UNIX/LINUX Operating System Security Audit/Assurance Program

ISACA®
Serving IT Governance Professionals

ISACA®
With more than 86,000 constituents in more than 160 countries, ISACA (*www.isaca.org*) is a recognized worldwide leader in IT governance, control, security and assurance. Founded in 1969, ISACA sponsors international conferences, publishes the *ISACA Journal*®, and develops international information systems auditing and control standards. It also administers the globally respected Certified Information Systems Auditor™ (CISA®) designation, earned by more than 60,000 professionals since 1978; the Certified Information Security Manager® (CISM®) designation, earned by more than 10,000 professionals since 2002; and the new Certified in the Governance of Enterprise IT™ (CGEIT™) designation.

Disclaimer
ISACA has designed and created *UNIX/LINUX Operating System Security Audit/Assurance Program* (the "Work"), primarily as an informational resource for audit and assurance professionals. ISACA makes no claim that use of any of the Work will assure a successful outcome. The Work should not be considered inclusive of all proper information, procedures and tests or exclusive of other information, procedures and tests that are reasonably directed to obtaining the same results. In determining the propriety of any specific information, procedure or test, audit/assurance professionals should apply their own professional judgment to the specific circumstances presented by the particular systems or IT environment.

Reservation of Rights

ISACA
3701 Algonquin Road, Suite 1010
Rolling Meadows, IL 60008 USA
Phone: +1.847.253.1545
Fax: +1.847.253.1443
E-mail: *info@isaca.org*
Web site: *www.isaca.org*

ISBN 978-1-60420-083-6
UNIX/LINUX Operating System Security Audit/Assurance Program
Printed in the United States of America

UNIX/LINUX Operating System Security Audit/Assurance Program

ISACA wishes to recognize:

Author
Norm Kelson, CISA, CGEIT, CPA, The Kelson Group, USA

Expert Reviewers
Claudio Cilli, PhD., CISA, CISM, CGEIT, University of Marche, Italy
Jimmy Heschl, CISA, CISM, CGEIT, KPMG, Austria
Shirish Ketkar, BDO Haribhakti Consulting Private Ltd., India
Sanjay Vaid, CISA, Fujitsu Siemens Computers, Belgium

UNIX/LINUX Operating System Security Audit/Assurance Program

Table of Contents

I. Introduction ... 4
II. Using This Document ... 5
III. Controls Maturity Analysis ... 8
IV. Assurance and Control Framework ... 9
V. Executive Summary of Audit/Assurance Focus ... 10
VI. Audit/Assurance Program ... 13
1. Planning and Scoping the Audit ... 13
2. Preparatory Steps ... 15
3. Access and Authorization ... 17
4. Network ... 28
5. Monitoring and Auditing the System ... 36
6. Operating System and Application Patches and Configuration Change Management ... 39
7. System Backup and Recovery ... 48
VII. Maturity Assessment ... 52
VIII. Assessment Maturity vs. Target Maturity UNIX/LINUX Only ... 55

I. Introduction

Overview

ISACA has developed the *IT Assurance Framework*^{TM} (ITAF^{TM}) as a comprehensive and good-practice-setting model. ITAF provides standards that are designed to be mandatory, and are the guiding principles under which the IT audit and assurance profession operates. The guidelines provide information and direction for the practice of IT audit and assurance. The tools and techniques provide methodologies, tools and templates to provide direction in the application of IT audit and assurance processes.

Purpose

The audit/assurance program is a tool and template to be used as a roadmap for the completion of a specific assurance process. The ISACA Assurance Committee has commissioned audit/assurance programs to be developed for use by IT audit and assurance practitioners. This audit/assurance program is intended to be utilized by IT audit and assurance professionals with the requisite knowledge of the subject matter under review, as described in ITAF, section 2200—General Standards. The audit/assurance programs are part of ITAF, section 4000—IT Assurance Tools and Techniques.

Control Framework

The audit/assurance programs have been developed in alignment with the IT Governance Institute® (ITGI^{TM}) framework *Control Objectives for Information and related Technology* (COBIT®)—specifically COBIT 4.1— using generally applicable and accepted good practices. They reflect ITAF, sections 3400—IT Management Processes, 3600-IT Audit and Assurance Processes, and 3800—IT Audit and Assurance Management.

Many organizations have embraced several frameworks at an enterprise level, including the Committee of Sponsoring Organizations of the Treadway Commission (COSO) Internal Control Framework. The importance of the control framework has been enhanced due to regulatory requirements by the US Securities and Exchange Commission (SEC) as directed by the US Sarbanes-Oxley Act of 2002 and similar legislation in other countries. They seek to integrate control framework elements used by the general audit/assurance team into the IT audit and assurance framework. Since COSO is widely used, it has been selected for inclusion in this audit/assurance program. The reviewer may delete or rename these columns to align with the enterprise's control framework.

IT Governance, Risk and Control

IT governance, risk and control are critical in the performance of any assurance management process. Governance of the process under review will be evaluated as part of the policies and management oversight controls. Risk plays an important role in evaluating what to audit and how management approaches and manages risk. Both issues will be evaluated as steps in the audit/assurance program. Controls are the primary evaluation point in the process. The audit/assurance program will identify the control objectives with steps to determine control design and effectiveness.

Responsibilities of IT Audit and Assurance Professionals

IT audit and assurance professionals are expected to customize this document to the environment in which they are performing an assurance process. This document is to be used as a review tool and starting point. It may be modified by the IT audit and assurance professional; it *is not* intended to be a checklist or questionnaire. It is assumed that the IT audit and assurance professional holds the Certified Information Systems Auditor (CISA) designation, or has the necessary subject matter expertise required to conduct the work and is supervised by a professional with the CISA designation and necessary subject matter expertise to adequately review the work performed.

II. Using This Document

This audit/assurance program was developed to assist the audit and assurance professional in designing and executing a review. Details regarding the format and use of the document follow.

Work Program Steps

The first column of the program describes the steps to be performed. The numbering scheme used provides built-in work paper numbering for ease of cross-reference to the specific work paper for that section. The physical document was designed in Microsoft® Word. The IT audit and assurance professional is encouraged to make modifications to this document to reflect the specific environment under review.

Step 1 is part of the fact gathering and pre-fieldwork preparation. Because the pre-fieldwork is essential to a successful and professional review, the steps have been itemized in this plan. The first-level steps, e.g., 1.1, are in **bold** type and provide the reviewer with a scope or high-level explanation of the purpose for the substeps.

Beginning in step 2, the steps associated with the work program are itemized. To simplify the use of the program, the audit/assurance program describes the audit/assurance objective—the reason for performing the steps in the topic area. The specific controls follow and are shown in **blue** type. Each review step is listed below the control. These steps may include assessing the control design by walking through a process, interviewing, observing or otherwise verifying the process and the controls that address that process. In many cases, once the control design has been verified, specific tests need to be performed to provide assurance that the process associated with the control is being followed. Test objectives are shown in **green** type. The specific test process follows the test objective.

The maturity assessment, which is described in more detail later in this document, makes up the last section of the program.

The audit/assurance plan wrap-up—those processes associated with the completion and review of work papers, preparation of issues and recommendations, report writing and report clearing—has been excluded from this document, since it is standard for the audit/assurance function and should be identified elsewhere in the enterprise's standards.

COBIT Cross-reference

The COBIT cross-reference provides the audit and assurance professional with the ability to refer to the specific COBIT control objective that supports the audit/assurance step. The COBIT control objective should be identified for each audit/assurance step in the section. Multiple cross-references are not uncommon. Processes at lower levels in the work program are too granular to be cross-referenced to COBIT. The audit/assurance program is organized in a manner to facilitate an evaluation through a structure parallel to the development process. COBIT provides in-depth control objectives and suggested control practices at each level. As the professional reviews each control, he/she should refer to COBIT 4.1 or the *IT Assurance Guide: Using COBIT* for good-practice control guidance.

COSO Components

As noted in the introduction, COSO and similar frameworks have become increasingly popular among audit and assurance professionals. This ties the assurance work to the enterprise's control framework. While the IT audit/assurance function has COBIT as a framework, operational audit and assurance professionals use the framework established by the enterprise. Since COSO is the most prevalent internal control framework, it has been included in this document and is a bridge to align IT audit/assurance with the rest of the audit/assurance function. Many audit/assurance organizations include the COSO control components within their report and summarize assurance activities to the audit committee of the board of directors.

For each control, the audit and assurance professional should indicate the COSO component(s) addressed. It is possible but generally not necessary to extend this analysis to the specific audit step level.

The original COSO internal control framework contained five components. In 2004, COSO was revised as the *Enterprise Risk Management (ERM) Intergrated Framework* and extended to eight components. The primary difference between the two frameworks is the additional focus on ERM and integration into the business decision model. ERM is in the process of being adopted by large enterprises. The two frameworks are compared in **figure 1**.

Figure 1—Comparison of COSO Internal Control and ERM Integrated Frameworks

Internal Control Framework	ERM Integrated Framework
Control Environment: The control environment sets the tone of an organization, influencing the control consciousness of its people. It is the foundation for all other components of internal control, providing discipline and structure. Control environment factors include the integrity, ethical values, management's operating style, delegation of authority systems, as well as the processes for managing and developing people in the organization.	**Internal Environment**: The internal environment encompasses the tone of an organization, and sets the basis for how risk is viewed and addressed by an enterprise's people, including risk management philosophy and risk appetite, integrity and ethical values, and the environment in which they operate.
	Objective Setting: Objectives must exist before management can identify potential events affecting their achievement. Enterprise risk management ensures that management has in place a process to set objectives and that the chosen objectives support and align with the enterprise's mission and are consistent with its risk appetite.
	Event Identification: Internal and external events affecting achievement of an enterprise's objectives must be identified, distinguishing between risks and opportunities. Opportunities are channeled back to management's strategy or objective-setting processes.

Figure 1—Comparison of COSO Internal Control and ERM Integrated Frameworks

Internal Control Framework	ERM Integrated Framework
Risk Assessment: Every enterprise faces a variety of risks from external and internal sources that must be assessed. A precondition to risk assessment is establishment of objectives, and thus risk assessment is the identification and analysis of relevant risks to achievement of assigned objectives. Risk assessment is a prerequisite for determining how the risks should be managed.	**Risk Assessment**: Risks are analyzed, considering the likelihood and impact, as a basis for determining how they could be managed. Risk areas are assessed on an inherent and residual basis.
	Risk Response: Management selects risk responses – avoiding, accepting, reducing, or sharing risk – developing a set of actions to align risks with the enterprise's risk tolerances and risk appetite.
Control Activities: Control activities are the policies and procedures that help ensure management directives are carried out. They help ensure that necessary actions are taken to address risks to achievement of the enterprise's objectives. Control activities occur throughout the organization, at all levels and in all functions. They include a range of activities as diverse as approvals, authorizations, verifications, reconciliations, reviews of operating performance, security of assets and segregation of duties.	**Control Activities:** Policies and procedures are established and implemented to help ensure the risk responses are effectively carried out.
Information and Communication: Information systems play a key role in internal control systems as they produce reports, including operational, financial and compliance-related information that make it possible to run and control the business. In a broader sense, effective communication must ensure information flows down, across and up the organization. Effective communication should also be ensured with external parties, such as customers, suppliers, regulators and shareholders.	**Information and Communication:** Relevant information is identified, captured, and communicated in a form and timeframe that enable people to carry out their responsibilities. Effective communication also occurs in a broader sense, flowing down, across, and up the enterprise.
Monitoring: Internal control systems need to be monitored—a process that assesses the quality of the system's performance over time. This is accomplished through ongoing monitoring activities or separate evaluations. Internal control deficiencies detected through these monitoring activities should be reported upstream and corrective actions should be taken to ensure continuous improvement of the system.	**Monitoring:** The entirety of enterprise risk management is monitored and modifications made as necessary. Monitoring is accomplished through ongoing management activities, separate evaluations, or both.

Information for **figure 1** was obtained from the COSO web site *www.coso.org/aboutus.htm.*

The original COSO internal control framework addresses the needs of the IT audit and assurance professional: control environment, risk assessment, control activities, information and communication, and monitoring. As such, ISACA has elected to utilize the five-component model for these audit/assurance programs. As more enterprises implement the ERM model, the additional three columns can be added, if relevant. When completing the COSO component columns, consider the definitions of the components as described in **figure 1**.

Reference/Hyperlink

Good practices require the audit and assurance professional to create a work paper for each line item, which describes the work performed, issues identified, and conclusions. The reference/hyperlink is to be used to cross-reference the audit/assurance step to the work paper that supports it. The numbering system of this document provides a ready numbering scheme for the work papers. If desired, a link to the work paper can be pasted into this column.

Issue Cross-reference

This column can be used to flag a finding/issue that the IT audit and assurance professional wants to further investigate or establish as a potential finding. The potential findings should be documented in a work paper that indicates the disposition of the findings (formally reported, reported as a memo or verbal finding, or waived).

Comments

The comments column can be used to indicate the waiving of a step or other notations. It is not to be used in place of a work paper describing the work performed.

III. Controls Maturity Analysis

One of the consistent requests of stakeholders who have undergone IT audit/assurance reviews is a desire to understand how their performance compares to good practices. Audit and assurance professionals must provide an objective basis for the review conclusions. Maturity modeling for management and control over IT processes is based on a method of evaluating the organization, so it can be rated from a maturity level of nonexistent (0) to optimized (5). This approach is derived from the maturity model that the Software Engineering Institute (SEI) of Carnegie Mellon University defined for the maturity of software development.

The *IT Assurance Guide Using COBIT*, Appendix VII—Maturity Model for Internal Control, in **figure 2**, provides a generic maturity model showing the status of the internal control environment and the establishment of internal controls in an enterprise. It shows how the management of internal control, and an awareness of the need to establish better internal controls, typically develops from an *ad hoc* to an optimized level. The model provides a high-level guide to help COBIT users appreciate what is required for effective internal controls in IT and to help position their enterprise on the maturity scale.

Figure 2—Maturity Model for Internal Control

Maturity Level	Status of the Internal Control Environment	Establishment of Internal Controls
0 Non-existent	There is no recognition of the need for internal control. Control is not part of the organization's culture or mission. There is a high risk of control deficiencies and incidents.	There is no intent to assess the need for internal control. Incidents are dealt with as they arise.
1 Initial/*ad hoc*	There is some recognition of the need for internal control. The approach to risk and control requirements is *ad hoc* and disorganized, without communication or monitoring. Deficiencies are not identified. Employees are not aware of their responsibilities.	There is no awareness of the need for assessment of what is needed in terms of IT controls. When performed, it is only on an *ad hoc* basis, at a high level and in reaction to significant incidents. Assessment addresses only the actual incident.
2 Repeatable but Intuitive	Controls are in place but are not documented. Their operation is dependent on the knowledge and motivation of individuals. Effectiveness is not adequately evaluated. Many control weaknesses exist and are not adequately addressed; the impact can be severe. Management actions to resolve control issues are not prioritized or consistent. Employees may not be aware of their responsibilities.	Assessment of control needs occurs only when needed for selected IT processes to determine the current level of control maturity, the target level that should be reached and the gaps that exist. An informal workshop approach, involving IT managers and the team involved in the process, is used to define an adequate approach to controls for the process and to motivate an agreed-upon action plan.
3 Defined	Controls are in place and adequately documented. Operating effectiveness is evaluated on a periodic basis and there is an average number of issues. However, the evaluation process is not documented. While management is able to deal predictably with most control issues, some control weaknesses persist and impacts could still be severe. Employees are aware of their responsibilities for control.	Critical IT processes are identified based on value and risk drivers. A detailed analysis is performed to identify control requirements and the root cause of gaps and to develop improvement opportunities. In addition to facilitated workshops, tools are used and interviews are performed to support the analysis and ensure that an IT process owner owns and drives the assessment and improvement process.
4 Managed and Measurable	There is an effective internal control and risk management environment. A formal, documented evaluation of controls occurs frequently. Many controls are automated and regularly reviewed. Management is likely to detect most control issues, but not all issues are routinely identified. There is consistent follow-up to address identified control weaknesses. A limited, tactical use of technology is applied to automate controls.	IT process criticality is regularly defined with full support and agreement from the relevant business process owners. Assessment of control requirements is based on policy and the actual maturity of these processes, following a thorough and measured analysis involving key stakeholders. Accountability for these assessments is clear and enforced. Improvement strategies are supported by business cases. Performance in achieving the desired outcomes is consistently monitored. External control reviews are organized occasionally.
5 Optimized	An enterprisewide risk and control program provides continuous and effective control and risk issues resolution. Internal control and risk management are integrated with enterprise practices, supported with automated real-time monitoring with full accountability for control monitoring, risk management and compliance enforcement. Control evaluation is continuous, based on self-assessments and gap and root cause analyses. Employees are proactively involved in control improvements.	Business changes consider the criticality of IT processes and cover any need to reassess process control capability. IT process owners regularly perform self-assessments to confirm that controls are at the right level of maturity to meet business needs and they consider maturity attributes to find ways to make controls more efficient and effective. The organization benchmarks to external best practices and seeks external advice on internal control effectiveness. For critical processes, independent reviews take place to provide assurance that the controls are at the desired level of maturity and working as planned.

The maturity model evaluation is one of the final steps in the evaluation process. The IT audit and assurance professional can address the key controls within the scope of the work program and formulate an objective assessment of the maturity level of the control practices. The maturity assessment can be a part of the audit/assurance report and can be used as a metric from year to year to document progression in the enhancement of controls. However, it must be noted that the perception as to the maturity level may vary between the process/IT asset owner and the auditor. Therefore, an auditor should obtain the concerned stakeholder's concurrence before submitting the final report to the management.

At the conclusion of the review, once all findings and recommendations are completed, the professional assesses the current state of the COBIT control framework and assigns it a maturity level using the six-level scale. Some practitioners utilize decimals (x.25, x.5, x.75) to indicate gradations in the maturity model. As a further reference, COBIT provides a definition of the maturity designations by control objective. While this approach is not mandatory, the process is provided as a separate section at the end of the audit/assurance program for those enterprises that wish to implement it. It is suggested that a maturity assessment be made at the COBIT control level. To provide further value to the client/customer, the professional can also obtain maturity targets from the client/customer. The sections can be summarized using the graphic presentation (section VIII) describing the achievement or gaps between the actual and targeted maturity levels. If this is used, it should be noted that this assessment addresses UNIX/LINUX only, as there are generally other operating systems in the enterprise.

IV. Assurance and Control Framework

ISACA IT Assurance Framework and Standards

ITAF section 3630.14—Operating Systems (OSs) Management and Controls—is relevant to UNIX/LINUX security.

ISACA has long recognized the specialized nature of IT assurance and strives to advance globally applicable standards. Guidelines and procedures provide detailed guidance on how to follow those standards. IS Auditing Standard S15 IT Controls, and IS Auditing Guideline G38 Access Controls are relevant to this audit/assurance program.

ISACA Controls Framework

COBIT is an IT governance framework and supporting tool set that allows managers to bridge the gap among control requirements, technical issues and business risks. COBIT enables clear policy development and good practice for IT control throughout enterprises.

Utilizing COBIT as the control framework on which IT audit/assurance activities are based aligns IT audit/assurance with good practices as developed by the enterprise.

The COBIT IT process DS9 *Manage the configuration* from the Deliver and Support (DS) domain, addresses good practices for ensuring the integrity of hardware and software configurations. This requires the establishment and maintenance of an accurate and complete configuration repository. DS5.3 *Identity management* and DS5.4 *User account management* address user identity, and the IT process AI6 *Manage changes* from the Acquire and Implement (AI) domain specifically addresses change management. Relevant COBIT control objectives are:

- AI6.1 *Change standards and procedures*—Set up formal change management procedures to handle in a standardized manner all requests (including maintenance and patches) for changes to applications, procedures, processes, system and service parameters, and the underlying platforms.
- AI6.2 *Impact assessment, prioritization and authorization*—Assess all requests for change in a structured way to determine the impact on the operational system and its functionality. Ensure that changes are categorized, prioritized, and authorized.

- AI6.4 *Change status tracking and reporting*—Establish a tracking and reporting system to document rejected changes, communicate the status of approved and in-process changes, and complete changes. Make certain that approved changes are implemented as planned.
- DS5.3 *Identity management*[1]—Ensure that all users (internal, external and temporary) and their activity on IT systems (business application, IT environment, system operations, development and maintenance) are uniquely identifiable. Enable user identities via authentication mechanisms. Confirm that user access rights to systems and data are in line with defined and documented business needs and that job requirements are attached to user identities. Ensure that user access rights are requested by user management, approved by system owners and implemented by the security-responsible person. Maintain user identities and access rights in a central repository. Deploy cost-effective technical and procedural measures, and keep them current to establish user identification, implement authentication and enforce access rights.
- DS5.4 *User account management*[2]—Address requesting, establishing, issuing, suspending, modifying and closing user accounts and related user privileges with a set of user account management procedures. Include an approval procedure outlining the data or system owner granting the access privileges. These procedures should apply for all users, including administrators (privileged users) and internal and external users, for normal and emergency cases. Rights and obligations relative to access to enterprise systems and information should be contractually arranged for all types of users. Perform regular management review of all accounts and related privileges.
- DS9.1 *Configuration repository and baseline*—Establish a supporting tool and a central repository to contain all relevant information on configuration items. Monitor and record all assets and changes to assets. Maintain a baseline of configuration items for every system and service as a checkpoint to which to return after changes.
- DS9.2 *Identification and maintenance of configuration items*—Establish configuration procedures to support management and logging of all changes to the configuration repository. Integrate these procedures with change management, incident management and problem management procedures.
- DS9.3 *Configuration integrity review*—Periodically review the configuration data to verify and confirm the integrity of the current and historical configuration. Periodically review installed software against the policy for software usage to identify personal or unlicensed software or any software instances in excess of current license agreements. Report, act on and correct errors and deviations.

Refer to the IT Governance Institute's *COBIT Control Practices: Guidance to Achieve Control Objectives for Successful IT Governance, 2nd Edition*, published in 2007, for the related control practice value and risk drivers.

V. Executive Summary of Audit/Assurance Focus

UNIX/LINUX Operating System Security

The review of the operating system ensures management that the computer platform that supports the various applications is secure. If the foundation (operating systems) is not secure, the applications can be compromised (see risk in the next section).

UNIX is one of the longest enduring operating systems in existence. Developed in 1969 by Bell Labs of AT&T, the operating system has evolved as the equipment on which UNIX runs has matured. Initially developed for internal AT&T and academic use, the commercial hardware vendors developed customized versions for their specific hardware. Its selling point has been portability among hardware. Previously, hardware vendors developed proprietary operating systems that only operated on their brand of

[1] Scope limitation—Identity management as it relates to superusers having access to the operating system
[2] Scope limitation—User account management as it relates to users accessing system functions

equipment. UNIX was the first operating system that would operate on multiple hardware platforms. The vendors wanted to keep their maintenance fees and so developed proprietary versions, which added administrative, security, performance, vendor hardware/software specific functionality. The major UNIX implementations are Sun Microsystems' Solaris, IBM's AIX, and Hewlett-Packard's HP-UX. The primary advantage of purchasing the operating system from the vendor is the support. However, the negatives are the initial purchase and annual maintenance fee to the vendor for its specific UNIX version, the reliance placed on the vendor for support, the inability to customize the vendor's version and the knowledge that the software is proprietary.

LINUX is a recent derivative of UNIX. Its primary difference is that it is open source, meaning that the computing community has the opportunity to add to the LINUX functionality. Linus Torvalds, the originator of LINUX, maintains direct development of the kernel (essential processing control, networking, and peripheral and file system access). The Free Software Foundation supports the GNU's Not Linux (GNU) components, ranging from graphic user interface to user applications libraries and utilities. The LINUX source code is free. Distributors have obtained the free source and added functionality to ease implementation (installation and configuration packages to eliminate the need to compile the source code as well as enhance system security.

Recognizing the interest in the more cost-effective LINUX, vendors have distributed secure, tested LINUX implementations. The major hardware vendors (Dell, IBM, HP, and Sun Microsystems) and two large software vendors (Red Hat and Novell) provide LINUX implementations that are subject to rigorous software change management and control. The hardware and software companies (with the addition of Oracle) provide fee-based support. The cost component of LINUX has increased its popularity over UNIX.

In the enterprise, UNIX and LINUX are the underlying computing platform for application servers that execute essential business applications (both centralized and distributed), database servers that manage the massive database used to store business data, web servers that provide the public face of the business on the Internet, and process transactions. Recognizing the development strategies of both UNIX and LINUX, it is essential that the source of the UNIX/LINUX operating system distribution be known, and care must be taken to ensure only authorized and tested functions, processes and configurations are allowed to enter the production environment.

Business Impact and Risk

UNIX/LINUX is widely used in the enterprise operating environment. The failure for UNIX/LINUX to be properly configured could result in the inability for the business to execute its critical processes. The collective development of LINUX adds additional risk. Unless the distribution is controlled and managed, dangerous processes could be introduced into the operating system as a result of the open nature of the operating system and its essential processes.

UNIX/LINUX risks resulting from ineffective or incorrect operating system configurations could permit the operating system to become compromised resulting in:

- Disclosure of privileged information
- Loss of physical assets
- Loss of intellectual property
- Loss of competitive advantage
- Loss of customer confidence
- Violation of regulatory requirements
- Disruption of the computer infrastructure resulting in the inability to perform critical business functions
- Infection of computer systems with viruses and the like to disrupt processing and require costly

disinfection
- Use of the computer systems as a launching pad for malicious activity against other entities (and the potential to be held liable for their damages)

Objective and Scope

Objective—The objective of the UNIX/LINUX audit/assurance review is to provide management with an independent assessment relating to the effectiveness of configuration and security of the UNIX/LINUX operations systems with the enterprise's computing environment.

Scope—The review will focus on configuration of the relevant UNIX/LINUX servers within the organization. The selection of the applications/functions and specific servers will be based upon the risks introduced to the organization by these systems.

UNIX/LINUX systems are subject to identity management, the process of identifying and authenticating users and superusers. Since this process is also addressed in the *Identity Management Audit/Assurance Program*, this review is limited to superuser access (access to the operating system's configuration and security mechanisms) and general user controls (excluding users from access to operating system resources). Refer to the *Identity Management Audit/Assurance Program* for controls relating to user identity.

{The remainder of this paragraph needs to be customized to describe which servers and applications within the enterprise will be reviewed.}

Minimum Audit Skills

This review is considered highly technical. The IT audit and assurance professional must have an understanding of the good practice UNIX and LINUX processes and requirements, and be highly conversant in UNIX/LINUX tools, exposures, and functionality. It should not be assumed that an audit and assurance professional holding the CISA designation has the requisite skills to perform this review.

VI. Audit/Assurance Program

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
1. PLANNING AND SCOPING THE AUDIT									
1.1 Define audit/assurance objectives. The audit/assurance objectives are high level and describe the overall audit goals.	ME2.1								
1.1.1 Review the audit/assurance objectives in the introduction to this audit/assurance program.									
1.1.2 Modify the audit/assurance objectives to align with the audit/assurance universe, annual plan and charter.									
1.2 Define boundaries of review. The review must have a defined scope. The reviewer must understand the operating environment and prepare a proposed scope, subject to a later risk assessment.	ME2.1								
1.2.1 Obtain and review the UNIX/LINUX operating system security and management policies.									
1.2.2 Obtain a list of UNIX/LINUX servers, their locations, the applications they process or support, the UNIX/LINUX distribution (vendor), and the version number.									
1.2.3 Establish initial boundaries of the audit/assurance review.									
1.2.4 Identify limitations and/or constraints limiting the audit of specific systems.									
1.3 Define assurance. The review requires two sources of standards. The corporate standards as defined in policy and procedure documentation establish the corporate expectations. At minimum, corporate standards should be implemented. The second source, a good practice reference, establishes industry standards. Gaps between the two should be proposed for enhancements.	ME2.1								
1.3.1 Obtain and review good practice UNIX/LINUX security and configuration standards.									
1.3.2 Obtain corporate UNIX/LINUX configuration standards.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
1.3.3 Determine if there are gaps in the corporate policy.									
1.4 Identify and document risks. The risk assessment is necessary to evaluate where audit resources should be focused. In most enterprises, audit resources are not available for all processes. The risk-based approach ensures utilization of audit resources in the most effective manner.	ME2.1								
1.4.1 Using the list of servers identified previously, determine the risk category for each server and establish a prioritized list of servers to be assessed.									
1.4.2 Review previous audits of UNIX/LINUX and other assessments.									
1.4.3 Determine if issues identified previously have been remediated.									
1.4.4 Evaluate the overall risk factor for performing the review.									
1.4.5 Based on the risk assessment, identify changes to the scope.									
1.4.6 Discuss the risks with IT, business, and operational audit management, and adjust the risk assessment.									
1.4.7 Based on the risk assessment, revise the scope.									
1.5 Define the change process. The initial audit approach is based upon the reviewer's understanding of the operating environment and associated risks. As further research and analysis is performed, changes to the scope and approach will result.	ME2.7								
1.5.1 Identify the senior IT assurance resource responsible for the review.									
1.5.2 Establish the process for suggesting and implementing changes to the audit/assurance program, and authorizations required.									
1.6 Define assignment success. The success factors need to be identified. Communication among the IT audit/assurance team, other assurance teams and the enterprise is essential.	ME2.1								

UNIX/LINUX Operating System Security Audit/Assurance Program

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
1.6.1 Identify the drivers for a successful review. (This should exist in the assurance function's standards and procedures document.)									
1.6.2 Communicate success attributes to the process owner or stakeholder and obtain agreement.									
1.7 Define audit/assurance resources required. The resources required are defined in the introduction to this audit/assurance program.	ME2.1								
1.7.1 Determine audit/assurance skills necessary for review.									
1.7.2 Determine estimated total resources (hours) and timeframe (start and end dates) required for review.									
1.8 Define deliverables. The deliverable is not limited to the final report. Communication between the audit/assurance teams and the process owner is essential to assignment success.	ME2.1								
1.8.1 Determine the interim deliverables, including initial findings, status reports, draft reports, due dates for response and the final report.									
1.9 Communications The audit/assurance process is clearly communicated to the customer/client.	ME2.1								
1.9.1 Conduct an opening conference to discuss the review objectives with the executive responsible for operating systems and infrastructure.									
2. PREPARATORY STEPS									
2.1 Obtain and review the current organization chart for the operating systems configuration and security functions.									
2.1.1 Identify the key staff and stakeholders.									
2.2 Select the servers to be included in the review.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
2.2.1 Based upon the prioritized list of servers developed previously, identify the servers to be included in the review. Be sure that there is a representative sample of high risk servers. A group of servers may have similar functions and can be aggregated into a group.									
2.2.2 Determine if there is a corporate standard server configuration and related settings for each type of server.									
2.3 Obtain documentation for the servers to be reviewed.									
2.3.1 Obtain the following file listings using UNIX/LINUX utilities or reporting software[3]: • Inittab • Group • Passrd • Shadow • Uucp • Uucp/systems devices • Uucp/devices • Uucp/systems • Uucp/permissions • Cron.allow and cron.deny • At.deny • Ftpusers • Inetd.conf • Hosts.lpd • Hosts.equiv • At.allow and at.deny • Crontab • /etc/system on some UNIX flavors									

[3] Consult UNIX/LINUX documentation for specific commands and locations.

UNIX/LINUX Operating System Security Audit/Assurance Program

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
• /etc/shells • Services • /etc/josts									
2.3.2 Obtain the access permissions for the following directories: • /etc • /bin • /dev • /lib • /usr • /kernel • /usr/spool/cron/cronbats • /tmp • /etc/ftpusers • /etc/security • /proc									
2.4 Obtain an understanding of the operating environment and management issues.									
2.4.1 Interview the senior operating systems management analyst (manager or director) to obtain an understanding of policy and procedures as well as known issues.									
2.4.2 Interview the system administrator and/or network administrator job role/function to obtain an understanding of physical communication interfaces/network adapters/protocols (fixed or virtual IPs/protocols).									
3. ACCESS AND AUTHORIZATION									
3.1 Root Audit/assurance objective: Root access should be limited to administrators requiring access; access should be monitored and logged.									
3.1.1 Limit access to root **Control: Direct root access is restricted to the console.**	AI3.2 DS5.3 DS5.4			X					

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
3.1.1.1 Determine if only one account is defined as root (uid=0) by issuing the command Awk –F":" /etc/passwd '$3 = = 0 {print $0}' | more.	DS5.5								
3.1.1.2 If more than one account is defined with uid=0, determine the account owner, the reason for this access right, and verify written authorization for this access by the information security officer.									
3.1.1.3 Determine if the root password is complex and its use restricted.									
3.1.1.4 Determine the process for monitoring use of the root password.									
3.1.1.5 Determine if the password is changed, how often and what events trigger a password change.[4]									
3.1.1.6 Determine if root logons are disabled and privileged users log on using a general user ID.									
3.1.1.7 Determine if the SUDO (superuser do) facility is installed to limit the command set for specific users, and log SUDO activities.									
3.1.1.8 Determine if LDAP or Pluggable Authentication Module (PAM) is being used. Verify the authentication process is used throughout the enterprise.									
3.1.1.9 If LDAP/PAM is being used, verify the fail-over mechanism of the authentication request.									
3.1.1.10 Verify that the authentication applications use the PAM mechanism. If not in use, suggest the implementation of a single authentication mechanism.									

[4] Checking the last date of password change by any user, including the root user, is a little tricky on UNIX. The /etc/shadow file has several fields—the 5th field will indicate a number, which is the number of days elapsed after 01-Jan-1970 on which the user has changed the password. In some versions of UNIX this may be the seconds elapsed after 00.00 hours on 01-Jan-1970.

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
3.1.2 Limit resources available at the root directory **Control: Files accessible through the root directory are limited to those resources needed to log into the system.**	AI3.2			X					
3.1.2.1 Root's logon files (e.g., the .profile, .cshrc, .kshrc) should not run any other files not owned by root or which are group or world writable. Any exceptions should be documented by the system administer and forwarded to the security administrator.									
3.1.2.2 Verify that the local directory "." should not be in root's search path and the root's PATH does not contain "." and extraneous colons ":" are removed.									
3.1.3 Built-in IDs **Control: Built-in IDs are renamed and/or are assigned new passwords and the built-in IDs are assigned to an individual for accountability.**	AI3.2 DS5.4								
3.1.3.1 Verify that all IDs built in by the manufacturer of hardware and software are renamed or assigned new password and/or assigned to a person with a sole responsibility. Examples are: oadmin, root, admin, sysadmin, shutdown, poweroff, guest, demo, gast, Informix, oracle, ingres, sam_exec and sap.									
3.2 Superusers Audit/assurance objective: Superuser accounts should be limited to those accounts required by the system; administrator access should be through the SUDO or SU command or similar processes.									
3.2.1 System users **Control: Systems users are limited to those required by the operating system.**	AI3.2 DS5.3			X					
3.2.1.1 Verify that each user on the system has an associated password using the commands pwck and grpck. (These two commands are available on most of the UNIX systems and will give a report on the inconsistencies in /etc/password and /etc/group files as under									

UNIX/LINUX Operating System Security Audit/Assurance Program

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
Validation of the number of fields, logon name, user ID, group ID, and if the logon directory and the program-to-use-as-shell exists.)									
3.2.1.2 Verify if the current directory (".") exists anywhere in the root search path.									
3.2.1.3 Verify that all "su" attempts are logged in the "sulog" or other relevant file.									
3.2.1.4 Verify that a home directory has been established for root, and root is restricted to that home directory									
3.2.2 Vulnerable commands **Control: Vulnerable commands are appropriately administered and monitored.**	AI3.2			X					
3.2.2.1 Verify that certain vulnerable commands (i.e., SU, SUDO, /bin/vi, /bin/sh) are granted by the administrator (root) to other users.									
3.2.2.2 Determine if their level of access (full or partial) is necessary for their job function.									
3.2.2.3 Determine if their use is monitored, by whom and the review process.									
3.2.2.4 Determine if their access rights are reviewed regularly (at least each quarter).									
3.3 PS (Process Status Command) Control: The PS command is restricted to root users.	AI3.2								
3.3.1 Verify that the PS command is restricted to root users and assigned through group permissions.							3.		
3.4 Script Usage Control: Script processing on servers is minimized or prohibited.	AI3.2 DS13.1 DS13.2								
3.4.1 Verify that script processing is either prohibited or limited.							3.		

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
3.4.2 Verify that scripts require appropriate documented approvals, including justification, prior to implementation.							3.4		
3.5 General user Audit/assurance objective: General user accounts should be assigned to individual users, the user ID (including number) should be unique, password composition should be in compliance with organization password complexity standards and unnecessary user IDs with special functions should be disabled.	DS5.3								
3.5.1 Unnecessary system accounts are disabled **Control: Unnecessary system user IDs are disabled.**	AI3.2			X					
3.5.1.1 Verify that sys, uucp, nuucp and guest are disabled.									
3.5.1.2 Verify that the following accounts are disabled unless there is a demonstrated need to enable: servdir, sync, shutdown, halt, mail, news, operator, games, gopher, mysql, ftp, anonymous.									
3.5.1.3 Verify that systems accounts may not use FTP: root, smtp, daemon, bin, sys, adm, uucp, nuucp, listen, lp, lpd, guest, nobody, noaccess.									
3.5.1.4 Verify that disabled accounts have a null shell (/dev/null or /bin/false) assigned and an invalid password in the /etc/shadow file.									
3.5.2 Unique user IDs and password complexity **Control: Each user logon is unique and password complexity policy is followed.**	DS5.3			X					
3.5.2.1 Verify that passwd complexity controls available in the vendor's UNIX /LINUX implementation are enabled.									
3.5.2.1.1 For AIX, determine if the AIX password configuration is set to provide maximum complexity.									
3.5.2.1.2 For Solaris, determine that the parameters in the /etc/default/passwd file are set for maximum password complexity.									
3.5.2.1.3 For LINUX, determine if the global settings are set to									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
maximum password complexity and determine if additional password complexity add-ins are used to enhance security.									
3.5.2.2 Review the /etc/passwd file to ensure that all users have a unique ID. The following commands can be used to identify the user: • Names that are not unique use cat /etc/passwd | awk –F”:” ‘{print $1}’ | sort | uniq –c | sort –r | grep –v “^ 1”|more • User IDs that are not unique use cat /etc/passwd | awk –F”:” ‘{print $3}’ | sort | uniq –c | sort –r | grep –v “^ 1”|more									
3.5.2.3 Obtain a utility for the version of UNIX running and determine that the user IDs in the password and shadow password files match and the password complexity is validated for adherence to policy.									
3.5.3 Password attributes **Control: Password attributes (frequency of change, length of password, reuse of passwords) are established according to policy and according to the sensitivity of information available to the user.**[5]	DS5.3								
3.5.3.1 Determine if password frequency change is in alignment with policy and is based on the sensitivity of data for which the user is responsible.									
3.5.3.2 Determine if password reuse (number of generations before being usable) is in alignment with policy.									
3.5.3.3 Verity that the quality of passwords is regularly evaluated using appropriate tools.									
3.6 Warning banner Audit/assurance objective: A warning banner should be installed, warning the user of the purpose, the computer environment and adherence to policy.	DS5.3								
3.6.1 Warning banner **A warning banner is installed that displays when anyone logs into the server.**				X					

[5] This step may have been a part of the identity management review.

UNIX/LINUX Operating System Security Audit/Assurance Program

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
3.6.1.1 Verify that the warning banner is set in /etc/motd and /etc/issue.									
3.7 Group Audit/assurance: Group identification should be to facilitate permission administration, but not for logon or accountability purposes.									
3.7.1 Group logon disabled **Control: logon using group accountability is not permitted.**	AI3.2			X					
3.7.1.1 For each group in the /etc/group file review members of user groups to ensure that access permitted to group members is authorized.									
3.8 Boot Setup Control: During the boot process, insecure commands cannot be executed by nonauthorized persons.									
3.8.1 Verify that a boot password is enabled in the BIOS or Open Boot Prompt (OBP).							3.		
3.8.2 Verify that the password mechanism has been enabled in the LILO or GRUB bootloader.							3.		
3.9 System directories Audit/assurance objective: Critical system directories should be protected from unauthorized access.									
3.9.1 System directories permissions set for maximum security **Control: System directories are set to maximum security and users are not assigned to groups that have access to systems directories**	AI3.2			X					
3.9.1.1 Verify that all system directories or executables have maximum permissions RWXR-XR-X; system control files and scripts RWXR--R--									
3.9.1.2 Verify that the /etc/shadow file is set to –rw------ (600), and the /etc/passwd file is set to –rw-r---r--- (644).									
3.9.1.3 Verify that all files that are writeable by OTHER are set to no write for group and world.									

UNIX/LINUX Operating System Security Audit/Assurance Program

<table>
<tr><th rowspan="2">Audit/Assurance Program Step</th><th rowspan="2">COBIT Cross-reference</th><th colspan="5">COSO</th><th rowspan="2">Reference Hyper-link</th><th rowspan="2">Issue Cross-reference</th><th rowspan="2">Comments</th></tr>
<tr><th>Control Environment</th><th>Risk Assessment</th><th>Control Activities</th><th>Information and Communication</th><th>Monitoring</th></tr>
<tr><td>3.9.1.4 Verify that the permissions for the critical directories below have the other (world) umask set to r-x (read/execute/no write). Critical directories include but are not limited to the following:
<table>
<tr><th>Directory</th><th>Others Per-mission Setting</th><th>Owner</th><th>Typical Group</th></tr>
<tr><td>/bin</td><td>r-x</td><td>bin</td><td>bin</td></tr>
<tr><td>/dev</td><td>r-x</td><td>root</td><td>sys</td></tr>
<tr><td>/dev/dsk</td><td>r-x</td><td>root</td><td>other</td></tr>
<tr><td>/dev/rdsk</td><td>r-x</td><td>root</td><td>other</td></tr>
<tr><td>/etc</td><td>r-x</td><td>root</td><td>sys</td></tr>
<tr><td>/etc/conf</td><td>r-x</td><td>root</td><td>sys</td></tr>
<tr><td>/etc/default</td><td>r-x</td><td>root</td><td>sys</td></tr>
<tr><td>/etc/init.d</td><td>r-x</td><td>root</td><td>sys</td></tr>
<tr><td>/etc/log</td><td>r-x</td><td>root</td><td>sys</td></tr>
<tr><td>/etc/perms</td><td>r-x</td><td>root</td><td>sys</td></tr>
<tr><td>/lib</td><td>r-x</td><td>bin</td><td>bin</td></tr>
<tr><td>/root</td><td>r-x</td><td>root</td><td>bin</td></tr>
<tr><td>/shlib</td><td>r-x</td><td>root</td><td>sys</td></tr>
</table></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3.9.1.5 Determine if there are other directories requiring additional security and verify that they have an owner of the root and appropriate permission settings.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3.9.1.6 Analyze the listing of all other directories containing operating system files.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3.9.1.7 Identify the users and groups that are permitted to access the list of operating system files.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3.9.1.8 Determine if this access is necessary.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3.9.1.9 Determine if there is a procedure for regularly analyzing use of these files and monitoring access maintenance as job functions change.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
3.9.2 Setuid and setgid **Control: Use of setuid and setgid are monitored; files assigned the higher privilege are monitored and reviewed.**	DS5.5			X					
3.9.2.1 Determine if the use of setuid and setgid requires approvals each time the command is used.									
3.9.2.2 Determine if a list of files having setuid and setgid applied exists.									
3.9.2.3 Determine if the baseline includes a list of files with the permission bit "s" set.									
3.9.2.4 Determine if the regular baseline compares permissions tests for this feature.	DS5.5								
3.9.2.5 Review programs with the setuid and/or setgid bit set. (Many of the setuid and setgid programs are used only by root, or by the user or group-ID to which they are set. They can have setuid and setgid removed without diminishing the user's abilities to get work done.)									
3.9.2.6 Verify that permissions prevent modification by other (world).									
3.10 User profiles and restricted shells Audit/assurance objective: User profiles should be managed by administrators; users should not be allowed to modify their profiles; standard shells should be assigned unless a special shell is required; and the default permission for user-created files should be secure.									
3.10.1 User profiles **Control: Standard user profiles are assigned to individuals, which provide the necessary permissions to perform their job function.**	AI3.2 DS5.3			X					
3.10.1.1 Identify the standard user profiles in use and determine if the default user profile is appropriate for most users.									
3.10.1.2 Verify that users cannot modify their profile.									
3.10.1.3 Verify that the permission for the profile files in the user's folder is to read/execute (-r-x).									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
3.10.1.4 Verify that the group is appropriate and that groups don't have access to the .profile.									
3.10.1.5 Verify that the default umask when creating files in their directory is set to 077 (-rwx------) user:rwx only) - review .profile file for s sample of users (both regular and superusers).									
3.10.1.6 Test objective: To verify that users can only create files in their own home directory and designated group directories, and that these files are protected from other users' access/.	DS5.5								
3.10.1.6.1 Select a representative sample of users.									
3.10.1.6.2 Verify that the .profile has a umask entry of 077									
3.10.1.6.3 Verify that the contents of the home directories have an effective permission of 077 (-rwx------).									
3.10.1.7 Test objective: To verify the use of standard profiles, and identify users who can change their own profiles.	DS5.5								
3.10.1.7.1 Select a sample of .profile from the user directories.									
3.10.1.7.2 Identify profiles that are not restricted by umask.									
3.10.1.7.3 For those profiles, examine the contents and determine if the user has modified the profile.									
3.10.2 Shells **Control: Users are assigned a standard shell unless their function requires enhanced or restricted access functions. Use of nonstandard shells is controlled and monitored.**	AI3.2 DS5.4 DS5.5			X					
3.10.2.1 Determine if nonstandard shells are used.									
3.10.2.2 If they are used, determine how restrictive and enhanced shells are assigned, managed and reviewed.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
3.10.2.3 Test objective: To verify that nonstandard shells are managed and re-evaluated based on need.	DS5.5								
3.10.2.3.1 Select a sample of users assigned nonstandard shells.									
3.10.2.3.2 Determine the scope of their job function and compare to the capabilities using the shell—identify mismatches.									
3.11 User and group directory permissions Audit/assurance objective: Users and groups should have access to directories required to fulfill job function.									
3.11.1 Groups **Control: Groups are assigned appropriate permissions, and users are assigned to groups based on job function.**	DS5.4			X					
3.11.1.1 Identify the groups in the /etc/group file.									
3.11.1.2 Determine if the permissions within the group file are appropriate based on the job function of the group.									
3.11.1.3 Determine if the members of the group are appropriate based on the scope of the group's access and the job functions of the group.									
3.11.1.4 Test objective: To verify that the group access rights are appropriate.	DS5.5								
3.11.1.4.1 Select a sample of groups from different business units and management levels.									
3.11.1.4.2 For those selected, review the directory permissions and compare the identify group access requirements against appropriate separation of duties (SOD) tables.									
3.11.2 User permissions **Control: User permissions are initially set by group; additional permissions are based on job function.**	DS5.4			X					
3.11.2.1 Determine if individual users have appropriate permissions in excess									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
of the group assignment based on job function.									
3.11.2.2 Test objective: To verify that the users' access rights are appropriate.	DS5.5								
3.11.2.2.1 Select a sample of users from different business units and management levels.									
3.11.2.2.2 For those selected, review the directory permissions and compare the identify user access requirements against appropriate SOD tables.									
3.11.2.3 Verify that all files and directories have owners.									
4. NETWORK									
4.1 Services Audit/assurance objectives: Only services required for the operation of the operating system should be enabled.									
4.1.1 Network services permitted through routers **Control: Only the external network services that are required to operate the specific server applications are permitted through the external or internal routers.**	AI3.2 DS5.10			X					
4.1.1.1 Verify that only those services that are required from outside the domain are allowed through router filters. If it is not required, routing should also be turned off on the UNIX system.									
4.1.1.1.1 Solaris: This is controlled by /etc/init.d/inetinit. To turn off routing on a Solaris 2.5 machine, touch /etc/notrouter.									
4.1.1.1.2 AIX: To dynamically turn off routing, use no -o ipforwarding=0. To turn it off at system boot add that line to the /etc/rc.net file.									
4.1.2 Services are enabled based on server functionality	AI3.2			X					

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
Control: Services requirements are evaluated based on the function of the server; unnecessary services are disabled.									
4.1.2.1 Determine if there are baseline configuration requirements for each type of server (i.e., database, file, print, web).									
4.1.2.2 Test Objective: To verify that the appropriate services are enabled on the servers.	DS5.5								
4.1.2.2.1 Collect and/or prepare server inventory and classify server (e.g., by server function/ service type) viz. application, network, print server, file server, security/firewall, storage, desktop, Desktop publishing (DTP), thin client, etc.									
4.1.2.2.2 Using the implementation-specific commands (inetd.conf), identify the services running and evaluate if they are appropriate to the mission of the server. Consider the following as potential candidates for disabling: rusersd, rstatd, rwalld, shell (rsh), comsat, tftp, netstat, login (rlogin), talk, finger, time, exec (rexec), uucp, sysstat, echo, name, discard, daytime, chargen, sprayd and bootps.									
4.1.2.2.3 Inspect the running services (command PS–EF) and listening ports (commands netstat–a and rpcinfo) to verify listening services.									
4.1.2.2.4 Verify xwindows is not configured to remotely export X sessions (TCP port 6000-6010)									
4.1.2.2.5 Verify that access to XServer is restricted.									
4.1.2.3 Determine if remote access is required for the servers under review.									
4.1.2.4 If remote access is not required, or the preferred ssh keys are used, verify that the .netrc files are deleted.									
4.1.2.5 If remote access is required, and the .netrc file is present, verify that the file allows only the owner (-400) to be permitted to read the file.									

UNIX/LINUX Operating System Security Audit/Assurance Program

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
4.1.2.6 Determine if remote shell access is not required. If that is the case, verify that rshd is commented out in the file /etc/inetd.conf or protect it with a Transmission control protocol (TCP) Wrapper.									
4.1.2.7 Verify that Single Network Management Protocol (SNMP) service is disabled, whenever not required.									
4.1.2.8 Verify that SNMP community string is hard to guess.									
4.1.2.9 Verify that SNMP version 3 is installed.									
4.1.2.10 Monitor all services started via the inetd or xinted Internet daemon. If inetd is used, all services should be secured via the TCP Wrapper.									
4.1.3 Access to network configuration files **Control: Access to network configuration files is limited to administrators, the host names are accurately identified and services are appropriately enabled.**	AI3.2			X					
4.1.3.1 Verify that access to /etc/hosts (host name), /etc/services (ports and protocols) and /etc/inetd.conf (daemon to start network services) is limited to the root directory.									
4.1.3.2 Verify that each host exists and has the correct IP address.									
4.1.3.3 Verify that network services in inetd.conf have been removed if not needed; /etc/services is only a reference file.									
4.1.3.4 Verify that network services in startup scripts have been removed if they are not required.									
4.1.4 System date and time **Control: System date and time stamp should be synchronized with a standard date and time using the Network Time Protocol (NTP)**									
4.1.4.1 Verify that the servers are utilizing the system time and date synchronization via NTP.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
4.2 File Transfer Protocol (FTP) controls Audit/assurance objectives: FTP should be disabled if not needed, and secured with passwords and best-practice permissions if needed.									
4.2.1 Disable FTP **Control: FTP is disabled.**	AI3.2								
4.2.1.1 Determine if the servers require FTP according to their function.									
4.2.1.2 Verify that the FTP daemon is disabled and ports 20-21 are disabled.									
4.2.2 FTP configuration is secure **Control: Good practice FTP configuration settings have been implemented.**	AI3.2			X					
4.2.2.1 Verify that ssh and scp are used for Telnet and FTP.									
4.2.2.2 Verify that root users are not allowed to log into the system via the ssh configuration.									
4.2.2.3 Verify that .netrc has been removed. Passwords contained in the file put other systems at risk.									
4.2.2.4 If FTP is required, review /etc/ftpusers to ensure that users who will not be permitted to use FTP have their user IDs in this file. Verify that the root user ID is also in the file to prevent the root user from using FTP.									
4.2.2.5 Verify that there are no real users or passwords in ftp/etc/passwd.									
4.2.2.6 Verify that FTP home is assigned its own file system and is isolated from the rest of the system.									
4.2.2.7 Verify that anonymous FTP has been disabled.									
4.2.2.8 Verify that the shell for the FTP user in /etc/passwd entry is set to /dev/null.									
4.2.2.9 Verify the owner permissions for the following: • ftp/root dr-xr-xr-x									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
• ftp/bin/root d--x--x—x • ftp/bin/ls d--x--x—x • ftp/etc/root d--x--x—x • ftp/etc/passwd/root -r--r--r— • ftp/etc/group/root -r--r--r— • ftp/pub/root drwxrwxrwt									
4.2.2.10 Determine if Trivial file transfer protocol (TFTP) is required.									
4.2.2.11 If TFTP is not required, it should be disabled or secured in the in the inetd.conf file.									
4.2.2.12 Check inetd.conf file to verify that it runs with secure option (-s) to force TFTP to restrict access to the restricted directory.									
4.3 Sendmail Audit/assurance objective: Sendmail should be controlled or disabled, based on the function of the server.									
4.3.1 Sendmail **Control: Sendmail should be disabled if the system is not running as a mail server.**	AI3.2								
4.3.1.1 Verify that the sendmail daemon has been disabled or is not listening for incoming mail connections (SMTP over TCP port 25) on systems that are not configured/required as a mail server.									
4.3.1.2 Determine if the more secure Smrsh sendmail alternative is in use and properly configured.									
4.4 Trusted hosts Audit/assurance objective: The use of remote access by defining a trusted host, should be avoided.									
4.4.1 Trusted hosts are disabled **Control: Trusted hosts facilities are disabled.**	AI3.2			X					
4.4.1.1 Verify that the file /etc/hosts.equiv has been removed from the system.									

UNIX/LINUX Operating System Security Audit/Assurance Program

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
4.4.1.2 Verify that no .rhosts files are in the user home directories or anywhere else on the server.									
4.4.2 Trusted hosts are limited in their permissions **Control: Trusted hosts are limited in their scope of resources.**	AI3.2			X					
4.4.2.1 Determine if trusted hosts are permitted in the information security policy.									
4.4.2.2 If trusted hosts are permitted, determine the justification and if other techniques have been considered.									
4.4.2.3 Determine if the information security officer has approved the use of trusted hosts.									
4.4.2.4 Determine if other control mechanisms (certificates, public key infrastructure [PKI]) can be employed.									
4.4.2.5 Verify that the /etc/hosts.equiv has the following: • Only a small number of trusted hosts listed based on a demonstrated need to use the resource. • Only trust hosts within the local domain or under an organization’s management are issued access. • The trusted host is listed in the /etc/hosts file. • The character ‘+’ is not listed by itself anywhere in the file since this may allow any user access to the system. • The characters ‘!’ or ‘#’ are not used since there is no comment character for this file. • The first character of the file is not ‘-’ (indicating a script). • The permissions are set to 400. (world:none;group:none:user:read/no write/no execute). • The owner is set to root.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
4.4.2.6 Verify that only users requiring access have an .rhost file in their home directory. Verify each .rhost file for the following: • The permissions are set to 600 (world:none;group:none:user:read/write/no execute). • The first character of the file is not '-' (indicating a script). • The owner of the file is the account's owner. • The file does not contain the symbol "+" on any line as this may allow any user access to this account. Usage of netgroups within .rhosts does not allow unintended access to this account. The characters '!' or '#' are not used since there is no comment character for this file.									
4.5 File systems Audit/assurance objective: Remote mounting of file systems should be minimized.									
4.5.1 Remote file system mounting **Control: Remote file system mounting using the network file system (NFS) should be disabled, if possible. Other file sharing mechanisms should be employed.**				X					
4.5.1.1 Determine if remote file system mounting is used and the justification for it.									
4.6 If it is not used, verify that NFS is not enabled.									
4.6.1 Using NFS **Control: If NFS is utilized, good practice configurations are implemented.**	AI3.2			X					
4.6.1.1 Determine if the following best practices are implemented: • The NFS server is not self-referenced in its own exports file. • Exports files don't contain a "localhost" entry. • Export file systems only to hosts that require them. • Export lists do not exceed 256 characters; aliases should not exceed 256 characters after the aliases have been expanded.									
4.6.1.2 Verify that only the file systems needed are exported.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
4.6.1.3 Verify that export file systems are not unintentionally permitted to the world. Use -access=host.domainname.com option or equivalent for each file system exported.									
4.6.1.4 Verify that file systems are set to read-only (-ro) whenever possible.									
4.6.1.5 Verify that the permissions on the export file is 644 and the owner is root.									
4.6.1.6 Verify that the NFS client maps the file systems as nobody, unless specifically required.									
4.6.1.7 Verify that the file systems use the –nosuid option for mounting folders, to prevent execution of setuid programs.									
4.6.1.8 Verify that the file systems use the option anon=-1 to completely disable the anonymous access.									
4.7 Network interface card (NIC) Audit/assurance objectives: The NIC should not operate in "promiscuous mode."									
4.7.1 NIC operating mode **Control: The NIC operates in standard mode, not promiscuous mode.**	AI3.2			X					
4.7.1.1 Verify that the NIC does not operate in promiscuous mode.									
4.7.1.2 Verify that unused physical and/or logical interfaces have been disabled.									
4.8 Bluetooth Audit/assurance objective: Bluetooth should be configured securely to prevent unauthorized access and use.									
4.8.1 Bluetooth Configuration **Control: Bluetooth is configured for maximum security**							4.		
4.8.1.1 Verify that Bluetooth device is not discoverable.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
4.8.1.2 Verify that Bluetooth device is configured so that a remote device is not able to identify the Bluetooth device class.									
4.8.1.3 Verify that only required Bluetooth Information Exchange Services (BIE) are enabled. All other services must be disabled.									
4.8.1.4 Determine that Bluetooth Personal Area Networking (PAN) is not allowed and all related services are disabled.									
4.8.1.5 Verify that Bluetooth COM port services are used for serial communication, modem connection and synchronization purposes.									
4.8.1.6 Verify that file transfer over Bluetooth uses a secure password mechanism, (whenever required) per agreed password policy.									
4.8.1.7 Verify that the user is informed interactively by a dialogue, when data/file is received over the Bluetooth interface.									
4.8.1.8 Verify that automatic association of received file, e.g., v-card is disabled.									
5. MONITORING AND AUDITING THE SYSTEM									
5.1 Audit Facilities Audit/assurance objective: Audit logs should be produced to effectively record essential activities, provide evidence of management review and actions based on incidents identified, and retained to provide evidence for forensic reviews and investigations.									
5.1.1 Sulog **Control: The sulog is maintained and reviewed for use of the su (sudo) command and cron events.**	DS5.5			X					
5.1.1.1 Determine if the systems administrator regularly maintains and reviews the sulog for unauthorized su attempts. (The ability to use superuser or the su command should be reserved for only the system administrator(s). The sulog will show how many attempts have been									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
made to a superuser account. If the sulog is not maintained, there are no other utilities available to track attempts to use superuser accounts. The sulog can generally be viewed with the command more /usr/adm/sulog. It can become quite large if not reviewed daily.)									
5.1.2 Syslog **Control: The syslog is configured to record significant activities, and is maintained and reviewed for failed logons and other system events.**	AI3.2 DS5.5 DS13.3			X					
5.1.2.1 Identify which activities are configured for recording in the syslog file and evaluate if the appropriate activities are monitored.									
5.1.2.2 Determine if the systems administrator regularly maintains and reviews the syslog for significant events.									
5.1.3 Cron log **Control: The cron log is maintained and reviewed for cron activities.**	AI3.2 DS5.5 DS13.3			X					
5.1.3.1 Determine if the cron log is generated and reviewed for cron activities.									
5.1.4 Log management **Control: Logs are printed, where appropriate; logs are reviewed; the review process is documented; incidents are placed into the issue management system and are investigated and closed. Those issues requiring escalation are forwarded to the appropriate personnel on a timely basis. Open issues are aged and monitored.**	DS5.5 DS5.7 DS9.2 DS13.3			X					
5.1.4.1 Determine how each of the logs described is managed, reviewed and approved.									
5.1.4.2 Determine the criteria for an incident for each of the logs and the procedure for initiating an incident.									
5.1.4.3 Determine how incidents are monitored, aged, closed or escalated.									
5.1.4.4 Test objective: To verify that systems logs are reviewed and incidents are monitored and managed.	DS5.5								

UNIX/LINUX Operating System Security Audit/Assurance Program

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
5.1.4.4.1 Obtain the sulog, syslog and cron log for a range of dates.									
5.1.4.4.2 Verify that each log has been reviewed with appropriate evidence of review.									
5.1.4.4.3 Determine if an incident should have been initiated based on the criteria established in the policy and if an incident has been established.									
5.1.4.4.4 Follow the incident through the resolution process. Determine if the incident was appropriately followed up, on a timely basis, closed or escalated.									
5.1.4.4.5 For escalated incidents, determine if they have been closed and evaluate the response: • Verifying the incident criteria includes: • Short or incomplete logs • Logs containing unusual timestamps • Logs with incorrect permissions or ownership • Records of reboots or starting of services • Missing logs • su entries or logons from unusual places • Too many failed logons									
5.1.4.5 Verify that all system logs are copied to a central file server for review and archiving.									
5.1.5 Log archive **Control: Logs are archived in secure locations and retained according to policy.**	AI3.2 DS5.5			X					
5.1.5.1 Determine the procedure for archiving the logs described.									
5.1.5.2 Determine how long the logs are retained and if this is in accordance with policy and the needs of potential forensic investigations.									
5.1.5.3 Determine if the security protecting the logs will meet the rules of evidence for the governing bodies.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
5.1.5.4 Test objective: To verify that the logs are maintained according to policy.	DS5.5								
5.1.5.4.1 Select a range of dates for system logs.									
5.1.5.4.2 Determine if the logs are available by requesting their retrieval.									
5.1.5.4.3 Evaluate the security and ability to access and review the logs.									
5.2 Monitoring facilities Audit/assurance objective: Relevant controls should be established in automated monitoring facilities.	DS5.5								
5.2.1 In case multiple systems/servers are available, verify that each facility/system/server writes messages into its own file and directs all messages to a central syslog server.									
5.2.2 Verify the list of specific keywords (e.g., Refused, Denied, WARN, Blocked, Cleaned, Quarantined) used. Tools, e.g., swatch, logsurfer, are used for these tasks.									
5.2.3 Verify if automated monitoring is not available and which of the alternative processes is used for raising alerts/incidents.									
5.2.4 Verify if automated monitoring is not available; log files are evaluated once per week or at regular intervals.									
5.3 Last Log **Control: Last log is reviewed regularly for users not logging out of the system or logged in for prolonged periods.**	DS5.4 DS5.5								
5.3.1 Obtain and review last log for logon/logoff details. Identify users logged on for prolonged periods or not logging off.									
6. OPERATING SYSTEM AND APPLICATION PATCHES AND CONFIGURATION CHANGE MANAGEMENT									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
6.1 Baseline Audit/assurance objective: A baseline of approved daemons, scripts, configurations and user information should be maintained for each server. The baseline should be evaluated for accuracy. The baseline should be compared to the servers on a regular basis to identify irregularities.									
6.1.1 Baseline comparisons **Control: Baseline comparisons are run regularly to identify unauthorized or erroneous baseline changes.**	DS5.5 DS9.3			X					
6.1.1.1 Verify that baseline comparisons are run regularly.									
6.1.1.2 Verify that the baseline comparisons include scripts in the baseline.									
6.1.2 Configuration baseline **Control: An approved list of daemons, scripts, programs, etc. is maintained for each server.**	DS9.1			X					
6.1.2.1 Verify that the baseline includes every executable file that is or could run as root, critical control files and their directories.									
6.1.3 Patch list **Control: A patch list of operating system patches, service packs, etc. is maintained. During the baseline test, patches are verified.**	AI3.3 DS5.5			X					
6.1.3.1 Obtain the patch list for the servers under review.									
6.1.3.2 Determine that the patch list is current as suggested by the operating system vendor.									
6.1.3.3 **Baseline verification** **Control: The baseline configuration is compared to the servers on a regular basis; the comparison is documented and the differences are analyzed.**	DS5.5			X					
6.1.3.4 Determine that the policy has a requirement for the frequency of									

UNIX/LINUX Operating System Security Audit/Assurance Program

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
baseline verification.									
6.1.3.5 Determine if baseline verifications are executed as prescribed in the policy.									
6.1.3.6 Determine if differences are investigated. The baseline or the live operating system may require modification.									
6.1.4 Accounting data recording **Control: An accounting recording and reporting process is installed to ensure that important events are recorded, but the accounting function does not affect performance.**	AI3.2 DS5.3			X					
6.1.4.1 Determine if accounting is turned on and reviewed regularly.									
6.1.4.2 Issue the command ls -l /usr/adm/pacct. If the file exists and has a recent modify date, then accounting is turned on.									
6.1.4.3 Determine if a reporting or filtering process is utilized.									
6.1.4.4 Evaluate the availability of accounting data for forensic review.									
6.2 Operating System Configuration Audit/assurance objective: The operating system should be implemented to provide a good practice security configuration.									
6.2.1 Cron and At **Control: The cron and at daemons are configured to prevent unauthorized access to scheduling capabilities to prevent unauthorized resources from being processed.**	AI3.2 DS13.2			X					
6.2.1.1 Determine if the cron.allow, at.allow, cron.deny and at.deny files are in use.									
6.2.1.2 Determine if the user IDs are identified in the cron.allow, at.allow, cron.deny and at.denyare appropriate. Note: Evidence of approval and justification should be available for users identified in cron.allow.									

UNIX/LINUX Operating System Security Audit/Assurance Program

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
6.2.1.3 Determine if all programs executed as cron jobs are secured from unauthorized modification using file system permissions.									
6.2.1.4 Determine if this table is reviewed regularly.									
6.2.2 UNIX logon program **Control: The appropriate UNIX logon program is used based upon the sensitivity of data on the server.**	AI3.2 DS5.3 DS5.4			X					
6.2.2.1 Determine which logon program is used by the UNIX operating system. Four versions of the logon program can be linked to /bin/login. The most secure version, login.secure, requires a password for the superuser account (root) and restricts the superuser account to log on only at the operator console.									
6.2.2.2 Review the table in the /etc/initab file, which contains the instructions for the init utility. The init utility executes at the system startup that calls the getty utility for each terminal. The getty utility displays a logon prompt and accepts the user name. Next, the getty utility calls the logon utility and validates the user name and password, and starts the shell to accept commands from the user.									
6.2.2.3 Determine if the getty and logon utilities are replaced by other utilities/ programs. These other programs/utilities are called by the init utility.									
6.2.2.4 Verify if NIS or NIS+ is used to authenticate users to the system. Confirm that the contents of the distributed password file cannot be viewed by unprivileged users (command ypcat passwd). Note: Implementation of NIS will allow all users with command line access into the system to view encrypted user account passwords. Ensure that the NIS domain name used cannot be easily found by an attacker.									
6.2.2.5 Operation of an NIS server should be viewed as a security risk. Deactivate NIS. Migrate to secure solutions such as NIS+ or LDPAS/Kerberos.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
6.3 Source code and compilers not present on production servers Audit/assurance objectives: Source code and compilers should not be present on the production servers to prevent unauthorized modification of the operating system kernel or applications.									
6.3.1 Source code **Control: Storing source code on production servers is prohibited.**	PO8.3 AI2.5 AI3.2			X					
6.3.1.1 Verify that the operating system source code is not stored on the server. Search the drives for instances of the operating system source code.									
6.3.2 Compilers **Control: C and other compilers are removed from production servers.**	AI3.2			X					
6.3.2.1 Verify that all compilers have been removed from production servers.									
6.4 Core Files Audit/assurance objective: Core files (generated during a memory dump after a system crash) should be secured if they contain sensitive information.									
6.4.1 Core File Security **Control: Core files containing passwords, etc. are secure.**							6.4		
6.4.1.1 Verify that no core files are retained on the server after they are no longer needed.							6.4		
6.5 Routine operating system configuration changes/updates Audit/assurance objective: Only operating systems configuration changes and updates/upgrades that are authorized, evaluated and prioritized should enter the change process.									
6.5.1 Management of operating system configuration changes **Control: Management classifies, reviews and approves operating system change requests. This control ensures that management has considered the changes in the queue and has approved the changes.**	AI6.2 AI6.4		X	X	X	X			
6.5.1.1 Obtain the enterprise's standards, procedures, and guidelines for									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
identifying, classifying and approving operating system configuration change requests.									
6.5.1.2 Based on reviewing documentation, interview and observation:									
6.5.1.2.1 Determine if a process exists to classify operating system change requests as an infrastructure change.									
6.5.1.2.2 Determine if a process exists to perform a risk assessment that is focused on the impact of the change on other systems or applications.									
6.5.1.2.3 Determine if there is a process to perform an impact analysis on changes to determine the affect the change would have on the integrity and availability of the business process.									
6.5.1.2.4 Determine if the appropriate approvers within IT operations and IT technical support (operating systems responsibility) document their approval of the change.									
6.5.1.2.5 Determine if the change requests are subject to prioritization.									
6.5.1.2.6 If prioritization is performed, determine if appropriate management regularly authorizes the priority.									
6.5.1.3 Test objective: To verify compliance with the review and prioritization process.	DS5.5								
6.5.1.3.1 Using the move ticket, obtain a population of requested changes.									
6.5.1.3.2 When making the selection, select representative samples resulting from emergency requests, systems requests and problem tickets.									
6.5.1.3.3 For each selected ticket:									
6.5.1.3.3.1 Trace the move request to the originating request (systems request, problem ticket or emergency									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
request, if different).									
6.5.1.3.3.2 Determine if there was a risk assessment performed for impact on other systems or applications.									
6.5.1.3.3.3 Determine if there was an impact analysis performed to determine the effect the change would have on the enterprise.									
6.5.1.3.3.4 Determine if the appropriate approvers within IT operations and IT technical support (operating systems responsibility) document their approval of the change									
6.5.1.3.3.5 Determine if the change request had been subject to prioritization.									
6.5.1.3.3.6 If prioritization had been granted, determine if appropriate management had authorized the priority.									
6.5.2 Completion of review and testing prior to implementation **Control: Program changes require signoffs by the appropriate management prior to implementation.**	AI3.2 AI6.2 DS5.5			X	X	X			
6.5.2.1 Determine that the sign-off process prior to a change moving into production includes the following:									
6.5.2.1.1 Technical support indicating completion of testing, quality assurance, documentation.									
6.5.2.1.2 IT operations, indicating acceptance of documentation, scheduling changes, backup changes, runtime changes, etc.									
6.5.2.1.3 Information security, indicating acceptance of information security changes.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
6.5.2.2 Test Objective: To verify the sign-off process to ensure that all signoffs were completed before implementation and that the appropriate personnel approved the move.	DS5.5								
6.5.2.2.1 Obtain a sample of operating system implementation tickets.									
6.5.2.2.2 For each ticket, verify timely approvals by:									
6.5.2.2.2.1 Technical support function, indicating completion of testing, quality assurance, documentation.									
6.5.2.2.2.2 IT operations, indicating acceptance of documentation, scheduling changes, backup changes, runtime changes, etc.									
6.5.2.2.2.3 Information security, indicating acceptance of information security changes.									
6.5.3 Management monitoring of configuration changes/upgrades **Control: Management reviews program changes to ensure that only authorized operating system configuration changes are included in the modification process.**	AI6.4 AI6.5			X		X			
6.5.3.1 Determine if a baseline comparison is executed after an update.									
6.5.3.2 Determine how management reviews the changes to ensure that only authorized changes have been implemented.									
6.5.3.3 Test objective: To verify the process of comparing authorized changes to completed changes:	DS5.5								
6.5.3.3.1 Select a sample of operating system configuration changes/upgrades.									
6.5.3.3.2 Obtain the request supporting the change.									
6.5.3.3.3 Based on the request, obtain an understanding of the changes required.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
6.5.3.3.4 Review the baseline comparison to determine if the changes requested were processed.									
6.6 Emergency changes Audit/assurance objective: Emergency operating system changes should be controlled, documented and initiated only in true emergencies.									
6.6.1 Emergency priority **Control: Changes using the emergency change procedure should be initiated only for changes where time is of the essence.**	AI6.3			X	X	X			
6.6.1.1 Through interviews, observation and review of documentation, determine if there is a definition for an emergency change.									
6.6.2 Emergency testing **Control: Emergency changes are adequately tested before being placed into production**	DS5.5 AI6.3			X	X				
6.6.2.1 Through interviews, observation and review of documentation, determine the process used to review testing procedures before an emergency change is accepted into production.									
6.6.2.2 Review the existence of test results and management review.	DS5.5								
6.6.3 Emergency change approval **Control: Emergency changes are authorized by appropriate management before being placed into production.**	AI6.3			X	X	X			
6.6.3.1 Through interviews, observation and review of documentation, determine the process used to authorize emergency moves to production. Differentiate between minor and major enhancements, operating system and configuration files.									
6.6.3.1.1 Ensure programming function, indicating completion of testing, quality assurance, documentation.									
6.6.3.1.2 Review with user, indicating satisfactory user acceptance									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
test and approval and knowledge of implementation date.									
6.7 Change management governance Control objective: Change management process is subject to management oversight to ensure consistent and timely processing of changes.									
6.7.1 Operating system change summaries **Control: Management receives timely reports summarizing change management activities, key performance indicators and escalation of issues requiring management attention.**	AI6.4	X		X	X	X			
6.7.1.1 Identify the reports that management receive and, the frequency and scope of the reports.									
6.7.1.2 Determine if service level agreements (SLA) are in use. If so, verify that the reports summarize SLA attainment and/or deficiency.									
6.7.1.3 Determine the escalation process for change management processes operating outside of normal conditions.									
7. SYSTEM BACKUP AND RECOVERY									
7.1 System backups Audit/assurance objective: The systems should be backed up on a regular basis in a manner to minimize data loss in the event of hardware or software failure, or a physical incident.									
7.1.1 Backup procedures **Control: Appropriate backup procedures are in effect to ensure backups are scheduled to be executed automatically, and include the necessary files to ensure the capability of restoring the operating system and applications.**	DS11.5			X					
7.1.1.1 Verify that relevant crontab file (usually root) contains the backup procedure.									
7.1.1.2 Compare this information with the contents of the file that is a text file that contains a list of file systems and partitions, which are backed up during a full backup. This file provides the command with a									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
description of the systems in order to execute a proper backup and restore.									
7.1.1.3 Determine that the backup program runs at least nightly.									
7.1.1.4 Determine the frequency with which the backup script is run.									
7.1.1.5 Verify that the frequency is appropriate for the operation being audited.									
7.1.1.6 Examine the backup scripts.									
7.1.1.7 Determine that the relevant files are backed up, including critical system files.									
7.1.1.8 Determine if the sync utility is periodically executed to copy disk buffers to disk so that loss of data is kept to a minimum in the event of system failure. This can be verified by reviewing the contents of the table stored in the \etc\crontab file that lists the programs executed periodically. These programs are executed by the cron utility as background processes.									
7.1.1.9 If the backup process records a status log (e.g., backup success or failure), verify that this is regularly reviewed by the system administrator (i.e., daily).									
7.1.1.10 Determine if write verify passes are made to enhance the reliability of the new archive. Write verify passes will occur by default unless they are disabled.									
7.1.2 Backup generations **Control: Multiple generations of the files are maintained off site.**	DS11.5			X					
7.1.2.1 Determine how many generations of the server files are retained. At least one month of files should be retained. If financial information is stored on the server, the backup must subscribe to regulatory requirements.									

Audit/Assurance Program Step	COBIT Cross-reference	COSO					Reference Hyper-link	Issue Cross-reference	Comments
		Control Environment	Risk Assessment	Control Activities	Information and Communication	Monitoring			
7.1.2.2 Test objective: To verify that the backup generations are adequate for the business processes executed on the server.	DS5.5								
7.1.2.2.1 Select several servers of varying business criticality.									
7.1.2.2.2 Determine the number of generations backup maintained for each server.									
7.1.2.2.3 Determine if the generations are adequate for the business priority.									
7.1.2.2.4 Determine if the offsite rotation of backups is adequate.									
7.1.3 Backup testing **Control: Backup tapes are subject to test restores to ensure readability.**	DS5.5 DS11.5			X					
7.1.3.1 Determine if there is a program to restore backups on a rotational basis and verify the accuracy and readability of the data contained therein.									

VII. Maturity Assessment

The maturity assessment is an opportunity for the reviewer to assess the maturity of the processes reviewed. Based on the results of audit/assurance review, and the reviewer's observations, assign a maturity level to each of the following COBIT control practices.

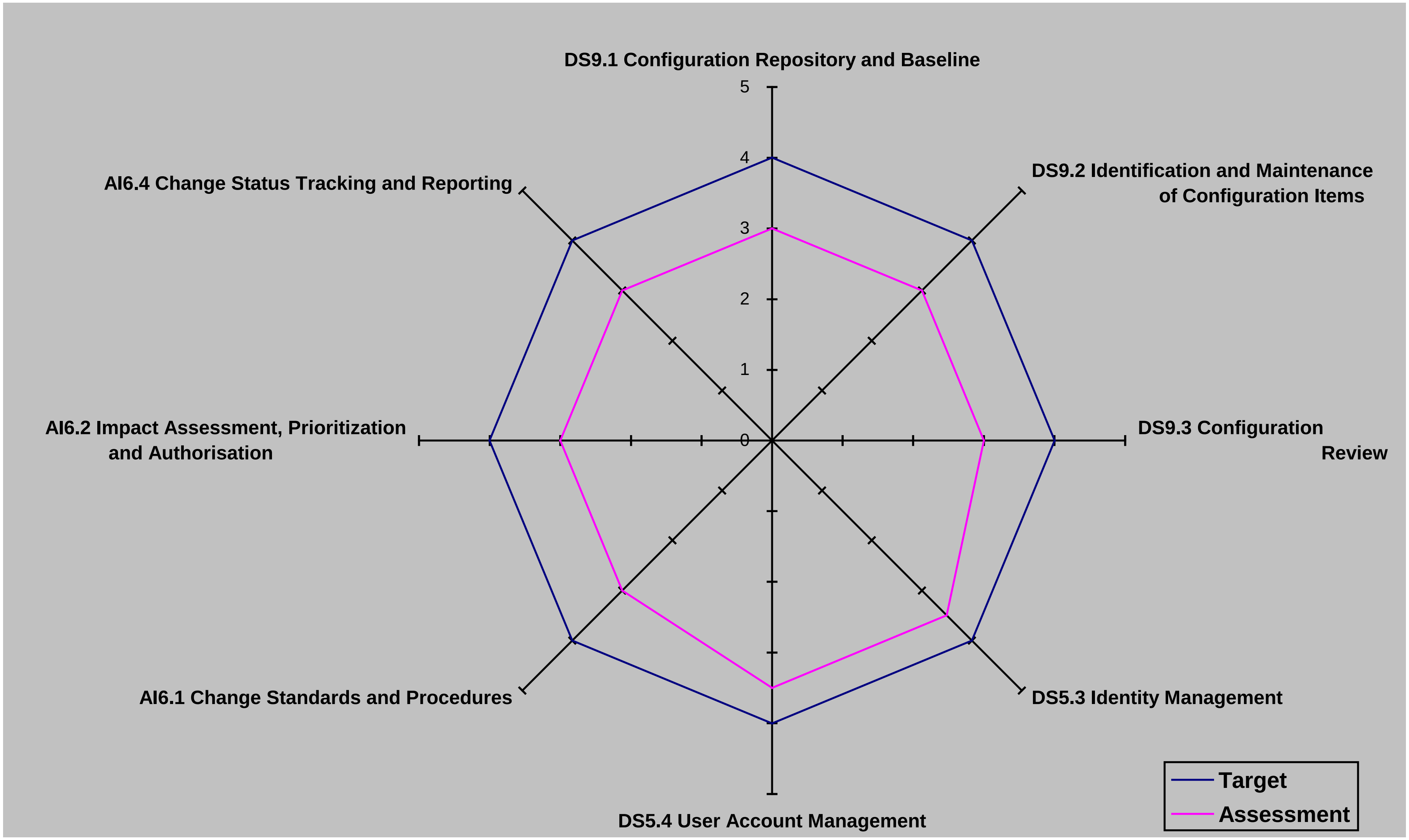
COBIT Control Practice	Assessed Maturity	Target Maturity	Reference Hyperlink	Comments
AI6.1 Change Standards and Procedures 1. Develop, document and promulgate a change management framework that specifies the policies and processes, including: • Roles and responsibilities • Classification and prioritization of all changes based on business risk • Assessment of impact • Authorization and approval of all changes by the business process owners and IT • Tracking and status of changes • Impact on data integrity (e.g., all changes to data files being made under system and application control rather than by direct user intervention) 2. Establish and maintain version control over all changes. 3. Implement roles and responsibilities that involve business process owners and appropriate technical IT functions. Ensure appropriate segregation of duties. 4. Establish appropriate record management practices and audit trails to record key steps in the change management process. Ensure timely closure of changes. Elevate and report to management changes that are not closed in a timely fashion. 5. Consider the impact of contracted services providers (e.g., of infrastructure, application development and shared services) on the change management process. Consider integration of organizational change management processes with change management processes of service providers. Consider the impact of the organizational change management process on contractual terms and SLAs.				
AI6.2 Impact Assessment, Prioritization and Authorization 1. Develop a process to allow business process owners and IT to request changes to infrastructure, systems or applications. Develop controls to ensure that all such changes arise only through the change request management process. 2. Categorize all requested changes (e.g., infrastructure, operating systems, networks, application systems, purchased/packaged application software). 3. Prioritize all requested changes. Ensure that the change management process identifies both the business and technical needs for the change. Consider legal, regulatory and contractual reasons for the requested change. 4. Assess all requests in a structured fashion. Ensure that the assessment process addresses impact analysis on infrastructure, systems and applications. Consider security, legal, contractual and compliance implications of the requested change. Consider also interdependencies amongst changes. Involve business process owners in the assessment process, as appropriate.				

COBIT Control Practice	Assessed Maturity	Target Maturity	Reference Hyperlink	Comments
5. Ensure that each change is formally approved by business process owners and IT technical stakeholders, as appropriate.				
AI6.4 Change Status Tracking and Reporting 1. Ensure that a documented process exists within the overall change management process to declare, assess, authorize and record an emergency change. 2. Ensure that emergency changes are processed in accordance with the emergency change element of the formal change management process. 3. Ensure that all emergency access arrangements for changes are appropriately authorized, documented and revoked after the change has been applied. 4. Conduct a post-implementation review of all emergency changes, involving all concerned parties. The review should consider implications for aspects such as further application system maintenance, impact on development and test environments, application software development quality, documentation and manuals, and data integrity.				
DS5.3 Identity Management 1. Establish and communicate policies and procedures to uniquely identify, authenticate and authorize access mechanisms and access rights for all users on a need-to-know/need-to-have basis, based on predetermined and preapproved roles. Clearly state accountability of any user for any action on any of the systems and/or applications involved. 2. Ensure that roles and access authorization criteria for assigning user access rights take into account: • Sensitivity of information and applications involved (data classification) • Policies for information protection and dissemination (legal, regulatory, internal policies and contractual requirements) • Roles and responsibilities as defined within the enterprise • The need-to-have access rights associated with the function • Standard but individual user access profiles for common job roles in the organization • Requirements to guarantee appropriate segregation of duties 3. Establish a method for authenticating and authorizing users to establish responsibility and enforce access rights in line with sensitivity of information and functional application requirements and infrastructure components, and in compliance with applicable laws, regulations, internal policies and contractual agreements. 4. Define and implement a procedure for identifying new users and recording, approving and maintaining access rights. This needs to be requested by user management, approved by the system owner and implemented by the responsible security person. 5. Ensure that a timely information flow is in place that reports changes in jobs (i.e., people in, people out, people change). Grant, revoke and adapt user access rights in co-ordination with human resources and user departments for users who are new, who have left the organization, or who have changed roles or jobs.				

COBIT Control Practice	Assessed Maturity	Target Maturity	Reference Hyperlink	Comments
DS5.4 User Account Management 1. Ensure that access control procedures include but are not limited to: • Using unique user IDs to enable users to be linked to and held accountable for their actions • Awareness that the use of group IDs results in the loss of individual accountability and are permitted only when justified for business or operational reasons and compensated by mitigating controls. Group IDs must be approved and documented. • Checking that the user has authorization from the system owner for the use of the information system or service, and the level of access granted is appropriate to the business purpose and consistent with the organizational security policy • A procedure to require users to understand and acknowledge their access rights and the conditions of such access • Ensuring that internal and external service providers do not provide access until authorization procedures have been completed • Maintaining a formal record, including access levels, of all persons registered to use the service • A timely and regular review of user IDs and access rights 2. Ensure that management reviews or reallocates user access rights at regular intervals using a formal process. User access rights should be reviewed or reallocated after any job changes, such as transfer, promotion, demotion or termination of employment. Authorizations for special privileged access rights should be reviewed independently at more frequent intervals.				
DS9.1 Configuration Repository and Baseline 1. Implement a configuration repository to capture and maintain configuration management items. The repository should include hardware; application software; middleware; parameters; documentation; procedures; and tools for operating, accessing and using the systems, services, version numbers and licencing details. 2. Implement a tool to enable the effective logging of configuration management information within a repository. 3. Provide a unique identifier to a configuration item so the item can be easily tracked and related to physical asset tags and financial records. 4. Define and document configuration baselines for components across development, test and production environments, to enable identification of system configuration at specific points in time (past, present and planned). 5. Establish a process to revert to the baseline configuration in the event of problems, if determined appropriate after initial investigation. 6. Install mechanisms to monitor changes against the defined repository and baseline. Provide management reports for exceptions, reconciliation and decision making.				
DS9.2 Identification and Maintenance of Configuration Items 1. Define and implement a policy requiring all configuration items and their attributes and versions to be identified and maintained.				

COBIT Control Practice	Assessed Maturity	Target Maturity	Reference Hyperlink	Comments
2. Tag physical assets according to a defined policy. Consider using an automated mechanism, such as barcodes. 3. Define a policy that integrates incident, change and problem management procedures with the maintenance of the configuration repository. 4. Define a process to record new, modified and deleted configuration items and their relative attributes and versions. Identify and maintain the relationships between configuration items in the configuration repository. 5. Establish a process to maintain an audit trail for all changes to configuration items. 6. Define a process to identify critical configuration items in relationship to business functions (component failure impact analysis). 7. Record all assets—including new hardware and software, procured or internally developed—within the configuration management data repository. 8. Define and implement a process to ensure that valid licenses are in place to prevent the inclusion of unauthorized software.				
DS9.3 Configuration Integrity Review 1. To validate the integrity of configuration data, implement a process to ensure that configuration items are monitored. Compare recorded data against actual physical existence, and ensure that errors and deviations are reported and corrected. 2. Using automated discovery tools where appropriate, reconcile actual installed software and hardware periodically against the configuration database, license records and physical tags. 3. Periodically review against the policy for software usage the existence of any software in violation or in excess of current policies and license agreements. Report deviations for correction.				

VIII. Assessment Maturity vs. Target Maturity—UNIX/LINUX Only



Auditing Linux/Unix Server Operating Systems

Muhammad Mushfiqur Rahman, CISA, CEH, CHFI, CCNA, ISO 27001 LA, ITIL V3, MCITP, MCP, MCSE, MCTS, OCP, SCSA, has 12 years of IT operations, project management and custom business solutions, enterprise resource planning implementation, and information security analysis and management experience. Rahman is an information security analyst at Eastern Bank Limited, Bangladesh. He also has 12 years of experience teaching IT courses for end users and IT professionals. He can be reached at *mushfique98@gmail.com.*

Server auditing is an important task to ensure platform-level security in an IT infrastructure and to ensure the proper configuration of Linux server security. The Linux system has its own security configuration and management system to address the security requirements in an enterprise environment. The system administrator needs to configure the Linux system to get more security assurance from the system, and IS auditors need to check the Linux system configuration as per audit standards to ensure the secure system is in place in the enterprise.

It is an exigent task for a system administrator to secure the production system from malicious attacks.

AUDITING PHYSICAL SYSTEM SECURITY

Physical security is the first and foremost task for any information system audit. Auditors must determine that the physical security of the systems configuration is standard, while also ensuring that the basic input-output system (BIOS) and the personal computer (PC) booting from CDs/DVDs, external devices and floppy drives in BIOS are rendered inoperative. Then, the auditor must ensure that the password is enabled in BIOS and that it also protects the GRand Unified Bootloader (GRUB) to ensure the restriction of physical access of the server.

In Linux or Unix-like systems, anyone can log in to the server in single-user mode using GRUB, as per the system configuration. Auditors must be certain that GRUB is protected with a strong password.

PROTECT GRUB USING PASSWORDS

To protect GRUB, administrators must use the strongest possible password and issue a command using a message-digest 5 (MD5) hash password:

```
[root@host-1[1]~]# grub-md5-crypt
```

After issuing the command, the administrator should open the /boot/grub/menu.lst or /boot/grub/grub.conf file and add the MD5 password:

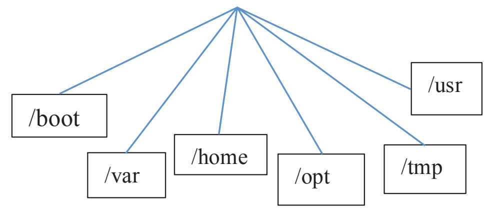
```
[root@ host-1  ~]# vi /boot/grub/menu.lst or
[root@ host-1  ~]# vi /boot/grub/grub.conf
```

The newly created MD5 password can then be added to the GRUB configuration file.

AUDITING DISK PARTITIONING IN THE AUDITED SYSTEM

In the system configuration, hard disk partitioning is critical. If any flaw exists in the partitioning, it will lead to data loss and possibly to disclosure, which could threaten the confidentiality of the data. During the audit, the auditor needs to examine and evaluate the different partitions in the audited server to ensure data security in case of any disaster. An administrator can group and separate the data among different partitions. This configuration ensures that only the data of that particular partition are lost if any unexpected accident occurs, despite the fact that the data on other partitions continue to exist. Auditors need to check that systems are configured in a way that allows separate partitions and ensure that third-party applications are installed on separate file systems. A secure directory structure is illustrated in **figure 1**.

Figure 1—Secure Directory Structure



Source: Muhammad Mushfiqur Rahman. Reprinted with permission.

AUDITING SERVERS FOR INSTALLED PACKAGES

It is recommended that when configuring the server, only the necessary packages should be installed. This ensures that the administrator may follow the standard configuration criteria of his/her organization and may scan the server using the Center for Internet Security Configuration Assessment Tool (CIS-CAT) and follow the recommendations of CIS-CAT. Unnecessary packages should not be installed into the system because such packages may create a vulnerability in the system. During the audit program, the auditor must evaluate and check the installed packages of the audited server to minimize the risk that compromising one service may lead to compromising other services. To ensure vulnerability minimization in the audited server, installed packages must be examined and unwanted installed services identified. Services that are running on run level 3 can be identified using the “chkconfig” command:

```
# /sbin/chkconfig --list |grep ‘3:on’
```

To examine all installed packages in a system, the following command can be used:

```
# sudo apt-get remove package-name
```

A sample script1 can be used to check the services running in the system:

```
#!/bin/bash
if (( $(ps -ef | grep -v grep | grep $service | wc -l) > 0 ))
then
echo “$service is running!!!”
else
/etc/init.d/$service start
Fi
```

AUDIT THE LISTENING PORTS

With the help of the Netstat networking command, all open ports and associated programs can be viewed. A sample script[2] is:

```
# netstat–tulpn
```

A script for port scanning is:

```
scan() {
  if [[ -z $1 || -z $2 ]]; then
    echo “Usage: $0 <host><port, ports, or port-range>”
    return
  fi

  local host=$1
  local ports=()
  case $2 in
    *-*)
      IFS=- read start end <<< “$2”
      for ((port=start; port <= end; port++)); do
        ports+=($port)
      done
      ;;
    *,*)
      IFS=, read -ra ports <<< “$2”
      ;;
    *)
      ports+=($2)
      ;;
esac
  for port in “${ports[@]}”; do
    alarm 1 “echo >/dev/tcp/$host/$port” &&
      echo “port $port is open” ||
      echo “port $port is closed”
  done
}
```

AUDIT REMOTE CONNECTIVITY OF THE AUDITED SERVER

Configuration of remote connectivity of the system in the network is penetrating and the remote protocol Telnet and rlogin is vulnerable because of the nonencrypted plaintext password and data transmission during the remote login. During the audit, the enabled remote connectivity services of the server should be checked, and the Secure Shell (SSH) protocol, which uses encryption technology during communication with the server should be examined. It is also necessary to check that the root login is disabled, the SSH port number is changed and the default port that is used by the audited server allows only specific authorized users access to the system. To do this, the auditor opens the main SSH configuration file and creates these parameters to restrict users’ access:

```
# vi /etc/ssh/sshd_config
```

DENYHOSTS AND FAIL2BAN

During the audit, the auditor should test the DenyHosts and Fail2ban feature. This is a log-based open-source intrusion prevention script used for SSH servers. This script is used by system administrators and users to monitor and analyze SSH server access logs for failed login attempts, known as dictionary-based attacks and brute-force attacks. In this script, the administrator can set the threshold for predefined failed logins from a specific Internet Protocol (IP) address and can ban the connection from specific IP addresses.

The features of DenyHosts include:

- Keeps and tracks logs from the /var/log/secure file, noting all successful and unsuccessful login attempts, and filters them.
- Regularly monitors the host as well as failed login attempts

- Sends email notification regarding blocked hosts and suspicious logins

The features of Fail2ban include:

- Keeps and tracks logs from /var/log/secure and /var/log/auth.log, /var/log/pwdfail
- Highly configurable and multithreaded
- Regularly monitors log files

AUDITING THE ROOT LOGIN STATUS

During the audit, the auditor should check whether Linux systems allow remote login using SSH for everyone with root user status. This configuration allows users with root user credentials to directly log in to the system. To protect the server from remote login, the root user administrator must disable the root access remotely. Systems can be saved by using the strongest passwords, but it is also recommended that administrators disable the root login from the remote connection and have a separate login ID. Another recommendation is that users use sudo to gain root access in the server.

AUDITING SSH PASSWORDLESS LOGIN

During the audit, the auditor should test the SSH passwordless login. Normally, system administrators use this feature for programmed backups, remotely executed required script, file transfers and remote script management, because it allows the administrator to perform these tasks without entering a password.

AUDITING THE SYSTEM FOR UPDATED PATCHES

Systems must be updated with the latest releases' patches, security fixes and kernels when those become available:

```
# yum updates
# yum check-update
```

AUDITING THE CRON JOBS STATUS

During audits, the auditor should check the built-in feature of cron jobs (cron) where it allows one to specify who may and who may not run jobs. This is controlled by the use of files called /etc/cron.allow and /etc/cron.deny. To lock a user using cron, usernames should be added to cron.deny. To allow a user to run cron, the user must be added to the cron.allow file. To disable all users from using cron, add the ALL line to the cron.deny file:

```
# echo ALL >>/etc/cron.deny
```
[3]

AUDITING THE STATUS OF USB DEVICES

During the audit, it is also important to examine and/or disable Universal Serial Bus (USB) devices. To mitigate data loss and control the spread of malware, users must be restricted from using USB devices in the systems.

AUDITING THE STATUS OF SELINUX

During audit, it is important to observe the status of Security-enhanced Linux (SELinux). It is an essential security mechanism for logical access control, which is provided in the kernel. This feature must be enabled in the system. Disabled SELinux demonstrates that the security mechanism has been deleted from the system.

The operations modes of SELinux include:

- **Enforcing**—This is the default mode of SELinux; it enforces the SELinux security policy in the machine.
- **Permissive**—This mode is used to troubleshoot SELinux-related issues; it tracks the log for each activity.
- **Disabled**—This mode speaks for itself and is not recommended.

During the audit, the auditor should use the following script to check the status of SELinux or use the system-config-selinux, getenforce or sestatus commands:

```
ENABLED=`cat /selinux/enforce`
if [ “$ENABLED” == 1 ]; then
  echo “SELinux is enabled, disable? (yes/no):”
  read disable
  if [ $disable == “yes” ]; then
    echo “disabling selinux”
setenforce 0
fi
fi
```

AUDITING THE IPV6 STATUS

During the audit, the auditor should check the activation and use of IPv6 in the system. If no one is using IPv6, it should be disabled in the system, because any unused service creates vulnerabilities for the system. During an audit, the auditor should check and confirm this. To do so, the auditor goes to the network configuration file and adds the following lines to disable IPv6:

```
# vi /etc/sysconfig/network

NETWORKING_IPV6=no
IPV6INIT=no
```

AUDITING EXISTING USER LISTS

The /etc/passwdfile stores users in Linux-based systems. To check existing users, the auditor should run the following script:

```
#!/bin/bash
# userslistinthesystem.sh

# count and Lists existing “real” users in the system.

echo
echo “[*] Existing users (sorted alphabetically):”
echo
grep ‘/bin/bash’ /etc/passwd | grep -v ‘root’ | cut -f1
-d’:’ | sort
echo

echo -n “[*] Number of real users found: “
grep ‘/bin/bash’ /etc/passwd | grep -v ‘root’ | wc -l
echo
```

AUDITING USER ACTIVITIES IN THE SYSTEM

During the audit, the auditor should check that audited systems are configured with psacct or acct. Both are open source applications for monitoring users’ activities in the system. Both psacct or acct applications run in the background and keep track of each user’s activity on the system, as well as what resources are being consumed. The auditor can use the following script[4] to audit user activities in the system:[5]

```
#!/usr/bin/envksh
last -Fa|awk ‘
    /wtmp begins/ { next; }
    /still logged in/ { next; }
    $0 == reboot { next; }

    NF > 0  {
        if( NR > 1 )
printf( “\n” );

printf( “      User:\t%s\n”, $1 );     # user
printf( “     Start:\t%s %s %s %s\n”, $3, $4, $5, $6 );
        if( $9 == “down” )
printf( “       End:\tshutdown\n” );
        else
printf( “       End:\t%s %s %s %s\n”, $9, $10, $11, $12 );

        if( substr( $NF, 1, 1 ) == “(“ )
        {
            t = $NF;
            h = “localhost”;
        }
        else
        {
            t = $(NF-1);
            h = $NF;
        }

gsub( “[()]”, “”, t );
printf( “   Time On:\t%s\n”, t );
printf( “Remote Host:\t%s\n”, h );
} ‘
```

Furthermore, during the audit, the auditor examines the documentation for the log retention policy of the organization to ensure compliance with the law and regulations of the organization and its regulatory body.

AUDITING USERS’ ABILITY TO USE OLD PASSWORDS

During the audit, it is important to check the configuration of password history in the system. It is recommended that administrators configure the system in a way so users are not able to revert to using old passwords when the password must be changed. The old password file is located at /etc/security/opasswd. This can be achieved through the following steps:[6]

- Open ‘/etc/pam.d/system-auth‘ file under RHEL:
  ```
  # vi /etc/pam.d/system-auth
  ```
- Open ‘/etc/pam.d/common-password‘ file under Ubuntu/Debian/Linux Mint:
  ```
  # vi /etc/pam.d/common-password
  ```
- Add the following line to ‘auth’ section:
  ```
  auth    sufficient    pam_unix.so likeauthnullok
  ```
- To disallow a user from reusing the last six passwords of his/hers, include the following line:
  ```
  Password   sufficient    pam_unix.so nullokuse_authtok md5
  shadow remember=6
  ```

After executing the command, the server stores the users’ previous six passwords, so if any user tries to update his/her password using any of his/her last six passwords, he/she will get an error message.

AUDITING THE STATUS OF USER PASSWORD EXPIRATION

During the audit, the auditor should check the configuration of the password expiration of users. In Linux systems, the /etc/shadow file stores users' passwords in an encrypted format. To check a user's password expiration, one can use the change command. This command results in detailed information regarding the password expiration date, as well as the date of change of the last password. Based on these details, the system will decide when a user must change his/her password.

The following command can be used to view existing users' information regarding the age of a password:

```
#chage -l username
```

Changes to password-aging of any user can be made with the following command:

```
#chage -M 60 username
#chage -M 60 -m 7 -W 7 userName
```

Parameters

The following parameters are used to set the password age in the system:

- Parameter -M is used to set password maximum age in days.
- Parameter -m is used to set password minimum age in days.
- Parameter -W is used to set the number of warnings in days.

AUDITING THE LOCK AND UNLOCK STATUS OF USER ACCOUNTS

During the audit, the auditor should check the list of locked and unlocked users. To examine this status, the following command can be used:

```
# passwd –s accountName
```

AUDITING PASSWORD STRENGTH IN THE SYSTEM

During the audit, the auditor should check the configuration of password strength to mitigate the risk from dictionary or brute-force attacks. System administrators must use pluggable authentication modules (PAM) to ensure that users set strong passwords.

The auditor can open the following file with an editor:

```
# vi /etc/pam.d/system-auth
```

AUDITING THE IPTABLES (FIREWALL) STATUS

During the audit, the auditor can check the configuration of the Linux firewall to prevent unauthorized access of the audited servers. To control the traffic, rules can be applied in iptables, which will filter incoming, outgoing and forwarding packets. Iptables can also allow and deny specific User Datagram Protocol/Transmission Control Protocol (UDP/TCP) port numbers.

AUDITING THE ACCOUNT FOR EMPTY PASSWORDS

During the audit, the auditor should check to identify any account having an empty password, which is prohibited and would allow anyone to access the system without entering a password. The auditor must check accounts for strong passwords and be certain that no one has any unauthorized access. Empty password accounts are a security risk and can be easily exploited by an attacker. Using the following command, one can determine the existence of accounts with empty passwords:

```
# cat /etc/shadow | awk -F: '($2=="“”"){print $1}'
```

AUDITING TIME STATISTICS OF USERS

Since organizations have a large number of users, they need to monitor the activities of users in the system, and, to do so, the auditor needs to ensure that the ac command is enabled in the system to review the activities of the users:

```
# ac
```

The command "ac -d" prints out the total login time in hours and by day:

```
# ac -d
```

The command to get the total login statistics time of user "isas" in hours is:

```
# ac isas
```

AUDITING THE LOG REVIEW STATUS

During the audit, check the logs and the frequency of the log review should also checked. As per the sensitivity of the data or based on business impact analysis (BIA), it is recommended that logs move in a dedicated log server. This may prevent intruders from easily modifying local logs. The common Linux default log file names and their usage, /var/log/message, include:[7]

1. /var/log/auth.log – Authentication logs.
2. /var/log/kern.log – Kernel logs.
3. /var/log/cron.log – Crond logs (cron job).
4. /var/log/maillog – Mail server logs.
5. /var/log/boot.log – System boot log.
6. /var/log/mysqld.log – MySQL database server log file.

7. /var/log/secure – Authentication log.
8. /var/log/utmp or /var/log/wtmp : Login records file.
9. /var/log/yum.log: Yum log files

AUDITING THE /BOOT DIRECTORY

During the audit, the auditor should check the status of the /boot directory. In Linux, kernel and its related files are placed in the /boot directory and auditors need to ensure that this folder is configured as read-only, which prevents unauthorized modification of the critical files in the Linux system. To ensure this configuration, the /etc/fstab file should be opened and the configuration checked:

```
# vi /etc/fstab
```

Then, the auditor should add the following line at the bottom, and save and close the file:

```
LABEL=/boot    /boot    ext2    defaults,ro    1 2
```

AUDITING INTERNET CONTROL MESSAGE PROTOCOL OR BROADCAST REQUEST

During the audit, the auditor should check that systems are configured in a way that ensures that the system ignores ping or broadcast requests, because excessive ping requests or broadcast echo replies slowdown the network and furthermore attackers may generate the denial-of-service (DoS)/distributed denial-of-service (DDoS) attack using the ICMP echo. To deny the ping or broadcast request, the following line should be added in the "/etc/sysctl.conf" file:[8]

```
Ignore ICMP request:
net.ipv4.icmp_echo_ignore_all = 1

Ignore Broadcast request:
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

New settings can be loaded by running following command:

```
#sysctl–p
```

AUDITING THE CONFIGURATION OF THE NTP SERVER

During the audit, it is important to check the status of the Network Time Protocol (NTP) because NTP is a client-server protocol and it uses the UDP 123. Time is critical in networked systems, and the system needs to identify and track each transaction and activity of users centrally to make them accountable for their activities with the data/information of the organization. To achieve this, the auditor must examine the enablement of NTP in the server and its configuration status. To check if NTP is configured to run at system start, the following command can be issued:

```
~]$ chkconfig --list ntpd
```

By default, when NTP is installed, it is configured to start at every system start.

To check if NTP is running, the following command can be issued:

```
~]$ ntpq -p
```

To obtain a brief status report from NTP, the following command can be issued:

```
~]$ ntpstat
```

VERIFY THE EXISTING STATUS OF NTP SERVER

The auditor should use the exit status of the NTP server to verify its operations:[9]

- Exit status 0 shows that the clock is synchronized.
- Exit status 1 shows that the clock is not synchronized.
- Exit status 2 shows that the clock state is indeterminant, e.g., if NTP cannot be contacted.

CONCLUSION

Assurance and auditing are the obligatory activities to secure the information of any organization. Auditing must be a continuous and ongoing process, no matter what system or provider is being used. The audit and assurance program needs to examine the system configuration and the status of information security on a periodic basis to avoid cyberattack. Because the operating system is a penetrating component in business, it is important to make sure that it is configured properly to ensure the security of business information.

A comprehensive, all-encompassing auditing solution that can easily accomplish each of the following at the operating system level must be implemented:

- Access and authentication auditing
- User and administrator auditing

- Suspicious activity auditing
- Vulnerability and threat auditing
- Change auditing

Without a sweeping auditing solution, organizations put critical information at risk. Corrupt, inaccurate or compromised data equal lost revenue, lost time, and compromised customer and employee relationships.

ENDNOTES

[1] Akamaras blog, *www.akamaras.com*

[2] Krumins, P.: *catonmat.com*

[3] Saive, R.; "25 Hardening Security Tips for Linux Servers," Techmint.com, 24 June 2013, *http://www.tecmint.com/linux-server-hardening-security-tips/*

[4] SK, "Monitoring Users Activity Using psacct or acct Tools in Linus," Unixmen, 11 May 2013, *www.unixmen.com/monitoring-users-activitiy-using-psacct-or-acct-tools-in-linux*

[5] Argoat.net, *http://argoat.net/Blog/?paged=20*

[6] *Op cit*, Saive

[7] *Ibid.*

[8] *Ibid.*

[9] Gile, Vivek; "How to: Verify My NTP Working or Not," nixCraft, 25 March 2010, *www.cyberciti.biz/faq/linus-unix-bsd-is-ntp-client-working.com*