**Emirates NBD**

**Application Baseline Security Checklist**

**For Vendors**

|  |  |
| --- | --- |
| Document Information | |
| Document Title | Emirates NBD Application Baseline Security Checklist |
| Document Owner: | Group Risk - IT Security |
| Contact Person: | Mr. Harpal Singh |
| Publish Date | May, 2013 |
| Classification | For Vendors |
| No. of Pages | 20 |

|  |  |  |  |
| --- | --- | --- | --- |
| Document History | | | |
| Date | **Version** | **Name** | **Notes** |
| October , 2010 | Version 2.0 | RISK (Harpal Singh) |  |
| October 23, 2012 | Version 3.0 | RISK (Harpal Singh) | Incorporated internal review comments. |
| May 27, 2013 | Version 3.1 | RISK (Dinesh Chauhan) | Minor updates |
| Oct 29, 2014 | Version 3.2 | RISK (Harpal Singh) | Minor updates + changes from GISM |

|  |  |
| --- | --- |
| Project Information | |
| Application Name |  |
| Business Owner |  |
| Business Contact Person |  |
| IT Project Manager/Domain Manager |  |
| In House / Vendor Developed |  |
| Software Vendor / Implementation Partner |  |
| List the departments that will use the system (Also indicate expected number of users in the first 3 months) |  |
| Is the application driven by an enterprise wide licensing arrangement? |  |
| Location of Installation |  |
| Service Type (Please note critical data cannot be hosted on cloud) | |
| Service Type – Internally Hosted |  |
| Service Type – Externally hosted / Cloud based (SaaS, PaaS, IaaS) |  |
|  |  |
| Whether system requires connectivity to any external server? e.g. to download upload signatures/patches etc. |  |
| Purpose of the Application (Brief Description) |  |
| Checklist Completed By |  |
| Approved By |  |

**Note1: Controls specified in the checklist are minimum baseline and may vary during the actual review based on current security best practices, application functionality and business requirement.**

| **S. No.** | **Control** | | | **Compliant**  **(Yes / No / NA)** | **Remarks** |
| --- | --- | --- | --- | --- | --- |
| Configuration Management | | | | | |
|  | Applications invoke and run services with minimal privileges rather than run them at system / administrator level privileges | | | yes |  |
|  | Application User must not have DBA rights on database. | | | Yes | DBO |
|  | Information about the following should not be divulged in responses or error messages (For web application : http response, thick client : responses intercepted)   * Database schema information * Server name * OS / Database name and version * Installed programs , version * Other system component | | | yes |  |
|  | Applications must display customized Error page for any unhandled condition instead of default error pages with detailed error information. | | | yes |  |
|  | Directory listing must be disabled on application and web servers. | | | yes |  |
|  | Sensitive links which must not be indexed by search engines must be listed within robots.txt files. On the other hand, if a critical webpage (e.g. administration panel) is not explicitly linked within the web application, it must not be included within robot.txt files as well. | | | N/A |  |
|  | Application Supports latest stable version of the Web server / Application Servers and other components of the solution. | | | Yes |  |
|  | All HTTP methods except GET and POST must be disabled if they are not in use. | | | Yes |  |
|  | Access to non-public resources (e.g. backup files, development test files) must be restricted for certain users and unnecessary applications (e.g. default web server sites, demo applications) must be removed. | | | Yes |  |
|  | The storage of application data must be located in a separate server from the application’s software libraries and content. Database must be on separate server | | | Yes |  |
|  | Application directories that contain application binaries, configuration files, metadata components, etc must not be world writable. | | | Yes |  |
|  | Names of critical directories like administration panels must not be easily guessable (e.g. admin, administration)and access to administration modules must be restricted to specific IP addresses | | | N/A |  |
|  | HTML comments leaking sensitive information about the directories/files/parameter etc. must be removed from source files that might help an attacker to plan further attacks. | | | Yes |  |
|  | Sensitive files and resources belonging to application domain that may provide reference to hidden files must be removed from the browsable directories and must not be indexed via search engines | | | N/A |  |
|  | Application developed must support stable versions of all popular browsers like Internet Explorer, Mozilla Firefox, and Chrome. Applied security controls must support all major browsers. | | | Yes |  |
|  | Default user accounts must be removed from applications, systems and databases. Default passwords must be changed. | | | Yes |  |
|  | Disable Weak SSL ciphers. Use SSLv3.0 or TLS 1.0 with at least 128 bit key cipher. | | | N/A |  |
|  | SSL renegotiation feature must be disabled against DoS and MITM attacks. Disable support for SSL/TLS renegotiations | | | N/A |  |
|  | Authentication codes (i.e. IDs, passwords, cryptographic keys, etc) must not be stored in clear text format in any of the configuration file or hardcoded in source code. Group standards mandate 3DES or AES for symmetric encryption and use one way hashing algorithms with salt (e.g. SHA 256/512).  *Explanatory Note: Ideally, authentication codes must not be hard-coded into the application. But if done, there must be an encryption-decryption mechanism to do the same. Authentication codes must be obfuscated and stored in the registry or in an encrypted license file and protected using access controls*. | | | Yes |  |
|  | Authentication codes (i.e. IDs, passwords, cryptographic keys, etc) transmitted across the network must be encrypted. For browser based applications this can be achieved through the use of SSL. Group standard mandate SSLv3.0 or TLSv1.0 & SFTP for file transfer.(Indicate the encryption algorithm used) | | | N/A |  |
|  | Passwords transmitted via ODBC, OLE and other internal links or intersystem profiles must be encrypted by implementing ODBC/JDBC over SSL  *Note: OLE implementation is preferred; ODBC to be implemented only where OLE is not feasible* | | | No |  |
|  | Ordinary users (business users) must not have access to application configuration management functions. | | | Yes |  |
|  | Application client must communicate to Database via Application server and should not be communicating directly to database. | | | YES |  |
|  | The traffic between client and server must be encrypted. End to End Encryption is recommended. (Client to server- and Server to Server and Network to Network) | | | No |  |
|  | Secure the permission on application program files. Give full control permission only to application service account and administrator account. Other users and groups must be restricted. | | | Yes |  |
|  | Restrict the permission on user and password file (if present). Store the password in one way hash format with Salt (e.g. SHA - 256/512) or use an algorithm specifically designed for the password storage such as bcrypt, PBKDF2 or scrypt. | | | No |  |
|  | Restrict the permission on database credential file. If database username and password are stored in this file then store the password in one way hash format with Salt (e.g. SHA-256/512) or use an algorithm specifically designed for the password storage such as bcrypt, PBKDF2 or scrypt | | | NO | We are using our own encryption algorithm |
|  | If the application stores database and user credentials in the OS registry, secure the registry as well. Access permission to the sensitive registry hive must be restricted to designated user only and user password must be stored in one way hashed format. | | | N/A |  |
|  | Application must not store sensitive information like user password in database, computer memory or registry in clear text format. It should rather be stored in one way hash format using salted hashing techniques (e.g. SHA 256/512). | | | Yes |  |
|  | Database must not run on a default port. Thick client must use non guessable port to connect the database. | | | Yes |  |
|  | Database must accept the connections only from the application server. | | | Yes |  |
| Authentication | | | | | |
|  | Application must display a message (like the one below) prior to completing the logon process warning the user that unauthorized access may result in prosecution.  "*Access to this application is restricted to authorized persons and programs for authorized purposes only. Unauthorized access is prohibited and is subject to prosecution. Evidence of criminal actions performed using this facility may be passed to law enforcement officials to support court cases. Do not proceed if you are not authorized."* | | | N/A |  |
|  | Windows/Active Directory based authentication must be used wherever possible. (Users must be prompted for login).Alternatively, in its absence, Form based authentication must be used. | | | N/A |  |
|  | In non Active Directory authentication, application must allow users to change their own passwords. | | | Yes |  |
|  | Applications must enforce a basic minimum password policy built-in as under: | | |  |  |
|  |  | | Enforces complex passwords (alphanumeric plus special characters OR alphanumeric with at least one Upper Case char – e.g. Qw3rgh\*@9 OR Q1dfc58g) | Yes |  |
|  |  | | Enforces a minimum password length of 8 characters | Yes |  |
|  |  | | Enforces a password history (no of passwords that are remembered and cannot be repeated) of at least 15 passwords or lesser | No |  |
|  |  | | Enforces a password expiry period of 45 days or lesser | No |  |
|  |  | | Enforces password change on first successful logon using any password that is issued by a security administrator (includes enforcement of change on initial and subsequent issuance). | Yes |  |
|  |  | | Allows logon with an expired password but enforce password change before completing the logon process; | No |  |
|  |  | | Disallows passwords from being the same as the User ID. | No |  |
|  |  | | Default Passwords must be changed. | Yes |  |
|  |  | | Enforces generation of random passwords (the password must be displayed on the screen for the security administrator to relay to the user) | Yes |  |
|  |  | | Enforces passwords masking | Yes |  |
|  |  | | Enforces user profile lock out (to be set to ‘for ever’) after 5 (or lesser) incorrect passwords attempts. If the user re-invokes the application, the counter must not be reset (zeroized) and the user must not be able to re-enter credentials again | Yes |  |
|  |  | | The current password must always be asked to users for password change functionalities. | No |  |
|  |  | | Password reset or unlock functionalities must be supported with secret questions and similar arguments. | No |  |
|  |  | | Password reset or unlock functionalities must not send user names and passwords back within emails. Instead, a secure link with certain lifetime must be sent that prompts a dialog for password change | No |  |
|  | Any critical data (e.g. password, credit card, and sensitive data) must be exchanged between clients and servers only over secure HTTPS protocol. | | | No |  |
|  | HTTP/HTML attributes (e.g. Autocomplete, cache-control, pragma) must be enabled and configured accordingly to prevent storage of sensitive information like passwords within caches. Browser Auto complete feature must be disabled at application code level. | | | N/A |  |
|  | Browsers must not cache critical form fields like username, secret answer, credit card, customer id etc. | | | Yes |  |
|  | Browsers must not cache the contents of the authenticated pages. | | | Yes |  |
|  | Applications that are used internally for which Out of Band or Hardware token is not feasible ,Application must provide configurable wide range of secret questions that are presented to the customer to perform critical transactions like beneficiary addition, profile maintenance, password reset, etc. | | | No |  |
|  | The account must be locked out after 5 unsuccessful tries of secret question | | | No |  |
|  | Passwords and secret question-answer of password retrieval functions must never be stored in plain text. | | | No |  |
|  | Salt value must be used as well by the generation of password hashes. | | | No |  |
|  | System must force users to change their password on first logon | | | Yes |  |
|  | The application automatically disables/inactivates User IDs that have not logged in for 90 days | | | No |  |
|  | A common message must be used for authentication failures to prevent user enumeration attacks. An example of such a message would be "Username and/or Password is wrong". | | | Yes |  |
|  | All successful and unsuccessful authentication attempts and access attempts to resources must be logged. | | | No |  |
|  | Upon Successful logon, application displays | | |  |  |
|  |  | The last logon date for successful attempts | | No |  |
|  |  | The last logon time for successful attempts | | No |  |
|  |  | The last logon date for unsuccessful attempts | | No |  |
|  |  | The last logon time for unsuccessful attempts | | No |  |
|  |  | The number of unsuccessful attempts | | No |  |
|  |  | Logged in User Name (Not User ID) | | Yes |  |
|  | All activities within applications must be logged at application and server levels including IP address, workstation ID, date and time.  E.g. security logs, audit logs, application logs, application parameter changes etc. | | | Yes |  |
|  | The application must have a transaction log to keep track / identify changes done (before and after modification) such as additions, deletions, modifications, updates, date / time stamp etc to records. | | | Yes |  |
| Session Management | | | | | |
|  | Unique values (e.g. session identifiers, token etc.) used for session management must be generated via secure random number generators. | | | N/A |  |
|  | The application supports the following session ID/token & cookie based features, where applicable. | | |  |  |
|  |  | The session ID must be unpredictable (random enough) to prevent guessing attacks. A good PRNG (Pseudo Random Number Generator) must be used | | No |  |
|  |  | The session ID must be long enough to prevent brute force attacks. The session ID length must be at least 128 bits (16 bytes).Session ID’s should not be easily guessable like sessionid:12345678 | | No |  |
|  |  | The session ID content (or value) must be meaningless to prevent information disclosure attacks. The session ID must simply be an identifier on the client side, and its value must never include sensitive information. | | No |  |
|  |  | Protects account credentials and session tokens. Do not use account credentials in session tokens. Enable secure attribute of the cookie for secure transmission of session cookies. | | No |  |
|  |  | Cookies, if used, have an expiration date so that they are valid only for a pre determined period of time (where applicable). | | No |  |
|  |  | Session tokens are invalidated at the server side when the user logs out. | | No |  |
|  |  | Session tokens are non-persistent at client & server (e.g. Session tokens are not written to the browser’s history or cache) | | No |  |
|  |  | Use Built-in Session Management Implementations of the web development frameworks. | | Yes |  |
|  |  | It is recommended to change the default session ID name of the web development framework to a generic name e.g.  Default Name: ASP.NET\_SessionId, ASPXFORMSAUTH  Recommended Name: ID | | No |  |
|  | Server must over write the browser generated session id upon successful authentication and server must add Auth parameter (like ASPXAUTH in ASP.NET) to session id that determine if a user is authenticated. This may reduce attack surface for session fixation and hijacking attacks. | | | No |  |
|  | After each authentication and re-authentication, a new session id must be created and the old session id must be invalidated. | | | No |  |
|  | An inactivity timeout must be set to 5 minute for applications sessions. | | | Yes |  |
|  | The number of concurrent sessions a user can run at a given time is limited to 1 only. | | | No |  |
|  | Solutions like tokens (challenge – response tokens), CAPTCHA must be integrated for critical operations in order to prevent Cross-Site-Request-Forgery (CSRF) attacks. | | | No |  |
|  | The domain and path for cookies containing authenticated session identifiers must be set to an appropriately restricted value for the site. | | | No |  |
|  |  | *httponly* attribute must be set on cookies. | | No |  |
|  |  | secure attribute must be set on cookies for HTTPS communications | | No |  |
|  | Logout tab must be available within all pages of accessed applications. | | | Yes |  |
|  | External link must be avoided. If required, all external pages must open in new browsers window with appropriate disclaimer note. | | | No |  |
| Authorization | | | | | |
|  | Application has proper ACL’s in place to control unauthorized access through Directory Path Traversal. e.g.  dot-dot-slash attack (../), directory traversal, directory climbing, or backtracking | | | No |  |
|  | Application has proper authorization controls in place to check Forceful Browsing (Unauthorized access to the files with or without authentication), unauthorized access to restricted resources without authentication/after logout. | | | No |  |
|  | Parameter manipulations within GET/POST requests must be taken into consideration and unauthorized access to legal user resources by attackers must be prevented. | | | No |  |
|  | Restricted URLs, functions, object references, services, application data, user information and security configuration files must be accessible for authorized users and roles. | | | Yes |  |
|  | Access rights /permissions /options granted to a user must be based on the principle of least privilege. Access matrix should be defined by business. | | | Yes |  |
|  | Application has controls in place to check Vertical and Horizontal Privilege Escalation. E.g. User with same privilege must not be able to access each other’s account information or impersonate identity and user with low privilege must not be able to access functionalities of higher privileged user. | | | Yes |  |
| Business Logic | | | | | |
|  | Application has adequate controls to perform logical checks where ever applicable. E.g. Malicious User cannot Bypass the actual workflow required to complete a process. | | | Yes |  |
|  | Application must have transaction password/tokens/SMS for critical transactions (like posting). User must enter transaction password /tokens/ SMS to complete such transactions. | | | No |  |
|  | Applications must be configured to operate under dual control arrangement, to complete any critical task (i.e. at least two individuals to be involved – ‘maker’ & ‘checker’) | | | No |  |
|  | Applications must prevent Checkers/Verifiers/Approvers from editing or changing transactions i.e. an unapproved transaction must be re-sent back to the maker for re-input. | | | No |  |
|  | Applications must prevent Originators/Makers from editing or canceling approved transactions. Approved transactions that require changes must undergo an approval process. | | | No |  |
|  | Application Administrators must not have roles that perform business transactions on the application | | | Yes |  |
|  | Application Administrators must not have a Security Administration ID on the underlying Operating System | | | Yes |  |
|  | Application Administrators must not be able to maintain security parameters. (i.e. segregation between System Maintenance & Security Maintenance functions ) | | | Yes |  |
| Data Validation | | | | | |
|  | All user inputs must be validated on client as well as server-side. Client side validations are not adequate. White-lists must be used for validation instead of black-lists. | | | Yes |  |
|  | Each input must be encoded to a common character set before validation (canonicalization).e.g.  & --> &amp;  < --> &lt;  > --> &gt; | | | No |  |
|  | User inputs must be escaped and validated before using as a part of command execution. | | | Yes |  |
|  | Prepared statement, parameterized query, bind variables and white list data validation must be implemented to prevent SQL injection attacks. | | | Yes |  |
|  | Output Escaping/Encoding must be applied on all user inputs before they are displayed on their screens | | | Yes |  |
|  | User inputs are checked for type, length and content. *Validation processes include ‘type checks’ e.g. numeric or alphanumeric etc, ‘Length checks’ e.g. right number of characters etc, ‘Range checks’ e.g. days between 1-31; months 1-12 etc.* | | | Yes |  |
|  | User inputs regarding arithmetic operations must be checked and validated for minimum and maximum values. | | | Yes |  |
|  | Data validation checks must be performed consistently throughout the Application in each form field, application parameter ,search field, session parameters etc. | | | yes |  |
|  | Disable right click and copy-paste options on all the pages. | | | No |  |
|  | User inputs regarding file access operations must be normalized and validated. | | | Yes |  |
|  | Application must check for Name, Length, Type and Contents in file Upload feature. | | | N/A |  |
|  | If data uploads (including batch uploads) are required and allowed (includes input file received from 3rd parties), the application or destination system must accept them only after checking the file for malicious code (e.g. via a virus/rootkit scan) before being processed and fed into Emirates NBD applications and eventually, databases in order to ensure that no compromise can occur. | | | N/A |  |
|  | If data uploads (including batch uploads) are required and allowed, Upload directory should be on separate volume other than where the application and system files resides. System must ensure that the uploaded file does not exhaust all the disk space available that might affect the availability of the system. | | | N/A |  |
|  | User inputs used for HTTP redirects must be validated by using whitelist method to prevent phishing attacks (open redirect problem).Redirect function must allow redirect only to specific authorized domain. | | | No |  |
|  | If Application uses Redirect parameter, Server must have a relation of the authorized redirections (i.e. in a database). | | | No |  |
|  | Appropriate solutions against frame busting and clickjacking attacks must be implemented within web applications. Browser must not allow framing from other domains. | | | N/A |  |
|  | CAPTCHA or similar anti-automation security controls must be implemented within HTML forms to prevent DoS, brute-forcing and dictionary attacks. | | | N/A |  |
|  | A timeout for search functionalities must be enabled against SQL Wildcard attacks which force databases to perform CPU-intensive queries by using several search wildcards like "%" or "\*". | | | N/A |  |
| Security Administration & Logical Security | | | | | |
|  | Applications must be able to disable user profiles on a given date. | | | No |  |
|  | Application (in-house/vendor developed or vendor purchased applications) incorporates a security administration module that mandates a maker/checker arrangement for the maintenance of users, passwords and parameters. | | | No |  |
|  | Applications must have the capability to disable user IDs e.g. to cater for staff on leave. | | | Yes |  |
|  | Application must provide a custom designed security activity Reviewer Profile i.e. a separate user ID which only allows viewing the audit log & other user related reports must be available. This id must not have any User Creation / Modification privileges | | | N/A |  |
|  | Application provides for alphanumeric User profiles. These user profiles are able to accommodate at least 10 characters. | | | N/A |  |
|  | Applications user IDs must have fields to capture detailed text information such as Names, Staff Numbers etc and the ID must belong to an existing Emirates NBD staff (User Profile Administration) | | | N/A |  |
|  | Application must ensure that Security Administration tasks are performed using specific menu-driven profiles only rather than using the command line. | | | N/A |  |
|  | Audit Logs/Reports can capture all user activities and automated system tasks. | | | Yes |  |
|  | All data processing activities involving sensitive static data such as keying material, must be logged to an audit trail, transaction journal etc. | | | Yes |  |
|  | The audit trail itself must be reliable and must be protected from any alteration or deletion. | | | Yes |  |
|  | Audit Logs/Reports are exportable to popular formats such as PDF, XLS, CSV, TXT, etc | | | No |  |
|  | Export of the audit logs must be restricted to Authorized Users only. | | | Yes |  |
|  | Application Audit Logs must be retained on the server or database for at least 30 days | | | No | On DB and deleted upon request |
|  | Application must support integration with Log Collection & Security Event and Information Management System (SEIM) | | | N/A |  |
|  | There is an Audit Log to capture activities performed by each user / administrator profile: | | |  |  |
|  |  | List of Users with Group/Role assigned  Filter required for Department, User, Group/Role and All | | Yes |  |
|  |  | List of Active / Disabled (Inactive) users  Filter required for Department & All | | No |  |
|  |  | List of User not logged in for more than X Number of Days (X is an input field) | | No |  |
|  |  | Login History  Filter required for All, User wise & Date wise | | No |  |
|  |  | List Locked-out accounts | | Yes |  |
|  |  | List Failed Login Attempts | | No |  |
|  |  | List of available Groups/Roles  Filter required for Role & All | | yes |  |
|  |  | List of Access Rights/Options assigned to Groups/Roles/Users  Filter required for Role & All | | yes |  |
|  |  | List of Group/Roles showing attached users  Filter required for Group/Role & All | | No |  |
|  | There is an Audit Log to capture Security Administrators Activities listed below: | | |  |  |
|  |  | User Creation Details | | Yes |  |
|  |  | User Deletion Details | | Yes |  |
|  |  | User Modification Details | | Yes |  |
|  |  | Change Password Details (Resetting, Generation, Change) | | No |  |
|  |  | User Activation Details | | No |  |
|  |  | User De-Activation Details | | No |  |
|  |  | User Locked Details | | Yes |  |
|  |  | User Unlocked Details | | Yes |  |
|  |  | Group/Role Creation Details | | Yes |  |
|  |  | Group/Role Deletion Details | | Yes |  |
|  |  | Group/Role Modification Details | | Yes |  |
|  |  | Changes to Application Parameters | | Yes |  |
|  |  | Force Logouts carried out for a user | | No |  |
|  | The following user profile related reports are available: | | |  |  |
|  |  | List of Users with Group/Role assigned.  Filter required for Department, User, Group/Role and All | | No |  |
|  |  | List of Active / Disabled (Inactive) users.  Filter required for Department & All | | No |  |
|  |  | List of User not logged in for more than X Number of Days (X is an input field) | | No |  |
|  |  | Login History  Filter required for All, User wise & Date wise | | No |  |
|  |  | List Locked-out accounts | | No |  |
|  |  | List Failed Login Attempts | | No |  |
|  |  | List of available Groups/Roles  Filter required for Role & All | | No |  |
|  |  | List of Access Rights/Options assigned to Groups/Roles/Users  Filter required for Role & All | | No |  |
|  |  | List of Group/Roles showing attached users  Filter required for Group/Role & All | | No |  |
| Interfaces & Middleware | | | | | |
|  | Interface User IDs must not be supervisor / administrator-equivalent. It must only have necessary minimum access rights to carry out the desired functions. | | | Yes |  |
|  | Interface User IDs must not have any interactive logon capability. If the application requires the interface ID to log on, it must be localized to the relevant systems / applications (i.e. must access only specified machines) only. | | | Yes |  |
|  | Authentication & Integrity checks over interfaces must be included to ensure that:   * Any data processed is bona fide * Data is not illicitly created or inserted * Data is not altered, and * Data is not replayed or deleted   *Explanatory Note: Systems that communicate with other systems without the direct involvement of a user need to authenticate each other’s identity to prevent unauthorized access*. | | | N/A |  |
|  | Where applications are processing data across an interface, there must be a reconciliation report at both sides of the interface to ensure completeness and integrity of processing across the interface  *Note: A reconciliation mechanism like a transaction report, allows abnormal or unusual transactions to be investigated* | | | Yes |  |
|  | Where the application is hosted on external servers (like 3rd party servers) and is required to interface with systems within Emirates NBD Group it must be able to connect and make transactions only via an Application Programming Interface (API) | | | N/A |  |
|  | If an application is designed such that it has different modules that may or may not be interfacing with each other, the security system of all these modules must be uniform and consistent in strength(e.g. If an application has a client, trader and admin modules, all global security parameters must be exactly the same for all the modules) | | | N/A |  |
| BCP /DR | | | | | |
|  | The Escrow Agreement must state the terms & conditions of release of the source code under various circumstances | | | No |  |
|  | Emirates NBD must be able to modify a copy of the source code provided and use it for business continuity purposes. | | | No |  |
| Contractual Agreements | | | | | |
|  | Vendor will cooperate with the Client and the 3rd party contracted to undertake the “Security Assessment” by providing the necessary documentation, source code (where feasible & agreeable), demonstration of the application and any other means required by the "Client". | | |  |  |
|  | Vendor will fix every security issue identified and reported by the Client and / or the 3rd party agency contracted to do the testing, at its own cost. Security issues rated as Critical or High must be fixed before the solution goes live or is rolled out to end-users or within 7 business days of being reported and those rated as Medium must be fixed within 30 days of being reported | | |  |  |