PART II

Auditing Techniques

Guidance for Executing Test Steps

When reading the test steps provided in Part II, the reader should keep in mind some important guidelines.

One System vs. the Environment

With the exception of Chapters 3 through 5, these steps are written from the standpoint that a single system (such as a server, database, or application) is being audited. When multiple systems are being audited as part of one audit, most of these steps should be performed on each system. However, some steps, particularly those involving the review of processes (as opposed to system configuration), can likely be audited once for the whole environment without being repeated for each system. This assumes, of course, that the same processes are being applied throughout the environment. When auditing multiple systems at once, the auditor should use judgment and adjust accordingly.

Exercising Judgment

The auditor should also use good judgment in assessing the true risk associated with these steps based on the environment and on the overall security posture of the system. For example, in Chapter 8 the controls referenced in the "Network Security and Controls" and "Account Management" sections tend to be some of the most important, as they deal with controls that prevent someone from accessing the system when they're not authorized to do so. Other controls, such as those mentioned in the "Permissions Management" section, deal with controls used to prevent someone who's already on the system from accessing things they shouldn't and/or escalating their privileges. If you have a system sitting on your internal network, with network services locked down and with user accounts existing only for a small number of system administration personnel, the risk represented by some of the steps in that section is minimal. For example, file permissions become less important in that case, as you're confident that the only people accessing the system are those responsible for administering it. It would still be good to keep everything locked down as part of a defense-in-depth strategy, but you might decide not to push as hard for some of the lesser controls.

On the other hand, systems in your DMZ usually need to be completely hardened and locked down, with even the smallest of holes closed. Likewise, systems housing critical data need to be locked down more than systems used for trivial purposes. The point is that the auditor should not use the audit steps in this section as a mindless checklist, raising an audit issue every time there is an instance of noncompliance.

Leveraging Scripts

In many of the steps discussed in this part of the book, you'll see commands that will generate the needed output. In some cases, these will simply be shown as they would be entered from the command line. In other cases, the code is written as it would appear in a shell script. It can be highly advantageous and efficient to create an audit script that you can give to the system administrator to collect needed information. This script should usually be run with the privileges of an elevated account (such as root for Unix and Linux) and can both list the information you need to see to complete the audit steps and, in some cases, actually evaluate that information for you.

Protecting Audit Data

Take care to protect the data generated by the audit, which may contain sensitive items such as account information. Encrypting this data in transit is always a good idea, using GnuPG for e-mail, for example, or other tools.

CHAPTER 3

Auditing Entity-Level Controls

In this chapter we will discuss how to audit entity-level controls, which are pervasive across an organization. We will be discussing the auditing of *information technology* (IT) areas such as

- Strategic planning and technology roadmaps
- Performance indicators and metrics
- Project approval and monitoring processes
- Policies, standards, and procedures
- Employee management
- Asset and capacity management
- System configuration change management

Background

Because entity-level controls are pervasive across an organization, you can audit them once and feel confident that you have covered the topic for the whole company. This chapter discusses areas that the auditor should expect to see centralized in an organization. If the topics covered in this chapter are not centralized, or at least centrally coordinated, at your company, questions as to their overall effectiveness should arise. Most of these topics set the overall "tone at the top" for the IT organization and provide governance of the entire IT environment. If they are not centralized and/or standardized, the auditor should question the ability of the overall IT environment to be well controlled.

What is and is not considered an entity-level control is not always consistently defined and will vary by organization, depending on how the IT environment is defined. An area that is an entity-level process at one company will not necessarily be an entity-level process at another company. However, there's really no mystery to it—it all comes down to what is centralized and pervasive at your company. If a critical IT process is centralized, it is a good candidate for an entity-level controls review.

For example, Chapter 5 covers the topic of auditing data centers and areas such as physical security, environmental controls, system monitoring, and so on. Many companies have multiple decentralized data centers, meaning that these controls are not centralized for those companies. However, some companies have one data center and one set of processes for executing these areas, so physical security, environmental controls, and system monitoring would qualify as entity-level controls because they are centralized and pervasive. (However, such areas are not covered in this chapter, as they are covered in Chapter 5.) Auditors must use good judgment and knowledge of the company to determine what is and is not an entity-level control.

As mentioned earlier, the topics covered in this chapter should be centralized to a large degree because they provide for the core principles of IT governance. If these areas have no central coordination, the auditor should dig deep before signing off as to their effectiveness. Put another way, the areas covered in this chapter should be considered the minimum for an entity-level controls review. Other areas (such as data center operations) might be added based on the environment at your company.

NOTE Strong IT entity-level controls form a foundation for the IT control environment within a company. They demonstrate that IT management is serious about internal controls, risk management, and governance. A strong overall control environment and attitude that originates from the top tends to trickle down throughout the organization and leads to strong controls over decentralized processes and functions. Conversely, weak entity-level controls increase the likelihood that controls will be weak throughout the organization, because upper management has not demonstrated and communicated to the organization that internal controls are valued. This often leads to inconsistency at the lower levels, because the personalities and values of lower-level managers will be the sole determining factors in how seriously internal controls are taken within the organization.

It is critical for upper management to communicate and set the tone that internal controls, risk management, and governance are valued and will be rewarded. Without this message, departments are more likely to focus on cutting costs, managing their budgets, and meeting their schedules, with no consideration given to internal controls.

Test Steps for Auditing Entity-Level Controls

1. Review the overall IT organization structure to ensure that it provides for clear assignment of authority and responsibility over IT operations and that it provides for adequate segregation of duties.

A poorly defined IT organization structure can lead to confusion regarding responsibilities, causing IT support functions to be performed inefficiently or ineffectively. For example, critical functions may be either neglected or performed redundantly.

Also, if lines of authority are not clearly established, it can lead to disagreement as to who has the ultimate ability to make a final decision. Finally, if IT duties are not segregated appropriately, it could lead to fraudulent activities and affect the integrity of the company's information and processes.

How

A "one size fits all" model for an IT organization doesn't exist, and you can't mechanically use a checklist to determine whether your company's IT organization is adequate. Instead, you must view the overall organization and apply judgment

in determining whether it adequately addresses the most essential elements. With this in mind, the following discussion covers some key areas to consider during this review.

Review IT organization charts and ensure that they clearly indicate reporting structures. The organization charts should provide an indication as to where in the company the various IT organizations meet. For example, in most companies, all IT organizations eventually report to the chief information officer (CIO) so that one ultimate authority is able to set rules for the overall IT environment. Ensure that your company has IT organization reporting structures that eventually report to a single source that is "close enough" to day-to-day IT operations to allow for effective governance and direction setting. If the IT organizations report to multiple CIOs or consolidate only to a high-level executive such as the chief executive officer (CEO), additional processes will likely be needed to develop an effective method for establishing overall policies, priorities, and governance for IT at the company. Otherwise, it is likely that "fiefdoms" will exist within IT, preventing the establishment of true entity-level IT controls.

Review IT organization charts and charters and ensure that they clearly delineate areas of responsibility. Determine whether it is clear how responsibilities are divided between organizations, or evaluate whether there is significant opportunity for confusion and overlap. In addition to reviewing documented organization charts and charters, consider interviewing a sample of IT employees and customers to determine whether there is a consistent understanding of the division of responsibility.

Evaluate the division of responsibilities within the IT organization to ensure that duties are segregated appropriately. You also should consider criticality in making judgments. It is more important that separation of duties be in place over critical financial systems than over systems providing support for minor convenience functions (such as the company's internal training system).

NOTE The specifics of which duties should be segregated from others will vary by company; however, the general idea is that the responsibilities for initiating, authorizing, inputting, processing, and checking data should be segregated so that one person does not have the ability to create a fraudulent transaction, authorize it, and hide the evidence. In other words, you're attempting to prevent one person from being able to subvert a critical process.

Following are some basic general guidelines that can be considered during the review. Again, this should not be used as a mechanical checklist, and the auditor should review for compensating controls when investigating potential exceptions.

- **IT personnel should not perform data entry.** Keep in mind that IT organizations differ in their composition across companies, so some data-entry personnel may be classified as IT in their companies. In this case, we're referring to IT personnel who are performing true systems support.
- **Programmers and those performing run/maintain support for systems should not directly be able to modify production code, production data, or the job-scheduling structure.** As with all these statements, when a segregation-of-duties issue seems apparent, the auditor should look for compensating controls before determining whether it is a true issue. Access to production data and code may not be a large risk if strict accountability and change-control procedures are in support of that access.
- **Programmers and those performing run/maintain support for systems should be separate from those performing IT operations support (such as support for networks, data centers, operating systems, and so on).**
- **An information security organization should be responsible for setting policies and monitoring for compliance with those policies.** This information security organization should have no operational responsibilities outside of those related to information security.

2. Review the IT strategic planning process and ensure that it aligns with business strategies. Evaluate the IT organization's processes for monitoring progress against the strategic plan.

To provide for long-term effectiveness, the IT organization must have some sort of strategy regarding where it plans to go, as opposed to being in reactive mode constantly, where day-to-day issues and crises are the only considerations. The IT organization must be aware of upcoming business needs and changes in the environment so that it can plan and react accordingly. It is important that IT priorities align with business priorities. Too many IT organizations lose sight of the fact that their only reason for existence is to support the company in meeting its business objectives. Instead, these IT organizations focus on becoming a "world-class IT shop," even when this goal doesn't directly support the overall company objectives. It is critical for IT organizations to stay grounded by tying their objectives to the company's objectives.

How

Look for evidence of a strategic planning process within IT, and understand how that planning is performed. Determine how company strategies and priorities were used in developing the IT strategies and priorities. Review documented short- and long-term IT priorities. Evaluate processes in place for periodically monitoring for progress against those priorities and for reevaluating and updating those priorities.

3. Determine whether technology and application strategies and roadmaps exist, and evaluate processes for long-range technical planning.

IT is a rapidly changing environment, and it is important that IT organizations understand and plan for change. Otherwise, the company's IT environment runs the risk of becoming obsolete and/or not fully leveraging technology to benefit the company.

How

Look for evidence that long-term technical planning is being performed. For purchased applications and technologies, determine whether IT understands the vendor's support roadmap for those products. The IT organization should understand when their versions of the products will cease to be supported and create plans for either upgrading or replacing the products. Determine whether processes are in place to monitor for changes in relevant technologies, consider how those changes will affect the company, and look for opportunities to use new

technologies to help the company.

4. Review performance indicators and measurements for IT. Ensure that processes and metrics are in place (and approved by key stakeholders) for measuring performance of day-to-day activities and for tracking performance against service level agreements, budgets, and other operational requirements.

The IT organization exists to support the business and its day-to-day operations. If minimum standards of performance are not established and measured, it is difficult for the business to determine whether the IT organization's services are being performed at an acceptable level.

How

Obtain a copy of any metrics being captured for the IT organization's routine activities (such as system uptime and response time). Determine the goals for those metrics, and ensure that the appropriate stakeholders have approved those goals. If actual performance is significantly inferior to goals, determine whether root-cause analyses have been performed to understand the problem and whether plans are in place to solve the problem.

Review any SLAs (service level agreements) that have been established for supporting IT's key stakeholders. Ensure that processes are in place for measuring actual performance against the requirements of the SLA and for correcting any deviations.

Ensure that processes are in place for establishing budgets and for holding the IT organization accountable for meeting its budget. Obtain copies of the IT budget for the current and preceding years, as well as copies of any "budget versus actual" analyses. Determine how any significant variances were reported and resolved.

5. Review the IT organization's process for approving and prioritizing new projects. Determine whether this process is adequate for ensuring that system acquisition and development projects cannot commence without approval. Ensure that management and key stakeholders review project status, schedule, and budget periodically throughout the life of significant projects.

Without a structured process for approving and prioritizing new IT projects, IT resources probably will not be deployed efficiently. Instead, they will be assigned on an ad hoc basis to whatever potential project comes up next. Also, IT projects may commence that do not meet the needs of the business and/or that are not as important as other potential projects to which those resources could be deployed. Without a structured process whereby management and key stakeholders periodically review the project's progress, it is more likely that the project will get off track and fail to meet key goals and milestones.

How

Review any available documentation regarding the project proposal and approval process. Evaluate the process for potential holes that might allow a project to commence without approval. Look for evidence that proposed projects have been prioritized prior to approval and that some discipline and commonality exist within this approval process. Consider selecting a sample of active IT projects and obtaining evidence that those projects went through an appropriate process of proposal, prioritization, and approval. Review evidence that management and key stakeholders are periodically reviewing the status, schedule, and budget for active IT projects. Ensure that the project approval process calls for a thorough cost analysis before project commencement so that management can make an informed decision regarding expected return on investment (ROI) for the project. These cost analyses should consider not only the project start-up costs but also ongoing costs, such as software maintenance, hardware maintenance, support (labor) costs, power and cooling requirements for system hardware, and so on. This element is often omitted erroneously, leading to misinformed decisions. Start-up costs are only a fraction of the total ongoing costs for implementing a new system. A multiyear (five years is often a good target) total cost model should be developed as part of the initial project analysis.

6. Evaluate standards for governing the execution of IT projects and for ensuring the quality of products developed or acquired by the IT organization. Determine how these standards are communicated and enforced.

If standards are not in place and enforced in the IT environment, projects probably will be executed in an undisciplined fashion, quality issues will exist in developed or purchased products, and the IT environment will be unnecessarily diverse (leading to increased support costs and potential interface issues).

How

Determine whether documented standards govern areas such as the following. If so, review those standards and ensure that they are adequate.

- **Project management** See Chapter 17 for guidelines regarding key elements that should exist within project management standards.
- **Software development** Standards should exist governing the development of code, including standards for naming, revision history, comments, and calls to other programs. Without such standards, the time and effort required for one person to support and troubleshoot another person's code increase significantly. Note that depending on the size of the IT organization, it may be acceptable for programming standards to be decentralized to a degree. However, each significant development organization should have a set of standards. See Chapter 17 for guidelines regarding key elements that should exist within these standards.
- **System configuration** This would include standard configuration for laptops, desktops, servers, and common user software packages. Common configuration will help ensure that the systems are supportable and that they have the appropriate security settings.
- **Hardware and software** Standards should exist governing the hardware and software that are approved and supported for use in the company. This should include the specific versions that are supported. Otherwise, the IT environment likely will consist of a multitude of products performing similar functions, driving up IT support costs and leading to problems with the ability of the various products to interface with each other.
- **Quality assurance standards** Standards should exist that ensure that the

development process includes the evaluation of security risks and internal control requirements.

Look for evidence that these standards are communicated to all relevant IT employees, and determine how these standards are enforced.

NOTE Consider reviewing a sample of recent and active IT projects for evidence that the standards were followed. Consider reviewing a sample of systems for deviations from configuration, hardware, and software standards.

7. Review and evaluate risk-assessment processes in place for the IT organization.

Without these processes, the IT organization will be unaware of risks to the achievement of its objectives and therefore will not have the ability to make conscious decisions regarding whether to accept or mitigate those risks.

How

Some overlap exists between this step and some of the other steps mentioned in this chapter, many of which are designed to determine how the IT organization is evaluating its own risks. You might consider this step to be adequately covered without explicitly performing it. However, you should look for evidence that the IT organization is periodically considering the risks to the IT environment and making conscious decisions as to whether to accept, mitigate, or avoid those risks. Risk-assessment mechanisms could include the following:

- Monitoring internal controls in the IT environment, including internal audits and self-assessments
- Performing formal threat and risk assessments of critical data centers and systems
- Performing periodic reviews of the strategic IT plans and technical roadmaps and assessing risks to the achievement of those plans
- Monitoring compliance with information security policies and other relevant IT policies

8. Review and evaluate processes for ensuring that IT employees at the company have the skills and knowledge necessary to perform their jobs.

If employees in the IT organization are not qualified to perform their jobs, the quality of IT services will be poor. If mechanisms are not in place for maintaining and enhancing the knowledge and skills of IT employees, their knowledge can become outdated and obsolete.

How

Review human resources (HR) policies and processes as they relate to IT employees. Look for mechanisms that ensure that qualified people are hired and that provide for continuous enhancements of employee skills and knowledge. Review evidence that these policies and processes are followed. Here are some examples:

- Ensure that job descriptions exist for all IT positions and that the job descriptions specifically state the knowledge and skills required for each job. Review evidence that these job descriptions are referenced during the hiring process. Review processes for keeping the job descriptions up to date.
- Review the IT organization's training policies and ensure that they provide the opportunity for employees to attend training classes and seminars for enhancing and updating their skills and knowledge. Look for evidence that IT employees have individual training plans and/or evidence that they have taken training over the past year.
- Review performance-review processes. Look for evidence that IT employees are receiving regular feedback on their performance. Ensure that processes exist for identifying poor performers, coaching them, and moving them out of the organization if performance does not improve. Conversely, ensure that processes exist for identifying top performers, rewarding them, and providing them with incentives to remain at the company.

9. Ensure that effective processes exist for complying with applicable laws and regulations that affect IT and for maintaining awareness of changes in the regulatory environment.

If your company is found to be in violation of applicable laws and regulations (such as Health Insurance Portability and Accountability Act [HIPAA] and Sarbanes-Oxley), it could face stiff penalties and fines, a damaged reputation, lawsuits, and possibly cessation of the company. If a robust process is not in place for monitoring the regulatory environment, the company may be unaware of new laws and regulations, resulting in noncompliance.

How

Look for a single point of contact that is responsible for monitoring the regulatory environment and its impact on IT. This person or organization should be responsible for identifying laws and regulations that apply to the company's IT environment, ensuring that the responsibility for complying with those rules has been explicitly assigned to the appropriate organization(s), and monitoring the regulatory environment for additions and changes that will affect the company. If no single person or organization is responsible for this (or a small subset of people, each with a specific regulatory domain to cover), it likely will be done on an ad hoc basis, providing no assurance of full coverage. Review the processes used to monitor the regulatory environment, and evaluate their effectiveness. Obtain a list of IT-applicable regulations that have been identified, and look for evidence that responsibility for compliance with those regulations has been assigned and is being monitored. See Chapter 20 for more information on laws and regulations that may be applicable to your company.

10. Review and evaluate processes for ensuring that end users of the IT environment can report problems, are appropriately involved in IT decisions, and are satisfied with the services provided by IT.

Because the IT environment exists to support the company's employees in per-

forming their jobs, it is critical that processes exist whereby those employees can provide input into the quality of service they are receiving. Otherwise, the IT organization may be misaligned with its users and not be aware of it.

How

Ensure that a helpdesk function provides end users with the ability to report problems. Review and evaluate processes for capturing problems and ensuring that they are tracked to resolution. Obtain a list of recent tickets and select a sample, ensuring that all tickets were resolved and that no tickets were closed without the consent of the user who entered the ticket.

Ensure that a process exists for obtaining end-user feedback after tickets are closed. Look for evidence that user-satisfaction metrics are kept and that management follows up on end-user feedback.

To ensure that the helpdesk does not seek customer satisfaction at the expense of security, review policies and processes for obtaining proper approvals prior to responding to user requests for having passwords reset and for obtaining system access. Review a sample of these sorts of tickets, and ensure that proper processes were followed and approvals obtained.

Look for the existence of customer steering teams to provide input and prioritization of IT projects and enhancements. For significant areas of the business, key stakeholders should be identified to provide guidance to the IT organization regarding projects and decisions that affect them. Otherwise, the IT organization will be making decisions in a vacuum and likely will work on projects or enhancements that do not provide the greatest value for the business.

Review any SLAs that have been established for supporting IT's key stakeholders. Ensure that processes are in place for measuring actual performance against the requirements of the SLA and for correcting any deviations.

11. Review and evaluate processes for managing third-party services, ensuring that their roles and responsibilities are clearly defined and monitoring their performance.

Many companies outsource some or all of their IT support processes, including areas such as PC support, web server hosting, system support, programming, and so on. If these vendors are not managed appropriately, it can lead to poor service and unacceptable quality in the IT environment. Depending on what portion of the IT environment has been outsourced, these problems could significantly affect the company's operations.

How

Review the process for selecting vendors. Ensure that the process requires soliciting multiple competitive bids, the comparison of each vendor against predefined criteria, involvement of knowledgeable procurement personnel to help negotiate the contract, evaluation of the vendor's technical support capabilities and experience providing support for companies of similar size and industries as yours, performance of a thorough cost analysis, and investigation of each vendor's qualifications and financial health. For a sample of recent vendor selections, review evidence that the process was followed.

Ensure that contracts with third-party service providers specifically define the roles and responsibilities of the vendor and include defined SLAs. Review a sample of contracts for evidence that expectations have been specifically defined.

Ensure that contracts include nondisclosure clauses, preventing the vendor from disclosing company information. Also ensure that contracts include right-to-audit clauses that allow you to audit vendor activities that are critical to your company. Review a sample of contracts for evidence that these clauses are in place where applicable.

Review processes for monitoring the performance and providing oversight of existing third-party service providers. For a sample of existing vendors, look for evidence that they are being monitored for compliance with SLAs and that they are performing the responsibilities defined in the contract.

See Chapter 16 for more details on auditing outsourced operations.

12. Review and evaluate processes for controlling nonemployee logical access.

Most companies employ some level of outsourcing and contract labor to supplement their internal workforce. Also, some companies allow third-party vendors a degree of logical access to purchased systems for troubleshooting and support purposes. Because these personnel are not employees of the company, they are less likely to have a personal investment in the company's success or an awareness of the company's policies and culture. If their access to company information assets is not governed, and if expectations regarding their use of that access are not communicated, it is more likely that company information assets will be exposed unnecessarily or misused.

How

Ensure that policies require approval and sponsorship from an employee prior to a nonemployee obtaining logical access to company systems. If feasible, obtain a sample of nonemployee accounts and validate that they have appropriate approval and sponsorship.

Review and evaluate processes for communicating company policies (including IT security policies) to nonemployees prior to granting them system access. Look for evidence that this communication has taken place. For example, if all nonemployees are required to sign a statement that they have read and agree to the policies, pull a sample of nonemployees and obtain copies of these agreements.

Review and evaluate processes for removing logical access from nonemployees when they have ceased to work with your company or otherwise no longer need access. Consider obtaining a sample of current nonemployee accounts and validating that those nonemployees are still working with your company and still have a need for their current level of access.

Ensure that nondisclosure agreements (NDAs) are signed by nonemployees to legally protect your company from inappropriate use of company data. Pull a sample of nonemployee accounts and obtain a copy of the NDAs for them.

Ensure that consideration has been given to identifying data that should not be accessed by nonemployees and activities that should not be performed by nonemployees. For example, your company may decide that access to certain levels of financial data should never be granted to nonemployees. Or it may decide that nonemployees should never be granted system administration duties. The answer will depend on your company's industry and philosophies; however, an evaluation process should take place, and the results of that evaluation should be documented in company policy and enforced. This evaluation should be part of the data classification effort described in Chapter 4 and should drive the restric-

tions on nonemployee logical access.

13. Review and evaluate processes for ensuring that the company is in compliance with applicable software licenses.

Using software illegally can lead to penalties, fines, and lawsuits. It is increasingly easy for company employees to download software from the Internet. If companies do not develop processes for preventing or tracking such activity (as well as tracking the use of company licenses for purchased software), they can find themselves subject to software vendor audits without the ability to account properly for the company's use of the vendor's software.

How

Look for evidence that the company maintains a list of enterprise software licenses (such as for Microsoft Office, ERP application accounts, and so on) and has developed a process for monitoring use of those licenses and complying with the terms of agreement. Determine how decentralized (nonenterprise) licenses are monitored and tracked. This would include software purchased by employees and placed on their company computers, as well as software downloaded from the Internet. Truly comprehensive software asset management requires a centralized database that contains information on exactly what software the company has the right to use (licenses purchased) and on exactly what software is being used in the environment (licenses used) and can compare the two. Test the effectiveness of the method used at your company either by performing your own scans on a sample of computers or by reviewing evidence from the company's processes.

14. Review and evaluate controls over remote access into the company's network (such as VPN and dedicated external connections).

Allowing remote access to a network basically results in that network being extended beyond its normal confines, bypassing normal perimeter controls such as firewalls. A lack of strong controls regarding this access can result in inappropriate access to the network and a compromised network.

How

Ensure that strong authentication (e.g., multifactor authentication) is required for remote access and that these credentials are transmitted over secure (such as encrypted) communication channels. Question any remote authentication schemes that require only an ID and password. IDs and passwords can and will be compromised, so they alone are not enough for verifying the identify of a user. Multifactor authentication requires at least two factors, such as a password plus a physical or virtual token, in order to authenticate, reducing the risk posed by a compromised password.

Determine whether approval processes are in place for granting remote access, especially for nonemployees. Pull a sample of users with remote access and look for evidence of approval. Also evaluate processes for removing remote access accounts when employees leave the company. Pull a sample of users with remote access and ensure that they are still active employees.

Evaluate controls for ensuring that dedicated external connections to business partners are removed when no longer needed. Work with the appropriate IT organization (for example, the network team) to pull a sample of current connections and, by means of interviews and documentation review, determine whether they are still legitimately necessary.

Evaluate controls for ensuring that unauthorized connections cannot be made to the network and/or for detecting them if they are. Evaluate controls for ensuring that unauthorized connection points cannot be placed on the network and/or for detecting them if they are.

Ensure that policies provide minimum security requirements that should be met by all machines accessing the network remotely. This should include requirements for operating system patch level and antimalware protection. Look for preventive or detective controls that enforce these requirements.

Ensure that machines that are remotely accessing the network are not permitted to be dual-homed, which would bridge networks. This should be enforced technically where possible and by explicit agreement otherwise.

15. Ensure that hiring and termination procedures are clear and comprehensive.

Hiring procedures ensure that employees are submitted to drug screens and background checks, where local laws permit, prior to beginning work within an organization. Termination procedures ensure that access to company systems and facilities is revoked before a disgruntled employee can cause damage and that company property is returned. Inadequate hiring or termination procedures would expose the company to sabotage or abuse of privileges that could result in an information security compromise.

How

Review HR policies and procedures for the hiring and termination of employees. Ensure that hiring procedures include background checks, drug screens, and confidentiality agreements. Ensure that termination procedures include physical and logical access revocation, return of company-owned equipment, and, where appropriate, supervision while the former employee collects his or her belongings.

16. Review and evaluate policies and procedures for controlling the procurement and movement of hardware.

Asset management is the controlling, tracking, and reporting of organizational assets to facilitate accounting for the assets. Without effective asset management, the company will be subject to the increased expense of duplicate equipment in situations where equipment is available but unaccounted for. The company will also be subject to unnecessary lease expenses if leased equipment is not adequately tracked and returned on time. Similarly, without adequate asset management, end-of-life equipment conditions may not be noted, resulting in increased risk of hardware failure. Additionally, theft of equipment that is not tracked likely would go unnoticed. In the context of this step, the assets being referred to are computer hardware, such as desktops, laptops, servers, and so on.

How

Review and evaluate the company's asset management policies and procedures, and ensure that they encompass the following:

- **Asset procurement process** Ensure that this process requires appropriate approvals prior to the purchase of hardware.
- **Asset tracking** Ensure that the company is using asset tags and has an

asset management database.

- **Current inventory of all equipment** Ensure that an inventory contains the asset number and location of all hardware, along with information about the equipment's warranty status, lease expiration, and overall life cycle (that is, when it is no longer eligible for vendor support). Ensure that an effective mechanism is in place for keeping this inventory up to date. A sample of asset tags also should be inspected visibly and tied back to the inventory.
- **Asset move and disposal procedures** Ensure that unused equipment is stored in a secure manner. Also ensure that data is erased properly from equipment prior to its disposal.

17. Ensure that system configurations are controlled with change management to avoid unnecessary system outages.

Configuration change management ensures that system changes are controlled and tracked to reduce the risk of system outages. It includes planning, scheduling, applying, and tracking changes to systems for the purpose of reducing the risk of those changes to the environment.

How

Change activities can affect two areas: hardware and software (including operating system–level changes). Ensure that the configuration management procedures include processes for the following:

- Requesting changes (including processes for end users to request changes)
- Determining the specifics of what should change
- Prioritizing and approving proposed changes
- Scheduling approved changes
- Testing and approving changes prior to implementation
- Communicating planned changes prior to implementation
- Implementing changes
- Rolling back (removing) changes that don't work as expected after implementation

Also review change-control documentation to verify that changes are fully documented, approved, and tracked. Approvals should incorporate a risk assessment and typically are granted by a committee made up of stakeholders. You should be able to obtain a sample of change-control requests, as well as other configuration management documentation, from IT management.

18. Ensure that media transportation, storage, reuse, and disposal are addressed adequately by company-wide policies and procedures.

Media controls ensure that information on data-storage media remains confidential and is protected from premature deterioration or destruction. Inadequate media transportation, storage, reuse, and disposal policies and procedures expose organizations to possible unauthorized disclosure or destruction of critical information. One increasingly common type of security incident is the loss of backup media in transit by third-party carriers. A number of high-profile companies have fallen victim to this threat in recent years and have incurred losses due to legal actions, reputation damage, and incident response costs.

How

Computer media, including but not limited to backup tapes, CDs and DVDs, hard disks, and USB drives, must be strictly controlled to ensure data privacy. Since backup operators, computer technicians, system administrators, third-party carriers, and even end users handle storage media, media policies and procedures should address these disparate roles. When auditing media control policies and procedures, look for the following:

- Requirements for sensitive information to be encrypted prior to transporting it through a third-party carrier
- Requirements for magnetic media to be digitally shredded or degaussed prior to reuse or disposal
- Requirements for optical and paper media to be physically shredded prior to disposal
- Requirements for users to be trained adequately on how to store and dispose of computer media, including removable media such as USB drives
- Requirements for computer media to be stored in a physically secure, temperature-controlled, and dry location to prevent damage to the media

You can obtain this information through the review of IT policies, procedures, and security awareness training documents, as well as user interviews.

19. Verify that capacity monitoring and planning are addressed adequately by company policies and procedures.

Anticipating and monitoring the capacity of data center facilities, computer systems, and applications are critical parts of ensuring system availability. When companies neglect these controls, they often experience system outages and data loss.

How

Review for the following:

- Selected architecture documents to ensure that systems and facilities are designed to anticipated capacity requirements
- System monitoring procedures, paying particular attention to capacity thresholds
- System monitoring logs to determine the percentage of systems that are approaching or exceeding capacity thresholds
- System availability reports to ensure that system capacity issues are not causing undue downtime

Since capacity management is addressed most often by the groups responsible for data centers, applications, or system management, specific procedures should be addressed within these areas.

20. Review and evaluate the company's identity and access management processes.

In practically every chapter of this book, you will find audit steps on this topic. As you evaluate each individual system and technology, it is important to understand how access to it is controlled. However, while it is possible that every system in your environment will have its own individual accounts and account management processes, hopefully there will be some level of centralization. Otherwise, you will be relying on each individual system to implement appropriate controls covering typical account tasks such as account creation, password management, and account deletion. Also, without centralized account management, users may be forced to track multiple IDs and passwords, making it more likely that they will write their passwords down and store them in an easy-to-find location. While just about every technology has its own native accounts and passwords, most also provide for some sort of ability to reference or sync with a centralized directory and authentication mechanism. This is often referred to as "federating" identities—where one system trusts the authentication coming from another system.

An enterprise identity and access management process will increase security (by allowing for centralized controls) and will also increase efficiency (by eliminating duplicate efforts).

How

Review for the existence of "enterprise" accounts. These are accounts (identities) that can be used across multiple systems and environments. Review the enterprise account processes for the following controls:

- Procedures for creating accounts and ensuring that each account is associated with and can be traced to a specific individual.
- Processes for ensuring accounts are removed or disabled in a timely fashion in the event of employee termination. Terminating an account in the central directory should result in a cascade effect, where that account is removed or disabled in all systems that subscribe to the central directory.
- Processes for suspending access to individual systems in the event of a job change within the company (or otherwise requiring revalidation of that access).

Determine what company systems are tied into the enterprise identity and access management process. Evaluate processes for identifying and prioritizing systems to be included in the enterprise identity and access management process.

If the identity and access management process includes a centralized authentication mechanism, verify that appropriate password and authentication controls are in place. Review password settings (such as password composition requirements and password aging rules) for appropriateness and for compliance with your company's policies. Review the need for and existence of stronger forms of authentication (e.g., two-factor authentication).

Again, each individual chapter in this section addresses this topic in terms of the specific technology or topic under review. The purpose of this step is to understand and evaluate to what extent these processes are centrally managed and controlled.

21. Review and evaluate the elements of the company's cybersecurity program.

Cybersecurity has become a critical concern for all companies and their ability to maintain the confidentiality, integrity, and availability of their information and processes.

How

This entity-level control is important enough that it has been separated into its own chapter. See Chapter 4 for details.

22. Based on the structure of your company's IT organization and processes, identify and audit other entity-level IT processes.

By identifying those baseline IT controls, you should be able to reduce testing during other audits and avoid repetition. For example, if your company has only one production data center, you can test the physical security and environmental controls of that data center once. Then, as you perform audits of individual systems that are housed in that data center, instead of auditing the physical security and environmental controls for each of those systems (which would be very repetitive because they're all in the same place), you can just reference your entity-level audit of those topics and move on. Also, by performing audits of centralized processes, you will have an understanding of potential compensating controls in the overall IT environment that may mitigate concerns you have with lower-level controls.

NOTE If a critical IT process at your company is centralized, it is a good candidate for being reviewed during an entity-level controls audit. By auditing it once at the company level, you will be able to rely on the results of that audit when performing audits of other IT systems and processes.

How

Review the topics covered in the other chapters in Part II of this book, and consider whether any of those areas are centralized at your company. Those topics are candidates for an entity-level controls review. Here are some likely candidates:

- Data center physical security and environmental controls (see Chapter 5)
- System monitoring (such as performance and availability) and incident reporting (see Chapter 5)
- Disaster recovery planning (see Chapter 5)
- Backup processes (see Chapter 5)
- Network security and management (see Chapter 6)
- Windows system administration processes (such as account management and security monitoring) (see Chapter 7)
- Security of baselines used for deployment of new Windows systems (see Chapter 7)
- Malware protection (such as antivirus, patching, and compliance checking) (see Chapters 7 and 14)
- Unix/Linux system administration processes (such as account management, security monitoring, and security patching) (see Chapter 8)
- Security of baselines used for deployment of new Unix and Linux systems (see Chapter 8)

- Software change controls for internally developed code (see Chapter 15)

Knowledge Base

As mentioned throughout this chapter, the specifics of entity-level controls will vary from company to company. However, the best general sources of information on IT-specific entity-level controls can be found on the Information Systems Audit and Control Association (ISACA) website (www.isaca.org), where details on the control objectives for information and related technology (COBIT) framework and guidelines for Sarbanes-Oxley IT compliance testing are available. In addition, general guidelines on entity-level controls (not specific to IT) and links to resources related to the popular Committee of Sponsoring Organizations (COSO) model of internal controls can be found on the website for the Institute of Internal Auditors (IIA) at www.theiia.org. Finally, your external auditors likely will have some published guidelines to share with you on this topic.

Master Checklist

The following table summarizes the steps listed herein for auditing entity-level controls.

Auditing Entity-Level Controls

Checklist for Auditing Entity-Level Controls

- ❑ 1. Review the overall IT organization structure to ensure that it provides for clear assignment of authority and responsibility over IT operations and that it provides for adequate segregation of duties.
- ❑ 2. Review the IT strategic planning process and ensure that it aligns with business strategies. Evaluate the IT organization's processes for monitoring progress against the strategic plan.
- ❑ 3. Determine whether technology and application strategies and roadmaps exist, and evaluate processes for long-range technical planning.
- ❑ 4. Review performance indicators and measurements for IT. Ensure that processes and metrics are in place (and approved by key stakeholders) for measuring performance of day-to-day activities and for tracking performance against service level agreements, budgets, and other operational requirements.
- ❑ 5. Review the IT organization's process for approving and prioritizing new projects. Determine whether this process is adequate for ensuring that system acquisition and development projects cannot commence without approval. Ensure that management and key stakeholders review project status, schedule, and budget periodically throughout the life of significant projects.
- ❑ 6. Evaluate standards for governing the execution of IT projects and for ensuring the quality of products developed or acquired by the IT organization. Determine how these standards are communicated and enforced.
- ❑ 7. Review and evaluate risk-assessment processes in place for the IT organization.
- ❑ 8. Review and evaluate processes for ensuring that IT employees at the company have the skills and knowledge necessary to perform their jobs.
- ❑ 9. Ensure that effective processes exist for complying with applicable laws and regulations that affect IT and for maintaining awareness of changes in the regulatory environment.
- ❑ 10. Review and evaluate processes for ensuring that end users of the IT environment can report problems, are appropriately involved in IT decisions, and are satisfied with the services provided by IT.
- ❑ 11. Review and evaluate processes for managing third-party services, ensuring that their roles and responsibilities are clearly defined and monitoring their performance.
- ❑ 12. Review and evaluate processes for controlling nonemployee logical access.
- ❑ 13. Review and evaluate processes for ensuring that the company is in compliance with applicable software licenses.
- ❑ 14. Review and evaluate controls over remote access into the company's network (such as VPN and dedicated external connections).
- ❑ 15. Ensure that hiring and termination procedures are clear and comprehensive.
- ❑ 16. Review and evaluate policies and procedures for controlling the procurement and movement of hardware.
- ❑ 17. Ensure that system configurations are controlled with change management to avoid unnecessary system outages.
- ❑ 18. Ensure that media transportation, storage, reuse, and disposal are addressed adequately by company-wide policies and procedures.
- ❑ 19. Verify that capacity monitoring and planning are addressed adequately by company policies and procedures.
- ❑ 20. Review and evaluate the company's identity and access management processes.
- ❑ 21. Review and evaluate the elements of the company's cybersecurity program.
- ❑ 22. Based on the structure of your company's IT organization and processes, identify and audit other entity-level IT processes.

CHAPTER

4

Auditing Cybersecurity Programs

High-profile incursions against technology and defense firms, breaches of credit card information, thefts of personal data—all of these have increased the awareness of security issues among boards of directors, executives, and others charged with making their companies successful. Globally, regulations dealing with the protection of data and systems have proliferated, with Payment Card Industry (PCI) standards, the European Union General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), and other requirements forcing companies to improve their security posture or face penalties or fines. Defending firms against cyber attacks and ensuring compliance with security and privacy regulations require training, vigilance, and a number of technical and procedural controls that aren't second nature to most people in an organization. In response to these and other concerns, companies around the world have increased their investments in security.

While larger or more mature businesses have had security programs for many years, over the last decade almost every company of any appreciable size has either created a security program or increased its attention to existing programs. As with any large investment, companies want to know that their efforts are meeting the needs of the organization. This is where the auditor comes in. This chapter provides an overview of considerations for auditing cybersecurity programs. Some of the topics covered here include

- Scope and structure of cybersecurity programs
- Organizational oversight and governance
- Common functions and services of security programs
- Specific technical and procedural items to review

Background

As an auditor, some of the first things you'll need to understand are the size, scope, and purpose of the firm's cybersecurity program. Cybersecurity programs take many different forms. In small companies, the entire security team might reside in one person, whose day job is to help with IT issues when not answering the phones. Very large organizations might have hundreds of security personnel spread across many functions, and some may have as many external security service providers as they have employees. While the size of a company's security team is often directly proportional to the revenue or profitability of a company, other

factors can also influence the size and scope of security programs.

Retail and financial organizations must comply with a range of regulations related to the kinds of information and assets they process. Retailers usually handle credit cards, and must comply with PCI rules in order to offer that service to their customers. Banks and brokerages must comply with a range of governmental rules. In addition to compliance with the different laws and standards, these kinds of businesses must protect themselves from reputational damage, as they don't want their customers to go to competitors because of security flaws. Retailers and financial organizations often have security programs focused heavily on compliance, and their budgets can be quite high.

Technology companies, or those dealing with the design and production of sophisticated equipment, such as for defense or aerospace industries, may have a completely different set of security concerns. They may be most intent on protecting their intellectual property from theft by competitors, insiders, or external groups. Since they may not have to comply with specific regulations, their security programs are dictated more by their internal risk appetites. More risk-tolerant companies may have leaner security teams, while more risk-averse businesses may have larger teams.

Manufacturing firms may also have intellectual property to protect but might primarily be concerned with avoiding cybersecurity issues that would affect their production lines. Depending on how the company is structured, responsibility for the security of production floors might reside within the factory teams themselves, or for larger companies, shared with a central security team. As with intellectual property firms, the scope of a security program is often related to the risk appetite of the business.

Credit agencies, nonprofits, governments, healthcare providers, and others might be targeted for the personal information they store and process. Some of the largest breaches to date have involved personal information. Personal data can be thought of like intellectual property but is often more regulated. Entities that deal heavily in personal data can incur fines and other costs when that data is stolen, so protecting it may be a high priority compared to other needs.

As in the previous chapter, there's no "one size fits all" approach. You must consider the overall organization and its risk posture in order to properly assess the program. Knowing the business needs and how they inform the company security program will help you evaluate the strength of a program and determine how to regard potential gaps. You may have to complete all of the steps discussed for contextual understanding before outlining strengths and improvement opportunities for each item.

Steps for Auditing Cybersecurity Programs

1. Assess the placement of the cybersecurity program within the overall organization and ensure appropriate oversight.

Most organizations ultimately identify someone to be responsible for information security for the company. This person is often designated as the chief information security officer (CISO), but could be identified as the director of information security, IT security manager, or might have a completely unrelated title, depending on the size and maturity of the team. In some companies, the chief information officer (CIO) is responsible for information security. Organizational reporting structures and lines of responsibility can vary greatly. This step helps set the stage for the rest of the audit by identifying the people responsible for information security and how they relate to the business at large.

How

Review organization charts to identify the information security team and management. The information security program is often part of the IT function, and the security manager may report to the CIO, but this is not always the case. In some larger organizations or those with higher cyber-related risk concerns, the CISO may report to the CEO or to another executive. Ensure that a clear reporting structure exists such that information security resources are guided through a single management chain or process. This provides the security group with clear objectives.

Interview the CISO or equivalent, or a delegate, to understand the level of oversight provided to the information security team under this structure. In many cases this will be through executive committees or regular presentations to company boards of directors. In some cases this oversight may come from the CIO or another high-ranking person in the company. It is important that the activities of the information security team fall under the guidance of the company's overall management structure in some way to ensure alignment with business actions.

For example, you could be part of a firm with a CISO, who reports to the CIO, who reports to the executive steering committee, who reports to the board of directors. On a periodic basis, the executive steering committee may meet with the CIO and/or CISO to review the state of information security, and the board of directors may be kept informed.

Among companies with boards, it's common for the CIO or CISO to report at least annually to the board of directors on the state of the information security program.

Again, there's no single "correct" structure. The key objective of this step is to understand who is responsible for cybersecurity and what corporate oversight is in place to advise the program.

2. Assess the information-related risk management processes of the organization and evaluate how cybersecurity risks are identified and managed.

For most businesses, information security is much like insurance. Security rarely adds value to whatever the company's business is; it often serves primarily to reduce risk. In order to reduce risk, you must first identify various risks and determine how your organization will respond to each. Managing risk in alignment with business goals is a primary function of the security organization. Without sound risk management processes, the information security organization will not be able to justify investment decisions related to security improvements.

How

Many very large and complex risk management processes exist, and many organizations follow them to the letter. If this is the case, you may only need to review the outputs of these processes to see that the firm is assessing and identifying risks adequately. More often though, risk processes are less formal. In any case, you should be able to look for evidence that the information security team is considering cybersecurity risks facing the company and making appropriate decisions in response. Some artifacts or processes that may demonstrate this include

- Periodic, formal threat and risk assessments for critical systems
- Third-party testing of security controls and correction of any identified deficiencies
- Compliance programs and monitoring of internal controls
- Strategic planning processes that prioritize initiatives based on risk or value
- Corporate-level risk planning processes

Some questions you may want to consider around this step include

- How are risks identified, and how does the team align on how to address them?
- What role do business leaders and other stakeholders play in the decision process?
- Are cybersecurity threats considered in the overall organization risk discussions?

All organizations should have a clear understanding of their cybersecurity risks and a process by which to prioritize security investments.

3. Evaluate the scope of the cybersecurity program and its relationship to other IT functions within the organization.

As discussed earlier, cybersecurity programs can differ greatly in size and scope. As an auditor, you'll need to determine how the security needs of the organization—defined in part by its risk posture—are met by the structure of the information security program. Regardless of organization, though, some common functions should be present in most information security teams, whether it's a one-person show or an entire security department.

How

Obtain organization charts or other artifacts describing the makeup and function of the security team. Interview functional leaders to gather more information.

A security program should be expected to cover a minimal set of practices in some form or fashion, including

- **Policy and compliance management** Defining security guidelines for the company
- **Awareness** Getting relevant security information into the hands of people who may need it
- **Vulnerability management** Helping the organization understand the risks and criticality of potential exploits and assisting with remediation
- **Security monitoring** Collecting log and alert data and detecting potential security events in the environment
- **Incident response** Dealing with viruses, breaches, or other malicious activities and helping to return the business to a normal state

In smaller organizations, the security team may also be responsible for managing operational aspects of security, such as firewall management, web or e-mail security, client or endpoint security, access control, remote access, authentication, and more. In larger organizations, these may be handled by a separate operations team or by a team that dual-reports to the security director.

As organizations grow, they may add other security functions, such as security architecture or penetration testing.

It is also important to ascertain whether separation-of-duties risks exist due to the organizational structure or makeup. For example, when a system administrator is also a security administrator, there is a higher risk that security controls may be bypassed or reconfigured for expediency.

4. Review the security policy and compliance functions of the organization, ensuring that IT security policies exist and provide adequate requirements for the security of the environment. Determine how those policies are communicated and how compliance is monitored and enforced.

IT security policy sets a baseline of expectations for employees of the company. If policies don't exist or provide adequate coverage, employees are forced to make up their own rules regarding security-related issues. The same concept extends to computer systems, which require a standard by which system security can be evaluated. If IT security policies are too lenient, they will not provide adequate protection of the company's information assets. If they are too strict, they either will be ignored or will place unnecessary overhead and costs on the business.

If the IT security policies aren't communicated to employees, they won't be followed. Additionally, if compliance with those policies is not monitored and enforced, employees will learn quickly that the policies can be ignored with no consequences, causing the policies to become "suggestions" rather than requirements.

How

Verify Adequate Policy Coverage Obtain a copy of your company's IT security policies. Ensure that they adequately cover your company's IT environment. At a minimum, the policies should include coverage of the following areas:

- Acceptable usage of the company's information assets by employees (for example, whether employees can use their computers, the Internet, and e-mail for personal reasons)
- Data classification, retention, and destruction
- Remote connectivity (for example, overall network security and security requirements for virtual private network [VPN] and other forms of connection to external parties)
- Passwords
- Server security (such as security requirements for Unix and Windows servers)
- Client security (such as security requirements for desktops and laptops)
- Logical access (such as requirements for obtaining and granting access to systems)

Review the policies for adequacy based on industry standards and the specific needs of your company. The audit steps in the other chapters of Part II can be used as guidelines.

Specifically review the company's password policy. It should provide adequate guidelines dictating requirements for the composition of company passwords (for example, minimum of eight characters, combination of letters and numbers, difficult to guess, and so on), for aging company passwords (such as requiring that they be changed every 90 days), for locking accounts after a certain number of unsuccessful logon attempts, for timing out login sessions after a period of inactivity, and for retaining a password history so that previous passwords cannot be reused for a certain period.

Specifically review the company's logical access policy. It should provide adequate guidelines dictating requirements for every user to have a unique ID, for accounts to be suspended upon employee termination or job change, and for users to be granted the minimum access necessary to perform their jobs.

Verify Stakeholder Buy-In Ensure that key stakeholders were included during policy creation. Obtain a list of employees involved in the creation and approval of the IT security policies, such as IT organizations that are expected to comply with the policy. If IT security policies are created in a vacuum by the IT security organization without involving others, they are likely to be viewed as unrealistic and will be ignored. Involvement from those who provide the day-to-day support of the IT environment will bring an important perspective to the policies and also will ensure buy-in from those who need to enforce and comply with the policies. Ensure that the IT security policies were approved by an executive, such as the CIO or CEO. This will provide the IT organization with the authority and backing necessary to enforce the policies.

Verify Processes Around the Policies Review processes for periodically reviewing and updating the policies to ensure that they keep up with the ever-changing IT environment. Look for evidence that these processes have been executed.

Review processes for periodically evaluating changes in the environment that might necessitate the development of new policies. Look for evidence that these processes have been executed.

Ensure that provisions have been made for obtaining approved exemptions from the policy. There inevitably will be occasions when people do not think that they can comply with the policy. A defined process should be in place whereby those people can formally request an exemption from the policy. They should be required to state why they need an exemption and define the compensating controls that will be put in place. The IT security organization should facilitate the exception process, including providing a recommendation and an opinion on the risk presented by the request, but they usually should avoid making the final decision as to whether or not to accept the risk. Instead, it should be a business decision. Review the escalation policy for the exemption process and ensure that business (as opposed to IT) management is involved at some point, at least for the acceptance of significant risks. Ensure that the final decisions are documented and retained.

NOTE Look for evidence that the IT security policies are communicated adequately to all company employees. Potential vectors include referencing the policies during new-hire orientation and/or having all employees periodically sign a statement that they have read and agree to the policies.

Review processes implemented by IT security and other IT organizations for monitoring compliance with the policies. Ensure that enforcement and escalation processes are in place that result in the correction of noncompliant situations. Review a sample of recent applicable compliance-monitoring reports, and ensure that significant issues were tracked to resolution.

5. Review the awareness and communications functions of the security team, reviewing methods to train employees on security risks and concerns.

Employees starting a new job will usually receive training about the expectations for the role and appropriate methods to do the job. When this training includes security information, employees can be better equipped to avoid security risks in their daily work, improving the overall security posture of the organization. An effective security awareness program educates employees on security concerns at a level and in a manner appropriate to their job functions. Most awareness programs will include several key elements; these should be assessed for their presence and effectiveness.

How

Discuss the scope of the security awareness program with the individual in charge of that function. You should expect the following primary elements in a complete program:

- General security training for new employees
- Periodic security training for current employees
- Ongoing general security awareness
- Role-specific security training for designated functions (for example, software developers)

Depending on the industry, size, and maturity of the organization, you may also find

- Education about nuisance or malicious e-mail, including unsolicited commercial e-mail (usually called "spam") and social engineering messages (usually termed "phishing")
- Exercises to test employee ability to detect phishing e-mails
- Training about specific data types, such as intellectual property or personally identifiable information (PII)

Review the processes for providing general security training for new hires. Review the content to ensure that basics are covered, such as employee expectations around security, information on company security policies, key areas of concern or risk for the company, how to report security issues, and so on. Sample records from training systems or orientations to ensure that employees attended or reviewed security material.

Ensure that training is provided on a periodic basis. Depending on risk level and industry requirements, organizations may be required to provide security training very frequently, but many companies mandate annual or biennial security training for current employees. Discuss how training is assigned, and review processes for ensuring that training is completed. This may involve reminders to employees

or escalations to supervisors. Sample training records to ensure employees have taken required security training.

Discuss the ongoing security awareness program. Awareness teams usually provide periodic updates to employees on various security topics. These could be very general, focusing on good security hygiene like password protection, or very specific, focusing on a new type of phishing message, for example. Ongoing security awareness may be delivered in various ways. Organizations may use posters, mass e-mail, web content, social media, or in-person presentations to raise security awareness. Obtain samples of this content as evidence of ongoing awareness efforts.

Specific roles in the organization, or individuals dealing with certain kinds of data or systems, may require additional training beyond the general information provided to all employees. For example, IT personnel may have access to more data on employees because they support those systems; additional training for IT personnel should be considered for these cases. Software developers should receive training on how to recognize and avoid software vulnerabilities in the code they write. Employees who handle PII or other personal data should be trained on any specific requirements for those data types. Any employee who comes in contact with the company's sensitive intellectual property should be trained on how to recognize and handle company data. Review the timing and content of these training programs. As with other periodic training programs discussed earlier, annual or biennial training should be considered the minimum standard for role-specific or data-specific training.

More mature security programs may also include phishing training and phishing exercises or assessments. As phishing and other forms of social engineering have grown, companies have responded by educating employees on how to recognize and avoid falling victim to phishing. If your organization has a phishing awareness program, discuss the parameters of the program with the team. This may include

- Content and timing of phishing awareness training for employees
- Phishing exercises (tests) to assess the effectiveness of awareness training
- Metrics or escalation processes associated with phishing exercises
- Phishing exercises tailored for high-risk groups such as finance, HR, or IT
- Methods for employees to report suspected phishing messages

Review the results of phishing exercises over time, if applicable. An effective program should see improved results as time goes by. Companies vary in how they handle the results of phishing exercises; discuss any escalation processes with the awareness team to understand how "failed" exercises are addressed. Ensure that any disciplinary action taken as a result of phishing exercises is handled in accordance with company policies.

6. Review the vulnerability management function of the organization, ensuring that the team is aware of emerging threats and vulnerabilities and has processes to identify at-risk systems in the environment.

If an organization takes no action to maintain its security posture on an ongoing basis, its risk over time will increase. Thousands of vulnerabilities are discovered in common software every year; what was thought to be secure a month ago is often known to be less secure today due to the ongoing discovery of security vulnerabilities. Malicious actors are always looking for an edge, and companies must have active vulnerability programs so that they can be aware of new threats and potential exploits, as well as ways to reduce risk.

How

Review the organization's approach in finding and resolving vulnerabilities. An effective program should have a few key elements, discussed next.

Awareness of new threats and vulnerabilities Interview the individual or team in charge of vulnerability management. Assess how the team becomes aware of newly discovered vulnerabilities in products common to the environment, such as operating systems or web browsers. Most commonly, security teams will subscribe to one or more feeds from organizations that distribute vulnerability information. For example, the United States Computer Emergency Readiness Team (US-CERT) publishes vulnerability information as part of its National Cyber Awareness System. Some organizations may use automated feeds from organizations like the National Institute of Standards and Technology (NIST). In addition, most major software companies provide methods for advising customers of security vulnerabilities in their products. The team should have a method of receiving and consuming new vulnerability data relevant to the products known to be in the environment.

Besides receiving information on known vulnerabilities, security teams should also receive information on emerging threats. These often come in the form of subscriptions to third-party threat intelligence services. This content can include information on how new vulnerabilities are being exploited, attack methods and other exploit tactics seen in the world, and general security news. Determine through interviews with the security team whether an external threat intelligence service is used and how the information is received and integrated with other processes. For example, some intelligence feeds can be integrated with monitoring systems to enhance detection capabilities. It's important for security teams to be aware of emerging threats in order to take proactive steps to protect company assets.

Vulnerability scanning and other methods to identify known vulnerabilities in the environment Vulnerability scanning can be a very effective control to ensure that patching procedures, firewall settings, and other security processes are providing the intended level of protection. Scanning can also be used to identify vulnerabilities that have not yet been mitigated. Common vulnerability scanners, like those available from Qualys, Rapid7, or Tenable, can identify applications on a system, find services answering on open network ports, and more. The system state is correlated with a known list of vulnerabilities to produce a report of potential system risks. All organizations should use a scanning tool or a third-party scanning service. If your organization doesn't have one, you may want to explore a free trial of one of the commercial tools listed earlier or consider an open-source tool like OpenVAS.

NOTE Some types of vulnerability scanners can also scan software for errors that lead to security flaws. These tools are often used with web-based applications and

will assess how code is written or how web content is constructed to identify application-level risks. A comprehensive vulnerability management program should include both system-level and application-level vulnerability scanning. This section primarily discusses system-level scanning.

Discuss the vulnerability scanning program with the security team. Scanning programs often have two primary objectives:

- Ensure that newly deployed systems are free of known vulnerabilities at the time of deployment
- Ensure the ongoing security of deployed systems by checking for vulnerabilities on a periodic basis

Your company should have a vulnerability scanning policy that lists timing requirements for scans, as well as the criteria for remediating issues discovered in a scan. For example, your policy may require that Internet-facing systems be scanned on a monthly basis and any vulnerability rated as "Medium" or higher must be resolved within seven days of discovery. The timing and criticality criteria will vary by company and risk tolerance, but some level of periodic scanning and risk remediation should take place.

Review scan records for both new systems and existing systems to ensure that systems are being scanned in line with company policy. A system may be scanned multiple times in the process of identifying and resolving vulnerabilities; a final, "clean" scan is an indication that all relevant vulnerabilities were mitigated.

Evaluation of discovered vulnerabilities Vulnerabilities published by software makers or by groups like US-CERT, as well as vulnerabilities identified by scanners, are usually assigned a severity rating, such as Low, Medium, High, or Critical. Organizations may decide to accept the external rating and work accordingly or employ a process to evaluate the risk based on their own situations. As most companies deploy multiple layers of protection in a defense-in-depth strategy, some vulnerabilities rated as Critical may not be as important once the full scope of defenses is taken into account.

Discuss with the vulnerability team whether published severity levels are evaluated independently of the external rating. Review the criteria for those decisions, and examine evidence of this activity. Common criteria include whether an exploit can occur remotely, whether there is evidence of active exploits in the world against a published vulnerability, whether a patch or other remediation is available, and the type of negative action that occurs as a result of the exploit.

Many organizations track metrics comparing the published severity level to the "judged" level. This can be particularly important for prioritization of remediation activity. A vulnerability judged to be Critical in nature might drive urgent action across the company, but if judged lower, might fall under routine maintenance or patching.

Processes to communicate and track results and actions Discuss how the vulnerabilities are resolved. In some organizations, the vulnerability management team may be in charge of mitigating discovered vulnerabilities. Other organizations may use an outsourced service. Some will turn over mitigation to the application teams responsible for each server. Whatever the case, there should be a clear process in place to scan systems and provide results to the appropriate teams.

Discuss how vulnerabilities are tracked to closure. Most organizations will track vulnerability status over time in order to ensure that teams take action as directed. Obtain relevant metrics as evidence. Review what happens when a team is not able to resolve a vulnerability within the specified time frame.

7. Assess the security monitoring function of the security team, reviewing log collection and alert processing and detection capabilities.

Organizations frequently task a dedicated individual or team with monitoring the environment for adverse security events. Whatever the organizational structure, the company should have people, processes, and technology in place to review system data and identify security issues. The auditor should ensure that the monitoring group has the appropriate systems available and is taking necessary action to monitor the environment and protect the company.

How

Review the processes and technologies to collect and correlate system log data and alerts. Discuss the monitoring system with members of the security operations center (SOC) team. You should be able to identify the key systems and interfaces used by the team.

A key technical component of most SOCs is a security information and event manager (SIEM), which collects and correlates log data from many sources to find events of interest. A SIEM usually stores log data from different sources in the environment, such as firewalls, proxies, antivirus systems, authentication servers, and more. This data is transmitted from remote systems and stored centrally. Review the SIEM policies to assess the amount of data stored, the sources used, and the retention time. While log sources can vary widely, a common data set will include the types of systems mentioned earlier. Retention time is dependent on the amount of storage available, but six to twelve months is a good reference point. SIEMs can be used to monitor for unusual input in near real time or can be used for forensic investigation, such as incident reconstruction.

Discuss with the SOC team how log data is protected from unauthorized access or manipulation. Access to read log data from the SIEM should be restricted to need-to-know individuals only; no one should have access or the ability to modify log data once stored in the SIEM. Ask the SOC to provide information on access controls for the log repository.

SIEMs and other tools in the SOC can be used to correlate data for the purposes of identifying unusual activity. Review the process for developing alerts from correlated data. Security events often happen right along with "normal" events, so it's important for a SOC to develop alerts at a proper level. The team should have an ongoing process to improve the fidelity of alert systems and should be able to provide evidence based on previous events that justifies why a threshold or alert level was set in a particular way. In addition, if the team subscribes to a threat intelligence service, determine how that information is integrated with the SIEM or other detection systems to improve monitoring.

You should also discuss how actions or alerts are tracked to resolution. Some companies use an incident tracking system to manage alerts and security incidents. Others may track issues manually in spreadsheets. Request a sample of incidents from the system in use; you should be able to see an alert being tracked to closure. If a rule in the SIEM or other system is generating alerts and no one is taking action, you should question why that alert is being generated or why nothing is being done in response.

Outside of SIEMs and other tools in use by the SOC to detect events, employees may also notice suspicious activity. Ensure that a mechanism exists for employees to report security incidents or concerns and that those reports are tracked to resolution. Review a sample of recently reported incidents, and determine whether they were resolved adequately.

8. Assess the incident response function of the security team, ensuring that the organization is able to respond effectively to various kinds of security events.

Almost every alert, virus detection, or phishing report generates some kind of response. In many companies, the severity of the incident dictates the type of response. When a significant security event occurs, the team may invoke a formal incident response (IR) process. This ensures that the right steps are followed and the right people within the company are notified of the situation. As an auditor, you should review the IR function to ensure that a documented process exists and will be followed should a security event occur.

How

Request a copy of the IR process. If the team does not have a documented IR process, you should try to assess how the organization would handle a larger-scale security event. Most organizations are well prepared to respond to a nuisance-level virus, for example, but if a ransomware incident affected half of the organization's systems and shut down business applications, how would the security team respond?

An effective IR plan should include the following components:

- Criteria for invoking different levels of response.
- Identification of key roles and responsibilities in the event of an incident. For example, an incident manager should be identified who would be "in charge" of the response process.
- Decision points for contacting other levels of the organization, including security management, IT management, business teams, legal advisors, communications teams, and others.
- Guidelines for the preferred course of action when dealing with certain event types. For example, incidents involving retail systems may drive a different response than incidents involving facilities systems. It's preferable to document these scenarios in advance where possible rather than make decisions during the stress of an actual incident.

The plan should have been reviewed by security and IT management as well as other stakeholders, such as legal teams or risk officers.

9. Assess other functions of the security team as appropriate.

Larger or more mature organizations or those in specific industries may have other elements in their security programs. If those elements are not being executed with appropriate controls, they might not meet the objectives for which they were created.

How

Some example functions are listed next; you can use basic auditing principles, including interviews and gathering documentation, to determine if the function meets its stated objectives for the company. Security architecture teams often serve as consultants for other areas of IT or for the other security teams. Frequently having deep technical expertise in one or more areas, security architects have a broad understanding of security principles and can apply this knowledge to many problems. Most commonly, security architects will participate in or drive major security-related initiatives for the company or will engage with other IT teams to ensure security concerns are properly addressed in project work.

Security infrastructure teams manage the devices and software responsible for providing much of the technical security for the company. These might include firewalls, web gateways, remote access systems, multifactor authentication platforms, authentication systems, identity and access management software, and more. Auditing many of these technologies is covered elsewhere in this book.

Security teams also have projects of their own and as a result may have project management activities within their scope. Sound project management practices are as important to the security effort as they are elsewhere in IT organizations. The elements in Chapter 17 on auditing company projects also apply to project teams within the security department.

10. Review and evaluate policies and processes for assigning ownership of company data, classifying the data, protecting the data in accordance with their classification, and defining the data's life cycle.

Although IT is responsible for providing the technology and mechanisms for protecting company data, a framework must be in place for making decisions as to what level of protection is necessary for any given data element (based on the criticality of the data). Without such a framework, there will be inconsistency in how data is protected, likely resulting in some data being underprotected (thereby placing critical information assets at risk) or overprotected (leading to unnecessary costs). If the life cycle of data is not defined, it will lead to data being retained longer than necessary (resulting in additional storage costs and possible legal liabilities) or being destroyed prematurely (leading to potential operational, legal, or tax issues).

How

Review the company's data classification policy. It should have provisions for identifying owners for all critical company data. It also should provide a framework for classifying that data based on its criticality (for example, confidential, internal data, public data). This framework should provide specific definitions of each classification level, along with specific requirements for how data at each level should be protected (for example, encryption).

Review evidence that the data classification policy has been implemented. Look for a list of data owners and documentation indicating that those owners have classified their data. For a sample of this data, review evidence that protection has been implemented in alignment with the classification.

Determine whether life cycle information has been created for company data. For a sample of major data elements, review documentation of the data's life cycle requirements, including retention, archive, and destruction requirements. Ideally, requirements will be identified for how long the data should be active (online, easily accessible, modifiable if appropriate, and backed up periodically), when and for how long they should be archived (possibly offline, not necessarily easy to ac-

cess, no longer modifiable, and no longer backed up periodically), and when they should be destroyed.

Review evidence that life cycle requirements have been implemented.

11. Determine how security policies and security risk are handled in organizational IT processes.

A firm can have the best security team with the best policies and a great awareness and vulnerability program, but if the organization-at-large doesn't account for security concerns in broader processes, security problems may arise that could have been avoided. As an auditor, you should ensure that various functions of the business, particularly in the IT space, understand security requirements and involve the security team when questions arise.

How

Discuss IT processes with members of your IT department's operations or development teams. Ask how security issues are addressed during project planning, and determine whether the security team is involved in aspects of IT project work. If there is a formal project approval process, determine if the information security team participates in the approvals.

Vulnerability management and software patching go hand in hand, and teams that are responsible for system administration or software development should be able to demonstrate via patch schedules that software vulnerabilities are being mitigated. Software development teams should also use security scanning tools to identify potential vulnerabilities during software creation.

If your company has a security architecture function, determine how it engages with the broader portfolio of IT efforts. Security architects can help address potential security concerns early in a project and ensure that strategic security concerns are taken into account.

In general, each area of the organization should have an understanding of security policies and should be able to articulate how those policies are being addressed in their processes.

As you review Chapter 17 on auditing company projects, consider how cybersecurity concerns are addressed as part of project management.

12. Review and evaluate processes for ensuring that security personnel have the skills and knowledge necessary for performing their jobs.

As discussed in the previous chapter, if employees are not qualified to perform their jobs, the quality of the work they do will be poor. This is no different for the information security team. If mechanisms are not in place for maintaining and enhancing the knowledge and skills of security personnel, their knowledge can become outdated and obsolete.

How

Review human resources (HR) policies and processes as they relate to the security team. These may be the same as processes covered for IT in the previous chapter. Look for mechanisms that ensure that qualified people are hired and that provide for continuous enhancements of employee skills and knowledge. Review evidence that these policies and processes are followed. Here are some examples:

- Ensure that job descriptions exist for all positions and that the job descriptions specifically state the knowledge and skills required for each job. Review evidence that these job descriptions are referenced during the hiring process. Review processes for keeping the job descriptions up to date.
- Review the security team's training policies and ensure that they provide the opportunity for employees to attend training classes and seminars for enhancing and updating their skills and knowledge. Look for evidence that employees have taken training over the past year.
- Review performance-review processes. Look for evidence that employees are receiving regular feedback on their performance. Ensure that processes exist for identifying poor performers, coaching them, and moving them out of the organization if performance does not improve. Conversely, ensure that processes exist for identifying top performers, rewarding them, and providing them with incentives to remain at the company.

13. Assess that metrics are collected commensurate with the goals of the security program and that metrics are reported to appropriate management personnel.

Performance-based organizations are fond of saying "you get what you measure." Metrics help an organization track various aspects of its performance, which can help drive decisions about investments in people, processes, or technologies. Metrics can take a number of different forms, from simple instrumentation (for example, how many e-mails are blocked as spam in a week) to maturity or capability (for example, mean time to remediate known vulnerabilities). Security teams should track metrics to improve their own operations but should also consider metrics to share with stakeholders. These may not always be the same items. As an auditor, you should identify the metrics that are collected, with whom they are shared, and what the organization does with them.

How

Obtain metrics from the owners of the various security services the organization provides. As you review the metrics, you should consider whether the organization takes action based on the metrics, or if the item is for instrumentation purposes only. While there is much debate in security circles about the effectiveness of different metrics, some you might commonly find include

- Number of security incidents
- Mean time to detect/mean time to resolve incidents
- Alerts from various security systems, such as antivirus software, data loss prevention systems, network firewalls, web application firewalls, and others
- Number of exceptions to security policies
- Percentage of systems patched/unpatched
- Number of vulnerabilities found/resolved/unresolved
- Expenses of the security program

If no or few metrics are collected, you will want to understand whether there is a technical limitation, or whether the team views those metrics as not helpful or interesting.

Metrics can also take the form of maturity measurements against external assessments or external standards. For example, if an organization employs an implementation standard such as NIST 800-53, metrics could show a percentage of compliance to that standard. Similar metrics could be created against the Center for Internet Security's Critical Security Controls, standards like ISO 27001, the NIST Cybersecurity Framework, and others. Metrics of this type can help identify gaps in a program and prioritize investment areas.

For any metric, it's important to understand who is consuming the data. If a measurement is not being reviewed or acted upon, it has very little value to the organization. Metrics that are of interest to system administrators will be different from those of interest to senior executives, but the organization should be able to provide metrics suitable for different audiences. If metrics are not being presented to management, an auditor should ask how the value of the program is understood by those managers.

14. Review processes around the use of managed security service providers (MSSPs) within the security team.

As discussed in Chapter 3, many companies outsource various IT support processes, and this can include parts of the security program. If these vendors are not selected and managed appropriately, the service may not meet the needs of the organization. Depending on what portions of the security function have been outsourced, these problems could reduce the effectiveness of the security team and increase company risk.

MSSPs commonly provide monitoring services for specific elements of a program. For example, an MSSP might receive all employee reports of spam or phishing and may perform a triage function to identify those that need additional attention. Some organizations may outsource their entire security operations center to an MSSP. Smaller organizations may leverage MSSPs to gain specific expertise, while larger organizations may use them to offload some functions and allow internal resources to focus on other efforts.

The audit steps for this item are also covered in Chapter 3 but are listed here for completeness.

How

Review the process for selecting vendors. Ensure that the process requires soliciting multiple competitive bids, the comparison of each vendor against predefined criteria, involvement of knowledgeable procurement personnel to help negotiate the contract, evaluation of the vendor's technical support capabilities and experience providing support for companies of similar size and industries as yours, performance of a thorough cost analysis, and investigation of each vendor's qualifications and financial health. For a sample of recent vendor selections, review evidence that the process was followed.

Ensure that contracts with third-party service providers specifically define the roles and responsibilities of the vendor and include defined service level agreements (SLAs). For security services, this may include delivery of metrics or reports related to the service. Review a sample of contracts for evidence that expectations have been specifically defined.

Ensure that contracts include nondisclosure clauses, preventing the vendor from disclosing company information. While this may be critical in many areas of the company, a disclosure of security-related information can be damaging, as sensitive vulnerability data or investigation-related information could be used maliciously by outsiders. Also ensure that contracts include right-to-audit clauses that allow you to audit vendor activities that are critical to your company. Review a sample of contracts for evidence that these clauses are in place where applicable.

Review processes for monitoring the performance and providing oversight of existing third-party service providers. For a sample of existing vendors, look for evidence that they are being monitored for compliance with SLAs and that they are performing the responsibilities defined in the contract.

15. Determine how the organization ensures that its security controls are effective.

When protecting its information assets, a business may consider hundreds of different security controls. Even the most secure organizations in the world don't implement every possible defense or control; whether due to cost, complexity, or a trade-off on usability or employee acceptance of a control, a certain set of controls is implemented. From the steps discussed earlier, you should be able to determine both how the organization sees risk and how it prioritizes mitigations. But how does a group know that what it has implemented will be effective in meeting its security needs?

Independent assessments, including attestations or certifications, external audits, incident response exercises, and penetration testing, can examine the effectiveness of controls and provide visibility to gaps.

Independent attestations or certifications may involve visits by external audit teams and may relate to widely known evaluations such as Statement on Standards for Attestation Engagements (SSAE) or the International Organization for Standardization (ISO). Within these families are many areas for control evaluation, but the most relevant for these purposes are the ISO 27000 series, dealing with information security management systems, and the System and Organization Controls (SOC) attestations of SSAE. In either case, a review of the security function and the various controls in place is performed by an external team. In the case of SSAE/SOC, some types of reports include evaluating the operational effectiveness of security controls.

External audits will include many of the same elements defined in this book. Depending on the scope and purpose of such an audit, the assessment could include detailed control evaluations. In an ideal case, an internal auditor and an external auditor would find the same sets of controls and potential gaps. Having an external audit provides an additional layer of independence to assure management that the audit is free from undue influence and represents an accurate picture of the control landscape.

Incident response exercises are intended to test the ability of the incident response team to handle various types of events by evaluating processes used by that team. An incident response exercise should involve various members of the response team as well as individuals or teams that might be affected by a major security event, such as legal teams, HR or privacy, operational teams, communications, and others as appropriate. This type of exercise can ensure that teams communicate adequately and that response processes are properly documented and followed.

Penetration testing is probably the best way to evaluate the security controls of an organization. Penetration tests, or "pen" tests, provide assurance that a control can withstand an effort by a determined individual or team to defeat that control. A pen test might involve social engineering, physical probing, vulnerability

scanning, use of known or unknown exploits, and more. These tests are usually performed by skilled individuals under specific rules of engagement determined by the parties involved. Pen tests may be conducted against a specific target or for a certain goal (evaluating a particular high-value application), or may be more general in nature (assessing perimeter defenses).

An auditor should evaluate the presence, scope, and frequency of independent assessments, as well as the processes used to handle the results of those evaluations.

How

Determine through interviews of the CISO or delegates which types of independent assessments, if any, are used by the organization to evaluate its security program. Examine the artifacts produced by these efforts to ensure they include actual testing of controls.

Review how the results are processed and resolved. The results of any external evaluation should include recommendations on how the organization can improve. While these are only recommendations, the organization should have some process to consider the information and determine how to proceed. One common practice is to review the results with appropriate management teams and align on which recommendations will be addressed. Actions resulting from these discussions should be tracked to closure.

Knowledge Base

As mentioned throughout this chapter, the composition and scope of information security programs will vary from company to company. However, many reference sources are available on security policy models, management structures, and more. Information on security program best practices is available at sources including SANS (www.sans.org), the Information Systems Audit and Control Association (ISACA, www.isaca.org), and the Institute of Internal Auditors (IIA, www.theiia.org). The ISO publishes standards (available for a fee), including ISO 27001, that describe information security management systems. Finally, your external auditors likely will have some information to share with you on this topic.

The following table lists various resources where you can find more information about the topics in this chapter.

Resource	Website
Information Systems Audit and Control Association	www.isaca.org
Institute of Internal Auditors	www.theiia.org
National Vulnerability Database	https://nvd.nist.gov
United States Computer Emergency Readiness Team (US-CERT)	www.us-cert.gov
SANS	www.sans.org
Common Vulnerabilities and Exposures (CVE)	cve.mitre.org
National Institute of Standards and Technology Cybersecurity Framework (NIST CSF)	www.nist.gov/cyberframework
International Organization for Standardization (ISO) 27001	www.iso.org/isoiec-27001-information-security.html
The Center for Internet Security	www.cisecurity.org
Computer Security Resource Center	https://csrc.nist.gov

Master Checklist

The following table summarizes the steps listed herein for auditing cybersecurity programs.

Auditing Cybersecurity Programs

Checklist for Auditing Cybersecurity Programs	
❑	1. Assess the placement of the cybersecurity program within the overall organization and ensure appropriate oversight.
❑	2. Assess the information-related risk management processes of the organization and evaluate how cybersecurity risks are identified and managed.
❑	3. Evaluate the scope of the cybersecurity program and its relationship to other IT functions within the organization.
❑	4. Review the security policy and compliance functions of the organization, ensuring that IT security policies exist and provide adequate requirements for the security of the environment. Determine how those policies are communicated and how compliance is monitored and enforced.
❑	5. Review the awareness and communications functions of the security team, reviewing methods to train employees on security risks and concerns.
❑	6. Review the vulnerability management function of the organization, ensuring that the team is aware of emerging threats and vulnerabilities and has processes to identify at-risk systems in the environment.
❑	7. Assess the security monitoring function of the security team, reviewing log collection and alert processing and detection capabilities.
❑	8. Assess the incident response function of the security team, ensuring that the organization is able to respond effectively to various kinds of security events.
❑	9. Assess other functions of the security team as appropriate.
❑	10. Review and evaluate policies and processes for assigning ownership of company data, classifying the data, protecting the data in accordance with their classification, and defining the data's life cycle.
❑	11. Determine how security policies and security risk are handled in organizational IT processes.
❑	12. Review and evaluate processes for ensuring that security personnel have the skills and knowledge necessary for performing their jobs.
❑	13. Assess that metrics are collected commensurate with the goals of the security program and that metrics are reported to appropriate management personnel.
❑	14. Review processes around the use of managed security service providers (MSSPs) within the security team.
❑	15. Determine how the organization ensures that its security controls are effective.

CHAPTER 5

Auditing Data Centers and Disaster Recovery

Information technology (IT) processing facilities, usually referred to as data centers, are at the core of most modern organizations' operations, supporting almost all critical business activities. In this chapter we will discuss the steps for auditing data center controls, including the following areas:

- Physical security and environmental controls
- Data center operations
- System and site resiliency
- Disaster preparedness

Background

Ever since the first general-purpose electronic computer (the Electronic Numerical Integrator and Computer, or ENIAC) was created in 1946, computer systems have had specific environmental, power, and physical security requirements. Beginning in the late 1950s, as mainframe computers became more widely available, data centers were created for the express purpose of meeting these requirements. Now, many organizations have their own data centers or co-locate their systems in a shared facility, although more and more companies are using the cloud for services that were previously hosted in dedicated or co-located data centers (see Chapter 16 for further discussion of this concept).

Today's data centers provide physical access control infrastructure, environmental controls, power and network connectivity, fire suppression systems, and alarm systems. This data center infrastructure is designed to maintain a constant optimal computing environment. The auditor's role is to verify and validate that all of the necessary systems and procedures are present and working properly to protect the confidentiality, integrity, and availability of the company's systems and data.

Data Center Auditing Essentials

A data center is a facility that is designed to house an organization's critical systems, which comprise computer hardware, operating systems, and applications. Applications are leveraged to support specific business processes such as order fulfillment, customer relationship management (CRM), and accounting. Figure 5-1 shows the relationships among data center facilities, system platforms, databases, applications, and business processes.

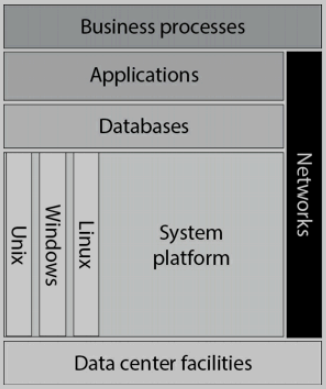


Figure 5-1 Data-processing hierarchy

As you can see, data center facilities are at the foundation of the hierarchy, which is why it is so important that they have the necessary controls to mitigate risk. Major data center threats include the following:

- Natural threats, such as weather events, flooding, earthquakes, and fire
- Manmade threats, such as terrorist incidents, riots, theft, and sabotage
- Environmental hazards, such as extreme temperatures and humidity
- Loss of utilities, such as electrical power and telecommunications

You may notice that most of these threats are physical in nature. In this age of advanced technology, it is easy to forget the importance of physical controls and focus your energy on logical controls. However, even with excellent logical access controls in place, these physical threats can compromise your systems' security and availability.

For those who have not worked in a data center environment, data centers can be a little overwhelming. Particularly in large environments and co-located facilities, data center access might be experienced through intimidating man-traps (doors specifically designed to allow only one person through at a time), physical guards, biometric readers, and card-key-access authentication systems.

Once you pass into the computing environment, you should notice racks of computer systems sitting on a raised floor. Most of the time, miles of power and network cables are run beneath the raised floor, although many data centers run cables through open conduits that hang from the ceiling. You also will notice generators, large power conditioners, and UPS (uninterruptible power supply) devices or rooms filled with batteries to ensure that clean, uninterrupted power is available at all times. Most data centers have industrial-strength heating, ventilation, and air conditioning systems to maintain optimal temperature and humidity levels within the facility.

The brain of the data center facility is the control center. It usually consists of a series of consoles and computer monitors that are used to monitor temperature, humidity levels, power consumption, alarms, and critical system status. Many times, if the control center is actually physically located within the data center, the

control center and tape operations may be the only areas consistently manned by data center personnel.

For the purpose of the data center audit, we will explore physical security and environmental controls; system and site resiliency controls; policies, plans, and procedures used in governing data center operations; and controls that enable disaster preparedness.

Physical Security and Environmental Controls

Data centers incorporate several types of facility-based controls, commonly referred to as physical security and environmental controls, including facility access control systems, alarm systems, and fire suppression systems. These systems are designed to prevent unauthorized intrusion, detect problems before they cause damage, and prevent the spread of fire.

Facility Access Control Systems

Facility access control systems authenticate workers prior to providing physical entry to facilities, with the goal of protecting the information systems that reside within the data center. Physical access control systems use the same concepts as logical access control systems for authentication based on something you know, something you have, and/or something you are. For example, the "something you know" may be a PIN code for a door. The "something you have" might include card-key systems or proximity badge systems, or you may have a physical key to unlock a door. In some cases, the access control system can be a standard key lock or simplex lock, although you'll see later that these are not preferred stand-alone mechanisms for controlling access. The "something you are" may include biometric devices that read fingerprints, hand geometry, and even retina characteristics to authenticate individuals who need to enter the facility.

Access control systems may use a man-trap to enforce the authentication mechanism. Man-traps consist of two doors that are separated by a corridor or a small closet-sized room. People entering the facility must first authenticate to open the door that allows them to enter the corridor. Once the first door closes behind them, they must authenticate again to open the door leading to the data center facility. The two doors cannot be open at the same time. Even if someone is able to circumvent security and gain access to the corridor via the first door, the person will be effectively trapped when the access control system blocks his or her access to the data center itself.

Alarm Systems

Because fire, water, extreme heat and humidity levels, power fluctuations, and physical intrusion threaten data center operations, data centers should implement several different types of alarm systems. Specifically, you will normally see the following types of alarms:

- Burglar alarms (with magnetic door, window, or cabinet sensors; motion sensors; and sometimes audio sensors)
- Fire alarms (usually heat- and/or smoke-activated sensors broken into zones that cover different parts of the facility)
- Water alarms (usually with sensors beneath the raised floor, near bathrooms, or in water pipe ducts)
- Humidity alarms (normally with sensors dispersed throughout the facility)
- Power fluctuation alarms (with sensors that can detect dips and spikes in the power grid)
- Chemical or gas alarms (sometimes in battery rooms and near air intakes)

These alarm systems usually feed into the data center operations center. During an alarm condition, the operator can drill down to specific sensors and reference a surveillance camera to isolate the cause of a problem.

Fire Suppression Systems

Because of the large amount of electrical equipment, fire is a major threat to data centers. Therefore, data centers normally are equipped with sophisticated fire suppression systems and should have a sufficient number of fire extinguishers. Generally speaking, fire-suppression systems come in two varieties: water-based systems and gas-based systems.

System and Site Resiliency

Because the computer systems that reside in a data center are leveraged to automate business functions, they must be available any time the business operates. Therefore, data centers incorporate various types of controls to ensure that systems remain available to perform critical business operations. These controls are designed to protect power, the computing environment, and wide area networks (WANs).

Power

Clean power is absolutely critical to maintain computer operations. Power fluctuations such as spikes, surges, sags, brownouts, and blackouts can damage computer components or cause outages. To mitigate this risk, data centers provide power redundancy in several layers, including the following:

- Redundant power feeds (connecting the data center to more than one power grid)
- Ground to earth (to carry power away from critical components during fault conditions)
- Power conditioning (to flatten out harmful spikes and sags in current)
- Battery backup systems or UPSs (to provide uninterrupted power in the event of power fluctuations, brownouts, or blackouts)
- Generators (to provide electrical power during prolonged power outages)

Heating, Ventilation, and Air Conditioning (HVAC)

Extreme temperature and humidity conditions can cause damage to computer systems. Because computers require specific environmental conditions to operate reliably, HVAC systems are required controls. Data centers typically provide sophisticated redundant systems to maintain constant temperature and humidity and often provide double the required capacity.

Network Connectivity

Whether from internal networks or the Internet, users access information systems

residing within data center facilities through network connections. Network connectivity is critical. More often than not, data center facilities have redundant Internet and WAN connections via multiple carriers. If one carrier experiences a network outage, service to the facilities can be provided by another carrier.

Data Center Operations

Although data centers are designed to be automated, they do require a staff to operate. As a result, data center operations should be governed by policies, plans, and procedures. The auditor should expect to find the following areas covered by policies, plans, and procedures:

- Physical access control
- System and facility monitoring
- Facility and equipment planning, tracking, and maintenance
- Response procedures for outages, emergencies, and alarm conditions

Disaster Preparedness

All data centers are susceptible to natural and manmade disasters. History shows that when disaster strikes a data center, the organizations such facilities serve come to a screeching halt. The auditor's job is to identify and measure physical and administrative controls at the facility that mitigate the risk of data-processing disruptions, including the following:

- System resiliency
- Data backup and restoration
- Disaster recovery planning

NOTE It is not within the scope of this chapter to cover business continuity. We instead focus on controls related to disaster recovery for the organization's IT environment specifically related to systems housed within data centers.

Test Steps for Auditing Data Centers

The following topic areas should be addressed during the data center audit:

- Neighborhood and external risk factors
- Physical access controls
- Environmental controls
- Power and electricity
- Fire suppression
- Data center operations
- System resiliency
- Data backup and restoration
- Disaster recovery planning

Test steps are detailed for each of these areas.

Neighborhood and External Risk Factors

When auditing a data center facility, you should first evaluate the environment in which the data center resides. The goal is to identify high-risk threats. For example, the data center you are auditing may be in the flight path of a regional airport, a Federal Emergency Management Agency (FEMA) flood zone, or a high-crime area. These types of environmental characteristics will reveal otherwise latent threats. In your audit, you will be looking for controls that reduce the likelihood of one of these threats being realized.

1. Review data center exterior lighting, building orientation, signage, fences, and neighborhood characteristics to identify facility-related risks.

Data center facilities should provide a physically secure environment for personnel and information systems. A breach of physical security, whether through a bomb, a physical intrusion, or a weather-related event, would compromise information and personnel security.

How

Perform a physical inspection of the data center facility. Pay attention to how far the building is set back from the curb and whether or not barriers (e.g., bollards) are in place to prevent cars from getting too close to the building. You're looking for controls that will reduce the risk of vehicle accidents or car bombs affecting the data center.

Determine on which floor of the building the data center resides. This information is important because below-ground and ground-level data centers are susceptible to flooding. Data centers on higher floors are more prone to lightning, wind, and tornado damage. The ideal is a single-story data center that is five feet or so above ground. If you're performing an audit of an existing data center, you obviously won't be able to get the auditees to move it just because you don't like the floor it's on; however, this information will help guide you in looking for compensating controls. For example, if the data center is below ground level, you will place extra focus on water-detection controls (discussed later in this chapter). This provides a great example of why it's valuable to be involved in your company's projects early. If you're invited to the table early during the development of a new data center, you might be able to influence the chosen location. Otherwise, all you can do is suggest controls to compensate for issues at the existing location.

Signage Review exterior signage to determine whether it's obvious to a passerby that the facility contains a data center. Data centers should be anonymous, away from main thoroughfares, and inconspicuously marked, if marked at all. In fact, most data centers employ what the security industry calls security through obscurity. Maintaining relative anonymity will reduce the possibility of the facility becoming a target for espionage, theft, or sabotage. Review interior signage as well. In general, it's best not to guide visitors in the building to the data center, especially if the building has frequent visitors from outside the company.

Neighborhood The next question is "Who are the neighbors of the data center facility?" Is it located in a multitenant building, or is it a stand-alone structure? If neighbors are within close proximity, in what sort of business are they engaged? A

data center that is located next to a warehouse or manufacturing facility may have an increased risk of being affected by hazardous material spills or fires. The ideal is a stand-alone structure without any close neighbors. Again, it will be difficult to influence this if you're auditing an existing data center, but knowing this information will help you identify necessary compensating controls. For example, if you're in a multitenant facility, you will want to suggest that the data center have stand-alone, segregated utilities (such as power feeds) so that the other tenants won't have a potential negative impact on the data center's power supply, water supply, and so on.

Exterior Lighting Evaluate exterior lighting. Proper lighting deters crime and loitering around the facility. Critical facilities should have exterior walls and parking lots illuminated uniformly at an intensity level that allows for viewing at a reasonable distance.

Fences Evaluate the adequacy of fences around the facility for deterring intruders. A three- to four-foot fence will deter common trespassers. A six-foot fence is too difficult to easily climb and will deter additional potential trespassers. An eight-foot fence with barbed wire at the top will deter all but the most determined intruders.

2. Research the data center location for environmental hazards and to determine the distance to emergency services.

Environmental threats such as floods, severe weather, and transportation-related accidents can destroy or severely damage a data center. In the event of an emergency, rapid response from authorities is critical. Therefore, the proximity to fire stations, police stations, and hospitals is important.

How

Perform research to identify environmental hazards that may not be evident during the onsite visit. Look for information on the following areas:

- Flood elevations
- Weather and earth movement threats
- Proximity to transportation-related hazards
- Local crime rate
- Proximity to industrial areas
- Proximity to emergency services

If you're reviewing an existing facility, you probably won't be able to affect the existence of these characteristics, as it's usually not realistic to recommend that the data center be moved. However, you can use this information during the audit to determine compensating controls that should be put in place. Ideally, you'll be able to consult during the construction of a new data center and influence its location based on the existence of these factors. However, even if the data center has already been built, as an auditor, it is your responsibility to inform management about risks to the business. It is management's responsibility to decide where to spend limited resources in an effort to mitigate those risks. Even if it's not realistic to relocate the data center, it may be reasonable to suggest that additional monitoring and disaster recovery capabilities be put in place.

Flood Elevations According to FEMA, floods are one of the most common hazards in the United States. Finding flood-zone information on the Internet is relatively easy.

NOTE The following Internet resources are available to assist auditors in evaluating flood risks: https://hazards.fema.gov and https://msc.fema.gov.

You should also identify any flooding hazards that are present as a result of the data center's location within the building. Determine what is located in the rooms immediately adjacent to and above the data center. Restrooms and other rooms involving frequent water usage introduce the threat of leaks and burst pipes flooding the data center.

Weather and Earth Movement Threats Since different geographic zones are prone to different weather and earth movement hazards, you should understand which of these threats are prevalent in the geographic area in which the data center resides. For example, if the data center you are auditing is in Dallas, Texas, the threats would be tornados, flooding, and extreme heat, whereas in northern California the threat would come from earthquakes.

NOTE Some excellent weather-related Internet resources include www.noaa.gov, https://earthquake.usgs.gov, and https://hazards.fema.gov.

Proximity to Transportation-Related Hazards Planes, trains, and automobiles represent another risk to data center operations. Specifically, research whether or not the data center you are auditing is in an airport flight path or if a rail line is near the facility. Though rare, planes do crash and trains do derail and can pose a risk. Maps and observation are good methods for identifying nearby transportation-related hazards.

Local Crime Rate Obviously, if your data center is in a high-crime area, there is a higher risk of theft and other crimes. Therefore, another statistic to research is the local crime rate. If the area has a high crime rate, you may recommend mitigating controls such as reinforced fences, an increased presence of security personnel, closed-circuit television (CCTV), and perimeter alarm systems.

NOTE Several useful sources of online crime statistics are available at the following websites: www.ucrdatatool.gov and www.cityrating.com/crimestatistics.asp.

Proximity to Industrial Areas Many data center facilities are situated in industrial zones near factories and warehouses. These areas generally have a higher crime rate and a higher risk of hazardous material spills affecting data center operations. Therefore, if the data center is situated in an industrial area, you should evaluate the risks inherent to the area and determine any needed compensating controls.

Similarly, within your own building, determine the usage of the rooms immediately adjacent to and above the data center. Manufacturing processes and other processes involving chemicals introduce the risk of chemical leaks and explosions.

Proximity to Emergency Services When an emergency occurs within a data center, every minute that passes can be costly. Therefore, it is important that you evaluate the distance to police stations, hospitals, and fire stations. Again, this is probably not an area you can influence after the fact, but it does provide good background information as you perform the rest of the audit, helping you gauge the level of capabilities you need onsite versus what you can rely on externally.

Physical Access Controls

Several information security incidents have occurred in which thieves gained unauthorized access to sensitive information by defeating physical access control mechanisms. Therefore, restricting physical access is just as critical as restricting logical access. In a data center environment, physical access control mechanisms consist of the following:

- Exterior doors and walls
- Access control procedures
- Physical authentication mechanisms
- Security guards
- Other mechanisms and procedures used to secure sensitive areas

3. Review data center doors and walls to determine whether they protect data center facilities adequately.

A data center's first and most formidable line of defense should be the walls and doors used in its construction. Look closely at how well doors and walls protect against intrusion and other hazards such as projectiles or blasts.

How

Through interviews and observation, identify all potential entry points into the data center. Verify that walls and doors are adequately reinforced. Exterior walls should be reinforced with steel and concrete to protect the facility. If the data center resides within a building, the walls may be constructed of sheetrock but should be reinforced with steel to prevent intrusion. Exterior doors should also be reinforced and should be able to withstand intrusion attempts. Ideally, there should be no exterior-facing doors or walls, which provides an extra layer of protection against forced entry. You can attempt to influence this if consulting prior to data center construction.

Raised Floors and Drop Ceilings Most data centers use either raised floors or drop ceilings to conceal ventilation ducts and power and network cables. Interior building walls sometimes are constructed with spaces below a raised floor or spaces above drop ceilings left unwalled. This would allow someone attempting to gain unauthorized access to the secured area to remove either a floor tile or a section of the drop ceiling to crawl over or under the wall. This is a common oversight that can allow intruders to bypass your physical security controls. During the building tour, remove a section of raised floor and a ceiling tile at a data center

wall to verify that walls extend from the structural floor to the structural ceiling. If they do not, you will need to encourage the addition of wall extensions or reinforced wire cages above and below the data center to prevent unauthorized entry.

Doors Ensure that doors are force-resistant, preferably with magnetic locks. Review the location of each door's hinges. If they are on the outside of the room, ensure they are protected to prevent an intruder from removing the door by popping it off its hinges.

Man-traps are an effective means of controlling access to critical facilities. Verify through observation that man-traps exist where appropriate and that they are working properly. Man-traps are equipped with two locking doors with a corridor in between. To ensure security, one door should be required to be locked before the other is allowed to open. Obviously, the man-trap should be constructed of reinforced walls and doors as well.

Note that in order to ensure personnel safety, emergency exit paths should not require authentication or other special measures to exit a data center (although it should be expected that there be some sort of alarm and investigation should these emergency exit paths be utilized).

Windows Identify any windows looking into the data center and ensure that all are constructed with reinforced shatterproof glass. In general, windows looking in on the data center should be avoided, as they advertise the location of the data center to passersby. If any windows provide a view into the data center from outside the building, determine whether they have been adequately covered with curtains or blinds or other obscuring mechanisms.

4. Evaluate physical authentication devices to determine whether they are appropriate and are working properly.

Physical authentication devices such as card-key readers, proximity badges, biometric devices, simplex (combination) locks, and traditional key locks serve to allow access to authorized personnel and keep out unauthorized personnel. The failure or misuse of these devices can allow unauthorized persons access to the data center or prevent authorized personnel from entering.

How

For each entry point into the data center, identify the physical authentication mechanism and ensure it has the following characteristics:

- Restricts access based on the individual's unique access needs or even restricts access to particular doors or to particular hours of the day
- Easily deactivated in the event an employee is terminated or changes jobs or in the event a key/card/badge is lost or stolen
- Difficult to duplicate or steal credentials

Obtain a sample of data center authentication device logs and verify that the device is logging the following information:

- User identification
- Date, time, and place of the access attempt
- Success or failure of the access attempt

Review processes for periodically reviewing and investigating these logs.

Card-Key and Proximity Devices
Card-key devices use magnetic stripes or radio frequency identification (RFID) chips to authenticate users who possess the card. Because a stolen card can be used for unauthorized authentication, a PIN-code device will preferably be coupled with the card-key reader. Verify that all card-key readers are working properly and are logging access attempts.

Biometric Devices Biometric devices are able to measure fingerprint, retina, and hand geometry. Because these biometric characteristics are unique to each individual, biometric authentication devices are difficult to defeat. However, that does not mean they are impossible to defeat, so they are still ideally paired with a second factor (e.g., a PIN code). Review the quality of the biometric system being used to determine whether an inordinate number of false negatives or any observed false positives have occurred.

Key Locks and Combination Locks Traditional key locks and simplex (combination) locks are the weakest forms of physical authentication and should be avoided. These forms of physical authentication offer no way to identify who has access to the data center. Keys can be lost, stolen, borrowed, or copied. Combination codes can be shared or can be stolen via shoulder surfing (watching someone enter the code). These are also the most difficult credentials to revoke when an employee no longer needs access to the data center.

5. Ensure that physical access control procedures are comprehensive and being followed by data center and security staff.

Physical access control procedures govern employee and guest access to the data center facility. If physical access control procedures are incomplete or not enforced consistently, data center physical access will be compromised.

How

Review the following related to physical access control procedures:

- Ensure that access authorization requirements are documented and clearly defined for both employees and guests. Approval from one (or more) of a predefined set of knowledgeable individuals should be required before data center access is granted. Standards for what constitutes a need for ongoing data center access should be established. For example, an employee who needs only occasional (such as quarterly) access to the data center does not need ongoing access but can instead arrange to be escorted on the occasions when access is needed. The philosophy of "minimum necessary access" should be embraced when it comes to granting access to data center facilities.
- Verify that guest access procedures include restrictions on taking pictures and outline conduct requirements within the data center. Visitors should be required to sign a visitor log indicating their name, company, and reason for visiting and should be required to wear identification badges that are a different color from employee badges. Visitors should be escorted at all times, and vendor service personnel (including cleaning personnel) should be supervised while onsite.
- Review a sample of both guest access and employee ID authorization requests to ensure that access control procedures are followed.
- Review procedures for ensuring that data center access is removed (including the collection of physical devices such as badges, keys, and cards) when it is no longer required. This should be part of the termination checklist and will preferably be automated. It should also encompass changes of jobs within the company so that employees don't retain data center access beyond the point when it is needed.
- Obtain a list of all individuals who have access to the data center, select a representative sample of employees with access to the data center, and determine whether access is appropriate.
- Determine whether management regularly reviews the physical access authorizations for validity. Management should periodically pull a list of people with data center access and review it for appropriateness. Review evidence that this is happening.

6. Ensure that intrusion alarms and surveillance systems are protecting the data center from physical intrusion.

Intrusion alarms and surveillance systems mitigate the risk of undetected physical intrusion by serving as a detective control as well as a deterrent for would-be intruders. The absence of these controls would increase the risk of theft and other criminal activities.

Most data centers employ either CCTV, audio surveillance systems, or a combination of the two. These systems typically feed into a guard station, where they are monitored by security personnel and recorded on either tape or a digital storage system. Data centers also often employ burglar alarms, generally through a series of sensors that are placed in strategic locations such as doors and hallways.

How

Review the placement of intrusion sensors, verifying that critical areas of the data center are covered adequately, and review maintenance logs to ensure that the system has been maintained and tested properly. Look for the following common types of sensors:

- Motion sensors that detect infrared motion
- Contact sensors that are placed on windows and doors to detect when they are opened or broken
- Audio sensors to detect breaking glass or changes in normal ambient noise
- Door prop alarms to detect when a data center door is left open for more than a specified length of time (typically 30 seconds)

Review camera quality and placement, ensuring that they are located at strategic points in the data center (such as each entry point). Verify that the surveillance systems are monitored and evaluate the frequency of the monitoring. Verify that the video surveillance is recorded for possible future playback and review tape rotation or mass-storage archival schedules.

These steps can be performed through a combination of document review and observation. The data center security manager should be able to provide this

information.

7. Review security guard building round logs and other documentation to evaluate the effectiveness of the security personnel function.

Security guards can be one of the most effective physical access controls. They act as a deterrent and can also control facility access and respond to incidents with cognitive reasoning. If the security personnel function is ineffective, emergency response most likely would be slow and ineffective, doors could be left unlocked, and unauthorized personnel could have the opportunity to enter the data center facility.

How

Verify that documentation of building rounds, access logs, and incident logs/reports exist and that this information is recorded properly by obtaining samples from the security staff. Look for consistent entry and exit times, regular building tours, and comprehensive incident logs/reports. Visit the main security post to obtain this documentation.

8. Verify that sensitive areas within the data center are secured adequately. Ensure that all computer processing equipment essential to data center operations (such as hardware systems and power supply breakers) is located within the computer processing room or in a secure area.

Data centers typically have some areas that are more sensitive than others, such as equipment staging areas, generators, and computer systems that are processing sensitive information. If a large number of people have access to the data center, sensitive equipment may need to be segregated in high-security areas. If these areas are not adequately secured, information could be altered or disclosed to unauthorized personnel or destroyed due to a system failure caused by either sabotage or an accident.

If equipment essential to data center operations is not located within the data center (or an equally controlled area), someone without data center access may be able to adversely affect data center availability and/or access sensitive information.

How

Based on the number of people with access to the data center and the nature of the equipment contained therein, evaluate the need for access to be further segregated within the data center. For example, computer systems that process sensitive information may be locked within a cage or cabinet, with only a select number of personnel given access. During interviews and tours of the data center, verify that these areas are protected appropriately with proper access control mechanisms and, if appropriate, are monitored by CCTV cameras and/or alarm systems.

Review the location of all data center systems, including power supplies, HVAC equipment, batteries, production servers, and so on, and ensure all are located within the data center or an equally secured facility.

Environmental Controls

Computer systems require specific environmental conditions such as controlled temperature and humidity. Data centers are designed to provide this type of controlled environment.

9. Verify that HVAC systems maintain constant temperatures within the data center.

HVAC systems are used to provide constant temperature and humidity levels. Computer systems can be damaged by extremes in either. High humidity can cause corrosion of computer components, and low humidity can cause static electricity discharges that can short-circuit system boards. High temperatures can reduce the lifespan of computer equipment and result in system freezes and crashes.

How

Review the following areas:

- Temperature and humidity logs to verify that each falls within acceptable ranges over a given period. In general, data center temperatures should range from 65 to 70°F (with temperatures above 85°F damaging computer equipment) and humidity levels should be between 45 and 55 percent. However, this will vary depending on the specifications of the equipment. Determine how the data center staff has established the parameters for the equipment.
- Temperature and humidity alarms to ensure data center personnel are notified of conditions when either factor falls outside of acceptable ranges. Sensors should be placed in all areas of the data center where electronic equipment is present. Ensure that sensors are placed in appropriate locations either by reviewing architecture diagrams or by touring the facility. Review maintenance and testing documentation to verify that the system is in good working order.
- HVAC design to verify that all areas of the data centers are covered appropriately. Determine whether the air flow within the data center has been modeled to ensure adequate and efficient coverage. Look for cold aisle and warm aisle configuration, which is a configuration of equipment racks where servers are faced such that hot and cold air are separated, thereby improving cooling efficiency.
- Configuration of the HVAC systems. The data center should use a self-contained air conditioning system that is isolated from other building systems and can be used with backup power. This will allow the HVAC controls to continue to function for the data center in the event of a power loss. Data center air conditioning ducts should be designed so as not to penetrate the perimeter walls. Otherwise, they could allow unauthorized access from outside the data center. Verify that there is enough HVAC capacity to service the data center even in the most extreme conditions.

This information usually can be obtained from the facility manager.

10. Ensure that a water alarm system is configured to detect water

in high-risk areas of the data center.

Water and electronic equipment do not mix. Data centers normally employ water sensors in strategic locations such as near water sources or under raised floors. Water sensors detect the presence of water and are designed to alert data center personnel prior to a major problem.

How

Identify potential water sources such as drains, air conditioning units, exterior doors, and water pipes to verify that water sensors are placed in locations where they will mitigate the most risk. The facility manager should be able to point out both water sources and sensors during a tour of the facility. Review maintenance records to ensure that the alarm system is maintained periodically.

Floor plans indicating shutoff valves for all water systems should be available. Data center managers should be aware of all water valves within the secured area. Determine whether this is the case.

Power and Electricity

Computer systems require uninterrupted, clean power to operate. Data centers typically employ several different types of controls to maintain clean power. These controls include the following:

- Redundant power feeds that provide power from two or more power stations
- Ground-to-earth to carry excess power away from systems during electrical faults
- Power conditioning systems to convert potentially dirty power to clean power
- Battery backup systems (UPSs) that provide immediate power, typically for short periods
- Generators to provide sustained power during extended power losses

11. Determine whether the data center has redundant power feeds.

Some data centers are built in locations where they can connect to more than one power station. When the power supplied by one feed is lost, the other often will remain live. As a result, redundant power feeds can be used to maintain utility power continuity.

How

This control is not always possible, but it is worth exploring with the data center facility manager during interviews.

12. Verify that ground-to-earth exists to protect computer systems.

Ungrounded electrical power can cause computer equipment damage, fire, injury, or death. These perils affect information systems, personnel, and the facility itself. Today, buildings that do not have grounded electrical outlets are probably in violation of building code. Unlike redundant power feeds, the ground-to-earth control should always be present. Ground-to-earth is a basic feature of all electrical installations that consists of a *green wire* that connects all electrical outlets to a rod that is sunk into the ground. When short circuits or electrical faults occur, excess voltage is passed through the ground wire safely into the ground rather than short-circuiting electrical equipment. This control should be present in any facility less than 30 years old or so, but it is definitely worth verifying. Older buildings that have not had electrical systems upgraded may not have an electrical ground. However, electrical ground normally is required in building codes.

How

This information can be obtained by interviewing the data center facility manager or through observation.

13. Ensure that power is conditioned to prevent data loss.

Power spikes and sags damage computer systems and destroy information. Power conditioning systems mitigate this risk by buffering the spikes and sags. Clean power can be represented as a wave pattern with symmetric peaks and valleys. Normal utility power has a wave pattern with peaks and valleys that are far from symmetric, causing momentary spikes and sags. These spikes and sags shorten the life of electronic components and sometimes cause system faults. Power conditioning systems smooth out the wave pattern to make it symmetric.

How

Through interviews and observation, verify that power is being conditioned by either a power conditioning system (such as surge protectors) or a battery backup system.

14. Verify that battery backup systems are providing continuous power during momentary blackouts and brownouts.

Power failures can cause data loss through abrupt system shutdowns. UPS battery systems mitigate this risk by typically providing 20 to 30 minutes of power as well as power conditioning during normal utility power conditions. Basically, they provide enough time for the generator (if available) to turn on and begin generating electricity, or for critical systems to be shut down gracefully to minimize data loss. They also perform a power conditioning function, because they logically sit in between utility power and computer center equipment. As a result, the batteries are actually powering the data center all the time. When utility power is live, the batteries are charged constantly. Conversely, when power is lost, they begin to drain.

How

Interview the data center facility manager and observe UPS battery backup systems to verify that the data center UPS system is protecting all critical computer systems and affords adequate run times (that is, make sure the batteries will run long enough for the generator to kick in and/or for critical systems to be shut down gracefully). In some cases, the UPS system may be able to initiate a graceful shutdown automatically when capacity reaches a certain threshold. Look for the existence and implementation of this feature.

Review a list of equipment tied into the UPS and ensure all critical systems are covered (such as critical production servers, network equipment, HVAC systems,

fire detection and suppression systems, monitoring systems, badge readers, and so on).

15. Ensure that generators protect against prolonged power loss and are in good working condition.

Mission-critical data centers, by their nature, cannot withstand any power loss. Since it is impractical to install enough batteries to power the data center for more than an hour or two, generators allow the data center to generate its own power in the event of a prolonged loss of utility power.

Generators come in two common varieties: powered by diesel or natural gas or powered by propane. Each has its benefits and drawbacks.

Diesel generators are the most common but have a finite amount of fuel stored in their tanks. Diesel fuel is also a biohazard. Spillage could result in significant cleanup expenses. Also, if the generator is in close proximity to the data center and a spill reaches the data center, it would be disastrous. These risks can be mitigated through fuel service contracts and spill barriers, however.

Natural gas generators run cleaner and theoretically have an infinite supply of fuel as long as the gas lines are intact. There is no danger of spills, but fire danger is increased. Natural gas generators are employed rarely, however, because of the expense.

Propane generators are also expensive but have a limited supply of fuel. Again, this can be mitigated with service contracts.

How

Through observation and interviews, verify that the data center has a generator. In addition, obtain the sustained and peak power loads from the facility manager and compare them with current power generation capacity. Generators should be able to produce at least double the sustained power load.

Determine the generators' ability to power operations for a sustained period by reviewing onsite fuel storage as well as service contracts for replenishing fuel. Review controls in place to mitigate the inherent risks for whichever type of fuel is being used (such as spill barriers if diesel fuel is used).

All types of generators require frequent maintenance and testing, so review both maintenance and test logs during a data center audit.

16. Evaluate the usage and protection of emergency power-off (EPO) switches.

EPO switches are designed to shut off power immediately to the computer and peripheral devices during emergencies, such as during a data center fire or emergency evacuation. If they are not adequately protected, it could result in inadvertent shutdown of the data center.

How

Through observation, review the EPO switch(es) for the data center. Ensure that they are clearly labeled and easily accessible yet still secured from unauthorized or accidental usage. They should be inside the secured area and underneath some sort of shield to prevent accidental activation.

Fire Suppression

Since data centers face a significant risk from fire, they typically have sophisticated fire suppression systems, generally one of two types: gas-based systems and water-based systems. The data center relies on more than just fire suppression systems, however, as controls. Other fire suppression controls include the following:

- Building construction
- Fire extinguishers
- Proper handling and storage of hazardous materials

17. Ensure that data center building construction incorporates appropriate fire suppression features.

For more than 30 years, building codes have required that buildings be constructed in such a way as to resist fire. Fire suppression features include the following:

- Fire-rated walls and doors to prevent fire from moving from one area of a building to another
- Firestops where fire-rated walls or floor assemblies are sealed to prevent the spread of fire
- Standpipe fire hose systems to provide a ready supply of water for fire suppression (a standpipe is like a fire hydrant within a building, providing a fixed water pipe to which a fire hose can be connected)

The absence of these features introduces the risk of a fire spreading more quickly and causing additional damage and possibly threatening lives.

How

Review the available fire suppression features built into the facility. The facility manager or local fire marshal should be able to provide information about wall/door fire rating and firestops. Standpipe water systems will be visible and observed easily during a building tour.

18. Ensure that data center personnel are trained in hazardous materials (hazmat) handling and storage and that hazmat procedures are appropriate. Also determine whether data center personnel are trained in how to respond to a fire emergency or hazardous material spill.

Hazardous and highly flammable materials are a common cause of fire. These materials include the following:

- Diesel and other fuels
- Solvents and thinners
- Propane or acetylene torches
- Chlorine or ammonia-based chemicals
- Glues and bonding compounds

These materials should be handled and stored in a proper manner to mitigate the risk of fire or spillage. Also, data center personnel should be trained in how

to respond to a fire or hazardous material spill (such as knowledge of emergency numbers to call, when and how to activate fire suppression systems, and so on) to minimize the threat to equipment and human life.

How

Review hazmat incident reports and hazmat and fire response training materials and procedures, as well as interview data center staff.

Through observation, determine whether anything is being unnecessarily stored in or near the data center of a combustible nature (such as paper stock, toners, cleaners, or other chemicals). If so, suggest that it be removed to reduce the need for hazmat procedures.

See step 26 for guidance on ensuring emergency response procedures are in place and communicated.

19. Verify that fire extinguishers are strategically placed throughout the data center and are maintained properly.

Fire extinguishers are often the first line of fire defense. In data centers, they should be placed in hallways and aisles every 50 feet or so. Three common types of extinguishers can be used: dry chemical based, water based, and inert gas based. In most cases, data centers should use inert gas–based fire extinguishers, such as CO_2 extinguishers, because water and dry chemicals damage electrical equipment. A lack of usable fire extinguishers could result in a small fire getting out of control.

How

Review the locations of fire extinguishers, as well as a sample of the attached service tags, during a data center tour. Ensure the location of each fire extinguisher is marked appropriately and easily visible. Since many data centers contain racks that are at least six feet tall, a marker should identify the location of each fire extinguisher that is visible above the racks.

The data center facility manager also should be able to supply maintenance records. Fire extinguishers should be inspected at least annually.

20. Ensure that fire suppression systems are protecting the data center from fire.

All data centers should have a fire suppression system to help contain fires. Most systems are gas based or water based and often use multistage processes in which the first sensor (usually a smoke sensor) activates the system and a second sensor (usually a heat sensor) causes a discharge of either water or gas.

Gas-Based Systems Varieties of gas-based fire suppression systems include CO_2, FM-200, and CEA-410. Gas-based systems are expensive and often impractical, but their use does not damage electronic equipment.

Water-Based Systems Water-based systems are less expensive and more common but can cause damage to computer equipment. To mitigate the risk of damaging all the computer equipment in a data center or in the extended area of a fire, fire suppression systems are designed to drop water from sprinkler heads only at the location of the fire. Four common types of fire suppression systems are used:

- **Wet pipe** Pipes are always filled with water. This is the least desirable type of system for a data center because a leaky pipe or broken sprinkler head would result in flooding.
- **Dry pipe** Pipes are filled with air and are filled with water at the time of a discharge.
- **Preaction** Pipes are filled at stage 1 activation, and water is discharged during stage 2.
- **Deluge** A dry pipe system that discharges a large amount of water to overwhelm a fire.

The absence of a fire suppression system would allow a fire to spread more quickly, resulting in more equipment loss and possibly loss of life.

How

Review system design, maintenance, and test records. This information can be obtained through a combination of interviews, document review, and observation. The data center facility manager should be able to provide the design, maintenance, and test documentation.

If water-based systems are used, determine whether the pipes above the data center are always filled with water (a wet pipe system). If so, determine what mitigating controls have been employed to minimize the chance of unintended water flow into the data center, such as from a broken sprinkler head or leaky pipe. For example, look for cages around the sprinkler heads, water flow sensors, and regular maintenance of the pipes.

If gas-based systems are used, determine the type of gas in use and ensure it is not harmful to humans if inhaled. Since gas-based systems can displace oxygen, ensure that data center personnel are aware of fire procedures and suppression system risks through training, signage, or a combination. If a fire suppressant that is harmful to humans is in use, ensure there are visual and/or audible alarms prior to and during discharge of the suppressant.

21. Verify that fire alarms are in place to protect the data center from the risk of fire.

Because of all the electrical equipment, the likelihood of fire can be significantly higher for data centers. Fire alarms alert data center personnel and local fire departments of a developing fire condition so that they can begin fire response procedures and evacuate the premises. A fire alarm failure would put data center operations and human lives at risk.

Data center fire alarm systems usually are multizone systems, which reduce the risk of false alarms due to a single malfunctioning sensor or zone. In such a system, sensors in two or more zones must detect the fire before an alarm sounds. Three types of sensors can be used:

- **Heat sensors** Activate when temperature reaches a predetermined threshold or when temperatures rise quickly
- **Smoke sensors** Activate when they detect smoke
- **Flame sensors** Activate when they sense infrared energy or flickering of a flame

Smoke and heat sensors are most common.

Hand-pull fire alarms should also be strategically located (such as near all

entrances) throughout the data center so that employees can raise an alarm when observing a fire condition.

How

Via physical observation and interviews, review fire alarm sensor type, placement, maintenance records, and testing procedures. Sensors should be located above and below the ceiling tiles and below the raised floor.

Observe whether hand-pull fire alarms are strategically located throughout the data center and review maintenance records and testing procedures.

Data Center Operations

Effective data center operations require strict adherence to formally adopted policies, procedures, and plans. The areas that should be covered include the following:

- Facility monitoring
- Roles and responsibilities of data center personnel
- Segregation of duties of data center personnel
- Responding to emergencies and disasters
- Facility and equipment maintenance
- Data center capacity planning
- Asset management

22. Review the alarm monitoring console(s), reports, and procedures to verify that alarms are monitored continually by data center personnel.

Alarm systems most often feed into a monitoring console that allows data center personnel to respond to an alarm condition before calling authorities, evacuating the building, or shutting down equipment. The absence of a monitoring console and appropriate response procedures would introduce the risk of an alarm condition going unnoticed.

How

Review alarm reports and observe the data center alarm-monitoring console to verify that intrusion, fire, water, humidity, and other alarm systems are monitored continually by data center personnel. Occasionally, the intrusion alarm is monitored by data center security staff. The main objective here is to verify that all applicable alarms are being monitored.

Review facility monitoring and response procedures to ensure that alarm conditions are addressed promptly. Facility monitoring procedures ensure that all critical alarm conditions are captured and acted on promptly. They should include a description of the alarm systems that will be monitored, as well as the steps that are to be taken in the event of all reasonably foreseeable alarms, including fire, intrusion, water, power outage, data circuit outage, system, and system component alarm conditions. Verify that the response is clearly outlined for each type of alarm condition. Obtain the actual monitoring procedures as well as monitoring logs from data center facility management.

23. Verify that network, operating system, and application monitoring provides adequate information to identify potential problems for systems located in the data center.

System monitoring provides insight into potential problems resulting from capacity issues, misconfigurations, and system component failures. Inadequate system monitoring gives rise to the threat of security violations going undetected and system outages. Although this function typically is managed by IT service groups rather than data center personnel, monitoring is a critical component of sound operations for the systems in the data center. System monitoring encompasses the monitoring of network devices, intrusion detection systems, operating systems, system hardware, and applications. Whereas intrusion detection system monitoring is focused primarily on monitoring for security violations, network device, operating system, system hardware, and application monitoring is focused primarily on items that can affect the availability of a system, such as hard disk usage, number of concurrent connections, and so forth. Therefore, when auditing the monitoring of system procedures, you need to understand the objective of the system.

How

Determine the criticality of specific system components within the data center and verify that monitoring systems provide near–real-time information to detect a problem with these system components. Determine how the computer systems are monitored and if an automated or manual problem log is maintained for hardware and software failures and downtime. Examples of items that may be monitored include system uptime, utilization, response time, and errors. In addition, review monitoring logs and reports to identify whether any components being monitored exceed predetermined thresholds, and then verify that actions have been taken to remediate the condition. Monitoring logs and reports typically can be obtained from system support groups, network support groups, and security and application monitoring teams.

24. Ensure that roles and responsibilities of data center personnel are clearly defined.

Well-defined employee roles and responsibilities ensure that responsibility and accountability for data center functions are clear. Inadequate roles and responsibilities can result in unclear job boundaries and data center functions going unaddressed, which could increase the risk of system outages.

How

Review documentation and verify that all job functions are covered and that responsibilities associated with job functions are clearly defined. Data center facility management should be able to provide job descriptions, including roles and responsibilities.

25. Verify that duties and job functions of data center personnel are segregated appropriately.

Segregation of duties is a basic security precept of personnel management. The goal is to spread high-risk duties across two or more employees to reduce the risk

of fraud or inadvertent errors. If high-risk functions are not segregated, the data center will have a higher degree of fraud risk.

How

Verify that high-risk job functions, such as access authorization, are segregated across two or more employees. These processes should be tracked with logs and forms that can be reviewed to verify that duties are segregated effectively.

26. Ensure that emergency response procedures address reasonably anticipated threats.

Data centers are faced with various threats, including the following:

- Fire
- Flood
- Physical or logical intrusion
- Power loss
- System failure
- Telecommunications outages
- Hazardous material spills

These and other identified threats should be addressed by emergency response plans. When a fire breaks out or a data center floor begins to flood, data center personnel need a clear plan to address the condition and minimize losses. Although used only during the unlikely event of an emergency, emergency response plans are absolutely critical for reducing the risk of an emergency escalating due to improper response from data center personnel. For example, suppose a generator catches fire while being tested. Without clear procedures and proper training, you probably would witness employees running around in the heat of the moment, responding in a way that they think is most appropriate but most likely not working together to solve the problem. With clear emergency response procedures, such decisions would have already been considered, and employees would not be forced to make decisions in the heat of the moment, resulting in a more coordinated response.

How

Review response plans. Verify that plans are present for all foreseeable threats, and ensure that response procedures are comprehensive and well thought out. Data center operations staff should be able to provide these plans. Verify that they have received appropriate communications and training regarding these plans. Observe whether emergency telephone numbers are posted or easy to access and that they include outside police, fire departments, and other emergency response groups.

27. Verify that data center facility-based systems and equipment are maintained properly.

When not properly maintained, facility-based systems and equipment are prone to premature failure. These breakdowns can cause loss of information and system outages. As a result, maintenance is critical.

How

Review maintenance logs for critical systems and equipment. Critical systems and equipment should be maintained at least semiannually. The data center facility manager should be able to provide the maintenance logs.

Determine whether procedures are in place for daily or weekly cleaning of the data center, including regular cleaning under the raised data center floor and of computer equipment. Dirt and dust in the data center can negatively affect the functioning of computer equipment.

28. Ensure that data center personnel are trained properly to perform their job functions.

Data center personnel cannot be expected to be proficient if they are not afforded job training. When not trained properly, data center personnel are more likely to cause data loss or system outages due to mistakes.

How

Review training history and schedules. Ensure that training is relevant to job functions and that all data center personnel are afforded training. Determine whether there is ongoing communication of employee responsibilities with respect to confidentiality, integrity, availability, reliability, and security of all IT resources. Look for policies that prohibit eating, drinking, and smoking within the data center, or those that at least restrict such activity to special break areas. Also, look for signs posted stating such prohibitions.

Data center management should be able to provide access to training history and schedules. Review history for the past full year and schedules for the next six months.

29. Ensure that data center capacity is planned to avoid unnecessary outages.

Capacity planning ensures that procedures are in place to monitor and analyze factors that could affect the data center's current or future power, network, heating, ventilation, air conditioning, and space requirements. Inadequate capacity planning could result in data loss, system outages, and/or delays in system deployments. Capacity management is a broad topic that was covered in detail in Chapter 3. A well-managed data center will be able to forecast how much rack space; network drops; network gear; electricity; and heating, ventilation, and air conditioning, just to name a few, are needed to support current and future operations.

How

Review monitoring thresholds and strategies that data center management uses to determine when facilities, equipment, or networks require upgrading. Data center management should be able to provide the capacity planning strategy and documented procedures, including thresholds for upgrading systems. Verify that these procedures are comprehensive and review evidence that they are being followed.

30. Verify that procedures are present to ensure secure storage

and disposal of electronic media.

Electronic media often contain sensitive information that, if disclosed, would constitute a compromise of information security. As a result, media storage and disposal must be closely controlled. Improper storage of electronic media could also result in accidental corruption of the information stored on the media.

How

Ensure that the following media storage and disposal controls exist within the data center:

- Electronic media are stored in a dry, temperature-controlled, and secure environment.
- Electronic media containing sensitive information is encrypted and tracked as it moves from one location to another.
- Electronic media is degaussed, overwritten with a Department of Defense (DOD)–compliant electronic shredding utility, or physically destroyed prior to disposal.

You should be able to obtain media tracking, storage, and disposal records from data center management. Tour electronic media storage facilities within the data center to verify that appropriate access and environmental controls are in place. For more information regarding electronic media management, see Chapter 3.

31. Review and evaluate asset management for data center equipment.

Asset management is the controlling, tracking, and reporting of assets to facilitate accounting for them. Without effective asset management, the company will be subject to the increased expense of duplicate equipment if assets are available but not locatable. The company will also be subject to unnecessary lease expenses if leased equipment is not adequately tracked and returned on time. Similarly, without adequate asset management, end-of-life equipment conditions may not be noted, resulting in increased risk of hardware failure. Theft of equipment that is not tracked could go unnoticed.

How

Review and evaluate the data center's asset management policies and procedures and ensure that they comply with company policy and encompass the following:

- **Asset procurement process** Ensure that this process requires appropriate approvals prior to the purchase of hardware.
- **Asset tracking** Ensure that the data center is using asset tags and has an asset management database.
- **Current inventory of all equipment** Ensure that an inventory contains the asset number and location of all hardware, along with information about the equipment's warranty status, lease expiration, and overall life cycle (that is, when it falls out of vendor support). Ensure that an effective mechanism is in place for keeping this inventory up to date. A sample of asset tags also should be inspected visibly and traced to the inventory.
- **Asset move and disposal procedures** Ensure that unused equipment is stored in a secure manner.

System Resiliency

Most information systems that reside within data centers process information that requires high system availability. Data center controls ensure high availability relative to the facility, whereas redundant system components and sites are used to ensure system availability in relation to the computer hardware.

32. Ensure that hardware redundancy (redundancy of components within a system) is used to provide high availability where required.

Failure of system components will cause system outages and data loss. When high system availability is required, systems should contain redundant system components such as Redundant Array of Inexpensive Disks (RAID) and redundant power supplies.

How

Determine whether standards for data center hardware include requirements for redundant components. For a sample of systems within the data center, ensure that critical system components such as disk storage and power supplies are redundant wherever possible. Information about hardware redundancy can be found within system specification documents. Data custodians (administration personnel) should be able to provide this documentation.

33. Verify that duplicate systems are used where very high system availability is required.

If system downtime will result in significant costs or loss of revenue to the business and system downtime cannot be tolerated, duplicate (redundant) systems are used to provide for automatic failover in the event of a system crash. This should not be confused with the preceding step, which evaluates the redundancy of components within a single system. This step is referencing the potential need for duplicating the system in its entirety. For the most critical systems, these redundant systems might be placed at two or more separate locations, allowing information to be copied to alternative sites at set intervals such as daily or in real time.

When reviewing system redundancy, you need to determine the manner in which data is copied from the main system to duplicate systems. Because most systems with this level of criticality are database applications, we will focus on database redundancy. Three types of systems provide database transaction redundancy:

- **Electronic vaulting** Provides periodic data copies through a batch process
- **Remote journaling** Provides real-time parallel processing over a network connection
- **Database shadowing** Provides real-time parallel processing over two or more network connections

How

For a sample of systems in the data center, ensure that the appropriate level

of system redundancy is being used for the level of system availability that is required. Include redundancy of network connectivity for the data center in this analysis. System redundancy information usually can be obtained from system architecture documentation and interviews with data center and system administrators.

Data Backup and Restoration

System backup is regularly performed on most systems. Often, however, restoration is tested for the first time when it is required because of a system corruption or hard disk failure. Sound backup and restoration procedures are critical for reconstructing systems after a disruptive event.

34. Ensure that backup procedures and capacity are appropriate for the respective systems.

Typically, backup procedures come in the form of backup schedules, tape rotations, and an offsite storage process. Depending on the maximum tolerable data loss (often referred to as recovery point objective, or RPO), system backup schedules could be as frequent as real time or as infrequent as monthly. If systems are backed up and/or taken offsite less frequently than required on critical systems, an unacceptable amount of data will be lost in the event of a system failure or disaster.

Backup schedules typically are one week in duration, with full backups normally occurring on weekends and incremental or differential backups at intervals during the week. Tape rotations generally are six to ten weeks in duration. Therefore, the organization will have the opportunity, for example, to retrieve a six- or eight-week-old version of a file if needed. This can be critical if a file corruption isn't discovered until more than a week after the corruption occurred.

How

Determine whether systems are backed up periodically and the backups are stored offsite in a secured location. Verify that processes are in place to determine the appropriate frequency of backup for each system in the data center, based on RPO, and to ensure the backup media have adequate space to store the appropriate system contents. Verify that backups are being performed and taken offsite in alignment with organizational backup practices and the requirements of each system. System backup procedures and logs can be obtained from data center staff. Consider retrieving and reviewing a sample of backup system logs.

35. Verify that systems can be restored from backup media.

There is no reason to back up information unless restoration is possible; unfortunately, however, organizations rarely test backup media to ensure that system restoration works properly. Backup media failure rates are high, especially with magnetic tapes. If it is not possible to restore from backup media, data will be lost.

How

Observe evidence of periodic testing of restoration procedures. Alternatively, ask a system administrator to order backup media from offsite storage facilities and observe the restoration of data from the media to a test server. Review the restoration logs to verify that all files were restored.

36. Ensure that backup media can be retrieved promptly from offsite storage facilities.

Often, backup media cannot be retrieved from offsite storage facilities. This is due to backup media being marked improperly or placed in the wrong location. This situation can cause either undue delay in restoring systems or a complete loss of data.

How

Verify that backup media can be retrieved within the time frames set forth in the service level agreement with the offsite storage vendor. This can be accomplished by reviewing the logs from recent retrieval requests or requesting retrieval during the audit and measuring the results. Also, ensure that a perpetual inventory is maintained of all tapes stored offsite.

Disaster Recovery Planning

The goal of disaster recovery planning is to reconstitute systems efficiently following a disaster, such as a hurricane or flood.

37. Ensure that a disaster recovery plan (DRP) exists and is comprehensive and that key employees are aware of their roles in the event of a disaster.

If a disaster strikes your only data center and you don't have a DRP, the overwhelming odds are that your organization will suffer a large enough loss to cause bankruptcy. Thus, disaster recovery is a serious matter.

How

Auditing DRPs can be difficult because of the complexity of successfully recovering data center operations. Perform the following steps:

- Ensure that a DRP exists.
- Verify that the DRP covers all systems and operational areas. It should include a formal schedule outlining the order in which systems should be restored and provide detailed step-by-step instructions for restoring critical systems. These instructions should provide sufficient detail that they could be followed by most any system administrator.
- Review the last data center threat assessment to verify that the DRP is still relevant and addresses the current risk to the data center.
- Ensure that disaster recovery roles and responsibilities are clearly defined.
- Verify that salvage, recovery, and reconstitution procedures are addressed.
- If an emergency operations center is used, verify that it has appropriate supplies, computers, and telecommunications connectivity.
- Ensure that emergency communications are addressed in the plan. This should include a contact list of all personnel to be notified in the event of a disaster, along with phone numbers. Personnel to be notified of a disaster could include key decision-making personnel, personnel who will be involved in the recovery, equipment vendors, and contacts at alternative processing facilities.

- Verify that the DRP identifies a critical recovery period during which business processing must be resumed before suffering significant or unrecoverable loss (often referred to as the recovery time objective, or RTO). Validate that the plan provides for recovery within that period.
- Determine whether the plan includes criteria for determining whether a situation is a disaster and procedures for declaring a disaster and invoking the plan.
- Verify that a current copy of the DRP is maintained at a secured, offsite location.
- Review the results of the last disaster recovery exercise.

This information can be obtained from reviewing the actual DRP or from interviewing the data center facility manager or disaster recovery planner.

38. Ensure that DRPs are updated and tested regularly.

If plans are not tested, there is no assurance that they will work when needed. Plans should be tested and updated at least annually, sometimes more frequently for organizations that are upgrading or procuring new systems, conducting mergers or acquisitions, or adding new lines of business. Failure to update or test DRPs will result in slower recovery times in the event of a disaster.

How

Review the update or version history that usually is included in the front of the plan. Plans should be updated at least annually or per your company's policy. Likewise, review disaster recovery test documentation to verify that tests are performed at least annually or per your company's policy. This information usually accompanies the plan in either electronic or paper form.

39. Verify that parts inventories and vendor agreements are accurate and current.

When disasters occur, organizations are faced with the task of recovering from scratch systems that often are completely destroyed. This requires hardware, software, and backup media. To speed up the process, data centers should keep certain equipment (such as servers and parts) at offsite facilities and enter into vendor agreements to receive expedited equipment in the event of a disaster. Often this spare equipment will be kept at a "hot site" or a "warm site," where systems are available and ready to use at an alternative data center to expedite recovery. The main difference between hot and warm sites is generally how in sync the data at the site is with production data and how quickly the site can be production ready. A hot site will be operational more quickly than a warm site but is also a more expensive option.

How

Review spare equipment inventories and vendor agreements to ensure that both are current for existing systems. Vendor agreements should accompany the DRP. Spare equipment inventories can be obtained from asset management or system personnel.

40. Ensure that emergency operations plans address various disaster scenarios adequately.

Several types of disasters can occur at a data center. The common ones include fire, flood, and other weather-related events. Different types of events will require different salvage and recovery efforts. Emergency operations plans should reflect any reasonably anticipated scenario. Inaccurate emergency operations plans increase recovery times.

How

Verify that any reasonably anticipated scenario is covered by emergency operations plans and that those plans accurately reflect specific needs relating to each scenario. This analysis can be performed by interviewing disaster recovery planners or simply by reviewing emergency operations plans.

41. Determine whether business teams are appropriately involved with the DRP and whether the DRP is tied in with the company's overall business continuity planning.

The DRP ultimately exists in order to support the recovery of business operations in the event of a disaster affecting the data center. Without the involvement of knowledgeable business personnel to provide requirements for recovery, the amount of time it takes to recover operations and/or the amount of data lost during a disaster might be out of alignment with business needs.

Business continuity planning (BCP) is sort of like a disaster recovery plan for the overall business, defining processes and contingencies for the business to recover and minimize impact from serious incidents and disasters. The DRP is often considered a subset of BCP, dealing with the IT aspects of recovery. If there is no tie between the company's DRP and BCP, it could lead to communications gaps during the recovery from an actual event.

How

Look for the existence of a business impact analysis (BIA) for the data center. A BIA is performed to obtain input from business users regarding the impact to the business in the event of an extended outage at the data center. This drives the engineering of the data center's backup and recovery mechanisms. Note that depending on the scale of the data center, there could be one BIA for the entire facility or there could be a series of BIAs, each covering a specific utility/application (or subset of utilities and applications, grouped by business area).

At a minimum, look for documented requirements regarding data center RTO, which dictates how quickly the data center needs to be back up after a disaster, and RPO, which dictates how much data the business can afford to lose in the event of a disaster.

Look for evidence that the appropriate business leaders were involved in the BIA(s) and signed off on the results, specifically on the RTO and RPO.

Through interviews with the data center facilities manager or disaster recovery planner, determine how the DRP is tied in with the company's BCP, particularly focusing in on documented plans for communications with the BCP process during an actual disaster event.

Knowledge Base

Several additional resources offer information about data centers and related controls. A number of good websites provide information about potential hazards (such as flood hazards) for specific geographic areas and general information on emergency and disaster activities:

- https://hazards.fema.gov
- https://msc.fema.gov
- www.fema.gov
- www.noaa.gov
- https://earthquake.usgs.gov

The Green Grid is a consortium of IT companies and professionals seeking to improve energy efficiency in data centers. Some useful background and guidelines for data center power efficiency can be found at its website at www.thegreengrid.org.

Disaster recovery is a deep discipline. While we touched on best practices and provided high-level audit procedures, several resources can be used by auditors for additional information, including the following:

Resource	Website
The Disaster Recovery Journal	www.drj.com
Disaster Recovery Institute International	www.drii.org
Disaster Recovery World	www.disasterrecoveryworld.com
ISACA	www.isaca.org

Master Checklists

The following table summarizes the steps listed herein for auditing data centers and disaster recovery.

Auditing Data Centers and Disaster Recovery

Checklist for Auditing Data Centers and Disaster Recovery
❑ 1. Review data center exterior lighting, building orientation, signage, fences, and neighborhood characteristics to identify facility-related risks.
❑ 2. Research the data center location for environmental hazards and to determine the distance to emergency services.
❑ 3. Review data center doors and walls to determine whether they protect data center facilities adequately.
❑ 4. Evaluate physical authentication devices to determine whether they are appropriate and are working properly.
❑ 5. Ensure that physical access control procedures are comprehensive and being followed by data center and security staff.
❑ 6. Ensure that intrusion alarms and surveillance systems are protecting the data center from physical intrusion.
❑ 7. Review security guard building round logs and other documentation to evaluate the effectiveness of the security personnel function.
❑ 8. Verify that sensitive areas within the data center are secured adequately. Ensure that all computer processing equipment essential to data center operations (such as hardware systems and power supply breakers) is located within the computer processing room or in a secure area.
❑ 9. Verify that HVAC systems maintain constant temperatures within the data center.
❑ 10. Ensure that a water alarm system is configured to detect water in high-risk areas of the data center.
❑ 11. Determine whether the data center has redundant power feeds.
❑ 12. Verify that ground-to-earth exists to protect computer systems.
❑ 13. Ensure that power is conditioned to prevent data loss.
❑ 14. Verify that battery backup systems are providing continuous power during momentary blackouts and brownouts.
❑ 15. Ensure that generators protect against prolonged power loss and are in good working condition.
❑ 16. Evaluate the usage and protection of emergency power-off (EPO) switches.
❑ 17. Ensure that data center building construction incorporates appropriate fire suppression features.
❑ 18. Ensure that data center personnel are trained in hazardous materials (hazmat) handling and storage and that hazmat procedures are appropriate. Also determine whether data center personnel are trained in how to respond to a fire emergency or hazardous material spill.
❑ 19. Verify that fire extinguishers are strategically placed throughout the data center and are maintained properly.
❑ 20. Ensure that fire suppression systems are protecting the data center from fire.
❑ 21. Verify that fire alarms are in place to protect the data center from the risk of fire.
❑ 22. Review the alarm monitoring console(s), reports, and procedures to verify that alarms are monitored continually by data center personnel.
❑ 23. Verify that network, operating system, and application monitoring provides adequate information to identify potential problems for systems located in the data center.
❑ 24. Ensure that roles and responsibilities of data center personnel are clearly defined.
❑ 25. Verify that duties and job functions of data center personnel are segregated appropriately.
❑ 26. Ensure that emergency response procedures address reasonably anticipated threats.
❑ 27. Verify that data center facility-based systems and equipment are maintained properly.
❑ 28. Ensure that data center personnel are trained properly to perform their job functions.
❑ 29. Ensure that data center capacity is planned to avoid unnecessary outages.
❑ 30. Verify that procedures are present to ensure secure storage and disposal of electronic media.
❑ 31. Review and evaluate asset management for data center equipment.
❑ 32. Ensure that hardware redundancy (redundancy of components within a system) is used to provide high availability where required.
❑ 33. Verify that duplicate systems are used where very high system availability is required.
❑ 34. Ensure that backup procedures and capacity are appropriate for the respective systems.
❑ 35. Verify that systems can be restored from backup media.
❑ 36. Ensure that backup media can be retrieved promptly from offsite storage facilities.
❑ 37. Ensure that a disaster recovery plan (DRP) exists and is comprehensive and that key employees are aware of their roles in the event of a disaster.
❑ 38. Ensure that DRPs are updated and tested regularly.
❑ 39. Verify that parts inventories and vendor agreements are accurate and current.
❑ 40. Ensure that emergency operations plans address various disaster scenarios adequately.
❑ 41. Determine whether business teams are appropriately involved with the DRP and whether the DRP is tied in with the company's overall business continuity planning.

CHAPTER 6

Auditing Networking Devices

The network is the backbone of your IT infrastructure, enabling the transmission of data between users, data storage, and applications. Routers, switches, and firewalls work together to enable data transfer while protecting networks, data, and end users. Wireless networks extend these capabilities for mobile users or in places where the traditional infrastructure is too costly or undesirable. As network gear has evolved and become more powerful, new capabilities have been added, and an entire market segment known as the next-generation firewall (NGFW) has emerged. This chapter discusses how to review these critical pieces of your infrastructure while helping you to do the following:

- Unravel the complexity of network equipment
- Understand critical network controls
- Review specific controls for network gear, including routers, switches, firewalls, and wireless components

Background

Today we take it for granted that computers and other devices can connect with each other almost anywhere the world, enabling online shopping, communications, social networking, and more. The Internet, as complex as it is, is regarded by many people as a utility, like water or electricity. How did we get here? How did we end up with a global, interconnected network, and how does your corporate network fit into the map?

The idea of computers talking to each other emerged almost as soon as the first computers were developed. It was nothing novel as a concept—the telegraph, enabling electrical communication between distant points, had been around for over a hundred years by then. But governments and military organizations, realizing the potential in electronic computing, began to study real-world applications of networking and started developing the building blocks of what we know today as the Internet.

One of the earliest efforts started in 1962, when Paul Baran of the RAND Corporation was commissioned by the U.S. Air Force to study how to maintain control over aircraft and nuclear weapons after a nuclear attack. A robust type of communication was needed, something more fault-tolerant than existing methods. Baran and others working on similar projects advanced the ideas of blocks of data operating on "packet-switched" networks that would allow data to be moved around via different routes rather than be subject to single points of failure.

These concepts were incorporated into and further developed with the construction of ARPANET (Advanced Research Projects Agency Network) in 1969. ARPANET initially consisted of 50-kbps circuits connecting four network nodes: the University of California at Los Angeles, Stanford Research Institute (SRI), the University of California at Santa Barbara, and the University of Utah.

Soon other groups began working on their own similar projects, and a new problem arose: different projects developed different protocols, and there was no simple way to allow one cluster of nodes to talk to another. This challenge drove the creation of what later became known as the *Transmission Control Protocol/Internet Protocol* (TCP/IP). The proposal for this protocol included the first use of the word "internet" to describe interconnected network communication. TCP/IP allowed different networks to talk to each other, no matter what protocols they used internally. TCP/IP took over as the foundation for ARPANET in 1983, replacing other protocols in use.

Beginning in the 1980s, commercial Internet service providers (ISPs) emerged, and the fledgling Internet began to be influenced more by the private sector. With the basic concepts complete, governments and research agencies stepped into the background, and the Internet took off. Today the Internet serves as the communications backbone of every major company on the planet, and the TCP/IP protocol is fundamental to corporate networks everywhere. Payroll systems, logistics, manufacturing, sales systems, websites, and more are all enabled by TCP/IP.

Starting in the 1990s, the Internet began to go wireless. In 1997, the Institute of Electrical and Electronic Engineers (IEEE) issued the 802.11 standard for wireless networking. The original 802.11 is not used much today, but its descendants are still going strong. The 802.11n and 802.11ac standards drive the prevalent platforms used in homes and businesses today. Wireless is everywhere, from laptops and printers, to smartphones and watches, to thermostats and refrigerators.

Larger enterprise networks today can look a lot like miniature models of the Internet. If you have, for example, a main campus, two satellite offices, and a data center, they may all act like small local networks communicating with each other using TCP/IP. And importantly, they most likely have similar network components at each location facilitating all of those communications.

In the next section we will uncover how these components work using simple explanations and analogies.

Network Auditing Essentials

Networks enable hosts to communicate using specialized hardware or software optimized for delivering data from one host to another. Fundamentally, the hardware is a computer running an operating system designed to move data. Network devices such as routers, switches, and firewalls have many of the same basic components you would find in your typical server, except they are highly customized. These devices contain specialized processors with embedded instructions designed to process data movement in a fast and efficient manner. They also have memory, an operating system, and a means for configuring the device. Today many types of network gear can be virtualized, particularly in cloud environments or in virtual machine infrastructure. While this can add some confusion to your audit effort, you should consider virtualized network gear to have similar requirements as their physical counterparts.

Networking giants in recent years have answered the call for simplicity and created slick graphical user interfaces (GUIs) to complement the sometimes-daunting command-line interfaces (CLIs) used to interact with, and configure, network devices. However, regardless of the method, you are still configuring the operating

system for a device that essentially is a computer designed to move data.

Let's start our review of network auditing essentials with a discussion covering protocols and the *International Standards Organization*'s (ISO) *Open System Interconnection* (OSI) model to gain a better understanding of routers, switches, and firewalls. This review will help you work with your network team to audit your networking environment. We will stick to simple analogies and examples while avoiding complex issues. It can take years to master advanced networking concepts. The purpose of this section is to help an auditor who is completely new to networking quickly understand the functionality of routers, switches, and firewalls.

NOTE This chapter does not deal specifically with software-defined networking (SDN) or software-defined wide area networking (SD-WAN). These technologies allow networks to be easily reconfigured from a central control point based on the dynamic needs of users, applications, and workloads. If your organization uses these capabilities, discuss with your network administrator and apply concepts from this and other chapters when assessing the systems.

Protocols

In networking, protocols define the structure or rules by which devices communicate. As an analogy, let's pretend there are two people who each speak several different languages, and one wants to tell the other his street address. First, they'll introduce themselves (handshake). Then they'll determine with a few words whether to use a common language (message type) or to use an interpreter. Next, they'll begin exchanging the actual information of interest (payload). Even the street address itself has rules and a specific order—you wouldn't state your address as "USA 123 Anytown Street Main." All of these ideas—handshake, message type, payload—make up the protocol needed to exchange that street address. In networking, the concepts are similar, if a bit more abstract.

If you've been around computers and networking, you may have noticed that there are more protocols than just TCP/IP. Why? Each protocol has features designed into it to make the protocol more efficient at communicating specific types of data or allowing for specific functions.

OSI Model

The seven-layer OSI model describes how data moves from one system to another system. This model helps describe how to build applications, protocols, and equipment that move data from your application to the physical wire, across hundreds or thousands of miles, to an application on the other side.

Two common layered models are the ISO OSI model and the TCP/IP model. The TCP/IP model has five layers that loosely relate to the seven layers in the ISO OSI model. For the purposes of this chapter, we will discuss and stick with the ISO OSI seven-layer model (Table 6-1). Keep in mind that this is just a model and that real implementations of protocols do not always align perfectly with the seven steps that follow.

Layer	Common Name	Description
Layer 7	Application	Represents the end-user application such as HTTP, File Transfer Protocol (FTP), Simple Mail Transport Protocol (SMTP), or Telnet.
Layer 6	Presentation	Handles formatting, encryption, compression, and presentation of data to the application. Examples include Secure Sockets Layer (SSL) and Transport Layer Security (TLS).
Layer 5	Session	Deals with the setup and management of sessions between computer applications. Examples include named pipes, NetBIOS, and session establishment for TCP.
Layer 4	Transport	Deals with transport issues, such as getting to the destination in one piece, and error control. TCP and User Datagram Protocol (UDP) are perhaps the best-known examples in this layer.
Layer 3	Network	Routes packets between networks. Examples include IP, Internet Control Message Protocol (ICMP), IP Security (IPSec), and Address Resolution Protocol (ARP). Routers operate at this layer typically using IP addresses.
Layer 2	Data Link	Links data on hosts from one location to another, typically on the local area network (LAN) but sometimes on the wide area network (WAN) too. Examples include Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), Frame Relay, and Asynchronous Transfer Mode (ATM). Switches and bridges operate at this layer, typically using Media Access Control (MAC) addresses. The wireless 802.11 protocols also operate at this layer.
Layer 1	Physical	Defines the physical link, cabling, and binary transmission. Modulation and flow control occur at this layer.

Table 6-1 Simplified OSI Model Description

As data moves through a network, the different layers of the OSI model work together to move the information along. Network transmission begins with layer 7, the application layer. Each layer adds a few bits of header information specific to the protocols it manages, then passes the payload to the next layer. That layer repeats the process. When the physical layer processes the payload, the information is sent to the receiver. The receiving system then begins the reverse process, using the bits of header information and discarding them, then handing the remaining payload to the next layer. The process of adding these different header bits to a chunk of data is called encapsulation. You can see a representation of this in Figure 6-1, where the various layers are adding their own header bits, designated H7, H6, etc., for each layer, to the initial payload.

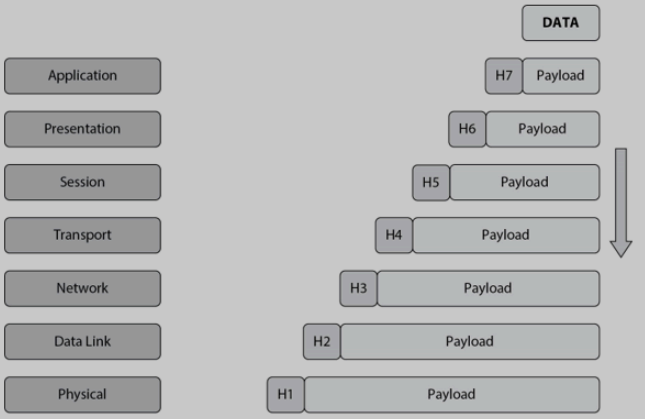


Figure 6-1 Data encapsulation

Routers and Switches

Two key hardware components of networks are switches and routers. Routers connect and route data between networks using layer 3 network addresses, usually IP addresses. Routers operate at OSI layer 3. Once data is routed to the destination network, it goes to a switch where the destination host resides. The switch uses the destination host's MAC address, at OSI layer 2, to send data the rest of the way to the host. Some network devices can combine the capabilities of switching and routing and operate at multiple OSI layers. Let's take a step back and discuss routers and switches, starting with an analogy.

An Oversimplified Switching and Routing Analogy

Consider a local school full of classrooms. Students can yell their names out to everyone in their own class, and everyone in the classroom can hear. The classroom is their own *broadcast domain,* and it doesn't take long for the people in the classroom to know each other's personal names. However, several communication challenges arise when the size of the classroom, or *broadcast domain,* becomes too large. There can be too many voices, and conversations are misunderstood or have to be repeated. To organize communications, let's say that anytime a student wants to talk to another student, she has to first talk to the teacher, and the teacher will relay the message. The teacher is now serving the function of a network switch, responsible for knowing each student and making sure the messages get to the right place.

Great, now our classroom is organized into students (endpoints, or network nodes) and a teacher (switch). Next, we need to determine how our students can talk to students in other classrooms. Let's assign a hall monitor to handle these messages and serve as our *router*. A student (sender) has a message for another student (recipient). She tells the teacher, who realizes the recipient isn't in the same classroom (on the same switch). The teacher gives the message to the hall monitor, who knows all of the other teachers, and takes the message to the right classroom and hands it off to the teacher. The message reaches the recipient, and the process continues.

With this analogy in mind, let's talk about some of the specific devices you'll see on a network.

Switches

A network switch handles messages to and from devices. It learns the unique identifier (MAC address) of each device connected to it and ensures that messages intended for a certain recipient are sent only to that recipient. If it doesn't know the recipient, it will send it up to the next level of the network for processing. Everything at the switch level typically is handled with the MAC address, represented by OSI layer 2.

If you connect a couple of devices to a switch, you've got a local area network, or LAN. LANs are typically limited in size to a small area, like a building or campus. You can even connect switches to other switches to increase the size of a LAN. Suppose, though, that you want to talk to a host in another campus. Your switches will have to be connected to a router.

Routers

Routers handle data traveling from one network to another, forwarding packets to other routers or to switches connected locally. Eventually, the packet reaches the remote LAN and then finally arrives at the host on the other side. Each router between you and the remote host simply looks at the IP address header information, located at layer 3, to see where to send it next.

Features specific to routers enable them to communicate across the Internet or company network. Routers dynamically build routing tables using protocols such as Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP). These enable the routers to send packets in the direction they need to go to get to the other side as quickly as possible. Routers also may have access control lists (ACLs) and quality-of-service (QoS) features. The association between routers and switches is shown in Table 6-2.

Layer	Name	Equipment Used	Depends On	Example
Layer 3	Network	Routers	IP Address—WAN	198.133.219.25
Layer 2	Data Link	Switches	MAC Address—LAN	00-14-22-F5-04-16

Table 6-2 Routers and Switches

Despite some of the differences between them, switches and routers typically are managed in similar manners using similar syntax and have many of the same concerns from an audit perspective. Keep in mind the purpose of the device as you step through the audit, and this will help you determine what additional steps, if any, you might want to perform.

LANs, VLANs, WANs, and WLANs

The LAN is a simple implementation of a network, but there are many other "area" networks that may come up in conversations with network teams. Let's define a few of them next.

Virtual Local Area Network (VLAN)

Virtual LANs, or VLANs, can be used on most types of switches to further segment networks connected to the switch. Routing between these VLANs can be performed by routers separate from the switch or, in some cases, this function can be integrated into the switch.

VLANs allow network administrators to create segregated networks based on levels of trust or types of traffic. For example, you can create a separate VLAN for managing sensitive hosts and prevent general access to the management console of network equipment or sensitive appliances. Breaking up the network into smaller VLANs also helps reduce the number of broadcasts that individual hosts are required to process, and VLANs also allow network administrators to move a host with a logical change in the switch rather than a cable move.

VLANs are extremely powerful. With a VLAN, a physical switch can be separated into multiple logical switches, each managing its own VLAN. Additionally, VLANs can be shared within the same network among multiple physical switches and routers, allowing two geographically separated devices to exist logically on the same virtual network.

Wide Area Network (WAN)

A WAN is designed to cover a very large geographical area, perhaps an entire region or even the entire globe. WANs are used to carry traffic between LANs and

use different transmission methods due to the large distances covered.

Wireless Local Area Network (WLAN)

Often referred to as Wi-Fi, the wireless LAN permits transmission of network data by electromagnetic waves, freeing endpoints from the limitations of a network cable. WLANs are often deployed using access points, physical devices containing the transmission equipment needed to translate wireless data into physical signals and allow users to access the rest of the network.

Wireless networks require additional consideration around authentication management and intrusion detection. Special audit steps for wireless networks are discussed later in the chapter.

Firewalls

While routers and switches are designed to send traffic from one network node to another, network firewalls are designed primarily to serve as a barrier, establishing rules to control the flow of traffic. Firewalls can help establish a protected area of your network that is accessible only to certain systems or via certain methods. They may be used to isolate sensitive areas of your network in order to reduce the risk of unauthorized access. Firewalls are most commonly used as perimeter protection for networks, keeping undesired traffic out.

In a very simple network model, an organization might deploy two firewalls to defend its networks from unwanted Internet traffic. One firewall might be placed at the boundary of the company network, directly facing the Internet. This would be configured to block all incoming packets except those destined for designated, Internet-accessible destinations, such as the company's public website. Another firewall might be placed at the boundary of the company's internal network, blocking any traffic coming from the company's public website or other systems into the internal network. In this way, the network is broken up into different security areas, sometimes called "trust zones." The portion of a company network closest to the Internet is usually called its demilitarized zone, or DMZ. Although it is protected by one or more firewalls, a DMZ is considered a less-trusted zone because it is directly accessible from the Internet, which means it is also often subject to direct attack. A trust zone such as the internal network might be considered a safe area for systems in the same segment of the network, but a firewall might be used to manage traffic coming from less-trusted zones, such as a DMZ. An example of a simple, two-firewall configuration is depicted in Figure 6-2.

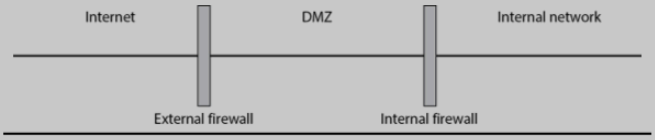


Figure 6-2 Simple firewall placement

There are many different types of firewalls today, available from dozens of different vendors. However, all are essentially designed to block unwanted traffic. For the purposes of this book, we will stick to the definitions in the National Institute of Standards and Technology (NIST) Publication 800-41-Rev1.

NOTE NIST Publication 800-41-Rev1, "Guidelines on Firewalls and Firewall Policy," can be found at http://csrc.nist.gov/publications/PubsSPs.html.

Packet Filtering Firewalls

Packet filtering firewalls are essentially routers operating at layer 3 using set ACLs. Decisions are made to allow and disallow traffic based on the source and destination IP address, protocol, and port number.

Stateful Inspection Firewalls

Stateful inspection, also stateful packet inspection (SPI) or dynamic packet-filtering firewalls, operate at layers 3 and 4. Your router at home allows you to establish and maintain a session externally with another address. The "state" refers to identifying and tracking sessions that occur in layers 4 and 5. The rules are changed dynamically when you establish an outbound connection to enable packets from the destination IP address to be returned to you. All other traffic is stopped from reaching your computer, protecting you from the dangers inherent in the Internet.

Application Firewalls

Application firewalls combine the functionality of the typical firewall operating in the lower OSI layers with the power and deep inspection of application awareness. Now, based on information at the application level, such as known malicious traffic, decisions can be made to allow or disallow traffic. An example might be an appliance or host that screens web traffic before it hits your web server. Based on the behavior and content of the web traffic, decisions might be made dynamically to refuse access to the web server.

Application-Proxy Gateway

Application-proxy gateways manage conversations between hosts, acting as an intermediary at the application level of the OSI model. Because proxies reestablish conversations to the destination, they effectively can hide the source of a conversation. Proxies might enforce authentication, logging, or content rules. One of the advantages of application-proxy gateways is the potential ability to stop an encrypted session, decode the data, read the data in cleartext, encode the data, and then reinitiate an encrypted session to the destination. This is extremely resource-intensive, and performance requirements become a concern for these applications.

Additional Firewall Technologies

Additional types of firewalls discussed in greater detail in NIST Publication 800-41-Rev1 demonstrate the specialization that has occurred in the firewall market. These firewall technologies include dedicated proxy servers, virtual private networking, network access control, unified threat management, web application firewalls, and firewalls for virtual infrastructures.

NOTE Packet filtering firewalls, stateful inspection firewalls, and dedicated proxy servers fall generally into what is termed "traditional" firewall technology. Firewalls operating at higher levels of the OSI stack, such as application firewalls, are generally considered NGFWs. NIST 800-41 does not include the term NGFW, but that term has been used to describe more advanced firewalls for many years.

Auditing Switches, Routers, and Firewalls

The audit steps are divided into *general steps* and *specific steps.* The general audit steps are applicable to network equipment in general, followed by specific sections for routers, switches, and firewalls. Work through the first section of general controls, regardless of your audit, and then move to the specific section(s) you need to complete the audit.

NOTE The general networking audit steps need to be performed, regardless of what networking device you audit. These steps apply to routers, switches, and firewalls, regardless of what layer they operate on and regardless of where they are located in your network.

General Network Equipment Audit Steps

Begin the audit by asking the network engineers for a copy of the configuration file and the version of the device you intend to audit. For routers and switches, nearly all of the information you want is located in the configuration file, and having this prevents you from having to log onto the device repeatedly. Modern switches and other network gear supply web-based management interfaces that can make examining a single device's configuration much simpler, but command-line interfaces are present as well. Discuss with your network administrator which access methods are available or preferred. In many environments you may not be permitted to directly access network gear. In these cases, be prepared to ask a network administrator to provide the evidence you need to complete the audit.

NOTE Many of the examples that follow are from the Cisco IOS. Your networking equipment may be different, but the concepts generally are the same. Your network engineers should know when differences occur and can show you adequate supporting documentation so that you feel confident that your network is secure and operating as it should.

1. Review controls around developing and maintaining configurations.

This step is a catch-all that addresses configuration management, the overarching concept of maintaining the secure configuration of the network. Failure to maintain a secure configuration may lead to lapses in technology or processes that affect the security of your network. Review changes on the network devices to ensure that the change did not unintentionally degrade the performance or security of the network.

How

Discuss change management practices with network administrators. Ensure that changes are planned, scheduled, documented (including the purpose of the change), and approved prior to implementation. Ensure that the company's configuration change management policies and processes are followed. See Chapter 3 for more information.

Note that this step lightly covers routine patch cycles, which is specifically covered again in step 2. Discuss the following as applicable with the administrator to ensure that proper configuration management controls are in place:

- Security mailing lists are monitored.
- The latest patches are applied in a routine patch cycle under the guidance of written and agreed-on policies and procedures.
- A configuration guideline exists for the equipment in the environment and is strictly followed. Exceptions are carefully documented and maintained.
- Regular vulnerability scanning from both internal and external perspectives is conducted to discover new risks quickly and to test planned changes to the environment.
- Regular internal reviews of the configuration are conducted to compare the existing infrastructure with the configuration guide.
- Regular status reports are issued to upper management documenting the overall security posture of the network.

Having a strong configuration standard is critical to a secure network. Network equipment, including routers, switches, and firewalls, has many configuration options that affect security and are rarely secure out of the box. Taking the time to understand these options and how to configure them to your environment is fundamental to maintaining a sound and secure network.

Numerous configuration and security hardening guides are available for network devices. A few resources are listed later in this chapter in the Knowledge Base. If your organization does not utilize a particular hardening guide, you should ensure that the principles in those documents are covered in your company's configuration processes.

2. Ensure that appropriate controls are in place for any vulnerabilities associated with the current software version. These controls might include software updates, configuration changes, or other compensating controls.

This step goes beyond configuration changes and targets specifically software updates and any associated vulnerabilities. This step is where you as an auditor will research critical vulnerabilities associated with the software and ensure that appropriate controls are in place, such as a software update, a configuration change, or other compensating control.

Note that it isn't necessary to install each and every update, but you generally should keep your network equipment current. As vulnerabilities become known to the security community, they are documented in various online databases such as the National Vulnerability Database (NVD) located at http://nvd.nist.gov. These lists should be checked, and if the version of code being used is found to have some known vulnerabilities, the device should be patched or have other mitigat-

ing controls employed to protect the network device and your network.

How

Discuss the software version information with the network administrator and the status of any pending patches or upgrades. Check the software and version against the NVD. Note and discuss any potential issues with the network administrator.

3. Verify that all unnecessary services are disabled.

Running unnecessary services can leave you susceptible to performance- and security-related risks. This is true of any host or device and adds to the attack surface available to potential attackers.

How

Discuss unnecessary services with the network administrator, and review the configuration of the device. If the device depends on another platform (for example, some firewalls), ensure that the underlying platform also has all unnecessary services disabled.

Discuss any exceptions with the administrator, and determine what additional risk exposure might exist and whether exceptions are necessary. For any other services enabled, discuss with the administrator to verify that there is a legitimate business need for the service. Services should be enabled only when needed. Refer to the vendor's website for the best source of required services and those that might be considered security risks. Note the presence or absence of these services.

4. Ensure that good SNMP management practices are followed.

Simple Network Management Protocol (SNMP) represents an often-overlooked way to obtain full administrative access to a network device. This step may not be applicable to your equipment if the equipment doesn't support SNMP management or it is disabled.

How

Discuss SNMP management practices with your network administrator. SNMP Versions 1 and 2 send password information in cleartext, and the packets are unauthenticated. SNMP Version 3 adds message integrity, authentication, and encryption to the packets. However, all versions have suffered from security issues. Refer to Cisco's website and carefully review the suggested compensating controls listed under "Workarounds" at https://www.cisco.com/c/en/us/support/docs/csa/cisco-sa-20080610-snmpv3.html.

SNMP community strings (passwords) should follow standard password policies for strength and change frequency. Management with SNMP should be restricted with an access list, and no management should be allowed from an untrusted network.

5. Review and evaluate procedures for creating user accounts and ensuring that accounts are created only when there is a legitimate business need. Also review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.

This step has a wide scope, covering controls around account usage and management. Inappropriately managed or used accounts could provide easy access to the network device, bypassing other additional security controls to prevent malicious attacks. This step should cover policies and procedures that are essential to ensure that only authorized administrators can log into a network device and that, once logged in, they have the proper privilege level. Login procedures should adhere to strong authentication, authorization, and accounting (AAA).

How

Discuss with the administrator and verify with the administrator's help that appropriate policies and procedures exist to add and remove account access to the device. Accounts should be controlled such that only those authorized to have access can log onto the device. Unused accounts, if applicable, should be removed from the configuration of the network device or completely disabled in accordance with your organization's account management policies.

Also review the process for removing accounts when access is no longer needed. The process could include a periodic review and validation of active accounts by system administrators and/or other knowledgeable managers. Obtain a sample of accounts and verify that they are owned by active employees and that those employees' job positions have not changed since the account's creation.

In general, accounts should never be shared among administrators. This can present risk, in that you lose accountability for actions taken on the system. Strong account policies always should be enforced by the network device. Additionally, discuss login procedures with the administrator to ensure that all users are managed appropriately using roles and that actions are logged appropriately. Individual IDs should be created for every person requiring access to the network device, or the network device should use a central authentication server for these accounts, sometimes called an *AAA server.* RADIUS (Remote Authentication Dial-In User Service) and TACACS+ (Terminal Access Controller Access-Control System Plus) are examples of protocols used for AAA services.

Some network infrastructures may use TACACS+ for AAA on routers and switches to enable integration with Active Directory or LDAP for single sign-on but require an additional "enable" password for privileged access. An example configuration file entry if TACACS+ servers are used might be this:

```
tacacs-server host <IP Address>
```

Individual IDs might look something like this in the configuration file:

```
username <name> password 5 <encrypted password>
```

These IDs should be created with a privilege level of 1 (which is the default), forcing the enable password to be required for additional access.

6. Ensure that appropriate password controls are used.

Weak and unencrypted passwords allow attackers to guess or read passwords easily in plaintext. Strong password controls are essential to protecting network equipment. Older versions of network software allowed storing passwords in cleartext by default. You probably won't see this, but you should verify that passwords are securely stored with the administrator. Cisco recommends that an AAA server be used with network devices; this allows password policies and controls to

be applied centrally.

How

Discuss password controls with the network administrator, and consult existing policies and procedures. Ensure that complex passwords are used and are changed with appropriate frequency (such as every 90 days). Passwords used for privileged modes of operation should never be the same as any other password used on the device.

Finally, ensure that appropriate controls exist so that the same password isn't shared on a large number of devices throughout your network or across trust zones.

7. Verify that secure management protocols are used where possible.

Telnet sends all its information in cleartext, allowing passwords and other information to be viewed with a sniffer. SNMP Versions 1 and 2 are similar. Secure alternatives are Secure Shell (SSH), IPSec, and SNMPv3. While it is imperative that secure protocols be used on untrusted networks, it is also important on more trusted zones as well.

How

Discuss management procedures with the network administrator and review the configuration of the network device. Ensure that policies and procedures exist to manage routers, switches, and firewalls as securely as possible. SSH Protocol, another secure management protocol, or an out-of-band management system should be used for remote administration.

Many modern network devices can be managed via browser by accessing a web interface on the device. Ensure that any web services used for device management are configured to use Hypertext Transfer Protocol Secure (HTTPS) to protect transmission of sensitive data.

8. Ensure that current backups exist for configuration files.

Keep copies of all network device configurations in a readily accessible, secure location—this is critical! These files should contain comments that can help give perspective to the configuration settings and filters. You can change filters with much more ease and accuracy when you can refer back to the old configuration files. These backups also can be invaluable for diagnosing and recovering from unexpected network failures.

How

Discuss policies and procedures with the network administrator and ask to see where the current configurations are kept. Verify with the administrator that the backup repository contains the latest configurations shown on the routers and switches. The configuration files should be stored in a secure location to which only the network team and appropriate administrators have access.

9. Review logging and monitoring processes for network equipment.

Logs should be collected for AAA and other key events and should be sent to a secure host to prevent tampering with the information. Failure to keep logs or monitor them appropriately may prevent administrators from properly diagnosing a network issue or detecting malicious behavior.

How

Interview the network administrator and review any relevant documentation to get an understanding of logging and security monitoring practices. Some level of monitoring is important, but the monitoring level required should be consistent with the criticality of the system and the inherent risk of the environment (for example, alerts related to unusual traffic in the DMZ may carry a higher criticality than normal traffic in a regional sales office). Ensure that logs are sent to a centralized repository.

Review the log levels for network gear and the retention period. On Cisco equipment, log levels range from 0 to 7, where level 0 includes only critical system events and level 7 is considered debug-level logging. Ensure that log retention complies with company policies.

If security monitoring is performed, assess the frequency of the monitoring and the quality with which it is performed. Look for evidence that the security monitoring tools are actually used. Review recent alerts and determine whether they were investigated and resolved.

Note that some organizations may configure network event logs to be sent to centralized logging environments, to be reviewed by dedicated teams or even by outside service providers. In these cases, discuss network monitoring practices with the monitoring team.

10. Evaluate use of the Network Time Protocol (NTP).

NTP provides time synchronization for system operations and helps ensure an accurate timestamp on all logged events. These timestamps are invaluable in reporting and troubleshooting.

How

Discuss the use of NTP with the administrator and review the configuration file. For Cisco devices, NTP is generally enabled by default and is configured to use Coordinated Universal Time (UTC). You can review the status of NTP at the command line using the `show ntp status` command.

In more risk-averse environments, devices should be configured to authenticate NTP source systems. Additional NTP information can be displayed using the `show running-config ntp` command.

11. Verify that a banner is configured to make all connecting users aware of the company's policy for use and monitoring.

A warning banner that clearly marks the router or switch as private property and disallowing unauthorized access is essential should a compromise ever result in legal action.

How

Verify with the administrator and a review of the device configuration that all connecting users are made aware of the company's policy for use and monitoring upon connection. Confirm that the message-of-the-day (MOTD) or LOGIN banner does not disclose any information about the company or network device. This information should be reserved for the EXEC banner, which is displayed only after a

successful login.

For Cisco equipment, you would review the configuration file for the `banner motd, banner login, or banner exec` directive. The order of display of the LOGIN and MOTD banners may vary by equipment, so ensure that neither contains sensitive information if present.

12. Ensure that access controls are applied to the console port.

Logical access to the switch can be gained via the console port. Often the console port will have no password for convenience. Ensure that a password is used to provide an additional layer of defense beyond the physical controls. A password is imperative if the location is not physically secure.

How

Discuss access controls with the administrator. Verify that physical access to the console port is protected and, logically, that the console port is password-protected.

13. Ensure that all network equipment is stored in a secure location.

Anyone with physical access to a network device might be able to gain full logical access using well-documented password recovery procedures. Someone also could unplug cables or otherwise disrupt service. Additionally, access should be limited to prevent nonmalicious accidents (such as tripping over a cable) from disrupting service.

How

Visually observe the location of the network equipment and discuss physical access to the equipment with the network administrator. Only authorized individuals should have physical access to network hardware.

14. Ensure that a standard naming convention is used for all devices.

Standard naming conventions make troubleshooting and finding issues easier. Standard naming conventions also help make managing the environment easier as the organization grows.

How

Discuss the naming conventions used with the network administrator.

15. Verify that standard, documented processes exist for configuring network devices.

Documented processes promote repeatability, helping to prevent common mistakes that might lead to a service disruption or network compromise.

How

Discuss documented policies and procedures for configuring network equipment with the network administrator. If possible, verify that the process was followed using a recently deployed device.

16. Evaluate the use of network access control (NAC) technology to ensure the network is accessed in accordance with company policy.

NAC allows organizations to enforce policy restrictions on network access. If NAC is not in use, it may be possible for someone to connect an unauthorized or noncompliant device to the network and gain access to internal systems.

How

Speak with the network administrator to determine the presence and scope of NAC usage in the environment. A fully realized NAC deployment includes network authentication and system profiling to ensure that the device is known, the user is known, and the system is compliant with security policies such as antivirus protection or patch level. Depending on the outcome of various checks performed by network gear or agents on the system, NAC software may also dynamically assign a device's level of network access. For example, if a system is found to be behind on security patches, the NAC software may assign the computer to a quarantine network so that patches may be applied.

Elements of NAC are more commonly found in wireless environments or in conjunction with remote access systems than in wired networks. However, even if an automated network access control solution is not in place, the organization should have processes in place to ensure that unexpected devices cannot access sensitive resources

17. Review the availability and access controls for web-based (GUI) access to network components.

A web interface can simplify access and management of devices, particularly in smaller environments. However, web servers are often attacked by malicious actors looking for vulnerabilities and can present an expanded attack surface if not properly controlled.

How

Discuss the web interface with the network administrator. If the web interface is not used, it should be disabled. Access to the web server should be restricted to those with a verified need.

In more risk-averse environments, access to the web interface should be limited to specific origin hosts (sometimes called hop hosts or jump hosts). Multifactor authentication provides an additional layer of protection and should be present in more secure environments. Together, these reduce the likelihood that an attacker will be able to access the interface.

18. Ensure that web sessions are configured with appropriate idle and session timeouts.

All web servers, particularly those used for administrative tasks, should be configured to automatically log a user out after a period of inactivity or after a specific duration.

How

Review the web interface configuration with the network device administrator. Ensure that the web server has appropriate idle and session timeouts configured.

For more information about web interface security, see Chapter 9.

19. Review disaster recovery plans related to network devices.

A complex, global network should be built for resilience. If key pieces of the network infrastructure are not comprehended in the organization's disaster recovery plan, the business could suffer a worse-than-expected disruption or recover to an insecure state. Since they can cover large areas without physical cabling, wireless networks can be an essential component of an organization's disaster recovery plan.

How

Discuss the network architecture and disaster recovery with the network administrator. Review key elements of network infrastructure, such as authentication systems, logging systems, wireless controllers, web access gateways, NAC devices, or other potential single points of failure. Components of the network responsible for managing network access, such as password systems, certificate management systems, or NAC components, must be included in the disaster recovery plan to ensure that the security posture of the network is maintained during a recovery.

You should expect that essential elements of network services have been comprehended in the disaster recovery plan. Ensure that the plan is fully documented and that recovery documentation is stored in a location that would be accessible in the event of a disaster, such as an offsite facility. Review the organization's processes to ensure that recovery plans for networks are exercised periodically.

For a more detailed look at disaster recovery plans, refer to Chapter 5.

Additional Switch Controls: Layer 2

The following are additional test steps for switches, or layer 2 devices.

1. Verify that administrators avoid using VLAN 1.

By default, all ports on a Cisco switch are members of VLAN 1. Avoiding the use of VLAN 1 prevents network intruders from plugging into unused ports and communicating with the rest of the network.

How

Discuss this practice with the administrator and review the configuration file for the use of VLAN 1.

2. Evaluate the use of trunk autonegotiation.

A trunk on a switch joins separate VLANs into an aggregate port, allowing traffic access to either VLAN. Autonegotiation allows devices to determine for themselves whether to form a trunk. Trunks should be defined explicitly and should be present only where intended by the network architect.

How

Discuss with the network administrator and review the configuration file. Ensure that switch ports do not attempt to negotiate trunking protocols. Look for the `switchport nonegotiate` directive in the configuration. Cisco describes a number of best practices in their "Network Security Baseline" documentation around the use of trunking and VLANs. Information on this guide is listed in the Knowledge Base.

3. Verify that Spanning Tree Protocol attack mitigation is enabled (BPDU Guard, STP Root Guard).

The Spanning Tree Protocol (STP) is designed to support networks where multiple paths between hosts may exist while preventing network loops from developing. The switch will learn the network topology and move a port through four stages —block, listen, learn, and forward—as it ensures that an endless loop isn't developing in the network traffic patterns. STP was designed without security in mind, leaving it subject to attacks that could alter the network topology or result in man-in-the-middle scenarios.

How

Discuss with the network administrator and review the configuration file. BDPU (bridge protocol data unit) Guard prevents ports from participating in STP. This should be done for all user-facing switch ports. For access ports, look for the following configuration:

```
spanning-tree portfast
spanning-tree bpduguard enable
```

STP Root Guard can be enabled with the following:

```
spanning-tree guard root
```

4. Evaluate the use of VLANs on the network.

VLANs should be used to break up broadcast domains and, where necessary, to help divide resources with different security levels.

How

Discuss the application of VLANs with the network administrator. Devices at different security levels ideally should be isolated on separate switches or layer 2 devices. For example, if you have equipment that for some reason cannot be protected with the company's standard antivirus software and security patches, you could place that equipment on a separate VLAN.

5. Disable all unused ports and put them in an unused VLAN.

This setup prevents network intruders from plugging into unused ports and communicating with the rest of the network.

How

Discuss this practice with the network administrator.

6. Evaluate use of the VLAN Trunking Protocol (VTP) in the environment.

VTP is a layer 2 messaging protocol that distributes VLAN configuration information over trunks. VTP allows the addition, deletion, and renaming of VLANs on

a network-wide basis. A network attacker could add or remove VLANs from the VTP domain, as well as create STP loops. Both situations can lead to disastrous results that are very difficult to troubleshoot. This type of event can also occur accidentally. A switch with a higher configuration version number in its VTP database has authority over other switches with a lower number. If a lab switch such as this one were placed on the production network, you might accidently reconfigure your entire network.

How

Discuss use of the VTP with the network administrator to ensure that passwords are used if the VTP is necessary. VTP should be turned off if it's not used. The VTP mode of a switch can be server, client, or transparent. Use transparent mode unless client or server is required.

If VTP is necessary, domains should be set up for different areas of the network, and passwords should be enabled. Look for these lines in the configuration file:

```
vtp domain domain_name
vtp password password
```

7. Verify that thresholds exist that limit broadcast/multicast traffic on ports.

Configuring storm controls helps mitigate the risk of a network outage in the event of a broadcast storm.

How

Discuss with the administrator and review the configuration file for the presence of `storm-control [broadcast | multicast | unicast] level.`

Additional Router Controls: Layer 3

The following are additional test steps for routers, or layer 3 devices.

1. Verify that inactive interfaces on the router are disabled.

Inactive interfaces that should be disabled include LAN and WAN interfaces such as Ethernet, Serial, and ATM. Open interfaces are possible sources of attack if someone plugs into the interface.

How

Discuss policies and procedures with the network administrator to ensure that this is a common practice. Ask the administrator for examples. The command `shutdown` is used to disable interfaces.

2. Ensure that the router is configured to save all core dumps.

Having a core dump (an image of the router's memory at the time of the crash) can be extremely useful to tech support in diagnosing a crash and determining the root cause.

How

Discuss how the router handles core dumps with the network administrator. The core dumps should be located in a protected area that is accessible only to the network administrator, because disclosure of important information could occur. You might review the configuration file for something similar to the following. Note that Trivial FTP (TFTP) and Remote Copy Protocol (RCP) also may be options here, but FTP is recommended.

```
ip ftp username username
ip ftp password password
exception protocol ftp
exception region-size 65536
exception dump ip address
```

NOTE Note that core dumps will cause the router to take longer to reboot after a crash because of the time it takes to dump the core file to the server.

3. Verify that all routing updates are authenticated.

Authentication ensures that the receiving router incorporates into its tables only the route information that the trusted sending router actually intended to send. It prevents a legitimate router from accepting and then employing unauthorized, malicious, or corrupted routing tables that would compromise the security or availability of the network. Such a compromise might lead to rerouting of traffic, a denial of service, or simply access to certain packets of data by an unauthorized person.

How

The authentication of routing advertisements is available with Routing Information Protocol (RIPv2), OSPF, intermediate system to intermediate system (IS-IS), Enhanced Interior Gateway Routing Protocol (EIGRP), and BGP. Most allow the use of plaintext authentication or an MD5 hash. The MD5 method should be used to prevent passwords from being sniffed.

RIPv2 authentication is configured on a per-interface basis. Look in the configuration file for something like this:

```
router rip
version 2
key chain name_of_keychain
key 1
key-string string
interface ethernet 0
ip rip authentication key-chain name_of_keychain
ip rip authentication mode md5
```

OSPF authentication is configured on a per-area basis, with keys additionally specified per interface. Look in the configuration file for something like this:

```
router ospf 1
area 0 authentication message-digest
interface ethernet 0
ip ospf message-digest-key 1 md5 authentication_key
```

BGP authentication is configured on a per-neighbor basis. There are various formats, but one example configuration might look like this (MD5 is the only option, so it does not need to be specified):

```
router bgp 1
neighbor ip_address password password
```

4. Verify that IP source routing and IP directed broadcasts are disabled.

IP source routing allows the sender of an IP packet to control the route of the packet to the destination, and IP directed broadcasts allow the network to be used as an unwitting tool in a smurf or fraggle attack.

How

Discuss the router configuration with the network administrator. An example configuration for disabling IP source routing might look something like this for Cisco routers:

```
no ip source-route
```

You should see the following on each interface in the configuration file for Cisco routers to disable IP directed broadcasts:

```
no ip directed-broadcast
```

Additional Firewall Controls

The following are additional test steps for firewalls. Note that some of these controls might be handled by a router in conjunction with a firewall, but a router by itself is a poor firewall for the perimeter of a corporate network.

1. Verify that all packets are denied by default.

All packets on a firewall should be denied except for packets coming from and headed to addresses and ports that are all explicitly defined. This is a much stronger defensive position than trying to keep track of what rules you have set up to block each specific address or service. For example, external SNMP queries from outside your network targeted to a router inside your network would be denied by default if the only traffic you allowed into your DMZ was to a web server.

How

Verify with the firewall administrator that all packets are denied by default. Ask the administrator to show you in the configuration how this is set up.

2. Ensure that inappropriate internal and external IP addresses are filtered.

Traffic coming from the internal address space should not have external addresses as the source address. Likewise, traffic coming from outside the network should not have your internal network as the source address.

How

Verify with the help of the firewall administrator that all packets entering from the exterior with source IP addresses set up for internal networks are denied. Likewise, all packets coming from the interior with source IP addresses not set up for the interior should be denied. Additionally, firewalls should hide internal Domain Naming Service (DNS) information from external networks.

3. Evaluate firewall rule sets to provide appropriate protection.

Failure to manage your firewall rules may expose you to unnecessary risk from open or inappropriate access. It wasn't long ago that a few hundred firewall rules would be considered inexcusable and difficult to manage. Today, many organizations have several hundred, or even thousands, of firewall rules on a single appliance. Firewall rules quickly accumulate and are difficult to remove because administrators are afraid to break applications, forget why specific rules exist, or simply can't navigate the complexity of hundreds of rules. Don't underestimate the importance of this step.

How

Interview the administrator and discuss what tools and processes exist to manage the configuration management process and the change management process. Verify that appropriate controls are in place to identify the purpose of the existing firewall rules. Change controls have already been covered by this point in the audit. However, this is a great opportunity to review the importance of change controls with the administrator.

At some point in the growing complexity of large data sets, such as hundreds of firewall rules across dozens of firewalls, specialized technologies and automated processes must be considered to support firewall management. Several excellent products are on the market, shown in Table 6-3, that can help administrators avoid mistakes and manage firewall rules in large environments. Some auditors may want to run vulnerability scans on the firewall and try various methods of using Nmap to reach assets that should be blocked. These approaches are a fine supplement; however, remember that although the current state of the firewall may secure your assets, inadequate or broken firewall management processes could leave your organization with an ongoing risk. Both are important. You need to be assured that your technical controls are effective now and that they will continue to be effective because of the additional controls in place to manage system changes.

Product	Company	Website
FireMon	FireMon	www.firemon.com
SecureTrack	Tufin	www.tufin.com
Firewall Analyzer	Algosec	www.algosec.com
Firewall Assurance	Skybox Security	www.skyboxsecurity.com
Playbook	Matasano Security	www.matasano.com/playbook

Table 6-3 Firewall Management Solutions

4. Evaluate the use of intrusion detection or other packet security monitoring technologies.

More advanced firewalls (so-called NGFWs) may include intrusion detection or other packet inspection technologies. If present, these should be maintained in order to provide ongoing protection.

How

Discuss intrusion detection system (IDS) rules with the network administrator. For many environments, this capability is built into a firewall. For others, a separate appliance or system may serve as a network IDS.

You should expect to find that IDS rules, often signature-based, are maintained and updated periodically. Many IDS providers offer regular updates to IDS signatures. Custom IDS rules should be documented per the organization's change process. If the IDS is not updated, emerging threats may not be detected by the system.

5. Evaluate the use of layer 7 (application layer) protections.

Advanced firewalls can inspect traffic payloads for much more than source, destination, and port. Application layer firewalls can defend web servers and other applications from threats embedded in traffic not inspected by traditional firewalls.

How

Interview the network administrator to determine the presence and scope of application firewalls. These may include web application firewalls (WAFs) or other packet inspection systems. Many organizations deploy this kind of technology at network perimeters or in front of web servers. They can be used to filter traffic to specific types of traffic, analyze documents, and more.

As with an IDS, you should expect to find a process for updating rules within an application firewall that follows a documented change or update process.

6. Determine how firewall data is reviewed or monitored.

When a firewall does its job, unwanted traffic is kept out, and large networks see high numbers of connections dropped by firewalls. However, when traffic is blocked by a firewall, this can be an indication that a threat is probing or inspecting your network, looking for weak points. Firewall data should be reviewed regularly to identify potential risks.

How

Depending on the type of firewall involved, you may need to discuss the situation with multiple teams. Some organizations may separate management of traditional firewalls from IDS and application firewalls. In all cases, you will need to identify the administrator to determine how firewall output is logged and reviewed.

In high-traffic organizations, it may not be practical to log all firewall data. However, a best practice is to log not only blocks and other rule-based alerts but also information on "accepted" traffic. For example, if a particular source IP address tries three destinations and is blocked but is able to pass through the firewall on trying a fourth, a log showing only blocked packets will not alert security teams that an unauthorized access attempt may have been successful.

For IDS and application firewall systems, you should expect that the system is monitored for activity triggering the rules. Determine if alerts are sent to a central monitoring team or reviewed by network administrators. If no one is monitoring the detection system, malicious activity could persist in the environment.

Additional Controls for Wireless Network Gear

The following are additional steps for wireless environments. You may choose to complete these steps as a separate audit, depending on how your network team has organized its responsibilities.

1. Ensure that access points are running the latest approved software.

Running old firmware on an access point (AP) may leave it open to known attacks or prevent the organization from taking advantage of more robust security features.

How

Evaluate a representative sample of APs with an administrator and verify that the code running on the AP is the latest version. Verify that the latest version is correct using the manufacturer's website or some other similar updated source of information from the manufacturer. Examine the change management processes used to evaluate and maintain current code releases for the APs. Note whether this process is automated and coordinated and whether it scales operationally across regional offices. Ensure that a backout plan exists for firmware updates.

2. Evaluate the controls around centralized WLAN management.

Verify that management software is used to the fullest extent possible and that it's kept under tight control with sound policies and procedures. Centrally managed WLAN tool suites are a powerful way to control the many APs likely under the network team's supervision, especially across different geographic locations. Often, management software is available from the same company that manufactured the APs. The access controls surrounding the tool suite must also be managed to prevent someone from purposefully or inadvertently wreaking havoc on your network and user population by changing AP settings.

How

Discuss the capabilities of the management software with your administrator and ask for a demonstration of the management suite and its capabilities. Ask for procedures discussing access to the management suite, including who has access and how that access is controlled. If passwords are used, ensure that passwords meet company policy and are rotated according to appropriate policies. Ensure that access to wireless administration systems is removed promptly upon resignation, termination, or role change.

3. Evaluate the security of the wireless authentication and encryption method.

Insecure authentication and encryption methods place the integrity of your network at risk. The tools and methods used to compromise some authentication systems are readily available and easy to use.

How

Wireless networks can be configured in many ways. Guest networks in lobbies or conference areas may not require any authentication when a user connects. Corporate internal wireless networks may require the device or user to provide proper credentials before allowing a connection. During your audit of the wireless

environment, consider the various networks under discussion, as answers to your inquiries may vary greatly depending on the scope of the audit. You will need to evaluate several key aspects of authentication and encryption through discussions with your wireless team, including:

- Does the network permit direct access to internal company information (i.e., is it an internal wireless network?)
- How do devices or employees authenticate to the wireless network? Is access password-based? Certificate-based? How are certificates distributed and revoked? Are they managed at a user level or at a device level? Is a wireless NAC system in use?
- What type of encryption is used for authentication and transmission? Private (internal) networks should use the Wi-Fi Protected Access version 2 (WPA2) standard for connections. Due to security vulnerabilities, Wired-Equivalent Privacy (WEP) or the original WPA should not be allowed to remain in use under any circumstances.

NOTE WPA3 was announced in 2018 and is beginning to find its way into devices, but it will take time for this to become an enterprise standard. WPA3 provides additional convenience and security features, including the ability to encrypt transmissions on "open" wireless networks.

A complete explanation of wireless authentication and encryption is beyond the scope of this book, but a useful primer on wireless security can be found in NIST Special Publication 800-97, *Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i.*

4. Verify that rogue access points are not present on the network.

This is the step that most people think about when someone brings up a wireless audit. A rogue AP is any unauthorized AP facilitating a connection to a private network. Rogue APs may allow unauthorized systems to join internal networks and access sensitive company data.

How

Review the organization's processes to identify and remove rogue APs with the network administrator. The network team should have a process for detecting unexpected APs and alerting the appropriate personnel to act. Determine how alerts are handled, and ensure that a documented, repeatable process is in place.

You can attempt to verify the absence of rogue APs in a number of ways, including specialized wireless appliances or software and searching through your network traffic. The use of commercial tools built into wireless infrastructure is the most effective means.

Commercial WLAN Security Tools Many modern wireless infrastructures include capabilities to detect unexpected wireless devices, particularly those attempting to use the same SSID as an "official" AP. If commercial WLAN security tools are available in your environment, you should work with the administrator to determine what settings are in use, who is notified if a rogue AP is detected, and what response processes are in place.

NOTE Scanning for wireless networks and access points, usually called wardriving, is less useful in enterprise environments today due to the rise of personal hotspots included with many carrier data plans. As long as these devices aren't also attached to your internal network, personal hotspots are primarily a concern relative to their impact on acceptable use policies.

Searching Through Network Traffic You theoretically could search through your network for MAC addresses belonging to wireless APs and attempt to identify unknown devices in this way. Each network card has a MAC address uniquely assigned to it by the manufacturer. Each MAC address contains as part of the address an identifier unique to each manufacturer of these devices. This organizationally unique identifier, or OUI, is a 24-bit globally unique assigned number. The OUI usually is concatenated with another 24 bits that are assigned by the company to make a 48-bit number that is unique to a particular piece of hardware. The 48-bit number is the MAC address. Each network card has a MAC address assigned to the card used to route packets from the network card to the next hop on the network. The idea is to address a piece of hardware uniquely. The problem with this approach is that as networks have grown, so, too, have the number of devices and device manufacturers, and as a result it is now very difficult to know what is supposed to be on a network based merely on the MAC address.

5. Evaluate procedures in place for tracking end-user trouble tickets.

Failure to establish ownership and tracking of end-user issues could result in users being unable to resolve connectivity problems.

How

End-user issues should be tracked through a trouble ticketing system. An owner for these issues should be assigned and a group should be made responsible for tracking the progress toward closure of any tickets opened because of WLAN issues. Discuss these processes with the administrator.

6. Ensure that appropriate security policies are in place for your WLAN.

Policies help to ensure compliance with a standard, help with repeatable processes, and allow the company to act against documented company violations. Since employees may not immediately understand the risks associated with the various wireless devices they may carry, your organization may find it useful to have a policy specific to wireless networking or wireless device use.

How

Determine whether WLAN policies exist and whether the administrator responsible for the WLAN knows and understands the content of the policies. Determine whether the policies are being followed or what barriers might exist that prevent them from being followed. Finally, ensure that relevant portions of the WLAN

policies are communicated to employees that use the wireless network. A few common policy items might include the following:

- All wireless transmissions must be encrypted to prevent eavesdropping.
- All APs must have updated firmware.
- Only authorized people on the wireless networking team may have direct administrative controls of the APs.
- Only authorized people on the wireless networking team may install APs.
- Passwords to APs must adhere to company policy.
- All efforts will be made to reduce propagation of radio waves outside the facility.
- Devices accessing the network must use endpoint firewalls and antivirus programs.
- The wireless team must monitor for rogue APs on a periodic basis.
- Only authorized systems owned by the company may access the network and only for appropriate business use.

Tools and Technology

These tools can be quite helpful in understanding configurations and traffic patterns on your network. Many general vulnerability scanners can also identify network configuration problems or security risks.

Tool	Website
Wireshark	https://www.wireshark.org
Nmap	https://nmap.org
"Top 125 Network Security Tools"	https://sectools.org

NOTE Automated tools can be quite harmful to production environments. Exercise care, and design any testing in a manner that will not affect production systems.

Knowledge Base

Resource	Website
Aruba (HP)	www.arubanetworks.com
Cisco	www.cisco.com
Fortinet	www.fortinet.com
Palo Alto Networks	www.paloaltonetworks.com
Institute of Electrical and Electronics Engineers (IEEE)	www.ieee.org
Internet Engineering Task Force (IETF)	www.ietf.org
National Vulnerability Database	http://nvd.nist.gov
Assigned port numbers, essential for reading access lists	https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml
OSI model	https://en.wikipedia.org/wiki/OSI_model
NIST Special Publication 800-97	https://csrc.nist.gov/publications/detail/sp/800-97/final

Master Checklists

The following tables summarize the steps listed herein for auditing routers, switches, and firewalls.

General Network Equipment Audit Steps

These controls should be evaluated in addition to performing the specific steps in the following checklists as they apply. For example, if you were to audit a switch, router, firewall, or wireless system, you would perform the steps in the following checklist and then additionally perform the steps under the specific checklist for that type of device.

Checklist for Auditing Network Equipment	
❑	1. Review controls around developing and maintaining configurations.
❑	2. Ensure that appropriate controls are in place for any vulnerabilities associated with the current software version. These controls might include software updates, configuration changes, or other compensating controls.
❑	3. Verify that all unnecessary services are disabled.
❑	4. Ensure that good SNMP management practices are followed.
❑	5. Review and evaluate procedures for creating user accounts and ensuring that accounts are created only when there is a legitimate business need. Also review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.
❑	6. Ensure that appropriate password controls are used.
❑	7. Verify that secure management protocols are used where possible.
❑	8. Ensure that current backups exist for configuration files.
❑	9. Review logging and monitoring processes for network equipment.
❑	10. Evaluate use of the Network Time Protocol (NTP).
❑	11. Verify that a banner is configured to make all connecting users aware of the company's policy for use and monitoring.
❑	12. Ensure that access controls are applied to the console port.
❑	13. Ensure that all network equipment is stored in a secure location.
❑	14. Ensure that a standard naming convention is used for all devices.
❑	15. Verify that standard, documented processes exist for configuring network devices.
❑	16. Evaluate the use of network access control (NAC) technology to ensure the network is accessed in accordance with company policy.
❑	17. Review the availability and access controls for web-based (GUI) access to network components.
❑	18. Ensure that web sessions are configured with appropriate idle and session timeouts.
❑	19. Review disaster recovery plans related to network devices.

Auditing Layer 2 Devices: Additional Controls for Switches

These controls should be evaluated in addition to performing the general steps for auditing network equipment.

Checklist for Auditing Layer 2 Devices: Additional Controls for Switches
❑ 1. Verify that administrators avoid using VLAN 1.
❑ 2. Evaluate the use of trunk autonegotiation.
❑ 3. Verify that Spanning Tree Protocol attack mitigation is enabled (BPDU Guard, STP Root Guard).
❑ 4. Evaluate the use of VLANs on the network.
❑ 5. Disable all unused ports and put them in an unused VLAN.
❑ 6. Evaluate use of the VLAN Trunking Protocol (VTP) in the environment.
❑ 7. Verify that thresholds exist that limit broadcast/multicast traffic on ports.

Auditing Layer 3 Devices: Additional Controls for Routers

These controls should be evaluated in addition to performing the general steps for auditing network equipment.

Checklist for Auditing Layer 3 Devices: Additional Controls for Routers
❑ 1. Verify that inactive interfaces on the router are disabled.
❑ 2. Ensure that the router is configured to save all core dumps.
❑ 3. Verify that all routing updates are authenticated.
❑ 4. Verify that IP source routing and IP directed broadcasts are disabled.

Auditing Firewalls: Additional Controls

These controls should be evaluated in addition to performing the general steps for auditing network equipment.

Checklist for Auditing Firewalls: Additional Controls
❑ 1. Verify that all packets are denied by default.
❑ 2. Ensure that inappropriate internal and external IP addresses are filtered.
❑ 3. Evaluate firewall rule sets to provide appropriate protection.
❑ 4. Evaluate the use of intrusion detection or other packet security monitoring technologies.
❑ 5. Evaluate the use of layer 7 (application layer) protections.
❑ 6. Determine how firewall data is reviewed or monitored.

Auditing Wireless Network Gear: Additional Controls

These controls should be evaluated for wireless gear in addition to the general steps for auditing network equipment.

Checklist for Auditing Wireless Network Gear: Additional Controls
❑ 1. Ensure that access points are running the latest approved software.
❑ 2. Evaluate the controls around centralized WLAN management.
❑ 3. Evaluate the security of the wireless authentication and encryption method.
❑ 4. Verify that rogue access points are not present on the network.
❑ 5. Evaluate procedures in place for tracking end-user trouble tickets.
❑ 6. Ensure that appropriate security policies are in place for your WLAN.

CHAPTER 7

Auditing Windows Servers

Since its introduction in 1985, Microsoft Windows has evolved into one of the world's most pervasive operating systems for both servers and clients. This chapter covers the basic components of a Windows server audit. Many of the tools and resources discussed here are also applicable to Windows clients and could be used in conjunction with Chapter 14 on end-user computing devices.

We will discuss the following:

- A brief history of Windows development
- Windows essentials: learning about the target host
- How to audit Windows servers
- Tools and resources for enhancing your Windows audits

Background

Microsoft and IBM worked jointly starting in the mid-1980s to develop the OS/2 platform as a planned successor to both DOS and Windows. But the growing success of the Windows platform led to disagreements over OS/2's direction, and the relationship turned sour. Microsoft and IBM split up, and Microsoft was free to pursue their vision in what became Windows NT 3.1, first released in July 1993. Windows NT was the first "professional" version of the Windows operating system, targeting businesses and government organizations. More than 25 years later, Windows NT is still going strong—both Windows 10 and Windows Server 2019 are built on the latest version of the NT core, or "kernel."

The server market has evolved over many years through various releases, including Windows NT, Windows 2000, Windows Server 2003/2008/2012/2016, and, presently, Windows Server 2019. In 2017, Microsoft began offering an alternative delivery model for Windows Server that includes twice-annual updates; these are designated simply as "Windows Server," with a year-based version or build number, such as 1709 or 1803. What this all means for the auditor is that many versions of the operating system may be present in large environments. You should spend time familiarizing yourself with the operating systems in your environment. Be prepared to be flexible; some utilities mentioned here may not work on all versions of Windows. In some situations, hosts might exist on your network that are no longer supported by Microsoft. Additional controls should be in place to protect these systems, such as technologies that prevent network attacks or malware propagation.

Microsoft's Windows platform serves as the foundation for dozens of components that may be present in many businesses, including directory and au-

thentication services, web infrastructure, databases, system management, e-mail, mobile device management, and more. System administrators may throw around acronyms or service names like Hyper-V, AD, Exchange, IIS, SQL Server, Intune, SCCM—these all run on Windows Server. It's important as an auditor to know what type of environment you are inspecting and which of your organization's security policies apply to that environment.

You should also take time to understand your organization's use, if any, of cloud services such as Amazon Web Services or Microsoft Azure, or offsite hosted environments such as Rackspace. We'll get to cloud computing in detail in Chapter 16, but depending on your organization's network architecture, you may be required to audit Windows Server systems outside the boundaries of your local data center. The principles discussed in this chapter apply to almost all Windows Server environments, whether installed on bare metal, running as virtual machines, sitting in your local data center, or running in a high-availability cloud environment.

Windows Auditing Essentials

The material in this chapter requires a basic understanding of the components that comprise the Windows environment. In addition, your role as an auditor and advisor will significantly improve if you understand how to approach a comprehensive audit of a Windows platform.

Figure 7-1 illustrates how the operating system serves as a vehicle for supporting applications. Many components surrounding the operating system should be considered in a complete review. For example, consider the danger of poorly maintained or configured applications. The more applications you add to the platform, the more potential trouble areas you have as an auditor as you increase your attack surface area. Several chapters in this book are devoted to applications that you might want to consider for your audit. In addition, the hardware, storage, and network affect the performance and protection of the operating system. Finally, the surrounding controls and management of the environment affect the support, risk, compliance, and business alignment of the server.

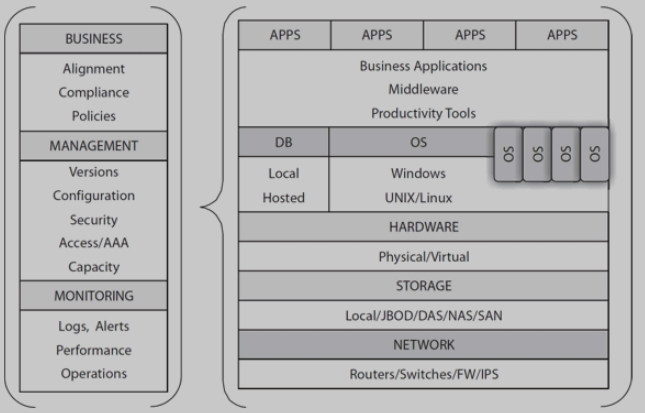


Figure 7-1 Model for auditing hosts

Consider scheduling time on your calendar to learn more about the tools discussed in this chapter. You might be surprised at how easy most of them are to use and how much more efficient you become because you know the shortcuts to getting the information you need. Sometimes it's too easy as auditors to continue using what you've always used because it works, instead of looking at new methods for improving your efficiency. After you've done a little homework, you can ask your company administrators to show you the ropes. Most administrators of any caliber actually enjoy doing this. You can be assured that if you show up to an administrator's office asking about an obscure tool, you'll get his or her attention, and one of you will walk away a little wiser for the visit.

Command-Line Tips

The Windows command-line environment is suitable for completing most tasks discussed here. More adventurous auditors may want to leverage PowerShell for more complex tasks or automation. PowerShell is installed by default on modern Windows platforms.

Windows Server may be installed for some uses as Server Core, which includes no GUI support and minimal services. If you are auditing a Server Core system, you will need to use only the command-line tools listed here, or use a remote administration tool. Standard server installations have a full Windows GUI that includes useful tools like Server Manager.

Those of you who are comfortable with the command line on a Unix machine may appreciate installing Unix functionality using Cygwin from www.cygwin.com, which allows you to access several utilities such as `ls`, `sed`, `grep`, `more`, and `cat`. It's also possible to create scripts based on these binaries, located in the bin directory, to manipulate the text output from standard Windows utilities. Finally, as long as you understand the risks involved, you power users may even want to add the <drive>:\cygwin\bin directory to the environment path.

Beginning with Windows Server 2019, the Windows Subsystem for Linux (WSL), previously available only for Windows 10, is supported on server platforms. WSL allows a full Linux installation to be installed directly as part of Windows rather than in a virtual machine. Depending on your organization's policies, this may be an option to explore if you prefer using a Unix command line.

NOTE If you like the command line and enjoy scripting, take advantage of the resources located in Microsoft's scripting center website at https://gallery.technet.microsoft.com/scriptcenter.

Essential Command-Line Tools

Several tools should be in every administrator's back pocket. Keep in mind that with today's complex firewalls and malware protection, not all of these tools may work properly. The commands referenced in this chapter were verified with Windows Server 2016, but you should test every tool in a lab environment for your specific operating system installation prior to running it on a production network. In addition, some environments may restrict the use of certain utilities. Obtain necessary approvals and notify your administrator before installing any software on a server.

NOTE The various tools discussed in this chapter can be powerful. Follow best practices. Learn how these tools work on another computer off the network in a test environment prior to using them on your own computer or production network and systems.

Resource Kit Tools

Earlier versions of Windows either shipped with or were compatible with add-on Resource Kits from Microsoft. These kits contained many additional tools for administering and troubleshooting systems, configuring security features, and much more. Beginning with Windows Server 2008, these tools have been replaced with more robust or more powerful tools. Many useful commands are now built into Windows and are accessible with PowerShell. Others can be found using the Remote Server Administration Tool or by using the Sysinternals suite of commands.

NOTE Microsoft offers outstanding command-line help at https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands. Type **help cmd** from the command prompt for general information about using the command line in Windows.

Sysinternals Tools

The Sysinternals tools have been used by administrators for over two decades. The package was so popular that Microsoft bought the suite in 2006. Sysinternals helps administrators and auditors perform complex tasks and detailed analysis. You can download Sysinternals tools from Microsoft via https://docs.microsoft.com/en-us/sysinternals/, or you can try the Sysinternals Live service to execute compatible Sysinternals tools on systems with Internet access. Dozens of GUI and command-line tools are available for remote administration, network analysis, process and registry monitoring, and other tasks. Several companies include a subset of these tools as part of the standard build for servers and clients. Many Sysinternals functions can be duplicated or scripted in PowerShell, but many administrators still prefer the simplicity of Sysinternals.

Other Tools

Many other tools are available as well, some of which are listed here and discussed in the various audit steps. You can script nearly everything in the following audit, and in some cases, you may find that your organization already has a commercial configuration management tool that can perform a detailed analysis of Windows systems. You will still find it helpful to sample critical servers and individually test them for appropriate controls. One interesting tool, the Windows Forensic Toolchest (WFT), written by Monty McDougal, serves as a wrapper for command-line tools. It can handle any of the tools listed here or others you may want to add. WFT is referenced as part of the SANS forensic track. You can learn more about it from www.foolmoon.net/security.

Common Commands

Table 7-1 presents a list of command-line tools used throughout this chapter.

Tool	Description	Where to Get It
PsInfo	List system information, including installed service packs, patches, applications, and drive information	www.sysinternals.com
systeminfo	List system information	Native command
pslist	List running processes	www.sysinternals.com
PsService	List all installed services	www.sysinternals.com
cmdkey	Create, list, or delete stored credentials	Native command
netsh	Display or modify network configuration	Native command
netstat	Provide network information	Native command
Sc	Tool for talking with service controller	Native command
tcpview	GUI view of processes mapped to ports	www.sysinternals.com
procexp	Powerful GUI process explorer	www.sysinternals.com
schtasks	List scheduled tasks at the command line	Native command
bootcfg	List boot partition information	Native command
pendmoves	List file move operations scheduled for the next reboot	www.sysinternals.com
autoruns	List everything scheduled to start when your computer starts up—the GUI version	www.sysinternals.com
autorunsc	List everything scheduled to start when your computer starts up—the command-line version	www.sysinternals.com
rsop.msc	Open the resulting set of security policies on your host when run from the Start | Run path or command line	Native command
secpol.msc	Open just the local computer policy	Native command

Table 7-1 Common Commands Used in This Chapter

Server Administration Tools

Newer Windows Server versions include a GUI dashboard called Server Manager, available from the Start menu. Server Manager displays a great deal of basic configuration information and has a Tools menu with shortcuts to commonly used features, like System Configuration, Services, and Computer Management.

Microsoft has also released a tool called Windows Admin Center, which complements Microsoft System Center Configuration Manager (SCCM) by allowing an administrator to dive into a specific server and assess system health, see installed updates, review running processes, and more. Many of the functions listed in this chapter are available in Windows Admin Center. Learn more at https://docs.microsoft.com/en-us/windows-server/manage/windows-admin-center/overview.

Remote Server Administration Tools (RSAT) enable a Windows 10 client to manage roles and features running on Windows Server systems. RSAT allows administrators to perform remote server management functions and includes several great tools that are otherwise difficult to duplicate in functionality.

NOTE You can easily add the Microsoft Windows RSAT to your desktop or laptop computer. Search for RSAT at www.microsoft.com. After downloading the installer package onto your computer, you need to run the file as an administrator to install the tools onto your system.

Performing the Audit

The key to a successful audit of Windows servers is to review the host thoroughly by itself and in conjunction with the many other possible connections that pass data to and from the host.

The following audit steps focus only on the host and do not cover extensive reviews of overlying applications or trust relationships with outside systems. Also not covered are data input and data output methods or their validity. You would deal with these on a per-host basis using techniques and tools covered elsewhere in this book. The steps shown here are typical of many server audits and represent a good trade-off between the number of risks covered and the amount of time it takes to review the host.

Test Steps for Auditing Windows

In an ideal world, you would audit against a reference set of controls and information covering every possible configuration setting. However, we don't live in an ideal world, and most of us don't have that much time per host. The test steps in this chapter are a recommended list of items to evaluate. From experience, we know that debate abounds regarding auditing Windows. Can a Windows server be secured? What makes your steps better than someone else's steps? The steps covered here have worked for several companies.

Many auditing programs fail to balance effective audits and effective time management. Related to time management, notice that we spend a lot of time discussing various ways to script the results. Configuration management tools can also be leveraged by the audit team to review scores of servers very quickly, and some audit packages promise the same. The only concerns here regard ensuring that all of the controls that affect the business are covered and occasionally validating the results of the tools with your own independent reviews.

Initial Steps

The following represents a check of the overall system setup and other basic controls to ensure overall system compliance with your organization's policy. These are mostly general, high-level controls, such as making certain that the system runs company-provisioned firewall and antivirus programs.

1. Obtain the system information and service pack version and compare with policy requirements.

Policies were written and approved to make your environment more secure, easily manageable, and auditable. Double-check the basic configuration information to ensure that the host is in compliance with policy. Older operating systems increase the difficulty in managing the server and increase the scope of administrator responsibilities as he or she attempts to maintain control over disparate operating system (OS) versions. Maintaining standard builds and patch levels greatly simplifies the process of managing the servers.

How

You could find this information using built-in command-line tools, hunting through the graphical user interface (GUI), and searching the registry. However, two efficient ways to pull up this information include the Sysinternals tool PsInfo and the native tool systeminfo. Use one of these tools to retrieve this information, and then compare the results with your organization's policies and requirements.

NOTE Download `PsTools` or the entire Sysinternals suite from www.sysinternals.com. The tool `PsInfo` is part of this set of tools. You may want to use several tools from Sysinternals for auditing your servers.

2. Determine whether the server is running the company-provisioned firewall.

Failure to use a firewall subjects the client to network attacks from malware, attackers, and curious people.

How

Most of the time, a check of the processes on the system shows that the company-provisioned firewall is installed and running on the system. An easy way to script this check is to run the Sysinternals tool `pslist`. Do this by running `pslist <process name>` on the system, and search for the appropriate running process by specifying the process you want to find.

For many organizations, the firewall is centrally managed and the same across all hosts in a group. You may want to verify the configuration of the firewall on the host.

If you are using the Windows Firewall, learn the `netsh` command set, which allows scripted output and changes to the firewall. Try running `netsh advfirewall show currentprofile` to see the configuration of the firewall on the host and whether the firewall is configured for particular adapters. Use `netsh advfirewall` to see other available options for the `netsh advfirewall` tool.

3. Determine whether the server is running a company-provisioned antivirus program.

Failure to have antivirus protection may allow harmful code to run on the system. Antivirus tools can also identify the presence or actions of hacking tools run by malicious actors. In Windows Server 2016, Windows Defender AV is installed and enabled by default, but some organizations will use an alternative antivirus program.

How

In GUI environments, the Server Manager screen for the Local Server lists the status of Windows Defender. For other AV packages, a visual check of the system tray usually shows that an antivirus program is installed and running on the system. As mentioned earlier, an easy way to script this check is to run `pslist` from Sysinternals on the system and search for the running process, MsMpEng in the example that follows:

```
pslist MsMpEng
PsList 1.4 - Process information lister
Copyright (C) 2000-2016 Mark Russinovich
```

```
Sysinternals - www.sysinternals.com
Process information for MK-PROD:
Name Pid Pri Thd Hnd Priv CPU Time Elapsed Time
MsMpEng 10012 8 30 981 416528 0:53:58.843 437:26:12.997
```

Depending on the nature of your audit, you also might want to check the configuration of the antivirus program on the host. For many organizations, the antivirus program is managed centrally and is the same across all hosts. Some organizations may use specialized security software or configurations on high-load, legacy, or other environments. Ideally, the configuration should not exclude any files or folders from scanning and should be set to protect the system in real time for all file operations. Antivirus tools should also be configured to automatically download and install signature updates. Deviations can put the system at greater risk.

4. Determine whether the server is running a company-provisioned patch management solution.

Failure to have a company-provisioned patch management solution may prevent the server from receiving the latest patches, allowing harmful code or hacking tools to run on the computer.

How

A visual check of the processes in the Task Manager usually shows that the company-provisioned patch management system for servers is installed and running on the system. For example, this may be evidenced by the existence of the process (e.g., CcmExec.exe) in the Task Manager or the output of `pslist`. You can also verify whether the system shows up on the SCCM console and validate the last patch cycle applied to a given machine.

Some organizations may enable automatic Windows updates for their systems. This will only cover patches and updates issued by Microsoft; any third-party applications on the system must be patched another way. To check the status of Windows Update, use the command `sconfig`, and review the attribute for Windows Update Settings.

5. Ensure that all approved patches are installed per your server management policy.

If all the OS and software patches are not installed, widely known security vulnerabilities could exist on the server.

How

Use `systeminfo or psinfo -s` to pull this information up for you, and then compare the results against the policies and requirements of your organization. You can use the output to compare with existing SCCM or other patch management data. You could also compare the output with data from a vulnerability scanner to identify possible disparities.

Account Management

Account management and related controls are fundamental components of server management. Tracking users over time is a difficult task and a common method for gaining access to systems that a user should never have had access to in the first place.

6. Review and evaluate procedures for creating user accounts and ensure that accounts are created only for a legitimate business need. Review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.

If effective controls for providing and removing access to the server are not in place, it could result in unnecessary access to system resources. This, in turn, can place the confidentiality, integrity, and availability of the server at risk.

How

Interview the system administrator, and review account creation procedures. This process should include some form of verification that the user has a legitimate need for access.

There are a few different ways to view account attributes on a system, but it may take some effort to paint a complete picture. You can start by viewing a listing of local accounts on a target system by opening the Computer Management GUI (`compmgmt.msc`) and selecting the Local Users and Groups tree. You can also try the `net user` command, which will list the local users of the system. For either method, review any evidence that the accounts were approved properly prior to being created.

Most system access in larger organizations is not managed via local accounts, but with Active Directory (AD) domain accounts, covered in the next section. To get a complete picture of accounts that can access a system, you'll need to understand how groups assigned to the system are constructed and which types of AD accounts are members of which group.

You should also review the process for removing accounts when access is no longer needed. This process could include an automated feed from the company's human resources (HR) system providing information on terminations and job changes, but this can be challenging with local accounts. The process could include a periodic review and validation of active accounts by the system administrator and/or other knowledgeable managers. Obtain a sample of accounts and verify that they are owned by active employees and that those employees' job roles have not changed since the account's creation.

Additional controls may be appropriate in your environment to monitor the use of sensitive administrator accounts. Review these controls if they are determined to be a critical part of your audit.

7. Ensure that all users are created at the domain level and clearly annotated in Active Directory. Each user should trace to a specific employee or team.

Most user accounts—including system administrator accounts—should be administered centrally by a domain controller, with the possible exception of accounts created on isolated systems that are not a member of a domain. Administering accounts centrally simplifies account management and control, since accounts can be created or deleted from a single location and account policies can be applied globally. Even shared accounts that might be found in a lab environment, kiosk, or

conference room should be owned by a specific employee.

How

Interview the system administrator or domain administrator to understand the process for creating user accounts and assigning them to servers. Discuss your findings with the administrator, and pay close attention to accounts that are not managed by the domain. In more secure environments, most local accounts will be disabled or deleted, and the default administrator account will be renamed to a less obvious name.

This is also a good time to investigate the use of shared accounts. Such accounts present risk in that you lose accountability for actions taken on the system, but may be unavoidable in certain situations. Organizations dealing with personally identifiable information (PII), Payment Card Industry (PCI), or Health Insurance Portability and Accountability Act (HIPAA) should closely examine their use of shared accounts.

8. Review and evaluate the use of groups, and determine the restrictiveness of their use.

Groups can greatly simplify the provisioning and deprovisioning process for adding or removing user access to systems as users join and leave a team. However, old members sometimes hang around inside a group when they leave a team.

How

Review the contents of the groups on the system for appropriate membership. Access Computer Management, then select the System Tools | Local Users and Groups | Groups tree. Each group in the panel can be expanded to reveal group membership. Remember that in an Active Directory environment, groups can be nested, and you may need to check the membership of the nested groups with an AD administrator. Pay particular attention to the Administrators, Guests, Users, Power Users, and Remote Desktop Users groups.

Additionally, ensure that the information security team, investigations team, and appropriate support personnel have proper access to the server. This may not pertain to all organizations, and there may be some exceptions. These users should be placed into a group and not added as individual users to the server. Organizations with higher sensitivity to AD security concerns may choose to leverage external solutions for temporary administrative access or use the new just-in-time (JIT) administration features in Windows. If these apply to your organization, talk to your administrators to understand how administrative privileges are assigned and removed.

NOTE A policy is rarely perfect; there are always exceptions. This is completely acceptable, provided that the organization has a process to document exception requests and require approval by the appropriate level of management. Many large organizations require the highest levels of management to sign off on such requests to discourage exceptions to policy.

9. Review and evaluate the strength of passwords and the use of password controls on the server, such as password aging, length, complexity, history, and lockout policies.

All accounts should have passwords. The methods used to test these controls depend on the password-provisioning process and controls enabled on the servers and Active Directory. At a minimum, you should review system settings that provide password controls. Password controls are essential to enforcing password complexity, length, age, and other factors that keep unauthorized users out of a system. Many organizations choose to assign more stringent password settings to privileged accounts, such as those with administrator rights on the server.

How

You can find the account policies as they affect your system by using the Server Manager interface in the Start menu. When the window opens, look for the Tools drop-down, and choose Local Security Policy. When the panel opens, choose the Account Policies tree, and examine the listings for Password Policy and Account Lockout Policy. You can also see much of the same information at the command line using `net accounts`. In general, verify that the policies listed in Table 7-2 are set in accordance with your local policies. Some common settings are listed.

Policy	Setting
Minimum password age	1 day
Maximum password age	30–90 days
Minimum password length	8–14 characters
Password complexity	Enabled
Password history	10–20 passwords remembered
Store passwords using reversible encryption	Disabled, if possible, but understand and test this before making this decision
Account lockout duration	10–30 minutes
Account lockout threshold	10–20 attempts
Reset account lockout after	10–30 minutes

Table 7-2 Account Policies

Previous editions of this book described processes to obtain and crack Windows account hashes to directly test password strength. While a thorough audit may still involve cracking passwords, we suggest at this point that you leave any Windows password-cracking exercises to professional penetration testers or consult your security team for help. Numerous resources are available on the Web that describe this process if you choose to venture down this path.

Permissions Management

Microsoft ships with a robust ability to configure user rights and security options. These are only effective, however, if they are configured properly.

10. Review and evaluate the use of user rights and security options assigned to the elements in the security policy settings.

Windows Server offers hundreds of user rights settings and security options. These settings and options allow broad, sweeping, and powerful changes to how the host behaves under many different situations.

CAUTION Be careful here. It is possible to lock yourself out, disable critical internal processes, and limit necessary functionality. It's strongly recommended that you thoroughly test any changes you make here in a test environment with any applications that may even possibly depend on the settings running on the system.

How

You'll find the security policies as they affect your system by accessing the Local Security Policy panel (`secpol.msc`, or via Server Manager's Tools menu). Remember that you can export these settings by right-clicking the folder icon and selecting Export List. Another helpful command-line option is to type `gpresult /v` to get a listing of Group Policy settings.

Evaluate the settings you found with the policies for your organization. Several guides suggest recommended settings, including Microsoft's own Windows Server 2016 Security Guide, the Center for Internet Security hardening guides (www.cisecurity.org), and, of course, SANS (www.sans.org). The bottom line here is that you need to decide what your organization is looking to accomplish and audit against these settings. If your organization isn't using these settings at all, you should take the initiative to spearhead a project to look into them. Here are some common settings for both.

Common security options include the following:

- Renaming guest and administrator accounts
- Disabling the guest account
- Choosing not to display the last logged-on user
- Prompting the user to change the password before expiration
- Refusing enumeration of SAM accounts and shares by anonymous
- Refusing to store network credentials (be careful with this!)

Common user rights assignments include the following:

- Changing who can access the computer across the network
- Defining who can log on locally
- Denying access to the computer from the network
- Denying logon through terminal services
- Defining who can take ownership of files or other objects

11. Look for and evaluate the use of file sharing on the host.

Inappropriate or open shares may needlessly compromise personal or company data. You need to identify all shares, shared directories, and permissions. For example, it's not uncommon to find open shares on a network with personal, group ranking, or payroll information. This type of data never should be kept on an open share.

How

Use the Computer Management panel by typing compmgmt.msc in the search bar. When the panel opens, go to System Tools | Shared Folders to view open shares, sessions, and files. You also can view a list of shares by running the command `net share`.

If you have a large set of shares on a server and want to spot-check it for unexpected content, use the built-in Windows search tools available in Explorer.

For each share you find, determine whether the permissions are appropriate. Open shares can increase risk, particularly if write permission is granted. You should consider disallowing public shares where the Authenticated Users or Domain Users group has full control permissions.

Network Security and Controls

Network access to servers must be controlled.

12. Review and evaluate the use and need for remote access, including RAS connections, FTP, Telnet, SSH, VPN, and other methods.

Not all remote access technologies are created equal. Some legacy services such as File Transfer Protocol (FTP) and Telnet transmit all data in cleartext, including authentication information. In an ideal environment, such legacy connectivity should be eliminated. All access to a system for both standard users and administrators should be through modern, encrypted methods, such as Secure Shell (SSH) or Remote Desktop Protocol (RDP). For external access, virtual private network (VPN) connectivity should be employed with multifactor authentication.

How

View the output of the services and port-mapping tools, and discuss these with the administrator. Ask the administrator about the remote access policies and the different methods of access. Question the need for any cleartext communications that aren't driven by business needs. In some cases, cleartext communications exist and are difficult to remove because of a legacy application, or the traffic just isn't that sensitive. However, where possible, an encrypted protocol should be used instead. For Microsoft hosts, encrypted protocols include RDP, Citrix (ICA protocol), SSH, and Secure Sockets Layer (SSL), among many others.

Most access to a Windows Server will be through Remote Desktop Services or through remote administration platforms such as PowerShell, SCCM, RSAT, or similar.

NOTE The use of secure protocols is particularly important in a DMZ and other high-risk environments. The auditor may determine that it is of less importance on the internal network. However, it is still advisable to use secure protocols even on internal networks to minimize attacks from within.

13. Ensure that a legal warning banner is displayed when connecting to the system.

A legal logon notice is a warning displayed whenever someone attempts to connect to the system. This warning should be displayed prior to actual login and should say something similar to this: "You're not allowed to use this system unless you've been authorized to do so." Verbiage of this sort may be needed to prosecute attackers in court.

How

Log in to your account using each available service that provides access, such as Remote Desktop. Determine whether a warning banner is displayed. Interview the system administrator to determine whether the verbiage for this warning banner has been developed in conjunction with the company's legal department.

14. Determine what services are enabled on the system, and validate their necessity with the system administrator.

Enabling network services creates a new potential vector of attack, increasing the risk of unauthorized entry into the system. Therefore, network services should be enabled only when there is a legitimate business need for them.

NOTE This is one of the most critical steps you will perform. Unnecessary and unsecured network services are the number-one vector of attack on Windows servers.

How

The tools shown in Table 7-3 reveal key pieces of information to help you identify services and how they are used. `Netstat` reveals the active sockets on your computer listening for external communications. `Psservice` and `sc query` list the running services. Windows Admin Center and Server Manager can also be used to view services. Next, you can map the running services to the open ports using `tcpvcon`. Finally, `procexp` is also capable of showing you much of this information but cannot be scripted. It is mentioned here because of its powerful capabilities and because it is free.

Tool	Description	Where to Get It
`netstat`	Provide network information	Native command
`PsService`	List service information	www.sysinternals.com
`Sc`	Native tool for talking with service controller	Native command
`Tcpvcon`	CLI view of processes mapped to ports	www.sysinternals.com
`TCPview`	GUI view of processes mapped to ports	www.sysinternals.com
`procexp`	Powerful GUI process explorer	www.sysinternals.com

Table 7-3 Tools for Viewing Service Information

These may seem like a lot of utilities, but it's worth your time to look through them to decide what information you need for your audit. In general, if the system is being used in an AD domain, ensure that the Group Policy Object (GPO) policy rules are periodically reviewed. These rules are applied to any system that joins the domain/specific branch.

You can use the native `netstat` command by typing `netstat –an` at the command line. Look for lines containing `LISTEN` or `LISTENING`. The host is available for incoming connections on these TCP and UDP ports. You can find a list of services using such tools as `psservice`, which is very much like the `netstat` service on *NIX systems.

Other utilities that map processes to port numbers include the built-in `sc` (try `sc query type= service`) command and tcpvcon from Sysinternals. We recommend tcpvcon from Sysinternals. The "Tools and Technology" section a bit later offers information about where to find these tools and more. You can run `tasklist /svc` if you quickly want to map existing process IDs to running services. If you want to know absolutely everything about a process, download and run the Sysinternals Process Explorer.

Once you have obtained a list of enabled services, discuss the results with the system administrator to understand the need for each service. Many services are enabled by default and therefore were not enabled consciously by the system administrator. For any services that are not needed, encourage the administrators to disable them. The Microsoft snap-in for the management console can be launched by typing `services.msc` from the Search option on the taskbar.

15. Evaluate vulnerability scanning procedures and ensure that known vulnerabilities are resolved.

New security vulnerabilities are discovered in Windows regularly, and information about them is distributed to the Windows community (which also includes potential attackers). If the system administrator is not aware of these issues and does not install security patches or deploy other mitigations, well-known security vulnerabilities could exist, providing a vector for compromise of the system.

How

Request access to the scan repository or scanning tool and review the results of recent scans directly, or ask the administrators to provide you with a copy of scan results. System scans will usually identify one or more vulnerabilities and a standard criticality rating to each. Ensure that the results of the scans are in line with company policy on vulnerability scanning.

See Chapter 4 for more information about vulnerability scanning.

Security Monitoring and Other General Controls

16. Ensure that only approved applications are installed on the system per your server management policy.

Administrators must manage the set of applications installed on their hosts for the following reasons:

- Not all applications play well together.

- Applications may have a dependency that's not installed.
- More applications mean more areas of potential compromise.

Unmanaged or unknown applications also may have configuration or coding issues that make the server vulnerable to compromise. For example, a poorly managed application could be missing patches, could allow access to a privileged process, or could inadvertently create a covert channel for an unprivileged user.

How

Use the results from the output of `psinfo –s`, which includes information about the installed applications. You might also consider looking through Process Explorer. Compare your findings with organizational policy and discuss them with the administrator.

17. Review and verify startup information.

Rogue partitions, processes, or programs in violation of your policies can sometimes be found during system startup. In addition, malware will sometimes make use of the next reboot to install kits deeper into the OS.

How

Several utilities can help you dissect what the next reboot will do to the system. Two excellent tools include `pendmoves` and `autorunsc`. You can use `pendmoves` by itself without any switches to understand what file moves are planned for the next system restart.

`Autoruns` is the GUI version of `autorunsc`. When you use `autorunsc` from the command line, it might be easier to output it to a comma-separated values (CSV) file with the `–c` switch and view the results inside Excel. It might be difficult to appreciate the power of `autorunsc` until you use the GUI `autoruns` version to see the information it's capable of uncovering for you.

18. Ensure that only approved scheduled tasks are running.

Scheduled tasks can stay hidden for weeks until an administrator takes the time to view the running scheduled tasks on the host. Scheduled tasks created by malicious or unknown sources could damage host or network resources.

How

Note that reading scheduled tasks from the command line doesn't show you what the task is really going to do. The task can be called anything an attacker wants to call it while setting it up. That being said, you can view tasks from the command line using `schtasks`:

```
The current directory is C:\>
schtasks
TaskName                         Next Run Time          Status
================================ ====================== ===========
Malicious Task                   12:27:00 PM, 6/13/2011
```

If you want to understand in depth exactly what each task does, you need to open the properties of each task independently. From there, you also can see the target file and review several other settings. Choose Search and type **schedule**. Then select Task Scheduler. Alternatively, you could type `taskschd.msc` at the command line to open the Task Scheduler.

19. Ensure that the server has auditing enabled per your organization's policies.

Auditing provides evidence in the aftermath of an event and helps with troubleshooting issues on the host. Ideally, an event correlation engine would filter and produce meaningful data for the system administrator. Until that day comes, it is important that you have auditing enabled to provide a record for what happens on the host.

How

You should view your audit settings manually with the Group Policy panel (`secpol.msc`). Find the settings under Security Settings | Advanced Audit Policy Configuration | System Audit Policies – Local Group Policy Object. Some suggested settings are shown in Table 7-4. These and other settings are discussed at https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/audit-policy-recommendations.

Audit Policy	Audit Settings	
Audit Audit Policy Change	Success	Failure
Audit Authentication Policy Change	Success	
Audit Computer Account Management	Success	Failure
Audit Credential Validation	Success	Failure
Audit IPSec Driver	Success	Failure
Audit Logoff	Success	
Audit Logon	Success	Failure
Audit Other Account Management Events	Success	Failure
Audit Process Creation	Success	
Audit Security Group Management	Success	Failure
Audit Security State Change	Success	Failure
Audit Security System Extension	Success	Failure
Audit Special Logon	Success	Failure
Audit System Integrity	Success	Failure
Audit User Account Management	Success	Failure

Table 7-4 Common Audit System Settings

You should consider that Windows audit policy settings can affect disk usage and system utilization. You can quickly fill your logs and tax your system with meaningless overhead if this is misused. Ensure a target system has sufficient resources before enabling these settings.

20. Review and evaluate system administrator procedures for monitoring and maintaining the state of security on the system.

If the system administrator doesn't monitor his or her systems for unexpected changes, security incidents could occur without his or her knowledge. By monitoring, we mean *actively* reviewing log data and system information. Merely enabling log collection without reviewing the resulting data is just barely preferable to having no log data at all.

System security must also be *maintained*. The world of security vulnerabilities is an ever-changing one, and it is unrealistic to believe that a static audit program can provide assurance of system security on a daily basis. A vulnerability scanning tool that is updated frequently can provide an effective mechanism for understanding the current security state of the machine. In addition, if the system administrator has a security patching process in place, this scan will provide some validation of the effectiveness of that process.

How

Interview the system administrator and review any relevant documentation to get an understanding of security monitoring practices. Some level of monitoring is important, but the monitoring level required should be consistent with the criticality of the system and the inherent risk of the environment (for example, a web server in the DMZ should have more robust security monitoring than a print server on the internal network). The system administrator is responsible for monitoring his or her hosts for issues such as those you have been auditing for throughout the audit steps in this chapter.

If security monitoring is performed, assess the frequency of the monitoring and the quality with which it is performed. Look for evidence that the security monitoring tools are actually used. Review recent results, and determine whether they were investigated and resolved. Leverage the results of the rest of the audit in performing this assessment. For example, if you found significant issues in an area they were supposedly monitoring, it might lead to questions as to the effectiveness of that monitoring.

Note that some organizations may configure server event logs to be sent to centralized logging environments to be reviewed by dedicated teams or even by outside service providers. In these cases, discuss server monitoring practices with the monitoring team.

21. If you are auditing a larger environment (as opposed to one or two isolated systems), determine whether there is a standard build for new systems and whether that baseline has adequate security settings.

Consider auditing a system freshly created from the baseline. One of the best ways to propagate security throughout an environment is to ensure that new systems are built correctly before moving into testing or production.

How

Through interviews with the system administrator, determine the methodology used for building and deploying new systems. If a standard build is used, consider auditing a newly created system using the steps in this chapter. Here is where something like Microsoft's Configuration Manager best comes into play; you can report on the deviations from the baseline and work on auditing just the deltas. Additionally, this is the time to ask your virtualization administrators for information about the baselines they use to create virtual servers.

NOTE Consider discussing an approval process for new standard builds in which an auditor would look over the changes and perform a full audit of new images. This is a great way for the audit team to create a working relationship with the Windows server team.

22. Perform the steps from Chapter 3 and 5 as they pertain to the system you are auditing.

In addition to auditing the logical security of the system, you need to ensure that appropriate physical controls and operations are in place to provide for system protection and availability.

How

Reference the steps from Chapter 3 and 5 and perform those that are relevant to the system being audited. For example, the following topics are likely to be pertinent:

- Asset inventory
- Physical security
- Environmental controls
- Capacity planning
- Change management
- Backup processes
- Disaster recovery planning

Tools and Technology

Several of the tools mentioned in this chapter are free and easily accessible. You are encouraged to download them and play with them on your personal machine, but be careful. Some of them are powerful and should be tested in a testing network prior to use in a production environment. Table 7-5 lists some of the tools you might consider as you look into auditing Windows.

Resource	Website
Microsoft Script Center	https://gallery.technet.microsoft.com/scriptcenter
Microsoft Command-Line Reference	https://technet.microsoft.com/en-us/library/cc754340(WS.10).aspx
Microsoft Sysinternals Tools	https://docs.microsoft.com/en-us/sysinternals/

Table 7-5 Tools and Technology: Auditing Windows

Knowledge Base

The following table shows additional resources where you can obtain information about Windows environments and related controls. Microsoft has a tremendous amount of information on its website for general consumption. Additionally, the community of helpful enthusiasts and social forums continues to grow.

Resource	Website
Microsoft Server and Tools	www.microsoft.com/en-us/cloud-platform/windows-server
Microsoft TechNet	www.technet.com
Microsoft System Center	www.microsoft.com/systemcenter
Windows Intune	www.microsoft.com/en-us/cloud-platform/microsoft-intune
Microsoft Tech-Ed Online	https:docs.microsoft.com/en-us/intune/what-is-intune
TCP/IP Fundamentals for Windows	https://technet.microsoft.com/en-us/library/cc307741.aspx
Secure Windows Server	https://technet.microsoft.com/en-us/library/dd548350(WS.10).aspx
Windows Firewall with Advanced Security	https://technet.microsoft.com/en-us/library/dd772715(WS.10).aspx
The Center for Internet Security	www.cisecurity.org
Computer Security Resource Center	https://csrc.nist.gov
KeePass Password Tool	https://keepass.sourceforge.net

Master Checklist

The following table summarizes the steps listed earlier for auditing Windows servers and clients.

Auditing Windows Servers

Checklist for Auditing Windows Servers
❑ 1. Obtain the system information and service pack version and compare with policy requirements.
❑ 2. Determine whether the server is running the company-provisioned firewall.
❑ 3. Determine whether the server is running a company-provisioned antivirus program.
❑ 4. Determine whether the server is running a company-provisioned patch management solution.
❑ 5. Ensure that all approved patches are installed per your server management policy.
❑ 6. Review and evaluate procedures for creating user accounts and ensure that accounts are created only when there's a legitimate business need. Review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.
❑ 7. Ensure that all users are created at the domain level and clearly annotated in Active Directory. Each user should trace to a specific employee or team.
❑ 8. Review and evaluate the use of groups, and determine the restrictiveness of their use.
❑ 9. Review and evaluate the strength of passwords and the use of password controls on the server, such as password aging, length, complexity, history, and lockout policies.
❑ 10. Review and evaluate the use of user rights and security options assigned to the elements in the security policy settings.
❑ 11. Look for and evaluate the use of file sharing on the host.
❑ 12. Review and evaluate the use and need for remote access, including RAS connections, FTP, Telnet, SSH, VPN, and other methods.
❑ 13. Ensure that a legal warning banner is displayed when users connect to the system.
❑ 14. Determine what services are enabled on the system and validate their necessity with the system administrator.
❑ 15. Evaluate vulnerability scanning procedures and ensure that known vulnerabilities are resolved.
❑ 16. Ensure that only approved applications are installed on the system per your server management policy.
❑ 17. Review and verify startup information.
❑ 18. Ensure that only approved scheduled tasks are running.
❑ 19. Ensure that the server has auditing enabled per your organization's policies.
❑ 20. Review and evaluate system administrator procedures for monitoring and maintaining the state of security on the system.
❑ 21. If you are auditing a larger environment (as opposed to one or two isolated systems), determine whether there is a standard for new systems and whether that baseline has adequate security settings.
❑ 22. Perform the steps from Chapters 3 and 5 as they pertain to the system you are auditing.

CHAPTER 8

Auditing Unix and Linux Operating Systems

This chapter discusses the steps required for auditing Unix- and Linux-based operating systems (also referred to as **nix systems*) and includes the following:

- The history of Unix and Linux
- Basic commands for getting around in the *nix environment
- How to audit Unix and Linux systems, focusing on the following main areas:
  - Account management
  - Permissions management
  - Network security and controls
  - Security monitoring and other general controls
- Tools and resources for enhancing your *nix audits

Background

Unix dates back to 1969, when it was developed by employees at AT&T for the purpose of providing an environment in which multiple users could run programs. Strong security was not one of the goals of its development.

In the late 1970s, students at the University of California, Berkeley, made extensive modifications to the AT&T Unix system, resulting in the *Berkeley Software Distribution (BSD)* Unix variant, which became very popular in academic circles. Around the same time, AT&T began a push to develop its Unix operating system into a legitimate commercial product called *AT&T System V* (or just *System V*).

During the 1980s, as commercial interest in the Unix operating system grew, companies faced the dilemma of deciding which of the two versions of Unix to adopt. Sun Microsystems' SunOS and Digital Equipment Corporation's Ultrix were based on the BSD. Other companies that tried to develop a Unix-based OS, including Hewlett-Packard (HP), IBM, and Silicon Graphics, used System V as their standard. Microsoft developed a third version of Unix, called *Xenix,* and licensed it to Santa Cruz Operations (SCO). Xenix was based on a prior version of the AT&T Unix operating system.

All these versions of Unix obviously resulted in confusion in the industry and frustration for vendors who were attempting to develop software for use on Unix-based platforms. This resulted in the merging of some versions, beginning with Xenix and AT&T's System V in 1988. Next was a merger of AT&T and Sun's

versions, called System V Release 4 (SVR4), which was to be compatible with programs written for either System V or BSD. Sun later named its proprietary version of this operating system Solaris (which was later renamed Oracle Solaris after Sun was acquired by Oracle). Not to be left out, a number of the other companies, such as IBM and HP, formed an organization called the Open Software Foundation (OSF), whose mission was to put control of Unix in the hands of a not-for-profit group. The OSF operating system (OSF/1) was never widely adopted, and the individual companies continued to develop and use their own proprietary Unix variants, such as IBM's AIX, HP's HP-UX, SCO Unix, and IRIX.

Linux, a "Unix-like" operating system, came on the scene with a Usenet posting in 1991 by its author, Linus Torvalds. Strictly speaking, Linux is a *kernel* and not an operating system, because what Torvalds developed was the piece that allows other programs to run. Most of these other programs that allow the system to be truly usable came from the GNU project. Hence, many people refer to Linux as *GNU/Linux* when speaking of it as an entire OS, but since this subject is a bit of a religious war, we won't discuss it further here.

From these humble, hobbyist beginnings in 1991, Linux grew to a 1.0 release in 1994. But even before the 1.0 release, a number of Linux "distributions" were developed, combining the Linux kernel with applications and system utilities. Some examples of today's popular distributions are Red Hat, Ubuntu, Debian, SUSE, and Mint. Although many aspects of all Linux distributions are identical or very similar, they included some differences as well, such as package management and the init system. Support models differ as well, and when you pay for a Linux distribution, you're typically paying for the support because the software itself is free. This free software, combined with the ability to run on generic x86/64-bit–based hardware, has made Linux a compelling choice for both enterprise and personal computing needs.

NOTE As you can see from this history, there are many variations of the Unix and Linux OSs. Although the information and concepts in this chapter are generic and applicable to all versions, it would take more space than is feasible to note the nuances for each *nix version. This chapter therefore focuses on Solaris (Unix) and Red Hat (Linux) where version-specific commands and examples are required.

Unix and Linux Auditing Essentials

If you are new to the Unix world, you'll find it helpful to obtain access to a Unix/Linux system while reading through this section. Try the commands for yourself to become familiar with them.

Windows users can easily turn their machine into a Linux system without altering the Windows file system. Just download an image and create either a bootable CD or USB drive, such as the popular Knoppix (www.knoppix.org), and boot into a full-featured Linux distribution. If you have a spare PC to work with, visit https://distrowatch.com and select Major Distributions. You'll find a wealth of information about the various free distributions available there.

NOTE When you're learning these commands, remember that you can access help at any time by typing `man <commandname>` for comprehensive help or `<commandname> --help` for abbreviated help.

Key Concepts

Before we start digging into the details, let's establish some key concepts upon which we can build:

- Everything in Unix is a file. For example, if you type in a command and press ENTER, you are actually executing a file within the system that has the same name as the command you entered. And if you attach a device, such as a printer or storage, to your Unix system, it will be represented on the system as a file.
- The root of the Unix file system is the directory called root or /. Every directory and every file branches off this root directory. Since everything in Unix is a file, if you do a recursive listing off of the / directory, you will see every component of the system.
- The system administrator (or superuser) account in Unix is called "root." This account has full control over the system.
- If you can alter a file that someone is executing, you can easily capture (i.e., compromise or "become") his or her account.

File System Layout and Navigation

The file system can be thought of as a tree, and the base of every tree is its root. So the root directory, designated /, is the trunk from which other directories branch. Every Unix system has a root directory, but you will find some variance in what you see from there. Table 8-1 lists some common directories that you usually will find.

Directory	Description
/bin	Location of most of the system binaries (programs)
/sbin	Contains binaries that are reserved for use by privileged accounts
/etc	Contains system configuration files
/boot	Contains the location of the kernel in many systems
/home	Typical location for user home directories
/var	Contains information that programs need to track as they run (such as their process ID on the system); usually contains log files as well
/lib	System and application libraries that aren't executed directly but are used by applications as they run
/opt	Includes many installed add-on packages
/usr	Another place for user-added packages; often duplicates many of the top-level directories within itself, so you'll have /usr/etc, /usr/bin, and so on; documentation is often placed into /usr/share
/root	Often contains the home directory for the root account
/tmp	Temporary directory that any user typically can access; often cleared when the system is booted
/mnt	Remote file systems or removable media may be mounted here
/dev	Represents the concept that everything is a file, so you will find device files here representing the hardware in your system
/proc	This pseudo-file system doesn't exist on a physical disk but contains memory-resident information about both the processes running on a system and the system itself

Table 8-1 Common Unix and Linux Directories

Several essential commands typed at the command prompt can be helpful for

navigating Linux and Unix file systems. The most essential commands are shown in Table 8-2 along with some common and helpful switches.

Command	Meaning	Description	Tips for Use
`cd`	Change directory	Changes directory location as you would from the Windows command prompt	`cd ~` changes directory to user's home directory `.` signifies current directory `..` signifies parent directory
`ls`	List directory contents	Lists the contents of a directory along with information, such as ownership, permissions, file size, and so on, when used with the `-l` option	`ls -l` uses long listing format for the files within the directory `ls -ld` provides the long listing format for the directory itself `ls -al` provides the long listing format for all directory contents, including hidden files `ls -alR` provides a recursive directory listing, using the long listing format and displaying hidden files `ls -altr` provides the long listing format, displaying the directory's contents in reverse chronological order
`pwd`	Print working directory	Displays the current working directory on the screen	Auditors can use this command when copying screen output for an audit to show on work papers where they are working on the system

`more`
`cat`
`less`
	List file contents	Lists the contents of a file	`cat` displays all the file's contents at once
			`more` displays the file's contents one page at a time
			`less` displays the file's contents one page at a time and allows backward navigation

`ypcat`	List NIS file contents	Lists the contents of a centralized NIS file	Displays the contents of the NIS password and group files if you're using NIS for centralized account management
`su`	Switch user	Allows a user to switch to another user ID	Works only if you have root access or if you know the password of the account to which you want to switch

Table 8-2 Common Linux and Unix Navigation Commands

NOTE When navigating a *nix system, the presence or absence of the leading `/` in the path is very important; if present, it serves to anchor the path at the root directory. Thus, if you are currently in /usr, `cd /bin` and `cd bin` will take you to two different places (/bin and /usr/bin, respectively). These are known as *absolute* or *relative path names.* The absolute path always starts with `/` and traces the entire path from the root directory. The relative path, with no leading `/`, starts with the present directory.

File System Permissions

File and directory permissions can be separated into user, group, and world permissions. In other words, each file and directory has permissions set for the owner of the file, for the group associated with the file, and for everyone else (often called "world" or "other"). Each of these entities can be granted read, write, and/or execute access. Both files and directories have their own permission sets. You can see how this can get tricky, but remember that the most restrictive set of permissions wins every time. For example, if a file has world-read permissions but is restricted under its parent directory to disallow world-read permissions, then the world (meaning everyone) will not be able to read the file.

You will notice that these permissions are shown in two ways. Some places use three sets of `rwx` for read, write, and execute. The three sets are for the owner, group, and world. An example might be `rwxr-xr--`. This means that the file's owner has read, write, and execute permissions on the file; the file's group has read and execute permissions; and everyone else has read permissions. Another example might be `rw-r-----`. This means that the file's owner has read and write permissions on the file, the file's group has read permissions, and everyone else has no permissions.

Other places use a three-digit number such as `754`, which is identical to the `rwxr-xr--` and is shown in Figure 8-1. For those who never studied binary numbers, just remember that read is worth 4 points, write is worth 2, and execute is worth 1. Add them up for each set (that is, owner, group, and world), and you have your permissions. Thus `754` is a way to say, "I don't mind if other people read this file and if those in my group execute this file, but only I should be able to modify it." As additional examples, permissions of `rw-r-----` would be represented as 640 and `rwxrwxrwx` would be represented as 777.

Permissions on a file or directory:

	Owner			Group			World		
Available permissions	Read	Write	Execute	Read	Write	Execute	Read	Write	Execute
Assigned permissions	r	w	x	r		x	r		
Binary (1 = Yes, 0 = No)	1	1	1	1		1	1		
Decimal value	4	2	1	4		1	4		
Cumulative result	7			5			4		

Resulting permissions: 754 (or) `rwxr-xr--`

Figure 8-1 Unix permissions

Finally, note that file permissions are not completely independent of the permissions of the directory that contains the file. This interaction is illustrated in Figure 8-2. For example, if you have rwx access to a file but that file is sitting in a directory to which you have no access, you will not be able to read, write to, delete, or otherwise access the file. Conversely, if you have no access to a file but that file is sitting in a directory to which you have write and execute access, you will be able to delete that file, as the combination of write and execute access is what allows a user to delete files from and add files to a directory. In this scenario, you would be able to delete the file and then create a new file with the same name within that directory, opening up the possibility of file spoofing. It is therefore critical when you're evaluating the security of a file to evaluate related directory permissions.

File Permissions \ Directory Permissions	-	r	x	wx
-	No access	No access	No access	Delete file
r	No access	No access	Read data	Delete file or read data
w	No access	No access	Add to or clear data	Delete file or add to or clear data
rw	No access	No access	Update data	Delete file or update data
x	Can't execute	Can't execute	Execute	Delete file or execute

Figure 8-2 Interaction between file and directory permissions

NOTE Execute permissions on all parent directories back to `/` are required of a user to perform operations on a file within that path. For example, permissions are `777` on a file in /home/andrew, but permissions in the andrew directory are `700`. Nonroot users other than Andrew will not be able to read or delete that file.

Users and Authentication

Access to a Unix system is typically controlled by means of a username and password. This authentication information may be kept on the local file system, or it may be kept in a central location on the network, where many systems can access the same information. In the simplest case, where all the information is local, we typically would consider three files: /etc/passwd, /etc/shadow, and /etc/group.

Unix Password File

The /etc/passwd file (Table 8-3) contains account information for all users. Each account on the local system will have a single line in the /etc/passwd file. The system refers to this file when a user attempts to authenticate.

Field	Use
account	Represents the user to the system. This name is used when the user logs in.
password	Encrypted password. It may be kept in /etc/shadow instead; if so, this field simply will contain an `*`, `x`, `!`, or other character.
UID	Numeric user ID.
GID	Numeric group ID for the user's primary group.
GECOS	Optional field used to store arbitrary additional information about the account. A typical use would be the real name and/or employee ID of the user.
directory	Location of the user's home directory.
shell	User's default shell, the command-line environment that interprets commands and passes them to the kernel.

Table 8-3 Components of a Unix Password File

Lines in /etc/password have this format:

```
account:password:UID:GID:GECOS:directory:shell
```

Unix Shadow File

By design, the /etc/passwd file (Table 8-4) allows world read access. Therefore, if the encrypted password is kept in that file, any user on the system would be able to download all users' encrypted passwords and attempt to crack them using freely available password-cracking software. To mitigate this risk, most systems store the encrypted password inside a shadow password file, which is readable only by root. The shadow password file is complementary to the /etc/passwd file, with a corresponding line for each user.

Field	Use
account	Name representing the user to the system.
password	Encrypted password; `*LK*` indicates that the account is locked.
lastchange	Number of days since the password was changed.
min	Minimum number of days allowed between password changes.
max	Maximum number of days allowed between password changes.
warn	Number of days before max, at which point the user will be warned to change his or her password.
inactive	Number of days of inactivity after which the user's account will be disabled.
expired	Number of days since January 1, 1970, that the account has been disabled.
reserved	An extra field that is not used.

Table 8-4 Components of a Unix Shadow File

Lines in /etc/shadow have this format:

```
account:password:lastchange:min:max:warn:inactive:expired:reserved
```

Unix Group File

The /etc/group file (Table 8-5) contains information on groups on the system.

Field	Use
name	Name of the group.
password	Group password, if one is used.
GID	Numeric group ID.
users	List of users who are members of the group, although members of the group who are assigned to it through their GID in /etc/password (see Table 8-3) won't necessarily be on this list.

Table 8-5 Components of a Unix Group File

Lines in /etc/group use this format:

```
name:password:GID:users
```

LDAP, NIS, or NIS+

In more complicated cases, credentials can be checked against an authentication database located on the network; typically, this is Lightweight Directory Access Protocol (LDAP), Active Directory, Network Information System (NIS), or NIS+. You may be able to determine whether one of these is used in preliminary discussions

with the system administrator, or you may want to look at the systems yourself.

Determine whether NIS, NIS+, or LDAP is used by looking at the line beginning with `passwd` in `/etc/nsswitch.conf`. The presence of `nis`, `nisplus`, or `ldap` on that line indicates use of those protocols. These typically will be present in addition to `files`, which refers to the local password file. You also may see `compat`, which enables the use of + and — in the local password file for NIS/NIS+. If `compat` mode is used, then a + at the beginning of a line in /etc/passwd would indicate that NIS/NIS+ is being used. Review of the `passwd_compat` entry in /etc/nsswitch.conf should allow you to distinguish between the two. Note that local access can show you only what you need to know about local Unix authentication. You may need more information to determine the effectiveness of a network authentication scheme such as NIS or LDAP. For these, you may want to do a separate review of the particular authentication infrastructure.

Network Services

To understand areas of potential risk in your environment, you must know the avenues by which a system can be accessed, and you need to be able to determine what network services are enabled on the system. On most systems, including older ones, you can use the `netstat` command to see this information. The most generic usage would be `netstat -an`, which will list a lot of information. Services running on Transmission Control Protocol (TCP) ports that are listening for external connections usually will say `LISTEN` in the output. User Datagram Protocol (UDP) ports may say `IDLE` on some systems such as Solaris. On Linux, look for UDP ports that have a listed `Remote Address` of `0.0.0.0`.

Once you have identified the open ports, you should determine what applications (often called *daemons*) are running on them. You often can determine this by mapping the port to the list of well-known ports maintained by the Internet Assigned Numbers Authority (IANA) at www.iana.org/assignments/port-numbers. However, you should be aware that, for example, even though TCP port 25 is supposed to be for SMTP, there's no reason you can't run a web server on that port instead. If you have any questions about port number assignment, ask the system administrator. You also may want to use some of the tools listed in the "Tools and Technology" section later in this chapter that can automate the process of identifying open ports and the applications running on them.

A newer alternative to netstat is the `ss` command, and invoking `ss -ap` will show both sockets and the applications to which they belong.

Test Steps for Auditing Unix and Linux

The following audit steps are divided into four sections:

- Account management
- Permissions management
- Network security and controls
- Security monitoring and other general controls

NOTE The test steps in this chapter focus on testing the logical security of Unix and Linux boxes, as well as processes for maintaining and monitoring that security.

However, other internal controls are critical to the overall operations of a Unix/Linux environment, such as physical security, disaster recovery planning, backup processes, change management, capacity planning, and system monitoring. These topics are covered in Chapter 5 and should be included in your audit of the Unix/Linux environment if they have not already been covered effectively in a separate data center or entity-level IT controls audit.

Account Management

Most of the steps in this section require some form of testing over the system's password file. Prior to commencing work on these steps, the auditor should determine whether the system is using only its local password file (/etc/passwd) or some additional form of centralized account management such as NIS or LDAP. If the latter form is used, the auditor must execute the following steps on both the centralized password file and the local password file. The same concept applies for the steps that reference the group file.

In the "How" sections of the following steps, we will not attempt to specify the commands for every possible centralized account management system, because there are a number of vendor-specific tools. We will include the details for pulling information from NIS, which is one of the most common of these systems, as an example. If your company uses a different tool, such as NIS+ or LDAP, you will need to work with your system administrator and review the documentation for these systems to determine the equivalent commands. However, the concepts described here for the local and NIS password and group files will apply.

1. Review and evaluate procedures for creating Unix or Linux user accounts and ensuring that accounts are created only when there's a legitimate business need. Also, review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.

If effective controls are not in place for providing and removing access to the server, it could result in users having unnecessary access to system resources. This, in turn, places the integrity and the availability of the server at risk.

How

Interview the system administrators, and review account creation procedures. This process should include some form of verification that every user has a legitimate need for access. Take a sample of accounts from the password file, and review evidence that they were approved properly prior to being created. Alternatively, take a sample of accounts from the password file and validate their legitimacy by investigating and understanding the job function of the account owners.

Also review the process for removing accounts when access is no longer needed. This process could include an automated feed from the company's human resources (HR) system providing information on terminations and job changes. Or the process could include a periodic review and validation of active accounts by the system administrators and/or other knowledgeable managers. Obtain a sample of accounts from the password file, and verify that they are owned by active employees and that those employees' job positions have not changed since the account's creation.

2. Ensure that all user IDs in the password file(s) are unique.

If two users have the same user ID (UID), they can fully access each other's files and directories and can "kill" each other's processes. This is true even if they have different usernames. The operating system uses the UID to identify the user. It merely maps the username to the corresponding UID in the password file.

How

For local accounts, use the command more /etc/passwd, and review the entries to ensure that there are no duplicate UIDs. If NIS is used, the command ypcat passwd also should be used so that NIS UIDs can be examined.

The following command will list any duplicate UIDs found in the local password file:

```
cat /etc/passwd | awk -F: '{print $3}' | uniq -d
```

3. Ensure that passwords are shadowed and use strong hashes where possible.

For the system to function appropriately, the password file needs to be world-readable. This means that if the encrypted passwords are contained within the file, every user on the system will have access to them. This, in turn, gives users the opportunity to copy the encrypted passwords and attempt to crack them via password-cracking tools that are freely available on the Internet. Given enough time, a brute-force cracking tool can guess even the most effective password. Also consider the form of the passwords. The crypt routine traditionally used for Unix passwords is a relatively weak form of encryption by today's standards, and the maximum effective password length is eight characters. A better choice is to use SHA-512 or MD5 hashes, which are more difficult to crack and allow more than eight characters for the password. An even better choice is to use a dedicated password-hashing scheme, such as bcrypt.

How

To determine whether a shadow password file is being used, type the `more /etc/passwd` command to view the file. Look within the password field for all accounts. If each account has an "`*`" or "`x`" or some other common character in it, the system uses a shadow password file. The shadow password file will be located at /etc/shadow for most systems. Systems using NIS create some special problems that make the use of shadowed passwords more difficult, and older systems cannot shadow these passwords at all. If NIS is used in your environment, consult with the system administrator to discuss the possibilities of shadowing these passwords. If it is not possible to do so, consider other password-related policies.

MD5 is the default hash on some Linux systems, although it has been found to have vulnerabilities, making SHA-512 (or other members of the SHA-2 family) a more secure option. Other distributions use the bcrypt function by default. The crypt form can be recognized because it is always 13 characters long; an MD5 or SHA-512 hash in /etc/passwd or /etc/shadow will be prepended with the characters `$1$` (for MD5) or `$6` (for SHA-512) and is longer. A bcrypt hash will be prepended with the characters `$2a$`, `$2b$`, or `$2y$` (and will also be longer). As you can see, this is a complex topic. Your best bet is to spend time with your system administrator to understand how passwords are currently hashed and to discuss improvement options.

4. Evaluate the file permissions for the password and shadow password files.

If a user can alter the contents of these files, he or she will be able to add and delete users, change user passwords, or become a superuser by changing his or her UID to `0`. If a user can read the contents of the shadow password file, he or she can copy the encrypted passwords and attempt to crack them.

How

View the file permissions for these files by using the `ls –l` command on them. The /etc/passwd file should be writable only by "root," and the /etc/shadow file also should be readable only by "root."

5. Review and evaluate the strength of system passwords and use of stronger forms of authentication.

If passwords on the system are easy to guess, it is more likely that an attacker will be able to break into that account, thus obtaining unauthorized access to the system and its resources. Also, standard account credentials (username and password) can be compromised via malware, users succumbing to phishing and other forms of social engineering, etc. A simple ID and password are often not enough to protect access to the most sensitive systems, in which case stronger authentication mechanisms (e.g., two-factor authentication) are called for.

How

Review system settings that provide password composition controls. For Solaris systems, the password policy is usually set in /etc/default/passwd. Use a `more` command on this file, and view the `PASSLENGTH` parameter to determine minimum password length. Compare the value of this parameter with your company's information security policy. Most Linux systems have /etc/login.defs, which provides basic controls such as minimum password length and maximum password age for locally created accounts.

Unfortunately, the standard Unix `passwd` program does not provide strong capabilities for preventing weak passwords. It will prevent a user from choosing his or her username as a password but not much else. Through discussions with the system administrator, you can determine whether other tools have been implemented either to replace or enhance the native `passwd` functionality for password composition requirements. Additional controls also can be provided through Pluggable Authentication Modules (PAM) by the use of `pam_cracklib`, `pam_passwdqc`, or a similar module (`pam_cracklib` is included in many Linux distributions). Look for lines beginning with `password` in /etc/pam.conf or the configuration files in /etc/pam.d/ to get an idea of what's in use on the system you're auditing. Perform a `more` command on these files to view their contents. System administrators can also supplement native Unix capabilities by implementing up-front or periodic after-the-fact testing of password strength using open-sourced password-cracking tools in order to identify weak passwords.

While a thorough audit may involve cracking passwords, we suggest in general that you leave any password-cracking exercises to professional penetration testers, or consult your information security team for help. However, if appropri-

ate per your company's policies and practices, consider obtaining a copy of the password file and the shadow password file and executing a password-cracking tool against the encrypted passwords to identify weak passwords. See the "Tools and Technology" section later in this chapter for information on password-cracking tools. Use good judgment in interpreting the results, because a brute-force cracking tool will eventually crack any password if given enough time. If the password files have been shadowed, you really need to worry only about truly weak passwords that are obvious and easy to guess. These sorts of passwords likely will be guessed within the first 30 to 60 minutes by someone running a password-cracking program. On the other hand, if the password files have not been shadowed, you likely will want to run the program for much longer, because anyone with access to the system will have the ability to do the same thing (and, more importantly, you would expect to see the system administrators implementing their own password-cracking, or equivalent, controls to compensate for the absence of shadowed passwords). If you go down the route of obtaining the encrypted passwords in order to attempt to test their strength, be sure that the data and results are stored in an appropriately protected location.

Through conversations with the system administrator, determine whether stronger forms of authentication have been implemented for accessing the server. Stronger authentication mechanisms (beyond a simple ID and password) might be appropriate based on the criticality of the system being reviewed, and a number of commercial and open-source tools are available that can supplement native Unix authentication capabilities. As mentioned earlier, standard account credentials can and will be compromised over time, so stronger forms of authentication (e.g., two-factor authentication) are sometimes needed in order to protect confidential information and critical processing. This strong authentication might be needed for all access to the server or possibly just for access to the most sensitive accounts. Explore the need for this additional protection through conversations with the system administrator and core users of the applications and/or services being hosted on the server under review.

6. Evaluate the use of password controls such as aging.

It is important to change passwords periodically for two primary reasons. First, without aging, an attacker with a copy of the encrypted or hashed passwords will have an unlimited amount of time to perform an offline brute-force cracking attack. Second, someone who already has unauthorized access (e.g., through cracking, hacking, or just password sharing) will be able to retain that access indefinitely.

How

Review system settings that provide password aging controls. For Solaris systems, the password policy is usually set in /etc/default/passwd. Perform a `more` command on this file and view the `MAXWEEKS` parameter to determine the maximum age for passwords and the `MINWEEKS` parameter to determine the minimum age for passwords. Minimum age is important to prevent a user from changing his or her password and then immediately changing it back to its previous value. View the settings of these parameters and compare them with your company's information security policy.

Most Linux systems have /etc/login.defs, which provides basic controls such as minimum password length and maximum password age for locally created accounts. Additional controls can be provided through PAM by the use of `pam_cracklib`, `pam_passwdqc`, or a similar module (`pam_cracklib` is included in many Linux distributions). Look for lines beginning with `password` in /etc/pam.conf or the configuration files in /etc/pam.d/ to get an idea of what's in use on the system you're auditing. Perform a `more` command on these files to view their contents.

The root account generally will not be subject to automatic aging to prevent the possibility of the account being locked. However, a manual process should be in place for periodically changing the password in accordance with company policy. Review the process for changing this password, and look for evidence that it is being followed.

7. Review the process used by the system administrator(s) for setting initial passwords for new users and communicating those passwords.

When new user accounts are created, the system administrator must assign an initial password to that user. If that password is easy to guess, it could allow the account to be hacked, resulting in unauthorized access to the server and its resources. If the initial password is not communicated via a secure channel, it could allow others to view the password and obtain unauthorized access to the account.

How

Interview the system administrator, and review documentation to understand the mechanism used for creating initial passwords. Ensure that this mechanism results in passwords that are difficult to guess and that comply with your company's IT security policy.

Also, review the channels used for communicating the new passwords to users. Ensure that unencrypted transmissions are not used. Finally, it is often a good idea for the user to be required to change his or her password immediately on first login. Interview the system administrator to determine whether or not this is done. Accounts can be expired, thus forcing the user to change his or her password on the next login by the use of `passwd –f` on Solaris and `passwd –e` on Linux. These commands will expire a user's account immediately, forcing the user to change it on the next login. These are not items that can really be checked for, other than asking the system administrator how he or she does things.

8. Ensure that each account is associated with and can be traced easily to a specific employee.

If the owner of an account is not readily apparent, it will impede forensic investigations regarding inappropriate actions performed by that account. If multiple people use an account, no accountability can be established for actions performed by that account.

How

Review the contents of the password file(s). The owner of each account should be obvious, with the user's name or other unique identifier (such as employee number) either used as the username or placed in the GECOS field. Question any accounts that seem to be shared, such as guest or application accounts. If accounts such as these are required, they should be configured with restricted shells and/or such that a user cannot directly log into them (thus requiring the user to log in as himself or herself first and then using `su` or `sudo` to access the shared account,

creating an audit trail).

9. Ensure that invalid shells have been placed on all disabled accounts.

This is only a significant risk if trusted access is allowed (see the "Network Security and Controls" section later in the chapter). If trusted access is allowed, a user with a certain username on one system (the trusted system) can log into an account with that same username on another system (the trusting system) without entering a password. This can be done as long as the user account on the trusting system has a valid shell assigned to it, even though the account may have been disabled. Therefore, if a system administrator disables an account but leaves it with a valid shell, a user on a remote, trusted system with the same username still could access that account.

How

View the contents of the password files (via the `more` command). If an account has been disabled, it will show an "*," "*LK*," or something similar in the password field (remember to look in the shadow password file if it is being used). For those accounts, review the contents of the shell field. If it contains anything other than `/dev/null, /bin/false`, or something similar, the account probably still can access a valid shell or program.

10. Review and evaluate access to superuser (root-level) accounts and other administration accounts.

An account with root-level access has the ability to do anything with the system, including deleting all files and shutting the system down. Access to this ability should be minimized. Other accounts may exist on the system for the purpose of administering specific applications and also should be tightly controlled to prevent system disruption.

How

Review the contents of the password files, and identify all accounts with a UID of `0`. Any account with a UID of `0` is treated by the system as if it were the root account. Question the need for any account besides root to have a UID of `0`. Determine via interviews who knows the passwords to the root and other UID `0` accounts, and evaluate the appropriateness of this list. Also, audit the process that is used by the system administrators to document and communicate the passwords for these accounts, as these will likely be shared between the members of the team. Preferably, these passwords will be stored in a password vault of some kind and require a controlled, auditable checkout feature in order to obtain the password.

Review the password file for the existence of other administration accounts (such as "oracle"). You likely will have identified potential candidates when performing step 8. Determine via interviews who knows the passwords to these accounts, and evaluate for appropriateness. Review the process for documenting and communicating these passwords as well.

Many environments use `sudo` or a similar tool to allow users to perform certain functions with elevated privileges. This is a useful way to allow a user to perform specific system administration duties without granting the user full root access. Even for users who require full root access, `sudo` can be configured to allow a user to run all commands with root access, allowing the user to perform system administration from his or her own account instead of logging into the root account. This is useful for audit trail purposes.

If `sudo` is used, review the /etc/sudoers file to evaluate the ability of users to run commands as root (and other sensitive accounts) with the `sudo` command. The `sudo` tool can be used to grant specific users the ability to run specific commands as if they were root (or any other account for that matter). This is generally preferable to giving users full root access.

The basic format of an entry in the sudoers file would look something like this:

```
Andrew ALL=(root) /usr/bin/cat

Micah ALL=(ALL) ALL
```

In this example, user Andrew would be allowed to run the command `/usr/bin/cat` as the user root on all systems, and user Micah would be allowed to run any command as any user on any system. Many other options will not be covered here. Consult the man page for sudoers for more information.

If `sudo` or an equivalent tool is used, review processes for managing the sudoers file (or equivalent). This file can quickly become complex—with lots of lines, each granting access to specific elevated privileges to specific users—and outdated. It therefore requires management similar to what you expect to see with firewall rule sets and the like. Look for processes to review, validate, and clean up the entries periodically in this file. If your Unix environment is large (it consists of a large number of servers), it may be preferable to implement some form of centralized sudoers file that is referenced by all systems rather than attempting to maintain the file on each individual system.

Finally, it is important that you review the use of trusted access for root and other administrative accounts. See steps 29 and 30 for information on testing for trusted access.

11. Review controls for preventing direct root logins.

Because several people usually know the root password, if they are allowed to log in directly as the root account, no accountability exists for actions performed by that account. If inappropriate actions are performed by the root account, there will be no way to trace those actions back to a specific user. It is preferable to force people to log in as themselves first and then use `su` or `sudo` to access the root account.

How

Review the `wtmp` log (by performing the `more` command on /usr/adm/wtmp, /var/adm/wtmp, or /etc/wtmp, depending on the type of system) to verify that there are no direct root logins. The `last` command can be used to view the contents of this file on most systems. Exceptions would be direct logins from the console, which may be needed for emergencies.

Review settings for preventing direct root logins via `telnet` and `rlogin`.

- The file /etc/default/login can be used to disable direct root logins on Solaris machines. If this file is available, the `CONSOLE=` parameter should be set to the pathname of a nonexistent device. If the administrator wants

creating an audit trail).

9. Ensure that invalid shells have been placed on all disabled accounts.

This is only a significant risk if trusted access is allowed (see the "Network Security and Controls" section later in the chapter). If trusted access is allowed, a user with a certain username on one system (the trusted system) can log into an account with that same username on another system (the trusting system) without entering a password. This can be done as long as the user account on the trusting system has a valid shell assigned to it, even though the account may have been disabled. Therefore, if a system administrator disables an account but leaves it with a valid shell, a user on a remote, trusted system with the same username still could access that account.

How

View the contents of the password files (via the `more` command). If an account has been disabled, it will show an "*," "*LK*," or something similar in the password field (remember to look in the shadow password file if it is being used). For those accounts, review the contents of the shell field. If it contains anything other than `/dev/null, /bin/false`, or something similar, the account probably still can access a valid shell or program.

10. Review and evaluate access to superuser (root-level) accounts and other administration accounts.

An account with root-level access has the ability to do anything with the system, including deleting all files and shutting the system down. Access to this ability should be minimized. Other accounts may exist on the system for the purpose of administering specific applications and also should be tightly controlled to prevent system disruption.

How

Review the contents of the password files, and identify all accounts with a UID of `0`. Any account with a UID of `0` is treated by the system as if it were the root account. Question the need for any account besides root to have a UID of `0`. Determine via interviews who knows the passwords to the root and other UID `0` accounts, and evaluate the appropriateness of this list. Also, audit the process that is used by the system administrators to document and communicate the passwords for these accounts, as these will likely be shared between the members of the team. Preferably, these passwords will be stored in a password vault of some kind and require a controlled, auditable checkout feature in order to obtain the password.

Review the password file for the existence of other administration accounts (such as "oracle"). You likely will have identified potential candidates when performing step 8. Determine via interviews who knows the passwords to these accounts, and evaluate for appropriateness. Review the process for documenting and communicating these passwords as well.

Many environments use `sudo` or a similar tool to allow users to perform certain functions with elevated privileges. This is a useful way to allow a user to perform specific system administration duties without granting the user full root access. Even for users who require full root access, `sudo` can be configured to allow a user to run all commands with root access, allowing the user to perform system administration from his or her own account instead of logging into the root account. This is useful for audit trail purposes.

If `sudo` is used, review the /etc/sudoers file to evaluate the ability of users to run commands as root (and other sensitive accounts) with the `sudo` command. The `sudo` tool can be used to grant specific users the ability to run specific commands as if they were root (or any other account for that matter). This is generally preferable to giving users full root access.

The basic format of an entry in the sudoers file would look something like this:

```
Andrew ALL=(root) /usr/bin/cat

Micah ALL=(ALL) ALL
```

In this example, user Andrew would be allowed to run the command `/usr/bin/cat` as the user root on all systems, and user Micah would be allowed to run any command as any user on any system. Many other options will not be covered here. Consult the man page for sudoers for more information.

If `sudo` or an equivalent tool is used, review processes for managing the sudoers file (or equivalent). This file can quickly become complex—with lots of lines, each granting access to specific elevated privileges to specific users—and outdated. It therefore requires management similar to what you expect to see with firewall rule sets and the like. Look for processes to review, validate, and clean up the entries periodically in this file. If your Unix environment is large (it consists of a large number of servers), it may be preferable to implement some form of centralized sudoers file that is referenced by all systems rather than attempting to maintain the file on each individual system.

Finally, it is important that you review the use of trusted access for root and other administrative accounts. See steps 29 and 30 for information on testing for trusted access.

11. Review controls for preventing direct root logins.

Because several people usually know the root password, if they are allowed to log in directly as the root account, no accountability exists for actions performed by that account. If inappropriate actions are performed by the root account, there will be no way to trace those actions back to a specific user. It is preferable to force people to log in as themselves first and then use `su` or `sudo` to access the root account.

How

Review the `wtmp` log (by performing the `more` command on /usr/adm/wtmp, /var/adm/wtmp, or /etc/wtmp, depending on the type of system) to verify that there are no direct root logins. The `last` command can be used to view the contents of this file on most systems. Exceptions would be direct logins from the console, which may be needed for emergencies.

Review settings for preventing direct root logins via `telnet` and `rlogin`.

- The file /etc/default/login can be used to disable direct root logins on Solaris machines. If this file is available, the `CONSOLE=` parameter should be set to the pathname of a nonexistent device. If the administrator wants

to place the pathname of the actual console device (the terminal directly linked to the Unix machine) into this parameter, the console should be in a secure location. The contents of this file can be viewed by executing the `more /etc/default/login` command.

- On Linux and HP systems, the /etc/securetty file can be used to prevent direct logins as root. The contents of the file should contain all terminals that are allowed direct root login. The file should exist but be empty. Sometimes the system administrator will want to allow direct root login from the console terminal. This is acceptable, as long as the console is in a secure location. The contents of this file can be viewed by executing `more /etc/securetty`.

Review settings for preventing direct root logins via SSH. The /etc/sshd_config or /etc/openssh/sshd_config file is used for this purpose. Review the contents of this file using the `more` command. Look for the `PermitRootLogin` parameter. If this parameter is set to a value of `no`, root logins are not permitted. If the parameter is not there or is set to a value of `yes`, root logins are permitted.

Review settings for preventing direct root logins via FTP. This can be done by placing a root entry in the /etc/ftpusers file. Review the contents of this file using the `more` command.

12. Review and evaluate the use of groups and determine the restrictiveness of their usage.

This information will provide a foundation for evaluating file permissions in later steps. If all users are placed in one or two large groups, group file permissions are not very useful. For example, if all users are part of one large group, a file that allows group "write" permissions effectively allows world "write" permissions. However, if users are placed in selective, well-thought-out groups, group file permissions are effective controls.

How

Review the contents of the /etc/group, /etc/passwd, and related centralized files (such as NIS) using the `more` (such as `more /etc/passwd`) and, for NIS, `ypcat` (such as `ypcat passwd` and `ypcat group`) commands.

Look at the password as well as the group files to get an idea of group assignments, because user primary group assignments from the password file do not need to be relisted in the group file. In other words, if a user is assigned to the "users" group in the /etc/passwd file, there is no need to list him or her as a member of that group in the /etc/group file. Therefore, to obtain a full listing of all members of the "users" group, you must determine who was assigned to that group in the /etc/group file and determine who was assigned to that group in the /etc/passwd file (along with any NIS, LDAP, and so on equivalents being used in your environment). It is important to note that a group does not need to be listed in the group file in order to exist. It is therefore necessary to identify all group IDs (GIDs) in the password file and determine the membership of those groups. If you rely on the group file to identify all groups on the system, you may not receive a complete picture.

13. Evaluate the use of passwords at the group level.

Group-level passwords allow people to become members of groups with which they are not associated. If a group has a password associated with it in the group file, a user can use the `newgrp <group name>` command and will be prompted to enter that group's password. Once the password is entered correctly, the user will be given the rights and privileges of a member of that group for the duration of the session. There is generally little need for this functionality, because users are usually granted membership to whichever groups they need to access. Creating a group-level password creates another vector of attack on the system by creating the opportunity for users to hack the group-level passwords and escalate their privileges.

How

Review the contents of the group file(s) by using `more /etc/group` for the local file and `ypcat group` for NIS. If the groups have anything other than a common character (such as an "*" or even nothing) in the password field (the second field for each entry), passwords are being used. If group-level passwords are being used, speak to the system administrators to understand the purpose and value of using such passwords, and review the process for restricting knowledge of these passwords.

To look for passwords in /etc/group, you could use this command in your audit script:

```
awk -F: '{if($2!="" && $2!="x" && $2!="*")print "A password is
set for group
"$1" in /etc/group\n"}' /etc/group
```

14. Review and evaluate the security of directories in the default path used by the system administrator when adding new users. Evaluate the use of the "current directory" in the path.

A user's path contains a set of directories that are to be searched each time the user issues a command without typing the full pathname. For example, suppose the `ls` command on your system is located at /bin/ls. To execute this program and view the permissions in the /home directory, you could type `/bin/ls /home`. By typing in the exact location of the file, you are using the full pathname. However, we rarely do this. Instead, the norm is to type `ls /home`. In this case, the user's path is the mechanism for finding the file that is to be executed.

For example, let's say that your path looks like this:

```
/usr/bin:/usr/local/bin:/bin
```

This means that when you type in a command, the operating system will first look for a file by that name in /usr/bin. If the file doesn't exist there, it will next look in /usr/local/bin. If it still doesn't find a file by that name there, it will look in /bin. If it is still unsuccessful, the command will fail. Thus, in our example, we have attempted to execute the `ls` command, which is located in /bin. The system will first look for a file called `ls` in the /usr/bin directory. Since there is no file in that directory, it will look in the /usr/local/bin directory. Since the file is not there either, it will look in /bin. A file called `ls` is in /bin on our system, so the operating system will attempt to execute that file. If the permissions on that file grant you execute permissions, you will be allowed to run the program.

Attackers who can write to a directory in a user's path can perform filename

spoofing. For example, if the directory that contains the `ls` command is not secured, an attacker could replace the `ls` command with his or her own version. Alternatively, if the "current directory" (meaning whatever directory the user happens to be in at the time the command is executed) or another unprotected directory is placed early in the user's path, the attacker could place his or her own version of the `ls` command in one of these and never have to touch the real `ls` command.

Because of all this, directories in the path should be user- or system-owned and should not be writable by the group or world.

A "`.`" or an empty entry (a space) represents the "current directory," which means whatever directory the user happens to be in at the time he or she executes a command. Since this is an unknown, it is generally safer to leave this out of the path. Otherwise, an attacker could trick a user or administrator into switching to a specific directory and then executing a common command, a malicious version of which could be located in that directory.

Each user has the ability to set his or her path in his or her initialization files. However, most users will never touch their paths, and it is important for the system administrator to provide a default path that is secure.

How

The easiest way to view your own path is by typing `echo $PATH` at the command line. The default setting for users' paths may be found in /etc/default/login, /etc/profile, or one of the files in /etc/skel. Ask the system administrator where the default setting is kept if you are unsure. If the user has modified his or her path, this typically will be done in one of the dot-files in the home directory. Look at the contents of such files as .login, .profile, .cshrc, .bash_login, and so on. A quick way to look is to use the command `grep "PATH=" .*` in the user's home directory. A user's home directory can be determined by viewing his or her entry in the password file.

Once you know the name of the file that contains the path, view the contents of the file using the `more` command. The `ls –ld` command can then be performed on each directory in the path to view directory permissions. The directories should be writable only by the user and system accounts. Group and world write access should not be allowed (unless the group contains only system-level accounts).

15. Review and evaluate the security of directories in root's path. Evaluate the use of the "current directory" in the path.

If a user can write to a directory in root's path, it is possible that the user could perform filename spoofing and obtain access to the root account. See step 14 for further explanation of this concept.

How

Have the system administrator display root's path for you (using the `echo $PATH` command when logged in as root), and then review the permissions of each directory using the `ls -ld` command. All directories in root's path should be system-owned and should not be group or world writable (unless the group contains only system-level accounts such as `bin` and `sys`). The "current directory" generally should not be part of root's path.

The following will print the permissions of root's path (assuming that the script is executed as root) and warn if there is a "`.`" in the path or if one of the directories is world writable:

```
#!/bin/sh
for i in `echo $PATH | sed 's/:/ /g'`
do
if [ "$i" = . ]
then
echo -e "WARNING: PATH contains .\n"
else
ls -ld $i
ls -ld $i | awk '{if(substr($1,9,1)=="w")print "\nWARNING - " $i " in
root'\'s' path is world writable"}'
fi
done
```

16. Review and evaluate the security of user home directories and config files. They generally should be writable only by the owner.

User config files are basically any file located in the user's home directory that starts with a dot (`.`), commonly called *dot-files.* These files define the user's environment, and if a third party can modify them, privileged access to the account can be obtained. For example, when a user first logs in, commands within his or her .login, .profile, .bashrc, or other file (depending on the shell) are executed. If an attacker is able to modify one of these files, he or she can insert arbitrary commands, and the user will execute those commands at the next login. For example, commands could be executed that copy the user's shell to another file and make it Set UID (SUID) (a concept that will be explained in step 19). The attacker then would be able to execute this new file and "become" that user. Access to these files also allows the attacker to change the user's path or create malicious aliases for common commands by modifying these files. Other config files, such as .cshrc and .kshrc, are executed at login, when a new shell is run, or when someone uses the `su` command to switch to the user's account. The ability to insert arbitrary commands into these files results in a similar risk as with the .login and .profile files.

Another config file that should be locked down is the .rhosts file. This file provides trusted access (access without the use of a password) to the user's account from specific accounts on specific other systems. A person who can modify this file can gain trusted access to the user's account.

Even though specific risks were not mentioned for other dot-files, it is generally a good idea to keep them locked down. There is generally no legitimate reason that others should be modifying a user's config files.

Access to a user's home directory also should be locked down. If an attacker has write privileges to the directory, he or she will have the ability to delete any of the user's config files and replace them with his or her own versions.

How

The location of user home directories can be obtained from the account entries in the password file. The `ls –ld` command should be performed on each directory to view directory permissions. The `ls –al` command should be performed on each directory to view the permission on the files (including the config files) within the directory.

Permissions Management

17. Evaluate the file permissions for a judgmental sample of criti-

cal files and their related directories.

If critical files are not protected properly, the data within these files can be changed or deleted by inappropriate users. This can result in system disruption or unauthorized disclosure and alteration of proprietary information.

How

Using the `ls –l` command, examine the permissions on critical system files and their related directories. Generally, the most critical files within the Unix and Linux operating systems are contained in the following directories:

- /bin, /usr/bin, /sbin, /usr/sbin, and/or /usr/local/bin (programs that interpret commands and control such things as changing passwords)
- /etc (files that contain information such as passwords, group memberships, and trusted hosts and files that control the execution of various daemons)
- /usr or /var (contain various accounting logs)

For these directories and the files contained therein, question the need for write access to be granted to anyone other than system administration personnel.

In addition, other critical data files (such as files containing key application data and company-proprietary information) will likely be on the system you are auditing and should be secured. Interview the system administrator to help identify these.

For ease of use and to get a full picture of the file system, you can ask the system administrator to run the `ls -alR` command (recursive file listing) against the entire file system and place the results in a file for you. You can then view the contents of this file in performing this and other steps. The system administrator must do this because only the superuser can access the contents of all directories.

You might want to look for several variations that are short of a full `ls –alR`. If, for example, you want to find all world-writable files (excluding symbolic links, or symlinks), use `find / -perm -777 ! -type l –print`. Check the man pages to get more ideas on how you can use that command in your audit.

18. Look for open directories (directories with permission set to drwxrwxrwx) on the system and determine whether they should have the sticky bit set.

If a directory is open, anyone can delete files within the directory and replace them with their own files of the same name. This is sometimes appropriate for /tmp directories and other repositories for noncritical, transitory data; however, it is not advisable for most directories. By placing the sticky bit on the directory (setting permissions to `drwxrwxrwt`), only the owner of a file can delete it.

How

Examine directory permissions within the recursive file listing obtained from the preceding step, and search for open directories. (In the listing of `ls –alR`, note that the directory permissions will be listed next to the "`.`".) To find just directories with world-write permissions, you can use the command `find / -type d –perm -777`. For any such directories discovered, discuss the function of those directories with the system administrator and determine the appropriateness of the open permissions.

19. Evaluate the security of all SUID files on the system, especially those that are SUID to root.

SUID files allow users to execute them under the privileges of another UID. In other words, while that file is being executed, the operating system "pretends" that the user executing it has the privileges of the UID that owns the file. For example, every user needs the ability to update the password file to change passwords periodically. However, it would not be wise to set the file permissions of the password file to allow world-write access, because doing so would give every user the ability to add, change, and delete accounts. The `passwd` command was therefore created to give users the ability to update their passwords without having the ability to alter the rest of the password file. The `passwd` file is owned by root and has the SUID bit set (`rwsr-xr-x`), meaning that when users execute it, they do so using the privileges of root.

If an SUID file is writable by someone other than the owner, it may be possible for the owning account to be compromised. Other users could change the program being run to execute arbitrary commands under the file owner's UID. For example, a command could be inserted such that the owner's shell is copied to a file and made to be SUID. Then, when the attacker executed this copied shell, it would run as if it were the owner of the SUID file, allowing the attacker to execute any command using the privilege level of the captured account.

How

For Solaris and Linux, a full list of SUID files can be viewed by using the following command:

```
find / -perm -u+s
```

Note that the results of this command will not be complete unless it is run by someone with superuser access.

Review the file permissions for those programs, particularly for those that are SUID to root. They should be writable only by the owner.

Also question the need for any programs that are SUID to a user account. There should be little reason for one user to run a program as if he or she were another user. Most SUID programs are SUID to root or some other system or application account. If you see a program that is SUID to a user account, it is possible that this program is being used to capture that user's account.

20. Review and evaluate security over the kernel.

The kernel is the core of the operating system. If it can be altered or deleted, an attacker could destroy the entire system.

How

Use the `ls –l` command on the location of the kernel for the system you are auditing. It should be owned and writable only by the superuser. The kernel could be stored in a number of possible locations. Some common kernel names are /unix (AIX), /stand/vmunix (HP), /vmunix (Tru64), /kernel/genunix (Solaris), and /boot/vmlinuz (Linux). Ask the system administrator for the location of the kernel on the system you are auditing.

21. Ensure that all files have a legal owner in the /etc/passwd file.

Each time a file is created, it is assigned an owner. If that owning account is subsequently deleted, the UID of that account still will be listed as the owner of the file unless ownership is transferred to a valid account. If another account is created later with that same UID, the owner of that account will, by definition, be given ownership of those files.

For example, suppose that Grant (UID 226) creates the file /grant/file. UID 226 (Grant) is listed as the owner of this file. Grant is then fired, and his account is deleted. However, ownership of his file is not transferred. The operating system still considers UID 226 to be the owner of that file, even though that UID no longer maps to a user in the password file. A few months later, Kate is hired and is assigned UID 226. The system now considers Kate to be the owner of the file /grant/file, and she has full privileges over it. If /grant/file contains highly sensitive information, this could be a problem. To avoid this problem, before deleting an account, the system administrators should disposition all files owned by that account, either by deleting them or by transferring ownership.

How

Have the system administrator perform the `quot` command (which has to be run by the superuser). This command will show all file owners on the system. Review this list, and ensure that a username, not a UID, is shown for every entry. If a UID appears, it means that there is no entry in the password file for that UID, which means that the password file could not convert the UID into a username. If a user is added later to the password file with that UID, that user would have ownership of these files.

NOTE The `quot` command is not available on all versions of Unix and Linux. If this is the case, the output of a `ls -alR` command will need to be reviewed manually to see if any files list an invalid username as the owner.

22. Ensure that the `chown` command cannot be used by users to compromise user accounts.

The `chown` command allows users to transfer ownership of their files to someone else. If a user can transfer an SUID file to another user, he or she then will be able to execute that file and "become" the user. For example, if a user copies his or her shell, makes it SUID and world-executable, and then transfers ownership to root, then by executing that file, the user becomes root.

How

Many versions of Unix allow only the superuser to execute `chown`. Many others do not allow SUID bits to be transferred to another user. To determine whether these controls are in place on the machine you are auditing, perform the following in order:

1. Review the password file and determine where your shell is located (it probably will be something like /bin/csh or /usr/bin/sh).
2. Run the command `cp <shell file name> ~/myshell` to create a copy of your shell file in your home directory.
3. Run the command `chmod 4777 ~/myshell` to make your new shell file SUID and world- executable.
4. Choose another user from the password file to transfer ownership to, preferably a fellow auditor.
5. Run the command `chown <new owner name> ~/myshell`, which will attempt to transfer ownership of the file to another user.
6. Run the command `ls -l ~/myshell` to see whether you transferred ownership successfully and, if so, whether the SUID bit also transferred.
7. If the SUID bit transferred to another owner, execute the file by typing `/myshell`. This will execute the shell.
8. Run the command `whoami`. This should show that you are now the other user and have taken over his or her account.
9. If this happens, the system administrator will need to contact his or her vendor for a fix.

23. Obtain and evaluate the default umask value for the server.

The umask determines what permissions new files and directories will have by default. If the default umask is not set properly, users could inadvertently be giving group and/or world access to their files and directories. The default should be for files to be created securely. Privileges then can be loosened based on need and conscious decisions by the users (as opposed to their being unaware that their new files and directories are not secure).

How

The default may be set in /etc/profile or in one of the files in /etc/skel. However, the easiest test is often just to view the umask value for your own account because this usually will be a representation of the default value for all new users. This can be done using the `umask` command.

The umask basically subtracts privileges when files and directories are created using the modular format of file permissions and assuming that the default is for all files and directories to be created fully open (777 permissions). In other words, with a umask of 000, all new files and directories will be created with default permissions of 777 (777 minus 000), meaning full access for the owner, group, and world.

For example, if the umask is set to 027, it will result in the following default permissions for newly created files and directories:

Normal default	777
Minus the umask	027
Default permissions on this server	750

This provides full access to the owner, read and execute access to the group, and no access to the world.

At a minimum, the default system generally should be set to a value of 027 (group write and all world access removed) or 037 (group write/execute and all world access removed).

24. Examine the system's crontabs, especially root's crontab, for unusual or suspicious entries.

A *cron* executes a program at a preset time. It is basically the Unix or Linux system's

native way of letting you schedule jobs. The *crontab* (short for cron table) contains all the crons scheduled on the system. Crons can be used to create time bombs or to compromise the owning account. For example, if an attacker managed to compromise a user's account, he or she could set up a cron that would copy the user's shell nightly and make it SUID and then delete this copy of the shell 15 minutes later. The attacker then could regain access to the account daily during that time period, but security-monitoring tools would not detect it unless the tools happened to run in that 15-minute window. An example of a time bomb would be a case where a system administrator is fired or quits and schedules a cron that crashes the system to run six months later.

How

The crontabs should be located within the directory /usr/spool/cron/crontabs or /var/spool/cron/crontabs. By performing the `ls –l` command on this directory, you will be able to list the contents. Each account with a crontab will have its own file in this directory. The contents of these files can be viewed with the `more` command. This will allow you to see the commands that are being executed and the schedule for that execution. Based on file permissions, you may need the administrator to display the contents of the crontabs. Also, depending on the level of your Unix knowledge, you may need the administrator's help in interpreting the contents of the files.

25. Review the security of the files referenced within crontab entries, particularly root's crontab. Ensure that the entries refer to files that are owned by and writable only by the owner of the crontab and that those files are located in directories that are owned by and writable only by the owner of the crontab.

All crons are run as if the owner of the crontab is running them, regardless of the owner of the file that is being executed. If someone besides the owner of the crontab can write to a file being executed by the crontab, it is possible for an unauthorized user to gain access to those accounts by altering the program being executed to cause the crontab owner to execute arbitrary commands (such as copying the cron owner's shell and making it SUID). For example, if root's crontab has an entry that executes the file /home/barry/flash and that file is owned by "Barry," then "Barry" has the ability to add any command he wants to the flash file, causing root to execute that command the next time the cron is executed.

If a crontab is executing a file that is in a directory that is not secure, this would allow other users to delete the program being run and replace it with their own, again potentially resulting in the owner of the crontab executing arbitrary commands.

How

The contents of each user's crontab should be reviewed (see the preceding step for more information). The `ls -l` command should be performed on each file being executed in a crontab, and the `ls –ld` command should be executed for each of the directories containing those files.

26. Examine the system's scheduled atjobs for unusual or suspicious entries.

Atjobs are one-time jobs that are scheduled to run sometime in the future. They operate much like cron jobs (except that they are executed only once) and can be used to create time bombs or to compromise an account.

How

The atjobs should be located within the directory /usr/spool/cron/atjobs or /var/spool/cron/atjobs. By performing the `ls –l` command on this directory, you can list the contents. The contents of these files can be viewed with the `more` command. This will allow you to see the commands that are being executed and the schedule for that execution. Based on file permissions, you may need the administrator to display the contents of the atjobs. Also, depending on the level of your Unix knowledge, you may need the administrator's help in interpreting the contents of the files.

Network Security and Controls

27. Determine what network services are enabled on the system, and validate their necessity with the system administrator. For necessary services, review and evaluate procedures for assessing vulnerabilities associated with those services and keeping them patched.

Whenever remote access is allowed (that is, whenever a network service is enabled), it creates a new potential vector of attack, thereby increasing the risk of unauthorized entry into the system. Therefore, network services should be enabled only when there is a legitimate business need for them.

New security holes are discovered and communicated frequently to the Unix/Linux community (including potential attackers). If the system administrator is not aware of these alerts, and if he or she does not install security patches, well-known security holes could exist on the system, providing a vector for compromising the system.

NOTE This is one of the most critical steps you will perform. Unnecessary and unsecured network services are the number-one vector of attack on *nix servers. They will allow someone who has no business being on the system either to gain access to the system or to disrupt the system.

How

Use the `netstat –an` or `ss -lp` command, and look for lines containing `LISTEN` or `LISTENING`. These are the TCP and UDP ports on which the host is available for incoming connections. If the `LSOF` (list open files) command is present on the system (more common on Linux), then `lsof -i` can be used.

Once you have obtained a list of enabled services, talk through the list with the system administrator to understand the need for each service. Many services are enabled by default and therefore were not enabled consciously by the system administrator. For any services that are not needed, encourage the administrator to disable them.

Understand the process used to keep abreast of new vulnerabilities for enabled

services and to receive and apply patches for removing those vulnerabilities. Common sources for vulnerability announcements include vendor notifications and Computer Emergency Response Team (CERT) notices. CERT covers the high-profile vulnerabilities, but you really should be getting notifications from your OS and add-on software vendors to ensure adequate coverage. Information on this process can be gathered via interviews and review of documentation.

If you need to validate a specific patch or package version, you can view installed packages and patches via the following commands:

- **Solaris** `showrev -p` will list the patches that have been applied; these can be cross-referenced with the patches listed in the security advisory from Sun.
- **Linux** `rpm -q -a` (Red Hat or other distributions using RPM) or `dpkg --list` (Debian and related distributions) will show the versions of installed packages.

Note that software can be installed outside the package-management system provided by the vendor, in which case these commands won't show you the requisite information. If you need to find the version of an executable, try running the command with the `-v` switch. In most cases, this will show you version information that you can compare with information in vulnerability notices.

A network scan of existing vulnerabilities also can be used to help validate the effectiveness of the patching process. See the next step for further details.

Consider the configuration of the services, not just whether they are allowed. The proper configuration of certain services such as Network File System (NFS), anonymous File Transfer Protocol (FTP), and those that allow trusted access are discussed later in this chapter. Space restrictions prevent us from detailing the proper configuration of every potential service (plus new vulnerabilities are discovered all the time). This is why the use of a network scanning tool is a critical component of an effective audit. Such a tool will keep up with and test for the latest vulnerabilities for you.

28. Execute a network vulnerability scanning tool to check for current vulnerabilities in the environment.

This will provide a snapshot of the current security level of the system (from a network services standpoint). The world of network vulnerabilities is an ever-changing one, and it is unrealistic to create a static audit program that will provide an up-to-date portrait of vulnerabilities that should be checked. Therefore, a scanning tool that is updated frequently is the most realistic mechanism for understanding the current security state of the machine. In addition, if the system administrator has a security-patching process in place, this scan will provide validation as to the effectiveness of that process (or as to whether it is really being executed).

How

See the "Tools and Technology" section later in this chapter for information on potential network vulnerability scanning tools. Even though many of these tools are designed to be nondisruptive and do not require access to the system, you should always inform the appropriate IT personnel (such as the system administrator, the network team, and information security) that you plan to run the tool and then get their approval and schedule with them a time to execute the tool. Scanning tools can interact in an unexpected fashion with a port and cause a disruption, so it is important that others be aware of your activities. These tools should usually be run in a "safe" (nondisruptive) mode such that they do not attempt to exploit any vulnerabilities discovered. On rare occasions, you will want to run an actual exploit to get more accurate results, but this should be done only with buy-in from and coordination with the system owner and administrator.

29. Review and evaluate the use of trusted access via the /etc/hosts.equiv file and user .rhosts files. Ensure that trusted access is not used or, if deemed to be absolutely necessary, is restricted to the extent possible.

Trusted access allows users to access the system remotely without the use of a password. Specifically, the /etc/hosts.equiv file creates trust relationships with specific machines, whereas the .rhosts file creates trust relationships with specific users on specific machines.

For example, if system "Trusting" has an /etc/hosts.equiv file that lists machine "Trusted" as a trusted host, then any user with an account using the same username on both systems will be able to access "Trusting" (the trusting machine) from "Trusted" (the trusted machine) without the use of a password. Thus, if the username "Hal" exists on both machines, the owner of the "Hal" account on "Trusted" will be able to access the "Hal" account on "Trusting" without using a password. Keep in mind that the key is the account name. If John Jones has an account on both machines but one has the account name "jjones" and the other has the account name "jjonzz," then the trust relationship won't work. The operating system won't acknowledge them as the same account.

The .rhosts files work similarly except that they are specific to a user. Each user can have a .rhosts file in his or her home directory that provides trusted access to his or her account. If username "Barry" on system "Trusting" has a .rhosts file in his home directory and that .rhosts file lists system "Trusted," then the "Barry" account on "Trusted" will be able to access the "Barry" account on "Trusting" without using a password. Alternatively, system and username pairs can be listed in the .rhosts file. The .rhosts file for "Barry" on "Trusting" could list username "Wally" on system "Trusted." This would mean that the "Wally" account on "Trusted" would be able to access the "Barry" account on "Trusting" without using a password.

If the system you are auditing has trust relationships with other machines, the security of the trusting system depends on the security of the trusted system. If the accounts that are trusted are compromised, then, by definition, the accounts on the system you are auditing will be compromised as well. This is the case because access to the trusted machine provides access to the trusting machine. It is best to avoid this sort of dependency if at all possible.

NOTE If NIS is used, it is also possible to grant trusted access to specific netgroups (groups of usernames).

Trusted access can also be used to bypass controls over shared accounts. As discussed in other steps, shared accounts can be locked down such that `su` or `sudo` is required for access. However, if a user has access to a shared account via one of

these mechanisms and then creates a .rhosts file for that account granting trusted access to his or her personal account, the user will be able to bypass the need to use `su` or `sudo` to access the account.

The first option should be to eliminate trusted access. If it becomes obvious to the auditor that this is not feasible in the environment, the steps in the "How" section that follows can be used to mitigate the risk.

NOTE Trusted access works via the usage of the Berkeley `r` commands (for example, `rlogin`, `rsh`, and `rexec`). These commands are designed to look for trusted relationships automatically via .rhosts and /etc/hosts.equiv files when executed. If a trusted relationship doesn't exist, these commands will require the entry of a password. If trusted relationships do exist, these commands will not require the entry of a password.

How

Examine the contents of the /etc/hosts.equiv file and any .rhosts files on the system. The contents of the /etc/hosts.equiv file can be viewed by using the `more /etc/hosts.equiv` command. To find .rhosts files, you will need to view the contents of each user's home directory via the `ls –l` command (the location of user home directories can be found in the password file) to see whether a .rhosts file exists. The contents of any .rhosts files found can be viewed by using the `more` command. If file permissions restrict you from viewing the contents of these files, you will need to have the system administrator perform these commands for you.

Discuss the contents of these files with the system administrator to understand the business need for each entry. Encourage the administrator to delete any unnecessary entries or preferably to eliminate the use of trusted access altogether. For essential trusted relationships, discuss the possibility of using trusted Secure Shell (SSH) keys, which is generally a preferred alternative to hosts.equiv and .rhosts (see the next step for more details).

Ensure that none of the files contain the + sign. This symbol defines all the systems on the network as trusted and enables them all to log on without using a password (if there is an equivalent username on the trusting server). If the + sign exists in the /etc/hosts.equiv file, then any user (except root) on any system on the network who has the same username as any of the accounts on the trusting system will be able to access the account without using a password. If the + sign exists in a .rhosts file, any user on any system on the network who has the same username as the owner of the .rhosts file will be able to access the account without using a password. This includes the root account, so a .rhosts file with a + in root's home directory is usually a particularly bad idea.

For any legitimate and necessary trust relationships, determine whether the administrator is comfortable in knowing that each system to which trusted access is given is as secure as the system being audited. As mentioned earlier, the system's security depends on the security of any system being trusted. System administrators generally should not give trusted access to systems they do not control. If they do, they should take steps to obtain assurance as to the security and integrity of the systems being trusted either by performing their own security scans or by conducting interviews with the system administrator of the trusted system.

If trusted hosts are needed in the /etc/hosts.equiv file, ensure that trusted users are not specified in this file. In some versions of Unix, a trusted user specified in this file will be allowed to log into the system as any username (except root) without entering a password.

If trusted access is allowed, usernames in the password files must be consistent across each system involved in the trusted relationship. Determine whether this is the case. If system2 trusts system1, then username "Bob" on system1 can log in as username "Bob" on system2 without entering a password. If "Bob" on system1 is Bob Feller, while "Bob" on system2 is Bobby Thomson, then Bobby Thomson's account now has been compromised.

Ensure that the /etc/hosts.equiv and .rhosts files are secured properly (using the `ls –l` command). The /etc/hosts.equiv file should be owned by a system account (such as root) and writable only by that account. If others can write to this file, they could list unauthorized machines in the trusted hosts list. The .rhosts files should be owned by the account in whose home directory they sit and should be writable only by that account. If a user can write to another user's .rhosts file, that user could make himself or herself, or someone else, trusted to log into that user's account from another machine.

Ensure that entries use the fully qualified domain name for systems being trusted (such as "rangers.mlb.com" instead of just "rangers"). An entry that does not use the fully qualified domain name could be spoofed by a machine with the same host name but a different domain.

Review processes used by the system administrators to detect and review any new trusted access established on the system. They should detect and review any new .rhosts files or entries and any new /etc/hosts.equiv entries.

30. Review and evaluate the usage of trusted access via SSH keys.

Trusted access via SSH keys is conceptually the same as trusted access via .rhosts files discussed in the preceding step, and is generally preferred if trusted access is required. It lets users access the system remotely via SSH without the use of a password, creating trust relationships with specific users on specific machines.

To establish a trust relationship via SSH keys, a user creates (or more likely uses an SSH key generation command to create) a subdirectory in his or her home directory on the trusted machine called .ssh and places two files within that directory: *id_rsa* is the private key and *id_rsa.pub* is the public key (if DSA encryption is being used instead of RSA, replace *rsa* with *dsa* in those filenames). The user then places the text from the public key file into a file called authorized_keys2 in the .ssh subdirectory of the home directory on the machine that the user wants to access (which becomes the trusting machine). Once this is done, the user will be able to access the trusting machine (the machine on which he or she created an authorized_keys2 file in his or her home directory) from the trusted machine (the machine containing the user's public and private key files) via SSH without the use of a password.

NOTE These default filenames (authorized_keys2, id_rsa, id_rsa.pub) can vary depending on the version of SSH being used and can even be changed by the user in some versions of SSH. Although this step is written using these standard filenames, talk with your administrator to understand the specifics for the environment you're auditing.

If the system you are auditing has trust relationships with other machines, the security of the trusting system depends on the security of the trusted system. If the accounts that are trusted are compromised, then, by definition, the accounts on the system you are auditing will be compromised as well. This is the case because access to the trusted machine provides access to the trusting machine. It is best to avoid this sort of dependency if at all possible.

Trusted access can also be used to bypass controls over shared accounts. As discussed in other steps, shared accounts can be locked down such that `su` or `sudo` is required for access. However, if a user has access to a shared account via one of these mechanisms and then places his or her personal public key in the shared account's authorized_keys2 file, the user will then be able to bypass the need to use `su` or `sudo` to access the account.

The first option should be to eliminate trusted access. If it becomes obvious to the auditor that this is not feasible in the environment, the steps in the "How" section next can be used to mitigate the risk.

How

Examine the contents of any authorized_keys2 files on the system. To find these files, you will need to view the contents of each user's home directory's .ssh subdirectory via the `ls –l` command (the location of user home directories can be found in the password file) in order to see whether an authorized_keys2 file exists. The contents of any authorized_keys2 files found can be viewed by using the `more` command. File permissions should restrict you from viewing the contents of these files, so you will likely need to have the system administrator perform these commands for you.

Discuss the contents of these files with the system administrator to understand the business need for each entry. Encourage the administrator to delete any unnecessary entries or preferably to eliminate the use of trusted access altogether.

For any legitimate and necessary trust relationships, determine whether the administrator is comfortable in knowing that each system to which trusted access is given is as secure as the system being audited. As mentioned earlier, the system's security depends on the security of any system being trusted. System administrators generally should not give trusted access to systems they do not control. If they do, they should take steps to obtain assurance as to the security and integrity of the systems being trusted either by performing their own security scans or by conducting interviews with the system administrator of the trusted system.

Ensure that the authorized_keys2 files and related .ssh subdirectories are secured properly (using the `ls –l` command). They should be owned by the account in whose home directory they reside and should be writable only by that account. If a user can write to another user's authorized_keys2 file, the user could set up additional trust relationships for the other user's account. For many versions of Unix, trusted access via SSH keys will not work unless permissions on these files and directories are set to 600.

Ensure that all id_rsa files on the system and related .ssh subdirectories are secured properly (using the `ls –l` command). The files should be owned by the account in whose home directory they reside and should be readable and writable only by that account. If a user can read another user's private key file, that user could use that information to spoof the other user and access trust relationships that user has established with other servers.

Passphrases can also be used to restrict further what activities can be performed using this form of trusted access. Talk with your administrator to determine whether passphrases are being used and to what extent. If they are being used, it will be important that you review the strength of and controls over those passphrases.

Review processes used by the system administrators to detect and review any new trusted access established on the system. They should detect and review any new authorized_keys2 files or entries.

Additional controls can be established over this function, depending on the version of SSH being used, such as disallowing key-based authentication or requiring any keys that are to be used for user authentication to be stored in a centralized location (instead of in various accounts' home directories). Talk with the administrator and perform research to determine what features are available and have been enabled in your environment.

31. If anonymous FTP is enabled and genuinely needed, ensure that it is locked down properly.

Anonymous FTP allows any user on the network to get files from or send files to restricted directories. It does not require the use of a password, so it should be controlled properly.

How

To determine whether anonymous FTP is enabled, examine the contents of the password file(s). If you see an "ftp" account in the password file and the FTP service is enabled, then anonymous FTP is available on the system. Once an anonymous FTP user has logged in, he or she is restricted only to those files and directories within the ftp account's home directory, which is specified in ftp's password entry (we'll assume that the home directory is at /ftp for this step). The ftp account should be disabled in the password file and should not have a valid shell.

Ensure that the FTP directory (/ftp) is owned and writable only by root and not by ftp. When using anonymous FTP, the user becomes user ftp. If ftp owns its own files and directories, anyone using anonymous FTP could alter the file permissions of anything owned by ftp. This can be determined by performing the `ls -l` command on the ftp home directory. Ftp should own only the /ftp/pub directory.

Examine the permissions of the /ftp directory and the subdirectories (by using the `ls -l` command).

- The /ftp/pub directory should have the sticky bit set so that people cannot delete files in the directory.
- The /ftp directory and its other subdirectories should be set with permissions at least as restrictive as `dr-xr-xr-x` so that users can't delete and replace files within the directories.

Ensure that the /ftp/etc/passwd file contains no user entries (just ftp) or passwords (by performing the `more` command on the file). Otherwise, anyone on the network can see usernames on the server and use those for attacking the system. It should not allow group or world write permissions (`ls -l /ftp/etc/passwd`).

Other files outside of the /ftp/pub directory should not allow group or world write access (verify by using the `ls -l` command).

Attackers could transfer large files to the /ftp directories and fill up the file

system (for example, to commit a denial-of-service attack and/or prevent audit logs from being written). The system administrator should consider placing a file quota on the ftp user or placing the /ftp home directory on a separate file system.

32. If NFS is enabled and genuinely needed, ensure that it is secured properly.

NFS allows different computers to share files over the network. Basically, it allows directories that are physically located on one system (the NFS server) to be mounted by another machine (the NFS client) as if they were part of the client's file structure. If the directories are not exported in a secure manner, the integrity and availability of that data can be exposed to unnecessary risks.

How

NFS use can be verified by examining the /etc/exports file or the /etc/dfs/dfstab file (using the `more` command). If this file shows that file systems are being exported, then NFS is enabled.

Because NFS authorizes users based on UID, UIDs on all NFS clients must be consistent. If, for example, Cathy's account is UID 111 on the system being audited but Bruce's account is UID 111 on an NFS client, then Bruce will have Cathy's access level for any files that are exported (because the operating system will consider them to be the same user). After determining which systems can mount critical directories from the system you're auditing, you will need to work with the system administrator to determine how UIDs are kept consistent on those systems. This may involve obtaining a copy of each system's password file and comparing UIDs that appear in both the NFS server and an NFS client. Note that the same risk exists and should be investigated for GIDs.

Review the /etc/exports file or the /etc/dfs/dfstab file (using the `more` command):

- Ask the system administrator to explain the need for each file system to be exported.
- Ensure that the `access=` option is used on each file system being exported. Otherwise, any machine on the network will be able to access the exported file system. This option should be used to specify the hosts or netgroups that are allowed to access the file system.
- Ensure that read-only access is given where possible using the `ro` option (note that read/write is the default access given if read-only is not specified).
- Ensure that root access is not being given to NFS clients (that is, the `root=` option is not being used) unless absolutely necessary and unless the NFS clients have the same system administrator as the server. The `root=` option allows remote superuser access for specified hosts.
- Ensure that root accounts logging in from NFS clients are not allowed root access. You should not see `anon=0`, which would allow all NFS clients superuser access.

Review the contents of the /etc/fstab or the /etc/vfstab (or /etc/checklist for HP systems) file (using the `more` command) to see if the system you are auditing is importing any files via NFS. If it is, ensure that the files are being imported "nosuid." If SUID files are allowed, the NFS client could import a file that is owned by root and has permissions set to `rwsr-xr-x`. Then, when a user on the NFS client runs this program, it will be run as that client's superuser. The root user on the NFS server could have inserted malicious commands into the program, such as a command that creates a .rhosts file in the client root user's home directory. This .rhosts file then could be used by the NFS server to obtain unauthorized superuser access to the NFS client. Note that if the system administrator is the same on both the NFS client and the NFS server, this is not a big risk.

On all these NFS steps, the auditor should use good judgment. The criticality of the files being exported should influence the scrutiny with which the auditor reviews them.

33. Review for the use of secure protocols.

Certain protocols (such as Telnet, FTP, remote shell [rsh], rlogin, and remote copy [rcp]) transmit all information in cleartext, including UID and password. This could allow someone to obtain this information by eavesdropping on the network.

How

Review the list of services that are enabled and determine whether `telnet`, `ftp`, and/or the `r` commands are enabled. If so, via interviews with the system administrator, determine the possibility of disabling them and replacing them with secure (encrypted) alternatives. Telnet, rsh, and rlogin can be replaced by SSH; FTP can be replaced by Secure File Transfer Protocol (SFTP) or Secure Copy Protocol (SCP); and rcp can be replaced by SCP.

NOTE The use of secure protocols is particularly important in a DMZ and other high-risk environments. However, it is still advisable to use secure protocols even on internal networks to minimize attacks from within and the ability for successful external attacks to escalate.

34. Review and evaluate the use of .netrc files.

The .netrc files are used to automate logons. If a confidential password is placed in one of these files, the password may be exposed to other users on the system.

How

The following command can be used to find and print the contents of all .netrc files on the system. You likely will need to have the system administrator run this command to search the entire system.

```
find / -name '.netrc' -print -exec more {} \;
```

For any .netrc files found, review the file contents. If read access is restricted, you will need the system administrator to do this for you. Look for indications of passwords being placed in these files. If you find them, review file permissions via the `ls –l` command, and ensure that no one besides the owner can "read" the file. Even if file permissions are locked down, anyone with superuser authority will be able to read the file, so it's better to avoid using these files at all. However, if they exist and are absolutely necessary, the auditor should ensure that they have

been secured to the extent possible.

35. Ensure that a legal warning banner is displayed when a user connects to the system.

A *legal logon notice* is a warning displayed whenever someone attempts to connect to a system. This warning should be displayed prior to actual login and basically should say, "You're not allowed to use this system unless you've been authorized to do so." Verbiage of this sort may assist your ability to prosecute attackers in court. The warning banner can also be used to inform authorized users that their activity on the system will be monitored.

How

Log in to your account using each available mechanism that provides shell access, such as Telnet and SSH. Determine whether a warning banner is displayed. The text for this banner frequently is located in files such as /etc/issue and /etc/sshd_config (or /etc/openssh/sshd_config). Via interviews with the system administrator, determine whether the verbiage for this warning banner has been developed in conjunction with the company's legal department.

36. Review and evaluate the use of modems on the server.

Modems bypass corporate perimeter security (such as firewalls) and allow direct access to the machine from outside the network. They present significant risk to the security of the machine on which they reside and may allow the modem user to "break out" of the machine being audited and access the rest of the network. Allowing dial-in modems to be placed on a production machine is usually a bad idea. It is almost always preferable to have access to a machine channeled through standard corporate external access mechanisms such as a virtual private network (VPN) or Remote Access Services (RAS).

How

Although modems are a dated technology, it is still possible for them to be used and it is therefore still worth checking. Unfortunately, there is no reliable method of determining whether a modem is connected to a machine outside of physical inspection. If physical inspection is not practical, the next best option is to interview the system administrator to understand whether modems are used. If they are used, alternative mechanisms for allowing external access to the machine should be investigated. If a dial-in modem is deemed truly necessary, consider implementing compensating controls such as dial-back to trusted numbers (that is, when a call is received, the machine hangs up and dials back to a trusted number) and authentication.

Security Monitoring and Other General Controls

37. Review the su and sudo command logs to ensure that when these commands are used, they are logged with the date, time, and user who typed the command.

The `su` command is a tool used frequently by attackers to try to break into a user's account. The `sudo` command allows authorized users to perform specific commands as if they were root (or another privileged account). The use of both commands should be logged to ensure accountability and to aid in investigations.

How

Attempt to perform a `more` command on the `su` log. However, the log may be protected, so you may not be able to do this. If this is the case, have the system administrator provide you with a copy of the log and perform the command on it. For some systems, the `su` log will be at /usr/adm/sulog, /var/adm/sulog, or /var/log/auth.log. For other systems, the /etc/default/su file will determine where the `su` log will be kept.

- Ensure that this file exists and is capturing information on `su` usage (such as who performed the command, to what account they switched, the date and time of the command, and indications as to whether or not the command succeeded).
- Question any instance of one user `su`-ing to another user's account. There should be little to no reason for one user to attempt to `su` to another user's account on the system. Most `su` commands should be issued from an administrator's account to root or from a user account to an application ID.

View the `sudo` log to ensure that it is capturing information on `sudo` usage (such as who performed the command, what command was performed, and the date and time of the command). By default, the `sudo` logs are written to the syslog, but this can be changed in /etc/sudoers, so check for the location on your system (using the `more` command).

38. Evaluate the syslog to ensure that adequate information is being captured.

If system audit logs are not kept, there will be no record of system problems or user activity and no way to track and investigate inappropriate activities.

How

View the contents of the /etc/syslog.conf file using the `more` command. The /etc/syslog.conf file determines where each message type is routed (to a filename, to a console, and/or to a user). At a minimum, `crit` and `err` messages related to `auth` (authorization systems—programs that ask for usernames and passwords), `daemon` (system daemons), and `cron` (cron daemon) probably should be captured, along with `emerg` and `alert` messages.

Each syslog message contains, in addition to the program name generating the message and the message text, the facility and priority of the message.

Following are some of the common potential syslog facilities (that is, the type of system function):

- `kern` Kernel
- `user` Normal user processes
- `mail` Mail system
- `lpr` Line printer system
- `auth` Authorization systems (programs that ask for usernames and passwords)
- `daemon` System daemons

- `cron` Cron daemon

Following are the potential priority levels that indicate the severity of the message:

- `emerg` Emergency condition (such as an imminent system crash)
- `alert` Immediate action needed
- `crit` Critical error
- `err` Normal error
- `warning` Warning
- `notice` Not an error but special handling needed
- `info` Informational message
- `debug` Used when debugging programs

Notice that these are listed in descending order—most critical to least critical. When specifying a logging level, the level encompasses that level and higher, so logging at the debug (lowest) level, for example, also would log all other levels. An asterisk for the facility or level indicates that all facilities or levels are logged.

On HP systems, the /etc/btmp file contains invalid login attempts. Determine whether this file exists. If not, it should be created. On Solaris, the file /var/adm/loginlog will log any time a user tries to log into the system but types a bad password five times in a row (this is the default—the number can be configured in the /etc/default/login file). If this file does not exist, it should be created.

39. Evaluate the security and retention of the wtmp log, sulog, syslog, and any other relevant audit logs.

If the audit logs are not secure, then unauthorized users could change their contents, thus damaging the logs' usefulness during investigations. If logs are not retained for an adequate period, the administrator may be unable to investigate inappropriate activities and other system issues if needed.

How

The locations of the log files are discussed in previous steps in this section. Perform an `ls -l` command on those files. They usually should be writable only by root or some other system account.

Interview the system administrator to determine retention, which could be either online or offline. It is generally preferable to retain these security logs for at least three to six months to allow for adequate history during investigations.

40. Evaluate security over the utmp file.

The utmp log keeps track of who is currently logged into the system and includes information regarding terminals from which users are logged in. By changing the terminal name in this file to that of a sensitive file, an attacker can get system programs that write to user terminals to overwrite the target file. This would cause this sensitive file to be corrupted.

How

Perform an `ls -l` command on the utmp file, which is usually located at /etc/utmp on Unix systems and at /var/run/utmp on Linux systems. The file should be owned by root or another system account and should allow only owner write.

41. Review and evaluate system administrator procedures for monitoring the state of security on the system.

If the system administrator does not have processes for performing security monitoring, security holes could exist, and security incidents could occur without his or her knowledge.

How

Interview the system administrator, and review any relevant documentation to get an understanding of security monitoring practices. Numerous levels and methods of security monitoring can be performed. Although all monitoring levels and methods do not need to be performed, you should see some level of monitoring, which should be consistent with the criticality of the system and the inherent risk of the environment. (For example, a web server in the DMZ should have more robust security monitoring than a print server on the internal network.) Basically, you want to know how the system administrator is monitoring for problems, such as what you've been auditing for throughout the other audit steps in this chapter.

Following are four primary levels of monitoring. Potential tools for performing these types of monitoring are discussed in the "Tools and Technology" section later in this chapter.

- **Network vulnerability scanning** This is probably the most important type of security monitoring in most environments. It monitors for potential vulnerabilities that could allow someone who has no business being on the system either to gain access to the system or disrupt the system. Since these vulnerabilities can be exploited by anyone on the network, you need to be aware of them and close them down.
- **Host-based vulnerability scanning** This is scanning for vulnerabilities that would allow someone who's already on the system to escalate his or her privileges (such as exploit the root account), obtain inappropriate access to sensitive data (due to poorly set file permissions, for example), or disrupt the system. This type of scanning generally is more important on systems with many nonadministrative end users.
- **Intrusion detection** This monitoring detects unauthorized entry (or attempts at unauthorized entry) into the system. Baseline monitoring tools (such as OSSEC or Tripwire) can be used to detect changes to critical files, and log-monitoring tools can be used to detect suspicious activities via the system logs.
- **Intrusion prevention** This type of monitoring detects an attempted attack and stops the attack before it compromises the system. Examples include host intrusion prevention system (IPS) tools and network-based IPS tools such as Snort or Suricata.

If security monitoring is being performed, assess the frequency of the monitoring and the quality with which it is performed. Look for evidence that the security monitoring tools actually are used and acted on. Review recent results, and determine whether they were investigated and acted on. Leverage the results of the rest of the audit in performing this assessment. For example, if you found significant issues in an area that is supposedly being monitored, it might lead to

questions as to the effectiveness of that monitoring.

42. If you are auditing a larger Unix/Linux environment (as opposed to one or two isolated systems), determine whether a standard build exists for new systems and whether that baseline has adequate security settings. Consider auditing a system freshly created from the baseline.

One of the best ways to propagate security throughout an environment is to ensure that new systems are built right. In this way, as new systems are deployed, you have confidence that they initially have the appropriate level of security.

How

Through interviews with the system administrator, determine the methodology used for building and deploying new systems. If a standard build is used, audit a newly created system using the steps in this chapter.

43. Perform steps from Chapter 5 as they pertain to the system you are auditing.

In addition to auditing the logical security of the system, you should ensure that appropriate physical controls and operations are in place to provide for system protection and availability.

How

Reference the steps from Chapter 5, and perform those that are relevant to the system being audited. For example, the following topics are likely to be pertinent:

- Physical security
- Environmental controls
- Capacity planning
- Change management
- System monitoring
- Backup processes
- Disaster recovery planning

Tools and Technology

The open-source community has provided numerous valuable tools that an auditor can take advantage of to increase both the accuracy and efficiency of his or her work. Some of the most commonly used open-source tools for auditing *nix systems are listed here with a few tips on their use.

Network Vulnerability Scanners

In past editions of this book, we have recommended the Nessus network vulnerability scanner, which was written by Renaud Deraison. It first appeared in 1998 and was arguably the most advanced and most popular open-source network vulnerability assessment tool. In a nutshell, Nessus operates by looking for open ports on the target host, trying to identify the services running on those ports, and then testing those services for specific vulnerabilities. The server operates on Unix/Linux only, but clients to control the server are also available for Windows. Beginning with version 3.0, Nessus is now closed source and is owned by Tenable Security. However, OpenVAS, an open-source network vulnerability scanner, can be used as an alternative and uses the last open-source version of Nessus as a base.

For more information, see www.openvas.org and www.tenable.com.

NMAP

NMAP can be a handy way to check for open ports on a server without running an all-out vulnerability scanner such as Nessus, perhaps to test the rules of a host-based firewall. NMAP affords the user many options, and the man page is a must-read to understand them all.

For more information, see https://nmap.org.

Malware Detection Tools

Malware detection tools can be run in the course of an audit to check for possible compromises and can be suggested to the system administrator as tool(s) that could be run on a regular basis for security monitoring. Multiple open-source tools are available.

Chkrootkit and Rootkit Hunter are designed to identify both known rootkits running on a system and "suspicious" files or processes. For more information, see chkrootkit.org and rkhunter.sourceforge.net.

ClamAV and Linux Malware Detect (LMD) provide more traditional antivirus protection and are designed to detect malware on Linux systems. See www.clamav.net and www.rfxn.com/projects/linux-malware-detect for more information.

Tools for Validating Password Strength

If checking the strength of user-chosen passwords is part of your audit scope, you'll want to take a look at these three tools. Alec Muffett's Crack dates back to the early 1990s and is still widely known, although it is an older tool and might not support newer password-hashing algorithms. John the Ripper, however, generally is faster and more full-featured. And hashcat, which had a proprietary code base up until 2015, proclaims itself to be the world's fastest and most advanced password recovery tool. Any of these three options probably will get the job done in most cases. Consider adding wordlists to the dictionaries of these tools, including non-English wordlists, to enhance their efficacy.

For more information, see ftp.cerias.purdue.edu/pub/tools/unix/pwdutils/crack, www.openwall.com/john, and hashcat.net/hashcat/.

Host-Based Vulnerability Scanners

Past editions of this book have recommended Tiger and TARA as host-based vulnerability scanning tools that can automate performance of a number of the test steps in this chapter, allowing you to avoid tedious manual execution and analysis of each step. Tiger is a security tool originally developed at Texas A&M University. TARA (Tiger Analytical Research Assistant) is a variant of Tiger. Unfortunately, development on both tools has stopped in recent years, although both

are still available and functional if you are looking for an open-source host-based vulnerability scanner to supplement your auditing. For more information, see savannah.nongnu.org/projects/tiger, www.nongnu.org/tiger, and www.arc.com/tara.

For a more current open-source host-based vulnerability scanner, consider investigating Lynis, which has been available since 2007 and has a similar focus as Tiger, at https://cisofy.com/lynis.

Shell/Awk/etc

Although not a tool in the same sense as the others, the *nix shell can prove valuable, especially with the help of additional tools such as `awk` or `sed`, which can chop up and process text output from commands. Much of the required information to perform the steps in this audit program could be obtained by the use of a shell script. This script can be provided to the system administrator, who would run it as root, providing the output to the auditor. Using logical operations to test the values returned can even automate the evaluation process, returning a simple pass/fail grade for some of the steps. A simple example is found in step 2, where the passwd file is checked for duplicate UIDs.

Knowledge Base

If you're interested in learning more about the subject of auditing *nix operating systems, many resources are available in print and on the Internet.

One of the "go-to" books on Unix security is *Practical UNIX & Internet Security* by Simson Garfinkel, Gene Spafford, and Alan Schwartz, published by O'Reilly Media. This book provides an excellent overview of the topic, along with detailed guidance on how to secure the Unix environment.

Another excellent print resource is *Essential System Administration* by Æleen Frisch, published by O'Reilly Media. This book is written for *nix administrators but also can serve as an excellent guide for auditors who are looking for details on how to implement many of the concepts discussed in this chapter.

Many websites are devoted to Unix; the problem is wading through them to determine which ones can be most useful. Following are some to consider:

Website	Description
https://www.isaca.org	Standards and security guidance
https://www.sans.org/reading-room	Certifications and other documents from SANS
https://www.cisecurity.org/controls	Center for Internet Security "top 20" controls
https://apps.nsa.gov/iaarchive/library/ia-guidance/security-configuration	Security configuration guides from the National Security Agency
https://csrc.nist.gov/publications and https://www.nist.gov/publications	Security guidelines from the National Institute of Standards and Technologies
https://.sectools.org	Top 125 security tools as generated from a survey of NMAP users
https://seclists.org	List of lists; good security-oriented mailing lists
https://www.securityfocus.com	Mailing lists, news, and vulnerabilities
cve.mitre.org	Along with the vulnerability database section of security focus, offers a good place to begin research on potential vulnerabilities

Remember that Google is your friend, and a wealth of information is available on the Internet about how Unix and Linux systems work. For example, try searching for **command list unix**.

Master Checklists

This chapter covers several methods for auditing Unix hosts and their variants. Because there are so many variants, it is impossible to list every occurrence you'll run across. Here is a list of the items we reviewed in this chapter.

Auditing Account Management

Checklist for Auditing Account Management	
❑	1. Review and evaluate procedures for creating Unix or Linux user accounts and ensuring that accounts are created only when there's a legitimate business need. Also, review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.
❑	2. Ensure that all user IDs in the password file(s) are unique.
❑	3. Ensure that passwords are shadowed and use strong hashes where possible.
❑	4. Evaluate the file permissions for the password and shadow password files.
❑	5. Review and evaluate the strength of system passwords and use of stronger forms of authentication.
❑	6. Evaluate the use of password controls such as aging.
❑	7. Review the process used by the system administrator(s) for setting initial passwords for new users and communicating those passwords.
❑	8. Ensure that each account is associated with and can be traced easily to a specific employee.
❑	9. Ensure that invalid shells have been placed on all disabled accounts.
❑	10. Review and evaluate access to superuser (root-level) accounts and other administration accounts.
❑	11. Review controls for preventing direct root logins.
❑	12. Review and evaluate the use of groups, and determine the restrictiveness of their use.
❑	13. Evaluate the use of passwords at the group level.
❑	14. Review and evaluate the security of directories in the default path used by the system administrator when adding new users. Evaluate the use of the "current directory" in the path.
❑	15. Review and evaluate the security of directories in root's path. Evaluate the use of the "current directory" in the path.
❑	16. Review and evaluate the security of user home directories and config files. They generally should be writable only by the owner.

Auditing Permissions Management

Checklist for Auditing Permissions Management	
❑	17. Evaluate the file permissions for a judgmental sample of critical files and their related directories.
❑	18. Look for open directories (directories with permission set to `drwxrwxrwx`) on the system and determine whether they should have the sticky bit set.
❑	19. Evaluate the security of all SUID files on the system, especially those that are SUID to root.
❑	20. Review and evaluate security over the kernel.
❑	21. Ensure that all files have a legal owner in the /etc/passwd file.
❑	22. Ensure that the `chown` command cannot be used by users to compromise user accounts.
❑	23. Obtain and evaluate the default umask value for the server.
❑	24. Examine the system's crontabs, especially root's crontab, for unusual or suspicious entries.
❑	25. Review the security of the files referenced within crontab entries, particularly root's crontab. Ensure that the entries refer to files that are owned by and writable only by the owner of the crontab and that those files are located in directories that are owned by and writable only by the owner of the crontab.
❑	26. Examine the system's scheduled atjobs for unusual or suspicious entries.

Auditing Network Security and Controls

Checklist for Auditing Network Security and Controls

- ❑ 27. Determine what network services are enabled on the system, and validate their necessity with the system administrator. For necessary services, review and evaluate procedures for assessing vulnerabilities associated with those services and keeping them patched.
- ❑ 28. Execute a network vulnerability scanning tool to check for current vulnerabilities in the environment.
- ❑ 29. Review and evaluate the use of trusted access via the /etc/hosts.equiv file and user .rhosts files. Ensure that trusted access is not used or, if deemed to be absolutely necessary, is restricted to the extent possible.
- ❑ 30. Review and evaluate the usage of trusted access via SSH keys.
- ❑ 31. If anonymous FTP is enabled and genuinely needed, ensure that it is locked down properly.
- ❑ 32. If NFS is enabled and genuinely needed, ensure that it is secured properly.
- ❑ 33. Review for the use of secure protocols.
- ❑ 34. Review and evaluate the use of .netrc files.
- ❑ 35. Ensure that a legal warning banner is displayed when a user connects to the system.
- ❑ 36. Review and evaluate the use of modems on the server.

Auditing Security Monitoring and Other General Controls

Checklist for Auditing Security Monitoring and Other General Controls

- ❑ 37. Review the `su` and `sudo` command logs to ensure that when these commands are used, they are logged with the date, time, and user who typed the command.
- ❑ 38. Evaluate the syslog to ensure that adequate information is being captured.
- ❑ 39. Evaluate the security and retention of the wtmp log, sulog, syslog, and any other relevant audit logs.
- ❑ 40. Evaluate security over the utmp file.
- ❑ 41. Review and evaluate system administrator procedures for monitoring the state of security on the system.
- ❑ 42. If you are auditing a larger Unix/Linux environment (as opposed to one or two isolated systems), determine whether a standard build exists for new systems and whether that baseline has adequate security settings. Consider auditing a system freshly created from the baseline.
- ❑ 43. Perform steps from Chapter 5 as they pertain to the system you are auditing.

CHAPTER 9

Auditing Web Servers and Web Applications

Web servers and web-based applications have provided interfaces for people and systems to access and manipulate information for over 30 years. As the Internet has evolved, web services have become more and more complex, supporting rich, interactive experiences, mobile applications, and more. While application models can vary, some basic principles of web service controls can apply to a wide range of systems. This chapter covers the following:

- How to audit a web server
- How to audit a web application

Background

The concept known as the World Wide Web began in the late 1980s with Tim Berners-Lee and Robert Cailliau as a way to improve references in text documentation. Berners-Lee, at the time a software engineer at the European Organization for Nuclear Research (CERN) in Switzerland, wanted a better way to access and share the information stored on computers. The foundations developed by Berners-Lee and Cailliau, including web servers, web browsers, and even the language and formatting used by web pages, are still the underpinnings of the Web 30 years later. The web interface has grown from simple, static pages to the interactive online world we experience today with browsers and mobile applications.

Businesses rely on web applications to connect with customers via websites, to manage human resources systems, for e-mail and collaboration, and more. Today, nearly every business has an online presence in the form of one or more websites, and many key business processes rely on web technology. With the prevalence and criticality of these applications, it's important for you as an auditor to ensure that appropriate controls are in place to protect company interests.

Web Auditing Essentials

Verizon's 2018 Data Breach Investigations Report identified web application attacks as a component in nearly 20 percent of successful company breaches. In addition, Verizon noted over 23,000 incidents in the 2018 report where a compromised web application was used for some other malicious purpose, such as spam or phishing, or was repurposed as part of an attack against another organization. Web servers are common targets. They can be difficult to secure, and they often

contain or are used to access company secrets, personal information, or financial data such as credit card records.

Like much of auditing, assessing a web server is not an exact science. Many different types of web servers are in use (servers running on Apache, nginx, or Microsoft Internet Information Services are the most common) and millions of different applications rely on web services. A higher and higher percentage of web traffic is application related today, with systems exposing application programming interfaces (APIs) as web services. The wide variety of systems reliant on web services can complicate an audit; you can simplify the situation by sticking to the basics, which are applicable to almost any web-based system.

An organization with a strong security program will have a number of additional controls designed to protect web servers. Web application firewalls (WAFs), network port controls, and reverse proxies are technologies you may find in a web environment. In addition, mature organizations will conduct or will contract with outside services for regular vulnerability testing and/or penetration testing related to web systems. A web server cannot rely merely on a clean audit for protection; layers of defense are strongly recommended. Some of these supplemental controls are discussed next.

The Center for Internet Security (CIS) publishes a number of hardening guides for commonly used technologies, including Apache and Microsoft Internet Information Services (IIS). These guides contain detailed technical configuration instructions as well as verification steps and are an excellent resource for auditors and system administrators. You should obtain a copy of the CIS hardening guides when available and review the recommendations therein.

One Audit with Multiple Components

A complete web audit is really an audit of three primary components, including the server operating system, web server, and web application. These three components are shown in Table 9-1. Additional components such as a supporting database or relevant network infrastructure may also be appropriate to consider as part of your audit.

Web Audit Component	Key Concerns
Server operating system	Security of the host platform and operating system
Web server	Default settings, sample code, general misconfigurations, logging
Web application	Development framework security settings, default application settings, input validation, incorrectly serving up data, access to company-confidential data, general misconfigurations

Table 9-1 Web Auditing Components

The first component we discuss is the underlying platform or operating system on which the web server and application are installed and operate. Next is the web server itself, such as Microsoft IIS or Apache, that is used to host the web application. Finally, we cover an audit of the web application.

A wealth of languages and structures are available for web application development, complicating the audit process. However, several tools and methods are also available to help us determine what needs attention. The steps that follow cover these tools and methods. Keep in mind that if the following steps don't fully cover your concerns as an auditor, you should review Chapter 15, which covers auditing applications. Chapter 15 is intentionally geared toward conceptually breaking down complex or infrequent audits.

Part 1: Test Steps for Auditing the Host Operating System

The host operating system audit should be conducted in conjunction with the audit of the web server and web application(s). Please see Chapters 7 and 8 on Windows or Unix, as appropriate, for the audit of the platform.

Part 2: Test Steps for Auditing Web Servers

Once the operating system controls are in order, the web server layer should be examined. The basic steps provided here apply to most web server environments, regardless of the underlying operating system platform or the specifics of the web application itself. Most of the audit steps involve interviews or discussions with application or server administrators. Where technical commands are referenced, they are generally for Apache-based servers. Your server or application administrator should be able to help you identify similar commands for IIS or other server types.

1. Verify that the web server is running on a dedicated logical system not shared with other critical applications.

A compromised web host may allow the attacker to compromise other applications on your server. You should use a dedicated logical machine for your web server, either in the form of a separate physical host or, more likely, a separate virtual machine. For example, you would never want to install your web server on the same system that runs an Active Directory domain controller.

How

Identify and discuss each application with the administrator. Carefully consider the legitimate business need to allow other applications to remain on the same host as the web server. If these applications must coexist, consider bringing each of the additional applications into the scope of the audit.

2. Verify that the web server is fully patched and updated with the latest approved code.

Failure to run adequately patched systems subjects the web server to an unnecessary risk of compromise from vulnerabilities that could have been patched with updated code releases.

How

Every organization has its own patch management systems and policies. With the help of an administrator, verify that the web server is running the latest approved code according to the policies and procedures in the environment. For Apache systems on Unix platforms, you can use the command `httpd –v` to see the Apache version in use. You should also review the policies and procedures for ensuring that systems are kept up to date with the latest code releases.

Web servers often fall into a "middleware" bucket from an enterprise support viewpoint. Many infrastructure teams consider web servers to be part of the application stack, while many application teams, noting that web servers are usually installed with server baselines, consider them part of the infrastructure scope.

Determining who in your organization has ownership of web server patching and maintenance will go a long way toward resolving patching questions and other concerns of web server audits.

You should keep in mind that some applications may have embedded web servers, and the patch management process for these may differ. Some customized web servers may also rely on specific libraries or binaries and may be difficult to patch. Your organization should have proper documentation for any web server falling into these groups.

3. Verify that unnecessary services, modules, objects, and APIs are removed or disabled. Running services and modules should be operating under the least privileged accounts.

Unnecessary services, modules, objects, and APIs present an additional attack surface area, resulting in more opportunities for malicious attackers and malware.

How

Discuss and verify, with the help of the administrator, that unnecessary services are disabled and that the running services are operating under the least privileged account possible. Verify that File Transfer Protocol (FTP), Simple Mail Transport Protocol (SMTP), Telnet, extra server extensions, and Network News Transfer Protocol (NNTP) services are disabled if they are not required (many of these are also covered under host operating system audits). Services such as these often transmit credentials in cleartext and are not needed for the operation of most applications. You can use `netstat` or a more robust port-mapping utility to identify running network services. Many web servers have robust management interfaces whereby you can review additional installed modules and plugins.

Review logs and configuration files to validate that only necessary modules are enabled. Question the need for anything else that might be running.

4. Verify that only appropriate protocols and ports are allowed to access the web server.

Minimizing the number of protocols and ports allowed to access the web server reduces the number of attack vectors available to compromise the server.

How

Discuss with the administrator and verify with the administrator's help that only necessary protocols are allowed to access the server. For example, the TCP/IP stack on the server should be hardened to allow only appropriate protocols. NetBIOS and Server Message Block (SMB) should be disabled on IIS servers. Note any additional controls that may be in place, such as firewall rules or network access control lists (ACLs) to limit the protocols and ports allowed to access the web server. In general, only TCP on ports 80 (http) and 443 (https) should be allowed to access the web server. In addition, in certain cases it may be necessary to review the negotiated ciphers allowed by Transport Layer Security (TLS) transactions. A vulnerability scanning tool can identify cipher-related issues. Review these decisions with the administrator.

In addition, you should examine the broader network protections in place for defending against web server attacks. Some common defenses include:

- **Web application firewall (WAF):** A WAF is an application-aware firewall that examines web requests for malicious content. A WAF can be installed as a stand-alone device, as a feature in other firewalls, as an add-on module for a web server, or as a third-party, network-based service. If a WAF is used, you should examine how it is maintained and how alerts from the WAF are handled by the organization.
- **Reverse proxy:** A reverse proxy is a special kind of intermediary web server that processes requests from the outside and directs them appropriately. Reverse proxies can be used as firewalls to enable very limited traffic patterns to the target web system. If a reverse proxy is used, you should work with the administrator to understand how it is configured and what additional protections are offered. If the system is merely a pass-through, it is effectively providing no protection.
- **Denial-of-service (DoS) prevention:** In a DoS attack, a system is overloaded with requests, causing a crash or other unpredictable behavior. While DoS attacks can be difficult to defend at the server itself, network architectures, including rate limiting, load balancing, and other defenses, can be deployed to reduce the likelihood or impact of a DoS condition. Discuss this with the network administrators. DoS protection may be applied strategically to critical applications rather than in an across-the-board fashion.
- **Bot defenses:** In 2017 a report was released by Imperva, a security firm, showing that over 50 percent of web traffic was generated by "bots" rather than by people. Some bots, such as those used by popular search engines, may be beneficial in indexing content for the Internet; some bots are malicious in nature and crawl sites looking for vulnerabilities to exploit. Depending on the nature of the application and content, your organization may not want bots crawling your site. Discuss this with your web server administrators or web application teams to determine if bot defenses are deemed necessary. If in place, determine how these defenses are maintained and what action, if any, is taken in response to alerts or other metrics generated by the defense platform.

5. Verify that accounts allowing access to the web server are managed appropriately and use strong passwords.

Inappropriately managed or used accounts could provide easy access to the web server, bypassing other additional security controls that prevent malicious attacks. This is a large step with a wide scope, covering controls around account use and management.

How

Discuss and verify with the system administrator that unused system accounts are removed from the server or completely disabled. The default administrator account on Windows servers should be renamed, and all accounts should be restricted from remote login except for those used for administration. The root account on Unix-flavored hosts (Linux, Solaris, and so on) should be strictly controlled and never used for direct remote administration.

In general, accounts never should be shared among administrators, and administrators should never share their accounts with users. Strong account and password policies always should be enforced by the server and by the web server application. Multifactor authentication should be considered for privileged ac-

counts or in any situation where an account might be accessible from the Internet, such as in public cloud infrastructures.

Web servers should always run under a distinct user and group and should never run under an administrator profile. In Unix, this means web servers should never run as root; in Windows, web servers should not run under accounts with administrative privileges. For Apache installations, you can use the command `apachectl –S` to list the user and group for Apache.

Additional considerations for IIS web servers include ensuring that the IUSR_MACHINE account is disabled if it is not used by the application. You also should create a separate, low-privilege anonymous account if your applications require anonymous access. Configure a separate anonymous user account for each application if you host multiple web applications.

6. Ensure that appropriate controls exist for files, directories, and virtual directories.

Inappropriate controls for files and directories used by the web server and the system in general allow attackers access to more information and tools than should be available. For example, remote administration utilities increase the likelihood of a web server compromise.

How

Verify that files and directories have appropriate permissions, especially those containing the following:

- Website content
- Website scripts
- System files (such as %windir%\system32 or web server directories)
- Tools, utilities, and software development kits

Sample applications and virtual directories should be removed. Discuss and verify with the administrator that logs and website content are stored on a nonsystem volume where possible.

Also verify that anonymous and everyone groups (world permissions) are restricted except where absolutely necessary. Additionally, no files or directories should be shared out by the system unless necessary.

7. Ensure that unnecessary information such as version and directory listings are not exposed through the web interface.

Web servers that expose information unnecessarily can leak sensitive data or make it easier for attackers to find vulnerabilities. Simple settings in server configuration can be checked to verify that information is not exposed.

How

For Apache servers, examine the httpd.conf file in the server's conf directory. Look for the following entries and ensure the values match:

```
ServerTokens Prod
ServerSignature Off
```

These settings configure Apache to provide minimal information in response headers.

While reviewing httpd.conf, you should also look for the Directory attribute and ensure that Options is set to either None or –Indexes:

```
<Directory /opt/apache/htdocs>
Options –Indexes
</Directory>
```

This setting prevents Apache from returning a directory listing when a referenced URL is a folder that does not contain index.html or another configured default page file.

8. Ensure that the web server has appropriate logging enabled and that monitoring processes are in place.

Logging auditable events helps administrators troubleshoot issues. Logging also allows incident response teams to gather forensic data.

How

Review logging policies in your organization as applicable to web servers and discuss compliance with your web server team. Web servers can be expected to have logs related to login and access events that include user information, IP addresses, and other data. Ensure that logs are retained per company policy. High-value web servers, including those with sensitive or regulated data, should have their logs transferred to a central log storage and analysis facility. These may also log additional transaction data.

You should also discuss monitoring practices with the security operations center (SOC) or other group responsible for monitoring server logs. Determine whether the SOC is able to identify alerts for critical web servers and respond accordingly. SOC processes and incident response activities are covered in detail in Chapter 4.

9. Ensure that script extensions are mapped appropriately.

Scripts might allow an attacker to execute the code of his or her choice, potentially compromising the web server.

How

Verify with the web administrator that script extensions not used by the web server are mapped to a 404 web page handler or simply denied altogether. Examples of extensions that you may or may not use include .idq, .htw, .ida, .shtml, .shtm, .stm, .idc, .htr, and .printer.

10. Verify the validity of any server certificates in use.

Server-side certificates enable clients to trust your web server's identity. Old or revoked certificates may prevent end users from visiting your site or may produce warnings or errors.

How

Verify with the help of the administrator that any certificates are valid, are being used for their intended purpose, and have not been revoked. Certificate date

ranges, public key, and metadata all should be valid. Discuss the certificate expiration and renewal process. While browsers have tended toward alerting users or disallowing access for certificate errors, this may not be the case for all browsers used to access your application. Ensure that any certificates in use are valid.

Certificates may be self-signed or may be issued by a certificate authority (CA). Self-signed certificates are sometimes used for internal websites or management interfaces when an organization can also configure internal browsers to accept self-signed certificates. CA-issued certificates usually come with a cost. CA-issued certificates assert the validity of your site from a trusted authority and are used by almost all Internet-facing websites. CA certificates are also easier to deploy from an end-user viewpoint, since browsers come preconfigured with trust information for many certificate authorities and do not automatically trust a self-signed certificate. Work with your web administrator to determine the certificate types in use.

Part 3: Test Steps for Auditing Web Applications

This section represents an approach to the application audit as represented by the Open Web Application Security Project (OWASP) Top Ten web application security risks.

According to its website, OWASP is "dedicated to enabling organizations to develop, purchase, and maintain applications that can be trusted." OWASP maintains a tremendous amount of information that can help you develop an audit program for your web applications. The OWASP Top Ten are updated periodically and are regarded as a set of *minimum* standards for web application security. As such, the steps in this chapter should be considered a starting point for a web application audit.

Your web application design may call for additional testing, including a partial or full code review, third-party penetration testing, use of commercial code scanners, or open-source review tools. Each of these can offer some additional assurance that your application is correctly designed and configured. Consider the business value of the web application, and invest in the appropriate resources to ensure that your application is secure. Additional guidance on how to effectively find vulnerabilities in web applications are provided in the OWASP Testing Guide and the OWASP Code Review Guide found at www.owasp.org.

NOTE Keep in mind that the audience of this book varies greatly in technical abilities, and an attempt has been made to simplify the content in this section as much as possible for the majority of the readers. You will find further guidance by visiting www.owasp.org to determine what scope and toolset are most appropriate for your environment.

1. Ensure that the web application is protected against injection attacks.

Injection attacks occur when an attacker adds unexpected commands or queries to expected parameters, tricking the application into performing a different task. A common type of injection attack is called SQL injection, where Structured Query Language (SQL) commands are inserted into web forms or other fields. Using SQL injection, an attacker can cause an application to delete data, add new data, or execute other commands against the database. Many websites have exposed credit card and Social Security information to hackers who have taken advantage of injection attacks.

Failure to realize the power of injection attacks and to review your systems for the likelihood of being exploited may result in the loss of critical and sensitive information.

How

Discuss injection attacks with the administrator and web application development team as appropriate to ensure that they understand how such attacks work, and then ask how they are guarding against injection attacks. Preventing all possible injection attacks is difficult, but you still can take appropriate steps to defend your system against such attacks. Some of the methods provided next are also helpful in protecting against other attacks. Your organization may use some or all of the following methods:

- Validate all inputs and reject any data that does not match the expected input, such as values, length, and character sets.
- Perform a code review, if possible, for all calls to external resources to determine whether the method could be compromised.
- Open-source and commercial scanning tools are available that may help find injection vulnerabilities. OWASP maintains a list of such scanners at www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools.
- Consider hiring third-party help if the application is particularly sensitive, if you lack the resources, or if you need to verify items such as regulatory compliance.

NOTE These steps apply to the application development life cycle as much as they apply to an existing application. Development teams should be aware of security concerns and should plan during design and development to protect their applications.

2. Review the application for authentication and session management vulnerabilities.

Account credentials and session tokens must be protected. Attackers who can compromise passwords, keys, session cookies, or other tokens can defeat authentication restrictions and assume other users' identities and level of authorized access.

How

Discuss with the application team the authentication mechanism used to authenticate users to the web application. The web application should have built-in facilities to handle the life cycle of user accounts and the life cycle of user sessions. Verify that helpdesk functionality, such as lost passwords, is handled securely; helpdesks should have a reasonable process to verify the identity of someone requesting help. Walk through the authentication implementation with the application administrator, and then ask the administrator to demonstrate the functionality to you.

The following list of guiding principles may be helpful when it comes to check-

ing the authentication mechanism used on a website. These continue to maintain relevance despite the many advances in web design and products that promise secure user and session management:

- When a user enters an invalid ID or password into a login page, the application shouldn't identify which element was invalid. Generic errors are preferable, such as, "Your login information is incorrect."
- Implement multifactor authentication where possible.
- Never submit login information via a GET request. Always use POST. This makes it more difficult for an attacker to manipulate login requests.
- Use TLS (https) to protect login page delivery and credential transmission.
- Use strong passwords and change or disable any default or vendor-provided accounts.
- Limit the number of failed login attempts or introduce a time delay when multiple logins have failed.
- Ensure session IDs are randomly generated, as sequential or predictable session IDs can be used in hijack attacks. Ensure session IDs are not visible in the URL.
- Ensure logout processes invalidate session IDs and other session information.

TIP OWASP has an excellent reference on authentication at www.owasp.org/index.php/Authentication_Cheat_Sheet.

3. Verify that sensitive data is identified and protected appropriately. Ensure proper use of encryption technologies to protect sensitive data.

Web applications may handle extremely sensitive information, such as health records, financial data, company secrets, and more. Sensitive data must be identified and handled appropriately.

How

Begin the discussion with the web application team by reviewing the sensitivity of the data handled by the application. Determine whether any industry or regulatory drivers require encryption of data at rest. Ensure that weak ciphers are not in use. Review key management procedures to ensure that keys are properly protected and available only to authorized processes or individuals.

Ask the administrator about the web services access policies and the different methods of access for private areas of the web application; ensure that each access method and ongoing communications with the web application are performed using secure protocols. Secure access methods during authentication ensure that the user information (such as user ID) and authentication tokens (such as passwords) are encrypted. Secure communications prevent data from being viewed by eavesdroppers. Ask the administrator about session cookies to verify that the secure flag is set to prevent the browser from sending them in the clear.

Question the need for any cleartext communications. There may be cases where clear-text communications exist and are difficult to remove because of a legacy application or where the traffic is deemed unimportant. However, where possible, an encrypted protocol should be used instead. Exceptions should be extremely rare and limited to business-driven cases where senior management is willing to sign off on and formally accept the risk of cleartext communications. Encrypted communications are absolutely required under some conditions with no exceptions. Packages exist for nearly every scenario to encrypt communications.

NOTE The use of secure protocols is particularly important for externally facing web applications and others hosted in high-risk environments. The auditor may determine that the web application is of less importance on isolated secured internal networks. However, it is still advisable that secure protocols be used, even on internal networks, to minimize attacks from within the organization. In many cases, regulations and standards (such as HIPAA and PCI) forbid the use of cleartext communications.

OWASP suggests avoiding pages combining TLS and cleartext traffic. Many sites still serve mixed pages, and the pop-up messages tend to be confusing for the users. Worse, it begins to desensitize users to pop-up messages while browsing secure sites. Furthermore, the server certificates should be legitimate, current, and properly configured for the appropriate web servers and domains that the web application uses.

4. Review the web server for exposure to XML external entities (XXE) attacks.

Applications that accept or process Extensible Markup Language (XML) data could be vulnerable to XXE. A malicious actor could construct an XML file with commands seeking to extract sensitive data from a system or perform other commands. A recent addition to the OWASP list, XXE attacks aren't as widely known as others like injection.

How

Review developer training materials to ensure application developers are aware of XXE and how to defend applications. Interview application developers to determine if applications that process XML are employing input validation, whitelisting, or other filtering of XML input data. Some WAFs can defend against XXE, and there are also dedicated XML security gateways in the marketplace.

5. Verify that proper access controls are enforced.

After a user is authenticated to the web server, the web server determines what kind of access the user should have and to what parts of the website the user should have access. Failure to enforce access controls (authorization) to each object may allow an attacker to step out of authorized boundaries, accessing other users' data or unauthorized areas. Specifically, attackers should not be allowed to change parameters used during an authorized user session to access another user's data. Client proxy and other tools allow attackers to view and change data during sessions.

How

A good initial step is to discuss policy requirements around access management with the administrator or security team. Failure to have a policy or other written documentation around access concerns is a red flag that access controls may not be correctly enforced. Access controls are complicated and difficult to get right without carefully documenting your desired results.

Each web page type or web form should be tested with both authenticated and unauthenticated (anonymous) users to verify that only authenticated users have access—and only to what they are authorized to view. Verify and map out access to privileged pages, and then verify that authentication is required to access each page. Depending on the purpose of the website, an authenticated user may be granted access to all areas of a site. If an authorization concept is in place for the site, ensure that the expected restrictions are still in place once a user is authenticated.

NOTE Mitre.org maintains a Common Weakness Enumeration (CWE) entry on this topic, CWE-285: Improper Access Control (Authorization), located at http://cwe.mitre.org/data/definitions/285.html.

Automated tools may help, but a code review is a more effective method for identifying these issues. The reality for most audit teams is that few have the hours or skill required to comb through the code to identify and verify all object references or to review authorized access in general. Tools that may be helpful include Paros Proxy from www.parosproxy.org and Burp Suite. Both have ample documentation available describing how to use them.

6. Review controls surrounding maintaining a secure configuration.

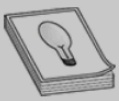
This is a catch-all that addresses configuration management, the overarching concept of maintaining the secure configuration of the web server. Failure to maintain a secure configuration subjects the web server to lapses in technology or processes that affect the security of the web platform and web application. Some of these items are also covered in other points in this chapter; they are included here for alignment with the current OWASP Top Ten.

How

Perform the web platform and server audit, and discuss any issues noted with the administrator. Determine whether any of the issues noted are due to inadequate configuration management. Discuss the following with the administrator to ensure that proper configuration management controls are in place:

- Security mailing lists for the web server, platform, and application are monitored.
- The latest security patches are applied in a routine patch cycle under the guidance of written and agreed-to policies and procedures.
- A security configuration guideline exists for the web servers, development frameworks, and applications in the environment covering default account management, installed components, and security settings and is strictly followed. Exceptions are carefully documented and maintained.
- Regular internal reviews of the server's security configuration are conducted to compare the existing infrastructure with the configuration guide.
- Regular vulnerability scanning from both internal and external perspectives is conducted to discover new risks quickly and to test planned changes to the environment.
- Management is aware of any outstanding security risks related to web servers.

A strong server configuration standard is critical to a secure web application. Take the time to understand the available security settings and how to configure them for your environment.



TIP Secure web applications start with secure development processes. Check out OWASP's Open Software Assurance Maturity Model (SAMM) project online at www.owasp.org/index.php/OWASP_SAMM_Project.

7. Review the website for cross-site-scripting vulnerabilities.

Cross-site scripting (XSS) allows the web application to transport an attack from one user to another end user's browser. A successful attack can disclose the second end user's session token, attack the local machine, or spoof content to fool the user. Damaging attacks include disclosing end-user files, installing Trojan horse programs, redirecting the user to some other page or site, and modifying the presentation of content.

How

Web code scanning tools can detect many XSS vulnerabilities in an application, but it may be difficult to detect all possible combinations of XSS. A thorough code review can help. As with many of the other issues described here, developer security training is very important; if a developer doesn't know what an XSS attack is, it's very likely that his or her code is vulnerable to this type of attack.

If you were to review the code, you would search for every possible path by which http input could make its way into the output going to a user's browser. The key method used to protect a web application from XSS attacks is to validate every header, cookie, query string, form field, and hidden field. Again, make sure to employ a positive validation method.

CIRT.net is home for an open-source scanner, Nikto, that you might be able to use to help you partially automate the task of looking for XSS vulnerabilities on your web server. Keep in mind that these tools are not as thorough as conducting a complete code review, but they can at least provide more information to those who don't have the skill set or resources to conduct a complete review. Nikto is available from www.cirt.net/Nikto2.

NOTE Always keep in mind that these tools may find well-known attacks, but they will not be nearly as good as performing a solid code review.

If you don't have the internal resources available to perform a code review, particularly on a homegrown application, and you believe that the data on the website warrants a deep review, you may consider hiring third-party help.

8. Review protections against exploitation of deserialization sequences.

Deserialization concerns were added into the OWASP Top Ten in 2017 based on industry survey data. While somewhat challenging to exploit, the impact of a successful attack can result in the complete compromise of a system. This type of issue can occur when an application processes a sequence of data passed as part of a command or request. If an attacker is able to manipulate the request and pass unexpected data as part of the query, the application may not execute as expected.

How

Discuss this vulnerability with your application development teams. As discussed in OWASP literature, the best methods to avoid this type of exploit are to process only trusted serialized data inputs or to restrict data types used for serialization to known, expected data values. Other protections described in this chapter, such as running applications as low-privilege users and adequate logging, can improve the control landscape.

9. Review processes to ensure vulnerabilities are not present in libraries, frameworks, or other components.

While any application may have hidden, difficult-to-discover vulnerabilities, you can greatly improve your defensive posture by ensuring that already-known vulnerabilities are not present. Unfortunately, the complexity of modern websites can make this a difficult task, as shared frameworks, libraries, databases, and other dependencies significantly expand the opportunity for vulnerabilities to exist in the site.

How

This step involves discussion with web application development teams and web administrators. Many of the controls here are similar to those in step 6 earlier. Ensure that the scope of patch management and vulnerability scanning programs includes all of the components, frameworks, and other dependencies in the web server environment. Review how components are obtained and whether an active support contract or support community exists for the component. Ensure that any unused features or files are removed from the environment to reduce the potential attack surface.

This activity can become a larger challenge as applications age, since formal support may end for various components and patches may no longer be available. The organization can be better prepared for this situation by keeping an accurate inventory of components needed on both client and server systems to run and support the application. With an up-to-date inventory, the teams are able to identify which components may no longer be supported.

10. Ensure that adequate logging is present and review processes for examining log data.

While some types of application attacks are very noisy and very noticeable, others are designed to avoid attention so that the attacker can persist in the environment. If an application does not log appropriate events and there's no monitoring of key activities or thresholds, an attacker could explore the system with impunity, looking for sensitive data and other systems to exploit. You should ensure that web systems are logging correctly and that the organization is able to identify and respond to adverse activity.

How

Review logging policies in your organization as applicable to web applications and discuss compliance with your web development team. Web applications can be expected to have logs related to login and access events that include user information, IP addresses, or other data. Ensure that logs are retained per company policy. High-value web servers, including those with sensitive or regulated data, should have their logs transferred to a central log storage and analysis facility. These may also log additional transaction data.

You should also discuss monitoring practices with the SOC or other group responsible for monitoring web application logs. Determine whether the SOC is able to identify alerts for critical web applications and respond accordingly. SOC processes and incident response activities are covered in detail in Chapter 4.

Additional Steps for Auditing Web Applications

Several of the following steps were part of earlier OWASP Top Ten publications and were discussed in previous editions of this book. Others are sound controls for any system. They remain relevant and appropriate to consider for your web application.

11. Review the security training provided to application development teams and ensure that development teams understand secure coding practices.

While this step isn't part of the OWASP Top Ten, security training is noted as a best practice for avoiding many of the issues discussed earlier. Development teams with proper security training can avoid many of the application vulnerabilities found in websites.

How

Discuss with application development teams how they are trained in security matters. You should expect that all application developers have a basic awareness of secure coding principles. Ideally, developers should be required to take some type of application security training on a periodic basis.

12. Verify that all input is validated prior to use by the web server.

Information must be validated before being used by a web application. Failure to validate web requests subjects the web server to increased risk from attackers attempting to manipulate input data to produce malicious results.

How

Discuss with the web application developer or web administrator the methodology used for input validation for the application you are testing.

Several tools effectively act as a proxy and allow you to see much of the content posted from your client to the remote web server. One such tool is Paros Proxy, located at www.parosproxy.org. In addition, most modern browsers contain development tools or inspection tools that can be used to examine content.

Another method used by professional web testers is to understand the movement of data during a code review. This isn't something that should be taken lightly because it may be beyond the scope of what you are trying to accomplish. There is a trade-off that you as an auditor must make regarding the level of rigor performed versus the value of the data or system you are reviewing.

In general, two ways to look at validation methods are negative methods and positive methods. Negative methods focus on knowing what bad input to filter out based on the "known bad." The problem with negative filtering is that we don't know now what tomorrow's vulnerabilities and input methods will bring. Positive filtering is much more effective and involves focusing on validating the data based on expected patterns and content. This is similar in approach to a firewall that denies everything except what should be accepted.

Common items for positive filtering include criteria you might find in a database or other systems that accept data. These include criteria such as

- Data type (e.g., string, integer, and real)
- Allowed character set
- Minimum and maximum length
- Whether null is allowed
- Whether the parameter is required or not
- Whether duplicates are allowed
- Numeric range
- Specific legal values (e.g., enumeration)
- Specific patterns (e.g., regular expressions)

13. Evaluate the use of proper error handling.

Improperly controlled error conditions allow attackers to gain detailed system information, deny service, cause security mechanisms to fail, or crash the server.

How

Proper error handling generally follows from careful planning around all potential inputs and outputs. Ask the administrator how error handling was designed into the web application and how errors are handled internally as the application interfaces with other compartmentalized functions. For example, how would the web application handle an error generated by the database? Does it make a difference whether the database is hosted internally by the application versus hosting the database externally on another server? How does the application handle input validation errors? What about username and password errors?

Error handling should be deliberate and should show structure during a code review. If the error handling looks haphazard or appears more like an afterthought, you may want to look much more closely at the application's ability to handle errors properly.

14. Review web application redirects and forwards to verify that only valid URLs are accessible.

Using an unvalidated redirect, attackers may be able to redirect users to the attacker's website using a URL that looks as if it comes from your domain. This is a preferred method for phishing scams to make requests appear valid by using the attacked organization's address in the first portion of the crafted URL. This is sometimes used in conjunction with a URL-shortening service for the target website to obfuscate the malicious intent of the URL.

```
http://www.mydomain.com/redirect.asp?url=badsite.com
```

In some cases, an unchecked forward can send a user to a privileged page that would otherwise be inaccessible if additional authorization controls are implemented incorrectly.

```
http://www.mydomain.com/somepage.asp?fwd=adminsite.jsp
```

How

Review with the administrator the use of any redirects and forwards within the web application to determine whether there are ways to avoid their use or implement safe controls around their use. Automated scanners may be used to scan and verify a website for proper handling of redirects and forwards.

Note that redirects and forwards are called transfers in Microsoft's .NET framework. OWASP recommends ensuring the supplied value is valid and authorized for the user when redirects and forwards cannot be avoided. Blind redirects and forwards are dangerous, and controls should limit the destination of both. There are many ways to implement redirects and forwards securely, but they should never be implemented blindly.

NOTE Mitre.org maintains a CWE related to this topic, CWE-601: URL Redirection to Untrusted Site ('Open Redirect'), located at cwe.mitre.org/data/definitions/601.html.

15. Verify that controls are in place to prevent cross-site request forgery (CSRF or XSRF).

Cross-site request forgery attacks exploit the trust a website places in the authenticated user. Attackers exploit this trust by sending embedded images, scripts, iframe elements, or other methods to call a command that executes on the web server while you are logged in with your credentials. Making matters worse for the user, this type of attack originates from the IP address of the user, and any logged data will appear as if the logged-in user entered it.

Web servers should validate the source of web requests to minimize the risk from attackers attempting to create authenticated malicious web requests that

originate from sources outside the control of the web application. Here is an example of how this type of attack might look as an image request:

```
<img                          src="http://mybank.com/transfer?
acct=mine&amt=100&to=attacker">
```

How

Discuss with the web application developer or web administrator the methodology used for uniquely creating tokens for each link and form for state-changing functions. Information generated by the client browser, such as the IP address or session cookie, is not a valid token element because it can be included in forged requests. If the token can be predicted, the web application is most likely subject to this type of attack.

Several tools can act as a proxy and allow you to see the content posted from your client to the remote web server. One such tool is Paros Proxy. If you can repeatedly replay the same URL over time to achieve the same result, your application may be vulnerable.

Another method used by professional web testers is to review the handling of requests during a code review. The preferred method for handling the unique token is outside of the URL, such as in a hidden field. OWASP provides tools for developers to create applications that securely create and manage unique tokens.

Tools and Technology

There are several reasons why an automated product can fail to thoroughly audit every possible component of your web server, but that doesn't mean these products should be ignored. Code reviews may not always be time consuming, but this depends on many variables. For example: How experienced is the reviewer? How well does the reviewer understand the web application? How well does the reviewer understand the constructs of the programming language used for the application? How complex is the application? What external interfaces exist, and how well does the reviewer understand these external interfaces?

If you have previous application development experience, code reviews may be easy for you. If not, you may want to consider augmenting your searches with automated tools, especially if third-party help isn't an option.

CAUTION Automated tools can be quite harmful to production environments. Exercise care, and design the test in a manner that will not affect production systems.

Automated tools can be very helpful and can guide you toward parts of your web platform or web application that need further review, and many are available. The following list includes only a sampling of what's out there. Many general vulnerability scanners also test commonly exploited vulnerabilities.

Tool	URL
Kali Linux	www.kali.org
Burp Suite	https://portswigger.net/burp/
Samurai Web Testing Framework	www.samurai-wtf.org
Paros Proxy	www.parosproxy.org
Nikto	www.cirt.net/Nikto2
XSS plug-in for Nessus	www.tenable.com/plugins/nessus/39466
Apache JMeter	jmeter.apache.org

Knowledge Base

Following you will find additional resources where you can obtain information about web application environments and related controls. Many vendors maintain a tremendous amount of information on their website for general consumption. Additionally, the community of helpful enthusiasts and social forums continues to grow.

Site	URL
Apache	www.apache.org
Microsoft IIS	www.iis.net
Center for Internet Security (CIS) Hardening Guides	www.cisecurity.org/cis-benchmarks
UrlScan tool for IIS	www.iis.net/downloads/microsoft/urlscan
CGI Security	www.cgisecurity.com
Common Weakness Enumeration (CWE)	cwe.mitre.org
Google Web Fundamentals—Web Security	https://developers.google.com/web/fundamentals/security/

Master Checklists

The following tables summarize the steps for auditing web servers and web applications.

Auditing Web Servers

Checklist for Auditing Web Servers
❑ 1. Verify that the web server is running on a dedicated logical system not shared with other critical applications.
❑ 2. Verify that the web server is fully patched and updated with the latest approved code.
❑ 3. Verify that unnecessary services, modules, objects, and APIs are removed or disabled. Running services and modules should be operating under the least privileged accounts.
❑ 4. Verify that only appropriate protocols and ports are allowed to access the web server.
❑ 5. Verify that accounts allowing access to the web server are managed appropriately and use strong passwords.
❑ 6. Ensure that appropriate controls exist for files, directories, and virtual directories.
❑ 7. Ensure that unnecessary information such as version and directory listings are not exposed through the web interface.
❑ 8. Ensure that the web server has appropriate logging enabled and that monitoring processes are in place.
❑ 9. Ensure that script extensions are mapped appropriately.
❑ 10. Verify the validity of any server certificates in use.

Auditing Web Applications

Checklist for Auditing Web Applications	
❑	1. Ensure that the web application is protected against injection attacks.
❑	2. Review the application for authentication and session management vulnerabilities.
❑	3. Verify that sensitive data is identified and protected appropriately. Ensure proper use of encryption technologies to protect sensitive data.
❑	4. Review the web server for exposure to XML external entities (XXE) attacks.
❑	5. Verify that proper access controls are enforced.
❑	6. Review controls surrounding maintaining a secure configuration.
❑	7. Review the website for cross-site-scripting vulnerabilities.
❑	8. Review protections against exploitation of deserialization sequences.
❑	9. Review processes to ensure vulnerabilities are not present in libraries, frameworks, or other components.
❑	10. Ensure that adequate logging is present and review processes for examining log data.
❑	11. Review the security training provided to application development teams and ensure that development teams understand secure coding practices.
❑	12. Verify that all input is validated prior to use by the web server.
❑	13. Evaluate the use of proper error handling.
❑	14. Review web application redirects and forwards to verify that only valid URLs are accessible.
❑	15. Verify that controls are in place to prevent cross-site request forgery (CSRF or XSRF).

CHAPTER 10

Auditing Databases

In this chapter we discuss auditing the lockboxes of company information. We will discuss how to conduct audits on the following components that affect the operational security of your data stores:

- Database permissions
- Operating system security
- Password strength and management features
- Activity monitoring
- Database encryption
- Database vulnerabilities, integrity, and the patching process

Background

The term *database* typically refers to a relational database management system (RDBMS). Database management systems (DBMS) maintain data records and their relationships, or indexes, in tables. Relationships can be created and maintained across and among the data and tables.

The more generic term *database* can be applied to any collection of data in any structured form. For instance, a flat file that contains customer records can serve as a database for an application. However, in this chapter, we primarily focus on auditing a full-blown RDBMS, but we will also cover to a lesser extent the newest type of nonrelational database management system, called NoSQL. NoSQL was born out of the necessity to often store and query disparate types of data that don't lend themselves easily to structured schemas. They are appropriate for the so-called "big data" implementations.

Typically, an audit includes a fairly in-depth review of various areas, including the perimeter, the operating system, policies, and so on. If time allows, an audit might cover one or two of the most critical databases. Databases are complex beasts requiring patience and technical know-how to audit and secure properly. However, neglecting a database audit is a serious error. Databases are the virtual lockboxes of the information age. Where do organizations store their most valuable assets? Not in perimeter devices, not in an e-mail system, and not in a flat file. They are stored in a database. When you hear about a security breach and sensitive data being stolen, ask yourself where that data "lived" when it was at-

tacked. In a database!

Databases have some distinct advantages and disadvantages. Because databases are almost always buried deep and far behind the firewall, they are rarely exposed to the types of attacks that your web servers, firewalls, and other systems confront. Most organizations are smart enough to know not to place their most valuable data out in the unsecured public network. In the "old" days, this was effective against external attacks; however, modern attackers have employed methods that can bypass external defenses. Of course, some attacks, such as SQL injection, can easily make their way through a firewall and hit the database.

Databases have disadvantages for the same reasons. Because databases are so far behind the firewall, securing and auditing your databases are often considered afterthoughts, something to be done if you have extra time and maybe just on one or two critical databases. This has led to a situation in which database security typically is left in a shabby condition. The typical database administrator believes that the database is far enough behind the firewall that even rudimentary security measures aren't necessary.

The secured perimeter might serve as enough protection for the database in a perfect world. Unfortunately, we don't live in a perfect world, and the firewall is no longer a valid "last line of defense." Focus is now shifting to protecting data right where it sits—in the database. As an auditor, you are likely to find that the database is the weak link in the security chain. And, luckily, a few relatively simple recommendations can result in vast improvements in database security.

Database Auditing Essentials

To audit a database effectively, you need to understand a broad set of components around how a database works. Here's a little history lesson.

In the early 1990s, applications were written using the client-server model, which comprised a desktop program connecting over a network directly to a database back end. This was referred to as a *two-tier application.* In the late 1990s, *three-tiered applications* became the norm. This new model consisted of a web browser connecting to a middle-tier web application. The middle tier then connected to the database back end. Three-tiered applications were a great step forward. It meant that custom software didn't need to be installed on every client workstation, and software updates could be applied to a central server. Clients could run any operating system that supported a basic browser. Moreover, in the three-tiered model, securing the database was much simpler.

Of course, the infrastructure required by the database to support two-tier applications still exists in database back ends for three-tiered applications. The danger now exists that an attacker will circumvent the web application to attack the back-end database.

Common Database Vendors

Typically, an audit engagement will focus on one of a few database vendors, such as Microsoft, Oracle, or IBM. However, any medium-sized or large organization typically will use a sampling of many different database platforms. Following is a summary of the most common databases and vendors, along with a short overview of each.

Oracle

Oracle Corporation is the largest database vendor and supplies multiple series of databases, including the Oracle Database, Oracle NoSQL, and MySQL product lines. In addition, Oracle Corporation has grown beyond standard database software to provide a variety of products, including but not limited to cloud services, web servers, development tools, identity-management software, a collaboration suite, and multiple enterprise resource planning (ERP) solutions.

In the database market, the Oracle Database has one of the largest install bases and an impressive feature set. The database comes in multiple flavors, including Standard Edition One, Standard Edition, Enterprise Edition, Express Edition, and Personal Edition. Most Oracle databases you audit will be either Standard Edition or Enterprise Edition. The basic features are fairly similar; however, the advanced features in Enterprise Edition are changing constantly, so you will need to access the Oracle website to check the exact feature sets included in the version you are auditing.

IBM

IBM is another of the largest database vendors. Although IBM's database software is a small piece of the company's business, it does supply multiple series of databases, including the Db2, Informix, and Information Management System product lines. IBM's main database is the Db2 product line that comprises two main products:

- Db2 Universal Database, providing database software for AIX, Linux, HP-UX, Sun, and Windows
- Db2 Universal Database for z/OS, providing software for the mainframe

A lot of confusion surrounds the nomenclature of these two products. Typically, people refer to Universal DB (UDB) as the Linux, Unix, and Windows version and Db2 as the mainframe version. This is a misnomer, because UDB is actually a term used for all of IBM's latest Db2 software. Understand what people mean when they use these terms, but try to use the correct terms to avoid confusion. As recently as 2017, IBM rebranded its DB2 product line as "Db2." At the time of this writing, IBM supports Db2 for Linux, Unix, Windows, z/OS, i (previously OS/400), VSE and VM, and IBM Cloud.

Note that modern implementations of Db2 support not only relational database management system models but also object relational features, as well as nonrelational structures.

MySQL

MySQL is an open-source database used extensively in small or medium-sized web applications. MySQL was developed under the GNU Public License by MySQL AB, a privately held Swedish company. MySQL has a large and growing grassroots following and is the *M* in the *LAMP* (Linux, Apache, MySQL, and PHP) open-source web platform. MySQL AB was purchased by Sun in February 2008, and Sun was later purchased by Oracle in 2010, making MySQL an Oracle product. MySQL is used extensively to support popular websites and web-based applications, including Facebook, Twitter, YouTube, Flickr, and others.

MySQL traditionally has been a bare-bones database, providing a small fraction of the functionality available from other database vendors. From the security perspective, this is good, because MySQL does exactly what it was meant to do very well—and little else. Administration costs are relatively low, and MySQL provides

adequate performance for all but the most demanding web applications.

MySQL AB invested heavily in the MySQL database early on, ensuring that it could compete directly with its more expensive proprietary competitors. MySQL 5.0 added significant functionality, including stored procedures, views, and triggers. As of this writing, MySQL Server 8 was released in April 2018 and included nonrelational database management capabilities as well. It is one of the simpler databases to secure from hacking because of the relatively small attack surface it exposes. In addition, MySQL source code is available for anyone to see, which has led to a relatively secure and vulnerability-free code base. Vulnerabilities have been discovered in the MySQL source code, but security holes are discovered early in the life cycle of each release and are patched quickly.

MySQL AB also developed a second open-source database called MaxDB, which is designed specifically as a high-reliability back end for SAP systems. SAP acquired MaxDB in 2007 and is currently on version 7.9, as of January 2019.

Microsoft

Microsoft SQL Server is one of the most popular databases due to its low price tag and its simplistic administration model. Microsoft has continued to develop and maintain several mainstream and specialized editions of Microsoft SQL Server, each with different feature sets and targeting different use cases. As of this writing, SQL Server 2017 is the most current version and the first SQL Server version to include support for Linux platforms.

Mainstream editions of Microsoft SQL Server are the traditional database installations and include editions such as Enterprise, Standard, Web, and Express. Specialized editions of Microsoft SQL Server are geared toward very specific use cases, platforms, or environments. Specialized editions include Azure SQL Database, the cloud version of Microsoft SQL Server, and SQL Server Express LocalDB, a minimal instance of SQL Server Express targeted to developers. Most Microsoft SQL Server databases you audit will be either Standard Edition or Enterprise Edition. Like Oracle, the basic features are fairly similar across SQL Server editions; however, the advanced features in Enterprise Edition continue to evolve, so you will need to access the Microsoft website to check the exact feature sets included in the version you are auditing.

Microsoft SQL Server is often referred to as *SQL, SQL Server, MSSQL,* and even *MS SQL Server.* Although it's best to stick to the proper nomenclature to avoid confusion, it's important that you also understand the common, although incorrect, lingo.

Because Microsoft SQL Server is so easy to install and administer, it is often used by people with relatively little knowledge about securing it properly. This can lead to problems, not because Microsoft SQL Server is insecure, but because many people using it haven't taken even the most basic steps to protect it.

Database Components

Each database vendor has a slightly different implementation of the various database components. However, the theories and principles apply to all the different platforms fairly universally. We will cover enough of these basics to give you a bird's-eye view. From there, you should have enough background to follow a technical guide on a specific database platform. Following are the major pieces of the database that you will need to understand as an auditor.

Program Files

A database is implemented as a software application, and as such, it comprises a core set of application and operating system–dependent files. These files include the executable files that will run the database management system, as well as library files that become embedded in the OS. It also may contain other nonexecutable program files such as help files, source and include files, sample files, and installation files.

These files should be protected, because the database relies on their integrity. They should be guarded from any form of modification—particularly any executable files. Access controls should be as restrictive as possible on the directory that holds these files. Ideally, only database administrators (and in fewer cases, system administrators) should have access to this directory.

Configuration Values

Databases rely heavily on configuration settings to determine how the system operates. Protecting these settings is important, because if the configuration can be manipulated, security can be subverted.

Configuration values reside in a variety of places, including the following:

- In operating system text files
- In the data files
- On Windows, stored in the registry
- In environment variables

Configuration values are used for a wide range of settings, such as these:

- Setting the type of authentication or trust model
- Setting which groups are database administrators
- Determining password management features
- Determining the encryption mechanism used by the database

Verifying the integrity of configuration values is a critical component of any audit.

Data Files

Databases need to store the data they hold in physical operating system files that typically comprise a series of files. The format of the files is typically proprietary, and the data files contain information such as the following:

- Data being stored
- Pointers from one field to the next field or from one row to the next row
- Index data, including pointers from the index to the physical data

NOTE Indexes contain a subset of the data to which they point. This means that if an attacker can access the index, he or she may not need access to the physical data itself. Ensure that access to any index is protected to the same degree as the data itself.

Usually, the database dictionary is stored in these data files, so any access to these files can be used to circumvent controls built into the database.

Client/Network Libraries

An important component of any database system is the client. Typically, the client is located on a remote system from the database. The client also can connect from the local system, which is frequently the case with batch processes.

In order for a client to connect to the database, a client library or driver is required on the client's machine. This usually consists of a set of executables such as DLLs and shared objects, as well as an API that the client can use to connect to the database. If the client drivers can be manipulated, credentials can be stolen fairly easily. The client libraries are hard to protect because they usually exist on remote systems where access controls are much more difficult to maintain. However, it is very important to maintain the integrity of the client drivers in locations from where administrators or even regular users will be connecting.

Communication over the network also requires network drivers on the database. These drivers are another point of focus for the auditor, because they are the avenue that the attacker will use to access the database. Most modern database implementations are able to interface directly with the operating system network stack, such as TCP/IP, so this is less of an issue than in the past.

Backup/Restore System

Backups are a very important piece of every database platform. Failure in some component of the database is not a question of *if* but *when*. Whether the problem is a hardware or a software failure, having a backup is critical to restoring the system. Backups contain a copy of the whole database. The backup can be to a separate file, to a tape, or to another storage facility.

Data is commonly stolen from, lost, or leaked through the backup facility. Backups often are secured by encrypting the data as they are written to a file or by encrypting the entire file after it is written. Securely storing the encryption key then becomes important to securing the backup properly. Just as important is ensuring that you have properly backed up the encryption keys along with the data so that the backup can be restored properly. If you can't restore the files, the backup becomes worthless. Backups that cannot be restored result in a loss of utility. Note that databases can be backed up in several different ways, to include a full backup of all data files, as well as the database support files, as well as transactional backups, which only back up the most recent set of records in near real time. Transactional backups are effective when they back up the most recent transactions quickly and are less time consuming than full database backups. It is important to understand what data is being backed up and what the time between backups is and to ensure that the backups are restorable.

SQL Statements

Structured Query Language (SQL) is used to access data in a relational database. Technically, SQL should be pronounced as three separate letters "S-Q-L," but the pronunciation "sequel" has become so commonplace it is also accepted as correct. SQL is a set-based language, meaning that it works on a set of data at a time. It is not a procedural language, meaning that it does not have any procedural components such as while loops, if statements, for loops, and so on. Most database platforms do have extensions to SQL to provide procedural components. For instance, Oracle has PL/SQL, and Sybase and Microsoft SQL Server have Transact-SQL.

SQL statements are used to pull data from the database. SQL is built around four core statements:

- **SELECT** View a subset of data from a table
- **INSERT** Add new data to a table
- **UPDATE** Modify existing data in a table
- **DELETE** Remove a subset of data from a table

The statement you will need to understand best is `SELECT`. The basic syntax of the `SELECT` statement is

```
SELECT <COLUMN LIST> FROM <TABLE NAME> WHERE <CONDITION>
```

In this statement, `<COLUMN LIST>` is a comma-separated list of column names that will be displayed. As a shortcut, you can use an asterisk to display all columns in the output. `<TABLE NAME>` is replaced with the name of the table to be displayed. `<CONDITION>` and the word `WHERE` are optional. If you do not indicate a `WHERE` clause, all rows in the table are returned. Using the `WHERE` clause, you can `SELECT` only the rows you want to include.

An example of selecting the first and last names of all employees who earn more than $20,000 is shown here:

```
SELECT FIRST_NAME, LAST_NAME FROM EMPLOYEES WHERE SALARY >
20000
```

`SELECT` statements can get much more complex than this. Your audit typically does not need to go much deeper than this, however.

Note that SQL statements can be entered directly from a provided interface in the database management system or as part of an included utility in the RDBMS (usually graphical). SQL statements can also be entered through third-party clients that have been allowed to connect to the core RDBMS system. Some database clients also allow intuitive drag-and-drop actions to create SQL statements, thereby abstracting the statements from the user and reducing the need for the user to know SQL very well.

Database Objects

A database comprises a variety of objects, each with a unique task or purpose. Understanding each object is not necessary, but you should have a grasp of the common object types.

Following are the most common types of database objects. Each database platform also has many proprietary object types, such as table spaces, schemas, rules, sequences, and synonyms. You should review the specific documentation for your database platform for more details.

- **Table** Stores rows of data in one or more columns.
- **View** A `SELECT` statement on top of a table or another view that creates a virtual table. Views can change the number or order of columns, can call functions, and can manipulate data in a variety of ways.
- **Stored procedure/function** Procedural code that can be called to execute complex functionality within the database. Functions return values. Pro-

cedures do not return values; rather, they execute a series of stored commands. Stored procedures are very efficient for data access.

- **Trigger** Procedural code that is called when a table is modified. Can be used to perform any actions, including modifications to other tables, when data is changed.
- **Index** Mechanism to provide fast lookup of data. Indexes are complex objects, and their proper tuning is critical to database performance.

Data Dictionary

The database stores metadata about itself, called the *data dictionary* or sometimes the *system tables*. The metadata tells the database about its own configuration, setup, and objects (or schema). Note that the metadata does not say anything about the content of the information in the database, only about the format of the database. The format of the data dictionary is static. The data dictionary does contain metadata about its own structure, but its format is not something that can be modified easily.

The metadata in the data dictionary is designed to be manipulated. Rarely is the data dictionary manipulated directly. Instead, special stored procedures with complex validation logic are used to manipulate the system tables. Direct access to the system tables is dangerous, because even a small misstep could corrupt the data dictionary, leading to serious database problems.

The data dictionary defines the structure of the database, including specifying where physical files are stored on disk; the names of tables; column types and lengths; and the code for stored procedures, triggers, and views.

NoSQL Database Systems

In recent years, database systems have changed significantly due to the large collections of disparate data that organizations collect and process. Combine that with cloud-based computing, and the term "big data" has emerged to describe a construct where data coming from multiple disparate sources is processed, aggregated, and queried. While RDBMS systems are in no danger of being replaced, they are not easily scaled to this type of use. An individual or organization could access what appears to be a central database containing multiple types and sources of data, including data that is not easily contained in structured fields, such as pictures, voice files, document types, and so on. This data could come from not just dozens, but even hundreds of different sources, all stored on different types of systems all over the world, using many different types of database management systems. As a result, a database management system called NoSQL was devised to accommodate this new paradigm of massive data collection, processing, and use.

NoSQL allows for data to be queried and aggregated, but not necessarily stored in strict structures, such as tables, rows, and columns. It also allows for a wide variety of data types, even those that are not easily defined or that are fluid in structure. Because of this, NoSQL database systems can be difficult to design, construct, and secure. This is especially true when a lot of their underlying data comes from so many dissimilar sources and systems.

Even with all their differences, both relational database management systems and NoSQL systems use some of the same security principles. These include positive identification, strong authentication methods, strict authorization, encryption, database object permissions, restricted views, and so on. Because this type of database management system is a break in the old relational database paradigm, any auditor examining a NoSQL DBMS should become very familiar with the constructs and theories of NoSQL, as well as the particular conventions and structures of the specific system they are auditing.

Test Steps for Auditing Databases

Before you conduct the audit, you will need a few basic tools. You should have a checklist of the items you need to verify. You can create your own checklist, you can find checklists on the Internet, or you can even use the basic checklist we provide here. You should try to customize your checklist for the particular type of database system you are auditing and the objectives of the audit, as many RDBMS have unique elements and configurations that should be audited.

Start off by meeting with and discussing the audit with the database administrator (DBA). Clearly, the DBA is not going to be excited about the idea of being audited. Therefore, do your best to approach the DBA in as friendly a way as possible. Make sure that the DBA understands that you are there to help, not hinder, his or her work.

Databases are very often 24/7 systems, meaning they are not allowed any downtime. You'll encounter pushback on anything you want to do that could, with even the remotest possibility, affect database availability or performance. The first time you as the auditor bring down the database, your job becomes infinitely more difficult.

Be ready to optimize the time you will be accessing the system. Ensure that any account you are given on the system runs with only the permissions you need. Immediately after you are completed with any work, have the DBA lock the account. Don't delete the account—simply lock it until you are officially done with the audit. Then, if you do need to gather more information, the DBA can simply unlock the account rather than re-create it.

Perform as much work offline as possible. Ideally, you want to download the system tables, password hashes, files permissions, and all other information onto a local source. Then you can disconnect from the database and perform your audit steps offline with no risk of affecting the database. For instance, you want to ensure that you never do password-strength testing on the database; the password hashes can be downloaded, and password-strength testing can be done offline.

By showing the DBA this level of caution with the database, he or she will, hopefully, give you the professional courtesy of letting you do your job. Being at odds with the DBA can result in an audit that provides little value to the organization.

Now that you are equipped with some background on databases, we need a plan for performing an audit. Many of the steps covered here are almost identical to steps you would perform on an operating system or network audit, but they need to be placed in the context of the database. Some steps are unique to the database.

Initial Steps

1. Obtain the database version and compare it with your corporate policy requirements. Verify that the database is running a version the vendor continues to support.

Policies were written and approved to make an environment more secure, easily manageable, and auditable. Double-check basic configuration information to en-

sure that the database is in compliance with the organization's policy. Older databases increase the difficulty in managing the environment and increase the scope of administrator responsibilities as he or she attempts to maintain control over disparate database versions. Maintaining standard builds and patch levels greatly simplifies the process of managing the databases. In addition, many legacy databases run versions of database software that are no longer supported by the database vendor. This becomes a problem when a security vulnerability is released and the database cannot be patched because no patches for the older versions are available from the vendor.

It also helps to understand the security model required by the organization, which is further implemented by the system and database administrators. For example, the organization may implement a default deny model, so that by default all access to the database is denied except for permissions that are explicitly given. The opposite end of this is the default allow model, which by default allows all access to the database data and functions except where expressly prohibited. Most organizational security models fall somewhere in between those two extremes.

How

Through conversations with the DBA and review of your company's IT standards and policies, determine what database versions and platforms are recommended and supported by your company. Verify with the database vendor which versions and platforms are supported and whether patches for new security issues will be provided. Inventory the versions of the database that are run, and check for any databases that fall under the unsupported versions. Ideally, you want to keep the databases upgraded to supported versions.

Operating System Security

Other sections of this book are dedicated to operating system security, so we'll discuss it only briefly here. Start with the premise that a database not secured can be used to break into the operating system. Conversely, an unsecured operating system can be used to break into the database. Locking down one but not the other fails to provide proper security to either.

NOTE Refer to Chapters 7 and 8 for detailed steps on auditing the security of the operating system on which the database resides.

2. Ensure that access to the operating system is properly restricted.

The best situation is to have the operating system dedicated to the database only. No users other than system administrators should have access to connect to the operating system from a Secure Shell (SSH), Secure File Transfer Protocol (SFTP), or any other method outside the application. DBAs may also need some level of access to the operating system or even play the role of system administrator in some organizations. For most applications, users should not be able to connect to the database directly (that is, outside of the application). All updates to data should usually be performed via the application. Direct update of the data outside of the application could corrupt the database, and users usually need to have a good reason to update data outside of the application.

How

Verify with the administrator that all access to the core operating system is restricted to authorized system administrators only, which may include DBAs depending on organizational needs. Verify that any remote desktop or system shell access occurs over a secure protocol, preferably SSH. Check for any accounts on the operating system that should be removed.

3. Ensure that permissions on the directory in which the database is installed, and the database files themselves, are properly restricted.

Inappropriate access and updates to the database's underlying database files can result in massive disruption of the database. For example, any direct alteration via the operating system of the data files containing the actual database data will corrupt the database. Also, in Oracle, redo log files allow for recovery of uncommitted data in the event of a database crash, and control files are used by the database to do such things as locate the last redo log and locate the data files. Any direct updates of these files through the operating system could damage database functionality or prevent the database from being brought up. Each DBMS has its own specific startup, logging, and configuration files, and it is critical that these files be protected to ensure the ongoing availability and integrity of the database.

How

Verify that permissions on the directory to which the database is installed are as restrictive as possible and owned by the appropriate DBA account. Unfortunately, some database functionality was written without security in mind, and we can break database functionality by making file permissions too restrictive.

In Windows, similar measures should be taken. File permissions on the directory in which the database is installed should be limited to the permissions of the account the database runs under. Ensure that the "Everyone" or "Anonymous" users do not have any permissions on database files. In addition, make sure that all drives being used to store database files use NTFS.

In an ideal situation, even the DBA would not need permissions on the underlying operating system files. However, given the need for the DBA to work with database files and backups, patch the database, and accomplish other chores, the DBA may need some access to the operating system files. Any users, including system administrators, who do not need access to the operating system should not be granted permissions to it.

Retrieve a list of file permissions on all database files and the directories in which they reside, either by connecting to the operating system and pulling this information yourself, or by obtaining the information from the administrator. Review the listing to find any excessive privileges. For example, on Unix systems, check that permissions are set to be no more permissive than 770 to ensure explicit file owner and group permissions are applied. Setting tight security is the goal, but you may have to make exceptions to this policy. If you do need to make an exception, be sure to document the reasons for this. The best practice is to grant appropriate permissions only to the system administrators and DBAs who require access.

4. Ensure that permissions on the registry keys used by the database are properly restricted.

For database platforms running on Windows, you must properly secure the registry keys being used by the database. The registry keys are used to store configuration values that are important to the secure functioning of the database. Make sure that only the account under which the database runs has permission to edit, create, delete, or even view these registry keys.

How

Review the security permissions through the Registry Editor, through an acceptable utility approved by the system administrator, or by obtaining the information from the system administrator. After retrieving a complete list of the permissions, review it to ensure that no excessive permissions exist.

Account Management

Account management is a difficult activity because of the number of teams and individuals involved in requesting, approving, assigning, and removing accounts and access. Ensuring that the right users and accounts have access to critical systems and data makes this section of the audit particularly important.

5. Review and evaluate procedures for creating user accounts and ensuring that accounts are created only with a legitimate business need and appropriate approval. Also review and evaluate processes for ensuring that user accounts are removed or disabled in a timely fashion in the event of termination or job change.

Effective controls should exist for providing and removing access to the database, limiting unnecessary access to database resources. Some RDBMS may require creating unique accounts that are separate from the operating system or network accounts. Other database systems, such as Microsoft's SQL Server, rely on either machine-local accounts or Active Directory accounts, which are then added to the appropriate groups or given permissions to the database objects.

How

Interview the database administrator and review account-creation procedures. This process should include some form of verification that the user has a legitimate need for access. Ensure that access to DBA-level accounts and privileges are minimized.

Review a sample of accounts and evidence that accounts are approved properly prior to being created. Alternatively, take a sample of accounts and validate their legitimacy by investigating and understanding the job function of the account owners. Ensure that each user on the system has his or her own user account. No guest or shared accounts should exist. If a large number of database accounts exists, question the need. Application end users should generally be accessing the database through the application and not by direct database access.

Also review the process for removing accounts when access is no longer needed. This process could include a mechanism by which user accounts are removed on terminations and job changes. The process could include a periodic review and validation of active accounts by the system administrator and/or other knowledgeable managers. Obtain a sample of accounts, and verify that they are owned by active employees and that the access is still appropriate for their job duties.

Password Strength and Management Features

Many database platforms maintain their own authentication settings. Ensure that passwords and the authentication mechanism do not become the weak links in the chain.

Other database platforms integrate with the operating system or some other security subsystem to provide authentication. For instance, Db2 UDB does not maintain its own usernames and passwords, instead using the operating system or Resource Access Control Facility (RACF) for authentication. Microsoft SQL Server in Windows mode uses Windows authentication. This does not mean that users are not maintained in the database. Usernames, or groups that contain users, continue to be maintained in the database because there needs to be a mapping of the users to permissions and other database settings. However, the authentication happens at the operating system level instead of in the database.

Using integrated operating security for any of the database platforms has many pros and cons. Pros include the following:

- Operating system authentication typically is more robust than database authentication.
- Operating system authentication typically includes more password management features, which are more likely to be implemented already at the operating system level.

Cons include the following:

- Authentication is out of the DBA's hands.
- A user with an operating system account can access the operating system of the database if it is not configured properly.

6. Check for default usernames and passwords.

An essential item to audit for is default usernames and passwords. This continues to be an issue for databases. Many successful attacks against databases are executed using default usernames and passwords. Table 10-1 classifies these default usernames and passwords into a few categories. Literally thousands of these default passwords can be found on various security websites.

Category	Description
Default database password	Created in a standard database install. Can depend on the installed components of the database. Most of the latest versions of databases have eliminated default database passwords, but default passwords continue to be a serious concern in older versions of database software.
Sample or example passwords	Many samples, examples, and demonstrations of new or existing features are shown in SQL scripts that include creation of a test or sample account.
Default application password	When you install third-party products on top of a database, the products often install and run using a default username and password to access the database. These are known to hackers and serve as a common access route.
User-defined default password	When a new account is created, the password is often set to an initial value and then reset on first use. Problems arise when an account is created but never accessed. Ensure that passwords set on new accounts are random, strong passwords.

Table 10-1 Default Passwords

How

Verify that all default usernames and passwords have been removed or locked, or that the passwords have been changed. Free and commercial utilities and tools are available to verify this.

7. Check for easily guessed passwords.

Users often choose passwords that can be easily guessed by automated programs or clever hackers. The most common passwords used to be *password* and *secret*. Minimum password complexity standards may be enforced that require people to select more secure passwords, but it is still important to ensure that passwords cannot be found in a dictionary or easily guessed.

How

Run a password-strength test on password hashes to determine whether any passwords are easily guessed. This can be a complex task and should be executed by a security team or partner with sufficient skills in this activity. If you detect passwords that are found in a dictionary or can be guessed, talk with the DBA about user awareness practices and about implementing password-strength-checking practices. Refer to step 8 for system configuration settings that can help strengthen passwords.

8. Check that password management capabilities are enabled.

Many of the database platforms provide support for rich password management features. Oracle leads this area by including capabilities for the following features:

- Password-strength validation functions
- Password expiration
- Password reuse limits
- Password expiration grace time
- Password lockout
- Password lockout reset

If you do not configure these settings, they will not provide any additional security. By default, these features are not enabled.

How

Review the configuration values from the database. Ensure that each password management feature is enabled and configured for an appropriate value for the environment and in accordance with your company's policies. You will need to review the documentation for the database platform to determine the exact password management features available and the commands required to view them.

Permissions Management

Database privileges are slightly different from operating system permissions. Privileges are managed using `GRANT` and `REVOKE` statements. For instance, the following SQL statement gives `USER1` the permission to `SELECT` from the `SALARY` table:

```
GRANT SELECT ON SALARY TO USER1
```

The `REVOKE` statement is used to remove permissions that have been granted:

```
REVOKE SELECT ON SALARY FROM USER1
```

The `GRANT` statement can be used selectively to give permissions, such as `SELECT`, `UPDATE`, `DELETE`, or `EXECUTE`. This allows you to grant access to read the data in the table but limit the ability to modify the table. `GRANT` and `REVOKE` also can be used more selectively on a column-by-column basis.

9. Verify that database permissions are granted or revoked appropriately for the required level of authorization.

If database permissions are not restricted properly, inappropriate access to critical data may occur. Database permissions also should be used to restrict people from using subsystems in the database that may be used to circumvent security. Security best practices dictate that permissions should be granted on a need-only basis. If permission is not specifically needed by an account, it should not be granted.

How

Talk with the database administrator to determine which user accounts are required to have access to what data. Some administrators may need access, some accounts may be used by a web application to access the data, and some accounts may be used by batch jobs. Accounts that do not require permissions or access should be locked, disabled, or even removed.

10. Review database permissions granted to individuals instead of groups or roles.

Database best practices dictate that you should attempt to grant permissions to roles or groups, and those permissions, in turn, should be granted to individuals within those roles or groups. Use of roles or groups to allocate permissions ensures consistent levels of access are granted. When new permissions need to be granted, they can be granted to a single group rather than to multiple accounts. In addition, when a user changes jobs, it is straightforward to revoke the role or group and

grant new individuals access within the role or group.

How

Review the list of permissions from the database dictionary. Review for any permissions granted to an account or user. Check that privileges are granted to roles or groups. Individual users can then be granted permissions by assigning them to roles or groups as needed.

You also will need to download the list of roles/groups and users/accounts to determine which are allowed to be granted. The lists of users and groups are stored in the data dictionary.

11. Ensure that database permissions are not implicitly granted incorrectly.

Database permissions can flow from many sources. For instance, ownership of an object grants implicit full control over that object in a database. Privileges such as `SELECT ANY TABLE` allow access to all data and can lead to unauthorized access to data. If you do not have a complete understanding of how database permissions are implicitly granted, permissions may be granted in a way that was not intended.

How

Review the specifics of the permission model for the database platform and verify that permissions are inherited appropriately. Also review system privileges that allow access to data, such as `SELECT ANY TABLE`, or grant a privileged role to a user. Document permissions that are implicitly as well as explicitly granted to ensure that permissions are not allowed when they are not appropriate.

12. Review dynamic SQL executed in stored procedures.

Access to an object also can be gained by running stored procedures or functions. On Microsoft SQL Server, when executing code objects, access to any other objects owned by the stored procedure owner is allowed. On Oracle, running a stored procedure allows you to access objects as the stored procedure owner. This can be dangerous if stored procedures are not constructed properly and can be manipulated.

How

With the DBA's assistance, review stored procedures, specifically looking for use of dynamic SQL and evaluate how input is sanitized. Restrict use of dynamic SQL in procedures that run with elevated privileges. In addition, ensure that any and all access to stored procedures that run under elevated privileges is being logged.

In a large data warehouse environment, the auditor should work with the DBA and application owner to identify a sampling of critical paths and then look for dynamic SQL in stored procedures.

13. Ensure that row-level access to table data is properly implemented.

Relational databases are designed to grant permissions on a table or column. Unfortunately, they are not well designed to restrict access to a subset of rows in a table. When you grant a user `SELECT` privileges on a table, the user will be able to read every row in the table.

Several technologies can be used to help manage this problem. For instance, Oracle offers virtual private databases (VPDs) that you can use to limit access to specific rows. You also can use views programmatically to restrict rows returned based on the user's context. A common and practical approach is to use stored procedures to access tables. Using this strategy, the DBA does not need to grant permissions on the table, thereby preventing the user from attempting to circumvent the stored procedure.

How

This will likely be a joint effort between the DBA and application owner, particularly in larger environments. Discuss with the appropriate administrators the method of row-level access controls in the database. Ensure that a user cannot access data in a table without proper authorization if the user circumvents the application or stored procedure providing access. Access the database through a user's account to verify that the "effective" ability of the user is as intended.

14. Revoke PUBLIC permissions where not needed.

Each database has a slightly different implementation of a PUBLIC role—generically, it represents everyone in the database. Although the PUBLIC role cannot be dropped and you cannot add users to this role, you can assign permissions to it. If permissions are applied to the PUBLIC role, the permissions granted will apply to everyone.

The PUBLIC role should not be used to grant permissions, as these would be inherited by all users. You will want to validate that no extra permissions have been granted to the PUBLIC role.

Remember that if you revoke permissions from the PUBLIC role that are needed, you may end up breaking necessary functionality unless the permissions are granted to the explicit logins or appropriate groups that require permissions. Blindly revoking all PUBLIC permissions is a recipe for disaster.

How

Start by gathering a list of all permissions, highlighting those granted to PUBLIC. Discuss with the DBA which permissions are required. Then determine how much risk would be introduced by revoking permissions from objects that are clearly not needed. If everyone agrees to have the permissions revoked, it makes sense to revoke them. Always make a backup and provide an undo script that can be used to roll back any changes if you later determine that you need those permissions or something unexpectedly breaks.

Data Encryption

Data encryption is applied to three distinct areas, or states. *Data-in-motion* describes data in transit across the network and is often encrypted using protocols such as Transport Layer Security (TLS). *Data-at-rest* describes data resident on storage, such as inside a database, and can be encrypted with a number of algorithms such as the Advanced Encryption Standard (AES). *Data-in-use* describes data processing in applications. Note that encryption of data in use can be very tricky and is not often implemented because of the dynamic nature of processing data, which may be constantly changing. Data that is being used at the moment is not easily encrypted and decrypted in a fashion that is conducive for accessing it and changing it. However, any risks from not encrypting data in use can

be mitigated through strengthening other operating system and RDBMS security functions.

15. Verify that network encryption is implemented.

Network encryption serves two main purposes: to protect authentication credentials as they move across the network and to protect the actual data in the database as it moves over the network. The network is not a secure environment—IP addresses can be spoofed, and network traffic can be redirected and sniffed. It is critical that network traffic be encrypted not just over the external network but also on your intranet.

How

Verify that the network and client drivers have been configured to support encrypting network traffic using protocols such as TLS. Verify settings at both the client and the database. In some cases, you may need to sample the traffic to demonstrate the encryption.

16. Verify that encryption of data-at-rest is implemented where appropriate.

Encryption of data-at-rest involves encrypting data as it is stored in the database. Arguably, encryption of data-at-rest is more important than other forms of encryption, because the lifetime of data on disk or in the database is much longer than the lifetime of data on the network. As databases are replicated, copied, or backed up, encryption of data-at-rest provides an additional layer of protection, in that the data is not accessible without the proper key. Proper key management (access control, protection, and management of the encryption key) is a critical component of reviewing encryption at rest.

How

Verify that data that should be encrypted is encrypted properly. Also review the location where the encryption keys are stored and who is able to access or use the keys, because the strength of encryption relies on the strength of protection of the encryption keys. If the encryption keys are stored with the encrypted data, an attacker can subvert the security simply by extracting the encryption keys.

Check the disaster recovery plan to ensure that encryption key management is included as a component. A mistake you do not want your DBA to make is to implement encryption features but fail to include key management in the backup procedures. Failing to back up encryption keys properly results in the inability to recover a database backup.

Security Log Monitoring and Management

Regulations require that access to sensitive data be properly monitored. Regulations such as PCI, HIPAA, and Sarbanes-Oxley have had a significant and positive impact on companies that store sensitive data.

17. Verify the appropriate use of database auditing and activity monitoring.

Ultimately, regardless of whether an outside organization has mandated database monitoring, if the stored data is of significant business value, the database should probably have appropriate monitoring in place to identify malicious attacks and inappropriate use of data.

A number of methods can be used to monitor activity:

- Enabling native audit logging in the database
- Monitoring network traffic of audit database activity
- Reviewing transaction logs to build an audit trail from the database
- Enabling auditing of object access in the operating system

Each method has particular strengths and weaknesses. For instance, native audit logging is relatively inexpensive, because it is typically included with the database. Other solutions are more expensive but provide additional capabilities, such as context intelligence whereby an attack can be identified, which native auditing fails to provide.

How

Audit logging can take many forms:

- **Access and authentication auditing** Record logs that describe who accessed which systems, when, from where, and how.
- **User and administrator activity auditing** Record logs that describe what activities were performed in the database by both users and administrators.
- **Suspicious activity auditing** Identify and flag any suspicious, unusual, or abnormal access to sensitive data or critical systems.
- **Vulnerability and threat auditing** Detect vulnerabilities in the database and then monitor for users attempting to exploit them.
- **Change auditing** Establish a baseline policy for database configuration, schema, users, privileges, and structure and then find and track deviations from that baseline.

Review the implemented methods of event logging and monitoring with the DBA and discuss the sensitivity of the data. Log monitoring should align with the business value of the information stored in the database and with the policies and requirements of the organization.

Review a list of sensitive data in the database, and verify that appropriate auditing is properly enabled for sensitive data. Consider reviewing a list of sensitive transactions for a specific period of time to demonstrate the ability of the monitoring system to audit such events.

18. Verify that policies and procedures are in place to identify when a patch is available and to apply the patch. Ensure that all approved patches are installed per your database management policy.

Most database vendors have regularly scheduled patch releases. You must be prepared for the scheduled releases so that you can plan appropriately for testing and installation of the patches. If all the database patches are not installed, publicly known security vulnerabilities could exist on the database.

How

Interview the DBA to determine who reviews advisories from vendors, what steps are taken to prepare for the patches, and how long the patches are tested before being applied to the production databases. Ask to review notes from the previous patching cycle.

Obtain as much information as possible about the latest patches, and determine the scope of the vulnerabilities addressed by the patches. Compare the available patches with the patches applied to the database. Talk with the DBA about steps taken to mitigate potential risk if the patches are not applied in a timely manner. Many DBAs attempt to mitigate the need to patch by removing vulnerable components from the system if they are not necessary.

Databases pose an interesting dilemma with regard to patching for most organizations. Many databases run on a 24/7 schedule, so they have no allowance for downtime. This means that no time is available to bring down the database to apply the patches.

The other major complication for database patching is that testing new patches is typically a lengthy process. Databases typically are so critical that patches cannot be installed without thorough testing. Given a quarterly patch cycle, the DBA's full-time job easily could become testing and applying new patches, and this likely will become a full-time job for DBAs moving forward, just as today teams of people are sometimes dedicated to patching Windows and Unix systems.

One solution to the downtime problem has been the use of *clustering*. In a clustered environment, a single node in the cluster can be taken offline, patched, and brought back online. This can work, but it introduces complexity to the process. Regardless of the solution, patches related to control weaknesses must be understood and the control weaknesses must be appropriately dealt with to protect the database.

19. Determine whether a standard build is available for new database systems and whether that baseline has adequate security settings.

One of the best ways to propagate security throughout an environment is to ensure that new systems are built and configured in a consistent manner before moving into testing or production.

How

Through interviews with the system administrator and DBA, determine the methodology used for building, configuring, and deploying new systems. If a standard build is used, consider auditing a newly created system using the steps in this chapter.

NOTE Consider discussing an approval process for including the implementation and updating of standard builds into the organizational change management process and including a requirement for an auditor to review the changes and perform a full audit of new images. This is a great way for the audit team to create a working relationship with the database management team.

20. Evaluate how capacity is managed for the database environment to support existing and anticipated business requirements.

Technical and business requirements for databases can change quickly and frequently, driven by changes in infrastructure, business relationships, customer needs, and regulatory requirements. Inadequate database infrastructure places the business at risk of losing important data and may impede critical business functions.

How

Verify that capacity requirements have been documented and agreed to with customers. Review processes for monitoring capacity usage, noting when it exceeds defined thresholds. Requirements may be evaluated or captured in part by the same team responsible for the storage environment. Evaluate processes for responding and taking action when capacity usage exceeds established thresholds. Discuss the methods used to determine present database requirements and anticipated growth. Review growth plans with the administrator to verify that the hardware can meet the performance requirements, capacity requirements, and feature requirements to support infrastructure and business growth.

21. Evaluate how performance is managed and monitored for the database environment to support existing and anticipated business requirements.

Inadequate database infrastructure places the business at risk of losing important data and may impede critical business functions that need either more storage or better performance. Database performance is driven by several factors, including the physical storage media, communication protocols, network, data size, CPU, memory, storage architecture, encryption strategies, query structures, and a host of other factors. It is important to note that modern cloud databases can simplify performance-tuning efforts by enabling automated resource tuning and providing real-time query optimization suggestions.

How

Verify that regular performance reviews of the processor, memory, and IO/network bandwidth loads on the database architecture are performed to identify growing stresses on the architecture. Verify that performance requirements have been documented and agreed to with customers. Review processes for monitoring performance and noting when performance falls below defined thresholds. Evaluate processes in place for responding and taking action when performance falls below established thresholds. Discuss the methods used to determine present performance requirements and anticipated changes.

NOTE Reviewing capacity management and performance planning is a critical step in this audit. Ensure that the administrator has a capacity management plan in place and verify that performance needs are appropriate for the organization.

Tools and Technology

Although you can perform most of your audit using manual methods, you'll often

find it helpful to use a set of tools to perform repetitive or technical chores. Tools allow you to spend more time working on results instead of wrestling with the technical details. Auditing and monitoring tools can provide the raw materials that you need to analyze and interpret. This is the added value that a human auditor brings when using one of these tools.

Auditing Tools

Tools are useful for evaluating configuration issues, missing patches, account permissions, and other database environment settings. These tools can increase the speed of data collection for an audit; however, the auditor must be knowledgeable about how a tool works and how to understand the auditing results. If a tool is out of date or not designed for a particular audit purpose, use of the tool can cause more issues than provide benefit.

It's also important that you understand that network and operating system auditing tools fail miserably at helping with database audits. Why is this? Databases are complex beasts. They have their own access-control systems, their own user accounts and passwords, their own auditing subsystems, and even their own network protocols. Generic scanners simply do not have the expertise to provide more than a cursory look at the database.

A number of tools, such as the following, are specialized to help the auditor run audits on a database:

Database Auditing Tool	Website
AppDetectivePRO by Trustwave	www.trustwave.com
NCC Auditor and NCC Squirrel by NCC Group	www.nccgroup.trust
Microsoft SQL Vulnerability Assessment Tool	https://docs.microsoft.com/en-us/sql/relational-databases/security/sql-vulnerability-assessment?view=sql-server-2017
Oracle DBSAT	https://www.oracle.com/database/technologies/security/dbsat.html
CIS-CAT	https://learn.cisecurity.org/cis-cat-lite

Monitoring Tools

Many tools are designed to assist you with database activity monitoring. As an auditor, you have influence over the use of these tools to record and detect unauthorized or malicious access to sensitive data. You will need to determine what regulations apply to the database and then translate them into specific items that can be implemented as native auditing or more in-depth activity monitoring.

Database monitoring solutions include approaches that monitor the database passively by watching the network or by using a client installed on the host. Some monitoring solutions use a hybrid approach combining these two methods. IBM's hybrid solution, for example, maintains an impressive set of features and reports but requires an agent to work in conjunction with the Audit Vault server in a best-practices setup. Although IBM states that it does not significantly harm database performance, many DBAs are wary of auditing databases using a client and would rather use an appliance that watches traffic over the network. Recognizing this, IBM acquired Guardium in late 2009. The product uses a network appliance that watches database traffic transparently, monitoring transactions, security events, and privileged access, without placing a client on the database host.

Several tools provide technology for monitoring activity in the database:

Monitoring Tool	Website
DbProtect from Trustwave	www.trustwave.com
IBM Guardium	www.ibm.com
Data Activity Monitoring from Imperva	www.imperva.com/products/data-protection

Encryption Tools

Auditors also need to understand the tools available to meet database encryption requirements. Most modern databases have full database-level encryption features using Transparent Data Encryption (TDE). Additionally, several vendors provide more robust solutions in this area. The most innovative and impressive solution is probably from Thales because of their deployment and management model. The Thales Vormetric Data Security Platform deploys without any application coding or knowledge, and can simultaneously manage database and file encryption permissions integrated with your LDAP, such as the following:

Data Encryption Tool	Website
Vormetric	www.thalesesecurity.com
Database Protector from Protegrity	www.protegrity.com
Encryptionizer from NetLib	www.netlib.com
SafeNet product suite from Gemalto	https://safenet.gemalto.com/data-encryption/
IBM Guardium	https://www.ibm.com/in-en/marketplace/guardium-file-and-database-encryption

Knowledge Base

Database security information is not nearly as vast as that for network or operating system security. You can find enough detail to get the job done effectively, however.

Following is a list of books that can help you understand database security. If you do need to run an audit, you can review one of the books that applies to your specific database platform.

- *Oracle Security Handbook,* by Marlene L. Theriault and Aaron C. Newman
- *Oracle Security Step-by-Step*, by Pete Finnigan
- *The Database Hacker's Handbook,* by David Litchfield, Chris Anley, Bill Grindlay, and John Heasman
- *Implementing Database Security and Auditing*, by Ron Ben Natan
- *SQL Server Security*, by Chip Andrews, David Litchfield, Chris Anley, and Bill Grindlay
- *SQL Server Security Distilled*, by Morris Lewis
- *SQL Server Security: What DBAs Need to Know*, by K. Brian Kelley
- *Oracle Privacy Security Auditing,* by Arup Nanda and Donald Burleson
- *Effective Oracle Database 10g Security by Design*, by David Knox
- *Special Ops: Host and Network Security for Microsoft, UNIX, and Oracle,* by Erik Birkholz
- *MySQL Security Handbook*, by John Stephens and Chad Russell
- *Cryptography in the Database: The Last Line of Defense*, by Kevin Keenan
- *Database Security,* by Maria Grazia Fugini, Silvana Castano, and Giancarlo Martella

- *Database Security and Auditing: Protecting Data Integrity and Accessibility*, by Sam Afyouni

Many online technical guides are also available. These guides are often free, are up-to-date, and can be accessed from anywhere. Of course, they are also typically incomplete and not nearly as comprehensive as the books just listed.

Resource	Website
Oracle Database Security Guide, by Oracle Corporation	https://docs.oracle.com/cd/B19306_01/network.102/b14266/toc.htm
NIST Security Checklists	https://web.nvd.nist.gov/view/ncp/repository
DISA Checklists	https://public.cyber.mil/stigs/
ISACA Auditing Guidelines	www.isaca.org
Links to papers and presentations covering Oracle security	www.petefinnigan.com/orasec.htm
Oracle security website	www.oracle.com/technetwork/topics/security/whatsnew/index.html
Microsoft SQL Server security	https://docs.microsoft.com/en-us/sql/relational-databases/security/securing-sql-server?view=sql-server-2017
Center for Internet Security configuration benchmarks	https://www.cisecurity.org/cis-benchmarks/

As always, never forget the most up-to-date source of database security—Google. Simply search on any term of interest such as "Oracle Exploits" or "Auditing MySQL." Google provides a great list of resources to explore to help you do your job. Take care to verify sources of information to ensure they are authoritative and accurate. Try to avoid sites that may be designed by malicious entities to give false information and persuade you to change database security functions to a less secure state.

Master Checklist

The following table summarizes the steps listed herein for auditing databases.

Auditing Databases

Checklist for Auditing Databases	
❑	1. Obtain the database version and compare it with your corporate policy requirements. Verify that the database is running a version the vendor continues to support.
❑	2. Ensure that access to the operating system is properly restricted.
❑	3. Ensure that permissions on the directory in which the database is installed, and the database files themselves, are properly restricted.
❑	4. Ensure that permissions on the registry keys used by the database are properly restricted.
❑	5. Review and evaluate procedures for creating user accounts and ensuring that accounts are created only with a legitimate business need and appropriate approval. Also review and evaluate processes for ensuring that user accounts are removed or disabled in a timely fashion in the event of termination or job change.
❑	6. Check for default usernames and passwords.
❑	7. Check for easily guessed passwords.
❑	8. Check that password management capabilities are enabled.
❑	9. Verify that database permissions are granted or revoked appropriately for the required level of authorization.
❑	10. Review database permissions granted to individuals instead of groups or roles.
❑	11. Ensure that database permissions are not implicitly granted incorrectly.
❑	12. Review dynamic SQL executed in stored procedures.
❑	13. Ensure that row-level access to table data is properly implemented.
❑	14. Revoke PUBLIC permissions where not needed.
❑	15. Verify that network encryption is implemented.
❑	16. Verify that encryption of data-at-rest is implemented where appropriate.
❑	17. Verify the appropriate use of database auditing and activity monitoring.
❑	18. Verify that policies and procedures are in place to identify when a patch is available and to apply the patch. Ensure that all approved patches are installed per your database management policy.
❑	19. Determine whether a standard build is available for new database systems and whether that baseline has adequate security settings.
❑	20. Evaluate how capacity is managed for the database environment to support existing and anticipated business requirements.
❑	21. Evaluate how performance is managed and monitored for the database environment to support existing and anticipated business requirements.

CHAPTER 1

1 Auditing Big Data and Data Repositories

The rapid growth in the amount of data created and collected by systems and processes has become an important issue for businesses over the last decade. Technological innovations have brought the capability to store and analyze very large data sets into the mainstream. Companies use these capabilities for a variety of purposes from analyzing customer purchase patterns, to predicting equipment maintenance needs, to forecasting revenue. As more and more data is collected and stored, the need to audit the systems handling this data becomes more critical. The technologies in use vary, but many basic principles are common to different systems. This chapter covers the following:

- How to audit data repositories
- Specific audit considerations for big data environments

Background

Every organization has to contend with data, and for most businesses, the amount of data generated and stored has grown considerably over time. Data comes in a lot of different forms but is usually categorized into one of two major types: structured data, which refers to data organized neatly into a specific construct, usually for a database; and unstructured data, which usually refers to files of just about any kind. The more data you have, the greater the need for organization of that data into formal repositories for simplified access, data management, and analysis. Data repositories can take many forms and are often created for specific uses or for specific content types. Some major data repository types include:

- Databases, discussed in more detail in Chapter 10
- File servers, used to provide access to files for people or systems
- Wikis, a web-based content editing and sharing platform
- SharePoint, Microsoft's web collaboration platform
- Bitbucket, a web-based version control system
- Big data repositories such as Hadoop or Splunk

"Big data" refers to very large and often disparate data sets that are difficult or impossible to process with traditional data handling and analysis solutions. In the case of big data, "very large" can mean data sets up to the exabyte scale (1 exabyte = 1,000 petabytes = 1,000,000 terabytes), although the same technologies can be applied effectively to much smaller data sets. Big data may involve a combination of data sources, typically pulled together into a repository designed for consumption of large data sets.

Big data is widely described using three key characteristics, or "vectors." Dubbed the "three Vs" of big data, they were first outlined by Gartner in 2001 and describe the primary ways in which big data differs from traditional data. The three Vs are

- Volume: describes the large amount of data available or collected
- Velocity: describes the rapid speed at which data is generated or collected
- Variety: describes the diversity of data types collected

The term "big data" has been around since the 1990s, but the problem of how to handle data sets that seem to be overwhelming in scope is hundreds of years old and has frequently inspired innovative solutions. The amount of data collected during the 1880 U.S. Census took nearly the rest of the decade to analyze; to solve this problem for the 1890 census, Herman Hollerith invented a punch-card tabulation system that reduced the effort drastically. During the mid-twentieth century, scholars were increasingly concerned about the "information explosion" and noted that the information humans were creating and collecting was growing exponentially. The invention of the relational database in 1970 served to simplify the management and analysis of some of this data. Other technologies followed, including Crystal Reports in 1992, which brought insights from various data sources into a single report.

NOTE You may not know much about Herman Hollerith, but his company's legacy is still going strong today. Hollerith's company was part of a group that merged in 1911 to form the Computing-Tabulating-Recording Company, which in 1924 changed its name to International Business Machines (IBM).

With the expansion of the Internet in the 1990s and the decreasing costs of computing and storage, the growth of data available for research and analysis has continued unabated. New paradigms for digesting all of this data emerged in the early twenty-first century, driven in part by Google's indexing system for the Web. In 2004 Google published a paper describing their technology for parallel processing of large data sets, called MapReduce; the paper led to the development of the open-source Apache Hadoop framework, which is one of the major big data technologies in use today.

Big Data and Data Repository Auditing Essentials

With databases, file servers, big data systems, and a myriad of web-based technologies, there's a lot to consider in the data space. Auditing the entire landscape of data storage and data management in even a medium-sized environment would be a daunting task. Remember to scope your audit carefully with specific goals and target environments in mind. Even with a small scope, the size of a data platform can be intimidating; keep in mind that you can always break the audit down into smaller, more manageable steps.

You will want to familiarize yourself with the various data repositories in use in your business. Leverage your relationships with other IT teams, particularly operations-related groups, to help identify major data systems, both by prevalence and by criticality. Once you've identified the major repositories in play, you can begin to prioritize the list and determine where to focus your audit.

If you are auditing a data repository that includes file shares, you will need to understand what type of system is serving the files. This could be a Windows or Linux-based server, or a storage system configured to serve file shares. This will help you determine what type of operating system is in use. If the repository is a storage system, it's a safe assumption that the system uses some variant of the Linux operating system.

The "Knowledge Base" section later in this chapter contains links to reference information on several major repositories. The speed of change in this area is considerable, and you will no doubt encounter systems, applications, or terms with which you are completely unfamiliar. Don't let the alphabet soup of unusual names scare you; they are still applications running on some type of operating system, and much of a data repository audit involves determining how access to the data is managed and governed rather than details of specific repository technologies.

Test Steps for Auditing Big Data and Data Repositories

The basic steps here apply to most data repository environments, regardless of technology. Many of the audit steps involve interviews of system administration or application administration personnel. The final two suggested audit steps are specific to big data environments and deal with the flexible nature of those platforms.

1. Audit the OS-level controls relevant to the base operating system(s) included in the environment.

Most data repositories can be thought of as applications designed to manage, serve, and/or analyze data. Controls for the underlying operating system should be in place to ensure the infrastructure baseline is well managed.

How

Depending on the type of operating system in use, audit the system using the steps in Chapter 7 and Chapter 8. Keep in mind that some environments, particularly big data systems, may be composed of multiple servers with different operating systems.

If a file share is part of your audit, you will need some additional information about the file system before proceeding. File shares may be hosted on a Windows or Unix/Linux server, or could be configured as shares on a storage platform. If the data repository you are auditing involves file shares served from a storage system, you may want to include steps from Chapter 12. Most storage platforms use some variant of Linux as the core operating system, so many of the steps in Chapter 8 may also be applicable to storage systems.

2. Verify that the application has appropriate password controls and other authentication controls as appropriate. Also, determine whether default application account passwords have been changed.

Password concepts are well established; for example, passwords should be difficult to guess and should not be reused for other systems. Applications should also have safeguards preventing brute-force attacks. Where more sensitive data is at risk, stronger controls, such as multifactor authentication, should be considered.

Many applications, particularly those that are purchased, have default accounts with well-known default passwords. Many of these default accounts are used for system administration and therefore have elevated privileges. If those default passwords are not changed, there is increased risk that an unauthorized user may be able to access the application.

How

Verify appropriate password controls with the help of the developer or the application administrator and by reviewing your company policy. For example, three-digit PIN numbers probably are inappropriate for applications that store credit card data, and a 20-character password probably is overly paranoid for someone trying to access his or her voicemail. Ensure that the security mechanism requires users to change their passwords periodically (such as every 30 to 90 days). When appropriate, the security mechanism also should enforce password composition standards such as the length and required characters. Additionally, the security mechanism should suspend user accounts after a certain number of consecutive unsuccessful log-in attempts. This is typically as low as 3 and can be as high as 25 depending on the application, other forms of authentication required, and the sensitivity of the data. If the system integrates with a single sign-on or federation platform for authentication, determine whether any accounts are able to access the system outside of the federation mechanism. For example, administrative or system setup accounts may still be available and should follow appropriate password practices.

If your organization uses a multifactor authentication system, determine whether this is in use for the data repository in question. Multifactor authentication can provide an extra layer of defense for privileged account activities, such as system configuration or other administrative access.

Determine whether default accounts and passwords exist with the help of the developer or application administrator and by review of system documentation and Internet research. If they do exist, one of the easiest ways to determine whether they have been changed is to attempt to log on using the default accounts and passwords (or by asking the application administrator to attempt to do so).

NOTE Organizations working with large data sets may use other software or command-line scripts to work with the data. Authentication keys may be employed in some cases to authorize certain individuals or systems to authenticate without using a traditional account name and password. If authentication keys are used in your organization, work with the system administrators and security team to understand policies around key management and key rotation.

3. Ensure that the data classification of the environment is understood and review the data ownership process for the environment.

All data stored by or used by a data repository platform should be assigned a business owner, and this owner should classify the data (for example, public, internal use only, or confidential). This provides assurance that the data is being protected in alignment with its sensitivity.

How

Determine the business owner of the data contained within the system and ask for evidence that the data has been classified according to your company's data classification system. This classification should appear on any reports or transactions that display system data. Also, determine whether the application's access control mechanisms are appropriate based on the classification.

Consider that data repositories may contain various kinds of data from different sources. Some systems, such as file servers or SharePoint, may provide access controls and partitioning in such a way that data of different classifications may be stored on the same system but with separate logical access lists. The overall controls of the environment should be commensurate with the highest classification (most sensitive) of data stored within or managed by the system. The classification of a particular data area (such as an individual SharePoint site or a single file share) should be made known to end users of that system so that data of a more sensitive classification is not accidentally mixed with data of a lower classification.

Big data environments increase the complexity of this problem, as they are designed to combine multiple data sets. Depending on the purpose, the aggregated data or reports/views derived from that data may be of a higher sensitivity than the source data. This problem is discussed in more detail in step 10.

4. Review the system for the existence and use of role-based access controls and the processes for granting privileged access.

The system's security mechanism should allow for each system user to be given a specific level of access to the system's data and transactions. Administrator functions should be tightly controlled and not available to typical users. Without the ability to provide granular access based on user need, users will likely be granted unnecessary levels of access.

How

Employees should be given only the amount of access to the system necessary to perform their jobs. Review the environment with the developer or administrator and verify this functionality in the application. In other words, it should be possible to specify which data sets or files a user will be able to access. In general, it also should be possible to specify what level of access (such as display, update, and delete) the user will receive to those resources.

Evaluate the use of the administrator function with the administrator. The user of this function should have the ability to add, delete, or modify user access to the application system and its resources. The security mechanism should also provide the ability to control who has access to this administrator function. Obtain a list

of all employees who have been granted the administrator access level and review each for appropriateness. Also ensure that the system's security mechanism provides the system's security administrator with the ability to view who has access to the system and what level of access they have.

5. Review processes for granting and removing user access to view or search data. Ensure that access is granted only when there is a legitimate business need.

Users should have access granted and governed by the environment administrator to prevent unauthorized access to areas outside the user's intended scope. The system should have controls and processes in place to prevent users from having more access than is required for their roles. This step embodies the concept of least-privilege access.

The system should also have procedures to remove access when no longer needed. Poor deprovisioning processes may leave a user with inappropriate access to your application long after the access or authority should have been removed.

How

Review processes for requesting and approving access to data and resources. Ensure that these processes are documented and that they require approval from a knowledgeable administrator before access is granted to a user. Select a sample of users and ensure that user access was approved appropriately. Verify that the authorization mechanism is working as designed.

Verify that appropriate deprovisioning processes are in place with the administrator. Review the administrative processes for periodically reviewing user access lists and validating that the access is still appropriate. Automated suspension of accounts in the event of termination or job change is preferable to processes that require manual intervention.

For systems that have been in regular use for some time, select a sample of system users and validate that their access is still appropriate. Alternatively, if possible, select a sample of system users who have changed jobs or left the company, and ensure that their access has been removed.

Be sure to consider all this for administrators as well. Administrator access should be removed promptly in the event of a termination or job change.

6. Ensure that company search systems follow data permissions rules if repositories or reports are indexed by systems outside of the repository scope.

Many organizations use internal search capabilities to help employees find useful or necessary information. This step helps ensure that sensitive data within a data repository is not unnecessarily exposed by search systems.

How

Identify the administration team responsible for any company search systems that may index the repository. With the repository administrator and search administrator, discuss the controls in place to limit unauthorized access to indexed data. Ideally, a search system will not permit a user to access any indexed data in a way that bypasses the authorization model in place for that data.

Some data in data repositories may be considered sensitive and prohibited for internal search indexing. Many search engines provide configuration options that indicate to the search engine that a specific data source should not be crawled for indexing. Determine if the repository in question has been authorized for indexing. Some organizations may take an approach of "index all," whereas others may decide only to index specific data sources.

7. Assess data retention, backup, and recovery procedures.

Data should be archived and retained in accordance with business, tax, and legal requirements. Failure to do so could result in penalties and operational issues caused by the inability to obtain needed data.

Failure to back up needed data may severely disrupt business operations in the event of a disaster. A disaster could result in total loss of the application and its data with no ability to recover it. Recovery procedures and testing are necessary to ensure that the recovery process is understood and that it functions operationally as intended.

How

Determine whether critical data and software are backed up periodically (generally weekly full backups with daily incremental backups for the data) and stored offsite in a secured location. If cost-effective and appropriate, duplicate transaction records should be created and stored to allow recovery of data files to the point of the last processed transaction. Ensure that the backup schedule is in alignment with the recovery point objective (RPO) and recovery time objective (RTO) for the data in question.

Also ensure that any application code and algorithms used to analyze data are backed up and stored offsite in a secured location, along with any tools necessary for compiling and using the code.

Discuss with the system administrator and appropriate personnel to ensure that detailed recovery procedures are documented that define what tasks are to be performed, who is to perform those tasks, and the sequence in which they are to be performed. Testing of the recovery from backup sources using the documented recovery procedures should be performed periodically. Ensure that the recovery processes are in alignment with the RTO established for the environment.

NOTE To minimize redundancy, only the basics of auditing disaster recovery are included in this chapter. See Chapter 4 for additional details and ideas for auditing your application's disaster recovery capabilities.

Review data retention requirements with the administrator team. These requirements will vary based on the type of data and should be acquired from the appropriate departments within your company.

Big data environments can complicate retention, backup, and recovery procedures. Any retention rules driven by regulatory compliance for source data should also be in place for aggregated data. Companies may not have the capacity to create complete backups of big data systems, and the exercise of a recovery process could take considerable time. If your organization has decided not to back up a big data environment, this decision should be in alignment with all business groups needing to use this data.

8. Review controls surrounding configuration management.

This step addresses configuration management, the overarching concept of maintaining the proper configuration of the repository. Failure to maintain a baseline configuration may subject the repository to security or operational risks and may result in unpredictable performance.

How

Many data repositories are highly customizable and may be tuned during installation to meet the organization's needs. The final configuration, which may include access settings, security controls, server names, IP addresses, and other information, should be protected from unauthorized change. Review configuration change controls with the system administrator to ensure that changes are documented and executed according to company policy. Ensure that logs generated by the system include documentation of changes to system configuration.

In addition, vendor or third-party guidelines may exist that define ideal or expected configuration guidelines for a secure environment. Determine whether these exist for the repository in question. If so, discuss with the environment administrator to see if the repository's configuration has been validated against known guidance.

In more risk-averse organizations, stronger controls could include keeping a copy of configuration options in a separate location and periodically verifying that the production configuration has not changed.

9. Review and evaluate procedures for monitoring and maintaining the security of the system.

If the system administrator doesn't monitor his or her systems for unexpected changes, security incidents could occur without his or her knowledge. By monitoring, we mean *actively* reviewing log data and system information. Merely enabling log collection without reviewing the resulting data is just barely preferable to having no log data at all.

System security must also be *maintained*. The world of security vulnerabilities is an ever-changing one, and it is unrealistic to believe that a static audit program can provide assurance of ongoing system security. A vulnerability scanning tool that is updated frequently can provide an effective mechanism for understanding the current security state of the machine. In addition, if the system administrator has a security patching process in place, this scan will provide some validation of the effectiveness of that process.

How

Interview the environment administrator and review any relevant documentation to get an understanding of security monitoring practices. Some level of monitoring is important, but the monitoring level required should be consistent with the criticality of the system and the inherent risk of the environment (for example, a repository of employee personal data should have more robust security monitoring than a wiki holding conference room instructions). The system administrator is responsible for monitoring his or her hosts not only for operational parameters but also for security issues; however, in larger organizations security monitoring may be delegated to dedicated security teams.

If security monitoring is performed, assess the frequency of the monitoring and the quality with which it is performed. Look for evidence that the security monitoring tools are actively used. Review recent alerts and determine whether they were investigated and resolved.

10. Review governance processes for adding data sources to the big data environment.

As different teams explore the power of big data tools, they are likely to want more and more data made available in the environment. Uncontrolled data growth can lead to risks, and combining some previously separated, less-sensitive data elements could result in a new, combined data set with a higher sensitivity.

How

Review the steps taken when a new data source or element is requested to be added to the big data environment. Additions should require the approval of a responsible delegate who understands the risks associated with combined data sources or who can identify appropriate individuals to consult on potential risks.

The impact of a data addition must be understood in terms of the system capacity, performance, security, and data classification. While capacity and performance are commonly addressed by operational teams, security changes and data classification are often missed.

Depending on the access model in use, the addition of sensitive data could create a situation where data is exposed to unauthorized personnel merely because their access was not verified or because they had overly broad access to begin with. To simplify access management for some environments, organizations may designate certain individuals with permission to access *all* of the data in a big data system. As long as every data addition is considered and formally approved with this in mind, this is an acceptable arrangement. Data should not be imported into an environment until the data access model is verified.

The addition of some data to an environment may also give rise to new data sets not previously considered by the organization. This could create a data classification problem if the resulting data set is more sensitive than either of the input sets. The approval process for adding data to an environment should consider whether the classification of the resulting aggregate data set should change.

11. Ensure that credentials or other methods used to load remote data sources are properly secured.

Big data systems commonly pull data from multiple sources using automation such as scripts. The credentials used by these scripts, if not properly secured, could be used for unauthorized access.

How

Discuss the flow of data in and out of the big data system with the administrator or architect. Assess where scripts or other automation processes are used to pull data into the environment. Ask an administrator to show you relevant scripts and determine how credentials for the target data sources are handled. If passwords are stored in plaintext in a script, ensure that the script is stored in a password-controlled area and accessible only by authorized personnel. If copies of scripts are stored in version control systems, ask to review those copies to ensure passwords are not also stored in version control systems or other script repositories. It is preferable for passwords to be referenced in variables and stored only on the

automation system in tightly controlled configuration files rather than be present in scripts themselves.

Knowledge Base

Resource	Website
Microsoft SharePoint (commercial)	https://docs.microsoft.com/en-us/sharepoint/sharepoint-server
MediaWiki (open source)	www.mediawiki.org/wiki/MediaWiki
Atlassian Bitbucket (commercial)	https://bitbucket.org/product
Big Data	https://en.wikipedia.org/wiki/Big_data
Apache Hadoop & Big Data 101 (video by Cloudera)	www.youtube.com/watch?v=AZovvBgRLIY
Apache Hadoop (open source)	hadoop.apache.org
Splunk (commercial)	www.splunk.com
Elasticsearch (open source and commercial)	www.elastic.co

Master Checklist

Checklist for Auditing Big Data and Data Repositories
❑ 1. Audit the OS-level controls relevant to the base operating system(s) included in the environment.
❑ 2. Verify that the application has appropriate password controls and other authentication controls as appropriate. Also, determine whether default application account passwords have been changed.
❑ 3. Ensure that the data classification of the environment is understood and review the data ownership process for the environment.
❑ 4. Review the system for the existence and use of role-based access controls and the processes for granting privileged access.
❑ 5. Review processes for granting and removing user access to view or search data. Ensure that access is granted only when there is a legitimate business need.
❑ 6. Ensure that company search systems follow data permissions rules if repositories or reports are indexed by systems outside of the repository scope.
❑ 7. Assess data retention, backup, and recovery procedures.
❑ 8. Review controls surrounding configuration management.
❑ 9. Review and evaluate procedures for monitoring and maintaining the security of the system.
❑ 10. Review governance processes for adding data sources to the big data environment.
❑ 11. Ensure that credentials or other methods used to load remote data sources are properly secured.

CHAPTER 12

Auditing Storage

This chapter covers auditing storage and begins with an overview of common storage technologies. The storage audit combines the concerns of the platform and the data. The platform has similar control requirements as those found in a server. The data has unique control requirements because of the necessity to keep appropriate controls in place for different classes of data. This chapter covers the following:

- A brief technical overview of storage
- How to audit the storage environment
- Tools and resources for enhancing your storage audits

Background

Storage extends the boundaries of the computing environment to allow data to be shared among users and applications. Storage platforms have grown so efficient that servers can use network-based storage platforms, as opposed to the storage native to the server and other forms of direct attached storage, for their primary storage requirements. Figure 12-1 illustrates the consolidation of data management across the data center to fewer points, simplifying management overhead with the use of shared storage.

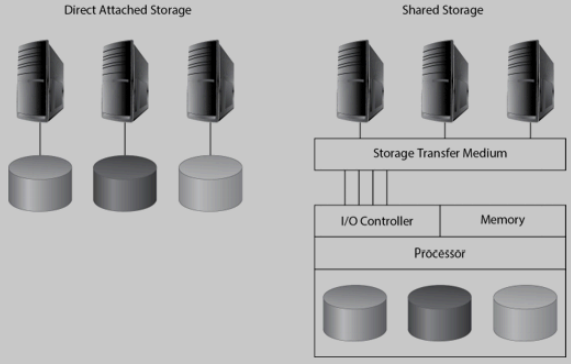


Figure 12-1 Consolidated storage architecture

The storage environment continues to evolve, as traditionally disparate technologies and storage platforms are combined into a single unit that manages both file data and application data within the same unit. Protocol smart switches capable of moving data at blistering speeds have broken bottlenecks to consolidating environments, which in turn enables downsizing of the data center. Add to this mix cool technologies such as data deduplication, storage virtualization, and solid-state drives, and it's easy to see why good storage administrators are in high demand.

Storage Auditing Essentials

To understand the material in this chapter, you need to understand the basic components that make up the storage environment. Your role as an auditor and advisor will significantly improve if you understand the major technology trends that challenge traditional storage models.

Key Storage Components

Storage infrastructure includes components associated with the host, network, and storage that work in conjunction to provide storage facilities to users and applications.

Redundant Array of Independent Disks (RAID)

RAID storage techniques allow multiple drives to be combined to provide more storage options than would be provided by a single disk, including more capacity, redundancy, and performance. The storage controller manages multiple drives in one of several configurations classified as RAID *levels*.

RAID-0: Striping Striping is a technique that offers the best performance of any RAID configuration. In a striped array, data is interleaved across all the drives in the array. If a file is saved to a RAID-0 array, the array distributes the file across the logical drive composed of multiple physical disks. In Figure 12-2, the file would span across all six disks. From a performance perspective, RAID-0 is the most efficient because it can write to all six disks at once. The drawback to RAID-0 is its lack of reliability. Any single disk failure results in the loss of all of that data stored in the array.

Figure 12-2 RAID-0: Striping across six disks

RAID-1: Mirroring RAID-1 is a disk array in which two disks are maintained as identical copies. The disks are mirrored to each other to protect against a drive failure. With mirroring, whatever you write to one drive gets written simultaneously to another. Thus, you always have an exact duplicate of your data on the other drive, as shown in Figure 12-3. RAID-1 is the most reliable of the RAID disk arrays because all data is mirrored after it is written; however, you can use only half of the total storage space on the disks. Although this may seem inefficient, given that storage media these days is both plentiful and reasonably cheap, RAID-1 is the preferred choice for data that requires the highest possible reliability.

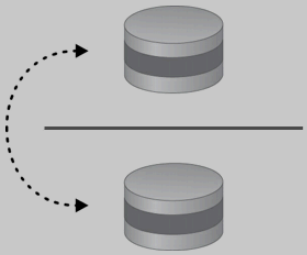


Figure 12-3 RAID-1: Mirroring

RAID-5: Reliability with Parity RAID-5 is a striped disk array, similar to RAID-0, in that data is distributed across the array; however, RAID-5 also includes data about the contents of the drives called parity. With parity, a mechanism maintains the integrity of the data stored in the array, so that if one disk in the array fails, the data can be reconstructed from the remaining disks. Parity is used to reconstruct data on a drive that has failed. RAID-5 is shown in Figure 12-4.

Figure 12-4 RAID-5: Reliability with parity

RAID-5 is a reliable storage solution. The RAID controller adds a parity byte to all binary information written to the array. These parity bytes add up to either an even or odd number. The controller can determine whether the information has been compromised in any way. If it has, it can replace the data automatically.

RAID-10: High Performance Striping with Mirrored Segments RAID-10 is implemented as a RAID-0 (striped array) whose segments are RAID-1 (mirrored) arrays. The result delivers high performance by striping RAID-1 segments and provides the same fault tolerance as RAID-1. The number of drives makes RAID-10 more expensive than other solutions, but the relatively low expense of storage media these days can make this a viable option. RAID-10 also comes with a high overhead. RAID-10 might be used to support a database server requiring high performance with fault tolerance. RAID-10 is shown in Figure 12-5.

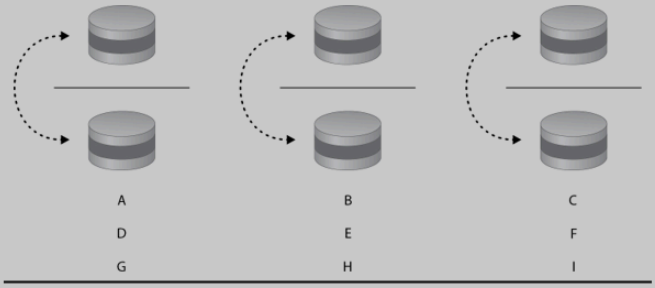


Figure 12-5 High-performance striping with mirrored segments

There are other RAID levels as well, but these are the most popularly used. Table 12-1 offers a summary of common RAID levels.

Level	Techniques	Description	Pros/Cons
RAID-0	Disk striping	Data is distributed in stripes that are sent to each disk in the array.	Best performance; no fault tolerance
RAID-1	Disk mirroring	Data on one drive is mirrored on another.	100% redundancy of data Slower performance and 50% loss of storage space
RAID-5	Block-level striping with distributed parity	Data and parity are striped in blocks across all disks.	High read data transaction rates Complex controller design
RAID-10	Disk striping and mirroring	Striped array's segments are RAID-1 arrays.	Offers redundancy along with high performance Common for high I/O databases

Table 12-1 Common RAID Levels

DAS, NAS, SAN, and CAS

Direct attached storage (DAS) is storage directly attached to the server by connectivity media such as parallel Small Computer System Interface (SCSI) cables. The media can either be internal drives or a dedicated RAID or JBOD (just a bunch of disks). This type of storage is the most limited and doesn't allow for the efficiencies that the other types of storage offer, because DAS is not networked. The storage can be shared by the host; however, file access performance will depend on the host system resources, and sharing can have an impact on the overall performance of the host system.

A *network attached storage* (NAS) device runs an operating system specifically designed to handle files and make them accessible to the network. NAS is also known as file storage and is often accessed by users and applications as mapped drives. Common protocols used in a NAS include Network File System (NFS) for Unix operating systems and Common Internet File System (CIFS) for Microsoft operating systems. Common NAS vendors include Dell EMC and NetApp.

A *storage area network* (SAN) is a scalable and flexible storage subsystem generally available to more than one host at the same time. The SAN operates using unique block-level communication protocols that require special hardware to work properly. The SAN comprises specialized devices such as host bus adapters (HBAs) in the host servers, switches that help route storage traffic, and disk storage subsystems that understand how to manage the special protocols required for SAN storage. Common protocols used in a SAN include SCSI and Fibre Channel (FC).

Table 12-2 compares SAN and NAS. Common SAN vendors include Dell EMC, Hitachi, and IBM.

Comparison	Storage Area Network	Network Attached Storage
Storage type	Block based	File based
Protocols	SCSI, iSCSI, HyperSCSI, Fibre Channel, ATA over Ethernet (AoE)	NFS (Unix), CIFS (MS), HTTP
File sharing	Large amounts of data	Easier file sharing
Traditional cost	$$$$	$$
Performance	High performance	Lower throughput
Access	Usually abstracted by a file system or database management system for use by applications and end users	Directly useable by end users as a "file share" or applications that don't need the throughput of SAN

Table 12-2 Comparison of SAN and NAS

The trend among storage vendors is to collapse as much complexity as possible into fewer components that have the capability to handle the functionality traditionally split among different product lines. For example, Dell EMC has gone to great lengths to embed NAS and SAN technologies into the same storage array, allowing the same chassis to perform functions that traditionally were split among NAS and SAN. The marketed long-term result is a solution that has a lower total cost of ownership, smaller footprint, and lower operating costs. Be aware of these and other trends that are discussed under "Key Storage Concepts."

Content addressed storage (CAS) is object-oriented storage designed specifically for archival storage of unique items that are not intended to be changed after they are stored. CAS is common for medical images and archival data for retention purposes. EMC coined the name CAS with its now-defunct Centera archive product, which could be set up to allow data to be written to the storage and never to be deleted, preventing malicious or unintentional deletion of archived data.

Key Storage Concepts

The following are important storage concepts that are gaining momentum. Storage is changing—permanently—to become smarter, faster, smaller, and more efficient.

Recovery Point Objective and Recovery Time Objective

Recovery point objective (RPO) determines how much data you will lose should an incident occur. Recovery time objective (RTO) determines how long it will take to recover data should an incident occur.

Tiered Storage

The *cost* of storage media is proportional to the *performance* of the media. Flash storage is the fastest media, and it's perfect for extreme performance environments. However, the cost (and some quirky side effects related to capacity decay over time) makes flash a poor choice for archiving data for long periods. You can buy different types of storage media for the same massive storage environment and classify your storage according to performance requirements. The storage array can then put data on the appropriate media for the performance that's required. Several different data-tiering models can be used.

Data Deduplication

Data deduplication technologies find duplicates at some level, whether identical files or identical components, blocks, streams, or sequential bits, and substitute duplicate copies with a pointer to a single copy of the data. For example, consider an e-mail sent to ten people with an attachment. Either the attachment can be saved ten times, or the attachment can be saved once with nine pointers to the original copy.

Several methods, technologies, and vendors can be used for deduplicating and reducing the size of the data stored. They vary greatly in terms of overhead, complexity, deployment, and effectiveness. Some, such as Dell EMC's Avamar, identify redundant data at the source, minimizing backup data before it is sent over the LAN/WAN. Other solutions are placed next to the storage target to identify and manage redundant data at the target. Regardless of the solution, data deduplication is an important strategy component for effective capacity utilization. The results are reduced capacity requirements, smaller footprints, and reduced operational costs.

Green Storage

The objective of green storage is ostensibly to conserve resources and the environment, but it also lets you get more out of your storage infrastructure for less money. Using smart technologies and architectures, companies are shrinking the amount of space, equipment, energy, and administrative overhead required to manage storage.

Business needs and compliance requirements drive the need for redundancy. Many companies store dozens of times more data than they actually use. RAID-10, backups, development, snapshots, overprovisioning, compliance archives, and disaster recovery sites continue to increase the amount of storage used. The good news is that technologies and architecture decisions can drastically reduce this number, cutting in half the amount of storage required.

The use of storage virtualization, compression, thin provisioning, nonmirrored RAID, deduplication, and resizable volumes can help reduce the storage footprint in a data center. This in turn reduces the energy requirements. Going green can also save a company operational and administrative costs of maintaining and managing storage. These discussions should be part of your capacity planning interview with the administrator.

Cloud-Based Storage

Storage solutions are not only bound to the physical network infrastructure of the organization. With the push to move many services to "the cloud," such as applications and even infrastructure, cloud-based storage has also become a common practice. Cloud-based storage is typically implemented through a service agreement with a cloud provider, including some of the larger players in the industry such as Amazon Web Services and Microsoft Azure. The organization relies on the infrastructure of the service provider for performance, storage capacity, and even security, to a degree.

The advantage of this storage model is that the organization does not have to dedicate equipment, personnel, or infrastructure to the storage platform. They simply contract this service out to the provider and pay the fees. The provider has the burden of providing acceptable performance, access, and capacity. This removes a lot of overhead management headaches from the organization and its users.

The disadvantages of this model are that the organization may exercise less control over the performance, availability, and security levels of the storage solution, and the organization is at the mercy of the provider and its policies, depending on how the service contract is written. Issues also could surface with data ownership and privacy, especially when the organization stores protected data (healthcare, financial, or personal data) in the cloud. These range from limiting the access to this data to only authorized persons, to liability in the event of a data breach.

Storage Virtualization

Traditional storage requires heavy administrator overhead and detailed knowledge of physical paths, device information, and data locations. Storage virtualization is an abstraction of detail that separates layers between the host's needs and the storage. Location and implementation are transparent to the host and appear to users as simple, unified file shares or mapped drives in most cases. Virtualized storage can be implemented using a combination of physical storage (e.g., SAN, NAS, attached storage) at an organization's site, or even in the cloud. The result is improved delivery and quality (uptime) of the storage infrastructure while increasing utilization and reducing capital cost and management overhead. Storage virtualization is a broad and somewhat complex topic because of the many vendors and implementations available, but you should be aware of the architecture. Storage virtualization has become a mainstay of any larger network requiring a large capacity of data storage space.

Test Steps for Auditing Storage

The following storage audit is designed to review critical controls that protect the confidentiality, integrity, and availability of storage for the supported systems and users that rely on the storage. Dozens of storage vendors—from Dell EMC, to Hitachi, to NetApp—cover every vertical of the market. Each of the steps that follow applies to some extent; however, use your judgment to determine the depth to which you decide to take any one step. For example, an auditor reviewing high-performance storage supporting a business-critical web application might spend more time asking questions and reviewing vendor-specific analysis output that verifies the storage has the capacity and performance necessary to handle peak loads. Note that cloud-based storage providers should be subjected to these audit steps as well; however, evaluation of their policies, procedures, and processes may be dependent upon the service agreement and may not be negotiable with the provider. In that event, you should at least gather all of the available information for each of these steps as it applies to the cloud provider and determine if it meets your performance, capacity, availability, and security requirements.

Initial Steps

1. Document the overall storage management architecture, including the hardware and supporting network infrastructure.

The team responsible for managing storage should maintain documentation illustrating the storage architecture and how the storage interfaces with the rest of the environment. This information should include data covering supported systems and the connecting network infrastructure, as well as any documentation from cloud storage providers. This information will be used by the auditor to help interpret the results of subsequent audit steps.

How

Discuss and review existing documentation with the administrator.

2. Obtain the software version and compare it against policy requirements.

Review the software version to ensure that the host is in compliance with policy. Older software may have reliability, performance, or security issues and increases the difficulty in managing the storage platforms. Additionally, disparate software versions may increase the scope of administrator responsibilities as he or she attempts to maintain control over the different versions running on the storage platforms.

How

Work with the administrator to obtain this information from the system and review vendor documentation. Ensure that the software is a version the vendor continues to support and does not contain widely known and patchable vulnerabilities that would bypass existing controls. Additionally, verify that the current running version does not contain performance or reliability issues that would affect your environment. Review any mitigating factors with the administrator, such as issues that have not been fixed but are not applicable to the environment.

3. Determine what services and features are enabled on the system and validate their necessity with the system administrator.

Unnecessary services and features increase the risk of exposure to misconfigurations, vulnerabilities, and performance issues and complicate troubleshooting efforts.

How

Today's storage systems range from the very simple to the extremely complex. Work closely with the storage administrator to discuss enabled services and their applicability to the environment. Review and evaluate procedures for assessing vulnerabilities associated with necessary services and keeping them patched.

Account Management

4. Review and evaluate procedures for creating administrative accounts and ensuring that accounts are created only when there's a legitimate business need. Also review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.

Effective controls should govern account creation and deletion. Inappropriate or lacking controls could result in unnecessary access to system resources, placing the integrity and availability of sensitive data at risk.

How

Interview the system administrator and review account-creation procedures. This process should include some form of verification that the user has a legitimate need for access. Take a sample of accounts and review evidence that they were approved properly prior to being created. Alternatively, take a sample of accounts and validate their legitimacy by investigating and understanding the job function of the account owners.

Review the process for removing accounts when access is no longer needed. This process could include a semiautomatic process driven by the company's human resources (HR) department providing information on terminations and job changes. Or the process could include a periodic review and validation of active accounts by the system administrator and/or other knowledgeable managers. Obtain a sample of accounts and verify that they are owned by active employees and that each employee has a legitimate business requirement for administrative access.

5. Evaluate the process and policies used for granting and revoking access to storage.

Written policies should govern the process used to create new storage allocations, including approval processes and procedures for setting up the new work area and the users who should have access to the new storage allocation. Policies or procedures should also exist for "cleaning up" or removing rights that are no longer needed when a project is completed. Failure to manage storage allocation could unnecessarily use storage capacity, and failure to govern rights management may allow users who should no longer have access to storage to maintain inappropriate levels of access.

How

Discuss policies and procedures for granting and revoking access to workspaces with the storage administrator.

Storage Management

6. Evaluate how capacity is managed for the storage environment to support existing and anticipated business requirements.

Technical and business requirements for storage can change quickly and frequently, driven by changes in infrastructure, business relationships, customer needs, and regulatory requirements. Inadequate storage infrastructure places the business at risk of losing important data and may impede critical business functions that need more storage.

How

Verify that capacity requirements have been documented and that customers have agreed to them. Review processes for monitoring capacity usage and noting when it exceeds defined thresholds. Evaluate processes in place for responding and taking action when capacity usage exceeds established thresholds. Discuss the methods used to determine present storage requirements and anticipated growth. Review growth plans with the administrator to verify that the hardware can meet the performance, capacity, logging, and feature requirements to support infrastructure and business growth.

Business drivers may affect the storage infrastructure design and architecture:

- Retention requirements may change because of new compliance drivers.
- Business continuity and disaster recovery plans may require faster response times and less data loss.
- Virtualization projects may require more storage.
- New high-performance databases may demand tiered storage technologies as you add faster storage media or network connections to support high-performance business requirements.
- Growing backup needs over strained networks might require a data deduplication solution to minimize the impact to the network.

7. Evaluate how performance is managed and monitored for the storage environment to support existing and anticipated business requirements.

Storage performance is driven by several factors, including the physical storage media, communication protocols, network, data size, CPU, memory, RAID architecture, data-tiering strategies, and a host of other factors. Inadequate storage infrastructure places the business at risk of losing important data and may impede critical business functions that need either more storage or better performance.

How

Regular periodic performance reviews of the processor, memory, and bandwidth loads on the storage architecture should be performed to identify growing stresses on the architecture. Verify that performance requirements have been documented and that customers have agreed to them. Review processes for monitoring performance and noting when performance falls below defined thresholds, as well as the action that should be taken. Discuss the methods used to determine present performance requirements and anticipated changes.

NOTE Reviewing capacity management and performance planning are two of the most critical steps in this audit. Ensure that the administrator has a capacity management plan in place, and verify that performance needs are appropriate for the organization.

8. Evaluate the policies, processes, and controls for data backup frequency, handling, and remote storage.

Processes and controls should meet policy requirements, support business continuity/disaster recovery (BC/DR), and protect sensitive information. Data backups present monumental challenges for organizations, particularly when it comes to the central data repositories in the organization, namely the databases and storage platforms. Vendors offer several solutions to manage the frequency, handling, and remote storage of data and system backups. The implemented solution mix should be appropriate to meet the stated goals of the BC/DR plans.

How

Review policy requirements for meeting RPOs, which affect how much data might be lost from a disaster, and RTOs, which affect how long it will take to restore data after a disaster occurs. The RPOs and RTOs should be aligned with the BC/DR programs.

Encryption and Permissions Management

9. Verify that encryption of data-at-rest is implemented where appropriate.

Encryption of data-at-rest involves encrypting data as it is stored. This step isn't appropriate for all environments and may be covered by other controls or applications. Encrypting data-at-rest protects against the loss of theft of storage devices and backup media and, in certain configurations, can act as an additional layer of access control because the application or user accessing the data must have access to the encryption keys.

How

Verify that data that should be encrypted is done so properly. Additionally, review the location where the encryption keys are stored, because the strength of encryption relies on the strength of protection of the encryption keys. If the encryption keys are stored with the encrypted data, an attacker can subvert the security simply by extracting the encryption keys.

Check the disaster recovery plan to ensure that encryption key management is included as a component. A mistake you do not want your administrator to make is to implement encryption features but fail to include key management in the backup procedures. Failing to back up encryption keys properly may result in the inability to recover a backup.

10. Verify that network encryption of data-in-motion is implemented where appropriate.

Policy requirements may require encrypted traffic for applications that contain sensitive information or for backing up storage to another location. Network encryption is implemented for two main reasons: to protect authentication credentials as they move across the network and to protect the actual data as it moves over the network. The network is not a secure environment—IP addresses can be spoofed, and network traffic can be redirected and sniffed.

How

Review policy requirements with the administrator and determine if any data is required to be encrypted in transit. For example, verify that network traffic used to back up or replicate sensitive data is encrypted.

11. Evaluate the low-level and technical controls in place to segregate or firewall highly sensitive data from the rest of the storage environment.

Controls should exist that restrict access to sensitive information, such as cardholder data (CHD), personally identifiable information (PII), source code, and other types of proprietary data, including administrative rights to the host. Sensitive information may be placed on isolated networks that require additional secure connections, such as using a VPN or bastion host to access the environment. Access control lists protecting file shares, database tables, and other repositories should be reviewed. Storage volumes on SANs should be configured so that only the authorized hosts can attach to the storage. Logical network isolation may be used so that firewall rules determine who can access a system and/or where a user can access the system from. If encryption is used, describe it here and evaluate the handling of keys, including the granting and revocation of rights, keys, and certificates. Other controls you should examine include identification and authentication methods (e.g., passwords, multifactor authentication, and so on), and even data loss prevention (DLP) technologies might be used to control the movement of sensitive data within an environment.

How

Review controls in place with the storage administrator to separate sensitive data. Review auditing and log management procedures for administrative access to the storage environment that could bypass intended controls. Consider compensating controls such as data encryption in the environment specific to an application. Identify technical and administrative controls that force separation between sets of data. Strong controls will prevent comingling of disparate data types and create actionable, nonrepudiated logs when these controls are bypassed.

Security Monitoring and Other General Controls

12. Review and evaluate system administrator procedures for security monitoring.

The storage administrator should regularly monitor the environment for changes and review the environment for security vulnerabilities. A poor monitoring program could allow security incidents to occur without the administrator's knowledge. By *monitoring,* we mean actively watching for issues (detection) and actively searching them out (finding and mitigating vulnerabilities).

How

Interview the system administrator and review relevant documentation to gain an understanding of security monitoring practices. Several methods of security monitoring may be performed. The level of monitoring should be consistent with the criticality of the system and the inherent risk of the environment. (For example, a storage environment supporting critical financial data should have robust security monitoring.) The system administrator is responsible for monitoring the environment to identify activity and trends that might indicate critical issues.

If security monitoring is performed, assess the frequency of the monitoring and the quality with which it is performed. Look for evidence that the security monitoring tools are actually used appropriately. It may be possible to review recent events and determine whether the events were investigated. Leverage the results of the rest of the audit in performing this assessment. For example, if you found significant issues in an area the administrator was supposedly monitoring, you might question the effectiveness of that monitoring.

13. Verify that policies and procedures are in place to identify when a patch is available and to evaluate and apply applicable patches. Ensure that all approved patches are installed per your policy.

Most storage vendors have regularly scheduled patch releases. You need to be prepared for the scheduled releases so that you can plan appropriately for testing and installation of the patches. If all the patches are not installed, widely known security vulnerabilities or critical performance issues could exist.

How

Interview the administrator to determine who reviews advisories from vendors, what steps are taken to prepare for the patches, and how long the patches are tested before being applied to the production storage systems. Ask to review notes from the previous patching cycle.

Obtain as much information as possible about the latest patches through conversations with the administrator and a review of vendor documentation, and determine the scope of the vulnerabilities addressed by the patches. Compare the available patches with the patches applied to the storage platform. Talk with the administrator about steps taken to mitigate potential risk if the patches are not applied in a timely manner.

14. Perform the steps from Chapter 5 as they pertain to the system you are auditing.

In addition to auditing the logical security of the system, you should ensure that appropriate physical controls and operations are in place to provide for system protection and availability.

How

Reference the steps from Chapter 5, and perform those that are relevant to the system being audited. For example, the following topics are likely to be pertinent:

- Asset inventory
- Physical security
- Environmental controls
- Capacity planning
- Change management
- System monitoring
- Backup processes
- Disaster recovery planning

Knowledge Base

Following are additional resources where you can obtain information about stor-

age and related controls. The vendors offer a tremendous amount of information on their websites for general consumption.

Resource	Website
Storage Networking Primer	www.snia.org/education/storage_networking_primer
RAID Primer	www.acnc.com/raid
RAID Recovery Guide	www.raidrecoveryguide.com
Dell EMC	www.dellemc.com
NetApp	www.netapp.com
HP	www.hp.com
Storage Glossary	www.webopedia.com/Hardware/Data_Storage

Master Checklists

The following checklist summarizes the steps for auditing storage.

Checklist for Auditing Storage

- ❑ 1. Document the overall storage management architecture, including the hardware and supporting network infrastructure.
- ❑ 2. Obtain the software version and compare it against policy requirements.
- ❑ 3. Determine what services and features are enabled on the system and validate their necessity with the system administrator.
- ❑ 4. Review and evaluate procedures for creating administrative accounts and ensuring that accounts are created only when there's a legitimate business need. Also review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.
- ❑ 5. Evaluate the process and policies used for granting and revoking access to storage.
- ❑ 6. Evaluate how capacity is managed for the storage environment to support existing and anticipated business requirements.
- ❑ 7. Evaluate how performance is managed and monitored for the storage environment to support existing and anticipated business requirements.
- ❑ 8. Evaluate the policies, processes, and controls for data backup frequency, handling, and remote storage.
- ❑ 9. Verify that encryption of data-at-rest is implemented where appropriate.
- ❑ 10. Verify that network encryption of data-in-motion is implemented where appropriate.
- ❑ 11. Evaluate the low-level and technical controls in place to segregate or firewall highly sensitive data from the rest of the storage environment.
- ❑ 12. Review and evaluate system administrator procedures for security monitoring.
- ❑ 13. Verify that policies and procedures are in place to identify when a patch is available and to evaluate and apply applicable patches. Ensure that all approved patches are installed per your policy.
- ❑ 14. Perform the steps from Chapter 5 as they pertain to the system you are auditing.

CHAPTER 13

Auditing Virtualized Environments

Innovations in virtualization permanently changed the footprint, architecture, and operations of data centers, and in the last five years these innovations have extended to the application space. This chapter discusses auditing virtualized environments and begins with an overview of common virtualization technologies and key controls. The virtualization audit combines the concerns of the hypervisor and the guest operating systems. Although the focus of this chapter is the hypervisor and server virtualization, you can apply many of the same steps and concepts to desktop or application virtualization. We make the assumption that the system components are under your control. You should reference Chapter 16 for guidance on how to ensure outsourced virtualized environments are properly managed and secured.

This chapter covers the following:

- A brief technical overview of virtualization
- How to audit the virtualization environment
- Tools and resources for enhancing your virtualization audits

Background

In the IT space, virtualization describes the implementation of an abstraction layer to represent or emulate computing resources for access by other elements of the environment. Virtualization can be applied to hardware or to various operating system or application components. For the purposes of this chapter, we'll be focusing primarily on *hardware virtualization*. In this situation, the physical hardware of a system is isolated and managed by a layer called a *hypervisor* or *host* operating system (OS). The hypervisor then allows one or more *guest* OS instances to be installed, providing virtual hardware resources to each and facilitating communication to and from each guest. This configuration provides tremendous flexibility for computing needs. Rather than purchasing individual servers for application teams, an infrastructure group can purchase a few larger servers and combine many workloads on a smaller number of physical systems. Virtualization also provides the ability to easily move a guest OS to another physical system

in the event of a problem or upgrade. Since the storage hardware can also be virtualized, the storage needs of hundreds of guests can be managed with a single storage array, simplifying backup/restore and business continuity needs. Figure 13-1 illustrates the separation of virtual machines from the physical hardware. This abstraction allows virtual machines to be implemented across a wide range of hardware configurations without disrupting the guest OS or applications. When virtualization platforms are combined through clustering, virtual machines can be moved between them with ease.

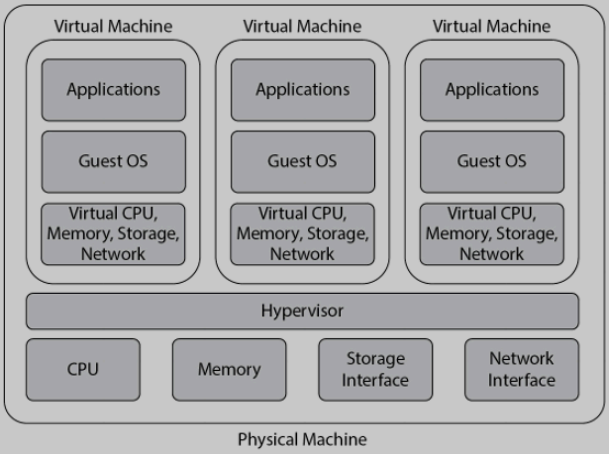


Figure 13-1 Virtualization model

The virtualization hypervisor is often installed onto a bare metal server as the fundamental operating system; many of the basic protections applicable to Windows or Linux server systems can be applied to the hypervisor layer. Many vendors also allow the hypervisor to be installed as an application on top of another operating system, although this is not common in server virtualization. The hypervisor is designed to utilize special processor instructions to support multiple operating systems. Processor manufacturers like Intel and AMD led this charge several years ago, and server manufacturers quickly adopted the capabilities and began developing highly customized hardware platforms packed with computing power, memory, and storage to support virtualization. The convergence of various computing, networking, and storage capabilities into single servers or racks, called a converged infrastructure, allows businesses to quickly deploy large-scale computing capacity in a small physical footprint. According to a 2016 Gartner report, many organizations now virtualize more than 75 percent of their data center infrastructure.

Hardware virtualization can also extend to desktop environments. This implementation, usually called desktop virtualization, separates an end user's physical hardware from the operating system and applications running on it. While sitting in the office, the virtual desktop may seem to operate just like a conventional workstation, but the increased flexibility of the virtualization layer may also allow the employee to access the same workstation, files, and applications from a phone, a home PC, or other device. The principles in this chapter generally apply to desktop virtualization environments. Companies often separate desktop or end-user environments from server workloads due to considerations around operational management.

Virtualization has also extended to the application space. In 2013, the Docker platform was released, which allows the packaging of applications inside a *container,* which can easily be moved to other physical systems as needed for load balancing, failover, and more. The Docker platform has the support of some major industry players, including Google, Amazon, IBM, Microsoft, Cisco, and others. Application development teams have embraced the container concept in recent years because of the flexibility it provides. This chapter does not address containers or application virtualization directly, but in concept, many of the same principles apply.

Commercial and Open-Source Projects

Several commercial players compete in the hardware virtualization market, including VMware, Microsoft, Citrix, and Oracle. Some of these companies maintain open-source projects, including Xen by Citrix and VirtualBox by Oracle. KVM is a popular open-source virtualization project for Linux. Links to each of these projects are located in "Knowledge Base" section at the end of the chapter.

Virtualization Auditing Essentials

To understand the material in this chapter, you need a basic understanding of the components that make up the virtualization environment. Your role as an auditor and advisor will significantly improve if you understand major technology trends challenging virtualization models.

Security models, business alignment, capacity planning, and performance management are more important than ever before in virtual environments. Smaller environments may have a few virtual servers running on a single powerful physical server, whereas larger environments may include hundreds or thousands of virtual servers and desktops running on a complex infrastructure of clustered servers connected to a massive storage area network (SAN). The scale may change the scope or approach to the audit, but the same business requirements and controls exist. Resource management and monitoring of each of the components, separately and collectively, enable the virtual environment to function.

Figure 13-2 illustrates an example collective environment and several auditing considerations. Notice that these considerations also apply to a normal server or storage audit. What's different? What are the security concerns that keep administrators awake? What should auditors explore? The hypervisor has control requirements similar to those found in a server, but it also has unique requirements to ensure that the hosted environment doesn't present additional control weaknesses to the guest operating systems. The guest operating systems have unique control requirements because of the necessity to keep appropriate segregation controls in place between servers. Mildly complicating this mix are different conceptual approaches to creating the virtual environment.

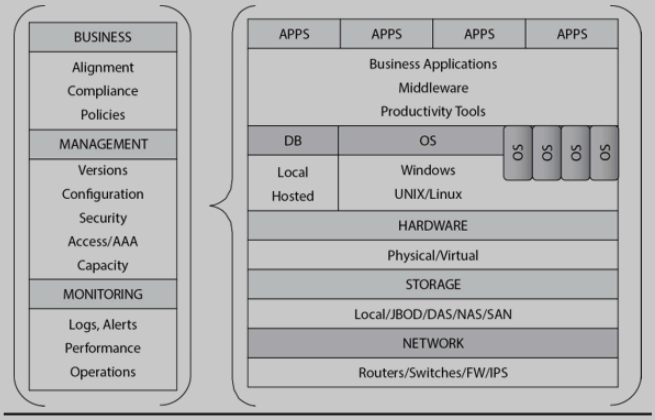


Figure 13-2 Example virtualization audit model

Test Steps for Auditing Virtualization

The virtualization audit covered here is designed to review key controls that protect the confidentiality, integrity, or availability of the environment for the supported operating systems and users that rely on the environment. Each of the following steps applies to some extent; however, use your judgment to determine the depth to which you decide to take any one step. For example, an auditor reviewing high-performance environments supporting a business-critical application might spend more time asking questions and reviewing vendor-specific analysis to verify that the virtualized environment has the capacity and performance necessary to handle peak loads.

Keep in mind that a large organization may support several different virtualization platforms for different purposes. As you outline the scope of virtualization in your organization, you may need to revisit the scope of the audit. This will ensure you maintain the right level of focus and will help operational and administrative teams align their resources to support your efforts.

NOTE This audit focuses on the hypervisor and management of the virtual environment, regardless of where the hypervisor is installed. If the hypervisor is installed as an application on another operating system, audit the underlying operating system separately using the appropriate test steps in Chapter 7 or Chapter 8.

Note that there are several excellent hardening guides and configuration checking utilities, and we encourage the use of these tools to help provide consistency across the environment. Vendors have different approaches for shipping products. Some vendors include unnecessary services and product features enabled. Others ship their products in a hardened state, requiring the administrator to enable additional services. Note that many of the hardening guides have a narrow scope that focuses on the compromise of the hypervisor as opposed to ensuring that controls support business processes and objectives. This is the value provided by Control Objectives for Information and Related Technologies (COBIT).

Initial Steps

1. Document the overall virtualization management architecture, including the hardware and supporting network infrastructure.

The team responsible for managing virtualization should maintain documentation illustrating the virtualization architecture and how it interfaces with the rest of the environment. Documentation should include supported systems, management systems, and the connecting network infrastructure. This information will be used by the auditor to help interpret the results of subsequent audit steps.

How

Discuss and review existing documentation with the administrator. As applicable, verify that document structure and management are aligned with corporate standards. Verify the entire environment, including management, storage, and network components, is properly documented.

2. Obtain the software version of the hypervisor and compare with policy requirements.

Review the software version to ensure that the hypervisor is in compliance with policy. Older software may have reliability, performance, or security issues that can increase the difficulty in managing the virtualization platform(s). Additionally, disparate software versions may increase the scope of the administrator's responsibilities as he or she attempts to maintain control over the different hypervisors and their feature, control, and administration differences.

How

Work with the administrator to obtain this information from the system, and review vendor documentation. Ensure that the software is a version the vendor continues to support and does not contain widely known and patchable vulnerabilities that would bypass existing controls. Also verify that the current running version does not contain performance or reliability issues that would affect your environment. Review any mitigating factors with the administrator, such as issues that have not been fixed but are not applicable to the environment or are protected through other controls.

3. Determine what services and features are enabled on the system and validate their necessity with the system administrator.

Unnecessary services and features increase risk of exposure to misconfigurations, vulnerabilities, and performance issues and complicate troubleshooting efforts.

How

Today's virtualization systems range from the very simple to the extremely complex. Work closely with the virtualization administrator to discuss enabled services and their applicability to the environment. Review and evaluate procedures for assessing vulnerabilities associated with necessary services and features and keeping them properly configured and patched.

Account Management and Resource Provisioning/Deprovisioning

Administrative accounts in the virtual environment must be managed appropriately, as should the provisioning and deprovisioning of virtual machines.

4. Review and evaluate procedures for creating accounts and ensure that accounts are created only when a legitimate business need has been identified. Also review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.

Effective controls should govern account creation and deletion. Inappropriate or inadequate controls could result in unnecessary access to system resources, placing the integrity and availability of sensitive data at risk.

How

Interview the system administrator and review account creation procedures. This process should include some form of verification that the user has a legitimate need for access. Take a sample of accounts and review evidence that they were approved properly prior to being created. Alternatively, take a sample of accounts and validate their legitimacy by investigating and understanding the job function of the account owners.

Review the process for removing accounts when access is no longer needed. This process could include a component driven by the company's human resources (HR) department providing information on terminations and job changes. Or the process could include a periodic review and validation of active accounts by the system administrator and/or other knowledgeable managers. Obtain a sample of accounts and verify that they are owned by active employees and that each employee has a legitimate business requirement for administrative access.

Best practice controls for administrative accounts include permitting only individually assigned accounts to have high-level access. If the technology in question does not support this, review other controls designed to prevent widespread use of shared accounts in the environment. In particular, discuss how passwords for shared admin accounts are distributed and when those passwords are changed. Termination controls are sometimes overlooked for shared accounts, so be prepared to review those procedures as well.

5. Verify the appropriate management of provisioning and deprovisioning new virtual machines, including appropriate operating system and application licenses.

Written policies should govern the process used to create new virtual machines, manage users, and allocate software licenses. The ease of spinning up new servers for development and testing has created a new challenge for managing hardware and license resources.

Policies or procedures should also exist for "cleaning up" or removing virtual machines, rights, and licenses that are no longer needed when a project is completed. Failure to manage virtual host allocation could unnecessarily expend virtualization capacity and software licenses.

Virtual machines should be accountable to specific groups or users. Failure to govern rights management may allow users that should no longer have access to hosts to maintain inappropriate levels of access.

How

Discuss policies and procedures for provisioning and deprovisioning new guest VMs and accounts with the virtualization administrator, including license allocation, user management, and host ownership. Be sure to include discussion on development environments, where server (and virtual system) sprawl tends to become a problem.

Virtual Environment Management

Virtual environments must be managed appropriately to support existing and future business objectives. Resources must be monitored and evaluated for capacity and performance. Resources must also support the organization's business continuity/disaster recovery objectives.

6. Evaluate how hardware capacity is managed for the virtualized environment to support existing and future business requirements.

Business and technical requirements for virtualization can change quickly and frequently, driven by changes in infrastructure, business relationships, customer needs, and regulatory requirements. The virtualization hardware and infrastructure must be managed to support both existing business needs and immediate anticipated growth. Inadequate infrastructure places the business at risk and may impede critical business functions that need more hardware capacity.

How

Virtual machine capacity is managed by the hypervisor to allocate a specific amount of storage, processor, and memory to each host. Verify that capacity requirements have been documented and that customers have agreed to abide by them. Capacity allocation may directly affect performance. Review processes for monitoring capacity usage for storage, memory, and processing, noting when they exceed defined thresholds. Evaluate processes in place for responding and taking action when capacity usage exceeds customer-approved thresholds. Capacity for guest VMs can often be adjusted in real time by an administrator. Discuss the methods used to determine the present virtualization requirements and anticipated growth. Review growth plans with the administrator to verify that the hardware can meet the performance requirements, capacity requirements, and feature requirements to support infrastructure and business growth.

Some organizations have processes to leverage external cloud providers for surge capacity. This concept, often called cloud bursting, allows spikes in demand to be offloaded to a cloud provider, such as Amazon Web Services or Microsoft Azure. If cloud bursting is used in your organization, discuss how these temporary loads are managed and how the cloud environment figures into overall capacity management.

7. Evaluate how performance is managed and monitored for the virtualization environment to support existing and anticipated

business requirements.

Virtualization performance of the infrastructure as a whole and for each virtual machine is driven by several factors, including the physical virtualization media, communication protocols, network, data size, CPU, memory, storage architecture, and a host of other factors. An inadequate virtualization infrastructure places the business at risk of losing access to critical business applications. It's possible to have adequate capacity but incorrectly configured and underperforming virtual machines that fail to deliver on the service level agreement (SLA).

How

Verify that regular periodic performance reviews of the processor, memory, and bandwidth loads on the virtualization architecture are performed to identify growing stresses on the architecture. A common performance measurement for virtual environments is based on input/output operations per second (IOPS). Verify that performance requirements have been documented and that customers have agreed to abide by them. Review processes for monitoring performance and noting when performance falls below defined thresholds. Evaluate processes in place for responding and taking action when performance falls below customer-agreed thresholds. Discuss the methods used to determine present performance requirements and anticipated changes.

NOTE A review of capacity management and performance planning is essential to this audit. Be careful to ensure that the administrator has a capacity management plan in place and verifies that performance needs are appropriate for the organization.

8. Evaluate the policies, processes, and controls for data backup frequency, handling, and offsite management.

Processes and controls should meet policy requirements, support business continuity/disaster recovery (BC/DR) objectives, and protect sensitive information. Data backups present complex challenges for organizations, particularly when it comes to virtualization platforms and other large, central systems. Vendors offer several solutions to manage the frequency, handling, and offsite delivery of data and system backups.

How

Review policy requirements around system backups. For virtualization platforms, you should ensure that all necessary components of the environment are being backed up. This may include remote management systems, hypervisor platforms, storage arrays, and configuration information, as well as the virtual machines themselves. Discuss the backup strategy with the virtualization administrators. Obtain the backup frequency, encryption status, and physical location of backups.

In addition to supporting day-to-day operational needs, backups are a key element in disaster recovery plans (DRP). You should ensure that the backup strategy for the virtualization platform is sufficient to meet DRP needs. Backup frequency directly influences the recovery point objective (RPO) capability, while backup method, media, and location may affect the recovery time objective (RTO). For more information on DRP, RPO, and RTO, refer to Chapter 5.

9. Review and evaluate the security of your remote hypervisor management.

Secure remote hypervisor management protects the hypervisor from remote attacks that might otherwise disrupt the hypervisor or hosted virtual machines. Each of the hypervisor products has its own management tools designed to allow remote administration of the hypervisor and virtual machines. Many commercial tools can also manage other commercial hypervisors to facilitate management of heterogeneous virtual environments. While remote management features vary by product, the areas that should be reviewed have many commonalities.

Unused services, accessible application programming interfaces (APIs), and installed applications may subject the hypervisor to additional attack vectors if a security flaw is discovered. In addition, remote users should be forced to access the hypervisor system using accounts that can be tied to a specific user for logging and tracking. The difference between this step and step 3 is the careful analysis of network-accessible components for the hypervisor with regard to remote management. Unless specifically required and appropriately controlled, network-accessible features should not be enabled. Enable only those components that are necessary and appropriately configured for remote management.

How

Each vendor provides specific security guides for enabling remote management. These security guides are generally easy to read and should be reviewed in detail prior to beginning the audit. The execution of this step consists of a policy review, account permissions review, and a configuration review.

Review remote access policies and access methods with the administrator. Verify that all remote access is logged to a system separate from the environment. Question the need for any cleartext communications used for remote access. Identify and validate the appropriateness of administrative accounts that have remote access. Determine whether remote access and remote administration are permitted from a wide range of network sources, or whether access sources are limited to a single or small number of origin systems.

NOTE The use of secure protocols is particularly important in a demilitarized zone (DMZ) and other high-risk environments. Where insecure protocols are in use, especially those permitting cleartext communication, an attacker could extract valuable information simply by snooping network traffic.

Obtain vendor-appropriate guidance for configuring secure remote hypervisor access. These should be used to identify and verify that the environment is securely configured for remote access. Most vendors offer security and configuration guides that include remote management access. In addition, the Center for Internet Security offers a hardening guide specifically for VMware.

As deployments mature, organizations turn to automation for many administration tasks. You should review the use of automation for virtualization management, including software and scripts used for these tasks. If remote management is limited to certain source systems, remote automation should be similarly limited.

Security Monitoring and Additional Security Controls

10. Review and evaluate system administrator procedures for monitoring and maintaining the state of security on the system.

If the virtualization administrator doesn't monitor his or her systems for unexpected changes, security incidents could occur without his or her knowledge. By monitoring, we mean *actively* reviewing log data and system information. Merely enabling log collection without reviewing the resulting data is just barely preferable to having no log data at all.

System security must also be *maintained*. The world of security vulnerabilities is an ever-changing one, and it is unrealistic to believe that a static audit program can provide assurance of system security on a daily basis. A vulnerability scanning tool that is updated frequently can provide an effective mechanism for understanding the current security state of the machine. In addition, if the administrator has a security patching process in place, this scan will provide some validation of the effectiveness of that process.

How

Interview the system administrator and review relevant documentation to gain an understanding of security monitoring practices. The level of monitoring should be consistent with the criticality of the system and the inherent risk of the environment (for example, a virtualization environment supporting critical financial data should have robust security monitoring). The system administrator is responsible for monitoring the environment to identify activity and trends that might allow the prevention of critical issues. Several excellent tools are available for monitoring virtual environments.

If security event monitoring is in place, assess the frequency of the monitoring and the quality with which it is performed. Look for evidence that the security monitoring tools are actively used. Review recent events and determine whether they were investigated and resolved. Leverage the results of the rest of the audit in performing this assessment. For example, if you found significant issues in an area the administrator was supposedly monitoring, it might lead to questions around the effectiveness of that monitoring.

Note that some organizations may configure event logs to be sent to centralized logging environments to be reviewed by dedicated teams or even by outside service providers. In these cases, discuss monitoring practices with the monitoring team.

11. Verify that policies and procedures are in place to identify when patches are available and to evaluate and apply applicable patches. Ensure that all approved patches are installed per your policy requirements.

Most virtualization vendors have regularly scheduled patch releases. Your business should be aware of the release schedule and should have plans in place for testing and installing patches. If all the patches are not installed, widely known security vulnerabilities or critical performance issues could exist. This is particularly important in virtualized environments, as vulnerabilities in the hypervisor could put all of the guest systems in those environments at risk.

How

Interview the administrator to determine who reviews security and patch advisories from vendors. Review what steps are taken to prepare for patches and how the patches are tested before being applied to the production systems. Ask to review notes from the previous patching cycle.

Obtain as much information as possible about the latest patches through conversations with the administrator and review of vendor documentation, and determine the scope of the vulnerabilities addressed by the patches. Compare the available patches with the patches applied to the hypervisor. Talk with the administrator about steps taken to mitigate potential risk if the patches are not applied in a timely manner.

12. Review and evaluate the security around the storage of virtual machine data.

Virtual machines are stored and manipulated as files that are easily transported, copied, and viewed. Shared storage for virtual machines should have controls in place to isolate sensitive virtual machines and content from the rest of the environment.

Some environments might encrypt data-at-rest. Encryption of data-at-rest involves encrypting data as it is stored on disk. Encryption of data-at-rest for virtual machines is important, because the storage system may be physically separate from the VM and may be shared by multiple VMs and even other kinds of systems. Limiting access to the guest VM data image on disk through encryption reduces the risk of data leakage. If you consider the likely avenues for data theft, removing data directly from its storage system is a key risk.

This step isn't appropriate for all environments and may be covered by other controls or applications.

How

Ensure virtual machines are stored in such a manner that sensitive virtual machines are isolated from the rest of the network and that only appropriate administrators have access. Consideration must also be given to managing and auditing administrative access to a storage environment containing sensitive virtual machines.

Verify that encrypted data is encrypted properly. Additionally, review the location where the encryption keys are stored, as the strength of an encryption system relies on protection of the encryption keys. If the encryption keys are stored along with the encrypted data, an attacker can subvert the security simply by accessing the encryption keys.

If encryption is used, verify that the disaster recovery plan includes encryption key management. Failing to back up encryption keys properly may result in the inability to recover a backup.

13. Verify that network encryption of data-in-motion is implemented where appropriate.

Policy requirements may require that traffic be encrypted for applications that contain sensitive information or for backing up some virtualized hosts to another location. Network encryption serves two main purposes: to protect authentication credentials as they move across the network and to protect the actual data as it

moves over the network. The network is not a secure environment—IP addresses can be spoofed, and network traffic can be redirected and sniffed.

How

Work with the administrator to verify that encrypted protocols are used for remote administration of the virtual environment. Review policy requirements with the administrator and determine if any of the virtualization data is required to be encrypted in transit. If the virtual hosts contain sensitive data, verify that network traffic used to back up or replicate the hosts is encrypted.

Given the additional potential complexity derived from dedicated networks for storage, backup, management, failover, and so on, an auditor might want to document the data flow between these components for the virtual environments. This may have been accomplished in step 1.

14. Evaluate the low-level and technical controls in place to segregate or firewall highly sensitive data on critical virtual machines from the rest of the virtualization environment.

Controls should exist that restrict access between virtual machines to protect sensitive information such as cardholder data (CD), personally identifiable information (PII), source code, and other types of proprietary data, including administrative rights to the host. Each of the hypervisors has specific settings and controls that can be implemented to assist with the segregation of data between hosts. Commonly discussed threats specifically include the use of shared folders and the ability to copy and paste between a host operating system and the hosted virtual machine. If encryption is used, describe it here and evaluate the handling of keys, including the granting and revocation of rights, keys, and certificates.

How

Review with the virtualization administrator the controls in place to isolate virtual machines that have different classification levels. Identify technical and administrative controls that force separation between sets of data. Strong controls will prevent commingling of disparate data types and create actionable, nonrepudiated logs when these controls are bypassed. Sensitive virtual machines should not be directly accessible by the rest of the environment.

Review auditing and log management procedures governing administrative access to the virtualization environment that could bypass intended controls. Consider compensating controls such as data encryption.

The different configuration options among the various hypervisors to protect virtual machines from each other and the host require that the auditor gather additional knowledge to identify vendor-recommended best practices. Discuss specific options with the administrator in the business context of environmental risk and compensating controls.

As virtualization platforms have grown in capability and complexity, the virtual network interfaces present in early platforms evolved to virtual switches and virtual firewalls. An entire virtual network architecture can be present in virtualization platforms. Your organization should make a conscious decision about whether these components of the virtualization system are managed by the network team or by the virtualization administrators.

15. Evaluate the use of baseline templates and the security of guest virtual machines as appropriate to the scope of the audit.

Baseline templates allow you to provision configured virtual machines quickly. One of the best ways to propagate security throughout an environment is to ensure that new systems are built correctly before moving into testing or production. In addition, if the scope of the audit includes evaluating guest virtual machines, refer to Chapters 7 and 8.

How

Through interviews with the system administrator, determine the methodology used for building and deploying new guest VMs. If a standard build is used, consider auditing a newly created system using the steps in Chapters 7 and 8. It's a good practice to include your baseline configurations as part of your normal audit routines.

16. Perform the steps from Chapter 5 and Chapter 12 as they pertain to the environment you are auditing.

In addition to auditing the logical controls of the system, you must ensure that appropriate environmental controls are in place to provide for system protection and availability. Also consider a deep review of the storage environment to ensure that data is protected and that capacity and performance are managed.

How

Reference the steps from Chapter 5, and perform those that are relevant to the system being audited. For example, the following topics are likely to be pertinent:

- Asset inventory
- Physical security
- Environmental controls
- Capacity planning
- Change management
- System monitoring
- Backup processes
- Disaster recovery planning

Reference the steps from Chapter 12, and perform those that are relevant to the system being audited. For example, the following topics are likely to be pertinent:

- Capacity management
- Performance management
- Data protection

Knowledge Base

Following are additional resources that can offer information about virtual environments and related controls. Vendors include a tremendous amount of information on their websites for general consumption. In addition, the community of helpful enthusiasts, open-source projects, and forums continues to grow daily.

Hypervisors

Platform	Website
VMware	www.vmware.com
Microsoft Hyper-V	www.microsoft.com/en-us/cloud-platform/server-virtualization
Oracle VM	www.oracle.com/virtualization
Citrix Hypervisor	www.citrix.com/products/citrix-hypervisor
KVM (Open Source)	www.linux-kvm.org
VirtualBox (Open Source/Oracle)	www.virtualbox.org

Tools

Tool	Website
VMware Open Source Tools	www.vmware.com/opensource.html
VMware Security Resources	www.vmware.com/security.html
Microsoft Hyper-V Security	https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/plan/plan-hyper-v-security-in-windows-server
CIS Benchmarks (VMware and Docker)	www.cisecurity.org/cis-benchmarks/

Master Checklists

The following checklist summarizes the steps for auditing virtualization.

Checklist for Auditing Virtualization
❑ 1. Document the overall virtualization management architecture, including the hardware and supporting network infrastructure.
❑ 2. Obtain the software version of the hypervisor and compare with policy requirements.
❑ 3. Determine what services and features are enabled on the system and validate their necessity with the system administrator.
❑ 4. Review and evaluate procedures for creating accounts and ensure that accounts are created only when a legitimate business need has been identified. Also review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.
❑ 5. Verify the appropriate management of provisioning and deprovisioning new virtual machines, including appropriate operating system and application licenses.
❑ 6. Evaluate how hardware capacity is managed for the virtualized environment to support existing and future business requirements.
❑ 7. Evaluate how performance is managed and monitored for the virtualization environment to support existing and anticipated business requirements.
❑ 8. Evaluate the policies, processes, and controls for data backup frequency, handling, and offsite management.
❑ 9. Review and evaluate the security of your remote hypervisor management.
❑ 10. Review and evaluate system administrator procedures for monitoring and maintaining the state of security on the system.
❑ 11. Verify that policies and procedures are in place to identify when patches are available and to evaluate and apply applicable patches. Ensure that all approved patches are installed per your policy requirements.
❑ 12. Review and evaluate the security around the storage of virtual machine data.
❑ 13. Verify that network encryption of data-in-motion is implemented where appropriate.
❑ 14. Evaluate the low-level and technical controls in place to segregate or firewall highly sensitive data on critical virtual machines from the rest of the virtualization environment.
❑ 15. Evaluate the use of baseline templates and the security of guest virtual machines as appropriate to the scope of the audit.
❑ 16. Perform the steps from Chapter 5 and Chapter 12 as they pertain to the environment you are auditing.

CHAPTER 14

Auditing End-User Computing Devices

This chapter discusses two separate audits, beginning with Windows and Mac client systems and then covering mobile devices such as Android and iOS phones and tablets. These audits include system management processes, administrative controls, policies, and basic controls that should be present on these systems. The following topics are discussed:

- Background of client and mobile computing technologies
- Essential auditing steps for these technologies
- Key technical resources for additional information

Background

The computing power available to the everyday user has grown exponentially over a generation to the point where today's small desktops and laptops are far more powerful than the room-filling supercomputers of 20 years ago, while mobile phones and tablets are now packing as much power as some modern laptops. The power of these devices means that end users can do more today with a laptop or phone than at any time in history. This also means that these devices need to be managed closely in a business environment, as a powerful, unmanaged system can present a significant risk to business operations.

NOTE For the purposes of this book, even though laptops and similar form factors are "mobile" in the sense that they are portable, we will use the term "mobile device" to refer to smartphones and tablets running smartphone-like operating systems. The terms "client" and "client device" will refer to laptops and desktops running Windows or macOS operating systems.

Until the early 1970s, computing tasks were performed mostly on large, centralized computers, and employees entered instructions by punching individual holes in special paper punch cards and inserting those cards in a precise sequence

into a feeder. During the late 1960s and early 1970s, terminal interfaces were developed, and people could use a keyboard to access the central computer remotely. With the advent of personal computers, including the IBM PC, which arrived in 1981, computing power began moving closer to the user. Over the last 40 years, these systems have evolved, with components miniaturizing, performance improving, and new technologies arriving, all while becoming less expensive. Computers became light enough to carry, battery technology advanced, and soon laptops emerged. As computers improved, employees have become more and more productive, and businesses have continually found new ways to leverage this power.

While computers were shrinking and becoming more powerful, companies began experimenting with pocket-sized computing power, a realm once dominated by calculators. One of the first personal digital assistants (PDAs), the Psion Organizer, was released in 1984. In the mid-1990s, companies like IBM and Nokia began adding telephone features to PDAs, and the smartphone was born. By the turn of the century, companies like Palm and Research in Motion, maker of the BlackBerry, along with Nokia, were considered leaders in the mobile space. Apple, which was an early player in the PDA space with the Newton device, became a major force in mobility in 2007 with the release of the first iPhone. Google wasn't far behind, with the first Android-based devices launching in 2008. Google's Android operating system now dominates the mobile landscape, powering over 85 percent of all smartphones and tablets. Apple's iOS is a distant second but covers nearly all of the remaining market share. Other OS offerings from the likes of Samsung, BlackBerry, and Microsoft now make up just a fraction of a percent.

Much of this would just be interesting trivia and not particularly relevant to managing end-user devices except for a few trends. First, the growing power and lower cost of laptops and phones over time meant that more and more people purchased these for personal use. Second, while laptops and phones became less expensive, they were still a cost for businesses; companies didn't necessarily want to buy them for everyone, and they expected devices to be used for several years before replacing them. This led to a situation in the early 2000s when, for the first time, employees began to have more modern computing technologies at home than at work. Believing they would be more productive and more competitive by leveraging their personal investments in the workplace, employees began to expect that they could bring their phones and laptops to work and use them in place of company-provided systems. This concept, called "consumerization," was a buzzword in enterprise environments in the 2000s and was embraced by many companies, who felt there was an advantage to allowing this kind of flexibility. Companies began to adopt bring your own device (BYOD) programs to allow employees to buy their own laptops or smartphones and use them to do their work. This made employees feel more productive and empowered, while reducing support and device costs for the business. By the 2010s, security, audit, and legal teams began to question the wisdom of this practice, as a personally owned device storing company-owned data can create a difficult legal situation if the employee leaves, and technical controls to address this were only partially effective. As a result, some companies backed away from BYOD programs, while others added policy language or made other changes to protect company resources. The challenge of device ownership remains a key point to consider when auditing end-user systems.

Part 1: Auditing Windows and Mac Client Systems

Microsoft Windows and Apple macOS are the predominant operating systems powering desktops and laptops in most homes and businesses. This section describes audit concepts and steps typical of a client system audit.

Windows and Mac Auditing Essentials

In most home settings, computers are stand-alone devices. You may have a local area network handling computers, printers, smart TVs, thermostats, and more, but the computer itself is usually "unmanaged"—no policy restrictions or management systems are in place to control the options available to you. In most business environments, this is not the case. Businesses spend a great deal of money on computing devices for employees and want to make sure the devices are usable, productive, secure, and supportable. To facilitate this, businesses will deploy management systems along with configuration rules or policies to manage the features, capabilities, and software available to employees. You will have to consider not only the state of the end user's desktop or laptop system but also the management system in place.

Operating a computer in an enterprise also involves processes around installing company software, configuring the system, managing users and passwords, backing up data, and responding to problems. These in turn lead to other systems in the client ecosystem, such as backup systems, trouble or ticketing systems and helpdesks, license management, and more. The steps described here focus on a typical business Windows or Mac client environment. Your business environment may be more or less complex. Work with your client team to understand the full scope of client management before proceeding.

NOTE The term "client" references the client-server computing relationship, where a system providing resources is called a server, and a system consuming resources is called a client. To some extent, referring to laptops and desktops as clients is legacy terminology; in a modern environment, laptops can act as servers, servers can be clients, and so on. In some organizations, the service for provisioning systems like laptops and desktops goes by other names, such as Device Management, but the "client" term is retained here.

Windows and macOS are the prevalent operating systems deployed to end-user desktop or laptop-style devices. Where technical steps are described, tips for both operating systems are included where applicable. These steps deal with high-level principles of managing business systems and are not an attempt to describe the hundreds of possible controls available in Windows and macOS. Resources dealing with detailed configuration options are covered later in the chapter. Work with your company's client system team for more details.

Windows auditors should also review Chapter 7 on the Windows Server platform, as many of the tools and techniques are applicable to the Windows client space.

Test Steps for Auditing Windows and Mac Client Systems

1. Review company policies around client devices and ensure device ownership and user responsibilities are covered.

As noted earlier, the presence of personally owned devices in the business environment creates challenges around device management and data protection. This step ensures your organization has considered these issues and has provided appropriate information to end users.

How

Obtain a copy of the company policy related to the use of personally owned or non–company-owned devices. The policy should clearly define the circumstances under which noncompany assets can be used and how they should be handled. In risk-averse organizations, personally owned computers might not be permitted under any circumstances, or they may be permitted to have only virtualized access to company resources.

Be sure to consider any supplemental labor engagements used by your company. Any policies that apply to employees should apply at least as stringently to third parties.

If personally owned computers are allowed, ensure that language addressing company data is present in the policies. If an employee is allowed to keep sensitive company data on a personal device and that employee later leaves the company or is terminated, it may be difficult or impossible for the organization to regain control of that data.

If personally owned computers are prohibited, determine whether your organization has technical controls to prevent these devices from connecting to company networks. These controls usually are in the form of Network Access Control or Network Admission Control (NAC) technologies that can interrogate a system and take appropriate action to allow or deny access.

NOTE NAC technologies are not covered in this book, but if they are present in your environment, a very simple verification step is to bring a personal laptop into the office and try to connect to either wired or wireless resources. If you are unable to gain access, the NAC system is probably working as expected. Discuss with a network administrator for more information.

Obtain a copy of the client security policy. This should address basic security requirements for client systems. A typical client security policy should include many of the items listed later in this audit, including disk encryption, antivirus software, and basic operating system configuration.

In addition, you should review the company's policies around user expectations. This is often called an acceptable use policy (AUP), which covers what users may and may not do with company-owned equipment or on company time. An AUP should clearly define what is allowed and what is not allowed and should explain the consequences of violating the policy.

Some organizations may allow personal computers to connect to special external networks that cannot access internal company resources. Similar in concept to a coffee shop network, these external networks might be provided as a convenience for both employees and visitors. For AUP purposes, these networks should be thought of as company networks unless separately addressed in the policy.

2. Ensure the organization has a device management infrastructure commensurate with policy goals and company strategy.

Client device management systems allow organizations to track device inventory, manage software, install patches, apply security policies, and more. This step assesses the existence and scope of device management systems. If device management is not in place, the company may not be able to properly secure and protect company assets and data.

How

Discuss the client device management systems with the client administrators. While a number of commercial and open-source products exist in the market, you should expect that any solution should, at a minimum, be able to facilitate the following key functions:

- Provide visibility to the device inventory for the company
- Provide information on the operating system version and patch status for all systems
- Manage the fleet application inventory, providing access to provisioned applications
- Apply security and system configuration policies to managed systems

In larger organizations, Windows systems are often managed as part of an Active Directory (AD) domain. Joining a computer to a domain facilitates the deployment of Group Policy Objects (GPOs) to Windows client systems. Various settings can be enforced via GPO, including account characteristics, Internet restrictions, and software features. While Mac systems don't support GPOs, they are sometimes joined to Windows AD domains. In other cases, Mac machines are managed separately from the Windows fleet.

If your organization uses GPOs, an additional audit check you might perform would be to ensure that a proper change management process is in place to govern GPO changes. Since these can affect the performance and operation of many end-user systems, it's important for businesses to have a sound change process around these configuration systems.

Over time, client device management systems have been merging with enterprise mobility management (EMM) systems. You may find that your organization uses a single platform to manage end-user client devices as well as mobile devices.

3. Ensure that guest accounts are disabled and default administrative accounts are disabled or renamed.

Default accounts on client systems can be exploited by outside threat actors to gain additional access or privilege in the environment. These accounts either should not be present or should be renamed in most business settings.

How

In Windows, you can use a PowerShell command to list local users and verify that the Administrator account and the Guest account are not enabled.

In the search bar, type **powershell** and select the Windows PowerShell app in the pop-up menu. When the shell appears, execute `get-localuser` as shown. You can see in the example output that the default Administrator and Guest accounts are disabled.

```
Windows PowerShell
Copyright © Microsoft Corporation. All rights reserved.
PS C:\Users\Mike> get-localuser
Name         Enabled   Description
----         -------   -----------
Administrator    False       Built-in account for administering the computer/domain
DefaultAccount    False       A user account managed by the system.
Guest        False       Built-in account for guest access to the computer/domain
HostAdmin        True
Mike         True
PS C:\Users\Mike
```

For macOS, you check the status of the Guest account using the Users & Groups icon in System Preferences. The Guest account should be listed along with other accounts on that system. If the account is listed as Off, then the Guest account is disabled. While reviewing this pane, you can also determine the privilege level of the current user, who could be a Standard user or an Admin user.

Checking the status of the "root" account, which has complete power over a Mac system, is more involved. Apple has provided steps to enable or disable the root user at https://support.apple.com/en-us/HT204012/. You can use these steps to verify the current status of the root account. By default, Apple disables the root account in macOS. You should discuss the management of the macOS root account with your client team, particularly if end users are allowed to have administrator rights on a system. Those rights can be used to enable the root user on demand.

4. Ensure that user accounts are provisioned through a centralized process and review policies around existence and use of local accounts.

The use of centralized account provisioning and management simplifies many processes related to the creation, maintenance, and deletion of end-user accounts. Central account management also simplifies application and web logins, as a single sign-on system can be leveraged to allow a user's identity to access many different parts of the environment.

Local accounts are difficult to manage and can create challenges in enterprise environments. A local account exists only on a specific system but could have privileges that increase the risk in the environment. Since local accounts aren't tied to a central identity system, it can be difficult for forensics teams to attribute actions taken using that account to any single individual. This type of attribution can be important in legal situations or when dealing with security incidents.

How

Review the end-user account process with the client administration team. Discuss how accounts are created and assigned to client systems. For many companies, end-user accounts for both Windows and Mac systems are managed via Active Directory and are linked to enterprise identity management systems. This keeps account records in a central location and allows accounts to be created when employees are hired and to be disabled or deleted when employees leave.

Review policies related to local accounts on end-user systems and discuss with the client administration team. If local accounts are permitted, check whether an inventory is kept that also records why each account is needed.

5. Review processes related to administration and remote support of client systems, ensuring that administrators use named accounts.

Client support teams are vital to the operation of the client environment, maintaining systems so that employees can be productive. To perform these tasks, support personnel usually have elevated privileges. This step ensures that client support teams have reasonable processes around their use of these privileges.

How

Interview the client administration team and a helpdesk team member or supervisor. Review the processes used when support teams need to connect to client systems. Determine if the end user must give consent or receives some visual notification when a remote support technician connects to a running session. You can easily verify this by asking a helpdesk team member to walk through the process of connecting to your system and by observing the resulting steps.

Assess processes for handling employee accounts and passwords when a system must be physically taken by the support team for repair. A best practice situation might include the following safeguards:

- The employee does not give any passwords to the technician team.
- The employee's active session is locked or logged out or the system is fully shut down before the support team takes possession of the system.
- The technician uses a unique, identifiable account to perform all maintenance tasks.

While support technicians usually have a high level of privilege on client systems and could use this power to access sensitive data, using a unique account provides additional protection and traceability for forensics teams in the event of a problem.

6. Review the device backup process, ensuring that restoration processes have been tested adequately.

Data from a system backup may be needed for various reasons. This step ensures that the organization has a suitable, proven backup and restore process for client data.

How

Interview the client backup administrator. Discuss practices around backup frequency, missed backups, and methods used to ensure that all in-scope clients are backed up on a regular basis. In most business environments, weekly backups are standard practice, while modern data backup solutions may record file changes in near real time.

Discuss processes for validating backup data and verifying that restore processes are functional. In a larger organization, file restoration may be exercised quite regularly when systems are upgraded or repaired, but in smaller groups, this may be less common.

7. Review the software licensing process and ensure users do not have access to unlicensed software.

Organizations run the risk of incurring large financial penalties for exceeding licensing agreements or using unlicensed software. This step ensures that the busi-

ness has appropriate safeguards to reduce license-related risks.

How

Interview the client administration team to determine how software licensing is managed for client systems. Some software may be licensed broadly for all employees (enterprise license or site license), while some may be limited to a set number of users or even to specific, named users. Many businesses are now provisioning internal "app stores" for client systems, giving users a one-stop shop for most of the software they may need.

Assess the processes for identifying and remediating software found to be out of license. For example, teams may compare a list of installed software to a list of licensed or allowed software and then may uninstall unexpected software remotely. Determine if the organization has processes addressing open-source software as well as freeware, shareware, and trialware, which may have differing license terms for business uses.

Some companies use special software installed on client systems to assist in software inventory or to provide additional controls on software usage. Through configuration, these packages can prevent or allow software installations based on company policy. These capabilities differ in various systems, but can be found in antivirus programs, system management clients, privilege management apps, security suites, and more.

8. Ensure that the organization has a sound process for responding to user problems.

Failure to establish ownership and tracking of end-user issues could result in lost productivity, unresolved security issues, and other risks.

How

End-user issues should be tracked through a trouble ticketing system. An owner for these issues should be assigned, and a group should be held responsible for tracking the progress to closure for any tickets opened because of client issues. Discuss these processes with the administrator or helpdesk supervisor.

9. Review and evaluate the strength of passwords and the use of password controls on client systems, such as password aging, length, complexity, history, and lockout policies.

All accounts should have passwords. The methods used to test these controls depend on the password-provisioning process and controls enabled on the client systems. Some of these controls may be configured centrally, such as in Active Directory. At a minimum, you should review system settings that provide password controls. Password controls are essential to enforcing password complexity, length, age, and other factors that keep unauthorized users out of a system. Many organizations choose to assign more stringent password settings to privileged accounts, such as those with administrator rights.

How

In Windows, you can find the account policies as they affect your system by using the `secpol` command from the search bar. This opens the Local Security Policy panel. From here, choose the Account Policies tree, and examine the listings for Password Policy and Account Lockout policy. You can also see much of the same information at the command line using `net accounts`. For Windows clients joined to an AD domain, these policies are usually set remotely via GPO and should be identical for all client systems. Systems not joined to a domain may not have any password or lockout policies.

For macOS systems linked to AD, password settings will be managed remotely. Systems controlled through a management system like Jamf Pro may have settings applied through that system. Discuss the settings with the client administration team. Unmanaged systems may not have any password or lockout policies.

For either operating system, verify that the policies listed in Table 14-1 are set in accordance with your local policies. Some common settings are listed.

Table 14-1 Account Policies

Policy	Setting
Minimum password age	1 day
Maximum password age	30–90 days
Minimum password length	8–14 characters
Password complexity	Enabled
Password history	10–20 passwords remembered
Store passwords using reversible encryption	Disabled, if possible, but understand and test this before making this decision
Account lockout duration	10–30 minutes
Account lockout threshold	10–20 attempts
Reset account lockout after	10–30 minutes

10. Ensure that end-user administrative privileges are established and maintained according to company policy.

By limiting end-user privileges, companies can reduce their risks and costs associated with rogue software installations, malware outbreaks, and unauthorized system configuration changes.

How

Interview the client administration team and ask whether end users are permitted to have local administrator rights on their systems. The administration team should be able to explain the standard posture and should be able to produce statistics or other reports listing which, if any, end users are permitted to have administrator rights. In many larger organizations, particularly in regulated industries, employees are not permitted any local administrator access. Some companies use commercial software to enable certain administrative features for standard user accounts.

If end users are not permitted administrator rights, you can verify the state of an account using a typical workstation. Select a user who should not have administrative access at random or from your workgroup (in many situations you can test this on your own workstation).

In Windows, open the Control Panel by typing **Control Panel** in the search bar and selecting the Control Panel desktop app in the pop-up. Select User Accounts. The account name of the logged-in user should appear toward the right of the resulting pane. If the user is an administrator, the word "Administrator" will appear

under the account name.

In macOS, open the System Preferences app. Often this is present in the dock, or you can use the Spotlight search bar. After opening System Preferences, select the Users & Groups icon. The resulting pane will show the list of users on the system. Below the usernames will appear the account type, either Admin or Standard.

11. Ensure that a legal warning banner is displayed when connecting to the system.

A legal logon notice is a warning displayed whenever someone attempts to connect to the system. This warning should be displayed prior to actual login and should say something similar to this: "You're not allowed to use this system unless you've been authorized to do so." Verbiage of this sort may be needed to prosecute attackers in court.

How

Log in to your system and determine whether a warning banner is displayed. If your organization permits remote access to client systems using a technology such as Remote Desktop or VNC, log in from another system using those as well and look for a warning banner. Interview the system administrator to determine whether the verbiage for this warning banner has been developed in conjunction with the company's legal department.

12. Verify that systems use a full-disk encryption (FDE) utility to protect company data.

Laptops are lost or stolen every day. If FDE is not in use, anyone in possession of a laptop or desktop can easily extract the data from the drive.

How

In macOS, you can check the status of FileVault, Apple's built-in disk encryption utility, using the System Preferences app and selecting the Security & Privacy icon. The FileVault tab shows the status of the disk. If your organization uses a disk encryption utility for Mac other than FileVault, check with your client administrator on how to verify that encryption is active.

In Windows, you can review BitLocker encryption status using Control Panel. Type **control panel** into the search bar and select the Control Panel desktop app from the pop-up. If your organization uses BitLocker encryption, you should see an option for BitLocker Drive Encryption in the list. Selecting this will provide the status for that system's drive. If your organization is using an alternative disk encryption product, discuss with your system administrator. Most alternative tools have a status icon in the system tray or a resident app to provide information about encryption status.

Your administrators should be able to provide metrics or reports as evidence to demonstrate the compliance of encryption throughout the in-scope fleet. If such metrics or reports are not available, the organization may not know which systems are not encrypted.

While discussing disk encryption with the administration team, you should review processes for storing encryption keys for maintenance or forensic needs. Security teams often require access to disk encryption keys during investigation work.

13. Determine whether the client is running a company-provisioned antivirus program.

Client systems are a primary target for outside hackers. Failure to have basic antivirus protection makes clients an easier target and may allow harmful code to run on the system. Antivirus tools can also identify the presence or actions of hacking tools run by malicious actors. In Windows 10, Windows Defender is installed and enabled by default, but some organizations will use an alternative antivirus program.

How

In Windows 10, you can access the status of security modules, including antivirus, by typing **security** in the search bar and selecting the Security and Maintenance app in the pop-up menu. Select the Security drop-down to see the status of security features. In the example shown in Figure 14-1, Windows Defender is listed under the Virus Protection entry.

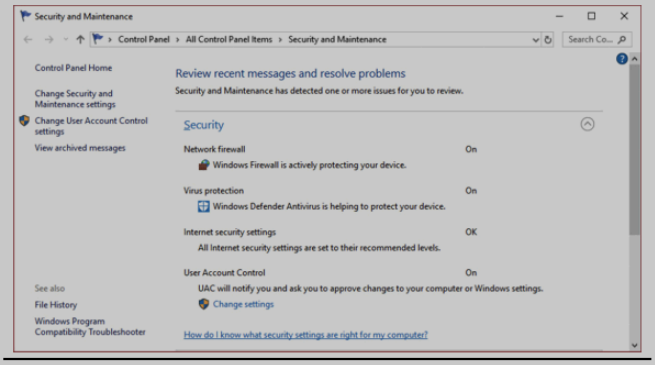


Figure 14-1 Windows Security and Maintenance Panel

Windows also offers the Windows Defender Security Center, which provides a similar view. Either will provide the status of antivirus and other tools. In later versions of Windows, some status information can be hidden from end users through system management configuration. If this is the case, discuss with your client administration team.

Mac clients should also use antivirus software, but macOS does not provide a simple tool to see the status of antivirus tools. Discuss with the client administration team to obtain information about the macOS antivirus software in use in your organization.

Depending on the nature of your audit, you also might want to check the configuration of the antivirus program on the client. For many organizations, the antivirus program is managed centrally, and enterprises will often use the same solution for both Windows and Mac platforms. Ideally, the configuration should not exclude any files or folders from periodic scanning and should be set to protect the system in real time for all file operations. Antivirus tools should also be configured to automatically download and install signature updates. Deviations can put the system at greater risk.

14. Verify that a client firewall is active and review firewall management practices.

Client firewalls protect the system from unexpected network connections. Failure to use a firewall can increase the risk of a malicious threat gaining access to a system.

How

For Windows, use the steps in the previous test to access the Security and Maintenance panel or, alternatively, the Windows Defender Security Center app. The panel indicates the status of the Windows firewall. If your organization uses a third-party firewall, such as those provided with many antivirus programs, the status should appear in the same Windows panels.

For Mac systems, select the System Preferences app, then choose the Security & Privacy icon. Switch to the Firewall panel. This will describe the status of the firewall.

Discuss the firewall configuration with the client administration team or the security team. Firewalls should be configured to block most inbound traffic.

15. Review client logging requirements and settings.

Appropriate client logging can assist operations and security teams in detecting issues with client systems.

How

In Chapter 4, we discussed the cybersecurity program, including policy requirements. Review policies related to logging to determine if requirements exist for client logging. Many companies choose to limit the use of centralized client log collection due to the number of clients, size of log data to be captured, or the potential for network impact from log transfers. If your company's logging policies have requirements for client systems, discuss with the client administration team. Ask to see the log configuration in the system management tool(s) in use. If logs are forwarded to a central system, ask the monitoring team to provide a sample of client logs.

16. Review the patching process for the operating system and key applications.

If applicable operating system and software patches are not installed, widely known security vulnerabilities could exist on the client, allowing harmful exploits from outside threats.

How

Discuss the client patching process with the administration team and the security team. Both Microsoft and Apple release patches for the OS and key applications periodically, as do many third-party software providers. The organization should have a process for assessing the applicability of patches and their resulting criticality. The teams should be able to provide supporting evidence of analysis for the most recent patch releases from either Microsoft or Apple at a minimum.

Discuss the timing of patch installation and the processes for ensuring that all relevant clients receive the expected patches. Companies should have reports or other metrics available describing the patch status of the client fleet.

To verify the metrics, select a client system at random—you can use your own system for this step if desired. In Windows, type **windows update** in the search bar. Open the Windows Update app, which will show the status of the system and provide a link to see recent installation history. The history should show recent installation activity, particularly under Quality updates. In macOS, use the About this Mac app, accessible via the Apple menu in the upper-left area of the screen. Select System report and then scroll down in the left-hand panel to find Installations under Software. You can select the Install Date column to sort the installation info, as depicted in Figure 14-2.

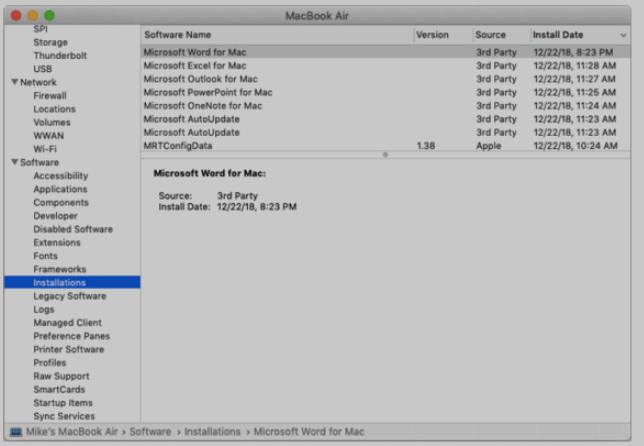


Figure 14-2 Viewing installed updates in macOS

Some organizations may enable automatic Windows and Mac updates for their systems, but others will manage updates carefully to ensure compatibility with company software. Discuss with the client administrators to determine if patches are installed from a central system or if clients update themselves using automatic updates.

You should pay particular attention to Windows 10 OS versions in use in the environment. With Windows 10, Microsoft moved to a semiannual release schedule for Windows, and when a new version is released, support for an older version is immediately discontinued. Depending on your company's licensing arrangement, you may have as little as 18 months of support for each new version of Windows before it is no longer supported or patched by Microsoft. See the Knowledge Base for additional information on Windows life cycles. Apple does not publish end-of-life information for macOS but in general supports security patches for the two prior releases.

17. Verify that the screen will automatically time out after a set interval and require a password to resume.

An unattended work session can be used by anyone walking up to the computer. A screen timeout reduces this risk.

How

In Windows 10, you can find the setting for screen timeout under the Screen Saver Settings application. Enter **screen saver** in the search bar and select the Change screen saver option in the pop-up menu. The panel displays the time duration. The box indicating "On resume, display logon" screen must be checked.

In macOS, several steps are needed to verify this item. First, open the System Preferences app. Next, select the Desktop & Screen Saver icon. Under the Screen Saver tab, you'll see a Start after field at the bottom. This should be set to a time matching your security policy, but preferably a brief duration, such as five or ten minutes. Next, click the back arrow in the same panel to return to System Preferences. Choose the Security & Privacy icon. Under General, the option for "Require password after sleep or screen saver begins" must be checked. Ideally, the drop-down box should be set for immediately or a very short duration. Power options can also affect this setting, but for most Mac deployments, the screen saver timer will be managed by the administration team and will represent the maximum timeout length.

Discuss with your client administration team to ensure that these options are enforced through the client management system.

18. Ensure that AutoPlay and AutoRun are disabled for removable devices.

In Windows, AutoPlay and AutoRun will automatically launch certain types of files upon the insertion of a USB drive, CD/DVD, or other media. If these features are enabled, a malicious file can execute without a user taking any specific action. Mac systems will not automatically execute files on USB drives, but systems with optical drives may play CDs or DVDs automatically.

How

Discuss with the client administration team to determine if AutoPlay and AutoRun are disabled. Both can be adjusted through GPO in Windows. For Mac systems, discuss with the Mac administrator. Most modern Macs are no longer shipping with optical drives, so this may not be an issue in some environments.

Tools and Technology

Most of the steps in this chapter are executed with basic GUI or command-line inputs in a working system. Many other tools are available, particularly for Windows, that will provide additional insight into the status of a client. Some of these are listed here.

Resource	Website
Microsoft Script Center	https://gallery.technet.microsoft.com/scriptcenter
Microsoft Command-Line Reference	https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands
Microsoft Sysinternals Tools	https://docs.microsoft.com/en-us/sysinternals/
Microsoft Security Compliance Toolkit	www.microsoft.com/en-us/download/details.aspx?id=55319

Knowledge Base

The following table lists additional resources where you can obtain information about Windows and Mac client environments and controls. Both Microsoft and Apple provide extensive online information regarding their platforms, and the Internet community has supplemented this with additional content. In addition, the Center for Internet Security (CIS) has developed hardening guides for both Windows and Mac client systems that contain hundreds of configuration options. The CIS guide for Windows 10 alone is over 1,000 pages!

Resource	Website
Windows 10 Reference	https://docs.microsoft.com/en-us/windows/windows-10/
Microsoft TechNet	https://technet.microsoft.com/en-us/
Microsoft System Center	www.microsoft.com/systemcenter
Windows Intune	www.microsoft.com/en-us/cloud-platform/microsoft-intune
Windows Lifecycle Fact Sheet	https://support.microsoft.com/en-us/help/13853/windows-lifecycle-fact-sheet
Windows Security Baselines	https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-security-baselines
macOS Security Overview	www.apple.com/business/resources/docs/macOS_Security_Overview.pdf
macOS Security Checklist	www.jamf.com/resources/white-papers/macos-security-checklist/
The Center for Internet Security	www.cisecurity.org
Computer Security Resource Center	http://csrc.nist.gov

Part 2: Auditing Mobile Devices

While laptops and other portable form factors like Microsoft's Surface line are mobile in the sense that they are easy to move, most organizations consider "mobile devices" to include smartphones, tablets, and other devices running Android, iOS, or another mobile-centric operating system. This section describes concepts and steps to consider when conducting an audit of an organization's mobile device environment.

Mobile Device Auditing Essentials

Conceptually, auditing a mobile device environment that includes smartphones

and tablets is fairly similar to auditing desktops and laptops. There are policies, device management systems, minimum security standards, and so on. As mobile management technologies converge with client management systems, the overlap between these two spaces is likely to increase. However, the complication of personal device ownership is a larger factor in the mobile space; this, along with a few differences in technology, suggests a separate audit is warranted.

There are a few basic questions you should consider as an auditor when assessing a mobile device program. Some of these include

- What company resources can be accessed by mobile devices?
- How is company data protected on mobile devices?
- How does the company handle personally owned devices and apps?
- Is the company considering risk from emerging mobile devices like wearables?

Companies may address mobile devices in a number of different ways. Some companies will purchase devices and carrier plans for employees, while others will pay for carrier plans but not the devices themselves. Some offer a stipend to employees who are free to select their own devices and plans.

Understanding your company's posture and policies around mobile device ownership and authorization for those devices to access company resources is a key element to a mobile device audit. In the simplest case, the company owns all devices and does not permit employees any personal use of the device. While some businesses may manage a subset of devices like this—for example, a shipping company may have special-use devices for delivery personnel—it's rare for all mobile devices to fall into this category.

As in the client audit, you'll need to review several policies in preparation. In addition to policies on noncompany devices and acceptable use, your company's policies should address mobile devices. In some cases, this could be merged with a client security policy or other document, but in many companies this is a separate policy. If a company allows personal mobile devices to access company data via a mobile access gateway, there can be implications for employee privacy and employee personal data; this usually drives a need for a separate policy or even a consent agreement for mobile devices.

Let's briefly establish some common terminology and define some industry terms. The technologies used to configure and secure mobile devices emerged under the term "mobile device management," or MDM. As mobile applications grew, technology providers added many app-related features and developed the mobile application management (MAM) field. After throwing in a few more acronyms, much of the industry settled around the more generic "enterprise mobility management," or EMM. As EMM technologies begin to converge with traditional client system management systems, yet another term, "unified endpoint management" (UEM), is emerging. For the purposes of this chapter, we'll mostly use the EMM terminology.

Test Steps for Auditing Mobile Devices

1. Review company policies around mobile devices and ensure device ownership and user responsibilities are covered.

As we've discussed, establishing policy around device ownership is a key step for companies dealing with client systems. This is also important for mobile devices. This step ensures your organization has considered these issues and has provided appropriate information to end users.

How

Obtain a copy of your company's mobile device policy, if it exists, as well as policies related to the use of personally owned or non–company-owned devices. The policy should clearly define how mobile devices may be used and how personal mobile devices should be handled. You should expect a mobile device policy to address what resources may be accessed by mobile devices, any minimum standards or settings for the device, how the employee and company will respond if the device is lost or stolen, and expectations placed on the employee. If your company permits personally owned devices, the mobile policy should also address privacy-related concerns around device location, personal apps, and personal expenses.

You should verify that policy language or employee agreements around privacy have been developed in conjunction with your company's legal and/or privacy teams. While this audit does not address privacy assessments related to various regulations around personal data protection, you should be aware that mobile device systems are often in scope for this kind of review.

In addition, you should review your company's policies around user expectations. For expectations around mobile devices, this may be covered in a mobile device policy or in an employee agreement, but the AUP may also be relevant, particularly if the mobile devices are company owned.

Some organizations may allow personal mobile devices to connect to special external networks that cannot access internal company resources. Similar in concept to a coffee shop network, these external networks might be provided as a convenience for both employees and visitors. For AUP purposes, these networks should be thought of as company networks unless separately addressed in the policy.

2. Ensure the organization has an EMM infrastructure commensurate with policy goals and company strategy.

Enterprise mobility management systems allow organizations to track device inventory, manage software, apply security policies, and more. This step assesses the existence and scope of EMM systems. If device management is not in place, the company may not be able to properly secure and protect company assets and data.

How

Discuss the EMM systems with the mobile device administrators. Some common commercial products in this area include AirWatch, MobileIron, and Intune. Both Google and Apple closely control how EMM systems interact with devices running Android and iOS, and the basic capabilities for most EMM solutions are very similar. You should expect that any solution should, at a minimum, be able to facilitate the following key functions:

- Provide visibility to the mobile device inventory for the company
- Provide information on the device type and operating system version
- Support application controls, including the ability to block specific applications

- Apply security and system configuration policies to managed devices
- Remotely wipe devices when lost, stolen, or no longer needed

Many EMM systems also facilitate access to company resources through network configuration or through a gateway device. In a typical configuration, mobile devices managed under EMM are permitted to access corporate e-mail, contacts, and calendar information, frequently via Microsoft Exchange protocols. The same systems that allow this kind of access may also allow access to other on-premises or cloud-based applications or services. You should review the architecture and configuration of these systems with your company's mobility team to ensure that the controls around device access are working as intended to protect company data. We'll discuss data protection controls in more detail later in this chapter.

As with other company infrastructure, you should also review access policies for the EMM system itself, ensuring that only designated individuals have access. You should also take this opportunity to review change management and business continuity aspects of the EMM environment. Finally, verify that the EMM system, gateways, and other components of the mobility environment are kept up to date with supported software and patches. The mobile device administration team should be able to assist with these items.

3. Ensure that mobile devices are configured to require a PIN or passcode to gain access to the device and review other PIN/passcode-related settings.

If access to mobile devices is not protected, company data could be exposed if an unauthorized person gains possession of a device.

How

You can discuss this and all of the following steps in this chapter with your mobile device administrators to obtain information about the expected behavior in each step. Discuss the passcode requirements, including minimum passcode length, passcode change requirements, support for alternative unlock methods such as facial recognition, and auto-wipe features. These will differ by company according to risk tolerance, but a standard practice is to require a six-character passcode with annual change, to automatically wipe the phone after ten failed passcode attempts, and to allow facial or fingerprint authentication.

To verify that the policy requirements are implemented, you can use any managed mobile device. First, you can verify that a managed device does, in fact, require a passcode to unlock it. To verify that the passcode cannot be removed, you can check additional settings. For iOS devices, access Settings, then Touch ID & Passcode or Face ID & Passcode, depending on the device model. Scroll down to Turn Passcode Off. This should be grayed out and not selectable for a managed device. For most Android devices, open the Settings app, then select Device, then Lock Screen, then Screen Security. Options to remove the different passcode methods, including pattern or PIN-based unlock, should be grayed out for managed devices.

Different models of Android-based devices may have slightly different menu options; if you are unable to find the settings, discuss with your mobility team.

4. Ensure that device encryption is enforced.

As with client devices, an unencrypted mobile device could reveal sensitive information to someone who gains physical possession of the device.

How

Most EMM solutions can verify device encryption status and permit or deny access for devices based on the result. Discuss this with your mobility administrators.

Since the release of iOS 8, Apple has included device encryption by default for iPhone and iPad. The only prerequisite is to set a passcode for the device. You can also verify the state of the device in the Touch ID & Passcode panel. Scroll to the bottom, where the screen should indicate "Data protection is enabled."

For Android, encryption is also a default setting beginning with the Nougat release. However, this can be disabled for some versions of Android. Open the Settings app and select Security. An Encryption option should indicate whether the device is encrypted. Earlier versions of Android and some lower-performance models did not support data encryption; these versions and devices should not be permitted to store corporate data.

5. Ensure that devices automatically lock after a set period.

An unattended device could be accessed by anyone without the need of a passcode or other authentication. A screen lock timer can reduce the risk of data theft or other malicious activity.

How

Discuss with your mobility team to ensure that this setting is configured in EMM policy for managed mobile devices.

To verify this setting in iOS, you can check the Settings | Display & Brightness panel and review the Auto-Lock setting. EMM administrators can set a maximum allowable time, and the user can select a shorter time if desired. The Never option should not be available.

In Android, the screen timeout is managed under Settings, then Display. The Sleep setting indicates how long the screen will remain on.

6. Review processes for keeping mobile devices up to date.

Obsolete or unpatched operating systems can expose security vulnerabilities that could be leveraged to access sensitive personal or company data or company e-mail accounts. This step verifies that the organization has a plan for maintaining a mobile device security posture.

How

As vulnerabilities are announced regularly in both iOS and Android, Apple and Google frequently release security updates. In addition, both companies typically release a new version of their OS each year. However, upgrade paths for their devices differ. The closed nature of the Apple environment (Apple makes all iPhones, iPads, iPods, and the iOS operating system) means that Apple determines which versions of hardware are compatible with which versions of software, and compatible devices can be updated quickly upon release of a new iOS version. For Android, however, which historically allows device makers and carriers to customize the OS for their needs, upgrades and patches take much longer to deploy, and in many cases, patches and upgrades are not issued even for very recent phone or tablet models. You should discuss the risks of the various operating systems with your security team and determine if the mobility team has implemented restrictions on which versions are allowed to connect to company resources.

Apple does not release patches for previous iOS versions but historically has supported newer iOS versions on devices that are several years old. This allows a fleet with a wide range of Apple devices to run the same, up-to-date software release. Companies therefore have more flexibility with Apple systems to limit the allowed versions without negatively affecting most users. For risk-averse organizations, a best practice is to allow only the currently released version of iOS to connect, with a reasonable grace period for employees to upgrade to the latest release. More risk-tolerant organizations may allow older iOS versions to connect but should have processes to review the specific vulnerabilities and risks present in older iOS releases. Your mobility team should be able to provide an inventory of managed Apple devices by iOS version.

For Android devices, practices vary widely by company. Some organizations attempt to allow only recent Android versions to access company resources; this may require either the company or the end user to upgrade devices frequently. As some Android device makers have a better track record at releasing updates than others, some companies have also taken the approach of restricting which device models are allowed. For example, Google's Pixel models run a "clean" version of Android that can be updated quickly—some organizations may choose to allow only these models. Your mobility team should be able to provide an inventory of managed Android devices by OS version.

7. Review processes for erasing or reclaiming devices in the event one is lost, stolen, or replaced or if the employee is terminated.

When a device is lost, stolen, or replaced, it may still contain sensitive data. Although other safeguards intended to protect this information may be in place, companies should have processes to safely retire devices and remove company data.

How

Discuss the steps involved in the case of a lost or stolen mobile device. People who lose a device may be embarrassed or may fear disciplinary action or financial penalties for losing a device, but it's important that employees can freely report a lost or stolen device to the appropriate personnel. Such reports may come to physical security teams, information security teams, or mobility teams, but the organization should have a process for notifying the mobility team that a device is lost.

EMM solutions support remote wipe features for lost or stolen devices. These can erase the entire device or just remove the company data and configuration. In most situations, it is better to wipe the device entirely if it has been lost or stolen. A remote wipe is not always successful, but it's an essential step. If the device is powered on and within range of a carrier signal or connected to an Internet-capable wireless network, it should receive the wipe signal.

Companies should also have processes for handling employee terminations. For company-owned devices, ensure that employee exit checklists include retrieval of the device. In the case of employee-owned devices, employee termination processes should include a partial or full device wipe. A partial wipe removes only company data and applications, leaving personal data intact. Ideally, these processes are automated and linked to other termination steps.

If you have access to a list of employees who have recently resigned, retired, or have been terminated, you can sample this list and compare against the list of active devices in the EMM system. Consult with your HR contacts and the EMM team.

8. Review additional options for protection of company data on the device.

This step covers additional data protection considerations for mobile devices in typical business deployments.

How

The increasing power of mobile devices and their increased penetration in business environments means that additional configuration areas should be considered to provide adequate controls to protect company information.

Access to company e-mail, calendar, and contacts has been the most popular use case for mobile device access for many years. In the past, this usually involved special network access from devices to on-premise e-mail systems like Microsoft Exchange. However, as cloud-based e-mail systems have expanded, many companies no longer have on-premise e-mail systems. You should review the configuration of e-mail access for mobile users, particularly if your company uses a cloud system like Office 365 or Google's G-Suite. Ensure that any provisions for employee termination also include termination of their mobile access to cloud-based e-mail systems.

Many organizations have deployed one or more mobile apps for business use. In some cases these are commercially purchased apps, like expense managers, HR applications, and the like. In other cases these are internally developed, custom applications supporting a specific business function or process. For any use of mobile applications for work, you should discuss how company data is protected. In best-practice situations, mobile applications used for company business are managed and deployed through the EMM system. This allows the company to revoke access to the application and its data if needed and to establish controls around what can be done with the data on the device.

Mobile devices may have access to or may store sensitive company data. Apple and Google have gradually added more and more business-friendly capabilities to their operating systems, allowing more isolation between company and personal data and inserting more control options. In late 2018, Google released a work profile feature for Android, which creates a separate application landscape for work-related apps and content. Apple offers restrictions to prevent opening company e-mail content in unmanaged applications, as well as settings to add virtual private network (VPN) connections and content protection for company-designated websites. Many options are available; discuss with your security team and mobility team to ensure that proper attention has been given to protecting company data on mobile devices.

Additional Considerations

Licensing

Mobile applications may have differing license terms than traditional desktop applications. While many applications are freely available on app stores, a review of licensing terms may show that they are not authorized for business usage. This can create a challenge for software management teams. Since mobile devices are more likely to be personally owned, a device with mixed use may have business and personal applications present. An employee who uses a personal application for a business function may unknowingly violate the terms of use of the applica-

tion and put the company at risk of financial penalties.

Closely managing mobile applications on employee-owned devices opens a number of privacy and legal issues. Managing applications on company-owned devices is more straightforward. Providing education to end users about the risks of using unauthorized applications for business is a best practice.

Higher-Security Environments

The tests and guidelines in this chapter apply to typical business settings, and you should expect to find most, if not all, of the controls described here to be in place in all environments. However, risk-averse organizations, those with high regulatory burdens, and those with very high security needs should consider deploying additional controls. The Center for Internet Security (CIS) has developed hardening guidelines for both Android and iOS-based devices. If you find that your organization's controls are not sufficient to mitigate company risks, consider reviewing the CIS guidance for mobile devices. See the Knowledge Base for more information.

Wearables and the Internet of Things (IoT)

While this chapter has focused on traditional end-user technology, new uses for computing capability have emerged over the last five years. Wearable technology, including smart watches, includes a class of systems that often pair with a phone to offload data or provide interactive capability. Some of these devices can also connect to wireless networks. Watches in particular are interesting to enterprises because they can be used for authentication; for example, the Apple Watch can be used to unlock a Mac. Companies should be aware of the capabilities employees are bringing to the workplace in these devices.

In addition, hundreds of new devices are being released each month in the IoT space. These include various kinds of sensors and control systems like thermostats, switches, automation systems, and more. Often used in industrial or facilities-related situations, there are also devices like Amazon's Echo series and other Alexa-compatible devices that may appear on your network. Many companies do not address these devices in existing policies, but they should be aware of the potential risks of having these unmanaged systems on corporate networks.

Tools and Technology

Tools	Website
VMware AirWatch	www.air-watch.com
MobileIron	www.mobileiron.com
Microsoft Intune	www.microsoft.com/en-us/cloud-platform/microsoft-intune

Knowledge Base

Resource	Website
Apple iOS Security Overview	www.apple.com/business/resources/docs/iOS_Security_Overview.pdf
Apple iOS Security Guide	www.apple.com/business/site/docs/iOS_Security_Guide.pdf
Android Security	https://source.android.com/security
Center for Internet Security	www.cisecurity.org
Department of Homeland Security study	www.dhs.gov (search for study on mobile device security)

Master Checklists

The following tables summarize the steps listed for auditing Windows and Mac clients and Android and iOS mobile devices.

Auditing Windows and Mac Client Systems

Checklist for Auditing Windows and Mac Client Systems
❑ 1. Review company policies around client devices and ensure device ownership and user responsibilities are covered.
❑ 2. Ensure the organization has a device management infrastructure commensurate with policy goals and company strategy.
❑ 3. Ensure that guest accounts are disabled and default administrative accounts are disabled or renamed.
❑ 4. Ensure that user accounts are provisioned through a centralized process and review policies around existence and use of local accounts.
❑ 5. Review processes related to administration and remote support of client systems, ensuring that administrators use named accounts.
❑ 6. Review the device backup process, ensuring that restoration processes have been tested adequately.
❑ 7. Review the software licensing process and ensure users do not have access to unlicensed software.
❑ 8. Ensure that the organization has a sound process for responding to user problems.
❑ 9. Review and evaluate the strength of passwords and the use of password controls on client systems, such as password aging, length, complexity, history, and lockout policies.
❑ 10. Ensure that end-user administrative privileges are established and maintained according to company policy.
❑ 11. Ensure that a legal warning banner is displayed when connecting to the system.
❑ 12. Verify that systems use a full-disk encryption (FDE) utility to protect company data.
❑ 13. Determine whether the client is running a company-provisioned antivirus program.
❑ 14. Verify that a client firewall is active and review firewall management practices.
❑ 15. Review client logging requirements and settings.
❑ 16. Review the patching process for the operating system and key applications.
❑ 17. Verify that the screen will automatically time out after a set interval and require a password to resume.
❑ 18. Ensure that AutoPlay and AutoRun are disabled for removable devices.

Auditing Mobile Devices

Checklist for Auditing Mobile Devices

- ❑ 1. Review company policies around mobile devices and ensure device ownership and user responsibilities are covered.
- ❑ 2. Ensure the organization has an EMM infrastructure commensurate with policy goals and company strategy.
- ❑ 3. Ensure that mobile devices are configured to require a PIN or passcode to gain access to the device and review other PIN/passcode-related settings.
- ❑ 4. Ensure that device encryption is enforced.
- ❑ 5. Ensure that devices automatically lock after a set period.
- ❑ 6. Review processes for keeping mobile devices up to date.
- ❑ 7. Review processes for erasing or reclaiming devices in the event one is lost, stolen, or replaced or if the employee is terminated.
- ❑ 8. Review additional options for protection of company data on the device.

CHAPTER 15

Auditing Applications

Each application is unique, whether it supports financial or operational functions, and therefore each has its own unique set of control requirements. It is impossible to document specific control requirements that will be applicable to every application. However, in this chapter, we will describe some general control guidelines that should be pertinent to any application, regardless of its function, programming language, and technology platform. The following topics are discussed in this chapter:

- Essential components of application audits
- How to drill down into possible issues with frameworks and key concepts
- Detailed steps for auditing applications, including the following:
  - Input controls
  - Interface controls
  - Audit trails and security monitoring
  - Account management
  - Permissions management
  - Software change controls
  - Backup and recovery
  - Data retention and classification and user involvement

Background

Business applications systems, or applications for short, are computer systems that are used to perform and support specific business processes. Your company likely has dozens of applications, each used to perform a particular business function, such as accounts receivable, purchasing, manufacturing, customer and contact management, and so on. Most of these applications have interfaces that allow end users to interact with and enter data into the systems, although some may consist purely of offline (batch) processing.

These applications may be systems that were purchased from an external vendor (for example, many companies use an enterprise resource planning [ERP] system such as SAP ECC [ERP Central Component] to perform their core financial functions), or they may be homegrown (that is, applications developed specifically

by your company for use within your company). Applications can range in size from an enterprise system that is accessed by every employee to a small client application accessed by one employee. Obviously, your audits will tend to focus on those larger applications that support critical business processes, but each application will need to be considered individually when you perform risk ranking and determine what to audit.

Each application has its own control nuances, depending on the business process it supports, the programming language that was used to develop it, and the technology platform(s) on which it resides (for example, the database management system, middleware, and operating system used). Although it is not realistic to provide detailed test steps and checklists for every possible permutation of an application, this chapter provides guidance on control concepts that are common to almost all applications and that can be used to generate thoughts and ideas regarding audit test steps more specific to the application being audited.

Staying on top of every new technology that attaches itself to your environment is tough. It's our job as auditors to drill down quickly into new or existing applications to find potential control weaknesses. We will therefore discuss how to examine applications conceptually using big-picture and abstract frameworks. We also will suggest a comprehensive set of checklists that will greatly assist you in covering the vast majority of common control weaknesses.

NOTE Chapter 9 contains test steps specific to auditing web-based applications, which can be used in conjunction with the standard application auditing test steps in this chapter.

Application Auditing Essentials

In a perfect scenario, you have a perfect audit program that you can apply quickly to your perfect application. However, although the test steps in this chapter will serve as a great starting point, in reality you're often faced with new ideas and approaches for solving business problems with new technology, all of which get bundled together to create a unique application that requires a unique audit program. As you struggle with the questions to ask, there are some general frameworks and best practices that you will find helpful. Examples have been included in Chapter 18 and, rather than duplicate them here, please reference the information contained in that chapter.

Test Steps for Auditing Applications

The following steps generally refer to controls specific to the application and do not address controls, for example, at the level of the network, operating system, and database management system. Refer to other chapters of this book for test steps for those topics and also consider the frameworks and concepts described earlier in this chapter as you approach developing the audit program for your application.

NOTE The audit steps in this chapter are written from the standpoint of auditing an application that has already been developed and implemented. See Chapter 17 for additional steps to be performed when auditing an application during the development process.

Input Controls

1. Review and evaluate controls built into system transactions over the input of data.

As much as possible, online transactions should perform up-front validation and editing to ensure the integrity of data before it is entered into the system's files and databases. Invalid data in the system can result in costly errors. It is preferable and much more cost-effective to catch a data entry error prior to that data being entered into and processed by the application. Otherwise, the error may not be caught at all, may only be caught after it results in system disruption, or may only be caught after time-consuming manual reconciliation procedures, and so on.

How

Verify that invalid data is rejected or edited on entry. You will need to understand the business function being supported by the system and the purpose and use of its various data elements. This likely will require discussion not only with the developers but also with the end users. Once you understand the purpose of the system and its data, you can think through the various data integrity risks associated with the application. In some cases, a code review may be appropriate if the developers are available and the auditor is a knowledgeable coder. Poorly written, commented, or formatted code is often a red flag that suggests that a deeper review is needed. If possible, obtain access to a test version of the system that mirrors the production environment and attempt to "break" the system by entering invalid data to see whether it is accepted by the application.

Following are some basic examples of good data input controls:

- Fields that are intended to contain only numbers should not allow entry of alphanumeric characters.
- Fields that are used to report such things as dates and hours should be set up either to require input in the correct format (such as *MMDDYY* or *HHMM*) or transform input into the correct format.
- Where applicable, transactions should perform "reasonableness" and "logic" checks on inputs. An example would be preventing users from reporting labor of more than 24 hours in a day or more than 60 minutes in an hour. Another example would be disallowing entry for time, costs, and so on for an employee who has been terminated or who is on leave. Or consider a transaction used by ticket agents to record how many seats were sold on a flight and the number of no-shows. The transaction should not allow the agent to input numbers indicating that there were more no-shows than seats sold.
- When a finite number of valid entries are available for a field, entries that are invalid should not be allowed. In other words, input screens should validate such things as cost centers, account numbers, product codes, employee numbers, and so on against the appropriate database(s) or file(s).
- Duplicate entries should not be allowed for data that is intended to be unique. For example, the transaction should not allow a product code to be added to the product database if that code already exists in the

database.

- Each input screen generally has certain fields that are required for the transaction to be processed accurately. Execution of a transaction should not be allowed until valid data is entered into each of those fields.
- Where applicable, transactions should perform "calculation" checks on inputs. For example, the system should ensure that journal entry credits and debits balance to zero before processing a transaction. Another example would be a labor entry system where hours charged for the week need to add up to at least 40.
- Programmed cutoff controls should be in place to help prevent users from recording transactions in the wrong period. For example, the screen should not allow users to record transactions in prior accounting periods.
- A user should be prevented from updating his or her own personal data in some systems. For example, a user, regardless of his or her access level, should not be allowed to change his or her own pay rate or vacation accrual rate.
- Database operatives (such as `*`, `=`, `or`, `select`) should be disallowed as valid input, as they can be used to disrupt or retrieve information from the database.

2. Determine the need for error/exception reports related to data integrity and evaluate whether this need has been filled.

Error or exception reports allow any potential data-integrity problems to be reviewed and corrected when it's not feasible or practical to use input controls to perform up-front validation of data entered into the system. For example, although it may not be inherently wrong for an employee to enter 80 hours of overtime for one week into a labor system, this sort of unusual event should be captured on a report for review by the appropriate level of management.

How

Discuss the application's error and exception handling with the developer or administrator. Based on the results of the analysis from step 1, look for opportunities for additional data integrity checks (which may not have been feasible to perform with "hard" up-front input requirements). Again, discussions with the end users can be very helpful here. Ask them what sorts of reporting would be helpful for them in catching anomalies and errors. For any error and exception reports that do exist, look for evidence that those reports are being regularly reviewed and appropriately handled.

Interface Controls

3. Review and evaluate the controls in place over data feeds to and from interfacing systems.

When an application passes and/or receives data to or from other applications, controls need to be employed to ensure that the data is transmitted completely and accurately. Failure to do so can result in costly errors and system disruption.

How

Determine what interfaces exist with the system you are auditing, including data flowing into and out of the system. These interfaces could be in the form of real-time data transmission or periodic transmission of data files via batch processes. Review system flow diagrams, review system code, and interview the application developer or administrator to obtain this information. Once you have a feel for the interfaces that exist, determine which controls are in place regarding those interfaces through code review and interviews with the application developer or administrator. Expect to see basic controls such as those discussed in the following paragraphs.

Control totals from interface transmissions should be generated and used by the system to determine whether the transmission completed accurately. If a problem is found, reports should be issued that notify the proper people of the problem. Some examples of control totals that may be applicable are hash totals (totals that have no inherent meaning, such as summing all account numbers or employee numbers in a file being transmitted), record counts, and total amounts (totals that do have inherent meaning, such as summing the total sales entered or salary paid in a file being transmitted). For example, prior to transmission, the sending system might generate a count of all records being sent. After transmission, the receiving system could generate a count of all records received. Those two numbers would then be compared. If they don't match, it would generate an error report, as this would indicate that some records were not received accurately. Another type of control total could flag missing record numbers when records are transmitted in a sequential fashion. All such methods are intended to detect instances when data from the sending system is not correctly received. If these controls do exist, review evidence that applicable error reports are being regularly reviewed and acted upon.

The system should handle items that did not transmit successfully in such a manner that reports and/or processes enable these items to be resolved quickly and appropriately, such as by placing them in a suspense file and generating reports of all items in the suspense file. Verify that any such suspense files and error reports are being reviewed and acted upon.

Data files that contain interface source or target information should be secured from unauthorized modifications. This may mean appropriate authentication controls, authorization controls, or encryption where necessary. Review the file security of any applicable files.

When it is not feasible to use control totals to verify accurate transmission of data, reconciliation reports should be generated that allow users to compare what was on one system with what was received on another system. If applicable, review evidence that reconciliation reports are regularly reviewed and acted upon.

Where applicable, data validation and editing, as described in the "Input Controls" section of this checklist, should be performed on data received from outside systems. Error/exception reports should be generated that allow any data integrity problems to be corrected, and those reports should be regularly reviewed.

4. If the same data is kept in multiple databases and/or systems, ensure that periodic sync processes are executed to detect any inconsistencies in the data.

Storing the same data in multiple places can lead to out-of-sync conditions that result in system errors. It can also have a negative impact on business decisions, as erroneous conclusions can be reached using inaccurate data.

How

Determine, with the help of the application developer or application administrator, where this sort of control is applicable and review for its existence and effectiveness. Ideally, one database or data file should be designated as the "master" for each data element, and other systems will reference the master location as opposed to keeping a separate copy of the data. Even if multiple copies of the data are kept, the location that represents the master copy should be designated so that the system can easily determine "who wins" in out-of-sync situations and perform automated corrections.

Audit Trails and Security Monitoring

5. Review and evaluate the audit trails present in the system and the controls over those audit trails.

Audit trails (or audit logs) are useful for troubleshooting and helping to track down possible breaches of your application.

How

Review the application with the developer or administrator to ensure that information is captured when key data elements are changed and key activities are performed. This information should include in most cases what activity was performed, the original and new values of the data (in the event of a data change), who performed the activity, and when it was performed. This information should be kept in a secured log to prevent unauthorized updates. The logs should be retained for a reasonable period, such as three or six months, to aid investigations into errors or inappropriate activities.

6. Ensure that the system provides a means of tracing a transaction or piece of data from the beginning to the end of the process enabled by the system.

This is important to verify that the transaction was fully processed and to pinpoint any errors or irregularities in the processing of that data.

How

Review the application with the developer or administrator and evaluate the existence of this ability. Identify a sample of recent transactions and attempt to trace them through the system's various processing steps.

7. Review and evaluate processes for monitoring and maintaining the state of security on the system.

If processes don't exist for security monitoring and maintenance, security holes could exist, and security incidents could occur without anyone's knowledge.

How

Interview the application administrator and review any relevant documentation to get an understanding of security practices for the application. This could include, for example, routine scans for and remediation of known vulnerabilities and/or alerts being sent and investigated when key activities are performed within the application (triggering off of audit logs kept within the system, as discussed in step 5). Some level of monitoring is important, but the monitoring level required should be consistent with the criticality of the system and the inherent risk of the environment.

If security monitoring is performed, assess the frequency of the monitoring and the quality with which it is performed. Review recent results, and determine whether exceptions were investigated and resolved.

If applicable for the application you are auditing, verify that policies and procedures are in place to identify when a security patch is available and to evaluate and apply applicable patches. Ensure that all approved patches are installed per your policy.

Account Management

8. Ensure that the application provides a mechanism that authenticates users based, at a minimum, on a unique identifier for each user and a confidential password.

Failure to authenticate users or just having a poor authentication scheme presents an open opportunity for curious users and malicious attackers to access your system.

How

Review the application with the developer or administrator and verify that appropriate authentication measures exist commensurate with the type of data on the application. For example, two-factor authentication might be required in some cases to authenticate users to sensitive applications or for end users accessing your applications from the Internet.

9. Review and evaluate the application's authorization mechanism to ensure that users are not allowed to access any sensitive transactions or data without first being authorized by the system's security mechanism.

The system's security mechanism should allow for each system user to be given a specific level of access to the application's data and transactions. Without the ability to provide granular access based on user need, users will likely be granted unnecessary levels of access.

How

Employees should be given only the amount of access to the system necessary to perform their jobs. Review the application with the developer or administrator, and verify this functionality in the application. In other words, it should be possible to specify which transactions and data sets or files a system user will be able to access. In general, it also should be possible to specify what level of access (such as display, update, and delete) the user will receive to application resources.

10. Ensure that the system's security/authorization mechanism has an administrator function with appropriate controls and functionality.

The administrator user function should exist to help administer users, data, and processes. This account or functionality should be tightly controlled in the application to prevent compromise and disruption of services to other users.

How

Evaluate the use of the administrator function with the developer or application administrator. The user of this function should have the ability to add, delete, or modify user access to the application system and its resources. The security mechanism should also provide the ability to control who has access to this administrator function. Obtain a list of all employees who have been granted the administrator access level and review each for appropriateness. Also ensure that the system's security mechanism provides the system's security administrator with the ability to view who has access to the system and what level of access they have.

11. Determine whether the security mechanism enables any applicable approval processes.

The application's security mechanism should support granular controls over who can perform what approval processes and then lock data that has been formally approved from modification by a lower authority. Otherwise, a lower-authority or malicious user could modify or corrupt data in the system.

How

Verify with the developer or application administrator that appropriate controls are in place. For example, if someone needs to approve journal entries before they can be passed on to the general ledger, the system's security mechanism should provide a means for defining who is authorized to perform this approval. Any data that has been through the approval process should be locked in order to prevent any further modifications.

Interviews with system users are a good way to help the auditor determine the need for this sort of ability. It is critical that the auditor understand not only the technical aspects of the application being reviewed but also the business purpose.

12. Review and evaluate processes for granting access to users. Ensure that access is granted only when there is a legitimate business need.

Users should have access granted and governed by the application administrator(s) to prevent unauthorized access to areas outside the user's intended scope. The application should have controls in place, and the administrator(s) should have processes in place to prevent users from having more access than is required for their roles. This step embodies the concept of least-privileged access.

How

Review processes for requesting and approving access to the application. Ensure that these processes are documented and that they require approval from a knowledgeable administrator before application access is granted to a user. Select a sample of users, and ensure that user access was approved appropriately. Verify that the authorization mechanism is working appropriately.

13. Review processes for removing user access when it is no longer needed. Ensure that a mechanism or process is in place that suspends user access on termination from the company or on a change of jobs within the company.

Poor deprovisioning processes may leave a user with inappropriate access to your application long after the access or authority should have been removed.

How

Verify that appropriate deprovisioning processes are in place with the developer and application administrator. Review the administrator(s)' processes for periodically reviewing user access lists and validating that the access is still appropriate. Be sure to look at both the application and the procedures around the application to ensure that they are being followed and are capable of being followed as written. Automated suspension of accounts in the event of termination or job change is preferable to processes that require manual intervention.

For applications that have been in "production" for some time, select a sample of system users and validate that their access is still appropriate. Alternatively, if possible, select a sample of system users who have changed jobs or left the company and ensure that their access has been removed.

14. Verify that the application has appropriate password controls. Also, determine whether default application account passwords have been changed.

The appropriateness of the password controls depends on the sensitivity of the data used within the application. Overly weak passwords make the application easier to compromise, and overly strong passwords could place unnecessary overhead on usage of the system.

Many applications, particularly those that are purchased, have default accounts with well-known default passwords. Many of these default accounts are used for system administration and therefore have elevated privileges. If those default passwords are not changed, it is easy for an unauthorized user to access the application.

How

Verify appropriate password controls with the help of the developer or the application administrator and by reviewing your company policy. For example, a three-digit PIN probably is inappropriate for an application that stores credit card data, and a 20-character password probably is overly paranoid for someone trying to access his or her voicemail. If your company policy requires periodic password changes (such as every 90 days), ensure that the security mechanism requires users to change their passwords in alignment with that policy. When appropriate, the security mechanism also should enforce password composition standards such as the length and required characters. Additionally, the security mechanism should suspend user accounts after a certain number of consecutive unsuccessful log-in attempts. This is typically as low as 3 and can be as high as 25 depending on the application, other forms of authentication required, and the sensitivity of the data.

Determine whether default accounts and passwords exist with the help of the developer or application administrator and by review of system documentation and Internet research. If they do exist, one of the easiest ways to determine whether they have been changed is to attempt to log on using the default accounts and passwords, but you're likely better advised to ask the application administrator to attempt to do so, as it might be against company policy to attempt to log in using someone else's account.

15. Ensure that users are automatically logged off from the application after a certain period of inactivity.

Without timeout controls, an unauthorized user could obtain access to the application by accessing a logged-in workstation where the legitimate user didn't log off and the application is still active.

How

Review the application with the developer or administrator to evaluate the existence of this ability.

Permissions Management

16. Evaluate the use of encryption techniques to protect application data.

The need for encryption is determined most often by either policy, regulation, the sensitivity of the network, or the sensitivity of the data in the application. Where possible, encryption techniques should be used for passwords and other confidential data that is sent across the network. This prevents other people on the network from "sniffing" and capturing this information. For sensitive data, such as passwords, encryption should also be considered when the data is at rest (in storage). This is particularly important for data that will be stored outside of your company's premises.

How

Review the application with the developer or administrator to evaluate the existence of encryption where appropriate.

17. Evaluate application developer access to alter production data.

In general, system developers should not be given access to alter production data in order to establish appropriate segregation of duties. Data entry and alteration should generally be performed by business users.

How

Discuss this with the developer or administrator and evaluate the separation of duties between developers and business users.

Software Change Controls

Software change management (SCM), used by a trained software development team, generally improves the quality of written code, reduces problems, and makes maintenance easier.

18. Ensure that the application software cannot be changed without going through a standard checkout/staging/testing/approval process after it is placed into production.

It should not be possible for developers to update production code directly. Your production code is your application, and it should be strictly controlled. Segregation of duties must be in place to ensure that all changes to the code are thoroughly reviewed and tested. Without these checks and balances, untested or unintended changes could be made to your production application, severely damaging the system's integrity and availability. Should a failure in the application occur without enforced software change controls, it might be difficult or impossible to track down the cause of the problem.

How

Evaluate this capability with the developers and application administrator. Determine the location of the production code and who has access to update that code. Preferably, the code will be controlled by some sort of librarian mechanism that provides granular control over how access to the code is managed.

Proper software change controls will require that the code first be checked out into a development environment, then checked into a testing or staging environment, and only then checked back into the production environment. Determine whether this is the case.

In addition, ensure that the software change mechanism requires sign-off before code will be moved into production. The system should require that this sign-off be performed by someone other than the person who developed or modified the code. In addition, the software change mechanism should allow for specific people to be authorized to perform sign-off on the system's programs. Review the people with this authorization and ensure that the privilege is kept to a minimum.

Evaluate controls in place to prevent code from being modified after it has been signed off on but before it has been moved to production. Otherwise, developers will be able to bypass approval processes.

Ensure that these controls are documented in a policy or procedure and communicated to developers. Consider selecting a sample of recent software changes and validating that all controls are implemented and working as designed.

19. Evaluate controls regarding code checkout and versioning.

Strong software controls regarding code checkout and versioning provide accountability, protect the integrity of the code, and have been shown to improve maintenance and reliability.

How

Verify with the developers that the software change mechanism requires developers to check out code that they want to modify. If another developer wants to modify the same code while it is still checked out, he or she should be prevented from doing so. Alternatively, the second developer could be warned of the conflict but allowed to perform the checkout. In such a case, a notification of the duplicate checkout should be sent automatically to the original developer.

Ensure that the software change mechanism "versions" software so that past versions of the code can be retrieved if necessary. This allows an easy mechanism for backing out changes, should issues be encountered.

20. Evaluate controls regarding the testing of application code before it is placed into a production environment.

Improperly tested code may have serious performance or vulnerability issues

when placed into production with live data.

How

Determine whether the software change process requires evidence of testing (including security testing), code walk-throughs, and adherence to software-development guidelines (including guidelines for secure coding). These should occur before the approver signs off on the code. Testing of any software development or modifications should take place in a test environment that mirrors the production environment, using test data. Ensure that these requirements are in place and documented. Pull a sample of recent software changes and look for evidence that these processes have been followed.

21. Evaluate controls regarding batch scheduling.

Many applications execute programs (often called "jobs") in batch (offline) mode. For example, an accounts receivable application may have jobs scheduled to run every night, and the application may acquire a feed of invoices and automatically apply payments to them. These functions are often performed by a series of jobs scheduled to run in sequence. If proper controls are not in place over the scheduling and monitoring of these jobs, it could result in inaccurate or failed processing.

How

Work with the application developers and administrators to understand what sort of batch processing is occurring and review applicable controls. Following are examples of common controls:

- Ensure that the batch scheduling tool can establish predecessor/successor relationships and that this ability is used where needed. Predecessor/successor relationships allow you to establish a sequence of jobs, where one job cannot kick off until another predetermined job successfully completes. This allows proper sequencing of processing.
- Determine whether the tool allows for jobs to be monitored for successful completion and has an alert mechanism in the event of unsuccessful completion. This alert mechanism should be used to alert some sort of central monitoring group, who in turn should have a contact and escalation list.
- The tool should provide the ability to control who can sign off on and implement changes to job scheduling and to job definitions (such as where the job is located, the name of the job, the user ID that runs the job, how often the job is scheduled, and so on). Review and evaluate who has the ability to sign off on and implement changes to job scheduling and job definitions. This ability should be limited.
- For changes to job scheduling and to job definitions, the tool should track who made the change, who signed off on the change, when the change was made, what was changed, and why the change was made. The tool should also allow you to retrieve previous versions of the schedule and of job definitions in the event of a problem with any changes. Determine whether this is the case.
- Ensure that the tool allows you to perform exception date processing. In other words, it should be able to accommodate changes in the schedule due to holidays, leap years, and so on.
- Ensure that recovery procedures have been developed that will allow for jobs that have ended abnormally to be restarted and reprocessed.

Backup and Recovery

22. Determine whether a business impact analysis (BIA) has been performed on the application to establish backup and recovery needs.

A BIA is performed to obtain input from the application's business users regarding the impact to the business in the event of an extended outage of the application (such as in the event of a disaster). This drives the engineering of the application's disaster recovery mechanisms.

How

Through interviews with the application support personnel and end users, determine what sort of BIA, if any, has been performed and review associated documentation. At a minimum, look for documented requirements regarding the application's RTO (recovery time objective, which dictates how quickly the system needs to be back up after a disaster) and RPO (recovery point objective, which dictates how much data the business can afford to lose in the event of a disaster).

23. Ensure that appropriate backup controls are in place.

Failure to back up critical application data may severely disrupt business operations in the event of a disaster (or possibly even a more common system outage), resulting in total loss of the application and its data with no ability to recover it.

How

Determine whether critical data and software are backed up periodically (for example, weekly full backups with daily incremental backups for the data) and stored offsite in a secured location (with appropriate encryption protections applied to the offsite data). If cost-beneficial and appropriate, duplicate transaction records should be created and stored to allow recovery of data files to the point of the last processed transaction. Ensure that the backup schedule is in alignment with the RPO and RTO established by the application's users.

Also ensure that the application code is backed up and stored offsite in a secured location, along with any tools necessary for compiling and using the code.

24. Ensure that appropriate recovery controls are in place.

Recovery procedures and testing are necessary to ensure that the recovery process is understood and that it functions operationally as intended.

How

Discuss this with the application administrator and appropriate personnel to ensure that detailed recovery procedures are documented that define what tasks are to be performed, who is to perform those tasks, and the sequence in which they are to be performed. Testing of the recovery from backup using the documented recovery procedures should be performed periodically. Ensure that the recovery processes are in alignment with the RTO established by the application's users.

NOTE To minimize redundancy, only the basics of auditing disaster recovery are included in this chapter. See Chapter 5 for additional details and ideas for auditing your application's disaster recovery capabilities.

Data Retention and Classification and User Involvement

25. Evaluate controls regarding the application's data retention.

Data should be archived and retained in accordance with business, tax, and legal requirements. Failure to do so could result in penalties and operational issues caused by the inability to obtain needed data.

How

These requirements will vary based on the type of data and should be acquired from the appropriate departments within your company. Ensure that inputs have been obtained from the appropriate personnel (such as the business owner of the data in the application and the legal and tax departments) to determine retention requirements. Evaluate the appropriateness of the retention controls with the developers and application administrator. Consider interviewing the business owner, legal department, and/or tax department to validate the retention requirements.

26. Evaluate the controls regarding data classification within the application.

All application data should be assigned a business owner, and this owner should classify the data (for example, public, internal use only, or confidential). This provides assurance that the data is being protected in alignment with its sensitivity.

How

Identify the business application owner (and/or the business owner of the data contained within the application) and ask for evidence that the data has been classified according to your company's data classification system. This classification should appear on any reports or transactions that display system data. Also, determine whether the application's access control mechanisms are appropriate based on the classification.

27. Evaluate overall user involvement and support for the application.

Without appropriate user involvement and support, the application may not adequately provide for user needs or appropriately support the business.

How

Interview the application's users and support personnel to determine what user involvement and support mechanisms have been put in place. Following are examples:

- Review and evaluate the existence of a formal steering team for the system. Generally, a steering team or some other form of user committee should exist to approve and prioritize system development and modifications.
- Ensure that changes to the functionality of the system are not made without user testing and approval.
- A mechanism should be in place that allows system users and developers to report and track system problems and to request system changes.
- For significant applications, some form of helpdesk functionality should exist to provide real-time help for user questions and problems.
- Ensure that system documentation and training exist that provides system users with adequate information to use the application effectively in performing their jobs.

Operating System, Database, and Other Infrastructure Controls

Detailed guidelines for controlling the operating system, database, and other related infrastructure components are beyond the scope of this chapter. However, security of the infrastructure on which the application resides is a critical part of application security. The applicable audit programs from this book's other chapters should be executed in addition to the application-specific steps provided in this chapter.

Master Checklists

The following table summarizes the steps discussed in this chapter for auditing applications.

Auditing Applications

Checklist for Auditing Applications
❑ 1. Review and evaluate controls built into system transactions over the input of data.
❑ 2. Determine the need for error/exception reports related to data integrity and evaluate whether this need has been filled.
❑ 3. Review and evaluate the controls in place over data feeds to and from interfacing systems.
❑ 4. If the same data is kept in multiple databases and/or systems, ensure that periodic sync processes are executed to detect any inconsistencies in the data.
❑ 5. Review and evaluate the audit trails present in the system and the controls over those audit trails.
❑ 6. Ensure that the system provides a means of tracing a transaction or piece of data from the beginning to the end of the process enabled by the system.
❑ 7. Review and evaluate processes for monitoring and maintaining the state of security on the system.
❑ 8. Ensure that the application provides a mechanism that authenticates users based, at a minimum, on a unique identifier for each user and a confidential password.
❑ 9. Review and evaluate the application's authorization mechanism to ensure that users are not allowed to access any sensitive transactions or data without first being authorized by the system's security mechanism.
❑ 10. Ensure that the system's security/authorization mechanism has an administrator function with appropriate controls and functionality.
❑ 11. Determine whether the security mechanism enables any applicable approval processes.
❑ 12. Review and evaluate processes for granting access to users. Ensure that access is granted only when there is a legitimate business need.
❑ 13. Review processes for removing user access when it is no longer needed. Ensure that a mechanism or process is in place that suspends user access on termination from the company or on a change of jobs within the company.
❑ 14. Verify that the application has appropriate password controls. Also, determine whether default application account passwords have been changed.
❑ 15. Ensure that users are automatically logged off from the application after a certain period of inactivity.
❑ 16. Evaluate the use of encryption techniques to protect application data.
❑ 17. Evaluate application developer access to alter production data.
❑ 18. Ensure that the application software cannot be changed without going through a standard checkout/staging/testing/approval process after it is placed into production.
❑ 19. Evaluate controls regarding code checkout and versioning.
❑ 20. Evaluate controls regarding the testing of application code before it is placed into a production environment.
❑ 21. Evaluate controls regarding batch scheduling.
❑ 22. Determine whether a business impact analysis (BIA) has been performed on the application to establish backup and recovery needs.
❑ 23. Ensure that appropriate backup controls are in place.
❑ 24. Ensure that appropriate recovery controls are in place.
❑ 25. Evaluate controls regarding the application's data retention.
❑ 26. Evaluate the controls regarding data classification within the application.
❑ 27. Evaluate overall user involvement and support for the application.

CHAPTER 16

Auditing Cloud Computing and Outsourced Operations

In this chapter, we will discuss key controls to look for when you are auditing IT operations that have been outsourced to external companies, including the following:

- Definitions of cloud computing and other forms of IT outsourcing
- Third-party attestations and certifications, such as ISO 27001
- Vendor selection controls
- Items to include in vendor contracts
- Data security requirements
- Operational concerns
- Legal concerns and regulatory compliance

Background

The concept of outsourcing IT operations to external service providers is not a new one. Companies have been implementing this concept for years, from hosting their applications via an application service provider (ASP), to storing their computer equipment in a co-location data center (also called a *colo*), to hiring an external company to run their IT operations. The decision to outsource operations is usually based on a desire to reduce costs and to allow a company to focus on its core competencies. For example, if you own a company that makes hockey sticks and your core competency is designing and building those hockey sticks, you might not want to invest the time and money required to run a data center to support your IT operations. Instead, you can pay someone who runs data centers for a living to do that for you. They can probably do it better than you could and at a lower cost, and it allows you to focus on those hockey sticks.

Over the last couple of decades, the concept of *cloud computing* has brought outsourced IT to the mainstream by providing IT services over the Internet using shared infrastructure. Cloud computing has grown from its initial stages to an industry buzzword and now to a legitimate and powerful business and operations model. The term is commonly misused to describe almost any type of IT outsourcing; we'll clarify the definition and attributes a little later.

Although outsourced operations can provide benefits to a company in terms of cost and resource efficiency, they also introduce additional risks, as the company gives up direct control over its data and IT environment.

Cloud Computing and Outsourced Operations Auditing Essentials

The methods used for outsourcing IT operations can be defined, separated, and categorized in multiple ways. None of those methods will be perfect or all-encompassing, but for the purposes of this chapter, they are divided into two major categories:

- IT systems, software, and infrastructure outsourcing
- IT service outsourcing

IT Systems, Software, and Infrastructure Outsourcing

IT systems, software, and infrastructure outsourcing is the practice of hiring another company to provide some or all of your IT environment, such as data center, servers, operating systems, business applications, and so on. This service can be provided using either cloud computing or dedicated hosting.

Cloud Computing

The National Institute of Standards and Technology (NIST) SP 800-145 defines cloud computing as "a model for enabling…convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Basically, cloud computing provides IT services over the Internet in such a way that the end user doesn't have to worry about where the data is being stored, where the infrastructure is located, and so on. The user receives the service without worrying about any of the details of how it's provided. Typically, as a consumer of cloud computing, you are sharing the back-end infrastructure that provides the service with other *tenants,* including some of the provider's other customers; it is not dedicated to you and your company. This is analogous to the utilities you use at home. You don't know or necessarily care how you get your electricity, but you do care that it works. You let the electric company worry about what it takes to provide the service. You don't have your own dedicated infrastructure at the electric company; you share it with all of your neighbors. Also, just like with your electricity at home, you pay for only what you use with cloud computing.

Some of the largest technology companies in the United States have developed cloud computing services for use by enterprises, small businesses, startups, and individual consumers. As of this writing, the largest services by market share in the United States include Amazon Web Services, Microsoft Azure, and Google Cloud. Each of these platforms offers a wide variety of services, including

- Virtual client and server systems, allowing you to run Windows, Linux, or other operating systems in the provider's cloud environment
- Storage platforms, such as Amazon S3, supporting general-purpose data storage

- Serverless computing, allowing your code to run on the provider's infrastructure, freeing you from the overhead of supporting an underlying operating system
- Databases, machine learning algorithms, and more

Businesses are increasingly using cloud services to enable speed to market while reducing startup costs. A new business can be started completely "in the cloud," foregoing the acquisition costs of computing equipment, IT support staff, data center space, and other expenses. Companies might use cloud computing for everything from experimentation, to running key business applications, to handling nearly all of their IT.

On a personal level, you've likely experienced cloud computing at home. If you have a personal e-mail address through a service such as Yahoo! or Gmail, you are receiving your e-mail in the cloud. You don't know and don't care where your data is stored and what sort of infrastructure is being used to provide the service to you. All you care about is that you can send and receive e-mail and manage your contacts. Also, you do not have a dedicated e-mail server on the back end; many other e-mail accounts are on the same server as yours. As to how many there are and who they are, you don't know and don't care. All you know and care about is that your e-mail is available and secure. (As we'll discuss later, organizations actually *do* care about some of these things when dealing with certain kinds of data, but you get the idea.)

As a buzzword, cloud computing as a term has often been overused, and is sometimes used improperly. For the purposes of this book, we'll use the NIST SP 800-145 definition.

Characteristics of Cloud Computing

According to NIST, for something to qualify as cloud computing, it must exhibit five characteristics.

On-Demand Self-Service This means that you can provision computing capabilities, such as storage, as needed automatically without requiring human interaction with each service's provider. It also implies that the implementation details are hidden from (and irrelevant to) the consumer. For example, the customer need not worry about what storage technology is used, but simply needs to define their business requirements and let the service provider determine how those requirements will be met.

Broad Network Access This means that capabilities should be accessible from anywhere and from any device (such as laptops and mobile devices) as long as Internet connectivity is available.

Resource Pooling This means that the provider's computing resources are pooled to serve multiple consumers using a *multitenant* model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand, with separation of data and users accomplished through logical rather than physical means. It provides a sense of location independence, in that the customer generally has no control over or knowledge of the exact location of the provided resources. Examples of resources in this context include storage, processing, memory, network bandwidth, and virtual machines.

Rapid Elasticity This means that capabilities can be rapidly and elastically provisioned (often automatically) to scale out quickly and rapidly released to scale in quickly. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

Measured Service This means that cloud systems automatically control and optimize resource usage by leveraging metering capabilities appropriate to the type of service (such as storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the service. This also implies transparency in cost, allowing the consumer to know that he is paying for only what he is using.

If a service your company is procuring does not meet those five criteria, it is likely not truly using cloud computing, but is instead using some form of dedicated hosting (discussed later in this chapter).

As we've discussed, cloud computing appeals to companies because it allows them to avoid the investment in physical infrastructure (and the operations for managing that infrastructure) and instead effectively "rent" infrastructure (hardware and software) from another company, paying for only the resources they use.

Cloud Computing Models

The next important concept to understand is the three primary models of cloud computing. The classifications of these three models have been relatively broadly accepted, but once again we'll lean on the NIST SP 800-145 definitions.

Software as a Service (SaaS) In this model, you will access the cloud provider's applications, which are running on a cloud infrastructure. The applications are accessible from client devices through a client interface such as a web browser (for example, web-based e-mail) or mobile application, or a program interface. As the consumer, you don't manage or control the data center, network, servers, operating systems, middleware, database management system (DBMS), or even individual application capabilities (with the possible exception of limited user-specific application configuration settings), but you may have some control over your data. Common examples of this form of cloud computing include salesforce.com, Gmail, Google's G-Suite, and Microsoft's Office 365 suite.

In the SaaS model, the customer is highly dependent on the cloud provider to manage most aspects of operations and security. An audit of this type of service should focus on company processes related to reviewing the security policies and processes of the provider, as well as general software controls, such as user access processes and software configuration change management.

Figure 16-1 shows a representation as to what layers of the infrastructure are dedicated to your company and what layers are shared with other customers in the SaaS model.

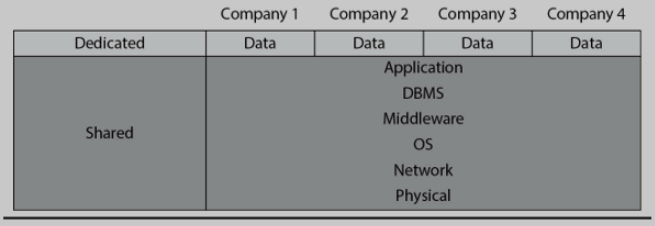


Figure 16-1 SaaS model

Platform as a Service (PaaS) In this model, you will deploy applications you created or acquired onto the provider's cloud infrastructure, using programming languages and tools supported by the cloud provider. As the consumer, you don't manage or control the data center, network, servers, operating systems, middleware, or DBMS, but you do have control over your data, the deployed applications, and possibly some configuration options.

In the PaaS model, the customer is dependent on the provider for operating system management, network and system security, and performance considerations. The customer is responsible for functionality and security of the application code itself.

This model, which include paradigms like Amazon Serverless Platform, introduces the concept of "shared responsibility," which is promoted by each of the major cloud providers. The notion of shared responsibility helps businesses understand where the cloud provider's efforts will end and where the business responsibilities begin. As an auditor, you should ensure that the organization understands the aspects of shared responsibility where PaaS systems are in use.

Figure 16-2 shows a representation as to what layers of the infrastructure are dedicated to your company and what layers are shared with other customers in the PaaS model.

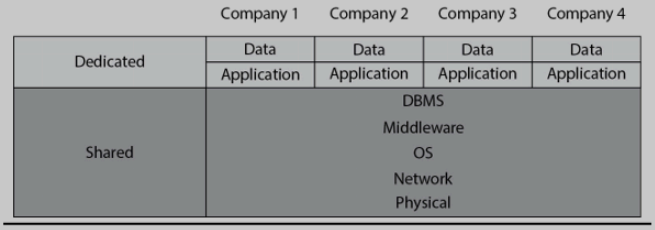


Figure 16-2 PaaS model

Infrastructure as a Service (IaaS) In this model, processing, storage, networks, and other fundamental computing resources are rented from the cloud provider. This allows you to deploy and run arbitrary software, which can include operating systems and applications. As the consumer, you don't manage or control the data center or network, but you do have control over your data and the operating systems, middleware, DBMS, and deployed applications.

In the IaaS model, the dividing line in the shared responsibility model moves closer to the customer organization. Operating system management, patching, network configuration, and more may be within the scope of the organization, while the provider is merely supporting network connectivity, physical hardware, and logical access controls necessary to enable the business to access the environment. An audit of this type of engagement may look very much like a data center or system audit, with the primary difference being that the data center is in a remote location.

Figure 16-3 shows a representation as to what layers of the infrastructure are dedicated to your company and what layers are shared with other customers in the IaaS model.

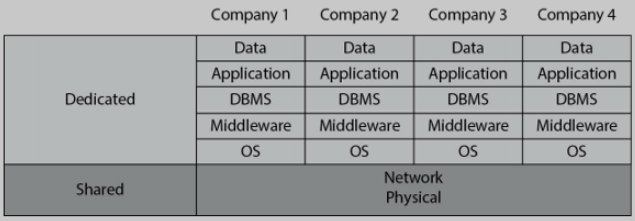


Figure 16-3 IaaS model

Dedicated Hosting

Dedicated hosting is conceptually similar to cloud computing, in that you're hiring someone else to provide (and probably manage) your infrastructure. The key difference is that with dedicated hosting, your company will have dedicated infrastructure, potentially sharing no more than the physical space with the vendor's other customers. An example of this would be a co-location (colo) data center, where you place your infrastructure (such as servers) in another company's data center, saving you the cost of building out and operating your own data center. Another example of this would be an ASP that hosts a business application for you, differentiated from SaaS only by the fact that you're on a dedicated server(s) not shared with the vendor's other customers. In contrast, with cloud computing, your data will be segregated logically but you may be sharing the rest of the infrastructure (such as network, servers, middleware, and so on) with the vendor's other customers. Figure 16-4 shows a representation as to what layers of the infrastructure are dedicated to your company and what layers are shared with other customers in the dedicated hosting model.

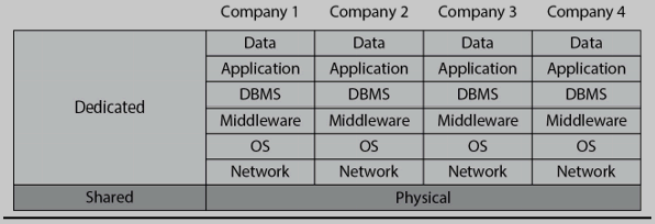


Figure 16-4 Dedicated hosting model

Although the concepts of what you need to protect may be the same between dedicated hosting and cloud computing, implementation will be vastly different. With dedicated hosting, you will look at how your network is isolated from other customers' (such as via firewalls). With cloud computing, you will look at how your data is segregated, since you're sharing the infrastructure. With dedicated hosting, encryption within your isolated network area may not be important. With cloud computing, you may want to see your data encrypted end-to-end, since it is commingled on the same infrastructure as other customers' data.

Because you're operating on dedicated infrastructure, dedicated hosting may not have the characteristics of cloud computing regarding on-demand self-service (the ability to provision additional capacity or other capabilities may not be automatic), broad network access (access may not be available via general Internet connections), resource pooling (you're on your own dedicated infrastructure), rapid elasticity (the ability to provision additional capacity or other capabilities may not be rapid, as procurement and setup time may need to be encompassed), or measured service (resource usage may not be automatically controlled and optimized).

There is often a fine line between whether you're using cloud computing or dedicated hosting. If you're not sure whether something is cloud or hosting, run a scenario by your provider. For example, tell them that you've just acquired another company and ask what it will take to scale the application to handle another 30,000 employees. If they say that they can handle it basically immediately, it's probably a cloud computing model. But if they say they need some time to expand your environment to accommodate the additional needs, it's probably dedicated hosting. This isn't a perfect test, as it will depend on your service provider and the amount of resources they have "on the bench" at the time, but it will give you a good indication.

Figure 16-5 shows a comparison of dedicated hosting and the three cloud computing models.

	Hosting	IaaS	PaaS	SaaS
Data	Dedicated	Dedicated	Dedicated	Dedicated
Application	Dedicated	Dedicated	Dedicated	Shared
DBMS	Dedicated	Dedicated	Shared	Shared
Middleware	Dedicated	Dedicated	Shared	Shared
OS	Dedicated	Dedicated	Shared	Shared
Network / Servers	Dedicated	Shared	Shared	Shared
Physical-Data Center	Shared	Shared	Shared	Shared

Figure 16-5 IT systems and infrastructure outsourcing model comparisons

NOTE Be aware that the definitions and distinctions among the various types of cloud computing and hosting are not always clear. Overlap can occur between these models and customizations (based on specific data protection requirements, cost constraints, and so on), leading to hybrid models. Also, people do not always use the terminology consistently or accurately. You will often find people who, for example, say they are using SaaS when they actually have dedicated hosting of their application (or vice versa). The auditor needs to be familiar with the concepts and standard models but should also realize that real-world scenarios will not always be as neat and tidy as what is reflected here. Not everyone will agree on the same terminology and definitions, so don't get too caught up in semantics.

IT Service Outsourcing

IT service outsourcing is the practice of hiring another company to perform some or all of your IT operations functions (that is, hiring the company to provide the people and processes necessary to perform the function). Commonly outsourced operations include helpdesk operations and PC support. This can obviously go hand-in-hand with the outsourcing of IT systems and infrastructure. For example, if you have placed your IT equipment in another company's data center, you are also likely to hire that company to perform data center operation activities (such as tape operations, hardware support, and so on). Similarly, if you deploy cloud computing, it is a given that the cloud provider will perform the operations over the cloud infrastructure.

Two types of IT service outsourcing are available, onsite and offsite, though there are obviously hybrids of these models, where portions of the function are performed onsite and portions are performed offsite.

Onsite

This model is used when a company outsources an operation but wants or needs for that function to be performed on company property. The external company is responsible for providing and training the people and establishing and monitoring the operational processes necessary for performing the function, managing all day-to-day aspects of the operation. However, the employees performing the function physically sit on the company's premises, using the company's network and IT environment.

Offsite

This model is used when a company outsources an operation without any onsite activity. Not only is the external company responsible for providing the personnel and processes necessary for performing the function, but they are also responsible for providing the facilities and infrastructure necessary for performing the function (often with connectivity back to the hiring company).

Other Considerations for IT Service Outsourcing

Additional topics related to IT service outsourcing are supplemental labor and remote operations.

Supplemental Labor

Many companies hire supplemental (contingent) labor to assist in their day-to-day operations. This is often done to assist with short-term needs or to perform jobs that require workers with skills that are easy to find and replace. This sort of activity should not be confused with truly outsourced operations. Supplemental labor workers perform activities under the day-to-day guidance and direction of your company's staff and therefore are subject to the controls and security already established for the functions your employees are performing. This is vastly different from a function where day-to-day operations have truly been outsourced.

Remote Operations

Many companies have expanded or relocated IT functions to various locations in the world. In some cases, this strategy provides an opportunity for 24-hour

(follow-the-sun) support; in others, lower labor costs provide a financial benefit. Although sourcing operations from remote locations can provide significant benefits, it also presents unique internal control challenges and additional complexities into the environment, especially in the areas of coordination and communication.

IT Service Outsourcing Models

In summary, when it comes to staffing IT services, the following basic models are used:

- Internal employees only
- Internal employees plus supplemental labor
- Outsourced: onsite
- Outsourced: offsite
- Outsourced: onsite/offsite mix

For each of these provisioning models, the following deployment options are used:

- Local (in a location very close to the business)
- Remote (in another region or country)
- Local/remote mix

Third-Party Reports and Certifications

When auditing cloud providers, you need to understand the various certifications, attestations, and reports available from independent, third-party assessors. Some of these include

- Statements on Standards for Attestation Engagements No. 18, or SSAE 18
- International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 27000 series, including ISO/IEC 27001, ISO/IEC 27002, and ISO/IEC 27018
- Health Information Trust Alliance (HITRUST)
- Cloud Security Alliance Security Trust Assurance and Risk (STAR)
- Federal Risk and Authorization Management Program (FedRAMP)

Some of these are certifications that a provider can achieve; others take the form of attestations, which are effectively the opinions of the auditing agent. Depending on your industry, you may encounter various forms of reports, and your organization may have specific requirements. In the United States, SSAE 18 is commonly seen, but ISO/IEC 27001 is a more widely known, international standard. We'll discuss SSAE 18 and ISO/IEC 27001 in more detail, as these are common across a broad range of industries.

SSAE 18

SSAE 18 is an auditing standard developed by the American Institute of Certified Public Accountants (AICPA) to deal with service organizations. It essentially provides a standard by which service organizations (such as those that provide IT services) can demonstrate the effectiveness of their internal controls without having to allow each of their customers to come in and perform their own audit. Without this standard, service organizations would expend a prohibitive amount of resources responding to audit requests from each customer. With this standard, service organizations can hire a certified independent service auditor (such as Ernst & Young) to perform an SSAE audit and issue various reports. These reports can in turn be presented to customers requiring evidence of the effectiveness of the service organization's internal controls.

NOTE SSAE 18 supersedes the previous SSAE 16, which superseded the much older Statement on Auditing Standards (SAS) 70 report. SSAE 18 became the effective reporting model in May 2017.

SSAE 18 auditor reports come in three varieties; two of these are broken down into additional subtypes. These Service Organization Controls (SOC) reports include

- SOC 1, intended to evaluate the effect of provider controls on client financial reporting
- SOC 2, intended to provide assurance on the security, availability, and privacy of provider systems and client data
- SOC 3, a less detailed variant of SOC 2

SOC 1 and SOC 2 reports come in two types: Type 1 and Type 2. Both types include a description of and independent opinion on the design of the service organization's internal controls. However, a Type 1 report looks at a point in time for the systems and services in scope. A Type 2 report includes an evaluation of operating effectiveness of controls based on the results of testing by the service auditor over the period under review to provide assurance that the control objectives were achieved. As an auditor, you will want your service providers to provide a Type 2 report, as Type 1 reports do not provide evidence that the controls are operating effectively.

SSAE 18 reports have become particularly important since the implementation of Section 404 of the Sarbanes-Oxley Act (SOX) in 2002, as companies can use them as evidence of the effectiveness of internal controls over any aspects of financial processing and reporting that have been outsourced. Without them, any company providing financial services would be bombarded with SOX audits from all of their customers, as opposed to being able to hand each customer the same SSAE 18 report.

NOTE SSAE 18 is an attestation report; it is not a standard. Organizations are never "certified" or "compliant" with SSAE 18 or any of the SOC reports. An organization advertising itself as such may not understand the nature of SSAE 18.

Service providers often undergo an SSAE 18 review on an annual basis; however, this may not be suitable for customers subject to SOX requirements. Consult your organization's SOX compliance team for information on the timing and reporting needs for artifacts like SSAE reports.

ISO/IEC 27001

The ISO/IEC 27000 series, which includes the widely known ISO/IEC 27001 as well as the less well-known ISO/IEC 27018, comprise a number of standards related to the practice of information security. These standards are jointly published by ISO and IEC; as a result you will usually see both as part of the standard name, such as ISO/IEC 27001. The shorter "ISO 27001" is also commonly used but refers to the same standard.

ISO/IEC 27001 deals with information security management and describes an appropriate program to properly manage the information security function. An organization with sufficient levels of control and compliance with the more than 100 controls in the document can apply to be certified as "compliant" with ISO/IEC 27001. This does not mean that the organization will never have a security incident, but that the organization has implemented an information security management program with sufficient rigor to manage the organization's risks. Many cloud service providers undergo ISO/IEC 27001 certification to demonstrate to potential customers that they have a viable security program.

ISO/IEC 27002 gets into more detail about information security practices. The various controls in ISO/IEC 27001 are described at a high level, while the 27002 standard provides additional guidance on implementation for each control. In practice, an organization probably implements much of ISO/IEC 27002 in order to achieve compliance with 27001.

ISO/IEC 27018 was adopted in 2014 and describes protections related to personal data and privacy. If your organization is entrusting personally identifiable information (PII) or other sensitive personal data to a cloud provider, you should add ISO/IEC 27018 compliance to your list of audit considerations.

Test Steps for Auditing Cloud Computing and Outsourced Operations

Here are a few notes on the test steps in this chapter.

First and foremost, whatever audit steps you would want to perform if the service were being performed by your company (that is, if it were not outsourced) should be considered when you're auditing an outsourced function. The same risks exist and will need to be mitigated. For example, if a business application is hosted in the cloud via SaaS, you will need to review for the sorts of application controls that are documented in Chapter 15. Those risks don't go away just because the application has been outsourced, and they are all still relevant to an audit program. However, the way you audit for them may be vastly different if the function has been outsourced.

Second, you need to determine whether you will be auditing the vendor and evaluating its controls or whether you will be auditing your own company and asking how it ensures that the vendor is providing the necessary controls. Both approaches are valid, and it may depend on what sort of right to audit and influence you have with the vendor. However, in general, it is preferable that you ask the questions of the vendor directly as opposed to using a middleman. You're more likely to get thorough and accurate answers. It's also sometimes interesting to ask the same question of both the vendor and your own internal IT team and compare their answers. This can tell you how well your company understands and reviews the controls over the outsourced operations.

Finally, for each step in this section, we will note which types of outsourcing (such as cloud computing, dedicated hosting, service outsourcing) are most applicable to that step. These are not intended to be absolute, because the scope of each outsourcing engagement is unique, but instead are intended to be guidelines.

Initial Steps

1. Review the audit steps in the other chapters in this part of the book and determine which risks and audit steps are applicable to the audit being performed over outsourced operations. Perform those audit steps that are applicable.

The risks present for an insourced function are also present for an outsourced function. Remember that the components and functions of what you've outsourced are similar in many cases to what you would have internally. They are simply being handled by a different entity. Regardless of who is responsible for your data and applications, you still have controls that must be put in place. Although additional risks are present when a function is outsourced, you still must review for the basic controls that you would expect of an internally sourced function. For example, if you outsource a business application, you will still be interested in access controls, data input controls, and software change controls over those applications. Those controls are still critical to the confidentiality, integrity, and availability of that application. And if you outsource your data center, you will still be concerned as to how the people running that data center ensure physical security and continuity of operations.

This step is applicable to all forms of outsourcing. For large cloud service providers such as AWS, Microsoft Azure, Google Cloud, or similar, an understanding of the scope of your company's use of available services will help you identify the steps you will need to perform.

How

Although you could argue that you would perform all of the same steps for an outsourced function as you would for an insourced one (again, because the risks are all still present), in reality, you probably won't have the same level of access with an outsourced process that you would get for an internal process, so you need to pick your battles. For example, if you want to review operating system security, the vendor may not give you access to accounts on its operating systems so that you can review system configuration. Maybe it will, and it's certainly worth asking, but you will often be limited by contractual rights. Instead, you may focus on their processes for keeping their systems patched and for regularly monitoring the security of the systems themselves (that is, review their processes regarding ensuring system security rather than reviewing the configuration of specific servers). If possible, you might ask the vendor to run a set of read-only scripts that pull key system configuration information from their environment and send you the output. After developing your wish list of steps you would like to perform during the audit, you might go ahead and determine which ones are the most critical to you so that you'll know which ones to fight for should you encounter resistance from the vendor.

Significant variability will be the norm with regard to how you perform these steps—it all depends on the rights, influence, and relationship you have with your vendor. Some may allow you to come in and audit their processes and infrastruc-

ture just as if you were their own internal auditors. Others will hand you an SSAE 18 report and be done with you, informing you that they have fulfilled their obligation. You will have to negotiate each instance separately and enlist the aid of your procurement, legal, and operations groups to see how far you can push for transparency from your supplier. This is why it is critical to establish robust "right to audit" clauses in your contracts to deal with these situations up-front, while you still have leverage.

NOTE This is a critical step. For efficiency's sake, we are not duplicating the audit steps from other chapters here. However, if, for example, you are performing an audit of data center operations that have been outsourced to a co-location facility, it is critical that you perform not only the steps in this chapter but also the steps in Chapter 5. Likewise, if you are performing an audit of a business application that uses the SaaS model, you must perform not only the steps in this chapter but also the steps in Chapter 15 (at a minimum). In fact, just as when you're auditing an internal application, you might also want to perform steps from Chapters 7 to 10 on auditing the pertinent operating system, the database, and so on.

2. Request your service provider to produce independent assurance from reputable third parties regarding the effectiveness of their internal controls and compliance with applicable regulations. Review the documentation for issues that have been noted.

dors and provide them to you.

Once you receive whatever assessments are available, you must review them in a number of areas. First, obviously, you must review the results of the assessment to understand any issues noted and the vendor's remediation plans. You will want to track these items to ensure that they have been remediated satisfactorily (which, again, you may need to determine via a third-party assessment). It is also important to ensure that the assessment was performed by a qualified independent third party and to determine the time period covered by the assessment to be sure it is still relevant.

You will also need to review the scope of the assessment performed and determine how many of your control objectives were addressed by it. You will likely see some gaps between your company's control objectives and the control objectives addressed by the independent assessment. Once you identify these gaps, you can attempt to perform your own assessments of those items not covered by the third-party assessment. You will have to negotiate each instance separately and enlist the aid of your procurement, legal, and operations groups to see how far you can push for the ability to perform your own audit. This again emphasizes the importance of placing a "right to audit" clause in your contracts.

If you find that your vendor does not have appropriate third-party assessments, you will have to attempt to perform all pertinent audit steps yourself (which may be limited by your right to audit). If this is the case, you should push your vendor to undergo an SSAE 18 SOC 2, Type 2 assessment and/or other pertinent independent assessments, possibly making this a negotiating point at contract renewal time. You should expect to see this type of assessment for any form of IT systems and infrastructure outsourcing (such as cloud computing). It may not be reasonable to expect it for IT service outsourcing models where you are providing significant guidance on day-to-day activities (such as when you outsource a function but manage it onsite using your own systems).

Also, determine how closely these certifications match your own company's control objectives and identify gaps.

Although you are attempting to perform your own audit of your service provider's controls, experienced service providers will already be engaging third parties for regular assessments. These assessments can be used to reduce your need to audit the service provider's functions, thereby reducing the scope of your audit. In fact, many service providers, especially the larger ones, will insist that you use these assessments in lieu of performing your own audit.

This step is most applicable to cloud computing, dedicated hosting, and offsite service outsourcing.

How

Request this information from your vendor. The type of independent assurance you ask for will vary depending on your industry, but the most common assessments you should look for will be an SSAE 18 report or ISO/IEC 27001 certification. Make sure you request a SOC 2, Type 2 SSAE 18 assessment when working with organizations that indicate they can provide SSAE 18 reports. You will need to determine what assessment(s) should be expected based on your industry, the type of outsourcing being performed, and the type of auditing you're performing. For example, if you're performing an audit of an outsourced website, you should expect to see some form of web security certification. As part of this exercise, you will need to determine whether your vendor subcontracts any relevant functions to additional third parties (for example, if you're using a SaaS vendor and it uses another vendor's data center facilities to host its systems). If so, request that your vendor obtain applicable independent assessments from those subcontract ven-

Vendor Selection and Contracts

3. Review applicable contracts to ensure that they adequately identify all deliverables, requirements, and responsibilities pertinent to your company's engagement.

The contract is your only true fallback mechanism should you have issues with the vendor. If it's not spelled out in the contract, it becomes very difficult, if not impossible, to enforce requirements and/or seek restitution should there be issues.

This step is applicable to all forms of outsourcing.

How

The best time to perform this step is before the contract is finalized and signed, because that's when you can make changes and influence the contents of the contract relatively easily. However, if you are performing the audit after the contract has been signed, it is still relevant for two reasons: First, it will give you an idea as to what you're working with and what sort of leverage you will have during the audit. Second, it will allow you to provide input as to what changes need to be made in the contract when it's time to renegotiate.

Regardless of whether you're reviewing a signed contract or providing input before the fact, you should make sure the following areas are addressed in the contract:

- Specify how performance will be measured, including service level agreements (SLAs) that specify requirements for availability (such as expected uptime), performance (such as speed of transaction response after the enter key is pressed), response time (such as whether the vendor will respond to problems 24/7 or only during normal business hours), and issue resolution time (such as how quickly you should expect issues to be fixed).
- SLAs for security (that is, requirements for controls to protect the confidentiality, integrity, and availability of data) can include requiring specific control frameworks (such as COBIT) to be followed and requirements for third-party assessments. It should also include requirements for how data should be stored (such as encryption, including requirements for the algorithm and key length), who may be granted access to it, how business continuity and disaster recovery will be ensured, how investigations will be supported, what security training and background checks are required for personnel who will access your systems and data, how data retention and destruction should occur, and so on. Overall, you want to make sure your vendor takes some contractual responsibility for security.
- Other key metrics and performance indicators should be included, which can be used by your company to measure the quality of the service. For example, if you have outsourced your helpdesk function, you might want to set an expectation as to tickets closed per analyst and customer satisfaction rating.
- Outline requirements for compliance with applicable laws and regulations (such as PCI, HIPAA), including requirements for independent assessments certifying compliance.
- Provide provisions for penalties upon nonperformance or delayed performance of SLAs and conditions for terminating the agreement if performance goals are not met.
- Add a right-to-audit clause, specifying what your company is allowed to audit and when. You obviously will want to push for a broad right to audit, allowing you to audit whatever you want, whenever you want (including the ability to perform surprise audits). You can negotiate from there. The broader you make this clause, the more freedom you will have.
- Include provisions for your right to audit and review independent assessments (such as SSAE 18) for functions that your vendor subcontracts out to other vendors (for example, if your SaaS vendor is hosting its systems with another third party). If possible, dictate in the contract what functions (if any) your vendor is allowed to subcontract and/or obtain the right of approval for any subcontracting relationships.
- Gain assurance that you can retrieve your data when you need it and in the format you desire, particularly if the agreement is terminated.
- Add language prohibiting the vendor from using your data for its own purposes (that is, for any purposes not specified by you).
- Include nondisclosure clauses to prevent the vendor from disclosing your company's information.
- Include evidence that the contract was reviewed by your procurement and legal organizations, as well as applicable operations groups.
- Basically, include anything you expect from the service provider that needs to be specifically outlined in the contract. Consider the other steps in this chapter for ideas as well.

4. Review and evaluate the process used for selecting the outsourcing vendor.

If the process for selecting the vendor is inadequate, it can lead to the purchase of services that do not meet the requirements of the business and/or poor financial decisions.

This step is applicable to all forms of outsourcing.

How

Obviously, your goal should be to perform this step prior to vendor selection, when you can influence the decision. However, if your audit is being performed after the fact, there is still value in understanding the vendor selection process. It can identify gaps that must be addressed and provide information that can be used when it's time to renew the contract or enter into other contracts.

Review the vendor selection process for elements such as these:

- Ensure that multiple vendors are evaluated and involved in the bid process. This provides for competitive bidding and lower prices.
- Determine whether each vendor's financial stability was investigated as part of the evaluation process. Failure to do so may result in your company signing up with a vendor that goes out of business, causing significant disruption to your operations as you attempt to bring them back in-house or move them to another vendor.
- Determine whether each vendor's experience with providing support for companies of similar size to yours and/or in a similar industry was evaluated. This can include obtaining and interviewing references from companies that currently use the vendor's services. You generally want to select a vendor who has already demonstrated that they can perform the types of services you're looking for at a similar scale.
- Ensure that the vendor's technical support capabilities were considered and evaluated.
- Ensure each vendor was compared against predefined criteria, providing for objective evaluations.
- Determine whether there was appropriate involvement of procurement personnel to help negotiate the contract, of operations personnel to provide expert evaluations as to the vendor's ability to meet requirements, and of legal personnel to provide guidance on potential regulatory and other legal ramifications of the outsourcing arrangement.
- Ensure that a thorough cost analysis was performed. The total cost of performing the operation in-house should be developed, as well as the total cost for using each vendor. This analysis should include all relevant costs, including costs for one-time startup activities, hardware and related power and cooling, software and its maintenance, hardware maintenance, storage, support (labor), and so on. Too often, companies make decisions without considering all relevant costs. For example, some of the cost savings from cloud computing may be offset by increased monitoring to ensure that requirements are met. These costs need to be included in the analysis to ensure that the company is making an informed decision.

Account Management and Data Security

For all of the steps in this section (except step 8), your first option should be to determine whether an evaluation of the area is available via a third-party assessment (such as SSAE 18). If it is not, you'll need to work with your operations, procurement, and legal departments to determine your rights to audit the vendor in this area. Hopefully, those rights are spelled out in the contract. If they are not, your company will need to attempt to press for that right, possibly using the next contract renewal as negotiating leverage.

If the area is not covered by an assessment such as SSAE 18 and if you have the right to audit the vendor, you will need to interview the vendor and review their documentation regarding their technical controls and processes, testing those controls as you're able.

You will also want to see your company's requirements for these controls spelled out in your contract and look for evidence that those specific requirements are being met.

5. Determine how your data is segregated from the data of other customers.

If your company chooses a form of outsourcing in which your data is being stored on the vendors' systems at their site (such as in cloud computing and dedicated hosting), you no longer have full control over your data. Your data may be commingled with other customers' data (a likely scenario with cloud computing). This creates a number of risks. For example, if data is not properly segregated, another customer (including one of your competitors) on the same shared infrastructure may be able to access your data. Likewise, if one customer's system is breached, the confidentiality, integrity, and availability of other customers in the same environment may be at risk. For example, viruses might be transmitted from one customer to another or an attacker might be able to download data from all customers in the environment.

This step is most applicable to cloud computing and dedicated hosting.

How

Review the technical controls and processes for assuring segregation and protection of your systems and data. There's no single way to do this, and the implementation will differ depending on the technologies being used by your vendor, but the vendor needs to demonstrate how they have segregated and compartmentalized systems, storage, network, and so on. For example, in a dedicated hosting environment, you'll be looking for network devices (such as firewalls) to segregate the network hosting your systems from the networks hosting other customers. In a SaaS environment, you'll be looking for segregation of databases containing customer data. Ideally, you would like to perform your own tests to validate that their controls are working as designed. Again, the nature of these tests will depend on the technology and the implementation.

6. Review and evaluate the usage of encryption to protect company data stored at and transmitted to the vendor's site.

If your data is no longer fully under your control (that is, it is being stored at a third-party site and possibly being commingled with data from other customers), it is critical that the data be encrypted to protect against possible compromise. This reduces the risk of a breach affecting the confidentiality or integrity of your data. If you have unencrypted data in a shared environment (such as cloud computing), you can assume that it is no longer confidential.

This step is most applicable to cloud computing, dedicated hosting, and offsite service outsourcing.

How

Look for encryption of data both in transit (for example, via TLS for browser-enabled transactions) and at rest (that is, in storage), because both are outside of your control if your data is stored at a third-party site. Evaluate the strength of the encryption. Hopefully, you will have contractually dictated requirements for encryption (such as algorithm and key length) against which you can compare the system.

Determine how key management is performed and how your keys are separated from those of other customers in your environment. Ideally, this function should be performed either by your company or by a separate vendor from your standard outsourcing vendor, providing for segregation of duties. However, many cloud providers also offer robust key management services, which can reduce the potential for failures and relieve your organization of the overhead and complexity of key management. For example, AWS offers a hosted key management solution that provides for data encryption but places key management responsibilities mostly on Amazon.

7. Determine how vendor employees access your systems and how data is controlled and limited.

If your data is being stored or processed by employees outside of your company and you do not maintain ownership regarding who has access to that data, you're putting its confidentiality, integrity, and availability at risk.

This step is applicable to all forms of outsourcing.

How

Determine who has access to your data and systems and review for appropriateness. Determine how appropriate segregation of duties is maintained. Ensure that the concept of "minimum necessary access" is followed.

Review the approval process for determining who will have access to your systems and data. Ideally, the data owner at your company will be the gatekeeper for approval. Your company should maintain the right (hopefully spelled out in the contract) to deny access to your data from vendor personnel.

Review your vendor's processes for hiring and screening employees, ensuring that appropriate background checks are performed and rules regarding security and management of your environment are communicated to the employees. In addition, review the vendor's use of third-party labor for system administration and support. Many vendors support their cloud environments using only internal employees, while some may outsource these functions. These requirements should be dictated in the contract.

Ask for a listing of any third-party relationships that your vendor has and any interfaces those additional parties have to your systems. Each of these represents additional exposure of your data.

8. Review and evaluate processes for controlling nonemployee logical access to your internal network and internal systems.

If you're using service outsourcing and/or supplemental (contract) labor, you are likely allowing a third-party vendor's personnel to have a degree of logical access to your network and systems. Because these personnel are not employees of your company, they are less likely to have a personal investment in the company's success or an awareness of its policies and culture. If their access to company information assets is not governed and if expectations regarding their usage of that access are not communicated, it is more likely that company information assets will be unnecessarily exposed or misused.

This step is most applicable to onsite and offsite service outsourcing plus supplemental labor.

How

Ensure that policies require approval and sponsorship from an employee prior to a nonemployee obtaining logical access to company systems. If feasible, obtain a sample of nonemployee accounts and validate that they have appropriate approval and sponsorship.

Review and evaluate processes for communicating company policies (including IT security policies) to nonemployees prior to granting them system access. Look for evidence that this communication has occurred. For example, if all nonemployees are required to sign a statement that they have read and agree to the policies, pull a sample of nonemployees and obtain copies of these agreements.

Review and evaluate processes for removing logical access from nonemployees when they have ceased to work with your company or otherwise no longer need access. Consider obtaining a sample of current nonemployee accounts and validating that those nonemployees are still working with your company and still have a need for their current level of access.

Ensure that nondisclosure agreements (NDAs) are signed by nonemployees to legally protect your company from inappropriate use of company data. Pull a sample of nonemployee accounts, and obtain a copy of the NDA for them.

Ensure that consideration has been given to identifying data that should not be accessed by nonemployees and activities that should not be performed by nonemployees. For example, your company may decide that access to certain levels of financial data should never be granted to nonemployees. Or it may decide that nonemployees should never be granted system administration duties. The answer will depend on your company's industry and philosophies; however, an evaluation process should take place, and the results of that evaluation should be documented in company policy and enforced.

9. Ensure that data stored at vendor locations is being protected in accordance with your internal policies.

No matter where you store your data, it is still subject to your internal policies. Outsourcing storage to a third party does not absolve your company of the responsibility to comply with policies and ensure proper security of the data.

This step is most applicable to cloud computing and dedicated hosting.

How

Ensure that data stored at third-party sites has been classified in accordance with your company's data classification policy and is being protected in accordance with that policy. Data with certain levels of classification might be inappropriate to store outside the company (such as employee and customer personal information). Review your company's policies on data security and ensure that offsite data is being protected in accordance with those policies. Encrypting data that is stored with the vendor will greatly benefit you in this area.

10. Review and evaluate controls to prevent, detect, and respond to attacks.

Without appropriate intrusion detection and prevention techniques, your systems and data are at an increased risk of compromise. This risk is increased in an outsourced model, specifically when outsourcing systems and infrastructure, because of the shared infrastructure—an attack and compromise on one customer could result in compromise of your systems.

This step is most applicable to cloud computing and dedicated hosting. Also, consider whether this risk is applicable if you're using offsite service outsourcing, as the service provider may store your data on their systems and/or have connectivity to your internal systems.

How

This step might be divided into separate substeps. For infrastructure and systems located at third-party sites, determine the effectiveness of the technology deployed, coverage of deployed technology, and processes to monitor and maintain the technology, such as those listed next.

Intrusion Detection Look for the usage and monitoring of intrusion detection systems (IDSs) to detect potential attacks on your systems and integrity-checking tools to detect potential unauthorized changes to system baselines.

Intrusion Prevention Look for the usage and monitoring of intrusion prevention systems (IPSs) to proactively detect and cut off potential attacks on your systems.

Incident Response Look for clearly defined processes for responding to alerts and potential security incidents, including notification and escalation procedures.

Discovering and Remediating Vulnerabilities Look for the usage and monitoring of vulnerability scanning tools to detect and mitigate potential vulnerabilities that might allow an intruder to access and/or disrupt your systems.

Logging Look for the logging of significant activities (successes and failures) on your systems, for the monitoring of these logs, and for the storage of these logs in a secure location for an adequate period of time.

Patching Look for procedures to receive and apply the latest security patches so that known security holes are closed.

Protection from Viruses and Other Malware Look for the usage of antivirus software and the application of new signature files as they are released.

For providers like AWS or Azure, consider the shared responsibility model. Review with security personnel to ensure they understand where provider responsibility ends and your company responsibility begins and that commensurate controls exist to cover customer responsibilities.

11. Determine how identity management is performed for cloud-based and hosted systems.

Proper identity management practices are critical for controlling access to your systems and data. If an outsourced operation is not integrated with your existing identity systems or single-sign-on processes, employees may end up with many accounts, with inconsistent password and account management practices across various providers.

This step is most applicable to cloud computing, particularly SaaS, and dedicated hosting, particularly of purchased applications.

How

Although it's possible to review the identity management controls over each outsourced system (checking each for appropriate password controls, account management controls, and so on), you should prefer to have a federated identity management capability. This will allow your users to authenticate to your internal systems with a single company ID and password, which will also serve as your authentication for the vendor system. This offers the benefits of centralized identity management and allows you to avoid storing user credentials with your vendor.

If you implement this form of federated identity management, using, for example, technologies such as Security Assertion Markup Language (SAML), be sure that your internal credential data (such as passwords) are not exposed or accessible to the vendor. These requirements will preferably be dictated in your contract. If you are unable to implement federated identities, you will need to review the identity management controls over your outsourced systems to ensure that they meet the requirements of your policies. An alternative solution is to use an identity management service as a "middleman" between your company and your vendor, but, of course, that solution introduces another third party that you must audit into your environment.

12. Ensure that data retention and destruction practices for data stored offsite comply with internal policy.

If the life cycle of data is not defined, data might be retained longer than necessary (resulting in additional storage costs and possible legal liabilities) or may be destroyed prematurely (leading to potential operational, legal, or tax issues).

This step is most applicable to cloud computing, dedicated hosting, and offsite service outsourcing (if the supplier is storing your data).

How

Determine whether life cycle requirements have been defined for data stored with vendors. For a sample, review the documentation of the data's life cycle requirements, including retention, archive, and destruction requirements. Ideally, requirements will be identified for how long the data should be active (online, easily accessible, modifiable if appropriate, and backed up periodically), when and for how long it should be archived (possibly offline, not necessarily easy to access, no longer modifiable, and no longer backed up periodically), and when it should be destroyed. Ensure that these requirements appropriately reflect the nature of the data (for example, external public content on your website should be treated differently than customer data). The contract should dictate that the vendor manage data per your life cycle requirements. Review evidence that life cycle requirements have been implemented, concentrating especially on evidence that your vendor has destroyed data per your requirements. Note that data destruction can often be very difficult to prove in the cloud, increasing the importance of using strong encryption for your data, as described earlier.

For services using a shared responsibility model, such as AWS or Google Cloud, determine which data falls under your company's responsibility and ensure data management controls are in place.

13. Review and evaluate the vendor's physical security.

Physical security affects logical security, because physical access can override some logical access controls. You can have excellent logical security, but if someone can walk in off the street and walk off with the computer (or perhaps just the disk drive or tape cartridges) containing your systems and data, you will, at a minimum, experience a disruption of service, and if the data is not adequately encrypted, you may also be looking at a security breach.

This step is most applicable to cloud computing, dedicated hosting, and offsite service outsourcing.

How

Review the vendor's physical security for controls such as these:

- Badge readers and/or biometric scanners
- Security cameras
- Security guards
- Fences
- Lighting
- Locks and sensors
- Processes for determining who will be granted physical access

See Chapter 5 for additional information on auditing physical security controls.

Operations and Governance

14. Review and evaluate your company's processes for monitoring the quality of outsourced operations. Determine how compliance with SLAs and other contractual requirements is monitored.

Although you have hopefully dictated expectations in your contract, unless you monitor for compliance with those expectations, you will have no way of knowing whether they're being met. If those expectations are not met, the availability, efficiency, and effectiveness of your operations and the security of your systems and data can be affected.

This step is applicable to all forms of outsourcing.

How

Review the contract to understand requirements. Interview your company's internal management to determine their processes for monitoring that each of those requirements is being met. Obtain and review metrics, slides from operations reviews, and other materials, and compare the results to the requirements stipulated in the contract. Where deviances have occurred, review for corrective action plans and evidence that those plans have been implemented and were effective.

If requirements have not been dictated in the contract, determine how the quality of services is monitored and how the vendor is held accountable. The inclusion of SLAs should be a requirement when the contract is renewed.

Ensure you cover the following basic topics in performing this step:

- Availability (such as expected uptime)
- Performance (such as speed of transaction response after the ENTER key is pressed)
- Response time (such as whether the vendor will respond to problems 24/7 or only during normal business hours)
- Issue resolution time (such as how quickly you should expect issues to be fixed)
- Security and compliance requirements
- Other key metrics and performance indicators that can be used by your company to measure the quality of the service

Some organizations may leverage existing departments to conduct performance monitoring for services like AWS or Azure. In other cases, dedicated cloud operations teams will handle these activities. Determine how cloud operational monitoring is handled if your company leverages this kind of service, and ensure the various monitoring areas are covered in accordance with the needs of the applications or systems deployed.

15. Ensure that adequate disaster recovery processes are in place to provide for business continuity in the event of a disaster at your service provider.

Just as with internally hosted systems, you must prepare for recovery from a disaster when outsourcing operations. Failure to do so will likely result in extended outages and business disruptions if a disaster occurs with your vendor.

This step is most applicable to cloud computing, dedicated hosting, and offsite service outsourcing.

How

You should expect that your vendor will follow sound disaster recovery disciplines, such as those you would look for when auditing your internal operations. This includes steps outlined elsewhere in this book, such as reviewing for offsite backups, up-to-date documented recovery procedures, periodic testing, hardware redundancy, and so on. Your first option should be to determine whether an evaluation of this area is available via a third-party assessment (such as SSAE 18). If not, you'll need to work with your operations, procurement, and legal departments to determine your rights to audit the vendor in this area. Ideally, that right is spelled out in the contract. If not, your company will need to attempt to press for that right, possibly using the next contract renewal as negotiating leverage.

If the area is not covered by a third-party assessment and if you have the right to audit it, you will need to interview the vendor and review their documentation regarding their controls and processes, testing those controls as you're able. You will also want to see the requirements for disaster recovery controls, including recovery time objectives (how quickly your systems should be back up after a disaster) and recovery point objectives (how many days' worth of data you're willing to lose), spelled out in your contract. Determine how the vendor ensures compliance with the requirements in the contract.

While it is important that you understand your vendor's disaster recovery procedures, you should also expect that your company will have documented procedures regarding how they would recover in the event of a disaster at your vendor. This should include notification and escalation procedures, any necessary hand-offs between your company and the vendor during the recovery, and potential manual workarounds while waiting for recovery. It should also include contingency plans should the vendor be unable to recover for an extended period of time (or ever). Request information regarding the location of your data and regarding any replication in the architecture. If the data and infrastructure are replicated across multiple sites, your vulnerability and need for contingency plans decrease. If your systems are at a single location, it becomes more critical for your company to document contingency plans, which need to include a method for obtaining your data and bringing it back in-house if necessary.

For services using a shared responsibility model, such as AWS or Microsoft Azure, determine if your company is responsible for configuring backup service capabilities in different service regions to ensure disaster recovery capabilities are available in the event of a vendor service outage.

16. Determine whether appropriate governance processes are in place over the engagement of new cloud services by your company's employees.

Cloud computing makes it easy for business unit personnel to meet their needs without ever engaging corporate IT. Because most cloud services can be accessed via an Internet-connected browser, a business unit can engage a cloud vendor and outsource the systems and data related to one of their business processes without really having to tell anyone else. This has the potential to bypass all of the governance processes normally in place to ensure proper security of company data, interoperability of systems, appropriate support capabilities, and so on.

This step is most applicable to cloud computing.

How

This step can be divided into several substeps. These may require the assistance of operations or security personnel, finance, or others.

Policy Review company policies to determine whether the topic of engaging with cloud services has been addressed. Policies should be in place requiring company personnel to follow specific procedures when engaging vendors for this sort of service. If this policy exists, review it for adequacy. It should require that IT be engaged and that specific security and operational needs be addressed. Determine how employees are made aware of the policy. Also, determine how the policy is enforced. For example, if your company has a centralized procurement organization that must be engaged to sign contracts and pay invoices, you can use them as the gatekeeper for ensuring that proper procedures are followed for new engagements.

Inventory of Services Determine whether the organization maintains an inventory of vendors and authorized external services. Discuss with procurement or operations personnel to assess the existence of an inventory.

Controlled Access to Cloud Services Discuss with security infrastructure or

operations teams whether controls exist to manage or limit access to external services. This may often be accomplished via a web security gateway or web proxy, or through a specialized solution known as a cloud access security broker (CASB). If your company does not manage or restrict access to services, many services may be in use by employees without proper contractual agreements, or may be used in violation of license restrictions.

Cloud Configuration Management Particularly where PaaS or IaaS solutions are in use, authorized use of cloud services should be subject to usage guidelines. These may take the form of templates or policy guidance related to specific services that may be used, required security configuration, logging, and more. If employees have no restrictions on usage of these services, company data could be put at risk. Discuss this with the operations teams involved with configuring these services.

Cost Management Cloud services may be less costly than on-premise services, but this is not always the case. Ensure the organization has a process to monitor and review costs related to cloud services, particularly with PaaS and IaaS providers. Without appropriate controls here, employees could engage with expensive cloud solutions unnecessarily.

17. Review and evaluate your company's plans in the event of expected or unexpected termination of the outsourcing relationship.

Your company might terminate the outsourcing relationship in the future for many reasons. The provider could go out of business or discontinue the service you're using. You could be unhappy with the provider's cost or performance. You might engage in a new competitive bid at the end of your contract and another vendor may win the business.

If you can't bring the service back in-house or switch it to another vendor, you'll find yourself locked in with your vendor, which greatly damages your leverage to influence price and service quality. And if that company goes out of business, you'll experience significant business disruption.

This step is applicable to all forms of outsourcing.

How

Determine whether your company has a documented plan indicating how they would bring the functions back in-house (or move them to another vendor) if necessary. If bringing the function in-house is unrealistic, you should see evidence that alternative service providers have been identified. Ensure that an analysis has been performed regarding how long it would take to transition the services, and determine whether appropriate contingency plans are in place to keep the business running in the interim.

Look for contractual requirements for your vendor to return your data and assets upon request. If this has not been indicated in the contract, the vendor can hold your data hostage or can commingle it with other customers' data in such a way that it's nearly impossible to extract your data. Your company should require that your vendor deliver copies of your data to you periodically in an agreed-upon format (one that can easily be ported to a new application). Where applicable, ensure that code is put in escrow to protect against the vendor going out of business.

For IaaS and PaaS, your systems should be developed and deployed so that they are easily portable to new environments. Review your company's processes for ensuring that portability is a key goal in any development for cloud-based services.

Certain elements of cloud platforms like AWS may be unique to a provider. For example, Amazon's Simple Storage Service (S3) or serverless architectures like AWS Lambda may operate differently or may be incompatible with competing providers. If your company has committed to these services, an exit plan may be more complex.

18. If IT services have been outsourced, review the service provider's processes for ensuring quality of staff and minimizing the impact of turnover. If those services are being performed in remote locations, look for additional controls to ensure effective communication and hand-offs with the main office.

If service provider employees aren't qualified to perform their jobs or the provider experiences high levels of turnover, the quality of IT services will obviously be poor. This risk generally increases with outsourced operations, where turnover tends to be higher.

Outsourced operations that are performed remotely require greater coordination and communication; a failure can affect the quality of service received.

This step is most applicable to IT service outsourcing (onsite and offsite).

How

Review the contract to ensure job descriptions and minimum qualifications for each position are documented (such as education level, skills, experience). Pull a sample of supplier employees and verify that these minimums have been met. Review the provider's employee screening process to verify that appropriate background checks and qualification reviews take place prior to employment offers.

Determine how continuity of services is ensured in the event of turnover of service provider employees. Review staffing assignments and determine whether any single points of failure exist. Review cross-training processes.

Review the vendor's processes for providing training to update skills and knowledge. Request evidence that the training policy is being followed for a sample of employees.

For remote outsourcing in other countries, ensure that personnel with appropriate language skills are available and that proper coordination takes place with local offices. Look for the existence of periodic hand-off and status meetings between offices. Depending on the criticality of the service, it may be advisable to have an employee of your company at the remote site (or at least in the same city with easy access to the site) to act as your liaison and perform oversight of the operations.

Requirements for all of these items should be dictated in the contract. Review the contract to verify this.

Legal Concerns and Regulatory Compliance

19. Review and evaluate your company's right and ability to obtain information from the vendor that may be necessary to sup-

port investigations.

Your company may be required to perform e-discovery (electronic discovery) in support of litigation. Inability to produce applicable data may result in legal ramifications, as your company will be held legally responsible for your information, even if it's being stored and processed by a third-party provider. Your company may also need to perform investigations for its own reasons (for example, to investigate inappropriate activities such as fraud or hacking attempts). An inability to access appropriate logging and other data will prevent you from performing your investigations, leaving you with no real recourse when those inappropriate activities occur.

This step is most applicable to cloud computing.

How

Because cloud providers often commingle their customers' data, especially logging data, it is critical that you receive a contractual commitment from your vendors to support investigations. Review the contract and ensure this is documented as a requirement, including details as to the kind of investigative support you may need (such as specific log information and data format requirements) and the required response time for requests. It is also important that the contract define the responsibilities of both the cloud provider and your company related to e-discovery (for example, who is responsible for conducting the searches, for freezing data, for providing expert testimony, and so on). Review the vendor's processes to ensure that a formal process is in place to cooperate with customer investigations and to handle subpoenas for information.

If you find that the cloud provider is incapable of providing adequate support of investigations (or is unwilling to), your company may need to maintain copies of its data in-house. If this is the case, the costs of doing so will affect the benefits of the cloud relationship.

20. Review requirements for security breach notification. Ensure that requirements are clearly defined regarding when and how the vendor should notify your company in the event of a security breach and that your company has clearly defined response procedures when they receive such notification.

A security breach at your service provider not only puts your data and operations in jeopardy but may also have legal implications. For example, if you're hosting personal information and a security incident occurs, you may be legally required to notify all users who may have been affected. It's therefore critical that the service provider notify you in a timely fashion as to what has happened so that you can put together any necessary response.

This step is most applicable to cloud computing and dedicated hosting.

How

Review the contract for the existence of requirements and evaluate them for adequacy. Look for requirements regarding what constitutes a breach, how quickly a breach needs to be communicated to your company, and the method by which it should be communicated. Determine whether penalties have been built into the contract so that your company can be compensated for the costs incurred because of a breach.

Obtain a copy of your company's response procedures and ensure that they cover the basic information regarding what processes should be followed, who should be notified, when they should be notified, and how any compensating processes should be enacted.

If a breach has been reported, review for evidence that the correct processes were followed.

21. Determine how compliance with applicable privacy laws and other regulations is ensured.

Regardless of where your data is stored and who manages it, you are still responsible for making sure that your company is complying with all applicable laws and regulations. If your company is found to be in violation of applicable laws and regulations, it can lead to stiff penalties and fines, a damaged reputation, lawsuits, and possibly the cessation of the company. The fact that it was being managed by a cloud provider will not be an acceptable defense.

This step is most applicable to cloud computing and dedicated hosting.

How

Review the contract and look for language requiring that the vendor obtain third-party certification regarding compliance with applicable regulations (such as PCI, GDPR, and HIPAA), as well as requiring SSAE 18 assessments. If you find such language, review evidence that your company is requesting these reports from the vendor and reviewing the results. Review the most recent reports for any issues that have been noted and determine how your company is tracking those issues.

The contract should require that the vendor disclose where your data is located and provide assurance that they are complying with local privacy requirements related to your data. The contract should also contain language specifying who is liable in the event of noncompliance.

If the contract does not require these certifications and/or the vendor will not undergo these assessments, determine how your company is certifying compliance with applicable regulations. If this is the case, your company should seriously consider a withdrawal strategy.

For services using a shared responsibility model, such as AWS or Microsoft Azure, the customer may have responsibility for compliance. In these services, it might be very easy for an application developer to enable system mirroring or failover (such as with "availability zones"). These teams should ensure that any data transfer associated with high-availability operations is in compliance with regulations and company policy.

22. Review and evaluate processes for ensuring that the company is in compliance with applicable software licenses for any software hosted offsite or used by nonemployees.

Using software illegally can lead to penalties, fines, and lawsuits. If companies do not develop processes for tracking the legal usage of software and licenses, they may be subject to software vendor audits and will not be able to account properly for the company's use of the vendor's software. This becomes more complex when dealing with outsourced operations, as purchased software may be hosted on third-party infrastructure and/or used by outsourced service provider employees, or may be installed on systems running at providers like AWS. You must ensure that copies of the software continue to be tracked and that the usage is in compli-

ance with the terms of the agreement.

This step is applicable to all forms of outsourcing.

How

Look for evidence that the company maintains a list of enterprise software licenses (such as Microsoft Office, ERP application accounts, and so on) and that it has developed a process for monitoring the usage of those licenses and compliance with the terms of the agreement. Ensure that this process incorporates copies of your software that are hosted by a third party and copies of the software used by nonemployees.

Tools and Technology

Several organizations, such as the Cloud Security Alliance and Center for Internet Security, offer resources to help organizations assess various cloud providers and platforms. In addition, many of the major providers have automated scripts or configuration assessment tools that could be used to get an overview of the environment and highlight any serious issues. Use of these tools may not be free, or may only be useful in the context of a particular service. For example, you would not want to use the AWS Config tool if your organization uses Google Cloud. Some available resources in this area include the following:

Tool	URL
AWS Config	https://aws.amazon.com/config
Microsoft Secure Score	https://docs.microsoft.com/en-us/office365/securitycompliance/microsoft-secure-score
Cloud Security Alliance Security Trust Assurance and Risk (STAR) Program	https://cloudsecurityalliance.org/star
Center for Internet Security	https://www.cisecurity.org/cis-benchmarks/
Shared Assessments Standardized Information Gathering (SIG)	https://sharedassessments.org/sig/
BitSight	https://www.bitsight.com
Security Scorecard	https://securityscorecard.com/

Knowledge Base

You can find a considerable amount of information online regarding cloud computing. The major providers have developed extensive free libraries, as well as paid training classes to encourage customers to adopt cloud services. Some of the available information on the topic includes the following:

Site	URL
Amazon Web Services	https://aws.amazon.com
Google Cloud	https://cloud.google.com
IBM Cloud	https://www.ibm.com/cloud
Microsoft Azure	https://azure.microsoft.com/en-us/
AWS Well-Architected Framework	https://aws.amazon.com/architecture/well-architected/
Cloud Security Alliance	https://cloudsecurityalliance.org
NIST Cloud Computing	https://csrc.nist.gov/projects/cloud-computing
What Is SaaS?	https://www.salesforce.com/saas
ISACA Cloud Computing Guidance	www.isaca.org/Knowledge-Center/Research/Pages/Cloud.aspx

Regarding materials on auditing general (non-cloud-specific) IT outsourcing, your best bets are to search for relevant materials on the ISACA website (http://isaca.org/), specifically within the COBIT framework.

Master Checklist

The following table summarizes the steps listed herein for auditing cloud computing and outsourced operations.

Auditing Cloud Computing and Outsourced Operations

Checklist for Auditing Cloud Computing and Outsourced Operations	
❑	1. Review the audit steps in the other chapters in this part of the book and determine which risks and audit steps are applicable to the audit being performed over outsourced operations. Perform those audit steps that are applicable.
❑	2. Request your service provider to produce independent assurance from reputable third parties regarding the effectiveness of their internal controls and compliance with applicable regulations. Review the documentation for issues that have been noted. Also, determine how closely these certifications match your own company's control objectives and identify gaps.
❑	3. Review applicable contracts to ensure that they adequately identify all deliverables, requirements, and responsibilities pertinent to your company's engagement.
❑	4. Review and evaluate the process used for selecting the outsourcing vendor.
❑	5. Determine how your data is segregated from the data of other customers.
❑	6. Review and evaluate the usage of encryption to protect company data stored at and transmitted to the vendor's site.
❑	7. Determine how vendor employees access your systems and how data is controlled and limited.
❑	8. Review and evaluate processes for controlling nonemployee logical access to your internal network and internal systems.
❑	9. Ensure that data stored at vendor locations is being protected in accordance with your internal policies.
❑	10. Review and evaluate controls to prevent, detect, and respond to attacks.
❑	11. Determine how identity management is performed for cloud-based and hosted systems.
❑	12. Ensure that data retention and destruction practices for data stored offsite comply with internal policy.
❑	13. Review and evaluate the vendor's physical security.
❑	14. Review and evaluate your company's processes for monitoring the quality of outsourced operations. Determine how compliance with SLAs and other contractual requirements is monitored.
❑	15. Ensure that adequate disaster recovery processes are in place to provide for business continuity in the event of a disaster at your service provider.

- ❑ 16. Determine whether appropriate governance processes are in place over the engagement of new cloud services by your company's employees.
- ❑ 17. Review and evaluate your company's plans in the event of expected or unexpected termination of the outsourcing relationship.
- ❑ 18. If IT services have been outsourced, review the service provider's processes for ensuring quality of staff and minimizing the impact of turnover. If those services are being performed in remote locations, look for additional controls to ensure effective communication and hand-offs with the main office.
- ❑ 19. Review and evaluate your company's right and ability to obtain information from the vendor that may be necessary to support investigations.
- ❑ 20. Review requirements for security breach notification. Ensure that requirements are clearly defined regarding when and how the vendor should notify your company in the event of a security breach and that your company has clearly defined response procedures when they receive such notification.
- ❑ 21. Determine how compliance with applicable privacy laws and other regulations is ensured.
- ❑ 22. Review and evaluate processes for ensuring that the company is in compliance with applicable software licenses for any software hosted offsite or used by nonemployees.

CHAPTER 17

Auditing Company Projects

In this chapter we will discuss key controls to look for when auditing the processes used for managing company projects, including understanding the following as it relates to information technology (IT) audit project management:

- Keys to successful project management
- Requirements gathering and initial design
- System design and development
- Testing
- Implementation
- Training
- Wrapping up the project

All of the other chapters in this part of the book have dealt with how to audit specific technologies and processes that are already in place and operating in a production environment (such as operating systems, data centers, applications, and so on). However, before any system or process can be implemented, a project must be funded and staffed to develop or procure that system or process. If proper disciplines are not followed throughout the project, the chances of failure in meeting requirements and/or of inefficient use of company assets are greatly increased.

NOTE This chapter does not discuss the concept of early involvement, which was covered in Chapter 1. The early-involvement concept is used to build internal controls into the systems and processes being developed at your company. Instead, this chapter deals with the processes used to ensure that those projects are being managed efficiently and effectively. The concept of building in controls at the start of the project certainly can be merged with an audit of project management processes, but they are two different topics. The early-involvement concept is briefly discussed in this chapter only as a reminder of how it can be used as part of a project audit.

Background

Proper project management techniques are essential elements in the success of any company endeavor. These techniques help to ensure that pertinent requirements are gathered and tested, project resources are used efficiently, and all elements of the system are tested properly. Without such techniques, it is likely that the system being developed won't work or won't perform as expected by key stakeholders. This leads to rework and extra costs to the company (and can sometimes lead to people losing their jobs).

Good project management does not ensure success, but it improves the chances of success. The intent of this chapter is not to provide a training course on the basics of project management or the software development life cycle (SDLC), but is instead intended to provide a list of basic risks you should review when auditing a systems project to ensure that the most essential project management disciplines are being followed.

NOTE The words *software, system,* and *process* are used interchangeably in the following test steps and in conjunction with one another. They are intended to represent "the thing that is being developed by the project team." The use of one word versus another in a given test step is not intended to convey any specific meaning.

Project Auditing Essentials

In this section, we will define the goals of a project audit and define the basic approaches to and elements of auditing projects.

High-Level Goals of a Project Audit

Project audits are performed to identify risks to the success of company projects. This chapter deals specifically with IT projects (such as software development, infrastructure deployment, and business application implementation), but the concepts could apply to any sort of project.

Following are some of the high-level goals of a project audit:

- Ensure that all appropriate stakeholders are involved in the development of requirements and testing of the system and that frequent and effective communication occurs with all stakeholders. Failure to gather customer requirements and to obtain ongoing customer involvement and buy-in lead to software, systems, and processes being developed or procured that do not align with business needs.
- Ensure that project issues, budgets, milestones, and so on are recorded, baselined, and tracked. Without these mechanisms, projects are more likely to go over budget and over schedule with unresolved issues.
- Ensure that effective testing encompasses all system requirements. Inadequate testing leads to unstable, low-quality systems that fail to meet customer requirements.
- Ensure that appropriate documentation is developed and maintained. Incomplete or out-of-date technical and user documentation could increase cost and cycle time to maintain the software, increase support and training costs, and limit the system's usefulness to the customer.
- Ensure that adequate training is provided to end users upon implementation. Inadequate training leads to systems, processes, and software that go unused or that are used improperly.

Basic Approaches to Project Auditing

Two basic approaches can be taken with project auditing. The first approach is quick and short term—the in-and-out approach. The second approach takes a long-term view of the project and is a more consistent approach.

The short-term approach can be challenging; auditors choose a point in the project to perform their audit, and then they review the project as of that point in time and make a judgment based on what has happened and what is planned. This approach suffers from two major downfalls.

First, it is difficult for the auditors to affect the phases that have already been completed. For example, the user acceptance testing phase is a bad time to learn that poorly controlled processes were used during the project definition phase. The project team either has to revisit and rework to improve earlier tasks or move forward, knowing that problems exist and hoping for the best. Either way, the auditors' input is not very timely and may even be seen as counterproductive, damaging relationships between the auditor and his or her customers.

Second, fully evaluating phases that have not yet begun is difficult. Auditors might be able to review plans for user acceptance testing at the beginning of the project, for example, but until those plans are fully developed and being executed, auditors will find it difficult to evaluate their true effectiveness.

The longer-term, or consistent involvement, approach allows auditors to perform some assessment activities during each major phase of the project. Each audit evaluates the processes within the current phase while simultaneously assessing and providing input on plans for future phases. This is an effective means of auditing projects and leads to a more collaborative approach with audit customers. On the negative side, this approach stretches out the audit over a long period and can be difficult to schedule. However, the positives far outweigh the negatives.

If the project spans an exceptionally long time, the auditors might consider one of two approaches:

- Release interim audit reports after each major project phase so that the information in the report doesn't become too stale.
- Meet with the project manager to discuss issues on a regular basis (such as every two weeks). At this meeting, the auditors can communicate new risks to the project discovered since the last meeting and also follow up on the status of previous issues to determine whether remediation is complete. If, in the auditors' opinion, the project risk is increasing to an unsatisfactory level or if issues are not being mitigated, the auditors can escalate to a higher level of management at their discretion. The auditors should reserve the right to issue a full-scale audit report at any time, but by trying to work with the project manager first, issues will more likely be resolved without escalation and without the issuance of interim audit reports.

Waterfall and Agile Software Development Methodologies

If you're going to be auditing a software development project, it's important to understand what software development methodology is being used by the project team, as that will inform how you go about conducting your audit of that project. There are two basic models of software development methodology, although you might find that your company is using a hybrid or customized version.

Waterfall Methodology

This is the more "traditional" of the two methodologies, where each phase of the software development life cycle (e.g., requirements gathering, design, construction, testing, implementation) is performed in sequence. For example, if the project is following a pure waterfall methodology, all requirements will be gathered before moving to the design phase, and once the design phase has begun, the requirements gathering phase will be closed out, never to be reopened.

This methodology works well in cases where there is a clear picture from the beginning of what the final deliverable should look like. For example, it tends to be appropriate for the implementation of large enterprise financial systems. But it is generally not the methodology of choice when speed is the key to success.

Agile Methodology

The agile methodology was essentially created because of perceived disadvantages with the waterfall methodology. While the waterfall methodology works well in some scenarios, it has a tendency to be slow and rigid. In contrast, the agile methodology is built to be fast and flexible. It follows an iterative process, where each phase of the life cycle is executed multiple times in a series of "sprints." Small capabilities will be implemented quickly, and then the project team will start to work on the next small set of changes.

This methodology works well in cases where speed is the key to success. For example, it tends to be the methodology of choice for adding capabilities and features to consumer websites, where there's really no "end" in sight, but rather it's critical to quickly keep up with the latest trends and developments.

Impact on Auditing Projects

You'll notice that this chapter does not contain separate steps for auditing a project using waterfall versus auditing a project using agile. That's because the key controls you'll want to see in place will be the same, regardless of how the project team decides to manage their project life cycle. However, this will likely be more challenging for the auditor when auditing projects using the agile methodology. That's because agile projects tend to be less structured and often use the "need for speed" as a reason to minimize processes, thereby removing key checks and balances. Your job as an auditor will sometimes be to challenge that thinking. Moving quickly is not an excuse for removing the internal project controls that are critical for ensuring the long-term success of the project. The implementation of those controls might look different in an agile project, but they still need to be there. Also, due to the iterative and incremental nature of agile projects, you might find the need to perform certain parts of the audit multiple times (for example, over a series of sprints) to be sure the controls are being implemented consistently.

Seven Major Parts of a Project Audit

Projects can be separated into seven major parts (Figure 17-1), each of which requires disciplines and controls that we will evaluate during the project audit:

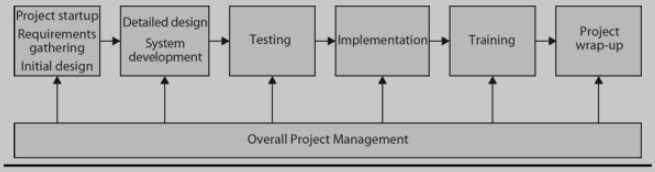


Figure 17-1 Major elements of projects

1. **Overall project management** Mechanisms that should be used throughout the project, such as issue tracking, project documentation, and change management.
2. **Project startup, requirements gathering, and initial design** Covers the birth of a project, where the need for the project is established, requirements are gathered, and initial design and feasibility studies are performed.
3. **Detailed design and system development** Covers the "meat" of the project, where the code is written, the product is procured or implemented, the processes are developed, and so on.
4. **Testing** The system, software, or process is tested to ensure that it meets requirements.
5. **Implementation** The system, software, or process is implemented or installed into a production environment.
6. **Training** Covers the activities for training end users on using the system, software, or process that has been developed and implemented.
7. **Project wrap-up** Covers post-implementation activities.

NOTE These project elements will not necessarily be performed in this precise order, nor will they necessarily be performed sequentially. Multiple iterations of each phase may exist, and some may be performed in parallel with each other (for example, user training is often performed in parallel with testing and implementation). However, just about every project should have some of each of these elements.

The rest of this chapter will focus on key audit steps and tests to perform with regard to these seven categories.

Test Steps for Auditing Company Projects

To provide some context and structure, the test steps in this section are provided according to project phase. However, the steps don't always work out as neatly as what has been laid out; each process has unique situations and requirements

to consider. For example, the time to address a step from the testing section may occur during the requirements gathering phase.

You should perform each step at the point in the project at which it makes the most sense, based on how the project is run. It is critical for the auditor to understand the methodology of the project and adjust his or her approach accordingly. For example, if the project is using agile development, where each project phase is executed multiple times, you may need to audit each phase concurrently or possibly even multiple times. The controls required for a project generally will be the same, regardless of the project methodology, but matching the audit phase to the project and coordinating the timing will be more difficult for some types of projects than for others. Part of the planning process should involve obtaining an understanding of the project methodology used and determining the appropriate timing and method for accomplishing the steps in this program.

When planning the audit, you should determine what project management tool is used by the project team and become familiar with the tool and its terminology. This will allow you to "speak the same language" as the people you are auditing and further enhance credibility.

In addition, some of these steps may be overkill for smaller projects. You should use judgment in determining which of these risks are material enough to address for each specific project.

Finally, these steps are written so that they can be used for any sort of IT project, whether it involves acquiring or developing software, procuring new technology, or developing a process. Use your judgment to determine which steps are most applicable based on the kind of project being audited.

Overall Project Management

The steps in this section should usually be performed thoroughly at the beginning of the project and then again lightly during each phase of the project to ensure that the disciplines are still being followed. Project management may start out strong, but it often wanes as people become busy and are scrambling to meet deadlines.

1. Ensure that sufficient project documentation and software development process documentation (if applicable) have been created. Ensure that the company's project methodology standards are being followed.

This sort of documentation will increase the likelihood that the project is being implemented in a disciplined manner and is following your company's established standards and methodologies. This, in turn, greatly increases the chances that the project will be executed successfully and will produce the business value desired. This documentation can also benefit future projects, allowing the company to leverage past efforts. Finally, your company may have specific standards for executing projects based on either internal or regulatory requirements.

How

Review copies of existing project documentation and compare it with your company's standards and requirements. The documents required will vary by company, but look for documents covering areas such as milestones, work breakdown structure (WBS), project approach, statement of work (SOW), requirements, test plans, and design documents. Obtain a copy of your company's project methodology standards and compare them with the methodology being executed on the project under review. When reviewing this documentation, look for evidence of adequate project and resource planning. Performing this step is not an exact science; you are trying to develop a feel for the overall level of documentation and processes established for the project. Some of this documentation will be examined in more detail during later steps. In this step, the auditor should obtain the document(s) that constitute the basic project plan and determine whether customer needs, deliverables, objectives, and scope are clearly defined.

2. Review procedures for ensuring that project documentation is kept up-to-date.

For reasons mentioned in the preceding step, this documentation enhances the quality of current and future projects. However, if it is allowed to become outdated, it quickly becomes useless.

How

Through interviews with the project team, understand the processes in place for updating these documents when necessary. Look for evidence that updates have been made.

3. Evaluate security and change management processes for critical project documentation.

If proper security and change controls are not in place, unauthorized, inaccurate, and/or unnecessary changes may be made to the project documentation.

How

Ensure that files containing project documentation are locked down and can be modified only by an appropriate subset of project personnel (using techniques described in Chapter 7 for Windows files and Chapter 8 for Unix or Linux files). Interview project personnel to understand processes for changing critical project documents. Ensure that an approval process is required before changes are made to significant project documents and that the approval process cannot be circumvented. The documents that constitute the basic project plan (such as customer needs, deliverables, objectives, scope, budget, risks, and communication strategies) should be baselined early in the project so that they cannot be changed without agreement from all key stakeholders.

4. Evaluate procedures for backing up critical project software and documentation. Ensure that backups are stored offsite and that documented procedures exist for recovery.

If these processes are not in place, a system crash or data center disaster could result in the permanent loss of project software and documentation.

How

Review processes or scripts indicating that project data is backed up and stored offsite. Review written recovery procedures, ensuring that they specify what steps are to be performed for recovery, the order of those steps, and who is to perform

each step. Note that these written recovery procedures may not be created for a particular project. They may instead be part of the IT team's standard recovery procedures for lost files. Consider requesting a test recovery of the critical project material.

5. Ensure that an effective process exists for capturing project issues, escalating those issues as appropriate, and tracking them to resolution.

During the course of any project, issues inevitably will arise regarding the project itself or the system, process, or software being developed. Without a robust method of capturing and resolving those issues, some issues will likely "slip through the cracks" and not be resolved, resulting in failures in the product or failures in execution of the project.

How

Review the issues database, spreadsheet, or other method established for recording and tracking issues. Ensure that the issue-tracking tool records adequate information regarding each issue, including description of the issue, priority level, due date, latest status, and resolution information. Ensure that controls exist over the tool used to track issues, such as backups to prevent the loss of information, and security and change management controls to prevent unauthorized updates and to, for example, prevent issues from being marked as "closed" without alignment from appropriate stakeholders. Review processes for escalating issues and for ensuring that issues are tracked to resolution. Review the issues list for evidence that issues are being closed. Interview customers to ensure that the process is working.

6. Ensure that an effective process exists for capturing project change requests, prioritizing them, and dispositioning them.

During most projects, requests for additional functionality will arise after the project has commenced and the requirements have been established and approved. Without a method for ensuring that these requests are approved, prioritized, and dispositioned, these requests may get lost and/or the scope of the project will shift continually, making it impossible to execute the project effectively. A change request process will help prevent *scope creep* and provide for ongoing discussions with the project's customers regarding how change requests will affect the project's budget and schedule.

How

Review the change management process and ensure that it provides allowance for entering, ranking, and approving change requests. Verify that it covers changes to scope, schedule, budget, requirements, design, and so on (that is, all major elements of the project). Ensure that it records adequate information regarding each change request, including description of the request, priority level, latest status, approval, and resolution information. Select a sample of change requests and walk them through the process, ensuring that proper approvals were received prior to final resolution.

7. Verify that a project schedule has been created and that it contains sufficient detail based on the size of the project. Ensure that a process is in place for monitoring progress and reporting significant delays.

Project schedules are used to ensure that the project is on track, resources are being used effectively, and all tasks have been accounted for and scheduled.

How

Review the project schedule, and look for items such as a work breakdown, milestone dates, task dependencies, and the critical path. Look for evidence that the schedule is followed and kept up-to-date. Seek explanations for any significant deltas. Ensure that an escalation procedure exists for any significant schedule or resource overruns, and review evidence that the process has been used. One potential way for the project to ensure schedule compliance is to create strategic points in the life cycle in which the project passes through a "tollgate process." At these points, the project team reports to a review panel to convey the status of the project, successes and issues, and progress versus the schedule and budget. This helps identify struggles and failures quickly as they occur.

8. Ensure that a method is in place for tracking project costs and reporting overruns. Ensure that all project costs, including labor, are considered and tracked.

Without these mechanisms, project budgets can often be exceeded, and often the appropriate levels of management are not made aware of these issues. Management presumably has placed a cap on the funding for a specific project. If all relevant expenses are not tracked, management will be unaware if that cap has been exceeded and will therefore be unable to make an informed decision regarding how to proceed.

How

Obtain a copy of the budget, and compare it with expenses to date. Seek explanations for any significant deltas. Ensure that the budget includes all costs associated with the project, including labor, software, and hardware. Ensure that an escalation procedure exists for any significant cost overruns, and review evidence that the process has been used. See the tollgate description from step 7 for a potential review methodology.

9. Evaluate the project leadership structure to ensure that both the business and IT are represented adequately. Ensure project sponsors have been clearly established and that they accept and understand their role.

Except for some pure infrastructure projects, most projects are undertaken at the request of the business to meet a business need. If the key business stakeholders are not part of the overall leadership and approval structure for the project, the odds of the project getting off track from the business needs increase because information and decisions about the project will be handled by IT people, who may not have the perspective necessary to make all decisions. Remember that IT exists to support the business, and therefore the IT organization should not be making decisions regarding the business's IT needs in a vacuum.

Conversely, IT personnel also should be part of the structure, because they generally bring important knowledge and perspective regarding the elements of

success for IT-related projects. They can help ensure that the system is being designed in a cost-effective way that enables long-term support. Systems that are developed without IT involvement are far more likely to have issues with scalability, interoperability, and supportability. They are also more likely to experience deployment issues, resulting in negative impacts to the project schedule.

On large-scale projects, a large number of business and IT leaders might be involved. When changes are requested for requirements, priority, or funding, there should be a limited number of sponsors that clearly have authority to make the final decision; otherwise, it might be impossible to arrive at a decision.

How

Obtain a copy of the project's leadership structure, and look for evidence that both business and IT leaders and stakeholders are represented. Also, look for evidence that project sponsors have been identified, that the number of project sponsors is limited and appropriate, that they have accepted their role as sponsor, and that they understand what that role entails.

Project Startup, Requirements Gathering, and Initial Design

10. Ensure that appropriate project approval processes were followed prior to project initiation.

Projects should not be initiated without approval from the appropriate members of management who are authorized to allocate resources and funds to new projects.

How

Review evidence that the project passed through the company's standard approval process. If no such process exists, review evidence that the appropriate manager(s) approved the project prior to startup. Look for evidence that alternative and cost-benefit analyses were performed. Ensure that cost-benefit analyses considered not only the project start-up costs but also ongoing costs, such as software maintenance, hardware maintenance, support (labor) costs, power and cooling requirements for system hardware, and other factors. This element is often omitted erroneously, leading to misinformed decisions. Startup costs are only a fraction of the total ongoing costs for implementing a new system. A multiyear (five years is often a good target) total cost model should be developed as part of the initial project analysis.

11. Ensure that a technical feasibility analysis has been performed along with, if applicable, a feasibility analysis by the company's legal department.

Prior to the startup of an IT project, qualified technical architects, network personnel, database administrators, and other applicable IT experts should agree that the proposed concept will work within the company's environment. If these experts are brought in early, it is likely that the technical professionals can find a way to make the concept work. However, if they are brought in after key elements of the system have been developed or procured, it may be determined that the solution is not technically feasible, leading to costly rework or discontinuation of the project. Likewise, it is important to engage the legal team to ensure that regulatory requirements are considered in the project.

How

Review evidence that appropriate technical and legal personnel were involved in the initial project proposal and that they agreed to the feasibility of the project.

12. Review and evaluate the requirements document. Determine whether and how customer requirements for the project are obtained and documented before development takes place. Ensure that the customers sign off on the requirements and that the requirements encompass standard IT elements.

Systems, software, and processes should be built based on the requirements of the end users. If end user requirements are not captured and approved by the customers, the product likely will not meet the customers' needs, requiring rework and changes. In addition, certain standard IT elements should be included in the requirements definition of any system. Customers may not be aware of these elements and therefore require guidance from the IT team. Establishing clearly defined requirements will also assist in discussions down the road regarding what is a bug fix (that is, when the system is not functioning as designed) and what is an enhancement request (that is, when the system is functioning as designed but the customer wants to make a change), which can be an important distinction depending on your IT organization's support and funding models.

How

Review project documentation for evidence that customer requirements were gathered. Ensure that all key stakeholders, including the project sponsors, were involved in this process. Look for evidence that the key stakeholders agreed to the final list of requirements.

Review customer requirements to ensure that they are documenting business requirements and are not dictating a solution. Often, business leaders will speak to a vendor or read an article and decide to create a project for the purpose of implementing a specific product or technology. However, that particular product may not be the most effective fit for your particular company's situation. For example, it may not fully meet the business's needs, it may be redundant with other products currently used in the environment, or it may not interface well with existing company technologies. It is critical that the customer focus on determining and documenting the business requirements and allow the IT organization the flexibility to determine what tool(s) best meet those requirements.

Ensure that the requirements encompass standard IT elements such as the following:

Distributed and centralized processing requirements (for example, the location of the storage and processing in a multitier architecture)	Service level agreements (for example, system availability, speed of response to problems)
Response time (for online transactions)	Interface requirements
Security	Backup/recovery/restart requirements
Execution frequency	Hardware requirements
Data retention requirements	Capacity, including needs for future anticipated growth
Requirements for output distribution	Fault tolerance and redundancy
Screen definitions	

If this project is intended to replace an existing system, look for evidence that an analysis of the current system was performed to determine what is working well and what is not. Also, look for evidence that the existing system has been carefully analyzed and all the existing use cases (functions) that it fulfills are met with the new system (or that there is a conscious decision to forego those capabilities in the new system). The results of this analysis should be reflected in the requirements documentation. (The requirements should call for the new system to do the things that work well in the old system and to improve on the things that don't.) Error logs and backlog requests from the old system can aid in the effort to determine what is not working well.

13. Evaluate the process for ensuring that all affected groups that will be helping to support the system, software, or process are involved in the project and will be part of the sign-off process, indicating their readiness to support it.

Multiple organizations in the IT environment are usually involved in supporting any new system, including network support, operating system support, database support, data center personnel, IT security, and the helpdesk. If these organizations are not involved in the project early on (and on an ongoing basis), they may not be prepared to support the system after it is ready and/or the system may not be in compliance with applicable standards and policies. They also might not have the time available to perform necessary project tasks, therefore putting the project schedule in jeopardy.

How

Review evidence that other affected IT organizations have been notified of the project, are involved in the project on an ongoing basis, and are part of the approval process as it relates to their readiness to support it.

14. Review the process for establishing the priority of requirements.

Often, more system requirements exist than can be encompassed in the project (or at least in the initial phase of the project). The most critical requirements must be identified, prioritized, and implemented.

How

Look for evidence that the requirements were prioritized and that the key stakeholders approved the prioritization.

15. Determine whether the system requirements and preliminary design ensure that appropriate internal control and security elements will be designed into the system, process, or software.

Internal controls are necessary to protect company systems and to ensure their integrity. It is much easier to build controls into new systems up-front than to attempt to add them post-implementation.

How

This step is referring to the execution of the early-involvement concepts discussed in Chapter 1. The auditor will need to determine what sorts of controls he or she would audit the system for post-implementation and ensure that those controls are being designed into the system. Appropriate application controls and infrastructure controls should be considered. The other chapters in Part II of this book provide most of the detail needed to perform this step. Although those chapters discuss the techniques for auditing systems, processes, and software post-implementation, the same information can be used for providing input as to what controls need to be built in during design. In addition, it might be appropriate to assign a financial/operational auditor to the project to ensure that the proper business controls are built into the system logic and workflows.

16. If the project involves the purchase of software, technology, or other external services, review and evaluate the vendor selection process and related contracts.

Purchasing a product from an outside vendor is usually a significant investment and represents a commitment to that vendor's product. If the process for selecting the vendor is inadequate or the contract does not provide the company with adequate protection, it can lead to the purchase of products that do not meet the requirements of the project and a lack of legal recourse.

How

Review the vendor selection process for elements such as these:

- Ensure that products from multiple vendors are evaluated as to their ability to meet all project requirements and their compatibility with the company's IT environment. This not only helps you select the best product for your requirements but also provides for competitive bidding and lower prices.
- Ensure that a cost analysis has been performed on the products being evaluated. This analysis should include all relevant costs, including product costs, one-time startup costs, hardware costs, licensing fees, and maintenance costs.
- Determine whether the vendors' financial stability was investigated as part of the evaluation process. Failure to do so may result in your company signing up with a vendor that goes out of business, causing significant disruption to your operations as you attempt to move them to another vendor.
- Determine whether the vendors' security practices were evaluated as part of the evaluation process. Failure to do so increases the chances that the vendor will be unable to protect themselves from cybersecurity incidents, potentially putting your company's data at risk, the integrity of the software they deliver to your company at risk, and/or the vendor's availability to support your company at risk.
- Determine whether the vendors' experience with providing support for the product for similar companies in the industry was evaluated. This may include obtaining and interviewing references from companies that currently use the product. You generally want to use vendors that have already demonstrated that they can perform the types of services you require at a scale similar to yours.
- Ensure that the vendors' technical support capabilities were considered and

evaluated.

- Ensure that each vendor was compared against predefined criteria, providing for objective evaluations.
- Determine whether there was appropriate involvement of procurement personnel to help negotiate the contract, of operations personnel to provide expert evaluations as to the vendor's ability to meet requirements, and of legal personnel to provide guidance on potential regulatory and other legal ramifications.
- After a vendor is chosen, ensure that the contract clearly identifies deliverables, requirements, and responsibilities. The contract should specify how performance will be measured and penalties for nonperformance or delayed performance. It also should provide conditions for terminating the agreement. Basically, anything you expect from the vendor needs to be specifically outlined in the contract.
- Ensure that the contract contains a nondisclosure clause preventing the vendor from disclosing company information.
- Ensure that the contract contains a "right to audit" clause that allows you to audit vendor activities that are critical to your company.
- Where applicable, ensure that code is put in escrow to protect against unavailability should the vendor go out of business and that an appropriate exit strategy is in place should the relationship between your company and the vendor be discontinued for any reason.

Detailed Design and System Development

How

Look for the equivalent of a detailed design document and for evidence of customer approval. Note that nontechnical personnel may not be in a position to understand the detailed design document, depending on how it is written. If this is the case, ensure that compensating design reviews or "use case" catalogs have been developed that allow the stakeholders to understand the planned design elements.

19. Review processes for ensuring ongoing customer involvement with the prioritization of tasks on the project.

Most projects experience fluidity, with the initial set of requirements rarely ending up as the final set of requirements. If key stakeholders are not involved throughout the project, the project runs the risk of straying away from customer requirements, and decisions can be made that are not in alignment with customer wishes.

How

Determine whether a direction-setting group has been established and contains key customers and whether they are involved in project decisions on a regular basis. Consider interviewing a small sample of customers to obtain their opinions on customer involvement. Look for evidence of periodic project review meetings and periodic communication with key stakeholders.

20. Look for evidence of peer reviews in design and development.

This quality-control discipline, which involves a review of code and configuration by the developer's peers, can help increase the odds that the system will be designed with sound logic and a minimum of errors.

How

Determine whether peer reviews are required by the process, and look for evidence that they are actually occurring.

21. Verify that appropriate internal controls and security have been designed into the system.

See step 15 for further information.

How

Validate (either through interviews or design reviews) that the input you provided in step 15 has been encompassed in the design of the system.

Testing

22. Verify that design and testing are occurring in a development/test environment and not in a production environment.

Failure to perform design and test work in dedicated environments could result in disruption of normal business activities.

How

17. Ensure that all requirements can be mapped to a design element.

A defined process for tracing requirements to the system design will provide assurance that all requirements are addressed, including such standard IT elements as interfaces, response time, and capacity.

How

If a requirements trace map exists, review it and verify that all requirements are represented and mapped to a design element. If a trace map doesn't exist, review the process for ensuring that all requirements are encompassed.

18. Verify that the key stakeholders have signed off on the detailed design document or "use case" catalog.

The detailed design document is used for the design of the system, software, or process. A "use case" catalog may be created for the project customers as a less technical document that details the system design from a more functional standpoint (that is, detailing exactly how each required system function will be implemented). This document will specify the success and failure criteria for each scenario within the application. For example, customer checkout for an e-commerce application would be a use case that would lead to multiple steps (such as verifying that the user logged in, validating the shipping address, and so on), all of which would be documented in detail.

If key stakeholders have not signed off on these appropriate documents, the chances are greater that the output of the project may not meet their needs.

View evidence that the environments being used during development and testing are separate from the environment being used for production. View a layout of the architecture, and validate segregation of the environments. View project member logins to the various environments, and confirm that the servers being used for design, testing, and production match the architecture layout. Also, ensure that the test environment closely mirrors the production environment. Otherwise, a successful test of code in the testing environment may not be an indicator that the code will work in the production environment or that it will be scalable with the production load.

23. Review and evaluate the testing process. Ensure that the project has an adequate test plan and that it follows this test plan.

Testing the system, software, or process will provide assurance that it works as intended.

How

Review the test plan for several elements. First, determine whether the test plan includes the following:

- **Unit testing** Testing of individual system modules or units or groups of related units
- **Integration testing** Testing of multiple modules or units to ensure that they work together correctly
- **System testing** Testing of the overall system by the development team
- **Acceptance testing** Testing performed by end users to validate that the system meets requirements and is acceptable
- **Regression testing** Retesting select areas to ensure that changes made to one part of the system did not cause problems in other parts of the system

Then review the plan for the following:

- Ensure that the test plan and related procedures and test cases are repeatable so that they can be used for regression testing and for future releases.
- Ensure that test plans and cases go through a peer review to ensure quality.
- Determine whether the test plan includes testing of bad/erroneous data, system error handling, and system recovery.
- Determine whether the test plan includes testing of security and internal controls.
- Ensure that results of testing are fully documented.
- Ensure that gaps identified during testing are documented, tracked, resolved, and retested.
- Ensure that the gap/bug-tracking process is approved up-front. This process needs to be baselined and a system of controlled change established quickly, or it can become a mess, with code being pulled in and out of production haphazardly.
- Ensure that the project team has agreed to metrics to be captured and reported during testing and that these metrics are reported in a timely fashion to the appropriate members of the project leadership.
- Ensure that the test plan includes the testing of performance requirements and thresholds.
- Ensure that each test case identifies the product, component, or module that it is testing.
- Evaluate the process for ensuring that all major functionality is tested and that all key logic paths are identified and tested. If a use case catalog is used, evaluate the process for ensuring all elements of all use cases are tested.
- Ensure that test data has been created and that the customers agree that test data is valid. Determine whether confidential information has been removed from the test data, as access provided to test environments is often broader than access provided to production environments. If doing so is not practical, ensure appropriate controls have been implemented over access to sensitive test data.
- Determine whether test steps define expected results and customer acceptance criteria.
- Ensure that all test tasks are identified and assigned an owner and that the "who, what, where, and when" of testing have been clearly identified for all parties involved.
- Ensure that appropriate sign-offs have been obtained for the plan.
- Determine whether the test plan lists the sequence in which test steps should be performed.
- Ensure that test planning includes the identification of and plans for obtaining hardware and software needed for testing.
- Determine whether the test plan includes the identification of a location at which testing will take place (a conference room, for example) and that the location has been reserved for the appropriate period of time.
- If using a combination of vendor software and internally developed code, determine whether a process has been defined for ensuring that both parties' code will be merged in a well-coordinated fashion.

NOTE This list should not be used as a mechanical checklist. The absence of one of these items should not automatically result in an audit issue. Instead, look at the testing process as a whole, and determine whether enough of the key elements are present to provide reasonable assurance that adequate and controlled testing is occurring.

24. Ensure that all requirements can be mapped to a test case.

A defined process for tracing requirements to the test plan will provide assurance that all requirements are addressed and tested.

How

If a requirements trace map exists, review it and verify that all requirements are represented and mapped to a test case. If a trace map doesn't exist, review the process for ensuring that all requirements are tested.

25. Ensure that users are involved in testing and agree that the

system meets requirements. This should include IT personnel who will be supporting the system and IT personnel who were involved in performing initial technical feasibility studies for the project.

The system, software, or process is being developed to meet a specific business need. The project cannot be a success if the key stakeholders are not satisfied. Therefore, they must be involved in testing and must sign off on the system prior to implementation. Also, as mentioned in step 13, multiple organizations in the IT environment usually will be involved in supporting any new system, including network support, operating system support, database support, data center personnel, IT security, and the helpdesk. If these organizations are not involved in system testing and sign-off, they may not be prepared to support it and/or the system may not be in compliance with applicable standards and policies.

How

Look for evidence of user acceptance testing. Ensure that key stakeholders who were involved in requesting and approving the project and in defining system requirements (including affected IT organizations) are also involved in project testing and sign-off.

26. Consider participating in user acceptance testing and validating that system security and internal controls are functioning as intended.

This is necessary for the same reasons outlined in step 15. By participating in testing, you will be able to validate these controls independently.

How

During earlier steps, you should have worked with the project team to identify the internal controls that should be built into the system, software, or process. Review the test plan to ensure that it encompasses testing of those internal controls. Participate as an acceptance tester of those test cases.

Implementation

27. Ensure that an effective process exists for recording, tracking, escalating, and resolving problems that arise after implementation.

Unforeseen problems arise after the implementation of almost any new system. Without a robust method for capturing and resolving those issues, issues can "slip through the cracks" and not be resolved in a timely fashion. Also, an issue-tracking system is needed to ensure that issues are being prioritized and fixed according to their importance.

How

Review the issues database, issues spreadsheet, or whatever other method has been established for recording and tracking post-implementation issues. Ensure that the issue-tracking tool records adequate information regarding each issue, including description of the issue, priority level, due date, latest status, and resolution information. Ensure that controls exist over the tool used to track issues, such as backups and security to prevent unauthorized updates. Review processes for escalating issues and for ensuring that issues are tracked to resolution. Review the issues list for evidence that issues are being closed. Interview customers to ensure that the process is working.

28. Review and evaluate the project's conversion plan. Ensure that the project has an adequate conversion plan and follows this plan.

If the project being reviewed involves replacing an existing system, at some point, users will switch over to the new system. It is critical that existing data be converted successfully to the new system prior to this time to ensure a smooth transition.

How

Review the conversion plan, and look for elements such as the following:

- Ensure that all critical data is identified and considered for conversion.
- Review controls for ensuring that all data is converted completely and accurately. Examples of such control mechanisms could be control totals on key fields, record counts, and user reconciliation procedures.
- Determine whether all conversion programs are fully tested with user involvement and that the test results are documented.
- If historical data is not converted, ensure that a method is developed for accessing the data if needed. For example, if financial data is involved, historical financial data may be needed in the future for tax reporting.
- Review and evaluate plans for parallel processing or other fallback methods in case difficulties are experienced during transition to the new system.
- Ensure that the conversion process includes establishing data that was not used in the legacy systems. For example, a record in the new system may contain fields that were not contained in a similar record on the legacy systems. Consideration should be given to populating these new fields.
- Review and evaluate the plan for a "conversion weekend." A detailed plan should contain criteria and checkpoints for making "go/no-go" decisions.

NOTE This list should not be used as a mechanical checklist. The absence of one of these items should not automatically result in an audit issue. Instead, look at the conversion process as a whole, and determine whether enough of the key elements are present to provide a reasonable assurance that adequate and controlled conversion is taking place.

29. Review plans for transitioning the support of the new system or software from the project team to an operational support team.

After the project has been completed, it is likely that project personnel will be redeployed to other projects. It is therefore critical that run/maintain support personnel be trained properly in the functionality of the system so that they will be prepared to support it when users identify issues or request enhancements. This is one of the most commonly overlooked elements of projects.

How

Through interviews or review of documentation, look for evidence that support personnel have been identified, adequately involved in the project, and appropriately trained on the system and its functionality.

30. Ensure that sufficient documentation has been created for use of the system or process being developed and maintenance of the system or software. Evaluate processes for keeping the documentation up to date. Evaluate change controls and security over that documentation.

Incomplete or outdated technical and user documentation could increase costs and cycle time to maintain the software; increase support and training costs; and limit the system, process, or software's usefulness to the customer.

How

Obtain copies of existing documentation, and evaluate its adequacy. Look for evidence that would indicate that documentation has been updated when the system has changed, and review processes for ensuring ongoing maintenance of the documentation. Ensure that files containing documentation are locked down and can be modified only by appropriate personnel (using techniques described in Chapters 7 and 8). Interview appropriate personnel to understand processes for changing critical documents. Ensure that an approval process is required before changes are made to significant documents and that the approval process cannot be circumvented.

Training

31. Review plans for ensuring that all affected users are trained in the use of the new system, software, or process.

Training is an essential element for preparing end users on the functionality and nuances of a newly developed system. If training is not provided or is inadequate, the new system, software, or process likely will be misused, used ineffectively, or avoided.

How

Review the training plans and interview users to develop an opinion on its adequacy. Compare a list of planned training recipients with the population of end users to ensure that no significant gaps exist.

32. Ensure that processes are in place for keeping training materials up to date. Evaluate change controls and security over the training materials.

As new employees and new users need to use the system, they will want to take advantage of the training materials. If these training materials have become outdated (for example, because of system changes), the training materials' effectiveness will be limited.

How

Look for evidence that would indicate that training has been updated when the system has changed, and review processes for ensuring ongoing maintenance of the documentation. Ensure that files containing documentation are locked down and that they can be modified only by appropriate personnel (using techniques described in Chapters 7 and 8). Interview appropriate personnel to understand processes for changing critical documents. Ensure that an approval process is required before changes are made to significant documents and that the approval process cannot be circumvented.

Project Wrap-Up

33. Ensure that a process exists for closing out the project and recording lessons learned and that the process is followed.

Finalized project documentation and recorded lessons learned can be used to aid in the effectiveness and efficiency of future company projects. This step is often missed, as the project team quickly moves on to other tasks after successful implementation.

How

Review the project documentation, and ensure that all relevant documents have been finalized and baselined. Look for evidence that a final list of lessons learned from the project has been documented.

Knowledge Base

The Project Management Institute (PMI) is responsible for publishing the well-known Project Management Professional (PMP) certification. For more information about PMI or the PMP, visit www.pmi.org.

The Software Engineering Institute (SEI) and the Capability Maturity Model Integration (CMMI) Institute both have useful tools for gathering best practices for software development methodology. The SEI conducts research in software engineering, systems engineering, cybersecurity, and many other areas of computing. Their website (www.sei.cmu.edu) contains research and publications related to software engineering and information assurance, system verification and validation, and other topics pertinent to a strong software development life cycle. The CMMI is a set of best practices focused on helping organizations improve performance, capabilities, and business processes. For more information on CMMI, visit https://cmmiinstitute.com.

Finally, the frameworks described in Chapter 18 can be useful tools when you are put on the spot to develop questions and evaluate possible risks associated with a project.

Master Checklists

The following tables summarize the steps listed herein for auditing company projects.

Auditing Overall Project Management

Checklist for Auditing Overall Project Management

- ❑ 1. Ensure that sufficient project documentation and software development process documentation (if applicable) have been created. Ensure that the company's project methodology standards are being followed.
- ❑ 2. Review procedures for ensuring that project documentation is kept up to date.
- ❑ 3. Evaluate security and change management processes for critical project documentation.
- ❑ 4. Evaluate procedures for backing up critical project software and documentation. Ensure that backups are stored offsite and that documented procedures exist for recovery.
- ❑ 5. Ensure that an effective process exists for capturing project issues, escalating those issues as appropriate, and tracking them to resolution.
- ❑ 6. Ensure that an effective process exists for capturing project change requests, prioritizing them, and dispositioning them.
- ❑ 7. Verify that a project schedule has been created and that it contains sufficient detail based on the size of the project. Ensure that a process is in place for monitoring progress and reporting significant delays.
- ❑ 8. Ensure that a method is in place for tracking project costs and reporting overruns. Ensure that all project costs, including labor, are considered and tracked.
- ❑ 9. Evaluate the project leadership structure to ensure that both the business and IT are represented adequately. Ensure project sponsors have been clearly established and that they accept and understand their role.

Auditing Project Startup

Checklist for Auditing Project Startup

- ❑ 10. Ensure that appropriate project approval processes were followed prior to project initiation.
- ❑ 11. Ensure that a technical feasibility analysis has been performed along with, if applicable, a feasibility analysis by the company's legal department.
- ❑ 12. Review and evaluate the requirements document. Determine whether and how customer requirements for the project are obtained and documented before development takes place. Ensure that the customers sign off on the requirements and that the requirements encompass standard IT elements.
- ❑ 13. Evaluate the process for ensuring that all affected groups that will be helping to support the system, software, or process are involved in the project and will be part of the sign-off process, indicating their readiness to support it.
- ❑ 14. Review the process for establishing the priority of requirements.
- ❑ 15. Determine whether the system requirements and preliminary design ensure that appropriate internal control and security elements will be designed into the system, process, or software.
- ❑ 16. If the project involves the purchase of software, technology, or other external services, review and evaluate the vendor selection process and related contracts.

Auditing Detailed Design and System Development

Checklist for Auditing Detailed Design and System Development

- ❑ 17. Ensure that all requirements can be mapped to a design element.
- ❑ 18. Verify that the key stakeholders have signed off on the detailed design document or "use case" catalog.
- ❑ 19. Review processes for ensuring ongoing customer involvement with the prioritization of tasks on the project.
- ❑ 20. Look for evidence of peer reviews in design and development.
- ❑ 21. Verify that appropriate internal controls and security have been designed into the system.

Auditing Testing

Checklist for Auditing Testing

- ❑ 22. Verify that design and testing are occurring in a development/test environment and not in a production environment.
- ❑ 23. Review and evaluate the testing process. Ensure that the project has an adequate test plan and that it follows this test plan.
- ❑ 24. Ensure that all requirements can be mapped to a test case.
- ❑ 25. Ensure that users are involved in testing and agree that the system meets requirements. This should include IT personnel who will be supporting the system and IT personnel who were involved in performing initial technical feasibility studies for the project.
- ❑ 26. Consider participating in user acceptance testing and validating that system security and internal controls are functioning as intended.

Auditing Implementation

Checklist for Auditing Implementation

- ❑ 27. Ensure that an effective process exists for recording, tracking, escalating, and resolving problems that arise after implementation.
- ❑ 28. Review and evaluate the project's conversion plan. Ensure that the project has an adequate conversion plan and follows this plan.
- ❑ 29. Review plans for transitioning the support of the new system or software from the project team to an operational support team.
- ❑ 30. Ensure that sufficient documentation has been created for use of the system or process being developed and maintenance of the system or software. Evaluate processes for keeping the documentation up-to-date. Evaluate change controls and security over that documentation.

Auditing Training

Checklist for Auditing Training

- ❑ 31. Review plans for ensuring that all affected users are trained in the use of the new system, software, or process.
- ❑ 32. Ensure that processes are in place for keeping training materials up to date. Evaluate change controls and security over the training materials.

Auditing Project Wrap-Up

Checklist for Auditing Project Wrap-Up

- ❑ 33. Ensure that a process exists for closing out the project and recording lessons learned and that the process is followed.

CHAPTER 18

Auditing New/Other Technologies

In this chapter we will discuss basic steps you can use to assist you in performing audits of technologies that are not covered elsewhere in this book. The steps will be divided into the following areas:

- Initial steps
- Account management
- Permissions management
- Network security and controls
- Security monitoring and other general controls

Background

While we wish this book could cover every possible technology you might encounter during your audits, such a book would be too long to read, too heavy to lift, and too expensive to purchase. We have attempted to cover some of the more common technologies in detail; however, there are a plethora of technologies out there (with new ones being introduced continuously), and you will undoubtedly find the need to audit technology that isn't specifically covered in this book.

Fortunately, when you break it down, the same basic concepts apply no matter what you're auditing. Accounts need to be created and managed. You need to have some method for authenticating those accounts, and you need to manage what those accounts are authorized to do. Systems need to be configured securely and monitored. Methods for connecting to the technology need to be secured and managed. You get the picture. While the technical implementation of these controls will vary based on the specific technology you're auditing, the same basic concepts will generally apply. The intent of this chapter is to provide you with a framework into which you can plug the detailed nuances of the particular technology you're reviewing.

New/Other Technology Auditing Essentials

As you begin exploring new technologies, some frameworks and best practices can assist you in structuring your thoughts.

Generalized Frameworks

Generalized frameworks are useful in meetings when you've been put on the spot to come up with questions and possible risks associated with an application, technology, or project. You might even find yourself walking into a meeting, taking out a blank sheet of paper, and writing "PPTM," "STRIDE," and "PDIO" (as explained in the following sections) at the top before the meeting starts. Then, as you discuss the system or project under review, you can ask questions and take notes regarding how each element of each framework is being addressed. At the end of the meeting, if you find "blanks" by any of the framework elements, it's possible that you've discovered a gap in the controls. This sort of quick-and-dirty thought process should never take the place of detailed and thorough testing, of course, but it can be useful when you're participating in initial discussions and consulting on controls. And these same frameworks can be helpful as you develop detailed audit steps for new technologies or systems under review.

PPTM

People, Processes, Tools, and Measures (PPTM) is a great brainstorming framework for examining a system from the macro level. Detailed specific technical review steps dominate the chapters in this section of the book. PPTM helps you come up with your own steps quickly and efficiently as they apply to your unique situation.

People People in PPTM describes every aspect of the system that deals with a human. For example, if you have the opportunity to provide input during system development, ensure that the right people are involved in the planning, design, implementation, or operations for the project and that the right stakeholders are involved. If the system involves end users, ensure that it has controls around provisioning and deprovisioning access and that the end users have been involved in the components with which they will ultimately interface. Little is more embarrassing than spending time and money rolling out a system just to find out that upper management doesn't approve it or that the end users find that the interface is too complicated to use.

Processes Processes in PPTM describe every aspect of the system that is involved in a policy, procedure, method, or course of action. Review the interaction of the system with interfacing systems and verify compliance to security models (for example, ensure that firewalls are in place to protect the system from external systems, users, business partners, and the like). Procedures and policies should be written to support how the system is intended to be used. Adequate documentation also should exist to support technicians who need to maintain the system.

Tools Tools in PPTM describe every aspect of the system that deals with a concrete technology or product. Ensure that the appropriate hardware and environment exist to support the system and that the system interfaces with recommended technologies appropriate to your intended policies and procedures. Verify that the system and infrastructure are tested and audited appropriately.

Measures Measures in PPTM describe every aspect of the system that is quantifiable conceptually, such as the business purpose or application performance. For example, you can verify that the system meets well-documented and well-

thought-out acceptance criteria. If it's intended to solve a quantifiable business problem, verify that it does indeed solve that problem. Verify that logs are meaningful and that you can measure the performance of the system.

STRIDE

The STRIDE acronym stands for the following: *S*poofing identity, *T*ampering with data, *R*epudiation, *I*nformation disclosure, *D*enial of service, and *E*levation of privilege. STRIDE is a methodology used for identifying known threats. It is an example of a simplified threat-risk model that is easy to remember and apply. When assessing a system, you can use the acronym to develop steps that address how each of the following risks is mitigated.

Spoofing Identity Identity spoofing is a key risk for systems that have many users but provide a single execution context. In particular, users should not be able to become any other user or assume the attributes of another user.

Tampering with Data Data should be stored in a secure location, with access appropriately controlled. The system should carefully check data received from the user and validate that it is sane and applicable before storing or using it. For web and other applications with a client component, you should perform your validation checks on the server and not the client, where the validation checks might be tampered with. This is particularly important for web applications, where users can potentially change data delivered to them, return it, and thereby potentially manipulate client-side validation. The application should not send data to the user, such as interest rates or periods, that are obtainable only from within the application itself and allow the user potentially to manipulate that data.

Repudiation Users may dispute transactions if there is insufficient auditing or recordkeeping of their activity. For example, if a user says, "But I didn't transfer any money to this external account!" and you cannot track his or her activities through the application, it is extremely likely that the transaction will have to be written off as a loss. Therefore, you should consider whether the system requires nonrepudiation controls, such as access logs and audit trails at each tier. Preferably, the system should run with the user's privileges and nothing more.

Information Disclosure Users are rightfully wary of submitting private details to a system. If it is possible for an attacker to reveal data publicly, especially user data, whether anonymously or as an authorized user, there will be an immediate loss of confidence and a substantial period of reputation loss. Therefore, systems must include strong controls to prevent user ID tampering and abuse and to secure system data stored in databases and data files.

Denial of Service System designers should be aware that their systems may be subject to a denial-of-service attack. Therefore, the use of expensive resources such as large files, complex calculations, heavy-duty searches, or long queries should be reserved for authenticated and authorized users and should not be available to anonymous users.

For systems that don't have this luxury, every facet of the system should be engineered to perform as little work as possible, to use fast and few database queries, and to avoid exposing large files or unique links per user to prevent simple denial-of-service attacks.

Elevation of Privilege If a system provides distinct user and administrative roles, ensure that the user cannot elevate his or her role to a more highly privileged one. In particular, it is not sufficient simply to not display privileged-role links. Instead, all actions should be gated through an authorization matrix to ensure that only the permitted roles can access privileged functionality.

PDIO

PDIO comes from Cisco Systems and stands for *P*lanning, *D*esign, *I*mplementation, and *O*perations. Sometimes you need to consider the potential challenges at each stage of a project. You might find this framework useful as you look at a new system and think ahead to the upcoming challenges. A problem might occur, for example, if system administrators are tossing around ideas in a planning or design session for a network solution and the senior networking engineer isn't in the room. If you, as an auditor, are asked to look at the implementation of a new solution, you should immediately ask questions about the ongoing operations of the solution. Refer to Chapter 17 for more details on auditing company projects.

Best Practices

These best practices can help you quickly spot common weaknesses and poor controls.

Apply Defense-in-Depth

Layered approaches provide more security over the long term than one complicated mass of security architecture. You might, for example, use Access Control Lists (ACLs) on the networking and firewall equipment to allow only necessary traffic to reach the system. This approach significantly lowers the overall risk of compromise to the system, because you quickly eliminate access to services, ports, and protocols that otherwise would be accessible to compromise.

Use a Positive Security Model

Positive (whitelist) security models allow only what is on the list, excluding everything else by default. However, negative (blacklist) security models allow everything by default, eliminating only the items you know are bad. This is the challenge for antivirus programs, which you must update constantly to keep up with the number of new possible attacks (viruses) that could affect your system. The problem with this model, if you are forced to use it, is that you absolutely must keep the model updated. Even with the model updated, however, a vulnerability could exist that you don't know about, and your attack surface is much larger than if you used a positive security model. The preferred practice is to deny by default and allow only those things that you consciously permit.

Fail Safely

When a system fails, it can be dealt with in three ways: allow, block, or error. In general, system errors should fail in the same manner as a disallow (block) operation as viewed from the end user. This is important, because it means the end user doesn't have additional information to use that may help him or her compromise the system. Log what you want and keep any messages that you want elsewhere, but don't give the user additional information he or she might use to compromise your system.

Run with Least Privilege

The principle of least privilege mandates that accounts have the least amount of privilege possible to perform their activity. This encompasses user rights and resource permissions such as CPU limits, memory capacity, network bandwidth, and file system permissions.

Avoid Security by Obscurity

Obfuscating data, or hiding it instead of encrypting it, is a very weak security mechanism. If a human could figure out how to hide the data, what's to keep another person from learning how to recover the data? Consider, for example, how some people hide a key to their house under the doormat. A criminal wants the easiest possible way into the house and will check common places, such as under the doormat, the rock closest to the door, and above the door frame, for a key. Never obfuscate critical data that can be encrypted (or, better yet, never stored in the first place). This is not to discount the value in making critical data difficult for an attacker to locate. For example, if you have a system that stores passwords, it's best not to name the server housing the system "passwordserver," as that just makes it easy for an attacker to know they should target that system. It would be better to name the server something obscure and uninteresting. However, you should not rely on that as your only defense mechanism, and you should secure the server and its data under the assumption that an attacker will eventually find it.

Keep Security Simple

Simple security mechanisms are easy to verify and easy to implement correctly. Cryptographer Bruce Schneier is famous for suggesting that the quickest method to break a cryptographic algorithm is to go around it. Avoid overly complex security mechanisms, if possible. Developers should avoid the use of double negatives and complex architectures when a simple approach would be faster and easier. Don't confuse complexity with layers. Layers are good; complexity isn't.

Detect Intrusions and Keep Logs

Systems should have built-in logging that's protected and easily read. Logs help you troubleshoot issues and, just as important, help you track down when or how an application might have been compromised.

Never Trust External Infrastructure and Services

Many organizations use the processing capabilities of third-party partners that more than likely have differing security policies and postures than yours. It is unlikely that you can fully control any external third party, be they home users or major suppliers or partners. Therefore, implicitly trusting externally run systems is dangerous.

Establish Secure Defaults

Your systems should arrive to you or be presented to the users with the most secure default settings possible that still allow business to function. This may require training end users or communications, but the end result is a significantly reduced attack surface, especially when a system is pushed out across a large population.

Use Open Standards

Where possible, base security on open standards for increased portability and interoperability. Open standards are standards that are publicly available and are usually developed and maintained via a collaborative, open process. Since your infrastructure is likely a heterogeneous mix of platforms, the use of open standards helps ensure compatibility between systems as you continue to grow. Additionally, open standards are often well known and scrutinized by peers in the security industry to ensure that they remain secure.

Test Steps for Auditing New and Other Technologies

Initial Steps

1. Research the technology under review to learn the specifics of how it works and key control points. Add audit steps to the audit plan presented in this chapter as appropriate.

Every technology has its own unique risks and security features. While the steps outlined in this chapter will provide a foundation and structure for you to start with, they need to be supplemented with specifics for the technology you're auditing.

How

This is part of the fun of auditing—learning about new systems and technologies. Your best bets are going to be reading technical documentation and conducting interviews with the system administrator. Start by searching the Internet for potential resources and by asking the system administrator for recommended resources (and, in fact, he or she might have some system manuals to loan you). Spend some time reading those resources, focusing on two things: (1) the business objective the technology is helping to achieve and the basics of how the technology works and (2) the technology risks and key control points within the technology. Key control points could include how accounts are created, how permission is granted to resources, and configuration parameters that dictate things such as password settings and the security of network services. Once you start to formulate your ideas and questions, sit down with the system administrator to validate your understanding and get answers to your remaining questions.

Review the "Generalized Frameworks" and "Best Practices" sections from earlier in this chapter to spark additional ideas for test steps that are pertinent to the technology under review.

Once you have completed the previous activities, determine any test steps you want to add to the list already contained in this chapter.

2. Obtain basic system information (e.g., version number, latest service pack installed, overall architecture) for the technology under review.

This information will be used by the auditor to help interpret the results of subsequent audit steps.

How

Work with the system administrator to obtain this information.

Account Management

3. Review and evaluate procedures for creating accounts and ensuring that accounts are created only when there's a legitimate business need. Ensure that each account is associated with and can be traced easily to a specific employee. Also review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.

Effective controls should govern account creation and deletion. Inappropriate or lacking controls could result in unnecessary access to system resources, placing the integrity and availability of sensitive data and processes at risk.

If the owner of an account is not readily apparent, it will impede forensic investigations regarding inappropriate actions performed by that account. If multiple people use an account, no accountability can be established for actions performed by that account.

How

Interview the system administrator and review account creation procedures. This process should include some form of verification that the user has a legitimate need for access. Take a sample of accounts and review evidence that they were approved properly prior to being created. Alternatively, take a sample of accounts and validate their legitimacy by investigating and understanding the job function of the account owners.

Review the process for removing accounts when access is no longer needed. This process could include an automated process driven by the company's Human Resources (HR) Department providing information on terminations and job changes. Or the process could include a periodic review and validation of active accounts by the system administrator and/or other knowledgeable managers. Obtain a sample of accounts and verify that they are owned by active employees and that each employee has a legitimate business requirement for access.

Review the list of accounts and ensure it is easy to attribute each account to a single employee. Question any accounts that seem to be shared, such as guest or application accounts. If accounts such as these are required, determine how their use is controlled and accountability maintained.

4. Verify that appropriate password and authentication controls are in place. Also, determine whether default account passwords have been changed.

The appropriateness of the password and other authentication controls depends on the sensitivity of the data and processes managed within the system under review. Overly weak controls make the system subject to compromise, and overly strong passwords could place unnecessary overhead on usage of the system.

Many systems, particularly those that are purchased, have default accounts with well-known default passwords. Many of these default accounts are used for system administration and therefore have elevated privileges. If those default passwords are not changed, it is easy for an unauthorized user to access the application.

How

Determine where password settings are controlled within the technology being reviewed. With the help of the system administrator, review those settings and compare them against company policy. Consider controls such as password aging, length, complexity, history, timeout, and lockout policies.

Explore the need for stronger forms of authentication (e.g., two-factor authentication) through conversations with the system administrator and core users of the technology under review.

Determine whether default accounts and passwords exist with the help of the administrator and by review of system documentation and Internet research. If they do exist, one of the easiest ways to determine whether they have been changed is to attempt to log on using the default accounts and passwords (although you're likely better advised to ask the application administrator to attempt to do so).

5. Ensure that administrator access to the system is appropriately controlled.

An administrator user account and/or function should exist to help administer users, data, and processes within the system being reviewed. This account or functionality should be tightly controlled to prevent compromise and disruption of services to other users.

How

Determine how the technology's system administration function works via review of documentation and interviews with the system administrator. Obtain a list of all employees who have been granted the administrator access level and review each for appropriateness.

Permissions Management

6. Review the system's authorization mechanism to understand how users are granted access to sensitive resources (e.g., transactions and data). Review the permissions of sensitive resources, as well as processes for granting access to those resources.

Employees should be given only the amount of access to the system necessary to perform their jobs. If critical resources are not protected properly (i.e., unnecessary and excessive access is provided), it could result in inappropriate disclosure or alteration of sensitive data or system disruption.

How

Through review of technical documentation and interviews with the system administrator and core users of the technology, identify the resources (e.g., files, shares, transactions) on the system that are most critical. Determine whether these resources are appropriately secured (access is limited to only those who need access) and that appropriate processes are in place for granting and revoking access to those resources.

7. Evaluate the use of encryption.

The need for encryption is determined most often by policy, regulation, the sensitivity of the network, or the sensitivity of the data. Where possible, encryption techniques should be used for passwords and other confidential data that is sent across the network. This prevents other people on the network from "sniffing" and capturing this information. For sensitive data, such as passwords, encryption should also be considered when the data is at rest (in storage). This is particularly important for data that will be stored outside of your company's premises.

How

Review the system with the administrator to evaluate the existence of encryption where appropriate.

Network Security and Controls

8. Determine what network services are enabled on the system, and validate their necessity with the system administrator. For necessary services, review and evaluate procedures for assessing vulnerabilities associated with those services and keeping them patched.

Whenever remote access is allowed (that is, whenever a network service is enabled), it creates a new potential vector of attack, increasing the risk of unauthorized entry into the system. Therefore, network services should be enabled only when there is a legitimate business need for them.

New security holes are discovered and communicated frequently for most technical platforms. If the system administrator is not aware of these alerts, and if he or she does not install security patches, well-known security holes could exist on the system, providing a vector for system compromise.

How

Through review of technical documentation and interviews with the system administrator, determine what services are enabled. Once you have obtained a list of enabled services, talk through the list with the system administrator to understand the need for each service. For any services that are not needed, encourage the administrator to disable them.

Understand and evaluate the process used to keep abreast of new vulnerabilities for enabled services and to receive and apply patches for removing those vulnerabilities. Information on this process can be gathered via interviews and review of documentation.

Based on your research and interviews, you might determine that certain services should be configured in a specific way in order to be enabled securely. Where applicable, review the configuration of enabled services.

9. If possible, execute a network vulnerability scanning tool to check for current vulnerabilities in the environment.

This will provide a snapshot of the current security level of the system from a network services standpoint. The world of network vulnerabilities is an ever-changing one, and it is unrealistic to create a static audit program that will provide an up-to-date portrait of vulnerabilities that should be checked. Therefore, a scanning tool that is updated frequently is the most realistic mechanism for understanding the current security state of the machine. In addition, if the system administrator has a security-patching process in place, this scan will provide validation as to the effectiveness of that process (or as to whether it is really being executed).

How

Through your technical research and interviews with the system administrator, determine whether there is a network vulnerability scanning tool that will work for the technology under review. If so, coordinate execution of that tool with the administrator and review the results. Note: Even though many of these tools are designed to be nondisruptive and do not require access to the system, you should always inform the appropriate IT personnel (such as the system administrator, the network team, and information security) that you plan to run the tool, and then get their approval and schedule with them a time to execute the tool. Scanning tools can interact in an unexpected fashion with a system and cause a disruption, so it is important that others be aware of your activities. These tools should usually be run in a "safe" (nondisruptive) mode such that they do not attempt to exploit any vulnerabilities discovered. On rare occasions, you will want to run an actual exploit to get more accurate results, but this should be done only with buy-in from and coordination with the system owner and administrator.

Security Monitoring and Other General Controls

10. Ensure that the system has audit logs that are being captured per your organization's policies.

Audit logs provide evidence in the aftermath of an event and help with troubleshooting operational and security issues.

How

Through review of technical documentation and interviews with the administrator, determine the system's logging capabilities and review the enablement of those logs. Also, evaluate the security and retention of the audit logs.

11. Review and evaluate system administrator procedures for monitoring and maintaining the state of security on the system.

If the system administrator does not have processes for performing security monitoring, security holes could exist, and security incidents could occur without his or her knowledge.

How

Interview the system administrator and review any relevant documentation to get an understanding of security monitoring practices. This could include, for example, routine scans for and remediation of known vulnerabilities and/or alerts being sent and investigated when key activities are performed within the system. Some level of monitoring is important, but the monitoring level required should be consistent with the criticality of the system and the inherent risk of the environment.

If security monitoring is performed, assess the frequency of the monitoring and the quality with which it is performed. Look for evidence that the security

monitoring tools are actually used. Review recent results, and determine whether exceptions were investigated and resolved. Leverage the results of the rest of the audit in performing this assessment. For example, if you found significant issues in an area they were supposedly monitoring, it might lead to questions as to the effectiveness of that monitoring.

12. Verify that policies and procedures are in place to identify when a patch is available and to evaluate and apply applicable patches. Ensure that all approved patches are installed per your policy.

Most vendors have regularly scheduled patch releases for their products. If patches are not installed, widely known security vulnerabilities or critical performance issues could exist. Note that we discussed the patching of network services in the "Network Security and Controls" section earlier. This step is referring to any other patches that might be released in the core product itself.

How

Interview the administrator to determine who reviews advisories from vendors, what steps are taken to prepare for the patches, and how long the patches are tested before being applied to the production systems. Ask to review notes from the previous patching cycle. Compare the available patches with the patches applied in the system. Talk with the administrator about steps taken to mitigate potential risk if the patches are not applied in a timely manner.

13. Perform steps from Chapter 5 as they pertain to the system you are auditing.

In addition to auditing the logical security of the system, you should ensure that appropriate physical controls and operations are in place to provide for system protection and availability.

How

Reference the steps from Chapter 5, and perform those that are relevant to the system being audited. For example, the following topics are likely to be pertinent:

- Physical security
- Environmental controls
- Capacity planning
- Change management
- System monitoring
- Backup processes
- Disaster recovery planning

Master Checklists

The following tables summarize the steps listed herein for auditing new/other technologies.

Auditing Initial Steps

Checklist for Auditing Initial Steps	
❑	1. Research the technology under review to learn the specifics of how it works and key control points. Add audit steps to the audit plan presented in this chapter as appropriate.
❑	2. Obtain basic system information (e.g., version number, latest service pack installed, overall architecture) for the technology under review.

Auditing Account Management

Checklist for Auditing Account Management	
❑	3. Review and evaluate procedures for creating accounts and ensuring that accounts are created only when there's a legitimate business need. Ensure that each account is associated with and can be traced easily to a specific employee. Also review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.
❑	4. Verify that appropriate password and authentication controls are in place. Also, determine whether default account passwords have been changed.
❑	5. Ensure that administrator access to the system is appropriately controlled.

Auditing Permissions Management

Checklist for Auditing Permissions Management	
❑	6. Review the system's authorization mechanism to understand how users are granted access to sensitive resources (e.g., transactions and data). Review the permissions of sensitive resources, as well as processes for granting access to those resources.
❑	7. Evaluate the use of encryption.

Auditing Network Security and Controls

Checklist for Auditing Network Security and Controls	
❑	8. Determine what network services are enabled on the system, and validate their necessity with the system administrator. For necessary services, review and evaluate procedures for assessing vulnerabilities associated with those services and keeping them patched.
❑	9. If possible, execute a network vulnerability scanning tool to check for current vulnerabilities in the environment.

Auditing Security Monitoring and Other General Controls

Checklist for Auditing Security Monitoring and Other General Controls	
❑	10. Ensure that the system has audit logs that are being captured per your organization's policies.
❑	11. Review and evaluate system administrator procedures for monitoring and maintaining the state of security on the system.
❑	12. Verify that policies and procedures are in place to identify when a patch is available and to evaluate and apply applicable patches. Ensure that all approved patches are installed per your policy.
❑	13. Perform steps from Chapter 5 as they pertain to the system you are auditing.