HBS Security Guidelines [OWASP compliant]

PRD Requirement Reference	ID	Description	Compatibility		Comments	References
			HB Desktop R7.5	HB Desktop PAT R7.5.1		

HBT-SEC-0100 - Authentication	V1.1	All pages used by the Application require authentication of the user prior to access (also NIST Control AC-2, AC-14)	Yes	Yes	Once after successful authentication via username-password validation, a unique secured random id is generated per user login. This unique id is later set in the current session. Henceforth every request for a page within the session, carry this unique id along, and the same is validated across the current session on the server. In case, if the unique id is not present or invalid then user gets redirected to the login/error page. Thus all the subsequent pages after login are authenticated.	

V1.2	All password fields (e.g. for login and password modification) shall not echo the user's password when it is entered and the fields shall not auto-complete.	Yes	Yes	This is implemented in all HBS Applications (even legacy)	
V1.3	All Authentication controls shall fail securely to ensure attackers cannot log in. This means to disguise which part of the credentials failed to authenticate as well as redirecting the user to the same login page without any additional information about the system structure behind the login page	Yes	Yes	The failed authentication error messages in all HBS applications are in compliance and gives no indication of why the authentication failed	
V1.4	User credentials shall not be traversed unencrypted (NIST Control AC-17)	Yes	Yes	All the applications (internal or on the public network) are secured using SSL (Secured Socket Layer).	

					1. Browser connects to a application secured with SSL (https). Browser requests that the server identify itself. 2. Server sends a copy of its SSL Certificate, including the server's public key. 3. Browser checks the certificate root against a list of trusted CAs and that the certificate is unexpired, unrevoked, and that its common name is valid for the website that it is connecting to. If the browser trusts the certificate, it creates, encrypts, and sends back a symmetric session key using the server's public key. 4. Server decrypts the symmetric session key	

				using its private key and sends back an acknowledgement encrypted with the session key to start the encrypted session. 5. Server and Browser now encrypt all transmitted data with the session key.	
V1.5	Verify that forgot password and other recovery paths do not send the existing or new passwords in clear text to the user.	Not Applicable	Not Applicable	Password recovery is not supported in any HBS application at this time. This feature is currently not in scope for all applications.	
V1.6	Username enumeration shall not be possible via login or password reset.	Yes	Yes	The failed authentication error messages in all HBS applications are in compliance and gives no indication of why the authentication failed.	https://www.owasp.org/index.php/Testing_for_User_Enumeration_and_Guessable_

V1.7	No default passwords (e.g. admin passwords of application frameworks) shall be used anywhere in the application	Yes	Yes	None of the HBS application uses a default password if not provided.	
V1.8	A mechanism shall be in place to protect against vertical brute force attacks (see separate Requirement HBT-SEC-0400)	No	Yes	These features are handled as part of new HB Tools and HB Desktop starting R7.5.1. User account lock on repeated failed attempts.	See https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks HBT-USM-0220 - User account soft lock on repeated failed logins. PAT-LOGIN-0360 - User lockout on repetetive failed login attempts
V1.9	Authentication controls shall be enforced on the server side of the application	Yes	Yes	All authentication is done on the server side for all HBS applications as secured services.	

V1.10	Verify password entry fields allow or encourage the use of passphrases, and do not prevent long passphrases or highly complex passwords being entered, and provide a sufficient minimum strength to protect against the use of commonly chosen passwords.	Yes	Yes	All the password rules (in Tools and HB Desktop) are implemented as defined in the customer's requirement. The current password rule set defined in the system are not that complex, and the change in the same needs to be adressed by the PRM or customer.	Clarification Required from PRM on this. As per Cigital, our current password rules set (i.e with letters, numbers, etc) defined in the account preference are not strong enough especially in Non-VA mode. Should we re-consider it for PAT R 7.5.1 or are we okay with it. This might require change in existing data model and design. It looks like the currently used password rules set was enforced by customers (VA)

V1.11	Verify all account management functions (such as registration, update profile, forgot username, forgot password, disabled / lost token, help desk or IVR) that might regain access to the account are at least as resistant to attack as the primary authentication mechanism.	Not Applicable	Not Applicable	Feature to regain access to the existing account is not supported in any of the HBS appications (i.e Desktop, Tools etc). If the existing password is disabled/forgotten then the user needs to contact customer support for futher assistance.	
V1.12	User credential changes shall be handled as resistant to attacks than the main authentication on login.	Yes	Yes	HB Desktop users can only change their passwords after successful login to the application.	
V1.13	User passwords shall expire after a period of time (HBT-SEC-0410)	Yes	Yes	This is enforced via a password rule that can be configured at the account level. Need SOP for customers to enforce this rule.	HBT-ACT-0250 Edit Account - Password Rules Settings

V1.14	All authentication decisions (success, fail, softlock) shall be logged in an audit trail (Nist Control AC-2).	Partial	Yes	Login success entry is logged in LOGIN_ENT_SUCCESS_LOG. Login failure entry is logged in LOGIN_FAILURE_LOG. Failure in case of repeated attempts to login will be implemented as part of HB Desktop R7.5.1	Implemeted as part of R7.5.1
V1.15	All passwords are salted with a salt that is unique to the user (e.g. user ID).	Yes	Yes	A unique salt is generated and mapped for every password and stored in the database.	

V1.15	(2) All passwords shall be hashed before storage using SHA256.	Partial	Yes	Current implementation (from legacy) supports only MD5. Migration to the SHA-256 will be considered as part of HB Desktop R7.5.1	Implemeted as part of R7.5.1. Old users passwords if hashed with MD5 initially will be migrated to SHA 256 algorithm on the very first login to the system (HBS R 7.5.1) and therafter the same will be compared during the authentication process. This desogn doesnt impact any change in user passwords.
V1.16	Authentication credentials used to access external shall be encrypted/persisted in a secure location (not in source code)	No	Yes	All the sensitive data and user credentials for secured resources access (such as database etc) stored in the property files need to be encrypted. This is implemented in HB Desktop R7.5.1 release by using 'Jasypt' framework.	Implemented as part of R7.5.1. An encryption utility script is provided to aid the deployment team to get the encrypted passwords before being mapped in the db.properties.

V1.17	Recovery of Authentication information (e.g. forgot passwords, activation) shall provide time limited access (not applicable for HB Tools)	Not Applicable	Not Applicable	Password recovery is not handled through web and is not currently supported feature of HB Dektop/Tools. In case of forgotton password, user needs to contact the support team for further assistance.	
V1.18	Recovery of authentication information shall not lock out or disable the account automatically. (Not applicable for HB Tools)	Not Applicable	Not Applicable	Password recovery is not handled through web and is not currently supported feature of HB Dektop/Tools. In case of forgotton password, user needs to contact the support team for further assistance.	

	V1.19	The system shall not use security questions for authentication recovery (not applicable for HB Tools)	Not Applicable	Not Applicable	Password recovery is not handled through web and is not currently supported feature of HB Dektop/Tools. In case of forgotton password, user needs to contact the support team for further assistance.	
	V1.20	User shall not reuse previous passwords (for details see HBT-SEC-0410).	Yes	Yes	This feature is supported across all HBS applications.	This again is dependent of the defined rules set in the account preferences.

HBT-SEC-0110 - Session Management	V2.1	The framework's default session management control implementation shall be used	Yes	Yes	The Play framework is stateless and doesn't implicitly provide the session management. But it provides an option to handle session explicitly is used for HBS and Tools. We store a unique securely generated session id in the play provided session (cookie) for each login. The same session id is mapped/stored in the time dependent play cache (managed by play server) as a key and mapped to the corresponding user logged in details.	

V2.2	Sessions shall be invalidated on user log-out	Yes	Yes	Session is invalidated/cleared using the framework support and all the data in cache linked to the corresponding session id is removed on user log-out or timeout.	
V2.3	Sessions shall time out after after a configurable amount of time of inactivity (between 15 and 60 minutes) (NIST Control AC-2, AC-11)	Yes	Yes	This is implemented as defined and the session gets timed out as per the configured value in PLAY config. In case of any inactivity for the defined time limit, cache data corresponding to the logged in user's session-id is automatically removed by framework (PLAY)	
V2.4	The application shall provide access to log-out functionality from all pages (NIST Control AC-12)	Yes	Yes	This feature is supported across all HBS applications.	

V2.5	The session ID shall not be disclosed other than in cookie headers (particularly urls, error messages or logs)	Partial	Yes	Session id is stored in the cookie (play provided) and the the server cache. Application never displays this information in the URL or error messages. But we found few traces of this data in the log files. Logs needs to be cleaned up in PAT R7.5.1.	This is already taken care in the current implementation. Corresponding to every session, a secured token along with the session id is generated. Only session tokens are printed in logs. This needs to be re-verified across all the features in HB Desktop R 7.5.1.
V2.6	Session IDs shall be changed or cleared on log-out.	Yes	Yes	On user logout or session timeout, session id is removed from the session/cookie and all the data related to the session present in the play cache is cleared.	

V2.7	Authenticated session tokens using cookies shall be protected by "HttpOnly"	Yes	Yes	This feature is in-built provided by PLAY framework. The cookie created for the storing the session tokens is managed by PLAY. Cookies are signed with a secret key so the client can’t modify the cookie data (or it will be invalidated)	
V2.8	Authenticated Session tokens using cookies shall be protected with the "secure attribute" and strict transport security headers shall be present	Yes	Yes	This feature is in-built provided by PLAY framework. The cookie created for the storing the session tokens is managed by PLAY. Cookies are signed with a secret key so the client can’t modify the cookie data (or it will be invalidated)	

V2.9	Session ID shall be changed on log-in to prevent session fixation.	Yes	Yes	A unique secured random number is generated for every new log-in anytime. (Secured random generation code needs to be enhanced.)	
V2.10	Session ID shall be changed on re-authenticati on.	Yes	Yes	A unique secured random number is generated for every new log-in anytime. (Secured random generation code needs to be enhanced.)	

V2.11	Only session IDs generated by the application framework shall be recognized as valid.	Yes	Yes	Unique session id's generated is stored in a static hashmap on the server perm memory. Every request containing session id is validated against the hashmap contents. If the match is found, user is allowed to proceed else user is redirected to the login page.	
V2.12	Authenticated session tokens shall be sufficiently long and random	Yes	Yes	Java secured random API is used to generate the session tokens/ids on the server side and later stored in the cookie and cache.	

V2.13	Authenticated Session Tokens using cookies shall have their path set to an appropriate restrictive value for the site. The domain cookie shall not be set.	Yes	Yes	The cookie created for the storing the session tokens is managed by PLAY. Cookies are signed with a secret key so the client can't modify the cookie data (or it will be invalidated).	Cigital : The path gives the chance to specify a directory where the cookie is active. Usually the path is set to /, which means the cookie is valid throughout the entire domain. Cookie path should be set to the application restrictive path. Bosch : Play framework support. No change required.

	V2.14	Duplicate concurrent user sessions from different machines shall be prevented (NIST SP 800-53 800-53 Control AC-10).	Yes	Yes	Same user cannot login through 2 different machines concurrently. Initially the second user trying to login is asked for a confirmation on whether he wants to logout from his already logged in session. If yes, then a new session is created and first user is logged out. This is handled using a static hashmap maintained for already logged users with username as a key.	
HBT-SEC-0120 - Access Control	V3.1	All functions in the system can be accessed only by authorized and authenticated users with the appropriate rights.	Yes	Yes	This is implemented using the deadbolt plugin for the PLAY framework. All the functions are provided access based on the user roles and permissions.	

V3.2	Users can only access secured urls for which the are authorized and authenticated	Yes	Yes	All HBS applications are SSL enabled. All the client-server communication is made over HTTPS protocol.	
V3.3	Users shall only be able to access secure files, if they are authorized (not applicable for HB Tools, since no secure files are used)	Not Applicable	Not Applicable	No secure files are ever accessed by the user directly.	Currently no feature in HBS enables access to the files on the server through web UI.
V3.4	Direct Object references are protected so that only authorized users can get access	Yes	Yes	Object references not exposed or sent over the URL as a plain text. References such as patient id's etc are encrypted and sent to the server.	
V3.5	Directory browsing is disabled unless desired for business reasons.	Not Applicable	Not Applicable	No function provides directory browsing.	

V3.6	The system shall ensure, that users do not access protected data for which they have no authentication	Yes	Yes	This is implemented using the deadbolt plugin for the PLAY framework. All the functions are provided access based on the user roles and permissions.	
V3.7	Access Control shall fail securely.	Yes	Yes	Failure of any access control operations display a valid error message describing of the unauthorised access.	
V3.8	All access control rules shall be applied on presentation layer as well as server side	Yes	Yes	Access control is handled at the controller level for every action defined. Few operations infact do check for some specific permissions for different conditions at the server side as per the defined	
V3.9	Access controls shall not be manipulable by users.	Yes	Yes	Access control check is done on the server side and no user interaction is involved in the process.	

V3.10	All access controls shall be enforced on the server side.	Yes	Yes	Access control is handled at the controller level for every action defined. Few operations infact do check for some specific permissions for different conditions at the server side as per the defined requirements	
V3.11	Access control decisions shall be logged (NIST SP 800-53 800-53 Control AU-2)	No	Yes	Access control decisions are logged only in the log files.	Cigital : If we authorization is denied, log it to file as of now. Later can be invested to check with Splunk integration. Bosch : Unauthorized access are getting logged into the file.
V3.12	The Application or framework shall generate and verify strong random anti-CSRF tokens unique to the user as part of all high value transactions.	No	Yes	Play supports CSRF guard protection and is applied for every request. The implementation is In-Progress.	

	V3.13	The system shall protect against aggregate or continuous access of information requiring access control (Lower priority compared to previous controls)	Yes	Yes	All access controls currently implemented is managed deadbolt frameowrk and doesn't allow access, if the user is not authorised. As of now, this is not of high priority.	Cigital : Velocity checking on access control is good option. If the number of attempts to access an authorized resources (even from an authenticated user) exceeds defined number, then just log him out. Bosch : Need to investigate more on implementation perspective if taken in consideration. Not of that priority as all the necessary check is in place to validate the access control and deny access if user is not authorised.
HBT-SEC-0130 - Input Validation	V4.1	The runtime environment shall not be susceptible to buffer overflows or controls additional controls prevent buffer overflow.	Yes	Yes	Proper input validation is performed on all the request parameters.	HBT-SEC-0130 - Input Validation

V4.2	The runtime environment shall not be susceptible to SQL Injection or shall provide security controls to prevent SQL Injection	Yes	Yes	Cigital didn't find any SQL injection within the system. Application is supported by Hibernate internally.	
V4.3	The runtime environment shall not be susceptible to Cross-Site-Scripting (XSS) or shall provide security controls to prevent Cross-Site-Scripting.	Yes	Yes	All input http request parameters are passed through an XSS filter to check for any vulnerablities.	
V4.4	The runtime environment shall not be susceptible to LDAP Injection (if applicable)	Not Applicable	Not Applicable	Not applicable to HB Desktop	
V4.5	The runtime environment shall not be susceptible to OS Command injections or shall provide security controls to prevent OS Command injection.	Not Applicable	Not Applicable	V32 is being eliminated from R7.5.1. No HBS application should have this vulnerability.	

V4.6	Input validation failures shall result in in rejection of the input	Yes	Yes	All the input fields presented to the user are validated both on client and server and also passed through a XSS filter to check for CSRF attack.	
V4.7	Input validation or encoding routines shall be performed and enforced on the server side of the application.	Yes	Yes	Input validations and Output encoding is handled as part of the current implementation. Input fields are valiated across the XSS filter before being validated for across business rules set.	
V4.8	The system shall properly escape output to HTML for untrusted data	Yes	Yes	Play supports scala programming in HTML. This supports mapping POJO to the HTML form fields directly. Output encoding is being incorporated.	

V4.9	The system shall specify a character set (e.g. UTF-8) for all sources of input	Yes	Yes	Application uses char set UTF-8 as defined. Meanwhile also supports ISO-8859-1	
V4.10	The system shall canonicalize input data for all downstream decoders or interpreters prior to validation	No	Yes	All the input request params are canonicalized and validated across a validation pattern.	https://www.owasp.org/index.php/Canonicalization,_locale_and_Unicode
V4.11	If the Application allows automatic mass parameter assignment (also called automatic variable binding) from the inbound request to the a model, it shall ensure that security sensitive fields such as role or password are protected from malicious automatic binding	Not Applicable	Not Applicable		
V4.12	The system shall provide defense against HTTP	Yes	Yes	Play server assigns the value of the last occurrence of same name	https://www.owasp.org/index.php/Testing_for_HTTP_Parameter_pollution_

		Parameter pollution attacks.			HTTP parameter currently as tested. According to the OWASP, if existing input validation and other security mechanisms are sufficient on single inputs, and if the server assigns only the first or last polluted parameters, then parameter pollution does not reveal a vulnerability. If the duplicate parameters are concatenated, different web application components use different occurrences or testing generates an error, there is an increased likelihood of being able to use parameter pollution to trigger	

					security vulnerabilities.	
HBT-SEC-0140 - Cryptography at rest	V5.1	The system shall implement all cryptographic functions used to protect secrets from the user on the server side.	Yes	Yes	All the user credentails and other sensitive information like (patient id's) are passed encrypted. Hashing is used to store the passwords along with the salt.	
	V5.2	All cryptographic modules shall fail securely, in case the fail.	Yes	Yes	In case of failure, proper error message is display to the user without mentioning the actual cause or any sensitive data.	
	V5.3	All access to master secrets shall be protected from unauthorized access.	Yes	Yes	Encryption keys are not stored on the file. Its randomly generated on server starup and stored and a static variable.	

	V5.4	All information generated using randomization (numbers, GUIDs, strings) shall use the cryptographic module's approved random number generator in case the values are intended to be unguessable by an attacker.	Yes	Yes	Java secured random API is used to generate.	
	V5.5	Cryptographic modules used by the Application shall be validated against FIPS 140-2. (Applicable for FISMA compliant applications)	No	Yes	MD5 is currently used for hashing for passwords. These will be modified in R 7.5.1 to SHA - 256.	

HBT-SEC-0150 - Secure Error handling and Logging	V6.1	The Application shall not output error messages or stack traces containing sensitive data, that could assist an attacker (including session id and personal information)	Partial	Yes	All the user displayed error messages are clean and doesn't contain any sensitive data. Log file needs to be relooked for any PHI traces. This will be addressed in PAT R 7.5.1.	
	V6.2	All error handling shall be performed on the application server as a trusted device.	Yes	Yes		
	V6.3	All logging controls shall be implemented on the server side.	Yes	Yes	Complete logging control is done on server side. This includes log4j framework logging and audit logs for specific event like login etc.	
	V6.4	Error handling logic in security controls shall deny access by default.	Yes	Yes	By default system deny access from all the security controls. Example : False is set as deafult for the input validations controls.	

V6.5	Security logging controls (such as authentication) shall log both success and failure events	Yes	Yes	Login success entry is logged in LOGIN_ENT_SUCCESS_LOG. Login failure entry is logged in LOGIN_FAILURE_LOG.	
V6.6	Log events shall include (NIST Control AU-14-2)	Yes	Yes		
	- Time stamp				
	- The originating component creating the audit log)				
	- Indication, if the log event is originating from a security related control if mixed with other events				
	- The user identity of the user causing the event				
	- The source IP address associated with the event				
	- Success or failure of the event				
	- A description of the event				

	V6.7	Logs related to security events shall be protected from unauthorized access.	Yes	Yes	Cigital : OS File system restriction is provided on all the log folders. Bosch : Is taken care by defining permissions by ETS infrastructure team for every user.	
	V6.8	The application shall not log application specific sensitive data, which could assist an attacker (such as session IDs, or personal information such as clean names or PII beyond sole application specific identifiers)	Partial	Yes	Log file needs to be relooked for any PHI traces. This will be addressed in PAT R 7.5.1.	
	V6.9	The logs shall be accessible to log analysis tools for authorized personnel.	Yes	Yes		

HBT-SEC-0160 - Data Protection	V7.1	The Application shall disable client side caching (including auto-complete if not required for business reasons) for all forms containing sensitive information by default. Auto-compelte should be disabled per default.	Yes	Yes	Sensitive information is not cached on client side anywhere.	
	V7.2	All sensitive data shall be sent to the server in the HTTP message body (e.g. URL parameters are not used to send sensitive data).	Yes	Yes	All the sensitive data is passed as POST request to server encrypted.	

	V7.3	All cached or temporary copies of sensitive data sent to the client shall be protected from unauthorized access or shall be purged/invalidated after the authorized user accessed it (e.g. no-cache and no-store cache-Control headers are set)	Yes	Yes	Cookies and cached information is cleared and deleted on logout or session timeout.	
	V7.4	All cached or temporary copies of sensitive data stored on the server shall be protected from unauthorized access or shall be purged/invalidated after the authorized user accessed it.	Yes	Yes	Cached information on the server pertaining to the user is cleared on user logout or session timeout.	

HBT-SEC-0170 - Communications Security	V8.1	The application shall ensure that a path can be built from a trusted Certificate Authority to each Transport Layer Security (TLS) server certificate.	Yes	Yes	CA certified certifcates are used with SSL enabled in play fra ewrk,	
	V8.2	The application shall use valid server certificates.	Yes	Yes	Valid CA certified server certifcates are used.	
	V8.3	The application shall use TLS (most current version of TLS at implementation finish) for all connections that are authenticated or The application shall log TLS connection failures on the back-end	Yes	Yes	TLS [SSL 3.1] is used for all client-server communications.	

	V8.4	The application shall ensure that connections to any external systems involving sensitive information are authenticated	Not Applicable	Not Applicable		
	V8.5	The application shall ensure that connections to external systems that involve sensitive information or functions use a "user" that has the minimum required privileges necessary to perform the function	Not Applicable	Not Applicable		

HBT-SEC-0180 - HTTP Security	V9.1	The application shall accept only a defined set of HTTP Request methods such as GET and POST. Unused methods shall be explicitly blocked	Yes	Yes	All the requests are either GET or POST defined in routes config. Any request sent with request methods other than GET/POST or the one defined in the routes will be blocked implicitly by the PLAY.	
	V9.2	The application shall ensure that every HTTP response contains a content type header specifying a safe character set (such as UTF-8)	Yes	Yes	Charset ISO-8859-1 is used.	
	V9.3	HTTP headers shall be included protecting against click jacking attacks.	Yes	Yes	Configuring all pages within the application to send the XFRAME-OPTIONS header with the DENY or SAMEORIGIN values or using an frame-breaker script in the pages needed protection	https://www.owasp.org/index.php/Clickjacking_Protection_for_Java_EE

	V9.4	HTTP headers shall contain only printable ASCII characters in both request and response.	Yes	Yes	Charset ISO-8859-1 is used.	
HBT-SEC-0190 - Business Logic Security	V11.1	The application shall process all high value business logic in a trusted environment (such as a protected and monitored server)	Yes	Yes	All business process are executed on the server hosted in a secured environment. All communication with the server is over HTTPS.	
	V11.2	The application shall prevent spoofing transactions such as allowing attacker "user A" to process a transaction as victim "user B" by tampering with sessions or replaying sessions, transaction states or user ID	Yes	Yes	Cookies are signed with a secret key so the client can't modify the cookie data (or it will be invalidated).	

V11.3	The application shall be protect against repudiation attacks including verifiable and protected transaction logs, audit trails or system logs.	Yes	Yes	All the user sensitive information identifying the valid system user is encrypted and sent through the POST request method to the server for every action. None of such data is stored in cookie or passed in URL parameter allowing attacker to forge	https://www.owasp.org/index.php/Repudiation_Attack
V11.4	The application shall prevent information disclosure attacks such as direct object reference, tampering, session brute force or other	Yes	Yes	This is duplicate of V1.8 and V3.4	

V11.5	The application shall have sufficient detection and controls to protect against brute force or denial of service attacks	Yes	Yes	Brute force Duplicate of V3.4. DoS attack makes the resources unavailable for valid users and the entire system becomes unresponsive after some time. This may occur because of many programming vulnerability. The DoS attack is highly possible when dealing with XML entities through XML injection. Polluting the XML entity with dump values and falling in an endless loop. Pat/Tools application code is reviewed to verify for any such possibilities. Standard Proxy has taken care of XML injection attack.	https://www.owasp.org/index.php/Denial_of_Service

	V11.6	The application shall prevent "elevation of privilege" attacks such as allowing anonymous users access of secure data or functions or allowing users to access each others detailed or privileged functions.	Yes	Yes	This is implemented using the deadbolt plugin for the PLAY framework. All the functions are provided access based on the user roles and permissions.	http://en.wikipedia.org/wiki/Privilege_escalation
	V11.7	The application shall only process business flow steps in sequential step order with all steps processed in realistic human time. It shall not process out of order, skipped steps, process steps of another user	Yes	Yes	Business implementation addresses this effectively. In web based applications, all the workflows are processed sequentially.	
	V11.8	The application shall ensure that url redirects and forwards do not include unvalidated data.	Yes	Yes		

HBT-SEC-0200 - Files and Resources	V12.1	The application shall canonicalize file names and path data from untrusted sources to eliminate path traversal attacks.	Not Applicable	Not Applicable	Need to be checked when trying to upload file during photo upload feature of user preferences.	
	V12.2	In case files from untrusted sources are used, those files should be scanned by the system's anti virus scanner to prevent upload of known malicious code.	Not Applicable	Not Applicable	System accepts only a set of file extensions like jpeg,gif etc	

V12.3	Parameters obtained from untrusted sources shall not be used in manipulating file names, path names or any file system object without being canonicalized and input validated to prevent local file inclusion attacks.	Not Applicable	Not Applicable	None of the features in HBS applications accepts parameters for usage in requires, include or similar functions. Does this have an impact on local file upload mechanism(e. g photo) in web based application ? Cigital : User input should not be used directly for constructing a file name present on the local server as such. Input should be first input validated and canonicalized. Bosch : Doesn't apply to us.	

V12.4	Parameters obtained from untrusted sources shall be canonicalized, input validated and output encoded to prevent remote file inclusion attacks.	Not Applicable	Not Applicable	This is somewhat similar to the local file inclusion attacks, with the difference of file being injected from a remote server. Like local file inclusion, none of the features in HBS applications accepts parameters for usage in requires, include or similar functions. Cigital : User input should not be used directly for constructing a file name located remotely on the server as such. Input should be first input validated and canonicalized. Bosch : Doesn't apply to us.	

V12.5	Remote IFRAMEs and HTML 5 cross domain resource sharing shall not allow inclusion of arbitrary remote content.	Yes	Yes	Addressed as part of remedy for click jacking attack.	
V12.6	The Application shall store files obtained from an untrusted source outside the webroot.	Not Applicable	Not Applicable	Data received from the files are stored in the db as clob.	
V12.7	The Application server shall by default be configured to deny access to remote resources or systems outside the application server.	Not Applicable	Not Applicable	No feature in the existing system enables file access outside application server.	
V12.8	The application code shall not execute uploaded data obtained from untrusted sources.	Not Applicable	Not Applicable		

	V12.9	In case Flash, Silverlight or other rich internet applications are used, the corresponding cross domain resource sharing configuration shall be configured to prevent unauthenticated or unauthorized remote access.	Not Applicable	Not Applicable	No feature in the existing system supports this feature.	
HBT-SEC-0420 - General Audit Log Requirements		The system shall keep an audit log on selected activities. These activities include at minimum	Yes	Yes	All the authentication (success, failure, softlock) are logged into the audit logs.Other activities are managed by log framework, audit log for such activites will not be addressed due to the performance issues.	
		Authentication events				
		User Management events				
		Execution of SQL reports				
		Modifications of Account preferences				
		Audit logs shall include				

		1. Type of event 2. Date/time of the Event)based on internal system time and mappable to UTC) 3. Result of the event 4. Description of the event 5. Identity of involved users				

Please follow the link to find the update on the Cigital and VA audit issue remediation identified in R7 release.

HBS R7.x Security issue remediation