Chapter 11

Auditing Computer-Based Information Systems

Learning Objectives

After studying this chapter, you should be able to:

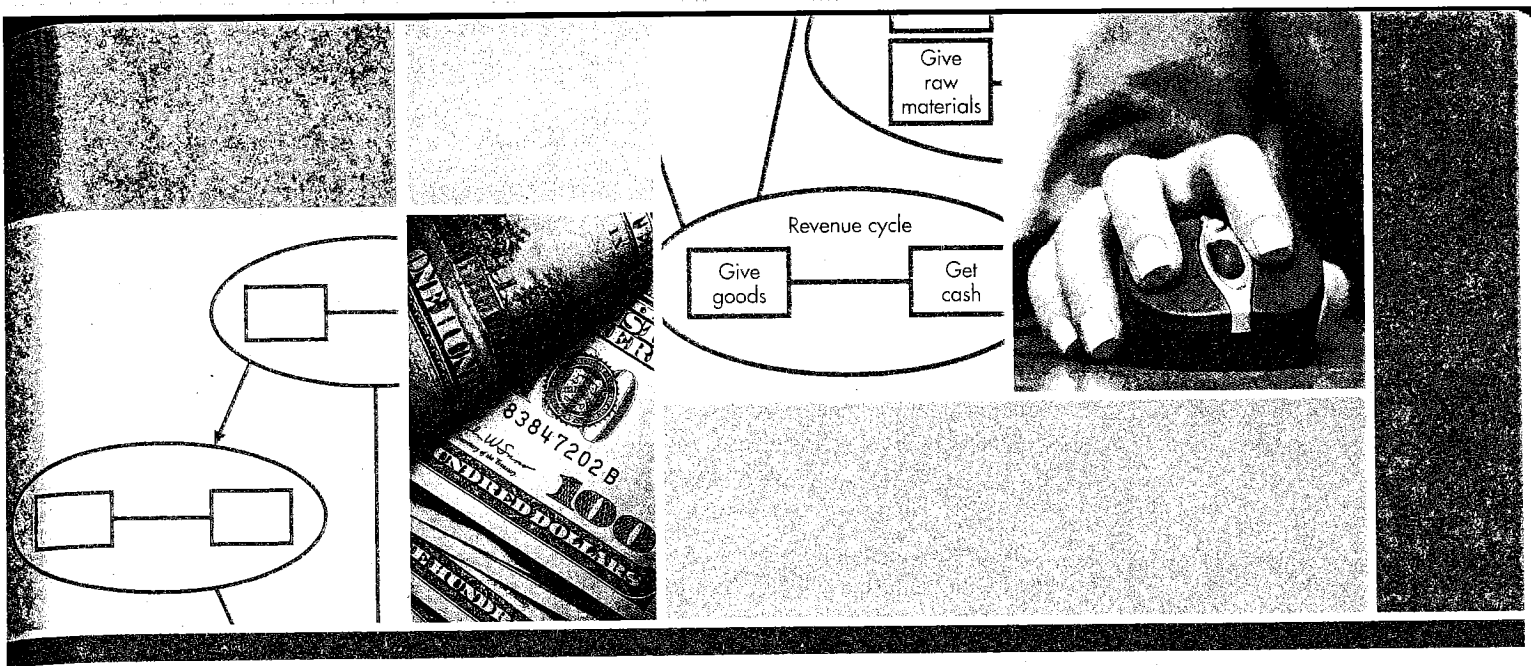
1. Describe the scope and objectives of audit work, and identify the major steps in the audit process.
2. Identify the objectives of an information system audit, and describe the four-step approach necessary for meeting these objectives.
3. Design a plan for the study and evaluation of internal control in an AIS.
4. Describe computer audit software, and explain how it is used in the audit of an AIS.
5. Describe the nature and scope of an operational audit.

INTEGRATIVE CASE SEATTLE PAPER PRODUCTS

Seattle Paper Products (SPP) is modifying its sales department payroll system to change the way it calculates sales commissions. Under the old system, commissions were a fixed percentage of dollar sales. The new system is considerably more complex, with commission rates varying according to the product sold and the total dollar volume of sales.

Jason Scott was assigned to use audit software to write a parallel simulation test program to calculate sales commissions and compare them with those generated by the new system. Jason obtained the necessary payroll system documentation and the details on the new sales commission policy and prepared his program.

Jason used the sales transaction data from the last payroll period to run his program. To his surprise, his calculations were $5,000 less than those produced by SPP's new program. Individual differences existed for about half of the company's salespeople. Jason double-checked his program code but could not locate any errors. He



selected a salesperson with a discrepancy and calculated the commission by hand. The result agreed with his program. He reviewed the new commission policy with the sales manager, line by line, and concluded that he understood the new policy completely. Jason is now convinced that his program is correct and that the error lies with SPP's new program. He is now asking himself the following questions:

1. How could a programming error of this significance be overlooked by experienced programmers who thoroughly reviewed and tested the new system?
2. Is this an inadvertent error, or could it be a fraud?
3. What can be done to find the error in the program?

Introduction

This chapter focuses on auditing an accounting information system (AIS). *Auditing* is the systematic process of obtaining and evaluating evidence regarding assertions about economic actions and events in order to determine how well they correspond with established criteria. The results of the audit are then communicated to interested users. Auditing requires careful planning and the collection, review, and documentation of audit evidence. In developing recommendations, the auditor uses established criteria, such as the principles of control described in previous chapters, as a basis for evaluation.

Many organizations in the United States employ internal auditors to evaluate company operations. Governments employ auditors to evaluate management performance and compliance with legislative intent. The Department of Defense employs auditors to review the financial records of companies with defense contracts. Publicly held companies hire external auditors to provide an independent review of their financial statements.

This chapter is written from the perspective of an internal auditor. *Internal auditing* is an independent, objective assurance and consulting activity designed to add value and improve organizational effectiveness and efficiency, including assisting in the design and implementation of an AIS. Internal auditing helps an organization accomplish its objectives by bringing a systematic, disciplined approach to evaluate and improve the effectiveness of risk management, control, and governance processes.

There are several different types of internal audits:

1. A *financial audit* examines the reliability and integrity of financial transactions, accounting records, and financial statements.

2. An *information systems*, or *internal control*, **audit** reviews the controls of an AIS to assess its compliance with internal control policies and procedures and its effectiveness in safeguarding assets. The audits usually evaluate system input and output; processing controls; backup and recovery plans; system security; and computer facilities.
3. An *operational* **audit** is concerned with the economical and efficient use of resources and the accomplishment of established goals and objectives.
4. A *compliance audit* determines whether entities are complying with applicable laws, regulations, policies, and procedures. These audits often result in recommendations to improve processes and controls used to ensure compliance with regulations.
5. An *investigative audit* examines incidents of possible fraud, misappropriation of assets, waste and abuse, or improper governmental activities.

In contrast, external auditors are responsible to corporate shareholders and are mostly concerned with gathering the evidence needed to express an opinion on the financial statements. They are only indirectly concerned with the effectiveness of a corporate AIS. However, external auditors are required to evaluate how audit strategy is affected by an organization's use of information technology (IT). External auditors may need specialized skills to (1) determine how the audit will be affected by IT, (2) assess and evaluate IT controls, and (3) design and perform both tests of IT controls and substantive tests.

Despite the distinction between internal and external auditing, many of the internal audit concepts and techniques discussed in this chapter also apply to external audits.

The first section of this chapter provides an overview of auditing and the steps in the auditing process. The second section describes a methodology and set of techniques for evaluating internal controls in an AIS and conducting an information system audit. The third section discusses the computer software and other techniques for evaluating the reliability and integrity of information in an AIS. Finally, operational audits of an AIS are reviewed.

The Nature of Auditing

Overview of the Audit Process

All audits follow a similar sequence of activities. Audits may be divided into four stages: planning, collecting evidence, evaluating evidence, and communicating audit results. Figure 11-1 is an overview of the auditing process and lists many of the procedures performed within each of these stages.

AUDIT PLANNING Audit planning determines why, how, when, and by whom the audit will be performed. The first step is to establish the audit's scope and objectives. For example, an audit of a publicly held corporation determines whether its financial statements are presented fairly. In contrast, an internal audit may examine a specific department or a computer application. It may focus on internal controls, financial information, operating performance, or some combination of the three.

An audit team with the necessary experience and expertise is formed. They become familiar with the auditee by conferring with supervisory and operating personnel, reviewing system documentation, and reviewing prior audit findings.

An audit is planned so the greatest amount of audit work focuses on the areas with the highest risk factors. There are three types of audit risk:

1. *Inherent risk* is the susceptibility to material risk in the absence of controls. For example, a system that employs online processing, networks, databases, telecommunications, and other forms of advanced technology has more inherent risk than a batch processing system.
2. *Control risk* is the risk that a material misstatement will get through the internal control structure and into the financial statements. A company with weak internal controls has a higher control risk than one with strong controls. Control risk can be determined by reviewing the control environment, testing internal controls, and considering control weaknesses identified in prior audits and evaluating how they have been rectified.

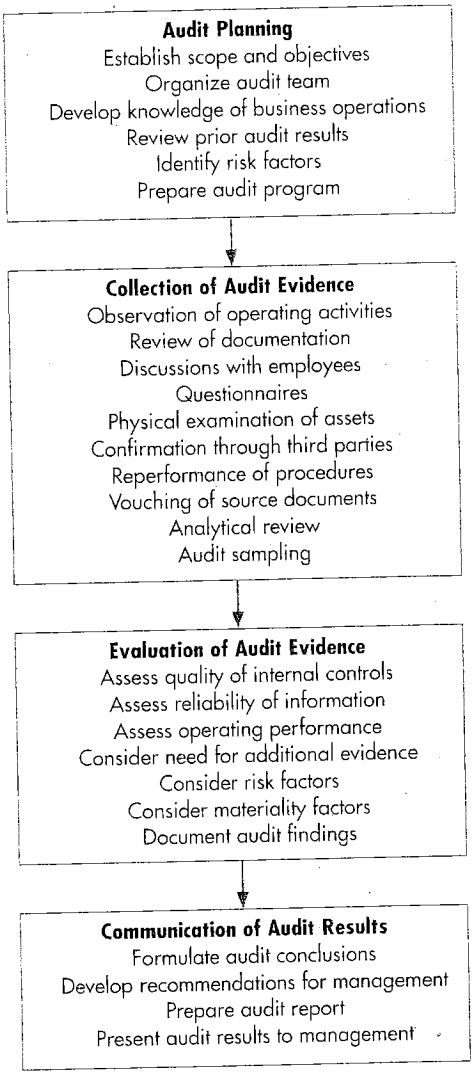


FIGURE 11-1
Overview of the Auditing Process

3. *Detection risk* is the risk that auditors and their audit procedures will fail to detect a material error or misstatement.

To conclude the planning stage, an audit program is prepared to show the nature, extent, and timing of the procedures needed to achieve audit objectives and minimize audit risks. A time budget is prepared, and staff members are assigned to perform specific audit steps.

COLLECTION OF AUDIT EVIDENCE Most audit effort is spent collecting evidence. Because many audit tests cannot be performed on all items under review, they are often performed on a sample basis. The following are the most common ways to collect audit evidence:

- *Observation* of the activities being audited (e.g., watching how data control personnel handle data processing work as it is received)
- *Review of documentation* to understand how a particular process or internal control system is supposed to function
- *Discussions* with employees about their jobs and about how they carry out certain procedures
- *Questionnaires* that gather data
- *Physical examination* of the quantity and/or condition of tangible assets, such as equipment and inventory
- *Confirmation* of the accuracy of information, such as customer account balances, through communication with independent third parties
- *Reperformance* of calculations to verify quantitative information (e.g., recalculating the annual depreciation expense)

- *Vouching* for the validity of a transaction by examining supporting documents, such as the purchase order, receiving report, and vendor invoice supporting an accounts payable transaction
- *Analytical review* of relationships and trends among information to detect items that should be further investigated. For example, an auditor for a chain store discovered that one store's ratio of accounts receivable to sales was too high. An investigation revealed that the manager was diverting collected funds to her personal use.

A typical audit has a mix of audit procedures. For example, an internal control audit makes greater use of observation, documentation review, employee interviews, and reperformance of control procedures. A financial audit focuses on physical examination, confirmation, vouching, analytical review, and reperformance of account balance calculations.

EVALUATION OF AUDIT EVIDENCE The auditor evaluates the evidence gathered and decides whether it supports a favorable or unfavorable conclusion. If inconclusive, the auditor performs sufficient additional procedures to reach a definitive conclusion.

Because errors exist in most systems, auditors focus on detecting and reporting those that significantly impact management's interpretation of the audit findings. Determining ***materiality***, what is and is not important in an audit, is a matter of professional judgment. Materiality is more important to external audits, where the emphasis is fairness of financial statement, than to internal audits, where the focus is on adherence to management policies.

The auditor seeks ***reasonable assurance*** that no material error exists in the information or process audited. Because it is prohibitively expensive to seek complete assurance, the auditor has some risk that the audit conclusion is incorrect. When inherent or control risk is high, the auditor must obtain greater assurance to offset the greater uncertainty and risks.

In all audit stages, findings and conclusions are documented in audit working papers. Documentation is especially important at the evaluation stage, when conclusions must be reached and supported.

COMMUNICATION OF AUDIT RESULTS The auditor submits a written report summarizing audit findings and recommendations to management, the audit committee, the board of directors, and other appropriate parties. Afterwards, auditors often do a follow-up study to ascertain whether recommendations were implemented.

The Risk-Based Audit Approach

The following internal control evaluation approach, called the risk-based audit approach, provides a framework for conducting information system audits:

1. *Determine the threats (fraud and errors) facing the company.* This is a list of the accidental or intentional abuse and damage to which the system is exposed.
2. *Identify the control procedures that prevent, detect, or correct the threats.* These are all the controls that management has put into place and that auditors should review and test, to minimize the threats.
3. *Evaluate control procedures.* Controls are evaluated two ways:
   a. A *systems review* determines whether control procedures are actually in place.
   b. *Tests of controls* are conducted to determine whether existing controls work as intended.
4. *Evaluate control weaknesses to determine their effect on the nature, timing, or extent of auditing procedures.* If the auditor determines that control risk is too high because the control system is inadequate, the auditor may have to gather more evidence, better evidence, or more timely evidence. Control weaknesses in one area may be acceptable if there are *compensating controls* in other areas.

The risk-based approach provides auditors with a clearer understanding of the fraud and errors that can occur and the related risks and exposures. It also helps them plan how to test and evaluate internal controls, as well as how to plan subsequent audit procedures. The result is a sound basis for developing recommendations to management on how the AIS control system should be improved.

Information Systems Audits

The purpose of an information systems audit is to review and evaluate the internal controls that protect the system. When performing an information systems audit, auditors should ascertain that the following six objectives are met:

1. Security provisions protect computer equipment, programs, communications, and data from unauthorized access, modification, or destruction.
2. Program development and acquisition are performed in accordance with management's general and specific authorization.
3. Program modifications have management's authorization and approval.
4. Processing of transactions, files, reports, and other computer records is accurate and complete.
5. Source data that are inaccurate or improperly authorized are identified and handled according to prescribed managerial policies.
6. Computer data files are accurate, complete, and confidential.

Figure 11-2 depicts the relationship among these six objectives and information systems components. Each of these objectives is discussed in detail in the following sections. Each description includes an audit plan to accomplish each objective, as well as the techniques and procedures to carry out the plan.

Objective 1: Overall Security

Table 11-1 uses the risk-based approach to present a framework for auditing overall computer security. It shows that overall system security threats include accidental or intentional damage to system assets; unauthorized access, disclosure, or modification of data and programs; theft; and interruption of crucial business activities.

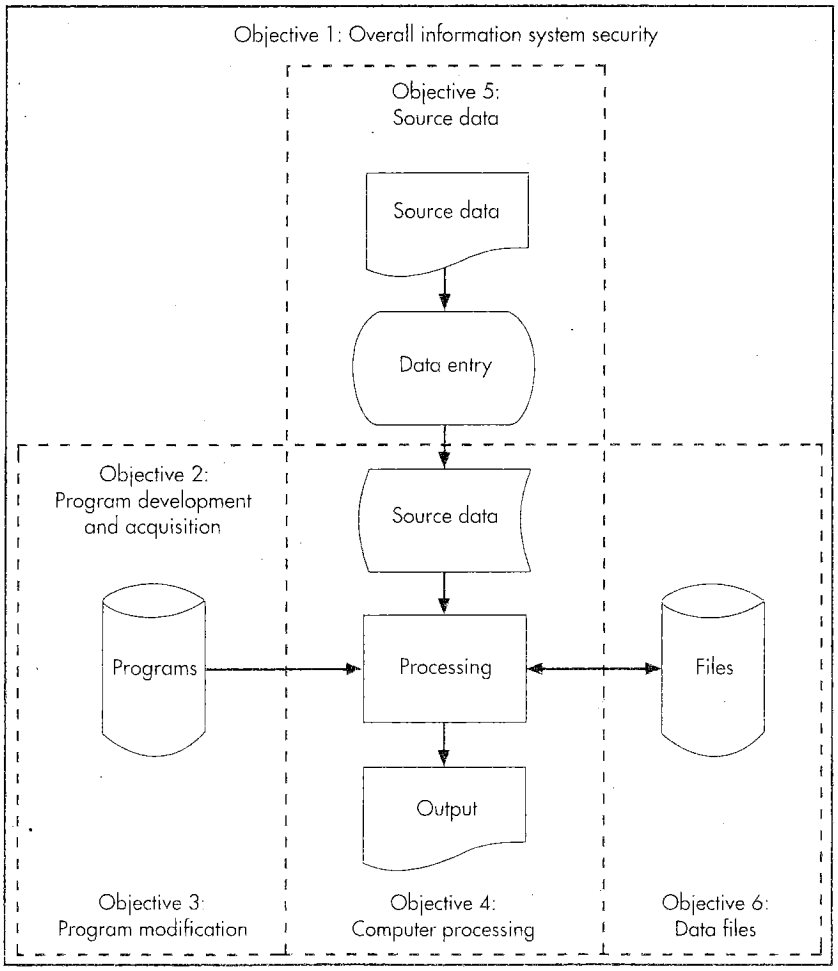


FIGURE 11-2
Information Systems Components and Related Audit Objectives

TABLE 11-1 Framework for Audit of Overall Computer Security

Types of Errors and Fraud

- Theft of or accidental or intentional damage to hardware
- Loss, theft, or unauthorized access to programs, data, and other system resources
- Loss, theft, or unauthorized disclosure of confidential data
- Unauthorized modification or use of programs and data files
- Interruption of crucial business activities

Control Procedures

- Information security/protection plan
- Limiting of physical access to computer equipment
- Limiting of logical access to system using authentication and authorization controls
- Data storage and transmission controls
- Virus protection procedures
- File backup and recovery procedures
- Fault-tolerant systems design
- Disaster recovery plan
- Preventive maintenance
- Firewalls
- Casualty and business interruption insurance

Audit Procedures: System Review

- Inspect computer sites
- Review the information security/protection and disaster recovery plans
- Interview information system personnel about security procedures
- Review physical and logical access policies and procedures
- Review file backup and recovery policies and procedures
- Review data storage and transmission policies and procedures
- Review procedures employed to minimize system downtime
- Review vendor maintenance contracts
- Examine system access logs
- Examine casualty and business interruption insurance policies

Audit Procedures: Tests of Controls

- Observe and test computer-site access procedures
- Observe the preparation of and off-site storage of backup files
- Test assignment and modification procedures for user IDs and passwords
- Investigate how unauthorized access attempts are dealt with
- Verify the extent and effectiveness of data encryption
- Verify the effective use of data transmission controls
- Verify the effective use of firewalls and virus protection procedures
- Verify the use of preventive maintenance and an uninterruptible power supply
- Verify amounts and limitations on insurance coverage
- Examine the results of disaster recovery plan test simulations

Compensating Controls

- Sound personnel policies, including segregation of incompatible duties
- Effective user controls

Control procedures to minimize these threats include developing an information security/protection plan, restricting physical and logical access, encrypting data, protecting against viruses, implementing firewalls, instituting data transmission controls, and preventing and recovering from system failures or disasters.

Systems review procedures include inspecting computer sites; interviewing personnel; reviewing policies and procedures; and examining access logs, insurance policies, and the disaster recovery plan.

Auditors test security controls by observing procedures, verifying that controls are in place and work as intended, investigating errors or problems to ensure they were handled correctly, and

examining any tests previously performed. For example, one way to test logical access controls is to try to break into a system. During a U.S. government security audit, auditors used agency terminals to gain unauthorized access to its computer system, disable its security-checking procedures, and control the system from the terminal. The security breakdown was possible because of poor administrative controls and inadequate security software.

Sound personnel policies and effective segregation of incompatible duties can partially compensate for poor computer security. Good user controls will also help, provided that user personnel can recognize unusual system output. Because it is unlikely these controls can compensate indefinitely for poor computer security, auditors should strongly recommend that security weaknesses be corrected.

Objective 2: Program Development and Acquisition

The auditor's role in systems development should be limited to an independent review of systems development activities. To maintain objectivity, auditors should not help develop the system.

Two things can go wrong in program development: (1) inadvertent programming errors due to misunderstanding system specifications or careless programming and (2) unauthorized instructions deliberately inserted into the programs.

These problems can be controlled by requiring management and user authorization and approval, thorough testing, and proper documentation.

During systems review, auditors should discuss development procedures with management, system users, and information system personnel. They should also review the policies, procedures, standards, and documentation listed in Table 11-2.

TABLE 11-2 Framework for Audit of Program Development

Types of Errors and Fraud

- Inadvertent programming errors or unauthorized program code

Control Procedures

- Review of software license agreements
- Management authorization for program development and software acquisition
- Management and user approval of programming specifications
- Thorough testing of new programs, including user acceptance tests
- Complete systems documentation, including approvals

Audit Procedures: System Review

- Independent review of the systems development process
- Review of systems development/acquisition policies and procedures
- Review of systems authorization and approval policies and procedures
- Review of programming evaluation standards
- Review of program and system documentation standards
- Review of test specifications, test data, and test results
- Review of test approval policies and procedures
- Review of acquisition of copyright license agreement policies and procedures
- Discussions with management, users, and information system personnel regarding development procedures

Audit Procedures: Tests of Controls

- Interview users about their systems acquisition/development and implementation involvement
- Review minutes of development team meetings for evidence of involvement
- Verify management and user sign-off approvals at development milestone points
- Review test specifications, test data, and systems test results
- Review software license agreements

Compensating Controls

- Strong processing controls
- Independent processing of test data by auditor

To test systems development controls, auditors should interview managers and system users, examine development approvals, and review development team meeting minutes. The auditor should review all documentation relating to the testing process to make sure all program changes were tested. The auditor should examine the test specifications and the test data and evaluate the test results. Auditors should ascertain how unexpected test result problems were resolved.

Strong processing controls may compensate for inadequate development controls if auditors obtain persuasive evidence of compliance with processing controls, using techniques such as independent test data processing. If this evidence is not obtained, auditors may have to conclude that a material internal control weakness exists and that the risk of significant threats in application programs is unacceptably high.

Objective 3: Program Modification

Table 11-3 presents a framework for auditing changes to application programs and system software. The same threats that occur during program development occur during program modification. For example, a programmer assigned to modify his company's payroll system inserted a command to erase all company files if he was terminated. When he was fired, the system crashed and erased key files.

TABLE 11-3 Framework for Audit of Program Modifications

Types of Errors and Fraud

- Inadvertent programming errors or unauthorized program code

Control Procedures

- List program components to be modified
- Management authorization and approval of program modifications
- User approval of program change specifications
- Thorough test of program changes, including user acceptance tests
- Complete program change documentation, including approvals
- Separate development, test, and production versions of programs
- Changes implemented by personnel independent of users and programmers
- Logical access controls

Audit Procedures: System Review

- Review program modification policies, standards, and procedures
- Review documentation standards for program modification
- Review final documentation of program modifications
- Review program modification testing and test approval procedures
- Review test specifications, test data, and test results
- Review test approval policies and procedures
- Review programming evaluation standards
- Discuss modification policies and procedures with management, users, and systems personnel
- Review logical access control policies and procedures

Audit Procedures: Tests of Controls

- Verify user and management signoff approval for program changes
- Verify that program components to be modified are identified and listed
- Verify that program change test procedures and documentation comply with standards
- Verify that logical access controls are in effect for program changes
- Observe program change implementation
- Verify that separate development, test, and production versions are maintained
- Verify that changes are not implemented by user or programming personnel
- Test for unauthorized or erroneous program changes using a source code comparison program, reprocessing, and parallel simulation

Compensating Controls

- Independent audit tests for unauthorized or erroneous program changes
- Strong processing controls

When a program change is submitted for approval, a list of all required updates should be compiled and approved by management and program users. All program changes should be tested and documented. During the change process, the developmental program must be kept separate from the production version. After the modified program is approved, the production version replaces the developmental version.

During systems review, auditors should discuss the change process with management and user personnel. The policies, procedures, and standards for approving, modifying, testing, and documenting the changes should be examined. All final documentation materials for program changes, including test procedures and results, should be reviewed. The procedures used to restrict logical access to the developmental program should be reviewed.

An important part of tests of controls is to verify that program changes were identified, listed, approved, tested, and documented. The auditor should verify that separate development and production programs are maintained and that changes are implemented by someone independent of the user and programming functions. The development program's access control table is reviewed to verify that only authorized users had access to the system.

Auditors should test programs on a surprise basis to guard against an employee inserting unauthorized program changes after the audit is completed and removing them prior to the next audit. There are three ways auditors test for unauthorized program changes:

1. After testing a new program, auditors keep a copy of its source code. Auditors use a ***source code comparison program*** to compare the current version of the program with the source code. If no changes were authorized, the two versions should be identical; any differences should be investigated. If the difference is an authorized change, auditors examine program change specifications to ensure that the changes were authorized and correctly incorporated.
2. In the ***reprocessing*** technique, auditors reprocess data using the source code and compare the output with the company's output. Discrepancies in the output are investigated.
3. In ***parallel simulation***, the auditor writes a program instead of using the source code, compares the outputs, and investigates any differences. Parallel simulation can be used to test a program during the implementation process. For example, Jason used this technique to test a portion of SPP's new sales department payroll system.

For each major program change, auditors observe testing and implementation, review authorizations and documents, and perform independent tests. If this step is skipped and program change controls subsequently prove to be inadequate, it may not be possible to rely on program outputs.

If program change controls are deficient, a compensating control is source code comparison, reprocessing, or parallel simulation performed by the auditor. Sound processing controls, independently tested by the auditor, can partially compensate for such deficiencies. However, if the deficiencies are caused by inadequate restrictions on program file access, the auditor should strongly recommend actions to strengthen the organization's logical access controls.

Objective 4: Computer Processing

Table 11-4 provides a framework for auditing the processing of transactions, files, and related computer records to update files and databases and to generate reports.

During computer processing, the system may fail to detect erroneous input, improperly correct input errors, process erroneous input, or improperly distribute or disclose output. Table 11-4 shows the control procedures to detect and prevent these threats and the systems review and tests of controls used to understand the controls, evaluate their adequacy, and test whether they function properly.

Auditors periodically reevaluate processing controls to ensure their continued reliability. If they are unsatisfactory, user and source data controls may be strong enough to compensate. If not, a material weakness exists, and steps should be taken to eliminate the control deficiencies.

Several specialized techniques are used to test processing controls, each of which has its own advantages and disadvantages. No technique is effective for all circumstances; all are more appropriate in some situations and less so in others. Auditors should not disclose which technique they use, because doing so may lessen their effectiveness. Each of these procedures is now explained.

TABLE 11-4 Framework for Audit of Computer Processing Controls

Types of Errors and Fraud

- Failure to detect incorrect, incomplete, or unauthorized input data
- Failure to properly correct errors flagged by data editing procedures
- Introduction of errors into files or databases during updating
- Improper distribution or disclosure of computer output
- Intentional or unintentional inaccuracies in reporting

Control Procedures

- Data editing routines
- Proper use of internal and external file labels
- Reconciliation of batch totals
- Effective error correction procedures
- Understandable operating documentation and run manuals
- Competent supervision of computer operations
- Effective handling of data input and output by data control personnel
- Preparation of file change listings and summaries for user department review
- Maintenance of proper environmental conditions in computer facility

Audit Procedures: System Review

- Review administrative documentation for processing control standards
- Review systems documentation for data editing and other processing controls
- Review operating documentation for completeness and clarity
- Review copies of error listings, batch total reports, and file change lists
- Observe computer operations and data control functions
- Discuss processing and output controls with operators and information system supervisors

Audit Procedures: Tests of Controls

- Evaluate adequacy of processing control standards and procedures
- Evaluate adequacy and completeness of data editing controls
- Verify adherence to processing control procedures by observing computer and data control operations
- Verify that application system output is properly distributed
- Reconcile a sample of batch totals; follow up on discrepancies
- Trace a sample of data edit routines errors to ensure proper handling
- Verify processing accuracy of sensitive transactions
- Verify processing accuracy of computer-generated transactions
- Search for erroneous or unauthorized code via analysis of program logic
- Check accuracy and completeness of processing controls using test data
- Monitor online processing systems using concurrent audit techniques
- Recreate selected reports to test for accuracy and completeness

Compensating Controls

- Strong user controls and effective controls of source data

PROCESSING TEST DATA One way to test a program is to process a hypothetical set of valid and invalid transactions. The program should process all valid transactions correctly and reject all invalid ones. All logic paths should be checked by one or more test transactions. Invalid data include records with missing data, fields containing unreasonably large amounts, invalid account numbers or processing codes, nonnumeric data in numeric fields, and records out of sequence.

The following resources are helpful when preparing test data:

- A list of actual transactions
- The test transactions the company used to test the program
- A *test data generator*, which prepares test data based on program specifications

In a batch processing system, the company's program and a copy of relevant files are used to process the test data. Results are compared with the predetermined correct output; discrepancies indicate processing errors or control deficiencies to be investigated.

In an online system, auditors enter test data and then observe and log the system's response. If the system accepts erroneous test transactions, the auditor reverses the effects of the transactions, investigates the problem, and recommends that the deficiency be corrected.

Processing test transactions has two disadvantages. First, the auditor must spend considerable time understanding the system and preparing the test transactions. Second, the auditor must ensure that test data do not affect company files and databases. The auditor can reverse the effects of the test transactions or process the transactions in a separate run using a copy of the file or database. However, a separate run removes some of the authenticity obtained from processing test data with regular transactions. Because the reversal procedures may reveal the existence and nature of the auditor's test to key personnel, it can be less effective than a concealed test.

CONCURRENT AUDIT TECHNIQUES Because transactions can be processed in an online system without leaving an audit trail, evidence gathered after data is processed is insufficient for audit purposes. In addition, because many online systems process transactions continuously, it is difficult to stop the system to perform audit tests. Thus, auditors use ***concurrent audit techniques*** to continually monitor the system and collect audit evidence while live data are processed during regular operating hours. Concurrent audit techniques use ***embedded audit modules***, which are program code segments that perform audit functions, report test results, and store the evidence collected for auditor review. Concurrent audit techniques are time-consuming and difficult to use but are less so if incorporated when programs are developed.

Auditors commonly use five concurrent audit techniques.

1. An ***integrated test facility (ITF)*** inserts fictitious records that represent a fictitious division, department, customer, or supplier in company master files. Processing test transactions to update them will not affect actual records. Because fictitious and actual records are processed together, company employees are unaware of the testing. The system distinguishes ITF records from actual records, collects information on the test transactions, and reports the results. The auditor compares processed data with expected results to verify that the system and its controls operate correctly. In a batch processing system, the ITF eliminates the need to reverse test transactions. ITF effectively tests online processing systems, because test transactions can be submitted frequently, processed with actual transactions, and traced through every processing stage without disrupting regular processing operations. The auditor must take care not to combine dummy and actual records during the reporting process.
2. In the ***snapshot technique***, selected transactions are marked with a special code. Audit modules record these transactions and their master file records before and after processing and store the data in a special file. The auditor reviews the data to verify that all processing steps were properly executed.
3. ***System control audit review file (SCARF)*** uses embedded audit modules to continuously monitor transaction activity, collect data on transactions with special audit significance, and store it in a SCARF file or ***audit log***. Transactions recorded include those exceeding a specified dollar limit, involving inactive accounts, deviating from company policy, or containing write-downs of asset values. Periodically, the auditor examines the audit log to identify and investigate questionable transactions.
4. ***Audit hooks*** are audit routines that notify auditors of questionable transactions, often as they occur. State Farm's use of audit hooks, including how the company detected a major fraud, is explained in Focus 11-1.
5. ***Continuous and intermittent simulation (CIS)*** embeds an audit module in a database management system (DBMS) that examines all transactions that update the database using criteria similar to those of SCARF. If a transaction has special audit significance, the CIS module independently processes the data (in a manner similar to parallel simulation), records the results, and compares them with those obtained by the DBMS. When discrepancies exist, they are stored in an audit log for subsequent investigation. If the discrepancies are serious, the CIS may prevent the DBMS from executing the update.

ANALYSIS OF PROGRAM LOGIC If auditors suspect that a program contains unauthorized code or serious errors, a detailed analysis of program logic may be necessary. This is time-consuming and requires proficiency in the appropriate programming language, so it should be used as a last

Using Audit Hooks at State Farm Life Insurance Company

The State Farm Life Insurance Company computer system has a host computer in Bloomington, Illinois, and smaller computers in regional offices. The system processes more than 30 million transactions per year for over 4 million individual policies worth more than $7 billion.

This online, real-time system updates files and databases as transactions occur. Paper audit trails have virtually vanished, and documents supporting changes to policyholder records have been eliminated or are held only a short time before disposition.

Because anyone with access and a working knowledge of the system could commit fraud, the internal audit staff was asked to identify all the ways fraud was possible. They brainstormed ways to defraud the system and interviewed system users, who provided extremely valuable insights.

Auditors implemented 33 embedded audit hooks to monitor 42 different types of transactions. One audit hook monitors unusual transactions in transfer accounts, which are clearing accounts for temporarily holding funds that are to be credited to multiple accounts.

The audit hooks have been very successful. One employee fraudulently processed a loan on her brother's life insurance policy, forged her brother's signature, and cashed the check. To conceal the fraud, she had to repay the loan before the annual status report was sent to her brother. She used a series of fictitious transactions involving a transfer account. The fraud was uncovered almost immediately when the transfer account audit hook recognized the first of these fictitious transactions and notified the auditor. Within a month of the notification, the case had been investigated and the employee terminated.

Source: Linda Marie Leinicke, W. Max Rexroad, and John D. Ward, "Computer Fraud Auditing: It Works," *Internal Auditor* (August 1990).

resort. Auditors analyze development, operating, and program documentation as well as a printout of the source code. They also use the following software packages:

- *Automated flowcharting programs* interpret source code and generate a program flowchart.
- *Automated decision table programs* interpret source code and generate a decision table.
- *Scanning routines* search a program for all occurrences of specified items.
- *Mapping programs* identify unexecuted program code. This software could have uncovered the program code that an unscrupulous programmer inserted to erase all computer files when he was terminated.
- *Program tracing* sequentially prints all program steps executed when a program runs, intermingled with regular output so the sequence of program execution events can be observed. Program tracing helps detect unauthorized program instructions, incorrect logic paths, and unexecuted program code.

Objective 5: Source Data

An *input controls matrix* is used to document the review of source data controls. The matrix in Figure 11-3 shows the control procedures applied to each input record field.

The data control function should be independent of other functions, maintain a data control log, handle errors, and ensure the overall efficiency of operations. It is usually not economically feasible for small businesses to have an independent data control function. To compensate, user department controls must be stronger with respect to data preparation, batch control totals, edit programs, restrictions on physical and logical access, and error-handling procedures. These procedures should be the focus of the auditor's systems review and tests of controls when there is no independent data control function.

Although source data controls may not change often, how strictly they are applied may change, and auditors should regularly test them. The auditor tests the system by evaluating source data samples for proper authorization, reconciling batch controls, and evaluating whether data edit errors were resolved and resubmitted for processing.

If source data controls are inadequate, user department and data processing controls may compensate. If not, auditors should recommend that source data control deficiencies be corrected.

Table 11-5 shows the internal controls that prevent, detect, and correct inaccurate or unauthorized source data. It also shows the system review and tests of control procedures auditors use. In an online system, the source data entry and processing functions are one operation. Therefore, source data controls are integrated with processing controls in Table 11-4.

Record Name: Employee Weekly Time Report / Field Names: / Input Controls	Employee number	Last name	Department number	Transaction code	Week ending (date)	Regular hours	Overtime hours		Comments
Financial totals									
Hash totals	✓					✓	✓		
Record counts									Yes
Cross-footing balance									No
Visual inspection									All fields
Check digit verification	✓								
Prenumbered forms									No
Turnaround document									No
Edit program									Yes
Sequence check	✓								
Field check	✓		✓			✓	✓		
Sign check									
Validity check	✓		✓	✓	✓				
Limit check						✓	✓		
Reasonableness test						✓	✓		
Completeness check				✓	✓	✓	✓		
Overflow procedure									
Other:									

FIGURE 11-3
Input Controls Matrix

Objective 6: Data Files

The sixth objective concerns the accuracy, integrity, and security of data stored on machine-readable files. Table 11-6 summarizes the errors, controls, and audit procedures for this objective. If file controls are seriously deficient, especially with respect to physical or logical access or to backup and recovery procedures, the auditor should recommend they be rectified.

TABLE 11-5 Framework for Audit of Source Data Controls

Types of Errors and Fraud

- Inaccurate or unauthorized source data

Control Procedures

- Effective handling of source data input by data control personnel
- User authorization of source data input
- Preparation and reconciliation of batch control totals
- Logging the receipt, movement, and disposition of source data input
- Check digit verification
- Key verification
- Use of turnaround documents
- Data editing routines
- User department review of file change listings and summaries
- Effective procedures for correcting and resubmitting erroneous data

Audit Procedures: System Review

- Review documentation about data control function responsibilities
- Review administrative documentation for source data control standards
- Review authorization methods and examine authorization signatures
- Review documentation to identify processing steps and source data content and controls
- Document source data controls using an input control matrix
- Discuss source data controls with data control personnel, system users, and managers

(Continued)

TABLE 11-5 Continued

Audit Procedures: Tests of Controls

- Observe and evaluate data control department operations and control procedures
- Verify proper maintenance and use of data control log
- Evaluate how error log items are dealt with
- Examine source data for proper authorization
- Reconcile batch totals and follow up on discrepancies
- Trace disposition of errors flagged by data edit routines

Compensating Controls

- Strong user and data processing controls

TABLE 11-6 Framework for Audit of Data File Controls

Types of Errors and Fraud

- Destruction of stored data due to errors, hardware or software malfunctions, and intentional acts of sabotage or vandalism
- Unauthorized modification or disclosure of stored data

Control Procedures

- Storage of data in a secure file library and restriction of physical access to data files
- Logical access controls and an access control matrix
- Proper use of file labels and write-protection mechanisms
- Concurrent update controls
- Data encryption for confidential data
- Virus protection software
- Off-site backup of all data files
- Checkpoint and rollback procedures to facilitate system recovery

Audit Procedures: System Review

- Review documentation for file library operation
- Review logical access policies and procedures
- Review standards for virus protection, off-site data storage, and system recovery procedures
- Review controls for concurrent updates, data encryption, file conversion, and reconciliation of master file totals with independent control totals
- Examine disaster recovery plan
- Discuss file control procedures with managers and operators

Audit Procedures: Tests of Controls

- Observe and evaluate file library operations
- Review records of password assignment and modification
- Observe and evaluate file-handling procedures by operations personnel
- Observe the preparation and off-site storage of backup files
- Verify the effective use of virus protection procedures
- Verify the use of concurrent update controls and data encryption
- Verify completeness, currency, and testing of disaster recovery plans
- Reconcile master file totals with separately maintained control totals
- Observe the procedures used to control file conversion

Compensating Controls

- Strong user and data processing controls
- Effective computer security controls

The auditing-by-objectives approach is a comprehensive, systematic, and effective means of evaluating internal controls. It can be implemented using an audit procedures checklist for each objective. The checklist helps auditors reach a separate conclusion for each objective and suggests compensating controls as appropriate. Each of the six checklists should be completed for each significant application.

Audit Software

Computer-assisted audit techniques (CAATS) refer to audit software, often called ***generalized audit software (GAS)***, that uses auditor-supplied specifications to generate a program that performs audit functions, thereby automating or simplifying the audit process. Two of the most popular software packages are Audit Control Language (ACL) and Interactive Data Extraction and Analysis (IDEA). CAATS is ideally suited for examining large data files to identify records needing further audit scrutiny.

The U.S. government discovered that computer-assisted audit techniques are a valuable tool in reducing massive federal budget deficits. The software is used to identify fraudulent Medicare claims and pinpoint excessive charges by defense contractors. The General Accounting Office (GAO) cross-checked figures with the IRS and discovered that thousands of veterans lied about their income to qualify for pension benefits. Some 116,000 veterans who received pensions based on need did not disclose $338 million in income from savings, dividends, or rents. More than 13,600 underreported income; one did not report income of over $300,000. When the Veterans Administration (VA) notified beneficiaries that their income would be verified with the IRS and the Social Security Administration, pension rolls dropped by more than 13,000, at a savings of $9 million a month. The VA plans to use the same system for checking income levels of those applying for medical care. If their income is found to be above a certain level, patients will be required to make copayments.

In another example, a new tax collector in a small New England town requested a tax audit. Using CAATS, the auditor accessed tax collection records for the previous four years, sorted them by date, summed collections by month, and created a report of monthly tax collections. The analysis revealed that collections during January and July, the two busiest months, had declined by 58% and 72%, respectively. Auditors then used CAATS to compare each tax collection record with property records. They identified several discrepancies, including one committed by the former tax collector, who used another taxpayer's payment to cover her own delinquent tax bills. The former tax collector was arrested for embezzlement.

To use CAATS, auditors decide on audit objectives, learn about the files and databases to be audited, design the audit reports, and determine how to produce them. This information is recorded on specification sheets and entered into the system. The CAATS program uses the specifications to produce an auditing program. The program uses a copy of the company's live data (to avoid introducing any errors) to perform the auditing procedures and produce the specified audit reports. CAATS cannot replace the auditor's judgment or free the auditor from other phases of the audit. For example, the auditor must still investigate items on exception reports, verify file totals against other sources of information, and examine and evaluate audit samples.

CAATS are especially valuable for companies with complex processes, distributed operations, high transaction volumes, or a wide variety of applications and systems.

The following are some of the more important uses of CAATS:

- Querying data files to retrieve records meeting specified criteria
- Creating, updating, comparing, downloading, and merging files
- Summarizing, sorting, and filtering data
- Accessing data in different formats and converting the data into a common format
- Examining records for quality, completeness, consistency, and correctness
- Stratifying records, selecting and analyzing statistical samples
- Testing for specific risks and identifying how to control for that risk
- Performing calculations, statistical analyses, and other mathematical operations
- Performing analytical tests, such as ratio and trend analysis, looking for unexpected or unexplained data patterns that may indicate fraud

- Identifying financial leakage, policy noncompliance, and data processing errors
- Reconciling physical counts to computed amounts, testing clerical accuracy of extensions and balances, testing for duplicate items
- Formatting and printing reports and documents
- Creating electronic work papers

Operational Audits of an AIS

The techniques and procedures used in operational audits are similar to audits of information systems and financial statements. The basic difference is audit scope. An information systems audit is confined to internal controls and a financial audit to systems output, whereas an operational audit encompasses all aspects of systems management. In addition, objectives of an operational audit include evaluating effectiveness, efficiency, and goal achievement.

The first step in an operational audit is audit planning, during which the scope and objectives of the audit are established, a preliminary system review is performed, and a tentative audit program is prepared. The next step, evidence collection, includes the following activities:

- Reviewing operating policies and documentation
- Confirming procedures with management and operating personnel
- Observing operating functions and activities
- Examining financial and operating plans and reports
- Testing the accuracy of operating information
- Testing controls

At the evidence evaluation stage, the auditor measures the system against one that follows the best systems management principles. One important consideration is that the results of management policies and practices are more significant than the policies and practices themselves. That is, if good results are achieved through policies and practices that are theoretically deficient, then the auditor must carefully consider whether recommended improvements would substantially improve results. Auditors document their findings and conclusions and communicate them to management.

The ideal operational auditor has audit training and experience as well as a few years' experience in a managerial position. Auditors with strong auditing backgrounds but weak management experience often lack the perspective necessary to understand the management process.

Summary and Case Conclusion

Jason is trying to determine how his parallel simulation program generated sales commission figures that were higher than those generated by SPP's program. Believing that this discrepancy meant there was a systematic error, he asked to review a copy of SPP's program.

The program was lengthy, so Jason used a scanning routine to search the code for occurrences of "40000," because that was the point at which the commission rate changes, according to the new policy. He discovered a commission rate of 0.085 for sales in excess of $40,000, whereas the policy called for only 0.075. Some quick calculations confirmed that this error caused the differences between the two programs.

Jason's audit manager met with the embarrassed development team, who acknowledged and corrected the coding error.

The audit manager called Jason to congratulate him. He informed Jason that the undetected programming error would have cost over $100,000 per year in excess sales commissions. Jason was grateful for the manager's praise and took the opportunity to point out deficiencies in the development team's programming practices. First, the commission rate table was embedded in the program code; good programming practice requires that it be stored in a separate table to be used by the program when needed. Second, the incident called into question the quality of SPP's program development and testing practices. Jason asked whether a more extensive operational audit of those practices was appropriate. The audit manager agreed it was worth examining and promised to raise the issue at his next meeting with Northwest's director of internal auditing.

a QuickTime video containing malicious software that replaced the links in the user's page with links to a phishing site. The devastating Conficker worm infected 25% of enterprise Window PCs.

Many viruses and worms exploit known software vulnerabilities than can be corrected with a software patch. Therefore, a good defense against them is making sure that all software patches are installed as soon as they are available.

Recent viruses and worms have attacked cell phones and personal electronic devices using text messages, Internet page downloads, and Bluetooth wireless technology. Flaws in Bluetooth applications open the system to attack. ***Bluesnarfing*** is stealing (snarfing) contact lists, images, and other data using Bluetooth. A reporter for TimesOnline accompanied Adam Laurie, a security expert, around London scanning for Bluetooth-compatible phones. Before a Bluetooth connection can be made, the person contacted must agree to accept the link. However, Laurie has written software to bypass this control and identified vulnerable handsets at an average rate of one per minute. He downloaded entire phonebooks, calendars, diary contents, and stored pictures. Phones up to 90 meters away were vulnerable.

Bluebugging is taking control of someone else's phone to make or listen to calls, send or read text messages, connect to the Internet, forward the victim's calls, and call numbers that charge fees. These attacks will become more popular as phones are used to pay for items purchased. When a hacker wants something, all he has to do is bluebug a nearby phone and make a purchase. To prevent these attacks, a bluetooth device can be set to make it hard for other devices to recognize it. Antivirus software for phones is being developed to deal with such problems.

In the future, many other devices—such as home security systems, home appliances, automobiles, and elevators—will be connected to the Internet and will be the target of viruses and worms.

Table 6-1 summarizes, in alphabetical order, the computer fraud and abuse techniques discussed in the chapter.

TABLE 6-1 Computer Fraud and Abuse Techniques

Technique	Description
Address Resolution Protocol (ARP) spoofing	Sending fake ARP messages to an Ethernet LAN. ARP is a computer networking protocol for determining a network host's hardware address when only its IP or network address is known.
Adware	Software that collects and forwards data to advertising companies or causes banner ads to pop up as the Internet is surfed.
Bluebugging	Taking control of a phone to make calls, send text messages, listen to calls, or read text messages.
Bluesnarfing	Stealing contact lists, images, and other data using Bluetooth.
Botnet, bot herders	A network of hijacked computers. Bot herders use the hijacked computers, called zombies, in a variety of Internet attacks.
Buffer overflow attack	Inputting so much data that the input buffer overflows. The overflow contains code that takes control of the computer.
Caller ID spoofing	Displaying an incorrect number on the recipient's caller ID display to hide the identity of the caller.
Carding	Verifying credit card validity; buying and selling stolen credit cards.
Chipping	Planting a chip that records transaction data in a legitimate credit card reader.
Cross-site scripting (XSS) attack	Exploits Web page security vulnerabilities to bypass browser security mechanisms and create a malicious link that injects unwanted code into a Web site.
Cyber-bullying	Using computer technology to harm another person.
Cyber-extortion	Requiring a company to pay money to keep an extortionist from harming a computer or a person.
Data diddling	Changing data before, during, or after it is entered into the system.
Data leakage	Unauthorized copying of company data.
Denial-of-service attack	An attack designed to make computer resources unavailable to its users. For example, so many e-mail messages that the Internet service provider's e-mail server is overloaded and shuts down.
Dictionary attack	Using software to guess company addresses, send employees blank e-mails, and add unreturned messages to spammer e-mail lists.
DNS spoofing	Sniffing the ID of a Domain Name System (server that converts a Web site name to an IP address) request and replying before the real DNS server.
Eavesdropping	Listening to private voice or data transmissions.

(*Continued*)

TABLE 6-1 Continued

Technique	Description
Economic espionage	The theft of information, trade secrets, and intellectual property.
E-mail threats	Sending a threatening message asking recipients to do something that makes it possible to defraud them.
E-mail spoofing	Making a sender address and other parts of an e-mail header appear as though the e-mail originated from a different source.
Evil twin	A wireless network with the same name as another wireless access point. Users unknowingly connect to the evil twin; hackers monitor the traffic looking for useful information.
Hacking	Unauthorized access, modification, or use of computer systems, usually by means of a PC and a communications network.
Hijacking	Gaining control of someone else's computer for illicit activities.
IP address spoofing	Creating Internet Protocol packets with a forged IP address to hide the sender's identity or to impersonate another computer system.
Identity theft	Assuming someone's identity by illegally obtaining confidential information such as a Social Security number.
Internet auction fraud	Using an Internet auction site to commit fraud.
Internet misinformation	Using the Internet to spread false or misleading information.
Internet terrorism	Using the Internet to disrupt communications and ecommerce.
Internet pump-and-dump fraud	Using the Internet to pump up the price of a stock and then sell it.
Key logger	Using spyware to record a user's keystrokes.
Lebanese looping	Inserting a sleeve into an ATM so that it will not eject the victim's card, pretending to help the victim as a means to discover his or her PIN, and then using the card and PIN to drain the account.
Logic bombs and time bombs	Software that sits idle until a specified circumstance or time triggers it, destroying programs, data, or both.
Malware	Software that can be used to do harm.
Man-in-the-middle (MITM) attack	A hacker placing himself between a client and a host to intercept network traffic; also called *session hijacking*.
Masquerading/impersonation	Accessing a system by pretending to be an authorized user. The impersonator enjoys the same privileges as the legitimate user.
Packet sniffing	Inspecting information packets as they travel the Internet and other networks.
Password cracking	Penetrating system defenses, stealing passwords, and decrypting them to access system programs, files, and data.
Pharming	Redirecting traffic to a spoofed Web site to obtain confidential information.
Phishing	Communications that request recipients to disclose confidential information by responding to an e-mail or visiting a Web site.
Phreaking	Attacking phone systems to get free phone access; using phone lines to transmit viruses and to access, steal, and destroy data.
Piggybacking	1. Clandestine use of someone's Wi-Fi network. 2. Tapping into a communications line and entering a system by latching onto a legitimate user. 3. Bypassing physical security controls by entering a secure door when an authorized person opens it.
Podslurping	Using a small device with storage capacity (iPod, Flash drive) to download unauthorized data from a computer.
Posing	Creating a seemingly legitimate business, collecting personal data while making a sale, and never delivering items sold.
Pretexting	Acting under false pretenses to gain confidential information.
Rootkit	Software that conceals processes, files, network connections, and system data from the operating system and other programs.
Round-down fraud	Truncating interest calculations at two decimal places and placing truncated amounts in the perpetrator's account.
Ransomware	Software that encrypts programs and data until a ransom is paid to remove it.
Salami technique	Stealing tiny slices of money over time.
Scareware	Malicious software of no benefit that is sold using scare tactics.

TABLE 6-1 Continued

Technique	Description
Scavenging/dumpster diving	Searching for confidential information by searching for documents and records in garbage cans, communal trash bins, and city dumps
Sexting	Exchanging explicit text messages and pictures.
Shoulder surfing	Watching or listening to people enter or disclose confidential data.
Skimming	Double-swiping a credit card or covertly swiping it in a card reader that records the data for later use.
SMS spoofing	Using short message service (SMS) to change the name or number a text message appears to come from.
Social engineering	Techniques that trick a person into disclosing confidential information.
Software piracy	Unauthorized copying or distribution of copyrighted software.
Spamming	E-mailing an unsolicited message to many people at the same time.
Splog	A spam blog that promotes Web sites to increase their Google PageRank (how often a Web page is referenced by other pages).
Spyware	Software that monitors computing habits and sends that data to someone else, often without the user's permission.
Spoofing	Making electronic communications look like someone else sent it.
SQL injection attack	Inserting a malicious SQL query in input in such a way that it is passed to and executed by an application program.
Steganography	Hiding data from one file inside a host file, such as a large image or sound file.
Superzapping	Using special software to bypass system controls and perform illegal acts.
Tabnapping	Secretly changing an already open browser tab using JavaScript.
Trap door	A back door into a system that bypasses normal system controls.
Trojan horse	Unauthorized code in an authorized and properly functioning program.
Typosquatting/URL hijacking	Web sites with names similar to real Web sites; users making typographical errors are sent to a site filled with malware.
Virus	Executable code that attaches itself to software, replicates itself, and spreads to other systems or files. Triggered by a predefined event, it damages system resources or displays messages.
Vishing	Voice phishing, in which e-mail recipients are asked to call a phone number that asks them to divulge confidential data.
War dialing	Dialing phone lines to find idle modems to use to enter a system, capture the attached computer, and gain access to its network(s).
War driving/rocketing	Looking for unprotected wireless networks using a car or a rocket.
Web cramming	Developing a free and worthless trial-version Web site and charging the subscriber's phone bill for months even if the subscriber cancels.
Web-page spoofing	Also called *phishing*.
Worm	Similar to a virus; a program rather than a code segment hidden in a host program. Actively transmits itself to other systems. It usually does not live long but is quite destructive while alive.
Zero-day attack	Attack between the time a software vulnerability is discovered and a patch to fix the problem is released.

Summary and Case Conclusion

It took RPC two days to get its system back up to the point that the audit team could continue their work. RPC had been hit with multiple problems at the same time. Hackers had used packet sniffers and eavesdropping to intercept a public key RPC had sent to Northwest. That led to a man-in-the-middle attack, which allowed the hacker to intercept all communications about the pending merger. It also opened the door to other attacks on both systems.

Law enforcement was called into investigate the problem, and they were following up on three possibilities. The first was that hackers had used the intercepted information to purchase stock in both companies, leak news of the purchase to others via Internet chat rooms, and, once

4. *Avoids potential for disagreement.* Both parties possess the same expectations, and pertinent information is captured in writing.

RFPs for exact hardware and software specifications have lower total costs and require less time to prepare and evaluate, but they do not permit the vendor to recommend alternative technology. Requesting a system that meets specific performance objectives and requirements leaves technical issues to the vendor but is harder to evaluate and often results in more costly bids.

The more information a company provides vendors, the better their chances of receiving a system that meets its requirements. Vendors need detailed specifications, including required applications, inputs and outputs, files and databases, frequency and methods of file updating and inquiry, and unique requirements. It is essential to distinguish mandatory requirements from desirable features.

Evaluating Proposals and Selecting a System

Proposals that lack important information, fail to meet minimum requirements, or are ambiguous are eliminated. Proposals passing this preliminary screening are compared with system requirements to determine whether all mandatory requirements are met and how many desirable requirements are met. Top vendors are invited to demonstrate their system using company-supplied data to measure system performance and validate vendor's claims. Table 21-1 presents hardware, software, and vendor evaluation criteria.

TABLE 21-1 Hardware, Software, and Vendor Evaluation Criteria

Hardware evaluation	Are hardware costs reasonable, based on capabilities and features? Are processing speed and capabilities adequate for the intended use? Are secondary storage capabilities adequate? Are the input and output speeds and capabilities adequate? Is the system expandable? Is the hardware based on old technology that will soon to be out-of-date? Is the hardware available now? If not, when? Is the hardware compatible with existing hardware, software, and peripherals? How do performance evaluations compare with competitors? What are the availability and cost of support and maintenance? What warranties come with the system? Is financing available (if applicable)?
Software evaluation	Does the software meet all mandatory specifications? How well does the software meet desirable specifications? Will program modifications be required to meet company needs? Does the software have adequate control capabilities? Is the performance (speed, accuracy, reliability) adequate? How many companies use the software? Are they satisfied? Is documentation adequate? Is the software compatible with existing software? Was the software demonstration/test drive adequate? Does the software have an adequate warranty? Is the software flexible, easily maintained, and user-friendly? Is online inquiry of files and records possible? Will the vendor keep the package up to date?
Vendor evaluation	How long has the vendor been in business? Is the vendor financially stable and secure? How experienced is the vendor with the hardware and software? Does the vendor stand behind its products? How good is its warranty? Does the vendor regularly update its products? Does the vendor provide financing? Will the vendor put promises in a contract? Will the vendor supply a list of customer references? Does the vendor have a reputation for reliability and dependability? Does the vendor provide timely support and maintenance? Does the vendor provide implementation and installation support? Does the vendor have high-quality, responsive, and experienced personnel? Does the vendor provide training?