCH methods shall be disabled unless required to support a RESTful Service interface. Strong input validation or other controls shall be in place to ensure that arbitrary content cannot be uploaded or deleted via these methods.

k. Unnecessary Services shall be removed from the Web Server Host. All Ports, services, and similar Applications installed on a computer or Network Systems, which are not specifically required for business functionality, shall be are disabled or removed.
l. The HTTP "Server" Response-Header shall be suppressed.
m. The web Server shall be configured to avoid Content Spoofing attacks on error pages.

14. Mobile Application Security

a. Push Notification (ACS-912):

i. Any mobile application that leverages push notifications, must implement controls to assure that no Restricted, Confidential, or Proprietary data, as outlined in the ATS-501: Information Classification Policy, is transmitted via push notification.
ii. Push notifications should not be leveraged to trigger action against a user's account without their interaction or consent.
iii. Push notifications for employee applications must be appropriate based on job role.
iv. Once user leaves a given role, they should no longer received push notifications that are relevant only to their previous role.
v. Once user leaves a role, it should not get push notification from applications tied to that role.

b. Version Update Notification (ACS-909):

i. All mobile applications deployed for either consumer or employee use should be capable of checking for new version availability upon launch.
ii. Applications should be capable of alerting users of the availability of a new version.
iii. In the event that a new version is being released to address a security vulnerability or defect, the application must have a means of disabling access and forcing the user to upgrade to the latest version.

c. Secure Publication, Deployment and Connectivity for Employee Mobile Applications (ACS-910):

i. Mobile applications developed for use by employees must leverage secure deployment and connectivity platforms provided by Global Security. This standard applies to both

applications developed by internal development teams, and those procured from third parties.

ii. Employee applications must leverage the provided privatized Appstore platform (Apperian) for deployment, and must establish secure communications to the Company via the Mocana/Atlas platform.

iii. Employee applications must also leverage security functionality provided by the Global Security provided security platform (Mocana), including jailbreak/root detection, DLP, and other controls, as deemed appropriate based on the classification of data stored on the device.

d. Application Certificate Pinning (ACS-911):

i. Mobile applications that process or store Restricted, Confidential, or Proprietary data as denoted in the ATS-501: Information Classification Policy, must leverage certificate pining to assure the integrity of Transport Layer Security (TLS) connectivity between the client and other resources.

e. Mobile Application Anti-Tamper Protection (ACS-907):

i. All mobile applications that display, store, or process Restricted, or Confidential data, as defined in the ATS-501: Information Classification Policy, must implement approved Global Security anti-tamper controls for all application components.

ii. Third party developed applications may leverage the Global Security provided solution, or may seek approval for an alternative technology.

f. Mobile Application Signing (ACS-908):

i. All mobile applications must be signed prior to release via application stores.

ii. Application signing must adhere to the standards outlined below:

a. Mobile applications may only be signed with keys or credentials officially assigned by the Company or its affiliates.

b. Keys or signing credentials must be controlled by the Company's employees or service provider employees only. Third party vendors must not have access to the app store release / distribution process.

c. Separation of duty must be enforced. Production code must not be signed by development groups or their managers (E.g. development groups) and must not hold the production cryptographic keys.

d. The signing party must ensure that the code to be signed is the valid source code.

e. The signing party must ensure that the key is properly secured.

f. The signing party must verify that the appropriate security reviews have been completed.

g. Separation of duties is required in application publication. The mobile application client code must not be uploaded to any marketplace app store by the development group directly.

g. Mobile Application Code Obfuscation and Reverse Engineering Protection (ACS-906):

i. All mobile applications that display, store, or process Restricted, or Confidential company data, as defined in the ATS-501: Information Classification Policy, must obfuscate code for the following subset of features:
   a. All application functionality that is leveraged to provide functional or non-functional security controls
   b. All invocations of security controls
   c. Components containing intellectual property, or restricted/ confidential/ proprietary CVS Health data.

ii. Third party developed applications may leverage the Global Security provided solution, or may seek approval for an alternative technology.

h. Mobile Application Binary Retention/Archiving (ACS-4712):

i. All mobile application teams who develop custom mobile applications must maintain binaries from last forced user upgrade. Retention is required to aid in forensic analysis of mobile applications following a security event and to assure appropriate retention of intellectual property.

i. Mobile Development Framework (ACS-1117):

i. All mobile applications development teams that choose to use a cross-platform mobile development framework and platform (E.g. Kony, PhoneGap) must address all high risk security items or findings from an Information Security review prior to deployment. Any platform selected must meet all mobile app security controls and also undergo an Enterprise Architecture review prior to usage.

j. All mobile applications should adhere to the following control standards:

ii. Mobile Static Analysis (ACS-1150)
iii. Mobile Binary Static and Dynamic Analysis (ACS-1151)
iv. Protection of Enterprise Data on Mobile Devices (ACS-914)

15. Publically Facing Applications

a. A publically facing Application is one which exposes an interface to two or more of its components within the CVS Health Network over a public Network such as the internet (as opposed to a private connection such as a B2B VPN or MPLS). Such an interface may include but is not limited to:

- Web Applications
- Web Service APIs
- File Transfer Interfaces (e.g., SFTP)
- Electronic Data Interchange Interfaces (e.g., EDI, AS2, Hl7, etc.)

i. Internet facing applications hosted at CVS Health data centers shall undergo periodic scanning for security vulnerabilities in Pre-PRODUCTION environments.
ii. Any publically accessible application must be subjected to Dynamic Scanning in pre-PRODUCTION prior to deployment of the application to PRODUCTION.
    i. The Dynamic Scan must be performed on an annual basis as a minimum. See Section 11.d
    ii. The Dynamic Scan will be performed by the Software Security Group or an approved Third Party.
    iii. Any Vulnerabilities found in the Dynamic Scan shall be remediated per the schedule identified in Section 11.a.
    iv. The Application owner is responsible for assuring an annual AIR assessment is completed (in Archer) and accurately reflects that **Application Internet Facing** is set to Yes.
    v. The Application Owner is responsible for funding as required to remediate findings from all Security Testing activities.

b. Application Security Standards (Secure Development Lifecycle SDLC) (ACS-6246)

This control standard applies to all internally developed applications, to include Cloud.

All high risk applications and systems shall perform threat modeling.

All applications shall be tested for vulnerabilities via the existing Global Security (GS) approved methodologies:

- Static source code analysis
- Dynamic application scanning, and
- Penetration testing, such as: SSG Ethical Hacking Team and/or 3rd Party Testing (Synopsis, Bugcrowd, etc).

All security defects discovered via static or dynamic analysis, as well as penetration testing shall be remediated in accordance with ACS-1111. (Also in section 11.a above.)

All open source components and third party libraries shall be scanned for vulnerabilities and managed throughout their lifecycle (inventory, patching, etc.)

Containerized applications shall leverage configuration management/lifecycle capabilities using GS approved tools, in accordance with CP-5011167.

Web application firewall capabilities shall be deployed to high risk applications using GS approved tools.

REFERENCES

1. OWASP Guide to Building Secure Web Applications and Web Services 2.1 (DRAFT 3): (http://www.owasp.org/index.php/OWASP_Guide_Project)

2. RFC 2617 - HTTP Authentication: Basic and Digest Access Authentication: (http://www.ietf.org/rfc/rfc2617.txt)
3. RFC 2965 - HTTP State Management Mechanism: (http://www.faqs.org/rfcs/rfc2965.html)
4. Mitigating Cross-site Scripting With HTTP-only Cookies: (http://msdn2.microsoft.com/en-us/library/ms533046.aspx)
5. Preventing SQL Injection in Java: (http://www.owasp.org/index.php/Preventing_SQL_Injection_in_Java)
6. OWASP Top 10: (http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)
7. What is HTTP TRACE? (http://www.cgisecurity.com/questions/httptrace.shtml)
8. Understanding Malicious Content Mitigation for Web Developers: (http://www.cert.org/tech_tips/malicious_code_mitigation.html)
9. Hypertext Transfer Protocol – HTTP/1.1: (http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.38)
10. US-DHS Build Security In – Design Principles - Fail Securely: (https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/principles/349.html)
11. US-DHS Build Security In – Guidelines – Use Well-known Cryptography Appropriately and Correctly: (https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/guidelines/334.html)
12. SANS Top 25 Most Dangerous Software Errors: (http://www.sans.org/top25-software-errors/)
13. Setting the HTTP charset parameter: (http://www.w3.org/International/O-HTTP-charset#scripting)

RECOMMENDED READING

1. The Absolute Minimum Every Software Developer Absolutely, Positively Shall Know About Unicode and Character Sets (No Excuses!): (http://www.joelonsoftware.com/articles/Unicode.html)
2. Mask Your Web Server For Enhanced Security by Joe Lima and Thomas Powell (http://www.evolt.org/node/60160)
3. NIST Guidelines on Securing Public Web Servers – Special Publication 800-44, Version 2.

POINT OF CONTACT

This Document was prepared by the Software Security Group. For questions or comments, please contact the Software Security Group.

DEFINITIONS

All defined words listed below are displayed throughout the Document with initial capitals, except for acronyms. Please refer to Information Governance Definitions Document, CIST-0061 for definitions of these defined words:

Agent; Application; Authorization; Cardholder Data (CHD); Cryptographic Hash; Custom Code; Database; Document; Email; Internal User Accounts; Change Control Board; Change Management; Confidential Information; CVS Health; Database; HTTP; HTTPS; LDAP; Management; Minimum Necessary; Operating System (OS); OWASP; Payment Card Industry

(PCI) Data; PHI; PII; POST; Privileged Access; Production; Risk; Sarbanes-Oxley Act of 2002 (SOX); Secure Coding; Sensitive Information; Server; Special Account ID; SQL; System; Third Party; Token; URL; User; User ID Enumeration Attacks; Username; Vendor; Vendor Supplied Applications; Vulnerabilities; Workstation.

REVIEW AND REVISION HISTORY

Date	Revision No.	Reason for Change	Sections Affected
10/20/10	1.00	New Enterprise version.	All
10/25/11	2.00	Updated definitions and terminology to reflect current standards.	All
03/07/12	3.00	All sections have been revised to reflect best security practices and align with Hi-Trust Framework. Included section on REMEDIATION.	All
03/07/13	4.00	Web Scans shall be performed annually, at a minimum. Secure Code Reviews shall be performed annually, at a minimum to ensure that Application Vulnerabilities are identified. Session Timeout periods rewritten based on Archer recommendation.	11.f 11.i. 6.g
02/05/14	5.00	Included under Web Server Security: The web Server shall be configured to avoid Content Spoofing attacks on error pages See Appendix B for PCI-DSS 3.0 clarifications to Section (s) 5, 6 and 11.	13.g 5, 6 and 11
05/07/14	6.00	Removed control 11.q - As part of the Change Management process, secure code review results shall be reviewed and approved by Application Development Managers in accordance with CVS Health Change Management Standard, CITD-0020.	11.q
06/04/14	7.00	Added control 11.q due to PCI-DSS 3.0 requirement 6.3.2: Secure Code-Review results shall be reviewed and approved by Management prior to release.	11.q
08/29/14	8.00	For public-facing web Applications, Secure Code Reviews shall be performed annually to ensure that Application Vulnerabilities are identified. All other Applications shall be reviewed when changes occur in accordance with CITD-0020 Change Management Standard. **PCI-DSS 3.0 requirement 6.6.	11.n
09/16/14	9.00	Reverted control in Section 11.n back to previous requirement as per CISO request: Secure Code Reviews shall be performed annually, at a minimum to ensure that Application Vulnerabilities are identified.	11.n
11/21/14	10.00	n. All Applications shall be reviewed when changes occur in accordance with CITD-0020 Change Management Standard. Additionally, Secure Code reviews shall be performed and recorded on annually for Applications that contain any of the following: i. Regulated Data ii. Is Externally-facing iii. An Application Exposure Risk Rating of 3 or greater Removed Section 7.c (Secure Socket Layer), as it is redundant with our Encryption and Key Management Standard which is referenced.	11.n 7.c
12/11/14	11.00	Added language for clarification, Section 11.o Secure Code reviews are not required on Vendor Supplied Applications.	11.o

03/02/15	12.00	Section 4.b Added: Applications shall not use SSN to authenticate.	4.b
04/22/15	13.00	Included Medium in the following: Vulnerabilities identified as Medium to High and above shall be remediated prior to deployment into Production. Removed reference to SSL in Section 8.b.5.	11.p 8.b.5
12/02/15	14.00	Added the following language to SCOPE: including Vendor hosted, Vendor managed, and Vendor Supplied Applications. Delete Section 11. o.	SCOPE Section 11.0
01/15/16	15.00	Added language for clarification, Section 10. j. i. This section now reads: The Application shall deliver its login form to the User's browser over HTTPS and shall submit the login form back to the server using the POST method.	Section 10. j. i
03/11/16	16.00	Added language to allow Enterprise level SSO solutions to share cookies between applications. This section now reads: To prevent session hijacking, the Application shall protect session identifiers in transit. The Application shall generate the session IDs using strong, non-predictable algorithms. Session identifiers shall not be stored in persistent cookies or in hidden fields. Instead, session IDs shall be stored in non-persistent cookies with the secure flag enabled. Session cookies shall be configured to restrict distribution beyond the Application by using the domain and path attributes in the cookie properly. NOTE: While cookies generated by an Application shall use domain and path attributes to restrict distribution beyond the Application, it is permissible for Enterprise level SSO solutions like SiteMinder to share cookies between applications. An application's individual SiteMinder cookie configuration shall include the setting of the Secure Flag and the HTTP Flag to avoid the sharing of sessions across unprotected resources.	Section 6. d. i.5)
03/22/16	17.00	Added language from PCI DSS V.3.1 for clarification.	Section 3.h.
12/13/16	18.00	Added new "Secure Coding Requirements" section. Deleted old section content in its entirety and added new content. Added new "Secure Code Reviews" section. Updated section content to reflect current processes. Changed the list level of this requirement. Updated content and added clarity to this requirement.	Section 11.a. Section 11.j. i.-v. Section 11.l. Section 11.n. o. Section 13.c.i. Section 13.c.ii.
02/10/17	19.00	Deleted the content that states "that contain any of the following: i. Regulated Data ii. Is Externally-facing iii. An Application Exposure Risk Rating of 3 or greater" Deleted the statement "based upon Application criticality." Add Compliance section.	Section 11 m. Section 11.j.ii Section 14.
03/16/17	20.00	Added new first paragraph to the Compliance section.	Section 14.
06/20/17	21.00	Added to the end of the sub section statement "Encryption and Key Management Standard, CIST 0018 and shall utilize a FIPS 140-2 Validated Cryptographic Module. Permitted algorithms, key strengths and cryptographic modules are identified in the Encryption and Key Management Procedure, CIST-0111." Added new subsection. Removed the words	Section 7. a. Section 7. b., Section 10. o. iv.

		"technically feasible" and replaced it with "provided by the authentication mechanism."	
11/29/17	22.00	Deleted subsection that stated: "After the User logs in, the Application shall display their most recent login time and location and also the number of invalid login attempts made since then if provided by the authentication mechanism."	Section 10. o. iv.
08/29/18	23.00	Added a new bullet to address the prohibiting of hard coding password. Added new section to address Publically Facing Applications	Section 2.d Section 15.
10/25/18	24.00	Added "Where development is outsourced, change control procedures to address security are included in the contract(s) and specifically require the developer to track security flaws and flaw resolution within the Application, System, component, or service and report findings to CVS Health defined personnel or roles" from HiTrust CSF v9.1 Section 10.k. Level 2. Added "Application Managers shall ensure that all proposed Application development and System changes are reviewed to check that they do not compromise the security of either the Application or the development, testing, and Production operating environments. Further, project and support environments are strictly controlled by the appropriate Technology or Application Owner." from HiTrust CSF v9.1 Section 10.k. Level 2. Added "Network Infrastructure shall ensure the storage of PCI-DSS Data is located outside the publicly accessible environments (e.g., on a storage platform existing on CVS Health's intranet" from HiTrust CSF v9.1 Section 10.k. Level 2. Added "CVS Health shall develop, a documented, control System for implementations and configuration Management plans" from HiTrust CSF v9.1 Section 10.k. Level 2. Added "Access rights to Application functions shall be limited to the minimum necessary in all environments" from HiTrust CSF v9.1 Section 1.v. Level 1. Added "Multi-factor authentication methods shall be used in accordance with CVS Health's Password and Credential Standard, CIST 0010." from HiTrust CSF v9.1 Section 1.q. Level 2. Added "and shall be encrypted during transmission and in storage on all System components" from HiTrust CSF v9.1 Section 1.d. Level 2. Added "Health shall limit authorization to privileged accounts on information Systems to a pre-defined subset of Users" from HiTrust CSF v9.1 Section 1.c. Level 3. Added "Attempted unauthorized remote connections to the information Systems shall be monitored and reviewed and appropriate action shall be taken if an unauthorized connection is disco	Section 1.c. Section 3.a. Section 3.b. Section 3.d. Section 3.g. Section 4. Section 4.a. Section 5.a.ii. Section 5.b.i.5) Section 6.d.ii. Section 6.e.iv. Section 7.b. Section 10.a.ii. Section 10.a.iii. Section 10.a.ix. Section 10.b. Section 10.i. i. Section 10.i. iii. Section 10.j.i. Section 11.j. Section 11.n.i. Section 13.d. Section 14. Section 15.b. Section 15.b.iv.
02/17/20	25.00	Minor edits, corrected grammar or spelling error(s), or updated definitions	Minor edits, corrected grammar or spelling error(s), or updated definitions

Appendix A

Compliance Effective Date

Section	Control Statement	Effective Date
6.d	a. Broken authentication and session Management i. Corroborate that broken authentication and session Management are addressed via coding techniques that commonly include: (1) Flagging session Tokens (for example Cookies) as "secure" (2) Not exposing session IDs in the URL (3) Incorporating appropriate time-outs and rotation of session IDs after a successful login.	**6/30/2015**
4.b	b. Applications shall not use SSN to authenticate	**9/1/2015**