Annexure-1: SSDLC Checklist

Bidder to provide response with below options, where appropriate in SSDLC checklist.

1. True
2. False
3. N/A
4. Other (explain)

Application Security Review Checklist (Bidder to provide response)			Availability Response	Bidder Response
INSTRUCTIONS: Fill in all applicable sections.		Guide		
1.0	Application Components	Acceptable Criteria	(Yes/No)	
1.1	Are Application Components identified ?	Identify all application components (either individual or groups of source files, libraries, and/or executables) that are present in the application		
1.2	Are Application Dependencies identified ?	Identify all components that are not part of the application but that the application relies on to operate.		
1.3	Is the Application Architecture Defined ?	Identify a high-level architecture of the application.		
1.4	Are Application Business/Security Functions Identified ?	Identify all application components and defined in terms of the business functions and/or security functions they provide.		
2.0	Identification and Authentication			
2.1	Authentication of End-users Is the authentication mechanism implemented for end users?	If the application contains only public information then user authentication may not be required. A user Authentication mechanism must be implemented if the application contains Confidential, Sensitive Information with PII, Sensitive or Private information. The strength of the authentication mechanism must be commensurate with the risk of the application. e.g. two factor authentication for Customer facing internet applications or digital Certificates.		
2.2	Authentication for Administrator: Is the authentication mechanism implemented for administrator?	An authentication mechanism for Administrator must be implemented for the application regardless of which class of data the application contains. The administrator authentication mechanism must be at least as strong as the user authentication mechanism.		
2.3	Unique ID for user Does the application use a unique login ID for each user?	The generation of login IDs of users must be uniquely identifiable to user. If the answer is "No" it is unacceptable.		

2.4	Error Message for Failed login Attempts Does the application use a generic message for login attempts failures and account lockout?	All authentication controls fail securely. The error message for any failed login attempts and account lockout must be generic to prevent ID/Password guessing attacks. E.g. Invalid ID, Incorrect Password messages are not allowed. Log all authentication decisions. This should include requests with missing required information, needed for security investigations.		
2.5	User ID generation for Customer Are the identities generated based on the non-public information?	User identities should be generated without containing personal data e.g. personal data like ATM/Credit Card number, CNIC number. Email IDs can contain user names.		
2.6	Initial Password Does the application prompt to change the initial password?	Initial password should be pre-expired. Application should immediately prompt to change the initial password after the users first log into the application.		
2.7	Clear Text Password Password is never displayed on the screen in clear text (with the exception of one time use password resets).	All password fields do not echo the user's password when it is entered, and that password fields (or the forms that contain them) have autocomplete		
2.8	Static Password Strength Policy If static password are being used for authentication, is the strength policy being enforced to ensure password meets IT Security policy criteria, Password Expiration Notification, Password history, Maximum failed login attempts?	Password parameters shall be configured in accordance to NBP IT Security Policy as mentioned below: Password history should be maintained for at least 10 passwords. Password Should be hard to guess. It should not constitute with the common predict phrases like names, Tel Number, Date of Birth, Anniversary, same as user name etc. Account should be locked after 3-5 consecutive unsuccessful login attempts, release of locked account automatically after 30 minutes is recommended. User Level • Minimum 8 characters • Users should be forced to change on or before the expiry period of 45 days. • Account should be locked after 3-6 unsuccessful login attempts, and should only be unlocked upon receipt of request from valid user to the administrator. Privilege Level • Minimum 11 characters • Privilege / admin users should change their password on or before password expiry policy setting of 90 days.		
2.9	Static Password Rules If static password are being used, are the following password rules enforced? Password should be different from user name Password should not be easily guessable Password should not be blank	The strength of any authentication credentials are sufficient to withstand attacks that are typical of the threats in the deployed environment. Password should be different from user name Password should not be easily guessable passwords e.g. 12345678, asdfasdf, etc. Password should not be blank		

2.10	Session Inactivity Timeout Does the application enforce a session inactivity timeout?	Inactivity timeout for the users should be implemented to prevent unauthorized access of an active login session when the user is not present. Inactivity timeout period should be based on the application IS risk level which may be 5 to 15 minutes.		
2.11	Secure Authentication Protocol Is the application using the authentication based on the international standards	A secure mutual authentication protocol with a proper key management scheme to encrypt credentials (e.g. password) should be used. Examples are Kerberos, TLS. One time password or dynamic password can be sent in the clear over the network.		
2.12	Security Contexts Does the authentication server create unique security contexts for the authenticated users	secure session IDs / secure cookies / Kerberos tickets		
2.13	Dynamic Password System If a Dynamic password (one-time) system is used for authentication, it is approved by the Information Security and relevant stake holders.	All Dynamic password system must be reviewed by the Information Security before implementation.		
2.14	Digital Certificates and Certificate Authority (CA) If digital certificates are used, are they issued by approved CA?	Digital Certificates used by the application should be issued by approved CA authority/certificates provider e.g. VeriSign CA. Self-signed certificates can be used for testing purposes. PGP and point-to-point secure file transfer can be used where endpoint authentication is not required.		
2.15	Biometric Authentication if a Biometric Authentication mechanism is used by the application, is it approved by the IS?	Biometric authentication mechanism tend to be one-off solutions and are driven by business requirements (like ATM Authentication) therefore, they should be reviewed and approved by the IS to ensure they are secure.		
2.16	Two Factor Authentication if a two factor Authentication mechanism is used by the application, is it approved by the IS?	Two factor authentication or MFA should be reviewed by the IS/SME before the implementation.		
2.17	Single Sign-On (SSO) If the internet application is using shared authentication services, is it reviewed and approved by IS?	SSO or any shared authentication services should be reviewed by the IS/SME before the implementation.		
2.18	Logout Does the application allow users to completely log out from the application?	The application must provide the logout capability such that the user can completely log out of the application.		
2.19	Brute Force Attacks Is the resource governor controls in place to protect the application against vertical & horizontal brute forcing attacks?	The resource governor is in place to protect against vertical (a single account tested against all possible passwords) and horizontal brute forcing (all accounts tested with the same password e.g. “Password1”). A correct credential entry should incur no delay. Both these governor mechanisms should be active simultaneously to protect against diagonal and distributed Attacks.		
3.0	Authorization / Access Control /			

	Entitlement			
3.1	Authorization / Access Control / Entitlement If the application contains private or higher data, does the application provide mechanisms to control access based on the identity of the authenticated user.	Access control should be implemented and auditable. Users are given only those privileges necessary to perform their function. e.g. via entitlement profile / group / role base which are based on the Least Privilege. Access controls fail securely.		
3.2	Inactive / Obsolete Entitlement review Does the application support a mechanism to review inactive / obsolete entitlements	To ease entitlement review, it is beneficial if inactive/ obsolete entitlement can shown by the application		
3.3	Entitlement Review report Does the application provides the complete details of all users entitlement in form of a report? Security Administrator should have the ability to generate these reports per department / unit for periodic user entitlement review.	To ease entitlement review, application should generate the complete entitlement report of users for periodic entitlement review. High risk application should be able to provide the fine-grained entitlements. Verify that all access control decisions are being logged and all failed decisions are logged.		
3.4	Functional ID Management Does the application have a defined owner who is responsible for all aspects of the Functional IDs including usage, entitlement review and password management			
3.5	Access Control for Privileged Actions Does the application enforce access control for the following privilege actions? • Create, modify and delete user accounts and groups • Configure passwords or account lockout policy • Change passwords or certificates of user • Establish log sizes, fill threshold and behavior.	Access control on privileged access should be enforced to maintain system integrity. If least privilege cannot be enforced, compensating controls should be implemented to mitigate the risk (e.g. activity log review)		
3.6	Account management functions are account management functions secure.	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.		
3.7	Re-Authentication is re-authentication required before any sensitive operations	Re-authentication is required before any application-specific sensitive operations are permitted. E.g. authenticating the customer again when conducting Financial Transaction or creating a beneficiary on an internet facing application		

3.8	Authenticity and Integrity of Authorization Data Is authorization data being stored on the client side (e.g. cookies, tickets) after the users get authenticated? If yes, is any mechanism being implemented to protect authorization data, prevent spoofing and maintain its integrity?	If authorization data is being stored on the client, it is important to ensure authorization data is protected/encrypted against unauthorized modification by the user. Ideally authorization data should be stored on the server to maintain data integrity.		
3.9	File and Directory Protection Is file and directory authorization enabled for user access control?	In addition to user entitlement, file and directory access control lists should be configured properly to protect the application's files against unauthorized access.		
3.10	Remote access System For internal application if non-NBP staff access this application remotely, is it over an approved solution which is reviewed and approved by IS?	All remote access to NBP systems / networks used by non-NBP staff (e.g. vendor) must be reviewed and approved by the IS.		
4.0	Data Confidentiality and Data Integrity			
4.1	Input Validation Does the application validate user inputs? Is input validation performed on the server or Client side?	Web Application that take data input can be exploited by the following attacks: buffer overflow, cross site scripting (XSS), SQL injection, code injection, denial of service and elevation of privileges. Input is validated to check for valid types, formats, lengths, and ranges and to reject invalid input. This validation is more critical if input filenames, URLs or user names are used for security decisions. Input validation must be performed on the server side instead on the client side to prevent it from being bypassed. Input validation should be enforced for the following: • Input from users • Parameter from URLs • Values from Cookies • Hidden fields to prevent SQL injection • Filter out character like single quotes, doNBPe quotes, slashes, back-slashes, semi colons, extended characters like NULL, carry return, new lines, etc. in all strings • Convert a numeric value to an integer or check whether it is an integer before parsing it into an SQL statement.		
4.2	Data Protection in Transit Is the sensitive or above category data protected during transmission in certain specific environments.	Identify the list of sensitive data processed by this application and there is an explicit policy for how access to this data must be controlled, and when this data must be encrypted (both at rest and in transit). The transmission of data can take many forms including, but not limited to electronic file transfer (e.g. FTP), web traffic, e-mail, tapes, CDs, DVDs, Disk and so on. Transmission of sensitive PII should be encrypted. All cached or temporary copies of sensitive data are protected from unauthorized access or		

		purged/invalidated after the authorized user accesses.		
4.3	Input length Does the application limit the length of each user input field?	The length of every field should be limited		
4.4	Input Manipulation Does the application prevent Sensitive and above data from being passed in clear ?	Sensitive and above data especially authentication data must not be passed in clear text to prevent it from being manipulated by users.		
4.5	Content Cache Does the application prevent Sensitive and above data from being cached on user's local disk.	Sensitive and above data should not be cached on user's local disk. It could be accomplished in web application by implementation HTTP response header with: 'Prama:No-cach' for ASP or 'Cache-control: No cache' for JSP/Servlet.		
4.6	File Upload If the application supports file upload functionality, does it enforce the following a. Validate file extension/type, and file format. b. Run virus/malware scan on the uploaded file.	If data files are being uploaded to NBP servers from the external entity, the receiving server must have anti virus software to scan files before processing. The controls should be inplace to check the integrity of files such as Hashes. Only process the allowed/agreed file extensions. It is a sound practice to validate file extension / type, file format and run scan on the uploaded files, especially executables or other file type that can carry and propagate viruses.		
4.7	Data Protection in Storage Is data at the sensitive or above category protected in storage?	Sensitive and above category data must be protected/encrypted in storage and only accessible by respective users.		
4.8	Password Protection in Storage Is password data protected in storage	Account passwords are salted using a salt that is unique to each account and hashed before storing. All authentication credentials for accessing services external to the application are encrypted and stored in a protected location (not in source code).		
4.9	PII data Mask If the application has a feature to deliver or display an e-statement for customer account activities or to export production PII data, are customer account number/CNIC/credit card number partially masked?	Any combination of PII (personally Identifiable Information) that identifies an individual human being in a manner that would facilitate identity theft, credit fraud or other financial fraud must be protected against unauthorized access. Customer name or contact information in combination with CNIC number or tax number, passport number or account number is an example of Sensitive PII. Also note that when production data is exported for testing, PII data should be masked.		
5.0	Cryptography & Key Management			

5.1	Cryptographic Keys List all type of cryptographic keys such as symmetric, asymmetric, secure hash keys used in the solution. Describe their use scenarios and the security mechanisms associated with each cryptographic algorithm used	if encryption is used for authentication, data protection, key management, digital signature or other purposes, all cryptographic keys including their type, cryptographic algorithms and key length and secure hash algorithm must be listed as a pre-requisite for key management assessment. For instance, an application may implement 2048-bit RSA digital certificates for user authentication and key management, 128-bit AES for data protection in transit, and SHA-2 for password protection.		
5.2	Cryptographic Algorithms and Key lengths Do all cryptographic keys comply with current market standards/requirements? (AES/3DES/RC4, SHA-2/256, RSA etc.) and key length?	Security of the data encryption shall depend on secrecy of the key, not secrecy of the algorithm. All the cryptographic algorithms and key length being used must be validated against FIPS 140-2 or an equivalent current market standards/requirements, Symmetric keys: 168-bit 3DES, 128 AES Asymmetric keys: 2048-bit RSA or 256-bit ECDSA or ECDH Source hash: SHA2 (i.e. SHA-256/384/512)		
5.3	Key Generation & Management Are all cryptographic keys randomly generated using approved Random Number Generator? What type of random number generator is used by the application?	All cryptographic keys must me randomly generated by approved random number generator (e.g. as per NIST FIPS 140-2), also define how cryptographic keys are managed (e.g., generated, distributed, revoked, expired)		
5.4	Key Data Display if a manual key entry process is used, do utilities used to load or enter keys or key components prevent the display of the data loaded or entered in the clear?	Distribute the key between multiple custodians and prevent to display the keys in clear during entry		
5.5	Key Renewal Frequency Do all cryptographic keys have a defined renewal frequency period to comply with	Cryptographic keys should be changed on a periodic basis commensurate with the frequency of key use or exposure to eavesdropping or unauthorized access. E.g. • Key encryption: At least once per month (automated) • Master keys or symmetric keys for authentication or key management : at least once per year (if manual renewal) • Cryptographic keys that cannot be renewed should have a pre-defined expiration period (e.g. 3 years of PIN generation keys)		
5.6	Key Protection in Storage Are all symmetric and asymmetric (private) keys, except public keys, encrypted and stored in a hardware or software cryptographic module with proper access control?	Cryptographic keys except public keys, must be encrypted in storage with proper access control. Access control must be prevent unauthorized access.		

5.7	Hard-coding Cryptographic Keys Does the application prevent any encryption keys or passkey being included in the source code or configuration files?	• Hard coded keys that are manually set into code or part of the application and cannot be changed are not acceptable as the keys are known to developers and all instances of the application should use the same set of keys. • Cryptographic keys being coded as the default should be configured and changeable during installation or configuration. • if Cryptographic keys are hardcoded they cannot be changed. Therefore, not acceptable.		
5.8	Certification Validation When digital certificates are used for the digital signature or authentication, is an up-to-date certification revocation list (CRL) used to verify the validity of the CA's and user's / server's certificates if an on-line certificate validation mechanism via Online Certificate Status Protocol (OCSP) or Server-based Certificate Validation Protocol (SCVP) is not available?	• When an online certificate validation protocol such as OCSP or SCVP is not supported. CRLs must be made available for servers to client's certificates or for the clients to server's certificate if certificates are used for digital signature or authentication.		
5.9	CRL renewal When a CRL is used for certification validation, are cached copies of CRLs updated regularly? If so, please describe the frequency (e.g. once per day). Is the frequency of update commensurate with the associated risk of the application.	• CRL (Certificate Revocation Lists) must be updated periodically.		
5.10	Key Recovery Compliance if the application is subject to supervisory or data retention requirements such as SBP - psd/2014/C3-Annex or section 7 of PS&EFT Act 2007 or any other key recovery requirements, is there a key recovery process to fulfill the regulatory requirements?	• To comply with NBP policy or other regulatory requirements, key recovery must be enforced, such as encrypted transactional messages or data can be decrypted and recorded for regulatory compliance, log all Cryptographic module failures		
5.11	Unique Key Does each cryptographic key have a unique application domain? For each party, there should be as many different keys as there are different cryptographic functionalities.	• Any particular key should be used for one particular purposes (e.g. signing, data encryption, Key encryption etc.) • Keys used for in production environment must not be used for development or testing.		
6.0	Error Handling & Audit Logging			

6.1	Auditing and Events Does the application have an auditing capability across application layers? Depending on the risk of the application, are audit logs and alerts of unauthorized access maintained?	The answer should be "Yes" since to comply with the NBP IT policy section 2.3.5.5 Logging is performed before executing the transaction. If logging was unsuccessful (e.g. disk full, insufficient permissions) the application fails safe, this is for when integrity and non-repudiation are a must. Following significant events should be available for review: • ID Management (Create, Delete, Modify, Suspend, Resume) • Successful / Unsuccessful user login attempt • Account Lock out • Account Lock/Unlock • Group Creation • Creation or modification of application roles/ profiles • Creation/modification/deletion of user rights • Password Forget • Password Resets • Password Change • Change in Application/System security configuration • Alarms associated with a firewall or IDS/IPS • Financial Transaction or Sensitive PII data • Security Validation failure • Session Management failure • Application/Service Start/Stop • Suspicious/Fraud and other criminal activities		
6.2	Audit Events contents Does the individual audit event contain the necessary attributes?	Each log entry needs to include sufficient information for the intended subsequent monitoring and analysis. The application logs must record "when, where, who and what" for each event. All auditable events must give enough information to trace the event to a particulars but not limited to the following: • Unique log identifier • The user ID or the process ID of the event • System, application, module or component • Data and Time of the event • Application address e.g. IP address or machine name and port number • Resource ID e.g. window, url, page, form, method • Service Protocol • Source Address e.g. user/service IP address • Device Identifier e.g. IMEI, MAC • User, object Identity • Type of the event • Log Level • Success or failure of an event • Starting and ending time of access to the application • Description		

6.3	Admin activities Does security Administration activities for this system are logged and traceable to User ID?	To achieve the non-repudiation all the generic IDs should be escrowed and only accessible in case of emergency, each user must have its own ID and guarantees that an action can be proven to have originated from specific person.		
6.4	Audit Log Protection Are audit logs in store and during transmission, protected from unauthorized deletion, modification and disclosure? i.e. Administrator should not have the ability to modify / edit the logs	Audit Logs must be protected against unauthorized access to ensure its integrity and maintain accountability. The application does not output error messages or stack traces containing sensitive data that could assist an attacker, including Session ID and personal information.		
6.5	Audit Log File Configuration Can the audit log files be configured for log size and rollover to prevent log file data loss and a denial of service attack?	Audit Logs should be automatically roll over unless it can be ensured that the audit logs have been backed up or archived. Audit log file size should be configurable to prevent a denial of service attack or application handles log size issue automatically. Rollover must always be configurable.		
6.6	SIEM Integration Is application capable to integrate with SIEM	Application should be able to integrate with SIEM solution. SIEM is available which allows the analyst to search for log events based on combinations of search criteria across all fields in the log record format supported by this system.		
6.7	Audit Log Notification Are administrators warned when the audit logs are nearly full?	It is desirable to have a mechanism to warn administrator when the audit logs are nearly full to prevent an application from shutting down, from a denial of service or from overriding previous logs.		
6.8	Reports Does the application have an ability to generate custom audit reports based on the criteria specified by the log reviewer?	It is desirable to have a capability to generate audit reports based on a number of criteria specified by the log reviewer.		
7.0	Security Administration			
7.1	Functional ID If the application is using functional IDs (e.g. root in Linux/Unix, Administrator in Windows), are they protected against unauthorized usage?	It is important to list any functional Ids that exist and if any internal application or third party controls can be placed on the system to decrease the privileges associated with these accounts. Also controls should be in place for any system functional IDs to prevent unauthorized usage. In general, the existence of all power administration IDs is not desirable.		
7.2	Service Accounts Are service/database ids only used by the applications and not used by individual users or other processes	The application backend IDs such as database, service accounts are restricted and only allowed by the application. These accounts would not be accessible by any individual users.		
7.3	Separation of Roles Does the application split administration privileges into several accounts (e.g. system administration, Security Administration)?	According to the principle of least privilege it is desirable for administrative accounts to have the least privilege needed to perform a particular function. It is describable to have separate administrative roles to perform system management, security management and audit. If separation of roles cannot be enforced, then the		

		application must have provisions (e.g. auditing to ensure the accountability of the privileged accounts.		
7.4	Administrator's Conflict of Interest Does the application prevent a security administrator from performing transactions or administrative functions for themselves that conflict with this role?	The application should not allow security administrator to create or modify user accounts for themselves. In case there is a business requirement to allow such action, then an independent verification or maker/checker process must be implemented and all such actions must be audited.		
7.5	Maker/Checker for Administrative Actions Does the application support maker/checker or dual-control for administrative actions (e.g. account creation / modification, entitlement management).	It is describable to have internal maker/checker or dual-control for administration actions such as account creation/modification and entitlement management. When maker/checker or dual-control cannot be implemented, an independent verification process must be enforced.		
8.0	System Security and Availability			
8.1	Application Identity Is the application or web server running as a non-privileged user (e.g. non-root or non-administrator)?	If possible, the application or web server should run as non-privileged user, especially when this is a customer facing application. This provision should be implemented to reduce/control damage in case the application is compromised.		
8.2	Application Integrity Is there a mechanism to protect application configuration stores and maintain the integrity of critical application files?	It is important to protect application configuration stores and critical files with access control. This include all share folder for data and system files.		
8.3	BCP/DR Does the application support redundancy or replication for continuity of business or automatic fail-over?	Automatic fail-over is commonly required for mission-critical applications for high availability. However, some low criticality application may not have a BCP/DR requirement or manual process (last updated data restoration on contingency server). It is a business decision to determine whether it is required or not.		
8.4	DDOS attack Are controls in place to prevent a denial of service (DOS) attack on this system?			
9.0	Network Architecture and Perimeter Security			
9.1	Perimeter Security For Applications deployed in the DMZ, does the application prevent unauthenticated users from the Internet from accessing a server on our intranet?	For Internet applications, unauthenticated users must not be allowed to directly access the server or Intranet to prevent hackers from exploiting vulnerabilities on the server that would first compromise the server and then internal infrastructure. Users must be authenticated on a server in the DMZ (e.g. a web or VPN server) before interacting with another server on intranet. For B2B application, B2B server/devices must be placed in the DMZ to perform authentication.		

9.2	3-Tier Architecture Does the application support at least 3-Tier Architecture (e.g. web server, application server and backend server/database) to protect data from being directly accessed from the web server in the DMZ.	For Internal web applications, especially financial application or application that provide personal data, it is desirable to have at least a 3-tier architecture (3 tiers may include the tiers of web server, application server and backend server or database server)to protect the backend server/database from being directly accessed from the web server and being compromised.		
9.3	Persistent Storage on the DMZ if this is an Internet-based application, does the application prevent Confidential and above data from being persistently stored on a system in the DMZ?	Confidential and above category data should not be persistently stored on a system in the DMZ. i.e. Persistently stored means storage beyond the session lifetime.		
9.4	System-to-System Authentication Does the application support system-to-system authentication or the authentication at the application layer for communication between any two servers to prevent unauthorized access?	System-to-system authentication or authentication at the application layer should be implemented for communication between any two servers to prevent unauthorized access. Network access control such as SSL or IPsec could be used as a compensating control if system-to-system authentication or authentication at the application layer is not implemented. Note that IPsec is consider as the last resort.		
10.0	Session Management			
10.1	Session Management Does the authentication server(s) implements a session management mechanism to manage active login sessions and to prevent spoofing/masquerading?	Session management for this case is used to record the states of active login sessions and to retire inactive sessions when they time out. Session management should have the following properties: • Unique session identification • Session Identification that are protected in transit and in storage against unauthorized access. • An inactivity time out mechanism		
10.2	Application Logout Does the application have a logout functionality on every screen that is available for authenticated user?	When a user logs out, the application must completely log the user out and prevent the user from accessing pages or information that is available to active authenticated users.		
10.3	Session Identifier Generation Does the application generate session identifiers (IDs) with a sound pseudo-random number generator?	Session management is required for web applications because they are based on the stateless HTTP protocol. Therefore, session management is critical to the overall security of web applications. A sound session management scheme should be able to generate unique and unpredictable session IDs, restrict session lifetime, and protect session IDs.		
10.4	Session State Store Does the application protect its session state store against unauthorized access?	The session state store can be local or remote. Session state data should be protected against eavesdropping and unauthorized access. If session state store is remote then the data in transit should be encrypted with a secure protocol such as SSL or IPsec and the data in		

		store should be protected against unauthorized access.		
10.5	Session Lifetime Does the application restrict session lifetime and enforce the maximum login period for a session?	Prolonged session lifetime would increase the risk of session hijacking and reply attacks. Therefore the application should restrict session lifetime to reduce the risk. IS Policy requirements of inactive user session • 5 minutes for Critical System that are classified as sensitive; • 10-15 minutes for other classified systems based on business need		
10.6	Session Identification Passage Does the application prevent session identifiers from being passed over unencrypted channels?	If session IDs are used to track session states, then session IDs or cookies containing session IDs should be passed via encrypted channels (e.g. SSL/TLS) to prevent eavesdropping.		
10.7	Session Identifier Manipulation Does the application prevent users from manipulating session identifiers that are being passed in query string or from fields?	Session IDs should not be passed via query string or from fields because they can be easily modified by the users in an attempt to impersonate other users.		
10.8	Session Cookies Does the application encrypt session cookies?	Session (authentication) cookies should be encrypted to prevent session cookies from being stolen. Session cookie encryption along with SSL/TLS can mitigate the risk of cross-site scripting (XSS) attacks. Session cookies values that are created in insecure session should not be inherited in secure session cookies.		
10.9	Session Cookies Validation If session cookies are used by application, does the application validate the cookies before granting access to protected pages?	Cookies that contain Restricted or authentication data must be marked secure, so that they are sent only over encrypted channel, namely SSL/TLS. Cookies that contain Sensitive PII must be marked secured when transmitting via non-NBP manage infrastructures.		
10.10	Secure Cookies If the application uses cookies containing Sensitive or Higher information, are the cookies marked secured so that the cookies are sent only over encrypted channels AND is the cookies content encrypted using approved method?	Cookies that contain Sensitive+ data must be marked secure, so that they are sent only over encrypted channels, namely SSL/TLS. Cookies that contain Sensitive PII must be marked secured when transmitting via non-NBP managed infrastructures.		
11.0	Database Access			

11.1	Database Authentication Does the application server utilize the database authentication to directly connect to the database instead of user account authentication at the application level?	The Application Server may use a database account or an application account to establish the database connectivity. When the user accounts in the Application are tightly coupled with the database accounts, the database is accessible by all legitimate users on the platform. To enforce the principle of least privilege, it is more secure for application to use separate database accounts for DB authentication. In case where a user account on the Application server used to access the database, a least privileged account should be created.		
11.2	Database Password Protection in Storage Does the application protect/encrypt database connection strings (e.g. passwords) in local storage?	Database connection string contain authentication data and therefore must be encrypted in storage. Encrypted connection string and encryption keys must be protected. The function of decrypting connection string should be a standalone utility to prevent the connection string from being decrypted and display in the clear. Instead, it should be embedded into or fully integrated within the application.		
11.3	Database Password Protection in Transit Does the application enable/implement a secure protocol (e.g. SSL/TLS) to protect database passwords in transit?	If database connection string contains passwords it must be encrypted in the transit. In general, most database system support a secure protocol (e.g. SSL / TLS) for this purpose. When a secure protocol cannot be enabled or applied. IPsec or other secure protocol can be considered as a last resort for host-to-host encryption.		
12.0	Legal / Regulatory Compliance, Management Approval & Awareness			
12.1	Additional Legal or Regulatory Requirements Does the application comply with all regulatory / local laws which are not included in the Information Security policy.	Each and every application must comply with Legal/Regulatory/IS requirements.		
12.2	Banner Text Approval Is the Legal Department approved banner text, when supported by the application, displayed at all entry points where a user initially signs on?	If there is a need to support banner tax, legal-approved banner text must be displayed at all entry points where a user initially signs on either from local or remote access.		
13.0	Application Configuration and IS Processes			

13.1	Information Classification Has the information been classified in accordance with NBP Information Standards?	NBP TBML system(s) assets must be given a classification level in accordance with the NBP approved classification standard. • Confidential Information that is considered to be very sensitive to business and is intended for internal use only by following the need to know principle. Unauthorized disclosure of this level of information could seriously and negatively impact bank's reputation and may cause significant business loss. • Sensitive Information that requires a higher level of protection than normal from unauthorized disclosure or alteration. Unauthorized disclosure / alteration of such information may negatively impact bank's reputation or can cause legal implications. • Private Proprietary information that is being developed for NBP internal use and being shared among the NBP employees only. Property of NBP and disclosure of such information could affect the NBP business or employees. • Public Information either collected from public sources or being produced for public review. Disclosure of such information will not have an impact to NBP business & employees.		
13.2	Inherent Risk of the Application Has an inherent risk analysis been completed by the business during the definition phase of the project?	IS Risk Assessment is a mandatory requirement, Risk Assessment lifecycle must be completed in coordination of IS Risk team.		
13.3	Vendor Support Product Is the application software supported by an approved vendor?	Application vendor must be in the NBP's approved vendor list.		
13.4	Default Access Capability Are all default access capabilities (including passwords) removed, disabled or protected to prevent their unauthorized use?	No default ID should be used or enabled. Default IDs should be renamed and password split and escrowed with IS		
13.5	Vulnerability Assessment If required, has the application undergone all of the Application vulnerability Assessment and remediated all security findings as specified in the VA process.	If VA is required for this application, the required VA (Internal or External) must be performed and all security findings with the medium and High risk level must be remediated with the timeframe specified in the VA process.		
13.6	Masking Data Is sensitive PII information masked within the application whenever possible (displaying as well as printing)?	By the GLBA act, financial institutions must protect the security and confidentiality of customer's nonpublic personal Information (NPI). When NPI is being displayed or printed, it should be partially masked and only the last four digits can be displayed or printed for identification or verification.		

13.7	Configuration location are application configuration files protected from unauthorized access?	All security-relevant configuration information is stored in locations that are protected from unauthorized access.		
13.8	Configuration Error is application capable of handling configuration errors?	If the application cannot access its security configuration, all access to the application should be denied and do not allow access using default configuration.		
13.9	Audit Configuration Changes is auditing enabled to track application configuration changes?	All changes to the security configuration settings managed by the application are logged in the security event log.		
13.10	Fax Are automated or manual fax processes used in connection with the transection of data to/from the system? If yes describe the controls around the fax process.	If sensitive or above information must be sent over Fax, specific procedures and guidance must be created and followed to mitigate the risk. Fax cannot support user authentication nor data confidentiality. Manual authentication of the source and verification of the data may to be conducted to mitigate the risk and such action may need to be logged/recorded for accountability.		
14.0	Compliance			
14.1	3rd Party Solution is it certified by PCI, Common Criteria?	The solution related to Card processing must be certified by PCI, Common Criteria min Level 3+.		
14.2	Have any non-compliance being found as a result of this review? If True, provide corrective action plan and/or RA numbers in the "Open Issues and Approvals" TAB of this document			

Web Secure Coding Checklist (Bidder to provide response) *Bidder should complete this checklist in order to ensure compliance. This reflects best industry practice and correlates directly to issues that are identified during Vulnerability Assessments.*		*Availability Response*	*Bidder Response*
1.0 Application Verification			
1.1	The integrity of interpreted code, libraries, executables, and configuration files is verified using checksums or hashes.		
2.0 Authentication			
2.1	All pages and resources require authentication except those specifically intended to be public.		
2.2	All password fields do not display the user's password when it is entered, and that password fields (or the forms that contain them) have autocomplete disabled.		
2.3	If a maximum number of authentication attempts is exceeded, the account is locked for a period of time long enough to deter brute force attacks.		
2.4	All connections to external systems that involve sensitive information or functions are authenticated.		

2.5	The forgotten password function and other recovery paths do not reveal the current password and that the new password is not sent in clear text to the user.		
2.6	The username enumeration is not possible via login, password reset, or forgot account functionality.		
2.7	All authentication controls are enforced on the server side.		
3.0 Session Management			
3.1	The framework's default session management control implementation is used by the application.		
3.2	Sessions are invalidated when the user logs out.		
3.3	Sessions timeout after a specified period of inactivity.		
3.4	Sessions timeout after an administratively-configurable maximum time period regardless of activity (an absolute timeout).		
3.5	All pages that require authentication to access them have logout links.		
3.6	The session id is never disclosed other than in cookie headers; particularly in URLs, error messages, or logs. This includes verifying that the application does not support URL rewriting of session cookies.		
3.7	The session id is changed on each login.		
3.8	The session id is changed on re-authentication.		
3.9	The session id is changed or cleared on logout.		
3.10	Only session ids generated by the application framework are recognized as valid by the application.		
3.11	Authenticated session tokens are sufficiently long and random to withstand attacks that are typical of the threats in the deployed environment.		
3.12	Cookies which contain authenticated session tokens/ids have their domain and path set to an appropriately restrictive value for that site. The domain cookie attribute restriction should not be set unless for a business requirement, such as single sign on.		
3.13	Verify that authenticated session tokens using cookies sent via HTTP, are protected by the use of "HttpOnly".		
3.14	Verify that authenticated session tokens using cookies are protected with the "secure" attribute and a strict transport security headers are present.		
3.15	Verify that the application does not permit duplicate concurrent user sessions, originating from different machines.		
4.0 Access Control			
4.1	Users can only access URLs for which they possess specific authorization.		
4.2	Direct object references are protected, such that only authorized objects are accessible to each user.		
4.3	All connections to external systems that involve sensitive information or functions use an account that has been set up to have the minimum privileges necessary for the application to function properly.		
4.4	Directory browsing is disabled unless deliberately desired.		
4.5	The same access control rules implied by the presentation layer which are enforced on the server side, such that controls and parameters cannot be re-enabled or re-added from higher privilege users.		

4.6	All user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized.		
4.7	Verify the system can protect against aggregate or continuous access of secured functions, resources, or data. For example, possibly by the use of a resource governor to limit the number of edits per hour or to prevent the entire database from being scraped by an individual user.		
4.8	There is a centralized mechanism (including libraries that call external authorization services) for protecting access to each type of protected resource.		
4.9	Verify that the application or framework generates strong random anti-CSRF tokens unique to the user as part of all high value transactions or accessing sensitive data, and that the application verifies the presence of this token with the proper value for the current user when processing these requests.		
4.10	Limitations on input and access imposed by the business on the application (such as daily transaction limits or sequencing of tasks) cannot be bypassed.		
4.11	All access controls are enforced on the server side.		
5.0 Input Validation			
5.1	The runtime environment is not susceptible to buffer overflows, or that security controls prevent buffer overflows.		
5.2	All input validation failures result in input rejection.		
5.3	All input validation or encoding routines are performed and enforced on the server side.		
5.4	Single/Centralized input validation control is used by the application for each type of data that is accepted.		
5.5	All input validation failures are logged.		
5.6	All input data is canonicalized for all downstream decoders or interpreters prior to validation.		
5.7	The runtime environment is not susceptible to SQL,LDAP,OS,XML Injection, or that security controls to prevent the Injection attacks.		
5.8	All untrusted data that are output to HTML (including HTML elements, HTML attributes, JavaScript data values, CSS blocks, and URI attributes) are properly escaped for the applicable context.		
5.9	If the application framework allows automatic mass parameter assignment (also called automatic variable binding) from the inbound request to a model, verify that security sensitive fields such as accountBalance,role, password etc. are protected from malicious automatic binding.		
5.10	The application has defenses against HTTP parameter pollution attacks, particularly if the application framework makes no distinction about the source of request parameters (GET, POST, cookies, headers, environment, etc.)		
6.0 Output Encoding/Escaping			
6.1	All untrusted data that are output to HTML (including HTML elements, HTML attributes, JavaScript data values, CSS blocks, and URI attributes) are properly escaped for the applicable context.		
6.2	All output encoding/escaping controls are implemented on the server side.		
6.3	Output encoding /escaping controls encode all characters not known to be safe for the intended interpreter.		

6.4	All untrusted data that is output to SQL interpreters use parameterized interfaces, prepared statements, or are escaped properly.		
6.5	All untrusted data that are output to XML,LDAP,OS use parameterized interfaces or are escaped properly.		
6.6	All untrusted data that are output to any interpreters not specifically listed above are escaped properly.		
6.7	For each type of output encoding/escaping performed by the application, there is a single/centralized security control for that type of output for the intended destination.		
7.0 Cryptography Requirements			
7.1	All cryptographic functions used to protect secrets from the application user are implemented on server side.		
7.2	All cryptographic modules fail securely.		
7.3	Access to any master secret(s) is protected from unauthorized access (a master secret is an application credential stored on disk which is used to protect access to security configuration information).		
7.4	Password hashes are salted uniquely when they are created.		
7.5	Cryptographic module failures are logged.		
8.0 Error Handling and Logging			
8.1	All logging controls are implemented on the server.		
8.2	Verify security logging controls, provide the ability to log both success and failure events that are identified as security-relevant.		
8.3	All events that include untrusted data will not execute as code in the intended log viewing software.		
8.4	Single logging implementation is used by the application.		
8.5	Application does not log application-specific sensitive data that could assist an attacker, including user's session ids and personal or sensitive information.		
8.6	All code implementing or using error handling and logging controls is not affected by any malicious code.		
9.0 Data Protection			
9.1	All forms containing sensitive information have disabled client side caching, including autocomplete features.		
9.2	All sensitive data is sent to the server in the HTTP message body (i.e., URL/GET parameters are never used to send sensitive data).		
9.3	All cached or temporary copies of sensitive data sent to the client are protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data (e.g., the proper no-cache and no-store Cache-Control headers are set).		
9.4	There is a method to remove each type of sensitive data from the application at the end of its required retention period.		
10.0 Network Communication Security			
10.1	A path can be built from a trusted CA to each Transport Layer Security (TLS) server certificate, and each certificate is valid.		
10.2	Failed SSL/TLS connections do not fall back to an insecure connection.		

10.3	SSL/TLS is used for all connections (including both external and backend connections) that are authenticated or that involve sensitive data or functions.		
10.4	SSL/TLS connection failures are logged.		
10.5	Certificate paths are built and verified for all client certificates using configured trust anchors and revocation information.		
10.6	All connections to external systems that involve sensitive information or functions use an account that has been set up to have the minimum privileges necessary for the application to function properly.		
10.7	There is a single standard SSL/TLS implementation that is used by the application that is configured to operate in an approved mode of operation		
11.0 HTTP Security			
11.1	"Redirect" (i.e. 302 Object moved) do not include unvalidated data. Form data redirect may be hijacked if compromised or mismanaged. If it is redirected to a site in a different domain, the users cannot tell whether the site is trusted or not before sensitive data contained in the form are submitted. Therefore, we recommend to NOT implement "redirect" when accepting sensitive information from user forms.		
11.2	The application accepts only a defined set of HTTP request methods, such as GET and POST.		
11.3	Every HTTP response contains a content type header specifying a safe character set (e.g., UTF-8).		
11.4	The HTTPOnly flag is used on all cookies that do not specifically require access from JavaScript.		
11.5	The secure flag is used on all cookies that contain sensitive data, including the session cookie.		
11.6	HTTP headers in both requests and responses contain only printable ASCII characters and do not expose detailed version information of system components.		
11.7	The HTTP header, X-Frame-Options is in use for sites where content should not be viewed in a 3rd-party X-Frame. A common middle ground is to send SAME ORIGIN, meaning only websites of the same origin may frame it.		
11.8	The application generates a strong random token as part of all links and forms associated with transactions or accessing sensitive data, and that the application verifies the presence of this token with the proper value for the current user when processing these requests.		
11.9	The HTTP header can be easily manipulated by an attacker and must not be used for security decisions.		

Web Secure Coding Checklist (Bidder to provide response) *Bidder should complete this checklist in order to ensure compliance. This reflects best industry practice and correlates directly to issues that are identified during Vulnerability Assessments.*		*Availability Response*	*Bidder Response*
1.0 Application Verification			
1.1	The integrity of interpreted code, libraries, executables, and configuration files is verified using checksums or hashes.		
2.0 Authentication			
2.1	All pages and resources require authentication except those specifically intended to be public.		
2.2	All password fields do not display the user's password when it is entered, and that password fields (or the forms that contain them) have autocomplete disabled.		
2.3	If a maximum number of authentication attempts is exceeded, the account is locked for a period of time long enough to deter brute force attacks.		
2.4	All connections to external systems that involve sensitive information or functions are authenticated.		
2.5	The forgotten password function and other recovery paths do not reveal the current password and that the new password is not sent in clear text to the user.		
2.6	The username enumeration is not possible via login, password reset, or forgot account functionality.		
2.7	All authentication controls are enforced on the server side.		
3.0 Session Management			
3.1	The framework's default session management control implementation is used by the application.		
3.2	Sessions are invalidated when the user logs out.		
3.3	Sessions timeout after a specified period of inactivity.		
3.4	Sessions timeout after an administratively-configurable maximum time period regardless of activity (an absolute timeout).		
3.5	All pages that require authentication to access them have logout links.		
3.6	The session id is never disclosed other than in cookie headers; particularly in URLs, error messages, or logs. This includes verifying that the application does not support URL rewriting of session cookies.		
3.7	The session id is changed on each login.		
3.8	The session id is changed on re-authentication.		
3.9	The session id is changed or cleared on logout.		
3.10	Only session ids generated by the application framework are recognized as valid by the application.		
3.11	Authenticated session tokens are sufficiently long and random to withstand attacks that are typical of the threats in the deployed environment.		
3.12	Cookies which contain authenticated session tokens/ids have their domain and path set to an appropriately restrictive value for that site. The domain cookie attribute restriction should not be set unless for a business requirement, such as single sign on.		

3.13	Verify that authenticated session tokens using cookies sent via HTTP, are protected by the use of "HttpOnly".		
3.14	Verify that authenticated session tokens using cookies are protected with the "secure" attribute and a strict transport security headers are present.		
3.15	Verify that the application does not permit duplicate concurrent user sessions, originating from different machines.		
4.0 Access Control			
4.1	Users can only access URLs for which they possess specific authorization.		
4.2	Direct object references are protected, such that only authorized objects are accessible to each user.		
4.3	All connections to external systems that involve sensitive information or functions use an account that has been set up to have the minimum privileges necessary for the application to function properly.		
4.4	Directory browsing is disabled unless deliberately desired.		
4.5	The same access control rules implied by the presentation layer which are enforced on the server side, such that controls and parameters cannot be re-enabled or re-added from higher privilege users.		
4.6	All user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized.		
4.7	Verify the system can protect against aggregate or continuous access of secured functions, resources, or data. For example, possibly by the use of a resource governor to limit the number of edits per hour or to prevent the entire database from being scraped by an individual user.		
4.8	There is a centralized mechanism (including libraries that call external authorization services) for protecting access to each type of protected resource.		
4.9	Verify that the application or framework generates strong random anti-CSRF tokens unique to the user as part of all high value transactions or accessing sensitive data, and that the application verifies the presence of this token with the proper value for the current user when processing these requests.		
4.10	Limitations on input and access imposed by the business on the application (such as daily transaction limits or sequencing of tasks) cannot be bypassed.		
4.11	All access controls are enforced on the server side.		
5.0 Input Validation			
5.1	The runtime environment is not susceptible to buffer overflows, or that security controls prevent buffer overflows.		
5.2	All input validation failures result in input rejection.		
5.3	All input validation or encoding routines are performed and enforced on the server side.		
5.4	Single/Centralized input validation control is used by the application for each type of data that is accepted.		
5.5	All input validation failures are logged.		
5.6	All input data is canonicalized for all downstream decoders or interpreters prior to validation.		

5.7	The runtime environment is not susceptible to SQL,LDAP,OS,XML Injection, or that security controls to prevent the Injection attacks.		
5.8	All untrusted data that are output to HTML (including HTML elements, HTML attributes, JavaScript data values, CSS blocks, and URI attributes) are properly escaped for the applicable context.		
5.9	If the application framework allows automatic mass parameter assignment (also called automatic variable binding) from the inbound request to a model, verify that security sensitive fields such as accountBalance,role, password etc. are protected from malicious automatic binding.		
5.10	The application has defenses against HTTP parameter pollution attacks, particularly if the application framework makes no distinction about the source of request parameters (GET, POST, cookies, headers, environment, etc.)		
6.0 Output Encoding/Escaping			
6.1	All untrusted data that are output to HTML (including HTML elements, HTML attributes, JavaScript data values, CSS blocks, and URI attributes) are properly escaped for the applicable context.		
6.2	All output encoding/escaping controls are implemented on the server side.		
6.3	Output encoding /escaping controls encode all characters not known to be safe for the intended interpreter.		
6.4	All untrusted data that is output to SQL interpreters use parameterized interfaces, prepared statements, or are escaped properly.		
6.5	All untrusted data that are output to XML,LDAP,OS use parameterized interfaces or are escaped properly.		
6.6	All untrusted data that are output to any interpreters not specifically listed above are escaped properly.		
6.7	For each type of output encoding/escaping performed by the application, there is a single/centralized security control for that type of output for the intended destination.		
7.0 Cryptography Requirements			
7.1	All cryptographic functions used to protect secrets from the application user are implemented on server side.		
7.2	All cryptographic modules fail securely.		
7.3	Access to any master secret(s) is protected from unauthorized access (a master secret is an application credential stored on disk which is used to protect access to security configuration information).		
7.4	Password hashes are salted uniquely when they are created.		
7.5	Cryptographic module failures are logged.		
8.0 Error Handling and Logging			
8.1	All logging controls are implemented on the server.		
8.2	Verify security logging controls, provide the ability to log both success and failure events that are identified as security-relevant.		
8.3	All events that include untrusted data will not execute as code in the intended log viewing software.		
8.4	Single logging implementation is used by the application.		

8.5	Application does not log application-specific sensitive data that could assist an attacker, including user's session ids and personal or sensitive information.		
8.6	All code implementing or using error handling and logging controls is not affected by any malicious code.		
9.0 Data Protection			
9.1	All forms containing sensitive information have disabled client side caching, including autocomplete features.		
9.2	All sensitive data is sent to the server in the HTTP message body (i.e., URL/GET parameters are never used to send sensitive data).		
9.3	All cached or temporary copies of sensitive data sent to the client are protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data (e.g., the proper no-cache and no-store Cache-Control headers are set).		
9.4	There is a method to remove each type of sensitive data from the application at the end of its required retention period.		
10.0 Network Communication Security			
10.1	A path can be built from a trusted CA to each Transport Layer Security (TLS) server certificate, and each certificate is valid.		
10.2	Failed SSL/TLS connections do not fall back to an insecure connection.		
10.3	SSL/TLS is used for all connections (including both external and backend connections) that are authenticated or that involve sensitive data or functions.		
10.4	SSL/TLS connection failures are logged.		
10.5	Certificate paths are built and verified for all client certificates using configured trust anchors and revocation information.		
10.6	All connections to external systems that involve sensitive information or functions use an account that has been set up to have the minimum privileges necessary for the application to function properly.		
10.7	There is a single standard SSL/TLS implementation that is used by the application that is configured to operate in an approved mode of operation		
11.0 HTTP Security			
11.1	"Redirect" (i.e. 302 Object moved) do not include unvalidated data. Form data redirect may be hijacked if compromised or mismanaged. If it is redirected to a site in a different domain, the users cannot tell whether the site is trusted or not before sensitive data contained in the form are submitted. Therefore, we recommend to NOT implement "redirect" when accepting sensitive information from user forms.		
11.2	The application accepts only a defined set of HTTP request methods, such as GET and POST.		
11.3	Every HTTP response contains a content type header specifying a safe character set (e.g., UTF-8).		
11.4	The HTTPOnly flag is used on all cookies that do not specifically require access from JavaScript.		
11.5	The secure flag is used on all cookies that contain sensitive data, including the session cookie.		

11.6	HTTP headers in both requests and responses contain only printable ASCII characters and do not expose detailed version information of system components.		
11.7	The HTTP header, X-Frame-Options is in use for sites where content should not be viewed in a 3rd-party X-Frame. A common middle ground is to send SAME ORIGIN, meaning only websites of the same origin may frame it.		
11.8	The application generates a strong random token as part of all links and forms associated with transactions or accessing sensitive data, and that the application verifies the presence of this token with the proper value for the current user when processing these requests.		
11.9	The HTTP header can be easily manipulated by an attacker and must not be used for security decisions.		

API Security Review Checklist (Bidder to provide responses, If applicable)			Availability Response	Bidder Response
INSTRUCTIONS: Fill in all applicable sections.		Guide		
1.0	Authentication & Authorization	Acceptable Criteria	(Yes/No)	
1.1	Use secure authentication mechanism	Don't use basic authentication with plaintext credentials e.g. plaintext password in URL parameter etc. Apply standard & secure authentication such as JWT tokens with dynamic mechanism per session/per request, OAuth 2.0, or public/private API keys combination.		
1.2	Ensure secure storage of passwords, API tokens & keys	Store API tokens/keys in secure key vaults via secure mechanism such as Windows Local Security Authrority so that tokens & keys remain secure including the config file data.		
1.3	Ensure encryption of sensitive data & tokens in transit and at rest	Sensitive data including credentials must be encrypted in transit and at rest. Use transport layer security protocols and strong encryption algorithms e.g. RSA, AES-256 etc.		
1.4	Never place the credentials in source code	Plaintext credentials or hashes must not be placed in source code to avoid misuse & brute force attacks by adversaries.		
1.5	Configure maximum login retries following the IS policy of organization	For NBP assets, 5 maximum retries should be allowed before the account gets locked. It prevents brute force attacks.		
2.0	Access Security			
2.1	Use HTTPS insead of HTTP	Implement SSL based communication over API connections.		
2.2	Display as minimum information as possible in you API request/response	Don't rely on client side to filter data; Avoid using generic methods such as to_json() and to_string(). Instead, cherry-pick specific properties & data you really want to return.		
3.0	Input Security			

3.1	Sensitive data protection in URL	Don't use any sensitive data (credentials, passwords, security tokens, and/or API keys) in the URL but use standard Authorization header.		
3.2	Use appropriate HTTP method according to the operation	Use GET (read), POST (create), PUT/PATCH (replace/update), and DELETE (to delete a record) methods appropriately in API communication. Respond with *405 Method Not Allowed* if the requested method is not appropriate for the requested resource.		
3.3	Ensure content validation controls for input security	Content Validation for Request: To validate the content type of response, use Accept header in HTTP request (Content Negotiation) to allow only the supported formats (e.g., application/xml, application/x-www-form-urlencoded, multipart/form-data, application/json etc.). Content Validation for SQL injection, RCE and XSS: Validate the user-submitted content for SQL injection, Remote Code Execution, and Cross-Site Scripting (XSS).		
3.4	Ensure Secure Coding Practice	Remove unused dependencies, unnecessary features, components, files, and documentation. Run depenciy check tools such as OWASP Dependency check.		
3.5	Make trusted updates of packages	Always check for trusted sources. Get the packages for your application with authorized signature so that no malicious component is included in the package.		
3.6	Apply caching and rate limiting	Use an API Gateway service to enable caching, rate limit policies (e.g., Quota, Spike Arrest, or Concurrent Rate Limit) and deploy API resources dynamically.		
4.0	Output Security			
4.1	Ensure content validation controls for output security	Content Validation for Response: Validate the content type of returned data via Content-type header of HTTP response. It should match with the request's Accept header. Respond with 406 Not Acceptable response if it is not matched.		
4.2	Use appropriate HTTP response headers for output security	Recommended Use: • X-Content-Type-Options: Nosniff • X-Frame-Options: Deny (if there are no frames used in application) • X-Frame-Options: Sameorigin (in case frames are to be used in application) • Content-Security-Policy: default-src 'none' • Remove fingerprinting headers like *X-Powered-By*, *X-AspNet-Version*, etc. • Don't return sensitive data like credentials or security tokens in response • Return the proper status code according to the operation completed (e.g., 200 OK, 400 Bad Request, 401 Unauthorized, 405 Method Not Allowed, etc.).		
5	Data Processing Security			

5.1	Ensure Object level authorization	• User's own resource ID should be avoided. Use /me/orders instead of /user/654321/orders • Don't auto-increment IDs. Use UUID instead		
5.2	Ensure XML External Entities (XXE) prevention	• Enternal entites' misconfiguration may lead to SSRF (Server-Side Request Forgery) and billion laugh attacks. Configure the XML parser to disable external entity resolution • The XML parser should be configured to use a local static DTD and disallow any declared DTD included in the XML document		
5.3	Ensure data rate limiting	.Rate limit the data processing wherever applicable in order to avoid brute force attacks.		
5.4	Do not use test environment in production mode	Make sure your application is set to production mode before deployment. Running a debug API in production could result in performance issues & unintended operations such as test endpoints and backdoors. It may expose data sensitive to the organization or development team.		
6	Monitoring Security			
6.0	Ensure API logging & monitoring mechanism	The API logs must be stored in a centralized log management system. API monitoring includes auditing, logging, and version control for all APIs and their components. This helps in the troubleshooting process when and if a problem occurs.		
6.1	Limit number of API calls	Set a quota on the API calls count, i.e. put limitations on the number of times an API is called.		

Annexure-II: Cloud Based Checklist

Cloud Based Outsourcing Service Arrangements Cloud Service Provider's (CSP) Compliance Matrix				
Clause ID	Clause Description			
		Consent/comments by Responsible functions	Existing Controls at CSP	Remarks (If Any)
E. PERMISSIBLE CLOUD OUTSOURCING ARRANGEMENTS				
1	Please confirm whether this service/function is material or non-material as per below mentioned definition provided by SBP. 1. For the purpose of these regulations, material workload means all systems, applications, and services that are fundamental for carrying out business of an RE (Regulated Entity)/bank, and if disrupted, have the potential to significantly impact an institution's business operations, reputation or profitability.			

	2. REs may outsource their workloads to CSPs in the following manner: a) All type of workloads (i.e. material and non-material) may be outsourced to reputable onshore (i.e. domestic) CSPs; b) EMIs, non-designated PSOs/ PSPs may outsource their material and non-material workloads to offshore (i.e. outside Pakistan) CSPs; c) Banks, MFBs, DBs, DFIs and designated PSOs/PSPs may outsource their non-material workloads to offshore CSPs. However, outsourcing of their material workloads to offshore CSPs shall be subject to SBP approval whereby SBP may grant approval on case to case basis, after considering the following: i. Systemic implications of the CO (cloud outsourcing) arrangement ii. Institution specific risks iii. Legal and other strategic risks iv. Data processing and storage v. Availability and quality of services vi. Security and other controls vii. Contingency and exit planning viii. Resilience ix. Sub-contracting x. Assurance mechanism xi. Role and responsibilities d) For approval to outsource material workloads to offshore CSPs, banks, MFBs, DBs and DFIs shall submit their request to BPRD whereas designated PSOs/PSPs shall submit it to PSP&OD; e) While granting approval to banks, MFBs, DBs, DFIs and designated PSOs/PSPs, SBP may impose additional terms and conditions over and above the requirements of this framework; f) For CO arrangements, REs may use any service and deployment model as per their requirements and risk appetite; g) REs shall give preference to onshore CSPs for outsourcing their workloads; h) REs shall not process and store their data in unfriendly or hostile jurisdictions;			
	3. Outsourcing of services to CSPs does not absolve the REs from their prime responsibilities including managing and running the business operations effectively, legal and regulatory compliance, and protection of customers' data.			
	4. SBP may instruct any RE to restrict outsourcing of their workloads to CSPs due to its systemic impact, unacceptable risks and any other concerns.			
	5. REs shall submit details of their CO arrangements to SBP, as and when required.			

	6. SBP may instruct REs to shift their cloud based workloads to SBP designated onshore community cloud as and when the same is available.			
F. GOVERNANCE				
2	1. Develop a comprehensive policy for CO duly approved by their BoD. In this regard, the REs can also amend their existing 'Policy for Outsourcing Arrangements' to include/update CO. The policy shall encompass all services, as per requirements provided in section E above that can be outsourced to CSP, and at least include all those aspects that have been prescribed in this framework.			
	2. Conduct appropriate due diligence of CSPs and proactively identify any risks emanating from their CO arrangements including risks associated with sub-contracting by CSPs.			
	3. Update their Enterprise Risk Management framework or other relevant policies for effective oversight and management of risks emanating from the CO arrangement(s). The framework shall include sub-contracting risks, assessment of cloud service location(s) especially if outside Pakistan with specific focus on areas including but not limited to legal aspects; regulatory issues; jurisdictional concerns; availability; security and resilience of information assets and services; connectivity; political and security situation; ease of oversight; plausibility of RE, SBP and external audit staff to travel for onsite assessment/ audit or alternate assurance mechanism.			
	4. Delegate cloud specific governance responsibilities such as for overseeing adherence to regulatory as well as performance requirements, including cloud service SLAs, reviewing of KPIs and KRIs, incidents (including cybersecurity incidents) and other relevant matters to the ITSC. In this regard, the ToRs of ITSC shall be suitably amended to include these responsibilities			
	5. Undertake all CO arrangements through legally binding SLAs, which shall at least include the areas prescribed in Appendix – I. These SLAs must be vetted by the legal function of the RE and be executed by the REs (except branches of foreign banks) themselves instead of their parent company or subsidiary, with governing law preferably as law of Pakistan. However, compliance with the laws of Pakistan, along with compliance to any legal obligation as prescribed by the host country of the CSP, shall be mandatory at all times.			
	6. Define and agree upon the roles and responsibilities of the IT and operational departments (e.g. requirement analysis, performance testing, UAT, responsibility for data validation, etc.), before transferring information assets and services to a CSP.			

	7. Ensure that the CO arrangements are compliant with the relevant legal and regulatory requirements, and the IA functions of the REs shall provide independent certification to their BoDs in this regard.			
G. DUE DILIGENCE OF CLOUD SERVICE PROVIDER				
3	REs shall exercise reasonable care before entering into CO arrangements. To ensure effective management of the associated risks, REs shall conduct reasonable due diligence of the CSPs and their material sub-contracting arrangements by using defined criteria which shall include the following:			
	1. Evaluation of feasibility of CO arrangements including cost effectiveness, quality of service, and legal / regulatory / compliance risks.			
	2. Ability of CSP to meet legal and regulatory requirements of Pakistan.			
	3. Assessment of financial strength and resources.			
	4. Competence, business structure, experience, track record in delivering such services.			
	5. Assessment of CSPs ability to comply with necessary minimum controls including physical security / internal controls based on the intended workloads, especially with respect to confidentiality, integrity, availability and resilience.			
	6. Assessment of corporate governance and entity level controls.			
	7. Assessment of CSPs ability to provide REs the control over data residency enabling them to shift over preferred data center instance, depending upon the cloud service model, in order to host these services at such locations/countries/regions considering geopolitical risks.			
	8. Cybersecurity and IT capabilities including adherence to international standards and best practices.			
	9. Sub-contracting risk management.			
	10. Data security related controls.			

	11. Access, audit and information rights of REs, SBP and external auditors of the REs.			
	12. Support services.			
	13. Contingency, resilience and exit arrangements.			
	14. Up to date certification and attestation of the CSPs including but not limited to IT service delivery, business continuity & disaster recovery, cyber / information security, and data center Tier III certification.			
	15. Liability of claims / penalties on CSPs for: a) Unauthorized transactions; b) Service disruptions; c) Security breaches; d) Enforcement and penal actions that may be taken by regulatory and legal authorities against the REs for not complying the regulatory and legal requirements, due to faults by CSPs.			
	16. For material workloads: a) Threat & Vulnerability Assessment or equivalent independent assessments of data centers to identify the security and operational weaknesses. The scope of such assessments shall include physical and environmental security, perimeter security, access controls, security & emergency procedures, monitoring, redundancy, natural disasters, and the political and economic climate of the country in which the data center resides. The assessment shall cover all data centers where the REs' data / systems will reside; b) Independent assessments instead of solely relying on attestations by the CSPs, and the results shall be reviewed by REs' IA as well as Information Security (IS) function. However, the REs may consider SOC Reports (level 1, 2 & 3).			
H. OVERSIGHT				
4	The risk exposures and effectiveness of the corresponding controls may vary over the tenure of the CO arrangements. In this regard, REs shall:			
	1. Develop and maintain an effective oversight mechanism including but not limited to the assessment of performance against desired service levels and ongoing viability of the CSP and its services, cybersecurity practices and controls, changes in service location(s), sub- contracting, change of ownership, control environment; and timely response to emerging risks and issues.			
	2. On an ongoing basis, review and monitor the CSP's compliance with legal, regulatory and contractual obligations.			

	3. Monitor access to their cloud data/workloads, wherever possible such as through cloud activity reports.			
	4. Review internal control assessment / audit reports of the CSPs, in order to obtain assurance regarding the security and resilience.			
	5. For material workload, conduct comprehensive audit of the CSPs, either themselves or through third party assessors / pooled audits, at least once in two years. The scope of the audit shall at least include the infrastructure and related software used to deliver cloud services to the RE. However, in case where audit/onsite assessment cannot be conducted due to a valid reason(s), REs may rely on internationally recognized third party certifications and reports made available by the CSPs, after sufficient understanding and review of their scope, methodology and the ability of the assessors.			
I. CONTINGENCY PLANNING				
5	REs shall develop contingency plan for their CO workloads in order to deal with any disruption/degradation of cloud related services. The contingency plan shall take into account all possible scenarios regarding the unavailability of CSP related services due to various reasons such as technical/connectivity issues, inability of CSP to provide services due to legal actions in their respective jurisdictions, etc. In this regard, REs shall:			
	1. Prudently select appropriate implementation option (service and deployment model, availability zones, server/server-less, etc.) along with related communication technologies, to offer better resilience that commensurate with their workload.			
	2. Maximize the redundancy by design and workload distribution, and implement health and monitoring checks for ensuring HA of their cloud workloads.			
	3. Ensure redundant and robust connectivity arrangements through different international internet cables and include penalties for disruption and downtimes in their agreements.			
	4. Define clear roles and responsibilities including responsibility for signing off, updating and activating the contingency plan.			
	5. Periodically review and update the contingency plan, taking into account developments which may affect their feasibility. These may include increase in number of availability zones, changes in business requirements, new viable alternate CSPs, technological changes, etc.			

	6. Periodically (at least once annually) test the contingency plan against various scenarios including disruption of internet / communication services, unavailability of CSP, etc. Where possible, REs may conduct collaborative testing of their contingency plan with their CSPs. Further, the REs shall ensure that the deficiencies identified during testing are recorded and corrective actions are implemented.			
J. RIGHT TO AUDIT, ACCESS AND INFORMATION				
6	REs shall ensure that CO does not hinder SBP in conducting its supervisory functions. In this regard, the REs shall comply with the following requirements:			
	1. Ensure that their internal & external auditors/ independent assessors and SBP have right to conduct audits and onsite assessments of the CSP and its sub-contractors, if required. Further, there should be no restriction or prohibition on access to REs' cloud related information assets and services for the RE, its auditors, independent assessors or SBP's authorized staff or such visits are otherwise not impractical.			
	2. Ensure that access, audit and information rights provided through the contractual arrangement include where relevant: a) Premises, data, devices, information, systems, and networks used for providing the cloud services or monitoring its performance. These may include CSP's (and its sub-contractors) policies, processes, and controls; b) Results of security testing carried out by CSP or on its behalf, on its applications, data, and systems to assess the effectiveness of the implemented cybersecurity processes and controls; c) Results of security testing carried out by the sub-contractors or on its behalf, on its applications, data, and systems, where applicable, to ascertain effectiveness of the cybersecurity processes and controls; d) Company and financial information; e) CSPs' external auditors, personnel, and premises.			
	3. In case, where audit/onsite assessment cannot be conducted for a valid reason(s), REs may rely on internationally recognized third party certifications, and reports made available by the CSP. However, reliance on these third party certifications and reports shall be supported by adequate understanding and review of the scope, the methodology applied therein and the ability of the third party and CSP to clarify matters relating to the assessment. Further, the REs shall have contractual right to request for the expansion of the scope of the certifications / assessments / audits to cover relevant controls and systems.			

	4. Ensure that CSPs timely provide any information requested by SBP, whenever required.			
	5. Follow up with the CSP to ensure that all appropriate and timely remediation actions are taken to address any audit findings.			
K. EXIT PLANNING				
7	REs shall develop an exit plan by considering the materiality and impact of their workloads outsourced to CSPs. In this regard, REs shall comply with the following requirements:			
	1. Ensure that the exit plan covers scenarios for stressed and non-stressed exit circumstances.			
	2. Ensure that the exit plan has defined trigger events, alternative solutions, transition plans, and roles and responsibilities including responsibility for signing off, updating and activating the plan.			
	3. Periodically review and test the exit plan, taking into account developments which may affect its feasibility.			
	4. Implement measures including but not limited to contractual or escrow arrangements, to ensure continuity of critical business services in case of exit.			
	5. Ensure complete removal of data including logs from all locations of CSP in case of exit.			
	6. To avoid the lock-in and dependency risks, REs shall in their CO arrangement contracts: a) Avoid inclusion of any lock-in clause or exclusivity arrangements; b) Ensure that they have right to terminate the CO agreement at least in the following circumstances: i. Change in ownership of the CSP ii. Insolvency or liquidation of the CSP iii. CSP goes into judicial administration iv. CSP is in breach of applicable laws, regulations or contractual provisions v. Significant and material breach of security or confidentiality vi. Demonstrable deterioration to perform the contracted service c) Ensure that the minimum termination period is documented in their CO contracts.			
L. SUB-CONTRACTING				

8	The CO arrangements expose REs to various risks relating to sub-contracting due to improper/non implementation of controls by the sub-contractors. For material sub-contracting, the REs shall comply with the following requirements:			
	1. Have visibility of the CSP supply chain by ensuring that the CSP maintains and provides an updated list of its sub-contractors.			
	2. Consider potential impact of large, complex sub-contracting by the CSPs on their operational resilience, and ability to oversee and monitor the emanating risks.			
	3. Only permit sub-contracting by CSPs if such arrangements do not give rise to excessive operational risks, and sub-contractor agrees to comply with all applicable legal, regulatory, contractual, audit and access requirements including granting REs and SBP contractual access, audit and information rights.			
	4. Review material sub-contracting agreements of the CSP and ascertain that the legal, regulatory, operational, and cybersecurity requirements are complied with, throughout the supply chain.			
	5. Ensure that the CSPs have the capability and capacity to oversee any material sub- contracting on an ongoing basis.			
M. USER ACCESS MANAGEMENT AND AUTHENTICATION				
	REs shall implement complete life cycle of user access management for their cloud related workloads, while complying with the following requirements:			
	1. Implement at-least four eye principle for user access administration.			
	2. Periodically review user access right changes independently.			
	3. Ensure that the use and access of service, generic and administrative accounts are controlled and monitored. Moreover, the REs shall implement MFA and limit use of these accounts through dedicated machines only.			
	4. Create separate account of administrative users for routine operations, and implement enhanced password controls (length, complexity, age).			
	5. Implement MFA and IP source restrictions (wherever possible) for their users accessing cloud environment.			
	6. Monitor and document the use of master account, and permit its usage only under exceptional circumstances.			

	7. Implement access controls for data backups including log data, of cloud related workloads.			
	8. Ensure that CSPs do not have access to REs' systems, software and data.			
N. CHANGE AND CONFIGURATION MANAGEMENT				
	REs shall plan and implement configuration management in conjunction with IT change management to ensure safe and secure operations of cloud services. In this regard, REs shall comply with the following requirements for their cloud related workloads:			
	1. Implement mechanism for detecting unauthorized changes to cloud environment, and configure automated alerts for the changes.			
	2. Ensure CM procedures for cloud related workloads are documented and mutually agreed with the CSP. These shall at least include change request and approval procedures, change prioritization and impact assessment, change reporting, roles and responsibilities, timeframe for patching and software releases.			
	3. Ensure that the CSPs have well-defined and robust CM controls, and notify the REs in advance of the changes.			
	4. Ensure that the changes are tested before their implementation on the production environment.			
	5. Define roles and responsibilities of REs staff for configuration management of the cloud environment, and at least segregate infrastructure, security and application roles.			
	6. Create and maintain baselines for cloud hosted systems, and periodically review, monitor, report and remediate non-compliance with the baselines.			
O. INCIDENT MANAGEMENT				
9	Effective and efficient remediation of the incidents requires timely detection and proper integration with the incident response and management processes. With the increase in sophistication of cyber-attacks, there is a need to use advance analytics to correlate events across multiple systems. For incident management, REs shall comply with the following requirements for their cloud related workloads:			

	1. Define and document criteria, performance requirements and procedures for escalation, notification, containment and closure of incidents (including IT, cyber incidents) in consensus with the CSP(s).			
	2. Ensure access to incident and root cause analysis reports of the CSPs.			
	3. Designate a SIRT to provide timely response to IT/cyber incidents. In this regard, roles and responsibilities of CSP and REs' teams shall be formalized.			
	4. Ensure that CSPs shall provide reasonable access to necessary information to assist in any REs' investigation arising due to an incident in the cloud.			
	5. Ensure that the CSPs conduct formal post incident review of the material cloud related incidents and provide their report to the RE.			
	6. Ensure that ITSC as part of their oversight reviews and discusses cloud related incidents.			
	7. Periodically review and test Incident Response Plan of cloud related workloads at least once annually, keeping in view cybersecurity as one of key considerations.			
P. DATA SECURITY				
10	Outsourcing of the workloads to the CSPs does not relieve the REs from the responsibility of safeguarding data confidentiality and integrity. In this regard, REs shall:			
	1. Encrypt data at rest (including backups) and in transit using strong and non-obsolete cryptographic algorithms.			
	2. Desensitize the production data before porting or using it on non-production environment(s).			
	3. Ensure that their data in the cloud environment is clearly identifiable and segregated.			
	4. Take appropriate measures for the protection of Personally Identifiable Information (PII); and ensure compliance with the requirements of laws of Pakistan at all times. Further, REs shall ensure that CSPs do not disclose their data to any third party including foreign governments / courts / law enforcement agencies without their consent.			
	5. Ensure that CSP does not use their data for any commercial purposes.			

	6. Implement reasonable backup and restoration testing mechanism, depending on the nature of the workloads in compliance with the defined RPOs. Further, the backup mechanism shall have ransomware protection, and the backup restoration testing exercise shall be conducted at least on a half-yearly basis.			
	7. Classify information assets in terms of their sensitivity, confidentiality & availability, and implement additional controls for high value cloud information assets.			
	8. Implement controls to prevent unauthorized exfiltration of data from the cloud environment. These controls shall include deployment of content inspection technologies, controls on data downloading and extraction, monitoring unusual data access, etc.			
	9. Ensure that they are notified about any proposed change in the location of their data, and have contractual right to reject such change, or terminate the CO arrangement on such grounds.			
	10. Ensure that data is deleted from all storage locations of the CSP following an exit or termination of the CO arrangement, and where applicable, from the systems of any sub- contractor by requesting written confirmation from the CSP.			
Q. CRYPTOGRAPHIC KEY MANAGEMENT				
11	CSPs use cryptographic controls to secure access and segregate customers' data. Hence, the security of the cryptographic keys is critical for ensuring the data security. CSPs offer a variety of key management options/features, which can be selected by the REs for implementation based on the significance of their workloads. In this regard, the REs shall comply with the following requirements:			
	1. Develop and implement policies and procedures governing the lifecycle of the cryptographic material.			
	2. Ensure that details of the cryptographic algorithms and other related parameters such as key lengths, renewals etc. are reviewed by a subject matter expert.			
	3. Ensure that details of the location, ownership and management of the encryption keys and HSM are agreed with the CSP and documented.			
	4. Ensure that the cryptographic keys are unique and generated only by the REs. Further, REs must have the sole ability to administer/manage these keys and HSM.			
	5. Periodically change the cryptographic keys in accordance with the international standards and best practices			

	6. Ensure that the encryption keys are stored separate from the virtual images and information assets.			
	7. For material workloads, deploy/implement HSM with due controls.			
R. TOKENIZATION				
12	Depending on the sensitivity of data workload, REs may implement tokenization to minimize the data footprint. While implementing tokenization, REs shall assess and evaluate the features and data interactions of the tokenization solution. REs shall also ensure that the CSPs do not have access or control over the tokenization solution.			
S. NETWORK ARCHITECTURE				
	The network structure and logical layout is of paramount importance in any cloud implementation. Therefore, the REs must implement controls for protection against plausible threats/attacks including cloud specific attacks, by implementing the following requirements:			
	1. Ensure security of their CO arrangements and on premise environments by implementing controls at appropriate locations to detect and mitigate security breaches and ongoing attacks. These controls shall include but not limited to perimeter security, network IDS/IPS, WAF, DDoS protection, etc.			
	2. Implement network segmentation based on type of workloads (e.g. production, pre- production, quality assurance, development, etc.) and purpose (e.g. end-users, critical servers, other servers, middleware, interface, etc.). In this regard, a dedicated network segment (i.e. management network) not accessible from other operational segments shall be implemented for administration purposes. Further, all internet traffic shall be routed through a dedicated network segment (i.e. security segment) and other network segments shall not have direct internet access.			
	3. Secure traffic between the cloud and on premise environment using a VPN or direct network connection with stringent access control rules configured to ensure routing of traffic from/to dedicated source and destination IPs.			
	4. Monitor and control access to and security of the cloud environments. In this regard, regularly review the firewall rules and access lists, especially after any changes.			
T. SECURITY TESTING				

13	The dynamic and evolving nature of cyber threats requires a high degree of validation and testing of security posture of an enterprise, on periodic basis. However, security testing of the systems and applications in the cloud environment is challenging due to the inherent shared service model. In this regard, REs shall comply with the following requirements:			
	1. Conduct vulnerability assessment, penetration testing and scenario based security testing of their systems hosted with the CSPs on periodic basis, at least once annually.			
	2. Ensure that CSPs conduct vulnerability assessment and penetration testing for the infrastructure and applications managed by them, at least once annually in order to provide security assurance to the REs. Further, the REs shall be fully cognizant of the scope of such assessments while examining the results.			
	3. Ensure that security testing is conducted while taking into consideration various scenarios and threats that are unique to cloud services including but not limited to hypervisor jumping, weak application programming interfaces, DoS hyper jacking, wrapping attacks, cloud malware injection, side channel attacks, etc.			
	4. Ensure that all vulnerabilities identified for their cloud related workloads are categorized in terms of risk, tracked and rectified (including post validated). For the infrastructure and applications managed by the CSPs, REs shall implement alternate mechanism for obtaining assurance that the vulnerabilities are timely rectified.			
	5. In case of material CO, ensure independent Threat & Vulnerability Assessment of CSP's data centers hosting the data/systems of REs, at least once annually.			
U. SECURITY EVENT MONITORING				
14	REs shall establish mechanisms for SEM by complying with the following requirements:			
	1. Wherever possible, leverage the controls/tools available in the cloud environment to enforce consistent security standards and baselines, automated response, remediation and notification.			
	2. Integrate CSP related services with their SIEM solutions to provide a detailed analysis of the security logs. In this regard, cloud specific incident scenarios with correlation rules shall be implemented. Further, AI and ML driven technologies may be explored and preferably adopted, where available.			
	3. Ensure that cloud related activities are effectively monitored by their SOC on 24x7 basis.			
V. OTHER REQUIREMENTS				

15	1. REs shall monitor and review capacity utilization of their cloud workloads.			
	2. REs shall provide adequate training of the cloud environment, to their end-users and privileged users.			
	3. All security incidents / breaches shall be reported to SBP in compliance with the requirements specified in the 'Enterprise Technology Governance & Risk Management Framework for Financial Institutions' or as advised by SBP from time to time. Further, REs shall conduct investigations to identify the root cause, take appropriate actions to prevent recurrence of such incidents in future and fix responsibility for such lapse.			
	4. For material workloads, REs shall provide following information to SBP1 one month before placing their services with the CSPs: a) Name of the CSPs, and their parent company (if any); b) Description of the activities and details of data to be placed with the CSP; c) Date of commencement / renewal / expiry of services; d) Last contract renewal date (where applicable); e) Service and Deployment Models.			