БАНК РОССИИ

BANK OF RUSSIA
RECOMMENDATIONS ON STANDARDISATION

RS BR IBBS-2.6-2014

MAINTENANCE OF INFORMATION SECURITY OF THE RUSSIAN BANKING SYSTEM ORGANISATIONS

MAINTENING INFORMATION SECURITY IN THE LIFE CYCLE PHASES OF AUTOMATED BANKING SYSTEMS*

Effective date: 2014-09-01

Moscow
2014

Foreword

ADOPTED AND ENFORCED by Bank of Russia Resolution No. R-556 dated 10 July 2014.

These recommendations on standardisation may not be fully or partially reproduced, duplicated or distributed as an official publication without the consent of the Bank of Russia.

Content

Introduction

Pursuant to the current Bank of Russia Standard entitled "Maintenance of Information Security of the Russian Banking System Organisations. General Provisions" (hereafter, STO BR IBBS-1.0), organisations of the Russian Federation banking system (RF BS) shall take appropriate action to ensure the information security (IS) of automated banking systems (ABS) in all life cycle phases.

IS is provided in the life cycle phases of the ABS to fulfil the following main tasks:

- implement the necessary requirements within the ABS to ensure IS as established by the legislation of the Russian Federation, including Bank of Russia regulations, STO BR IBBS-1.0, and internal documents of a RF BS organisation;
- reduce the risks of IS breaches associated with ABS vulnerabilities;
- monitor the provision of IS within the framework of ABS operations;
- reduce IS breach risks, including information leakage risks during the maintenance and upgrading of ABS, and during the decommissioning of ABS;
- effective ABS upgrading where impermissible IS breach risks associated with the operation of the ABS are identified.

To determine the recommendations related to the fulfilment of these tasks, this document specifies regulations:

- for work management in the life cycle phases of the ABS, including work that enables monitoring to build confidence in the performance of the given works and, therefore, confidence in the provision of ABS IS;
- for the delineation of the standard defects in the implementation of the requirements for the provision of ABS IS, generating the conditions for the occurrence of impermissible IS breach risks in ABS operations (hereafter, standard defects in ABS IS);
- for the delineation, content and procedure of ABS software source code control, ABS security assessment and controls establishing the parameters for technical safeguards (identification of configuration errors).

BANK OF RUSSIA
RECOMMENDATIONS ON STANDARDISATION

MAINTENANCE OF INFORMATION SECURITY OF THE RUSSIAN BANKING SYSTEM ORGANISATIONS

MAINTENING INFORMATION SECURITY IN THE LIFE CYCLE PHASES OF AUTOMATED BANKING SYSTEMS

Effective date: 2014-09-01

1. Scope of Application

These recommendations apply to RF BS organisations implementing the requirements of STO BR IBBS-1.0 for IS provision in the life cycle phases of the ABS to build up (or improve) the IS maintenance system, and to entities engaged by RF BS organisations to carry out work in the life cycle phases of the ABS.

This document shall be adopted by RF BS organisations and other entities as a reference and/or its provisions may be directly employed in the internal documents and contracts executed by RF BS organisations.

This document is a recommendation, and the RF BS organisation may replace certain provisions with other provisions of their own that ensure the equivalent (similar) level of ABS IS in various life cycle phases.

2. Regulatory References

These Bank of Russia recommendations on standardisation contain normative references to the following documents:

STO BR IBBS-1.0;

Bank of Russia Recommendations on Standardisation — Maintenance of Information Security of the Russian Banking System Organisations. IS Methodology for Assessing the Risks of IS Breaches (RS BR IBBS-2.2).

3. Terms and Definitions

These recommendations apply terms in accordance with STO BR IBBS-1.0, and also the following terms defined accordingly:

3.1. Confidence means the state of confidence in the fact that ABS meets the IS requirements determined for ABS.

3.2. IS function means the implemented functionality of one or more ABS components associated with IS.

3.3. IS function (application) interface means the description and implementation of the methods for employing the IS function.

3.4. Functional requirements for IS maintenance means the requirements for the IS functions relating to ABS components, as well as their application interfaces.

4. Designations and Abbreviations

ABS refers to Automated Banking System
WS refers to workstation
DBMS refers to Database Management System
TR refers to Technical Requirements
ITR refers to Individual Technical Requirements
IS refers to information security
RF BS refers to the Russian Federation Banking System

5. General Provisions

5.1. Within the framework of these recommendations, ABS is defined as an interconnected combination of software and hardware including: telecommunications, computer aids, system software, application software, and information security means.

The main functionality of ABS is its provision of the automation of bank information and payment processes, including material safeguards, and it is implemented through one or more specialised bank applications incorporated in an ABS. The remaining components, including system software, computer aids, and information security means are defined as the environment for the functioning of the specialised bank applications (hereafter, ABS associated components).

5.2. The provision of ABS IS is implemented through the provision of IS for ABS components, which consists in the application and operation of safeguards for specialised bank applications, as well as safeguards for all ABS associated components. A combination of safeguards for specialised ABS bank applications, and the safeguards for all ABS associated components is defined as an ABS IS subsystem.

It should be taken into account that ABS associated components may be used to ensure the operation of a few different specialised bank applications; therefore, IS functions of such associated components can be used with different ABS systems, and their safeguards are incorporated in the IS subsystems of different ABS systems.

5.3. Considering that ABS associated components may be targeted by an intruder, ensuring IS in the life cycle phases of the ABS requires the implementation of appropriate arrangements both for specialised bank applications and for all ABS associated components.

Regarding work management in the life cycle phases of the ABS, keep in mind that in a number of cases, ABS associated components are made by different entities, they are mostly delivered on an as-is basis, and the entity developing specialised bank applications (hereafter, developer) lacks complete and reliable information about the consistency of the implementation of security functions related to ABS associated components.

5.4. Subject to STO BR IBBS-1.0, the life cycle phases of the ABS are as follows:

1) Development of Technical Requirements (TR);
2) Engineering;
3) Making and Testing;
4) Commissioning and Start-Up;
5) Operation;
6) Maintenance and Upgrading;
7) Decommissioning.

5.5. You may be confident in the implementation of ABS IS only if there is certain evidence of the completeness and consistency of IS arrangements in the life cycle phases of ABS components, at least as pertains to specialised bank applications. In terms of verifying the reliability of the above, the following should be reviewed:

- regulations used to manage IS activities in the life cycle phases of the ABS;
- documented results from the performance of IS activities in the life cycle phases of the ABS.

The compilation of evidence of reliability created by each organisation is collected in each life cycle phase, and based on the assessment of such evidence, the completeness and consistency of implementing the IS requirements specified for ABS may be established.

5.6. Management of the creation of the ABS, including the IS subsystem, should be based on the provisions of the set of standards and guidelines for automated systems entitled "Information Technology:, including GOST 34.601-90, "Information Technology. Set of Standards for Automated Systems. Automated Systems. Stages of Development" (hereafter, GOST 34.601-90).

6. Technical Requirements Development Phase

6.1. The main objective in the TR development phase in terms of ensuring IS is to determine the requirements for ensuring IS for the ABS which are to be included in the TR (hereafter, TR for IS maintenance).

Keep in mind that at this phase, the information required to establish the specific functional requirements for the implementation of IS by ABS components is usually incomplete. Specific functional requirements for IS maintenance can be established only after determining the basic engineering solutions of the ABS. Therefore, the TR for IS maintenance should be general in nature (not specific), without any reference to concrete implementations; however, requirements should nonetheless be clear and unambiguous.

6.2. TR for IS maintenance should comply with GOST 34.602-89, "Information Technology. Set of Standards for Automated Systems. Technical Directions for Creating Automated Systems".

6.3. TR for IS maintenance are determined (or drawn up) on the basis of IS requirements established by the legislation of the Russian Federation, including Bank of Russia regulations, STO BR IBBS-1.0, and internal documents of the RF BS organisation, which shall be implemented for the ABS.

Documents containing the IS requirements used to make the TR for IS maintenance should be created on the basis of the following data:

- the types of information (information assets) for processing and/or storage in the ABS;
- bank processes of the RF BS organisation established for their automation;
- technologies and means of information processing to be used for ABS implementation (where similar data is available).

6.4. Where TR for IS maintenance are specified, the following should be established:

- the necessity and expediency of using information security means certified under information security requirements;
- the necessity and expediency of involving an organisation with a license for conducting activity related to the technical protection of confidential information, in order to make, upgrade, operate and decommission the ABS.

6.5. Where TR for IS maintenance are formulated, an additional preliminary analysis should be carried out in respect to relevant information security threats. In this phase, IS threats should be formulated generally in terms of business processes, operations and functions of the RF BS organisation.

In the event that functional requirements for IS maintenance can be formulated in this phase to neutralize the relevant threats or to provide compensation for potential damages, these requirements should be included in the scope of the TR for IS.

6.6. TR for IS should comprise requirements for using IS functions for the ABS associated components, which are applied to provide IS to specialised bank applications of different ABS systems on the part of specialised bank applications (requirements for the integration of specialised bank applications with shared ABS associated components).

6.7. TR for IS maintenance should also include:

– IS requirements associated with the assignment and distribution of roles in ABS;
– IS requirements associated with access and registration control;
– IS requirements associated with protection against the impact of malicious code;
– IS requirements associated with the use of public networks and data transmission channels;
– IS requirements associated with the use of cryptographic information protection facilities;
– IS requirements associated with the implementation of controls over the implemented security measures;
– IS requirements associated with the implementation of IS monitoring, including monitoring to identify IS incidents in ABS;
– requirements for safe technologies used in information processing (process information security measures).

6.8. TR for the ABS is the recommended primary source for the IS requirements in the ABS engineering phase.

7. ABS Engineering Phase

7.1. The main objectives in the ABS engineering phase in terms of IS include:

— establishing and documenting the functional requirements implemented by ABS components that provide the observance of TR for IS maintenance;
— determining the scope of functions involved in providing IS that are implemented by shared ABS associated components;
— selecting the scope of safeguards (technical and/or organisational) for the implementation of IS functions in accordance with the functional requirements for IS linked to ABS components, including the selection of information security means certified as meeting information security requirements;
— formulating the primary definition of parameters for configuring the settings of technical safeguards (configuration standards);
— determining and documenting the interfaces for ensuring IS functions that are implemented in ABS components;
— developing the primary determination of rules for executing technical safeguards, including rules for the updating, management and control of their configuration parameters;
— determining the requirements (scope and content) for regulations associated with the implementation of organisational safeguards;
— formulating the primary determination of the scope of roles of subjects with access to ABS (operating personnel, users, program processes), the scope of access resources (databases, file resources, virtual machines, other access resources), access rights and roles of access subjects (reading, recording, operational or other types of access) where access resources are accessed;
— formulating the primary determination of requirements for ABS IS subsystem staffing.

7.2. ABS engineering in terms of IS provision should commence from the development of ABS IS subsystem architecture that should include the following descriptions:

— the proposed implementation of TR for IS maintenance by means of the planned ABS components;
— the proposed implementation of IS functions for shared ABS associated components;
— the proposed interface of ABS components for ensuring IS in ABS.

ABS architecture for ensuring IS should be developed on the basis of:

— data on the division of ABS into the components, scope and functions of specialised bank applications and ABS associated components;
— the identification (for standard ones) or description (for those developed as part of ABS or individually) of interfaces between ABS components;
— data on ABS software, including system software purchased in boxes (for ABS systems made through the modification of specialised applications — data on the packages of specialised applications).

7.3. The design for the ABS IS subsystem should be based on the expediency of implementing:

— centralized management and control of technical safeguards, including software updates and updates to applicable signature bases, and the identification and control of their configuration parameters;
— ABS integration with IS monitoring infrastructure components, and identification of IS incidents applicable to the RF BS organisation. In this, the functional requirements for IS maintenance should include requirements for the scope of data for monitoring IS generated by ABS components during ABS operation;
— the maximum possible degree of use of the functions for ensuring IS generated by the shared ABS associated components.

The baseless upgrading of shared ABS associated components used in operations should not be part of designs for the ABS IS subsystem. Where the said components are upgraded, upgrading and testing should be arranged and performed to ensure IS for all ABS systems using the shared supporting component that has been upgraded.

7.4. Functional requirements for IS maintenance should be established in the engineering phase, including:

— functional requirements for IS maintenance in respect of specialised bank applications;

- functional requirements for IS maintenance in respect of providing components of the ABS developed;
- requirements for using functions that ensure IS in respect of shared ABS providing components.

Functional requirements for IS maintenance should be documented in the individual TR for ABS (hereafter, ITR for an ABS IS subsystem).

In the event that functional requirements for IS maintenance are established in the TR development phase, it is unnecessary to document them again with ITR for ABS IS subsystems.

7.5. In order to ensure the complete implementation of the TR for IS maintenance, procedures should be implemented for compliance control related to TR for IS, and the functional requirements for IS maintenance included in ITR for the IS subsystem.

The functional requirements of ITR for the ABS IS subsystem should be classified based on the performance of Clause 7.4 hereof, and documented with individual ITR subsections for the ABS IS subsystem.

Where identical functional requirements are specified for various ABS components, they should be reproduced in the relevant ITR subsections for the ABS IS subsystem.

7.6. Functional ITR for an ABS IS subsystem should be regarded as the main document to be followed in certifying reliability in subsequent phases of the ABS life cycle.

7.7. When designing an ABS IS subsystem, configuration standards should be determined and formulated (i.e. documents containing the list and reference values of the configuration parameters of ABS components, including technical safeguards). The adoption of configuration standards and control over the compliance of actual values of configuration parameters with their reference values is the main technique in preventing vulnerabilities caused by errors in configuring ABS components.

In order to establish configuration standards for ABS associated components from commercially available software (including operating systems, database management systems, other system software), standardised reference books with configuration parameters for ensuring IS should be used (e.g. the National Checklist Program Repository [1]).

7.8 As for ABS systems whose components are supposed to be located on computer equipment belonging to clients of the RF BS organisation, the following specifications and documentation are advised:

- the scope of components transferred to the client;
- actions taken to ensure the integrity of software components transferred to the client;
- requirements for the functional environment of the client's ABS component;
- requirements and procedure of client transferring information about IS problems and incidents that have occurred when the client used ABS components;
- requirements and methods for restoring ABS components operated by the client, and requirements for upgrading their operational environment.

7.9. Design documents developed in the engineering design development phase comprise design documents for the ABS IS subsystem. Determination of the scope and structure of the design documentation for an ABS should be based on the provisions laid out in the guiding document RD 50-34.698-90, "Automated Systems. Requirements for Document Contents", and GOST 34.201-89, 'Information Technology. Set of Standards for Automated Systems. The Type, Comprehensiveness and Designation of Documents when Building Automated Systems", and GOST 19.201-78, "Unified System for Program Documentation. Technical Specifications for Development. Requirements for Contents and Formatting".

7.10. Design documents for ABS IS subsystems should be developed in compliance with the following principles:

- Design documents shall contain documented results of the tasks set out in Clause 7.1 hereof.
- Design documents shall enable the monitoring of the completeness and consistency of implementation of IS functions in accordance with the requirements specified in the ITR for an ABS IS subsystem in the implemented engineering solutions.

8. The ABS Development and Testing Phase

8.1. The main objectives in the ABS development and testing phase in terms of ensuring IS include:

- managing versions and modifications of the specialised bank applications that have been developed;
- providing IS for the development and testing environment of the ABS components;
- testing (pre-testing) ABS components, including specialised bank applications;
- testing (pre-testing) ABS components designed for operation on the computer equipment of the clients of the RF BS organisation;
- developing operational documentation.

8.2. The main recommended objectives to manage versions and modifications of ABS software components that have been developed to ensure IS include:

- monitoring the compliance with specific requirements that have been implemented in ITR for an ABS IS subsystem in the particular version (assembly) of specialised bank applications being developed;
- formalising the procedure for input file storage and handling, and taking appropriate actions to prevent unauthorised changes to versions of specialised bank applications.

8.3. In order to manage versions and modifications of specialised bank applications being developed, a version and modification management system should be used that provides:

- the labelling (numbering) of interim versions of the specialised bank applications being developed;
- identifying source files used to assemble each interim version of specialised bank applications being developed, including source code files, resource files, documentation files;
- the labelling of versions (revisions) of source files.

8.4. As for ensuring IS in respect of the ABS components' development and testing environment, protection against the following IS threats should be provided:

- unauthorised changes to source files of the software being developed;
- suspension of the development and testing process resulting in failure to meet the timing set to issue the final software version developed, including suspension due to the abnormal performance of development tools or destruction of a certain portion of source files;
- unauthorised familiarisation of third parties with source files and software documentation, and any other information leakage during the development of ABS components;
- loss of rights (licenses) to use development tools.

To counteract threats to developers, protective measures should be adopted and documented, including:

- monitoring of physical access to computer equipment used in the development and testing phase for ABS software components;
- identifying computer network segments containing computing machinery that is used in the development of specialised bank applications;
- identifying computer network segments containing computing machinery used in testing specialised bank applications and ABS associated components;
- arranging and monitoring the isolation and information interface of the development segment, testing segment and computer network segments containing computing machinery that is used to implement bank processes;
- managing access to resources and the means of developing and testing specialised bank applications, including access to input files;
- recording and monitoring activity with the input files of specialised bank applications;
- providing virus protection;
- monitoring the use of communication ports in computer equipment.

8.5. Actual data resulting from the implementation of bank processes should not be used in the development and testing environment.

In the event that testing requires data approximating actual data to the maximum extent, data testing sets should be made up through irreversible depersonalization, concealment and/or distortion of details resulting from the implementation of bank processes. Testing should not include the use of any data subject to requirements for ensuring IS established in the legislation of the Russian Federation, including Bank of Russia regulations, internal documents of the RF BS organisation and/or under agreements with clients and counterparties.

8.6. The completeness and consistency of implementation of requirements specified in ITR for ABS IS subsystem should be tested in three steps:

- directly during the development of ABS software components;
- prior to the issue of the final version of ABS software components developed;
- during ABS pre-testing.

8.7. During testing within the framework of development of specialised bank applications, self-testing should be provided to check consistency of implementation of requirements specified in ITR for ABS IS subsystem in respect of specialised bank applications developed. In such case, the interaction of specialised bank applications developed with ABS associated components and their IS functions is not tested and the functions themselves are emulated by means of test programs. Such testing should be performed by developers in the development environment of the specialised bank application.

8.8. Prior to issuing the final version of specialised bank applications, the comprehensiveness and consistency of the performance of the requirements specified in ITR for an ABS IS subsystem in respect of specialised bank applications being developed should be tested, and this testing should be based on an interaction with ABS associated components, including those that are shared. Such testing should be carried out by developers in a testing environment comprising all ABS components located and set in accordance with the design documentation, and ensuring reproduction of their operational conditions that is close to the actual.

8.9. In order to test specialised bank applications, it is advised to develop and maintain a test program comprising control over the performance of all the requirements for ensuring IS as specified in ITR for ABS IS subsystem, including situations with incorrect values for input data, non-operability of IS provision functions of other ABS associated components and other potential non-standard modes of functioning by the ABS.. The testing program shall identify all tests and demonstrate, through relevant tests, that the requirements specified in ITR for the ABS IS subsystem are fully implemented.

The testing methodology for each test should be documented based on information about interfaces for the IS functions, which includes input data, sequence of checks, expected test results, and test success/failure criteria.

Test completion should be supported by a test report containing the test date, testing method, input data used for testing, outcome, and assessment of success or failure of the test.

By the issue of the final version of ABS software components being developed, the consistency of implementation of their security functions shall be supported by test reports verifying the successful completion of all tests envisaged by the test program.

8.10. As for the ABS software components which implement the bank payment process or which are designed to process personal data or any other information, and for which a security requirement has been established by the legislation of the Russian Federation or a decision made by the RF BS organisation, source code control should be provided prior to pre-testing in order to identify standard programming errors and any other defects involving the occurrence of vulnerabilities.

Source code control recommendations are given in Appendix 2 to this document.

8.11. During ABS pre-testing, complete testing should be carried out, independently or jointly with the developer, in order to check the completeness and consistency of the implementation of all the requirements specified in ITR for an ABS IS subsystem in connection with all ABS components. Preliminary testing should be performed in a testing environment that comprises all the ABS components configured and adjusted to comply with the design documentation. In this, the reproduction of operational conditions must be close to actual operations.

Preliminary testing should be carried under a test methodology and program that provides, in respect of each interface of each IS function, for test procedures that are consistent with the interface. Testing shall support the consistency of:

- implementation of the function ensuring IS where it is accessed through the interface being tested;

– calling the required IS function for ABS components, including shared functions.

Tests shall demonstrate compliance of the results of performing the IS function on the pre-established sets of input data with security requirements specified in the ITR.

8.12. Preliminary testing should be based on GOST 34.603-92, "Information Technology. Types of Tests of Automated Systems".

8.13. The design documentation for ABS should be adjusted as required subsequent to the results of implementing the build and testing of the ABS.

8.14 Operational documents, including instructions for operating personnel (and the ABS IS Administrator), should comprise the following information:

- a description of the scope of safeguards for ABS components;
- a description of the scope, location requirements, and configuration parameters (configuration standards) of technical safeguards;
- a description of the rules for executing technical safeguards, including the rules for safeguard updates, management and operation control, along with the configuration of parameters;
- requirements and regulations for implementing organisational safeguards;
- requirements for ABS IS subsystem personnel, description of the roles and functions of operating personnel;
- requirements for the scope and contents of organisational arrangements to be conducted to deploy and operate the ABS IS subsystem,including role assignment to operating personnel, training, information sharing and improving the knowledge base of both operating personnel and users;
- a description of rules and procedures for ensuring information security if the ABS is decommissioned or in the event of a cessation of information processing.

8.15. For ABS systems with components that are to be located on computer equipment of clients of the RF BS organisation, operational documents should comprise individual documents designed to regulate the operation of the ABS components on the client's side:

- a description of the scope of ABS components on the client's side;
- the procedure for implementing actions to ensure the integrity of specialised bank applications and ABS associated components transferred to the client's side;
- requirements for the scope, versions and necessary settings for IS software in the ABS functioning environment on the client's side;
- the procedure for upgrading ABS components operated on the client's side, and requirements for upgrading the software components of their functioning environment;
- requirements for the scope, versions, upgrading, and settings of technical safeguards applied on the client's side.

9. The Acceptance and Commissioning Phase

9.1. The main objectives in the acceptance and commissioning phase in terms of ensuring IS include:

- control over the deployment of ABS components in the information infrastructure of the RF BS organisation used to implement bank processes (hereafter, industrial and production environment);
- a trial operation;
- the elimination of defects in the implementation of the requirements specified in ITR for ABS IS subsystem;
- acceptance testing.

9.2. The following is advisable to control the deployment of ABS components in the industrial environment:

- monitoring the consistency of versions and the integrity of specialised bank applications during the transfer from the development and testing environment to the industrial environment;
- monitoring the performance of requirements set out in the design and operational documentation as related to the location and establishment of the configuration parameters for technical safeguards, implementing organisational safeguards, and the determination and assignment of roles.

9.3. GOST 34.603-92 should be referenced during the trial ABS operation.

9.4. Within the framework of trial operation in terms of IS provision, the consistency of the operations of the ABS IS subsystem in the industrial environment should be checked, as well as the capability of implementing design and operational documents during operation related to:

- monitoring the operation of technical safeguards, including updating rules, and the management and control of their configuration parameters;
- monitoring the implementation of organisational safeguards;
- requirements for ABS IS subsystem personnel.

9.5. Additionally, within the framework of trial operations, an integrated assessment of security is advised, including:

- testing for breaches;
- the identification of known vulnerabilities in ABS components.

An integrated assessment of security should be provided without any notification made to personnel involved in the trial ABS operation, which, inter alia, will allow the assessment of personnel readiness to fulfil the requirements specified in the documents of the RF BS organisation in terms of IS incident response.

Recommendations for security assessment are given in Appendix 3 to this document.

9.6. Based on trial operation results, the following is recommended:

- Documentation of the defects identified in the implementation of the ABS IS subsystem;
- In respect of each defect, an assessment of risks and a decision on whether they can be eliminated during operation;
- The creation of defect elimination plans as related to the implementation of the ABS IS subsystem;

- The conduction of activities to eliminate defects that are critical in terms of ensuring IS during the operation of the ABS IS subsystem.

9.7. Upon the elimination of defects in the implementation of the ABS IS subsystem, it is advisable to decide on the scope and necessity of arrangements related to the re-testing and trial operation of the ABS and/or its components in view of the updates that have been made and the level of criticality of the eliminated defects.

9.8. Based on the results of trial operation, the necessity of updating the design should be considered, and where such updating is required, the operational documentation may also need to be updated.

9.9. Upon the elimination of critical defects in the implementation of ABS IS subsystem identified during trial operation, acceptance tests are carried out. The determination of the scope and procedure of acceptance testing should be based on GOST 34.603-92.

9.10. Acceptance testing is based on the results of preliminary tests, a trial operation, and the outcome of the elimination of critical defects identified during trial operation. Moreover, within the framework of acceptance testing, selective arrangements may be conducted to test IS functions as envisaged within preliminary testing.

10. Operation Phase

10.1. The main objectives in the operation phase in terms of ensuring IS include:

- monitoring the scope, locations and configuration parameters of technical safeguards;
- monitoring the compliance with instructions for technical safeguards, including upgrading and management rules;
- monitoring the performance of regulations associated with the implementation of organisational safeguards;
- monitoring the implementation of actions taken to ensure the integrity of specialised bank applications and ABS associated components that have been transferred to the client's side, and the delivery of the necessary documents included in the operational documentation for clients;
- ensuring that ABS IS subsystem personnel are fully qualified;
- monitoring the carrying-out of organisational arrangements required to ensure the operation of the ABS IS subsystem, including the assignment of operating personnel roles, training, information sharing and improving the knowledge base of both operating personnel and users;
- monitoring the readiness of operating personnel to operate the ABS IS subsystem;
- ensuring that users are informed of the instructions for operating the ABS IS subsystem;
- regularly assessing ABS security (identifying standard vulnerabilities of ABS software components, testing for breaches);
- ensuring that notifications are made of ABS vulnerabilities and responding to such notifications.

Recommendations for monitoring the configuration parameters of technical safeguards (identifying configuration errors) are given in Appendix 4 of this document.

10.2. The frequency of consistency assessment is determined by the RF BS organisation. As for ABS systems used to implement the bank payment process, an integrated consistency assessment is advisable at least once a year.

10.3. Software vulnerability notifications can be obtained from various sources, such as:

- notifications published by Computer Incident Response Centres (e.g. CERT Notifications [2]), payment systems (e.g. VISA Payment System Notifications [3]), hardware and software manufacturers (e.g. ORACLE Notifications [4]);
- notifications published in public vulnerability databases and distributed by subscription;
- notifications of ABS vulnerabilities addressed by external specialists to the RF BS organisation or published in public sources, to which end methods for effective communication with relevant specialists of the RF BS organisation should be provided.

10.4. The following activities should be arranged:

- the identification of ABS systems whose components include software with the identified vulnerabilities;
- determining the degree of criticality of the identified vulnerabilities for the implementation of bank processes of RF BS processes;
- taking decisions to eliminate any vulnerability within ABS maintenance and upgrading arrangements where such vulnerability is confirmed.

11. ABS Maintenance and Upgrading

11.1. The main objectives in the ABS maintenance and upgrading phase in terms of ensuring IS include:

- checking the performance of the ABS IS subsystem in the testing environment when ABS components are updated as part of ABS maintenance;
- updating operational documents where applicable versions of ABS associated components are modified;
- preventing information leakage during ABS maintenance operations, including those involving external companies;
- preventing information leakage when computer equipment is repaired by external companies;
- monitoring the complete performance of events in the life cycle phases of the ABS when the ABS is upgraded.

11.2. In order to prevent information leakage under ABS maintenance (or upgrading) operations, including those involving external companies, persons carrying out ABS maintenance (or upgrading) should be supervised by RF BS organisation employees, and liability should be imposed on these RF BS employees for unauthorised and/or unregulated operations performed during maintenance (or upgrading).

11.3. ABS upgrading comprises, to the necessary extent, such phases as the development of technical requirements, engineering, building and testing, acceptance and commissioning.

12. Decommissioning Phase

12.1. The main objectives in the decommissioning phase in terms of ensuring IS include:

- monitoring compliance with the rules and procedures for ensuring information security during the decommissioning of an ABS;
- archiving the information contained in the ABS when it is to be used subsequently;
- ensuring the complete destruction (or erasure) of data and residual information from the technical ABS information carriers and/or destruction of the technical ABS information carriers.This should be carried out in compliance with the legislation of the Russian Federation, including Bank of Russia regulations and the internal documents of the RF BS organisation.

Appendix 1

Standard Defects in the Implementation of Automated System Security Functions

1. Common Defects in ABS systems and Bank Applications

1.1 Access control

1.1.1. The possession by users of access rights that are not required for their carrying out process operations corresponding to their responsibilities within the RF BS organisation.

1.1.2. The presence in the processing account, on behalf of which the ABS component is functioning, of access rights that are not required to carry out operations provided for the given ABS component in the design documentation.

1.1.3. The presence of process accounts in ABS with standard passwords set automatically when software is installed.

1.1.4. Implementation of discretionary, mandatory or other models of access control in ABS, instead of the role-based model.

1.1.5. The absence of methods incorporated in the ABS for filing reports on users and their privileges.

1.1.6. The implementation of access control functions only at the ABS level.

1.1.7. The presence, in the ABS graphical user interface, of control elements designed to carry out operations that the user is not authorised to perform.

1.1.8. No restrictions to the number of the user's concurrent connections (sessions) in ABS. This makes it easier for intruders to gain access to accounts belonging to RF BS organisation employees.

1.2. Identification and authentication

1.2.1. Lack of authentication from the server side in case of an interface between the user and ABS, and between various ABS components.

1.2.2. An interaction between ABS components that lacks authentication by the interaction initiator.

1.2.3. Use of authentication protocols that allow the insecure transfer of user authentication data (including unencrypted data transfer or through reversible transformation).

1.2.4. A narrowing in the conversions of authentication data in authentication algorithms (e.g. reducing user identifier letters and/or passwords to one case, limiting the number of significant password characters).

1.2.5. The use of predictable identifiers (e.g. derivatives from the user's first and last names that agree with identifiers in e-mails, serial numbers, identifier formation under the same algorithm).

1.2.6. No obligatory limits on the minimum complexity of passwords (e.g. a limitation on the minimum password length, availability of characters of various classes, lack of correspondence between the password and user identifier, lack of correspondence between the new password and a previous password).

1.2.7. The use, where new accounts are created, of the same original password, or creating passwords under the same algorithm, and the failure to require changes in the original password when the user logs in for the first time.

1.2.8. Storage of user passwords in ABS through reversible transformations. Unauthorised access by an intruder to the OS or DBMS of ABS server components compromises all the accounts of the given ABS and individual accounts in other ABS systems of the RF BS organisation.

1.2.9. The employment of procedures enabling the independent recovery or changes to passwords forgotten by users.

1.2.10. A lack of preliminary authentication when the user changes a password. In some instances, the intruder may pass authentication by setting a new user password.

1.2.11. Password character display during entry.

1.2.12. Lack of response to the automated selection of user identifiers and passwords including:

– a lack of automatic temporary account lockout when the predetermined number of unsuccessful authentication attempts is exceeded;
– a lack of arrangements to exclude automated password selection (such as CAPTCHA).

1.2.13. Where the predetermined number of unsuccessful authentication attempts is exceeded triggering an automated account lockout, the lack of automatic account unlocking in the set time interval, thus enabling the intruder to lock user access to ABS.

1.2.14. The necessity of executing certain ABS software modules with OS administrator rights. In the event that software code vulnerabilities of the application are available, the intruder may completely control both the application and the operating system.

1.2.15. User authentication through the software code in the workstation (hereafter, WS) where user authentication through ABS server components is unavailable, thus enabling the intruder to bypass authentication.

1.2.16. The presence of authentication data required to provide ABS component access to other ABS systems of the RF BS organisation in the ABS component software code and/or in configuration files available to users.

1.2.17. The use of interface protocols that are vulnerable to line-tapping and the repeated use of post-authentication data (hash values of passwords, session identifiers, authentication markers, etc.), which is vulnerable to line-tapping and repeated use.

1.3. The recording of events and review of event logs

1.3.1. The absence or shutdown of time synchronizers of the operating system.

1.3.2. The absence of recording mechanisms for particular types of events that are essential for incident investigations, including:

- the creation of new accounts, and change in account access rights;
- unsuccessful operations (e.g. authentication errors, insufficient access rights when operations are performed, inaccessibility of interfaces to ABS components);
- the response of security functions directed at counteracting computer attacks (e.g. automatic lockout of accounts, automatic session termination, receipt of incorrect input data at external interfaces and ABS interfaces);
- the performance of operations envisaged by the threat model as part of threat implementation.

1.3.3. The lack of significant information in the event log data about logged events that enable determining the circumstances in which events have occurred.

1.3.4. The availability of confidential and sensitive data (user passwords, payment card data, etc.) in the event log data.

1.3.5. Logging of certain events only by ABS components that may be available to the intruder (e.g. User WS, publicly available web servers).

1.3.6. Insecure storage of event logs (e.g. in a publicly available file that may be changed by users or database table).

1.3.7. The possibility for users to alter event logging parameters

1.3.8. A lack of incorporated or dedicated tools for event log analysis, including event searches by predetermined criteria (by the name or user identifier, date, time, etc.).

1.3.9. A lack of arrangements for the effective notification of ABS administrators about events bearing the marks of a security incident.

1.4. Input and output processing

1.4.1 A lack of preliminary checks of input data correctness (e.g. verifying limitations in respect of the text string length, the absence of inadmissible characters or combinations of characters, the agreement of numeric values and boundary conditions).

1.4.2. The presence of sensitive information in error messages seen by users (e.g. authentication data and information used to identify software for ABS components, diagnostic information).

1.4.3. Lack of checks of input data correctness including:

- the possible composition of executable files and scenarios by ABS server components based on input data that are set by users;
- possible inclusion of fragments that fail to meet specifications of interface protocols and/or are used to operate standard vulnerabilities in output data transmitted between ABS components.

1.5. Cryptographic protection

1.5.1. The use of protocols that fail to provide cryptographic data protection to ensure the interface of ABS components (including those within the control area).

1.5.2. The lack of technological capability to use the certified DET application during operations that require cryptographic data protection (including cases when non-certified DET can be used as decided by the RF BS organisation management).

1.5.3. During the use of a certified DET application, the performance of cryptographic operations using a software interface that is only standard for the particular DET model.

1.5.4. Use of a procedure for generating cryptographic keys that enables the user to replicate a symmetric key and/or a private part of an asymmetric key.

1.5.5. The use of software generators that are not part of DET to generate pseudo-random sequences (e.g. to make session identifiers, challenge requests, GUID).

1.5.6. The use of DET in modes and conditions that are not envisaged by the operational documentation for DET.

1.6. Secure architecture and development

1.6.1. The refusal to use protection mechanisms provided by the processor architecture, operating system and code compilers in the software code of ABS components (e.g. buffer overflow protection, protection against disturbed exception handling, protection against code execution in stack and data segments, random segment placement in the address space).

1.6.2. The use of functions of standard libraries that are vulnerable to buffer overflow attacks where similar functions with built-in protection are available.

1.6.3. The lack of preliminary initialization of variables and data structures during random-access memory allotment.

1.6.4. The presence of operating network services for interfaces that are not envisaged by the design documentation in the operating system, DBMS, server components of application software.

1.7. Data protection

1.7.1. The lack of ABS arrangements for residual information clearing where data is deleted.

1.7.2. The lack of protection against unauthorised access to shared resources of the operating system (e.g. to the shared memory, named channels, memory-mapped files).

1.7.3. The inconsistent use of methods to synchronize access to shared resources of the operating system (e.g. critical sections, semaphores).

1.8. Security configuration

1.8.1. The lack of arrangements for providing protection against unauthorised access to application settings.

1.8.2. The lack of capabilities for exporting application settings to a format that can be used for analysis by specialists.

1.9. Integrity and reliability control

1.9.1. The lack of a method in the ABS for controlling software code integrity and the consistency of ABS component settings.

1.9.2. The lack of arrangements for error handling and rollback to a previous state during particular operations.

1.9.3. The lack of mechanisms for the ABS to switch to an emergency mode of operation when a disturbance in software code integrity or setting inconsistencies are identified.

1.9.4. The shutoff of certain security functions when the ABS is switched to the emergency mode operation.

1.9.5. The lack of arrangements for generating diagnostic information when the ABS is switched to the emergency mode operation.

2. Standard Defects of Remote Bank Service Applications and Electronic Payment Means

2.1. Identification and authentication

2.1.1. The use of single-factor authentication during financial operations.

2.1.2. A predictable algorithm for forming one-time passwords and/or the possible repeated use of one-time passwords.

2.2. Transaction security

2.2.1. The use of authorization means (e.g. a simple digital signature) to confirm transactions, thus enabling confirmation by third parties, including employees of the RF BS organisation.

2.2.2. The selection of authorization methods should be based on transaction criticality and the severity of potential problems that may be associated with the provision of authenticity and data integrity. Examples of defects related to the implementation of authorisation methods may include:

– the lack of methods for confirming for non-payment transactions affecting the payment process (creating payment order templates, maintaining reference books with payment recipient details, changing limits, etc.);
– the use of key carriers to make a digital signature, which allow for the export of the private part of the signature key;
– the absence of the capability of signing electronic payment orders by legal entities requiring digital signatures from two authorised persons;
– the possible repeated use of the electronic payment document;
– the lack of pass-through validation of digital signatures in the electronic payment document in all the document handling phases.

3. Standard defects in web applications

3.1. The location of web application components

3.1.1. The location of web servers and other components of several ABS systems in the common demilitarized zone.

3.1.2. The storage of data used by the web server and event logs on the system partition of the hard drive.

3.1.3. The collocation of event logs and system files.

3.1.4. The presence on the web server of test applications and scenarios, and software components that do not form part of ABS.

3.2. Session management

3.2.1. The use of predictable session identifiers.

3.2.2. The possible repeated use of session identifiers (including use of similar identifiers in a few sessions by a user, the constancy of the session identifier after repeated user authentication).

3.2.3. The possible use of the session identifier after session termination.

3.2.4. The disclosure of session identifiers, including unencrypted identifier transmission and identifier inclusion in event log records, as well as in error messages.

3.3. Access management

3.3.1. The lack of access control at the resource identifier level (URI), including possible unauthorised access to separate website sections and objects by specifying their URI in the user web browser.

3.3.2 The potential review of website catalogue contents where such a review is not required.

3.3.3 The use of External Entities, External Parameter Entities, and External Doctypes for XML data processing.

3.4. Data protection

3.4.1. A lack of directives prohibiting data caching in web form parameters intended for confidential information entry.

3.4.2. The transfer of confidential and authentication information in HTTP-GET messages.

3.4.3. Lack of an HTTPOnly attribute in cookie parameters, whose values shall not be available to scenarios implemented by the web browser.

3.4.4. The lack of the secure attribute in the cookie parameters containing sensitive information.

3.5. Input and output processing

3.5.1. The lack of verifications of the consistency of the data entered by the user, or the performance of such verification only by scenarios implemented by the web browser.

3.5.2. Lack of a directive to determine encoding used in HTTP message headers, and also the use of different encodings for various input data sources.

3.5.3. The refusal to apply built-in methods to check the consistency of the input parameters implemented in standard software libraries.

3.5.4. The lack or shutdown of methods to prevent attacks that are associated with the use of standard vulnerabilities of web applications.

3.5.5. The lack of consistency controls for input data to be processed subsequently by software modules allowing for command interpretation (SQL, XPath, LINQ, LDAP, OS command shell, etc.).

3.5.6. Lack of conversion of special characters as provided by HTML language specifications (e.g. replacement of characters ‘<’ and ‘>’ with special characters of HTML language).

4. Standard deficiencies in Database Management Systems

4.1. Operation and availability of DBMS interface protocols that are not envisaged in the design documentation.

4.2. Possible unauthorised access by ABS components to DBMS functions.

4.3. The possession by DBMS administrators of operating system accounts with rights that are not required for DBMS servicing.

4.4. The availability of rights that are not required to carry out operations envisaged by documents in processing accounts used by ABS components for access to DBMS.

4.5. The installation of the DBMS on a server used by other ABS components.

4.6. The location of the DBMS in a demilitarized zone that may be accessed directly by external users.

4.7. The potential access to system tables and configuration settings by users who are not administrators.

4.8. The availability of demonstration databases delivered as part of the DBMS software distribution package in DBMS.

4.9. Allocation of data of several new applications in the same DBMS section where such allocation is not directly envisaged by the design documentation.

5. Standard defects in operating systems

5.1. Access management

5.1.1. Lack of limitations concerning the categories of users entitled to remote access to both the operating system, and the IP addresses that may be used for such access.

5.1.2. Use of insecure and low-security protocols for remote access to the operating system (e.g. TELNET, PPTP).

5.1.3. Possible access to operating system parameter settings, requirements, event logs, and system files by users that are not OS administrators.

5.1.4. The assignment of individual rights of access to operating system objects by certain users (instead of including these users in the relevant groups).

5.1.5. The potential interactive log-in for system accounts used by applications and services.

5.1.6. The possession by a user of rights to write and/or modify files in the home directories of other users.

5.1.7. The lack of disk quotas for accounts (including process accounts).

5.1.8. The non-correspondence of operating system settings with developer's recommendations for secure system settings.

5.1.9. The presence of software in operating systems of ABS server components that is not established in operational documentation.

5.2. Identification and authentication

5.2.1. The display on the log-in invitation of the information that may be used for determining the names of operating system users, or to obtain any details of user passwords.

5.2.2. The capability of accessing the operating system without any authentication via auxiliary and/or infrequently used interfaces (serial ports, etc.).

5.2.3. The lack of user authentication while accessing BIOS parameters, OS core boot loader parameters when the system is in recovery mode (safe mode, single-user mode, etc.).

5.3. System management

5.3.1. The deactivation in the operating system core settings of mechanisms that prevent code execution in the stack and data segments.

5.3.2. The deactivation in the operating system core settings of the file cleaning function / virtual memory swap sector.

5.3.3. The activation in the operating system settings of the capability of loading memory images (dumps) on the disk.

5.3.4. The potential hibernation (switch to the standby mode) via activation in operating system settings.

5.3.5. The disabling of the OS built-in firewall and lack of filtering rules in built-in fire wall settings that block interactions, which is not provided in the operational documentation for ABS, and the disabling of protections for external manufacturers.

6. Standard defects in telecommunication equipment

6.1. Where the operating system may be selected for installation on telecommunication equipment — the installation of operating systems with obviously redundant functions.

6.2. The use in the ABS telecommunication infrastructure of switching equipment that fails to enable disabling redundant interfaces, the controlled connection of network devices (e.g. by MAC addresses or using IEEE 802.1x protocol), protection against ARP spoofing attacks, and network segmentation using VLAN technology.

6.3. VLAN segment settings that allows for the presence of users and ABS servers in the same WS segment, as well as user WS segments and ABS administrator WS segments.

7. Standard defects in virtualization technologies

7.1. The potential access to virtual machine data (e.g. the settings for virtual hardware, disk images) of users who are not virtualization server administrators.

7.2. The provision of virtual machines with access to the shared resources of the operating system of the virtualization server where such access is not directly envisaged in the operational documentation for ABS.

7.3. The lack of methods for monitoring the volume of free resources of the virtualization server.

7.4. The lack of limitations to remote access by virtualization server administrators by limiting IP addresses from which access is allowed, as well as limits to the network interface for access by administrators.

7.5. The use of network interfaces utilized by virtual machines for the remote administration of the virtualization server.

7.6. The storage of event logs of virtualization methods in catalogues available for reading and/or recording by virtual machines.

7.7. Use of hard disk images with dynamically modified sizing in virtual machines.

7.8. Direct access by virtual machines to physical disks and logical partitions of virtualization server memory.

7.9. The use of enhanced mechanisms for data exchange between virtual machines and a virtualization server in the graphic interface of the virtualization server (e.g. drag and drop, copy and paste).

7.10. The use of enhanced mechanisms for data exchange between virtual machines (e.g. the program interface of the virtualization server, virtual sockets).

7.11. The potential modification of virtual machine boot mode by a user.

Appendix 2

Recommendations for Reviewing Source Code

1. General Provisions

1.1. A code review includes measures in respect of certain parts of the source text (source code) of the computer program, written by one or more developers, or by another developer (that was not involved in the creation of this code portion), or duly appointed by any other specialists with the required training and consists in the detailed checking (study, analysis, investigation) of relevant source codes in order to identify unknown vulnerabilities, including those associated with programming errors, violations of established requirements, and other significant defects.

1.2. The subject of investigation is the programme code developed for the ABS components, in particular, the programme code for specialised bank applications.

1.3. A code review may be carried out, where called for, by a few persons, and may include the participation of the developer who created and/or modified the checked code.

1.4. A code review may be performed by the person checking the code both manually (also using effective software code reading techniques) and through methods and means of automated input code analysis, including those that provide:

- static code analysis;
- dynamic code analysis.

2. Manual code review

2.1. A manual source code review (or check) is provided through code examination, analysis and assessment by a person other than the developer. Code assessment may comprise:

- assessment of code compliance with the requirements for code structuring and execution, object naming, division into modules, use of special facilities to ensure appropriate code quality as provided by programming languages and development tools;
- assessment of completeness and quality of code documentation, including documentation of software module headers, function prototypes and data structures, and comments concerning the performance of essential functions;
- assessment of the compliance of algorithms implemented in the source code, software documentation (including the identification of express undeclared capabilities (backdoors), code errors, attempts to obfuscate code, and any other methods used to hinder the review process.

2.2. Effective code reading methods include double reading (firstly, to understand the general structure, and to take note of the main designations and agreements, to be followed by repeated reading to identify defects or variances), the use of scenarios, etc.

2.3. Apart from individual code check techniques, there are methods of effective collaboration between code reviewers, including walkthroughs, code inspections, etc. These are rather resource-intensive, and appropriate skills are necessary for their application; however, when judiciously employed, they may be highly efficient in cases where special examination is required.

3. Static code analysis

3.1. Static code analysis (static_program_analysis) is carried out using automated means (software tools), and it is intended to identify potentially hazardous code fragments, including:

- calling functions, methods, procedures (hereafter, functions), that get as arguments the data entered by the user or input from external sources;
- function text for data typecasting;
- system calls and calls for IS functions for shared ABS components, including IS functions for the operating system and dedicated technical safeguards, I/O functions, and control over memory and system resources;
- function texts for generating verification of access rights and the acceptance of decisions based on security attribute values;
- function texts for independently implementing IS functionality, including cryptographic functions, user authentication and access right verification, and the generation of IS monitoring data;
- function texts for the connection with external components to which authentication data is transmitted;
- texts for error and exception handlers.

3.2. During static code analysis, it is advisable that searches be conducted for typical programming errors (insufficient checks of input parameters for functions, inclusion of authentication data immediately into the program text, incorrect typecasting, insufficient handling of errors and exceptions), and also that static program paths should be defined.

4. Dynamic code analysis

4.1. Dynamic code analysis is performed through the execution or emulation of the execution of program performance using a certain combination of sets of test input data. Prior to, or during execution, the program is sometimes reviewed by adding execution tracing functions to assign and control invariants, pre-conditions, post-conditions, etc. A dynamic review is carried out using dedicated automated facilities, and it may comprise, in particular:

- a study of performance features of potentially dangerous functions where known incorrect arguments are assigned;
- the construction of dynamic program paths and identification of decision points that are essential to fulfil IS functions;
- a search for sensitive information in random-access memory and function arguments;

– a study of program execution features during typical attacks (buffer overflow, injection of SQL operators into the data used to make DBMS requests).

5. Additional practical aspects of code review

5.1. Regardless of the adopted code review methods and techniques, classifiers of typical programming errors should be used, as well as relevant techniques to identify various types of errors (e.g. Common Weakness Enumeration Catalogue [5]).

5.2. It is expedient to assign a degree of criticality to vulnerabilities identified as part of the code review of ABS components (e.g. high, medium, low) to ensure IS of the RF BS organisation. As to each identified vulnerability, depending on its criticality, a decision is taken on modifying the ABS software component (including the priority of any modification), or on accepting the risks associated with the presence of the vulnerability.

5.3. Code review results are specified in code review protocols (such documents may be named otherwise), which are signed by developers, i.e. the immediate performers of the verification of the source code and persons involved in the verification process (code controllers); the protocol specifies details of the event date, source code portions checked, vulnerabilities identified, and other defects (in any), and a repeated code review with confirmation of the elimination of vulnerabilities or defects that were detected.

Note: It is recommended that code review protocols and other similar documents be executed as data in an electronic format that has been created, transferred, and reliably stored in the Electronic Document Management System provided for the given type of information (or documents) (e.g. in the archive of e-mail messages), with details (name, unique number, signatures, dates, etc.) that permit access to them during auditing as an electronic document (or document package) signed by means of simple digital signatures and, as required, produced and certified in hard copy format. (The requirement for initial protocol execution on paper or using supported digital signatures may block the performance of systematic code reviews.)

5.4. Code review arrangements are planned and executed in respect of the entire source code (modified or newly created) that is subject to review, with notification and, where necessary, involvement of IS service representatives as code controllers.

Appendix 3

Recommendations for Security Assessment

Security assessment is the study of ABS and ABS components with the aim of identifying vulnerabilities that may be used by an intruder to implement IS threats. The following main methods of security assessment are specified:

- penetration testing;
- the identification of known software vulnerabilities.

1. Penetration testing

1.1. Method description

1.1.1. Penetration testing is the main method of security assessment and covers all aspects of the ABS IS subsystem operations, including actions taken by personnel to respond to IS incidents, and cyber-attack counteraction.

1.1.2. Advantages of penetration testing as a method of security assessment include:

- the high reliability of information about identified vulnerabilities due to the actual confirmation of their possible use by the intruder;
- sufficiency of study results to assess the criticality of the identified vulnerabilities;
- the visibility of the obtained results.

Disadvantages of penetration testing are as follows:

- the investigator's ability to simulate only those actions taken by an intruder that is equal to or inferior in terms of expertise and, as a consequence, the essential requirements for the investigator's expertise are manifest, while the information on the absence of vulnerabilities is not highly reliable;
- the low degree of automation in the activities of the investigator and, as a consequence, high labour costs against other security assessment techniques.

1.1.3. When penetration testing, the investigator searches for ABS vulnerabilities simulating the intruder's actions. Prior to work commencement, the investigator is provided with conditions equivalent to those of the potential intruder. Conditions for penetration testing vary in the following parameters:

- the access rights to the ABS available to the investigator;
- the investigator's location as relates to the ABS network security perimeter;
- the strategy for providing the investigator with ABS information.

1.1.4. Penetration testing is divided into investigations with and without the provision of access to the ABS. When access is provided for an investigation, the investigator is given accounts to access the ABS. When access to the ABS is not provided for an investigation, the independent generation of accounts for access to the ABS is part of the process for penetration testing.

1.1.5. As for the location of the investigator in relation to the ABS network perimeter, penetration testing may be external or internal. In the case of internal penetration testing, the investigator can connect to the telecommunication equipment at a point that is within the network security perimeter of the RF BS organisation, thus enabling a network interface with ABS components. In the case of external penetration testing, the initial actions taken by the investigator are limited only by the network protocols interfacing with the ABS that are available from outside of the network security perimeter of the RF BS entity. If the ABS lacks interfaces for interaction from outside of the network perimeter, independently bypassing the network security perimeter of the RF BS organisation is part of penetration testing.

1.1.6. When conducting penetration testing, two strategies may be used to deliver ABS information to the investigator Where the black box strategy is applied, the investigator uses only the ABS information independently obtained by the investigator during penetration testing. If the white box strategy is applied, the investigator is given in advance any available ABS information including design and operational documentation, input codes of ABS software components, and the opportunity to review ABS component settings.

1.1.7. Penetration testing should be conducted only after notifying the operating personnel of the RF BS organisation, unless they can take actions to hinder the investigator.

1.1.8. Prior to penetration testing, the initial conditions for such testing should be determined. You should consider that the maximum degree of assessment is achieved with internal penetration testing that include the provision of access to the ABS and the white box strategy. Penetration testing under such initial conditions should be performed in the acceptance and commissioning phase and after each ABS upgrade.

1.1.9. Any action that may result in damage to the RF BS organisation should be carried out only after approval has been granted by the management of the RF BS organisation.

1.1.10. During penetration testing, the investigator may be granted access to information protected in accordance with the legislation of the Russian Federation and regulations of the RF BS organisation. Employment agreements entered into with the staff of the RF BS organisation, and service contracts concluded with penetration testing organisations shall comprise:

- a requirement for maintaining the confidentiality of information that could potentially be accessed during penetration testing, in accordance with the legislation of the Russian Federation including Bank of Russia regulations and documents of the RF BS organisation;

- the distribution and establishment of liability for cases when actions performed within the framework of penetration testing lead to negative consequences and damage to the RF BS organisation.

1.1.11. The penetration testing report shall contain:

- a description of the initial conditions for the investigation and problem definition;
- a description of the sequence of actions that resulted in the identification of vulnerabilities or changes in the resources of the investigator, and decisions to refuse to perform the actions requested;
- a description of the identified vulnerabilities with an assessment of the degree of their criticality for ensuring IS for the RF BS organisation;
- recommendations for the elimination of identified vulnerabilities.

1.2. The scope of penetration testing work

1.2.1. Penetration testing comprises the following investigative areas:

- assessment of the security of the network perimeter, network devices and protocols;
- assessment of the security of wireless networks;
- assessment of the security of web applications;
- assessment of the security of operating systems;
- assessment of the security of Database Management Systems (DBMS);
- assessment of the security of virtualization means;
- assessment of the security of specialised bank applications;
- assessment of the security of mobile devices.

1.2.2. The following arrangements are advised when the security of the network perimeter, network devices and protocols is assessed:

- the identification of network devices and interface protocols available to the investigator;
- the identification of types of devices, as well as series and versions of software used to implement network protocols, based on the information provided and the specific nature of their response to interaction;
- a search for remote access interfaces and other interfaces that should not be available from the given point under the IS requirements of the RF BS organisation or the practice implemented by the secured automated system;
- verification of whether it is possible to re-direct network traffic using features in the protocols for the channel and network levels, the protocols for the automatic mutual coordination of the parameters for the telecommunication equipment, the creation of false network services for automatic routing, name resolution, and the creation of false network services;
- the selection of authentication data (user names, passwords, keys) to gain access to network services based on directories with standard and frequent values;
- interception and repeated sending of authentication data;
- verification of whether it is possible to bypass network perimeter safeguards by changing essential fields in network packages, tunnelling and encryption of data, overloading event logs of information security systems with meaningless data;
- the identification of available web interfaces and irregular interface protocols for subsequent analysis.

1.2.3. The following arrangements are advisable during the security assessment of wireless networks:

- traffic monitoring, including the detection of available wireless network devices and their identifiers; determination of the current radio coverage zone; accumulation of information about client facilities (e.g. MAC addresses); collection of available network identifiers; determination of applicable encryption algorithms;
- dissemination of sample requests (e.g. to search the identification data of facilities) and analysis of responses to sample requests;
- the identification of defects in the settings for built-in cryptographic protection facilities for wireless connections;
- the imposition of false access points or duplicate access points for client facilities.

1.2.4. The following arrangements are advisable when the security of web applications is assessed:

- identification of vulnerabilities associated with the disclosure of sensitive information about the application including through sending incorrect messages; analysis of standard system error messages; search for sensitive information both in the code and in web page comments;
- receipt of information about the file system structure by searching file paths and names (a comprehensive search, directory search, check of availability of standard files belonging to platforms and development tools, searches for backup files);
- verifications of the consistency with which special characters in request parameters are processed (characters for output formatting, line feeds and carriage returns, switches to the parent directory, double URL encoding);
- verification of whether parameters of various lengths are processed consistently;
- validating the processing of numerical parameters, including those for technologies that are not equipped to process large variable, or negative and zero values;
- verification of whether parameter types are reduced and cast correctly;
- verification of whether user data presented in various ways is processed correctly (including duplication of request headers and scenario parameters);
- verification of whether parameters for the uniform resource identifier (URI) are processed correctly, including whether it is possible to connect an arbitrary external data source or redirect data to an external or internal website, and also the possibility of reference to network protocols or replacement of the full path to a resource with a relative one;
- verification of whether or not there are errors associated with loaded file processing, including the handling of file names without an extension, extension disagreement with the file type, alternative extensions for files of the same type, special characters (including the null character) in the file name;

- verification of whether scenarios are executed correctly during the manipulation of input parameters, including security attributes used for access control;
- verification of whether authentication data (passwords, including dictionary passwords, session identifiers, attributes used for password recovery) can be selected;
- verification of whether user session identifiers are processed correctly, including the handling of session closing events, inactivity intervals, and the comparison of session identifiers with additional attributes that identify the user or its workstation directly or indirectly, as well as the prevention of repeated or numerous applications involving session identifiers;
- verification of whether authorisation arrangements are executed correctly;
- verification of whether attacks against client applications are responded to correctly, including responses using cross-site scripting and cross-site request forgery;
- verification of whether input parameter scenarios are handled correctly where commands of operating systems, syntactic structures for programming languages and formatting are incorporated;
- validating the inability to bypass application-level firewalls using data fragmentation, parameter mixing, changes to the coding algorithm and data format, and the replacement of special characters with alternatives.

1.2.5. The following arrangements are advisable when the security of operating systems is assessed:

- the identification of the network services of the operating system by typical attributes (standard parameters for network protocols, typical response to connections, unique features in the implementation of network protocols), the availability of typical service messages in the network traffic;
- verification of whether access to the network services of the operating system is properly restricted, including when anonymous/guest access is used, during password selection, through interception and repeated/numerous uses of authorisation markers;
- verification of whether measures to counteract attempts to guess passwords, including an estimation of an intruder's ability to lock user accounts through numerous unsuccessful authentication attempts;
- verification of whether the intruder could obtain sensitive information using service network protocols (SNMP, RPC, CIFS);
- verification of whether cyber-attacks could be implemented using vulnerabilities in network services and in software applications at workstations.

Where the investigator has access to operating system management interfaces, security assessment also comprises:

- the possible loading of the operating system in a special mode or from a removable carrier (in case of physical access to the computer equipment), and the loading of the operating system in a special mode (e.g. in the recovery mode);
- the procurement of user names;
- review of event log data and residual information (deleted files, random-access memory images stored during failures);
- search for authentication user data in residual information, configuration parameters for software, the source code of applications and scripts;
- verification of whether operating system management may be transferred to a remote computer through a reverse connection and tunnelling protocols;
- verification of whether privileges may be advanced using locally operated vulnerabilities and errors in software settings;
- redirection and monitoring of network traffic in domain conflicts in the computer equipment analysed through forgery of ARP protocol tables, false servers in dynamic configurations for equipment and name resolutions.

1.2.6. The following measures are advised when DBMS security is assessed:

- verification of whether mechanisms for identification, authentication and access control are functioning correctly when interacting with DBMS control interfaces, including anonymous and guest access blocks, the absence of standard accounts or accounts with dictionary passwords;
- verification of whether modified incoming requests are processed correctly, including changes to protocol parameters, the insertion of special characters and commands for the operating system in parameters for input queries in SQL language;
- the exploitation of vulnerabilities in DBMS network services.

When there is access to interfaces to the operating system and DBMS management, the following activities are also called for:

- checking consistency of rights of access to DBMS files;
- verification of controls over the integrity of executable DBMS files, including protection against overriding files;
- a search for sensitive information (including user passwords) in service files (files of databases, logs, backup copies, configuration, command history), as well as in the variable system processes of DBMS;
- exploitation of vulnerabilities in stored procedures directed at improving privileges, the performance of ad-hoc commands or direct access to table contents, and also system setting changes;
- verification of whether it is possible to use stored procedures to access protected areas in the DBMS and operating system;
- verification of whether rogue parameter values for stored procedures are handled correctly (transmission of arbitrary values of parameters, implementation of SQL operators and PL/SQL and T-SQL commands, cursor application; transmission of parameter values of various lengths);
- verification of whether sensitive application information can be read, including the recovery of DBMS user passwords and applications from hash values.

1.2.7. Where the security of virtualization methods is assessed, verification should be made of whether virtualization environment management interfaces and protected objects can be accessed, including:

- whether it is possible to select passwords for management interfaces;
- whether user access rights to virtualization objects are correct, including the potential for unauthorised readings or changes to virtual disks, random-access memory images, configuration files, and virtual machine shots;
- use of vulnerabilities of the hypervisor and virtualization environment controls.

1.2.8. The following measures are advised when the security of specialised bank applications is assessed:

- network traffic monitoring, and searches for sensitive information in such traffic, including passwords and hash values in user passwords, session identifiers, authorisation markers, and cryptographic keys;
- the launch of programs with various parameters (including irregular ones), also using values of various lengths, the duplication of certain parameters that are assigned different values, the inclusion of special characters, operating system commands, and operators of interpreted programming languages in parameter values;
- the monitoring of the interface pattern between the application and the operating system during operations, including identification of data files containing sensitive information, and tracing system call;
- validation of the access rights to data files containing sensitive information, and monitoring of the integrity of application files.

1.2.9. The following measures are advised when the security of mobile facilities is assessed:

- verification of sensitive information in data files, event logs, the random-access memory of the device, and transmission of clear sensitive information;
- verification of whether it is possible to read encryption keys and digital signatures, as well as records and replacements of key certificates;
- identification of interface protocols, and verification of whether it is possible to force the use of insecure protocol versions by the device (HTTP instead of HTTPS, TELNET instead of SSH, SSH1 instead of SSH2);
- verification of whether the mobile application enables the correct handling of input parameters, including use of values of various lengths, the duplication of certain parameters that are assigned different values, inclusion of special characters, operating system commands, and the operators of interpreted programming languages in parameter values.

2. Identification of known vulnerabilities

2.1. Identification of known vulnerabilities comprises:

- identification of known vulnerabilities in network services;
- identification of typical vulnerabilities in web applications;
- identification of known vulnerabilities in software;
- identification of accounts with passwords contained in directories used during investigation.

This security assessment method may be part of the penetration testing method. It does not require an investigator with special qualifications, and it allows complete automation.

2.2. Known vulnerabilities may be identified through:

- identification of names and versions of ABS software, and searches for vulnerabilities expected in databases;
- the launch of test programs (exploits) fully or partially reproducing computer attacks using known vulnerabilities.

2.3. Depending on initial conditions for identifying known vulnerabilities, either the black box strategy or the white box strategy may be applied.

When the black box strategy is used, the investigator is provided with access to ABS components at the IP protocol level. Vulnerabilities in the network services of ABS components discovered by the investigator be the target of the investigation.

When the white box strategy is used, the investigator is provided with access to the interfaces of management of operating systems, telecommunication equipment, DBMS and application servers along with the necessary access rights. All vulnerabilities in ABS software components with information contained in the vulnerability database utilized by the investigator are the target of the investigation.

When the white box strategy is applied, investigations may be carried out both using automated means of security analysis and manually; if the black box strategy is used, investigations are performed utilizing automated means.

2.4. The advantages in the identification of known vulnerabilities as a security assessment method include:

- the high reliability of information about the vulnerabilities identified when the white box strategy is used;
- the high degree of automation, low requirements for investigator expertise where automated means of security analysis are used;
- the applicability of investigation results to estimate whether it is possible to implement threats, including the severity of the consequences of such implementation;
- the investigation is reproducible.

The disadvantages in the identification of known vulnerabilities as a security assessment method include:

- the low reliability of information about vulnerabilities identified when the black box strategy is used;
- the possible failure or faults in the operation of ABS components when the investigation is carried out using the black box strategy;
- the need to provide the investigator with privileged access to ABS components if the white box strategy is used.

2.5. Known vulnerabilities in network services are identified using the black box strategy. The investigation comprises:

- the identification of servers and workstations by their IP addresses or names;
- the identification of network protocols available for interface;
- the identification of programs providing implementation of the specified network protocols (their names and versions to be determined by information transmitted during the network interface);
- a selection of database vulnerabilities related to the identified program versions;
- the identification of vulnerabilities in network services through launching exploits that are potentially applicable.

2.6. Typical vulnerabilities in web applications are identified using the black box strategy. The investigation comprises the identification of the following types of vulnerabilities:

- injections, particularly SQL injections, OS Command, LDAP and XPath injections;
- the selection of authentication data;
- insecure data transfer, including during authentication;
- errors in access control (e.g. insecure direct references to facilities, the inability to restrict access by URL and directory bypasses);
- Cross Site Scripting (XSS);
- Cross Site Request Forgery (CSRF);
- HTTP request splitting, non-disclosure of HTTP response;
- open redirection;
- disclosure of information about directories/scenarios;
- predictable resource layouts;
- application identification;
- random file reading;
- disclosure of sensitive information;
- Back path in directories;
- buffer overflow.

The investigation is carried out through web form data analysis, sending test requests with variable request parameter values for the web server, and response analysis.

2.7. Known software vulnerabilities are identified using the white box strategy. The investigation comprises:

- an inventory of software installed in the processing equipment being investigated in which the program names and versions and provided security updates are identified;
- a selection of database vulnerabilities related to the identified program versions;
- the exclusion of vulnerabilities eliminated by security updates from the resulting selection.

2.8. Accounts with passwords contained in directories used during the investigation which are identified using both black box strategy and white box strategy When the black box strategy is used, authentication is attempted using names and passwords from the directory. If the white box strategy is used, hash values in passwords from configuration files and database tables are selected, and they are compared with hash values in passwords from the directory.

2.9. Based on the investigation, a report is compiled, which shall contain:

- the list of ABS components;
- the list of identified vulnerabilities, an assessment of the degree of their criticality for ensuring IS at the RF BS organisation;
- recommendations for the elimination of identified vulnerabilities.

Vulnerability criticality should be assessed in accordance with the methodology of the Common Vulnerability Scoring System (CVSS).

Appendix 4

Recommendations for Monitoring the Settings for Technical Safeguards (Configuration Error Identification)

Recommendations for Monitoring the Settings for Technical Safeguards (Configuration Error Identification)

1. Configuration error identification is intended to maintain the consistent operation of the ABS IS sub-system. To this end, configuration standards for ABS software and hardware components are developed and approved within the scope of the operational documentation for ABS or as individual internal documents of the RF BS organisation. Configuration error identification is an investigation aimed at finding clashes between the actual values of the parameters for technical safeguards and their reference values as specified in configuration standards.

2. The investigator shall be provided with access to the interfaces of ABS software components, including the operating system, specialised bank applications, or an alternative way of obtaining actual settings for technical safeguards.

3. The investigation is performed using the white box strategy. The investigation may be carried out manually, or by using an automated method for security analysis. Configuration errors are identified by obtaining the actual values of settings and their comparison with reference values. In this connection:

– if the actual value of the setting is explicitly assigned, it is compared with the reference value;

– if the actual value of the configuration parameter is not assigned or when it is not explicitly set, the effective value of the setting is calculated and then is compared with the reference value.

In the event that the setting is not explicitly assigned, the apparatus or program uses a default value that may depend on the version of the apparatus or program. In a number of cases, actual settings are not explicitly assigned; rather, they are assigned in the form of expressions calculated on the basis of actual values for other parameters and environment variables. In such cases, the effective values of settings shall be determined during the investigation based on the special nature of their determination within the framework of the ABS component being analysed.

4. The advantages of this method include:

– the high reliability of information about the identified clashes with configuration standards;

– the high degree of automation, low requirements for special qualifications by the investigator when an automated method of analysis is used;

– the investigation is reproducible.

5. The disadvantages of this method include:

– the inability to assess, in a number of cases, the potential to implement threats when particular settings fail to meet configuration standards. Recommendations of software and hardware developers are used in the development of configuration standards. As a rule, developers do not disclose information about threats that may be implemented if such recommendations are not met;

– the need to provide the investigator with privileged access to ABS components;

– the high qualifications required in the investigator to calculate the effective values of configuration parameters, and when the investigation is carried out without any automated methods of analysis.

6. Based on the investigation, a report is compiled; such report shall contain the list of ABS components analysed, and the list of identified clashes with configuration standards, and recommendations for their elimination.

7. If a change in the settings has not been caused by technical requirements, the ABS components should be reset.

If a change in the settings has been caused by technical requirements, and a reference value reset may involve a disturbance in the operation of the ABS, criticality of the impact of the settings on ABS security should be assessed. Where there is no such impact or if the increase in the risk of IS breaches is insignificant, corresponding changes should be introduced in the operational documentation (or configuration standards). If there is a significant increase in the risk of IS breaches, the ABS or specific components should be upgraded.

References

1. National Checklist Program Repository [electronic resource]: official website.
   URL: http://web.nvd.nist.gov/view/ncp/repository (last accessed date:
2. CERT Notifications [electronic resource]: official website.
   URL: http://www.us-cert.gov/ncas/bulletins (last accessed date: 25.03.2014).
3. Alerts, Bulletins & Webinars [electronic resource]: official website.
   URL: http://usa.visa.com/merchants/risk_management/cisp_alerts.html#anchor_3 (last accessed date: 25.03.2014).
4. Critical Patch Updates, Security Alerts and Third Party Bulletin [electronic resource]: official website.
   URL: http://www.oracle.com/technetwork/topics/security/alerts-086861.html (last accessed date: 25.03.2014).
5. CCE List — Archive [electronic resource]: US official website.
   URL: http://cce.mitre.org/lists/cce_list.html (last accessed date: 25.03.2014).

*In case of any translation ambiguity the Russian version shall prevail.