National Institute of
Standards and Technology
U.S. Department of Commerce

Special Publication 800-115
(Draft)

Technical Guide to Information Security Testing (Draft)

Recommendations of the National Institute of Standards and Technology

Murugiah Souppaya
Karen Scarfone
Amanda Cody
Angela Orebaugh

NIST Special Publication 800-115 (Draft)

Technical Guide to Information Security Testing (Draft)

Recommendations of the National Institute of Standards and Technology

Murugiah Souppaya
Karen Scarfone
Amanda Cody
Angela Orebaugh

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

November 2007

U.S. Department of Commerce

Carlos M. Gutierrez, Secretary

National Institute of Standards and Technology

James Turner, Acting Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology (IT). ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-115 (Draft)
Natl. Inst. Stand. Technol. Spec. Publ. 800-115, 82 pages (Nov. 2007)

Acknowledgements

The authors, Murugiah Souppaya and Karen Scarfone of the National Institute of Standards and Technology (NIST) and Amanda Cody and Angela Orebaugh of Booz Allen Hamilton, wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content. The authors would like to acknowledge Tim Grance, Arnold Johnson, Matt Scholl, and Pat Toth of NIST and Steve Allison, Derrick Dicoi, Victoria Thompson, Selena Tonti, Theodore Winograd, and Gregg Zepp of Booz Allen Hamilton for their keen and insightful assistance throughout the development of the document. Additional acknowledgements will be added to the final version of the publication.

Table of Contents

List of Appendices

List of Figures

List of Tables

Executive Summary

Information security testing is the process of validating the effective implementation of security controls for information systems and networks, based on the organization's security requirements. Information security testing augments an organization's information security program and is useful for the following:

- Evaluating the effectiveness of implemented security measures or controls
- Identifying, validating, and assessing security weaknesses so that they can be addressed
- Increasing an organization's ability to maintain a proactive computer network defense.

This document is a technical guide for information security testing. It presents the methods and techniques an organization might use to conduct testing. The guide offers guidance to a security test team in executing technical testing techniques and understanding their impact on systems and networks. In addition, for information security testing to be successful and have a positive impact on the security posture of a system, and ultimately the entire organization, there are elements beyond the execution of testing that support the technical testing process. Suggestions for these activities, including a robust planning process, root cause analysis, and tailored reporting, are also presented in this guide.

The processes and technical guidance presented in this document enable organizations to do the following:

- Develop information security testing policy, methodology, and individual roles and responsibilities
- Accurately plan for an information security test by providing guidance on determining the systems to test and the approach for testing, addressing logistical considerations, developing the rules of engagement (ROE), and ensuring legal and policy considerations are addressed
- Safely and effectively execute an information security test using the presented methods and techniques and respond to incidents during the testing
- Appropriately handle data (collection, storage, transmission, and destruction) throughout the information security testing process
- Conduct analysis and reporting to translate findings into risk mitigation actions to improve the organization's security posture.

To accomplish these activities and ensure that technical security testing provides maximum value, NIST recommends the following:

- **Establish an information security testing policy**. Policies identify the organization's requirements for executing information security testing. This provides accountability to the appropriate individuals to ensure testing is conducted in accordance with the stated requirements. Topics that a testing policy should address include the organizational requirements with which security tests must comply, roles and responsibilities, frequency of assessments, and documentation requirements.
- **Implement a repeatable methodology designed to reduce the risk associated with information security testing**. A repeatable methodology enables organizations to maximize the value of information security testing while minimizing the risk. Risks range from not gathering sufficient information on the security posture of the organization because of fear of impacting system functionality, to affecting the availability of a system or network by executing techniques without the proper safeguards in place. Processes to reduce risk include using skilled testers, developing

comprehensive ROE, and logging the activities of the test team. Organizations need to determine what level of risk they are willing to accept for each security test and tailor their approach accordingly while maintaining compliance with applicable federal regulations.

- **Authorize only skilled, experienced, and objective individuals to conduct testing**. Security testing must be performed by capable and trained staffed. Often, individuals recruited for security testing are already involved in system administration. It is important that sufficient separation of duties between system administrators and the security test team be maintained. Security testers should have significant skills and experience with the testing method and techniques identified for the engagement.
- **Determine the goals and objectives of each security test and tailor the approach accordingly**. Security tests have specific objectives, acceptable level of risk, and available resources. No individual technique provides a comprehensive picture of an organization's security when executed by itself. For the best picture of the security posture, organizations should use a combination of techniques.
- **Analyze findings and develop risk mitigation techniques to address weaknesses**. To ensure security testing provides the ultimate value, organizations should conduct root cause analysis upon completion of testing, to enable the translation of findings into actionable mitigation techniques. The results may indicate that organizations should address not only technical weaknesses but also weaknesses in organizational processes and procedures.

1. Introduction

1.1 Authority

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets; but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b (3), "Securing Agency Information Systems," as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This guideline has been prepared for use by federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority; nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official.

1.2 Purpose and Scope

The purpose of this document is to assist organizations in planning and conducting technical information security testing, analyzing findings, and developing mitigation strategies. The guide provides practical recommendations for designing, implementing, and maintaining information security testing processes and procedures. The guide is not intended to present a comprehensive information security testing program but rather an overview of key elements of security testing, with an emphasis on specific technical security testing techniques, the benefits and limitations of each, and recommendations for their use.

Throughout this document, the term *system* is used. System may refer to an individual information system, defined as a discrete set of information resources organized for the collection, processing, maintenance, use, sharing, dissemination, or disposition of information, or to a group of information systems such as a network. A distinction will be made in specific sections as needed to ensure clarity.

1.3 Audience

This guide is for computer security staff and program managers, system and network administrators, and other technical staff who are responsible for the technical aspects of preparing, operating, and securing systems and network infrastructures. Managers can also use the information presented in the guide to facilitate the decision-making processes associated with security testing. The material in this document is technically oriented, and it is assumed that readers have at least a basic understanding of system and network security.

1.4 Document Structure

The remainder of this document is organized into seven major sections:

- Section 2 presents an overview of information security testing, including information security testing policies, roles and responsibilities, methodologies, and techniques.
- Section 3 provides a detailed description of several review security testing techniques, such as documentation review, log review, network sniffing, and file integrity checking.
- Section 4 describes several techniques for identifying targets and analyzing them for potential vulnerabilities. Examples of these techniques are network discovery and vulnerability scanning.
- Section 5 explains techniques commonly used to validate the existence of vulnerabilities, such as password cracking and penetration testing.
- Section 6 presents an approach and process for planning a security test.
- Section 7 discusses those factors that are key to the execution of security testing, including coordination, testing, analysis, and data handling.
- Section 8 presents an approach for reporting test findings and provides an overview of remediation activities.

This guide also contains five appendices. Appendix A describes two live operating system (OS) CD distributions, which allow the user to boot a computer to a CD that contains a fully operational OS and testing tools. Appendix B contains a template for creating security testing rules of engagement (ROE). Appendix C offers a list of resources that may facilitate the security testing process. Appendix D provides a glossary of terms, and Appendix E provides a list of acronyms and abbreviations used in this document.

2. Information Security Testing Overview

Information security testing is the process of validating the effective implementation of security controls for information systems and networks, based on the organization's security requirements. Technical information security testing can identify, validate, and assess technical vulnerabilities, which helps organizations to understand and improve the security posture of their systems and networks. Security testing is required by FISMA[1] and other regulations.

This section presents suggestions for developing an information security testing policy, including typical roles and responsibilities of individuals involved in the security testing process. It also provides an overview of information security testing methodologies and testing techniques.

2.1 Information Security Testing Policy

Organizations should develop an information security testing policy to provide the organization with direction and guidance on security testing. This policy identifies the requirements for security tests and holds accountable those individuals responsible for ensuring that information security tests comply with organizational requirements. The policy should address the following:

- Organizational requirements with which security tests must comply
- Appropriate roles and responsibilities
- Adherence to established methodology
- Frequency of assessments
- Documentation requirements, such as rules of engagement (ROE), test plans, and test results.

Once developed and approved by the senior officials responsible for security testing policy, the policy should be disseminated to the appropriate staff, which may include the offices of the chief information officer (CIO), chief information security officer (CISO), and chief technology officer (CTO). If third-party test teams conduct any of the organization's security testing, leadership should communicate the policy to the third parties.

It is recommended that organizations review the testing policy at least annually and when there are new requirements related to testing. This review will determine the continued applicability of the policy, address any necessary modifications, and provide an opportunity for incorporating lessons learned into the policy.

A testing policy should define roles and responsibilities for security testing to ensure all stakeholders are aware of their responsibilities. At a minimum, organizations should identify roles and responsibilities for those approving security testing and those executing security testing. The description of the roles and responsibilities presented illustrates the individual roles as they relate to information security testing. Organizations may use titles that differ from those presented below.

[1] Section 3544 requires the "periodic testing and evaluation of the effectiveness of information security policies, procedures, and practices, to be performed with a frequency depending on risk, but no less than annually." FISMA is available at http://csrc.nist.gov/drivers/documents/FISMA-final.pdf.

2.1.1 Senior IT Management and the CIO

The senior IT management and the CIO ensure that the organization's security posture is adequate and provide direction and advisory services for the protection of the organization's systems. The senior IT management and the CIO are responsible for the following activities associated with security testing:

- Coordinating the development and maintenance of the organization's information security policies, standards, and procedures
- Ensuring the establishment of, and compliance with, consistent security evaluation processes throughout the organization
- Participating in developing processes for decision-making and prioritization of systems for security testing
- Approving the ROE and test plan (see Section 6.4), which govern the test activities, and the plan of action and milestones (POA&M) (see Section 8.3), which provides the mitigation actions.

2.1.2 CISO and Information Systems Security Program Manager

The CISO and the information systems security program manager (ISSPM)[2] oversee adherence to and compliance with the organization's security policy. The CISO or ISSPM is responsible for the following activities associated with security testing:

- Developing and implementing standard operating procedures and security policy
- Complying with security policies, standards, and requirements set forth by higher authority
- Ensuring that systems are identified and scheduled for periodic testing in accordance with the security policy requirements of each respective system and their priority levels
- Allocating resources to conduct security testing
- Approving the ROE, test plan, and resulting corrective action plan
- Ensuring the mitigation strategy is implemented by providing necessary resources and coordinating with the information systems security officer (ISSO) and system managers.

2.1.3 Information Systems Security Officer

The ISSO is responsible for overseeing all aspects of information security in a specific organizational entity. The ISSO ensures that the organization's information security practices comply with organizational and departmental policies, standards, and procedures. ISSOs are responsible for the following activities associated with security testing:

- Developing security standards and procedures for their area of responsibility
- Maintaining operational integrity of systems by conducting tests and ensuring that designated IT security professionals are conducting scheduled testing on authorized systems
- Developing the ROE or test plan with the test team

[2] Some organizations call this role the Senior Agency Information Security Officer (SAISO).

- Ensuring the documentation and analysis of security test results and the development of a mitigation strategy for identified vulnerabilities.

2.1.4 System and Network Administrators

System and network administrators address the security requirements of the specific systems and networks for which they are responsible on a daily basis. System and network administrators are responsible for the following activities associated with security testing:

- Monitoring system integrity, protection levels, and security related events
- Resolving detected security anomalies associated with their information system resources
- Conducting security tests as required
- Assessing and verifying the implemented security measures.

2.1.5 Program Managers and System Owners

Program managers and system owners oversee the compliance of their assets with their stated security requirements. Because they thoroughly understand the criticality and sensitivity of their systems and data, they can assist the organization in identifying the impact a security incident would have on specific systems. Program managers and system owners should work with the test team and security managers, such as the CISO, ISSPM, and ISSO, to do the following:

- Ensure system criticality and potential impact are communicated during the security testing planning phase
- Identify major system changes that may affect security so the appropriate decisions on retesting systems can be made
- Ensure the adoption of mitigation recommendations and the justification for rejected recommendations.

2.1.6 Internal Security Test Team

The internal security test team, also known as the assessment team, conducts information security tests for the organization. Only designated individuals should conduct security testing. Depending on the organization structure, size, location, and available resources, the test team may be divided by geographical location, or the team may be centralized and deployed to various sites to conduct security testing. Organizations may structure their test teams on the basis of security testing competencies (e.g., functional area such as wireless security testing). Alternatively, teams may be organized to be able to address each functional area at varying levels of depth. For instance, the team may have members capable of reviewing a system configuration, others who can use automated vulnerability assessment tools to identify known vulnerabilities, and others who can actively exploit vulnerabilities to demonstrate ineffective security measures. The assessment team's responsibilities include the following:

- Informing the appropriate parties—such as security officers, management, and users—of security test activities
- Developing the ROE and test plans with the system managers, the ISSO, and the ISSPM
- Executing testing and collecting all relevant data

- Analyzing collected data and developing mitigation recommendations
- Conducting additional testing when needed to validate the mitigation actions.

2.1.7 External Entities

Having external entities, such as auditors, third-party testers, and contractor support staff, conduct information security tests for the organization offers an independent view and approach to the testing, which an internal test team may not be able to provide. Organizations also may use external entities to provide specific subject matter expertise not available internally. While it can be beneficial to have an external perspective on the security posture, external entities introduce additional risk. Because the organization being assessed is giving an external group access to its systems, external entities should be properly vetted by the organization to ensure that they possess the necessary skills, experience, and integrity. Some of the risk associated with the security testing may be assumed by external entities, in that they may be responsible for damages incurred by the organization being assessed. External entities should understand and comply with the organization's applicable policies and operational and security requirements.

External entities are responsible for all activities identified in Section 2.1.6. In addition, they should do the following:

- Coordinate and communicate with the organization being assessed
- Ensure proper authority is granted and maintain a signed copy of the ROE, ensuring all updates to it are documented
- Sign and abide by any required nondisclosure agreements
- Properly protect data in accordance with the organization's regulations, including handling, transmission, storage, and deletion of all collected data and the resulting reports.

2.1.8 Users

During security testing, users should cooperate with the testers and comply with valid requests of the test team, such as providing documentation (e.g., security policy or process or architecture diagrams) or allowing the system to be physically accessed by the test team using a test team account. Users should be made aware of how to verify the validity of requests made by people claiming to be testers. Users of a system are also responsible for reporting any anomalous or suspicious activity (e.g., increase in computer processing usage on their system or individuals without appropriate identification), which may or may not be a result of the testing.

2.2 Information Security Testing Methodology

A repeatable and documented security testing methodology is beneficial, in that it can do the following:

- Provide consistency and structure to security testing, which can minimize the risks of testing
- Expedite the transition of new testing staff, enabling continuous functionality of the testing team
- Address resource constraints associated with security testing.

Information security testing requires resources such as time, staff, hardware, and software. The availability of resources is often a limiting factor in the type and frequency of security testing. Evaluating the types of security tests the organization will execute, developing an appropriate methodology,

identifying the resources required, and structuring the testing process to support the expected requirements can mitigate the resource challenge. This gives the organization the ability to reuse pre-established resources, such as trained staff and standardized testing platforms; decreases the time required to conduct the assessment and the need to purchase testing equipment and software; and reduces the overall cost of the assessment.

A phased information security testing methodology offers a number of advantages. The structure is easy to follow and provides natural breaking points for staff transition. The testing methodology should contain at least the following phases:

- **Planning**. The planning phase is critical to a successful security test. The planning phase is used to gather information necessary to execute the test—such as system data, threat data, and required security controls—and to develop the approach to testing the system. A security test should be treated as any other project; it is recommended that a project management plan be developed to address goals and objectives, scope, requirements, team roles and responsibilities, limitations, success factors, assumptions, resources, timeline, and deliverables. Development of the ROE and test plan, which detail how the testing will be executed, occurs during this phase. The test plan and ROE sometimes contain sufficient information so that a separate project management plan is not required. At a minimum, development of the ROE should occur for every testing engagement. Section 6 of this guide covers planning.

- **Execution**. The primary goals for the execution phase are to identify vulnerabilities and, when appropriate, validate those vulnerabilities. This phase should address the activities associated with the intended assessment method and technique. While the specific activities for this phase differ for each assessment type, upon completion of this phase the assessors will have identified system, network, and organizational process vulnerabilities. Section 7 of this document covers the execution phase.

- **Post-Execution**. The post-execution phase focuses on analyzing identified vulnerabilities to determine root causes, establish mitigation recommendations, and develop a final report. Section 8 of this guide addresses reporting and mitigation.

There are several accepted methodologies for conducting information security tests. NIST does not endorse one methodology over another; the intent is to provide options to organizations so they can make an informed decision to adopt an existing methodology or take several others to develop a unique methodology that best suits the organization. Appendix C contains references to several methodologies. One of these methodologies was created by NIST and is documented in Special Publication (SP) 800-53A, *Guide for Assessing the Security Controls in Federal Information Systems (Draft)*, which offers suggestions for assessing the effectiveness of security controls recommended in NIST SP 800-53.[3] The guide discusses the framework for the development of assessment procedures, describes the process of assessing security controls, and offers assessment procedures for each control. NIST SP 800-53A was developed to be used in conjunction with NIST SP 800-37, *Guide for the Security Certification and Accreditation of Federal Information Systems*.[4]

2.3 Information Security Testing Techniques

Security testing techniques are the actual tests that are performed to assess the security posture of systems and networks. There are dozens of techniques that can be used for security testing. For the purposes of

[3] NIST SP 800-53 and NIST SP 800-53A (Draft) are available at http://csrc.nist.gov/publications/nistpubs/.
[4] NIST SP 800-37 is available at http://csrc.nist.gov/publications/nistpubs/.

this document, commonly used techniques have been selected for further discussion in later sections. These techniques have been grouped into the following three categories:

- **Review Techniques**. These are passive techniques to evaluate systems, applications, networks, policies, and procedures to discover vulnerabilities. Review techniques, which are generally conducted manually, include documentation review, log review, ruleset review, system configuration review, network sniffing, and file integrity checking. Section 3 provides additional information on review techniques.
- **Target Identification and Analysis Techniques**. These techniques, mostly active, can identify systems, ports, services, and potential vulnerabilities. These techniques may be performed manually but are generally performed using automated tools. Target identification and analysis techniques include network discovery, network port and service identification, vulnerability scanning, wireless scanning, and application security testing. Further discussion of these techniques is presented in Section 4.
- **Target Vulnerability Validation Techniques**. These are active techniques that corroborate the existence of vulnerabilities. These techniques may be performed manually or by using automatic tools, depending on the testing technique and skill of the test team. Target vulnerability validation techniques include password cracking, remote access testing, penetration testing, social engineering, and physical security testing. For additional information on these techniques, see Section 5.

No single technique can provide a complete picture of a system or network's security. Organizations should use techniques in combination to perform robust security tests. For example, penetration testing usually relies on performing network port and service identification and vulnerability scanning to identify vulnerable hosts and services that may be viable targets for later penetration. When planning which techniques should be used for a particular testing need, an organization should consider several factors, including how much the testing can impact the organization's systems and networks, what perspective the testers should have (outside or inside the organization's network perimeter), how much knowledge of the targeted systems and networks the testers should have, and who in the organization's staff (e.g., IT operations staff, incident response staff) should be aware of the testing. These factors are discussed below.

2.3.1 Passive and Active

Passive security testing primarily involves reviews of documents such as policies, procedures, security plans, security requirements, standard operating procedures, architecture diagrams, engineering documentation, system configurations, rulesets, and system logs. While passive testing does not provide a comprehensive picture of an organization's security posture, it does provide important insight into fundamental aspects of the security program and implementation.

The purpose of conducting passive testing is to determine whether a system is properly documented and to gain an insight into security aspects of the system that are only available through documentation. Documentation identifies the intended design, installation, configuration, operation, and maintenance of the systems and network. Review and cross-referencing of this documentation ensures conformance and consistency. For example, an environment's security requirements should drive documentation such as system security plans and standard operating procedures; the test team should ensure that all plans, procedures, architectures, and configurations are compliant with the stated security requirements and applicable policies. Another example is reviewing a firewall's ruleset to ensure compliance with an organization's security policies regarding Internet usage, such as the use of instant messaging, peer-to-peer (P2P) file sharing, and other prohibited activities.

Passive testing typically has no impact on the actual systems or networks in the target environment aside from accessing the necessary documentation, logs, or rulesets.[5] While the impact is negligible, if system configuration files or logs are going to be retrieved from a given system such as a router or firewall, only trained individuals, such as system administrators, should do so to ensure settings are not inadvertently modified or deleted.

Active security testing involves hands-on testing of systems and networks to identify their security vulnerabilities. Active security testing can be executed across the entire enterprise or on select systems. Active security testing using scanning and penetration techniques provides valuable information on potential vulnerabilities and the likelihood that an adversary or intruder can exploit those vulnerabilities. It also allows organizations to measure levels of compliance in areas such as patch management, antivirus management, password policy, and configuration management.

Active security testing can provide a more accurate picture of an organization's security posture than passive security testing, but active security testing is more intrusive than passive and can impact systems or networks in the target environment. The level of potential impact depends on the types of active testing techniques used. Active techniques interact with the target systems and networks in various ways, such as sending normal network packets to determine open and closed ports or sending specially crafted packets to test for vulnerabilities. Any time a test or tester directly interacts with a system or network, there is a potential to cause unexpected system halts and other denial of service conditions. Organizations should consider the acceptable level of intrusiveness when determining which technique to use. Excluding tests known to create denial of service conditions and other disruptions should reduce the level of impact.

Active security testing has several limitations, including the following:

- Active security testing does not provide a comprehensive evaluation of the security posture of an organization. One reason is that many security weaknesses related to policy or configuration are more likely to be identified through passive techniques than active ones. Another reason is that active security testing often has a narrow scope because of resource limitations, particularly time. Malicious attackers are not constrained by these considerations and often take as much time as necessary to exploit and penetrate a system or network. Also, organizations often avoid using techniques that could impact systems or networks, but attackers are likely to use whatever techniques are necessary.
- Active security testing results, especially penetration testing, may vary depending on the skill level of the testers. An active security testing technique such as vulnerability scanning requires a lesser skill set than penetration testing techniques, such as exploiting systems and gaining remote access.
- The nature of active security testing produces highly technical results that may be difficult to interpret in a business context. This requires effective communication between the test team and management. When possible, active security testing findings and recommendations should be presented in a business risk context.
- If any of the vulnerabilities that the testers exploit with active techniques are not mitigated, an attacker could potentially follow the steps of the test team to exploit the system or network.

2.3.2 External and Internal

External security testing is conducted from outside the organization's security perimeter. This approach provides an organization the ability to determine the security posture of its environment as viewed from outside the security perimeter, usually the Internet. The goal is to reveal vulnerabilities that could be

[5] One passive testing technique that can potentially impact networks is network sniffing, which involves connecting a sniffer to a hub, tap, or span port on the network.

exploited by an external attacker. External security testing includes all of the active testing techniques identified in Sections 4 and 5 with the exception of those techniques (e.g., file integrity checking) that can only be performed by hosts on the organization's internal network.

External security testing often begins with reconnaissance techniques that search public registration data, Domain Name System (DNS) server information, newsgroup postings, and other publicly available information to collect information (e.g., system names, IP addresses, operating systems, technical points of contact) that may assist the tester in identifying vulnerabilities. Next, enumeration begins by using network discovery and scanning techniques to determine external hosts and listening services. With external security testing, perimeter defenses such as firewalls, routers, and access control lists often limit the types of traffic that are allowed into the internal network. Testers often use techniques to evade these defenses, just as external attackers would. Depending on what protocols are allowed through, initial attacks are generally focused on commonly used and allowed application protocols such as File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and Post Office Protocol (POP). Externally accessible servers are tested for vulnerabilities that could allow access to internal servers and private information. External security testing also focuses on discovering vulnerabilities in access methods such as wireless access points, modems, and portals to internal servers.

Internal security testing is similar to external security testing except that the testers are on the internal network behind the perimeter defenses. The test team assumes the identity of either a trusted insider or an attacker who is able to penetrate the perimeter defenses. This type of testing can reveal vulnerabilities that could be exploited and can demonstrate the potential damage this type of attacker could cause. Internal security testing also focuses on system-level security and configuration, including application and service configuration, authentication, access control, and system hardening.

To perform internal security testing, testers are often granted some level of access to the network, usually as a general user. Testers are provided with information about the network that somebody with similar privileges would normally have. While these are generally the privileges of a standard employee or user, they can also be anything up to and including privileges of a system or network administrator. The level of access provided to the testers depends on the goals of the test. Using the level of access provided to them, testers then try to gain a greater level of access to the network and systems through privilege escalation (i.e., increasing their user-level privileges to administrator-level privileges or increasing their system administrator privileges to domain administrator privileges).

Internal security testing is not as limited as external testing because the testing takes place behind perimeter defenses, although there may be internal firewalls, routers, and switches that pose limitations to the testing. Also, passive techniques such as network sniffing may be used in addition to active techniques.

If both internal and external security testing are to be performed, the external testing usually occurs first. Conducting the external testing first is beneficial if the same test team will be performing both the internal and external testing. The test team will not have acquired any insider information, such as network architecture or system configuration, which would provide them an unfair advantage in comparison to an adversary, thereby reducing the realism of the security test.

2.3.3 Blue Team and Red Team

Blue team security testing is an overt method of assessing an organization's security posture. Blue teaming involves performing testing with the knowledge and consent of the organization's IT staff. This provides a comprehensive evaluation of the security posture by working side by side with the security staff. Blue team security testing can involve external and/or internal security testing.

Because the organization's IT staff is fully aware of and involved in the security testing, blue team testing may be able to provide guidance to limit the impact of the testing. Additionally, the testing may provide a training opportunity for the IT staff. During the test, the staff can observe the activities and methods with which the testers evaluate and potentially circumvent the implemented security measures. This gives context to the security requirements the IT staff implements or maintains, and also may help teach IT staff how to conduct security testing.

Red team security testing is a covert method that takes an adversarial approach to assessing an organization's security posture. Red teaming involves performing a security test without the knowledge of the organization's IT staff but with full knowledge and permission of upper management. Some organizations designate a trusted third party to ensure that the target organization does not initiate response measures associated with the attack without verifying that an attack is indeed underway (e.g., the activity it is seeing does not originate from a test). The trusted third party provides an agent for the testers, the management, the IT staff, and the security staff that mediates the activities and facilitates communications. This type of test is useful for testing technical security controls and the IT staff's response to perceived security incidents and its knowledge and implementation of the organization's security policy. Red teaming may be conducted with or without warning.

The purpose of red teaming is to examine the damage or impact an adversary can cause and does not focus on identifying all vulnerabilities. This type of testing does not test each security control, identify every vulnerability, or assess each system in an organization. Red teaming examines the organization from an adversarial perspective. It generally identifies and exploits the most rudimentary vulnerabilities to gain access to the network. If an organization's goal is to mirror a specific adversary, this type of testing requires special considerations, including acquiring and modeling threat data. The resulting scenarios provide an overall strategic viewpoint of the potential methods of exploit, attacks, risk, and impact of an intrusion.

In addition to failing to identify many vulnerabilities, red team security testing is often time-consuming and costly because of its stealth requirements. To operate in a stealth environment, a red team will have to slow its scans and other actions to move below the radar of the target organization's security staff. If security testing is to be performed in-house, training must also be considered in terms of time and budget. In addition, an organization may have staff trained to perform regular activities such as scanning and vulnerability assessments, but not specialized techniques such as penetration testing or application security testing. Blue team tests, the more overt form of security testing, are less expensive, carry less risk than red team testing, and are more frequently used. Red teaming provides a better indication of everyday security of the target organization since system administrators will not have heightened awareness.

3. Review Techniques

Review techniques passively examine systems, applications, networks, policies, and procedures to discover security vulnerabilities. Reviews also gather information to facilitate and optimize other testing techniques. Because review techniques are passive in nature, they pose minimal risk to systems and networks. This section covers several common review techniques for technical security testing: documentation review, log review, ruleset review, system configuration review, network sniffing, and file integrity checking.

3.1 Documentation Review

Documentation review determines if the technical aspects of policies and procedures are current and comprehensive. These documents provide the foundation for an organization's security posture but are often overlooked during testing. The organization's security groups should provide the testers with appropriate documentation to ensure a comprehensive review. Documents to review for technical accuracy and completeness include security policies, architectures, and requirements; standard operating procedures; system security plans and authorization agreements; memoranda of understanding and agreement for system interconnections; and incident response plans.

Documentation review can discover gaps and weaknesses that could lead to missing or improperly implemented security controls. Testers typically verify that the organization's documentation is compliant with standards and regulations such as FISMA. They also look for policies that are deficient or outdated. Common documentation weaknesses include security procedures for OSs or protocols that are no longer used, and failure to include new OSs and protocols. Documentation review will not ensure that security controls are implemented properly, only that the necessary direction and guidance exist to support the security infrastructure.

The results of documentation review can be used to fine-tune other testing techniques. For example, if a password management policy specifies minimum password length and complexity requirements, this information can be used to configure password cracking tools, which will result in more efficient password cracking.

3.2 Log Review

Log review determines if security controls are logging the proper information and if the organization is adhering to log management policies.[6] As a source of historical information, audit logs can be used to help validate that the system is operating according to policies. For example, if the logging policy states that all authentication attempts to critical servers must be logged, the log review will determine if this information is being logged and with the appropriate level of detail. Log review may also reveal problems such as misconfigured services and security controls, unauthorized access, and attempted intrusions. For example, if an intrusion detection system (IDS) sensor is placed behind a firewall, its logs can be used to examine the communications that are allowed into the network by the firewall. If this sensor registers activities that the firewall should be blocking, it indicates that the firewall is not configured securely.

The following are examples of log information that may be useful when conducting security testing:

- Authentication server or system logs may include successful and failed authentication attempts.

[6] NIST SP 800-92, *Guide to Security Log Management*, provides more information on security log management methods and techniques, including log review. It is available at http://csrc.nist.gov/publications/nistpubs/.

- System logs may include system and service startup and shutdown information, installation of unauthorized software, file accesses, security policy changes, account changes (e.g., account creation and deletion, account privilege assignment), and use of privileges.
- Intrusion detection and prevention system logs may include malicious activity and inappropriate use.
- Firewall and router logs may include outbound connections that indicate compromised internal devices (e.g., rootkits, bots, Trojan horses, spyware).
- Firewall logs may include unauthorized connection attempts and inappropriate use.
- Application logs may include unauthorized connection attempts, account changes, use of privileges, and application or database usage information.
- Antivirus logs may include update failures and other indications of outdated signatures and software.
- Security logs, in particular patch management and some IDS and intrusion detection system (IPS) products, may record information on known vulnerable services and applications.

Manual log review can be extremely cumbersome and time-consuming. Automated audit tools can significantly reduce review time and generate reports (predefined and customized) that summarize the log contents to a set of specific activities. Testers can also use automated tools to facilitate analysis of logs by converting logs in different formats to a single standard format for analysis. If testers are reviewing a specific action, such as the number of failed logon attempts in an organization, they can use tools to filter logs based on the activity they are checking.

3.3 Ruleset Review

A ruleset is a collection of rules or signatures that network traffic or system activity is compared against to determine an action to take, such as forwarding or rejecting a packet, creating an alert, or allowing a system event. A ruleset review ensures ruleset comprehensiveness and identifies gaps and weaknesses on security devices and throughout layered defenses, such as network vulnerabilities, policy violations, and unintended or vulnerable communication paths. A ruleset review can also indicate inefficiencies that could be negatively impacting the performance of the ruleset.

Rulesets to review include network- and host-based firewall and IDS/IPS rulesets and router access control lists. The following list provides examples of the types of checks most commonly performed in ruleset reviews:

- Router access control lists
  - Each rule is still required (for example, rules that were added for temporary purposes are removed as soon as they are no longer needed).
  - Only traffic that is authorized per policy is permitted and all other traffic is denied by default.
- Firewall rulesets
  - Each rule is still required.
  - The rules enforce least privilege access, such as specifying only required IP addresses and ports.
  - More specific rules are triggered before general rules.
  - There are no unnecessary open ports that could be closed to tighten the perimeter security.

- The ruleset does not allow traffic to bypass other security defenses.
- For host-based firewall rulesets, the rules do not indicate the presence of backdoors, spyware activity, or prohibited applications such as peer-to-peer file sharing programs.

- IDS/IPS rulesets
  - Unnecessary signatures have been disabled or removed to eliminate false positives and improve performance.
  - Necessary signatures are enabled and have been fine-tuned and properly maintained.

3.4 System Configuration Review

System configuration review is the process of identifying weaknesses in security configuration controls, such as systems not being hardened properly or not being configured according to security policies. For example, system configuration review will reveal unnecessary services and applications, improper user account and password settings, and improper logging and backup settings. Examples of security configuration files that may be reviewed are Windows security policy settings and Unix *inet.d* files.

Testers using manual review techniques use security configuration guides or checklists to verify that system settings are configured to minimize security risks.[7] To perform a manual system configuration review, testers access various security settings on the device and compare them with the recommended settings on the checklist. Those settings that do not meet the minimum security standards are flagged and reported.

The NIST Information Security Automation Program (ISAP) is an initiative to enable automation and standardization of technical security operations. The Security Content Automation Protocol (SCAP) is a method for using specific standards to enable automated vulnerability management, measurement, and policy compliance evaluation.[8] The NIST SCAP files are written for FISMA compliance and NIST SP 800-53A security control testing. Other tools can retrieve and report security settings and can provide remediation guidance. Automated tools are usually executed directly on the device under test, but they can also be executed on a system that has network access to the device under test. While automated system configuration reviews are faster than manual methods, there may be some settings that must be checked manually. Both manual and automated methods require root or administrator privileges to view some security settings.

3.5 Network Sniffing

Network sniffing is a passive technique[9] that monitors network communication, decodes protocols, and examines headers and payloads for information of interest. In addition to being a review technique, network sniffing is also a target identification and analysis technique (see Section 4.1). Reasons for using network sniffing include the following:

- Capturing and replaying network traffic
- Performing passive network discovery (identifying active devices on the network)

---

7 NIST maintains a repository of security configuration checklists for IT products at http://checklists.nist.gov/.

8 More information on SCAP is located at http://nvd.nist.gov/scap/scap.cfm.

9 Sniffers may perform domain name lookups for the traffic they collect, in which case they generate network traffic. Domain name lookups can be disabled for stealthy network sniffing.

- Identifying OSs, applications, services, and protocols, including unsecured protocols (e.g., telnet) and unauthorized protocols (e.g., peer-to-peer file sharing)
- Identifying unauthorized and inappropriate activity, such as the transmission of proprietary information
- Collecting information, such as unencrypted usernames and passwords.

Network sniffing is a passive technique that has little impact on systems and networks. The most noticeable impact is on bandwidth or computing power utilization. The sniffer, which is the tool used to conduct network sniffing, requires a means to connect to the network, such as a hub, tap, or switch with port spanning. Port spanning is the process of copying the traffic transmitted on all other ports to the one port that has the sniffer installed. Organizations may deploy network sniffers in a number of locations in an environment. Common locations include the following:

- At the perimeter to assess the traffic entering and exiting the network
- Behind firewalls to assess that rulesets are accurately filtering traffic
- Behind IDSs/IPSs to determine if signatures are triggering and being responded to appropriately
- In front of a critical system or application to assess activity
- On a specific network segment to validate encrypted protocols.

One limitation for network sniffing is the use of encryption. Many attackers take advantage of encryption to cover their tracks and hide their activity. Security testers can see that communication is taking place but cannot see the contents. Another limitation of network sniffing is that it can only sniff the traffic of the local segment where it is installed. The tester must move the sniffer from segment to segment, install multiple sniffers throughout the network, and/or use port spanning. Testers may find it challenging to find an open physical network port to use for scanning on each segment. Network sniffing is also a somewhat labor-intensive activity that requires a high degree of human involvement in interpreting the network traffic.

3.6 File Integrity Checking

File integrity checkers provide a way to identify that system files have been changed. A file integrity checker computes and stores a checksum for every guarded file and establishes a database of file checksums. Stored checksums are subsequently recomputed to compare the current value with the stored value, thus identifying file modifications. A file integrity checker capability is usually included with any commercial host-based IDS and is also available as a standalone utility.

An integrity checker does not require a high degree of human interaction, but it needs to be used carefully to ensure that it is effective. File integrity checking is most effective when system files are compared with a database built using a system that is known to be secure, to create the initial reference database. This helps to ensure the reference database was not created with compromised files. The reference database should be stored offline so that attackers cannot compromise the system and hide their tracks by modifying the database. Testers can also compare their checksums against the NIST National Software Reference Library (NSRL)[10] or build their own libraries of checksums. For example, an organization may choose to run a calculation of the checksums of known Trojan horses. Also, because patches and other updates change files, the checksum database needs to be kept up-to-date.

[10] Additional information on the NSRL is available at http://www.nsrl.nist.gov/.

For file integrity checking, strong cryptographic checksums such as SHA-1 should be used to ensure the integrity of data stored in the checksum database. Federal agencies are required by the Federal Information Processing Standard (FIPS) PUB 140-2, *Security Requirements for Cryptographic Modules*[11] to use Secure Hash Algorithm (SHA) (e.g., SHA-1, SHA-256).

3.7 Summary

Table 3-1 summarizes the major capabilities of the review techniques discussed in Section 3.

Table 3-1. Review Techniques

Type of Test	Capabilities
Documentation Review	• Evaluates policies and procedures for technical accuracy and completeness
Log Review	• Provides historical information on system use, configuration, and modification • Could reveal potential problems and policy deviations
Ruleset Review	• Reveals holes in ruleset-based security controls
System Configuration Review	• Evaluates the strength of system configuration • Validates systems are configured in accordance with the hardening policy
Network Sniffing	• Monitors network traffic on the local segment to capture information such as active systems, OSs, communication protocols, services, and applications • Verifies encryption of communications
File Integrity Checking	• Identifies changes to important files; can also identify some forms of unwanted files, such as well-known attacker tools

There are risks associated with each technique and combination of techniques. To ensure each technique is executed safely and accurately, each member of the test team should have a certain baseline skill set. Table 3-2 provides guidelines for the minimum skill set for each testing technique presented in Section 3.

Table 3-2. Baseline Skill Set for Review Techniques

Technique	Baseline Skill Set
Documentation Review	General knowledge of security from a policy perspective
Log Review	Knowledge of log formats and ability to interpret and analyze log data; ability to use automated log analysis and log correlation tools
Ruleset Review	Knowledge of ruleset formats and structures; ability to correlate and analyze rulesets from a variety of devices
System Configuration Review	Knowledge of secure system configuration, including OS hardening and security policy configuration for a variety of OSs; ability to use automated security configuration testing tools
Network Sniffing	General TCP/IP and networking knowledge; ability to interpret and analyze network traffic; ability to deploy and use network sniffing tools
File Integrity Checking	General file system knowledge; ability to use automated file integrity checking tools and interpret the results

[11] FIPS PUB 140-2 is available at http://csrc.nist.gov/publications/fips/.

4. Target Identification and Analysis Techniques

This section addresses target identification and analysis techniques, which focus on identifying active devices and their associated ports and services, and analyzing them for potential vulnerabilities. The tester can then use this information to further explore devices to validate the existence of the vulnerabilities.

4.1 Network Discovery

Network discovery uses various methods to discover active and responding hosts on a network, identify weaknesses, and learn how the network operates. There are both passive and active techniques for discovering devices on a network. Passive techniques use a network sniffer to monitor network traffic and record the IP addresses of the active hosts, and they can report which ports are in use and which OSs have been discovered on the network. Passive discovery is usually performed from a host on the internal network where it can monitor host communications. It does this without sending out a single probing packet. Passive discovery takes more time to gather information than active discovery does, and hosts that do not send or receive traffic during the monitoring period might not be reported.

Active techniques send various types of network packets, such as Internet Control Message Protocol (ICMP) pings, to solicit responses from network hosts. This is generally done through the use of an automated tool. One activity, which is known as OS fingerprinting, enables the tester to determine the system's OS by sending it a mix of normal, abnormal, and illegal network traffic. Another activity is sending packets to common port numbers to generate responses that indicate the ports are active. The tool analyzes the responses from these activities and compares the information it discovers with known traits of packets from particular OSs and of particular network services. This allows the tool to identify hosts, the OSs they run, their ports, and the state of those ports. Active discovery also identifies relationships between hosts, including which hosts communicate with each other, how frequently the communication occurs, and what type of traffic is being passed between hosts. This information can be used for several purposes, including gathering information on targets for penetration testing, generating topology maps, determining firewall and IDS configurations, and discovering vulnerabilities on systems and in network configurations.

Discovery tools have many ways of acquiring information, including the following:

- **TCP SYN scan (also known as *half open scanning*)**. A TCP SYN frame is sent to gather open port information, but the handshake is reset before it can be completed. If the discovery tool receives an acknowledgement (ACK), then the port is open. Since a true connection never occurs in a SYN scan and many systems do not record the activity in their logs, they are stealthier than a completed handshake. A second advantage of a SYN scan is its speed. A concern associated with SYN scans is the possibility that the target system could become flooded with outstanding SYNs, resulting in an accidental denial of service (DoS) attack.
- **TCP ACK scan**. A TCP ACK scan, which is a TCP packet with the ACK flag set, often will pass through simple firewalls when SYN scans are blocked and will elicit responses from hosts that might have been configured to ignore ICMP pinging. Scans of port 80 (HTTP) can identify active web sites that block ICMP echo requests (pings), whereas an open port will elicit a RST (reset) response. This type of scanning seeks to determine whether the protecting firewall is using simple versus advanced packet filtering techniques.
- **Xmas scan**. An Xmas scan sends a TCP packet with FIN, URG, and PSH flags set. If the port is open, no response is sent. If the port is closed, a RST/ACK response is sent.

- **ACK value**. IP stacks differ in the sequence value they use for the ACK field. On sending an Xmas scan to a closed port, Windows will usually send an ACK with the initial sequence number (ISN) incremented by one. Most Unix-based OSs will send an ACK with the same ISN set as the probe packet.
- **FIN scan**. A FIN packet is sent to an open port. The correct behavior is for the port not to respond.[12] Nevertheless, many stack implementations, such as in older versions of Windows, will respond.
- **SYN/FIN scan**. This scan combines SYN and FIN flags. These packets should not normally appear on a network and indicate a malicious scan.
- **NULL scan**. A NULL scan sends packets with no flags set. These packets normally should not appear on a network, nor should systems respond to them.
- **ICMP Echo scan (ping sweep)**. This technique seeks to identify active hosts by using the ICMP echo request (ping) command. An ICMP echo reply message in response to an ICMP echo request message indicates that the target host is alive.
- **TCP options**. Sending a packet with multiple options set, such as no operation, timestamps, and maximum segment size, allows the tester to maximize the information gained with each packet received. More conclusive target assumptions can be made concerning its OS.

Enterprise firewalls and intrusion detection systems can usually identify many instances of these scans, particularly those that use the most suspicious packets (e.g., SYN/FIN scan, NULL scan). Testers planning on performing discovery through firewalls and intrusion detection systems should consider which types of scans are most likely to provide results without drawing the attention of security administrators, or how scans can be conducted in a more stealthy manner to improve their chances of success (such as more slowly or from a variety of source IP addresses). Testers should also be cautious when selecting what types of scans to use against older systems, particularly those that are known to have weak security, because certain scans can cause system failures. Typically, the closer the scan is to normal activity, the less likely it is to cause operational problems.

Network discovery may also find unauthorized or rogue devices operating on the network. For example, an organization that only uses a few OSs could quickly identify rogue devices using other OSs. Once a wired rogue device is identified,[13] it can be located by using existing network maps and the information already collected on the rogue device's network activity to attempt to identify the switch to which it is connected. It may be necessary to generate additional network activity with the rogue device, such as pings, to locate the correct switch. The next step is to identify the switch port on that switch associated with the rogue device, and finally to physically trace the cable from that switch port to the rogue device.

There are a number of tools for network discovery. (Many active discovery tools can be used for passive network sniffing and port scanning as well.) Most tools offer a graphical user interface (GUI), and some also offer a command-line interface. Command-line interfaces may take longer to learn than GUIs because of the number of commands and switches, which specify what tests the tool should perform, that a tester must learn to use the tool effectively. In addition, developers have written a number of modules for open source tools that allow testers to easily parse tool output. For example, using a tool's eXtensible Markup Language (XML) output capabilities, a little scripting, and a database creates a more powerful tool, which can monitor the network for unauthorized services and machines. Learning what the many commands do and how to combine them is more efficiently achieved with an experienced security engineer. Most experienced IT professionals, such as system administrators and other network engineers,

[12] RFC 793, which is available at http://www.ietf.org/rfc/rfc0793.txt, provides early TCP specifications.
[13] See Section 4.4 for information on locating wireless rogue devices.

should be able to interpret the results, but working with the discovery tools themselves is more efficiently achieved by an experienced security engineer.

Some of the advantages of active discovery, compared to passive discovery, are that the assessment can be conducted from a different network and usually requires little time to gather information. In passive discovery, to ensure that all the hosts are captured, traffic needs to hit all points, which can be time-consuming, especially in larger enterprise networks.

A disadvantage to active discovery is that it tends to generate network noise that can result in network latency. Since active discovery sends out queries to receive responses, this added network activity could slow down traffic or cause packets to be dropped in poorly configured networks, if performed at high volume. Active discovery can also trigger IDS alerts. Unlike passive discovery, active discovery reveals where it is originating. In addition, the ability to successfully discover all network systems can be affected by environments that have protected network segments and perimeter security devices and techniques. For example, an environment that is using network address translation (NAT), which allows organizations to have internal, non-publicly routed IP addresses that are translated to a different set of public IP addresses for external traffic, may not be able to be accurately discovered from points external to the network or from protected segments. In addition, personal and host-based firewalls on the target devices may also block discovery traffic. There may be misinformation received as a result of trying to instigate activity from devices. Active discovery presents information from which conclusions have to be drawn about settings on the target network.

For both passive and active discovery, the information received is usually not completely accurate. For instance, only the hosts that are on and connected during active discovery will be identified; if systems or a segment of the network is offline during the test, there is a potential for a large gap in discovering devices. Passive discovery will only find devices that transmit or receive communications during the discovery period. However, products such as network management software can provide continuous discovery capabilities, automatically alerting when a new device is present on the network. Continuous discovery may scan IP address ranges for new addresses or monitor new IP address requests. Also, many discovery tools can be scheduled to run regularly, such as every x days at a particular time, providing more accurate results than running such tools only occasionally.

4.2 Network Port and Service Identification

Network port and service identification involves using a port scanner to identify the network ports and services operating on active hosts, such as FTP and HTTP, and the specific application running each identified service, such as Microsoft Internet Information Server (IIS) or Apache for the HTTP service. Organizations should conduct network port and service identification to identify hosts, if that has not already been done by other means (e.g., network discovery), and identify potentially vulnerable services. This information can be used to select targets for penetration testing.

If a port scanner is run against a range of IP addresses, it will first identify active hosts in the address range specified by the user. The port scanner may use a number of host discovery techniques, such as ICMP ECHO and ECHO_REPLY packets or TCP SYN/ACK packets. Once active hosts are identified, they are scanned for open TCP and User Datagram Protocol (UDP) ports.[14] All basic scanners can identify active hosts and open ports, but some scanners provide additional information on the scanned hosts. The information gathered during an open port scan often assists in the identification of the target OS. This process is called OS *fingerprinting*. For example, if a host has TCP ports 135, 139, and 445

[14] In TCP/IP terminology, a port is where an application receives information from the transport (TCP/UDP) layers. For example, all data received on TCP port 80 is forwarded to the Web server application. If an IP address identifies a particular host, the port is used to identify a particular service (HTTP, FTP, SMTP, etc.) running on that host.

open, it is probably a Windows host, or possibly a Unix host running Samba. Other items—such as the TCP packet sequence number generation and responses to packets—also provide a clue to identifying the OS. Nevertheless, OS fingerprinting is not foolproof. For example, firewalls filter (block) certain ports and types of traffic, and system administrators can configure their systems to respond in nonstandard ways to camouflage the true OS.

Some scanners can assist in identifying the application running on a particular port. This process is called service identification. Many scanners use a services file that lists common port numbers and typical associated services; for example, a scanner that identifies that TCP port 80 is open on a host may report that a web server is listening at that port. However, this is only a guess, and additional steps are needed to confirm it. Some scanners can initiate communications with an observed port and analyze the communications sent by that port to determine what service is there, often comparing the observed activity to a repository of information on common services and service implementations. These techniques can also be used to attempt to identify the service application and application version, such as which web server software is in use; this is known as *version scanning*. A well-known form of version scanning, *banner grabbing*, involves capturing the banner information transmitted by the remote port when a connection is initiated. Banner information can include the application type, application version, and even OS type and version. Version scanning is not foolproof, because a security-conscious administrator can alter the transmitted banners or other characteristics of a service to attempt to conceal its true nature. Still, version scanning is far more accurate than simply relying on a scanner's services file.

A number of scanners support different scanning methods with different strengths and weaknesses, which are usually explained in the scanner documentation. For example, certain scanners are better suited for scans through firewalls, and others are better suited for scans inside the firewall. Depending on the port scanner, the results will differ. Some scanners respond with a simple open or closed response for each port, while others offer additional detail (e.g., filtered or unfiltered) that may assist the tester in determining what other types of scans would be most helpful for gaining additional information.

Network port and service identification often uses the IP address results of network discovery as the devices to scan. Port scans can also be run independently on entire blocks of IP addresses. In this case, port scanning performs network discovery by default by identifying active hosts on the network. The result of network discovery and network port and service identification is a list of all active devices operating in the address space that responded to the port scanning tool, along with responding ports. There could be additional active devices that do not respond to scanning, such as those that are shielded by firewalls or turned off. Testers can try to find these devices by performing the scan on the devices themselves, placing the scanner on a segment that can access the devices, or attempting to evade the firewall through the use of alternate scan types (e.g., SYN/FIN or Xmas scan).[15]

It is recommended that if both external and internal scanning are to be used, and the test team is intentionally performing the testing "blind", that the external scanning be performed first. When scans are performed outside the firewall first, logs can be reviewed and compared before and during internal testing. When performing external scanning, testers may use any of the latest stealth techniques to attempt to get packets through firewalls while evading detection by IDS and IPS.[16] Tools that use fragmentation, duplication, overlap, out-of-order, and timing techniques to alter packets so that they blend into and appear more like normal traffic are recommended. Internal testing tends to use less aggressive scanning methods than external testing because internal scans are usually not blocked as often as external

[15] Many firewalls can recognize and block various alternate scan types, so testers may not be able to use them to evade firewalls in many environments.

[16] This can be particularly helpful in improving the tuning and configuration of IDSs and IPSs.

scans. Because of this, using more aggressive scans internally significantly increases the changes of disrupting operations, while not necessarily improving scan results (less aggressive scans often work well when done internally). Being able to scan a network with customized packets is also ideal for internal testing, because checking for specific vulnerabilities requires highly customized packets. Tools with a packet-builder ability are helpful with this process. Once built, the packets can be sent through the use of another scanning program that will collect the results. Because customized packets could trigger a DoS, this type of test should be conducted during periods of low network traffic, such as overnight or on weekends.

Although port scanners identify active hosts, OSs, ports, services, and applications, they do not identify vulnerabilities. Further investigation would be needed to confirm the presence of insecure protocols (e.g., TFTP, telnet), malware, unauthorized applications, and vulnerable services. To identify vulnerable services, the tester compares the identified version numbers of the services with a list of known vulnerable versions. Alternatively, the tester can perform automated vulnerability scanning, which is discussed in Section 4.3. Thus, with port scanners, the scanning process itself is highly automated, but the interpretation of scanned data is not.

Port scanning may disrupt network operations by consuming bandwidth and slowing network response times. Nevertheless, port scanning does enable an organization to ensure that its hosts are configured to run only approved network services. Scanning software should be carefully selected to minimize disruptions to operations. Port scanning can also be conducted after hours to ensure minimal impact to operations.

4.3 Vulnerability Scanning

Like network port and service identification, vulnerability scanning identifies hosts and host attributes, such as OS, applications, and open ports, but it also attempts to identify associated vulnerabilities, as opposed to relying on human interpretation of the results. (Many vulnerability scanners can accept results from network discovery and network port and service identification, thus reducing the amount of work needed for vulnerability scanning.) Vulnerability scanning can also help identify out-of-date software versions, missing patches, and misconfigurations, and validate compliance with or deviations from the organization's security policy. To accomplish this, vulnerability scanners identify OSs and major software applications running on hosts and match them with information on known vulnerabilities stored in the scanners' vulnerability databases.

Vulnerability scanners provide the following capabilities:

- Testing compliance with host application usage and security policies
- Providing information on targets for penetration testing
- Providing information on mitigating discovered vulnerabilities.

Vulnerability scanners can be run against a host either from the network or locally. A network-based scanner is installed on a single system to quickly locate and test numerous hosts over the network. Network-based scanning is used primarily for performing network discovery and identifying open ports and related vulnerabilities. In most cases, network-based scanning is not limited by the OS of targeted systems. Network-based scanning can be conducted both internally and externally; internal scanning usually uncovers more vulnerabilities than external scanning, but testing from both viewpoints is important. External scanning generally results in the discovery of fewer vulnerabilities because of perimeter security devices that block traffic, limiting the testers to scanning only the ports that are authorized to pass traffic. The testers may find challenges similar to those faced with network discovery,

such as the use of NAT or personal and host-based firewalls. To overcome the challenges of NAT enabling the team to successfully conduct network-based scanning, the test team can request the firewall administrator to enable port forwarding on specific IP addresses or groups of addresses, provided the firewall supports it. To address the challenges of personal or host-based firewalls, the test team could request they be configured to permit traffic from the test systems' IP addresses during the testing period. While this will give the tester additional insight into the network, it does not accurately reflect the capabilities of an external attacker. Alternatively, the test team can perform scanning on the individual hosts.

To perform local vulnerability scanning, a scanner is installed on each host to be tested. Local scanning is used primarily to identify host OS and application misconfigurations and vulnerabilities, both network exploitable and locally exploitable. Local scanning is able to detect vulnerabilities with a higher degree of detail than network-based scanning, because local scanning usually requires not only host (local) access but also a root or administrative account. Some scanners also offer the capability of repairing local misconfigurations.

A vulnerability scanner is a relatively fast and easy way to quantify an organization's exposure to surface vulnerabilities. A surface vulnerability is a weakness as it exists in isolation, independent from other vulnerabilities. The system's behaviors and outputs in response to attack patterns submitted by the scanner are compared against the behaviors and outputs that characterize the signatures of known vulnerabilities. Matches between the behavior or output and the tool's vulnerability signatures are reported by the tool. In addition to signature-based scanning, some vulnerability scanners attempt to simulate the reconnaissance attack patterns used by attackers to probe for exposed, exploitable vulnerabilities, and report the vulnerabilities found when such techniques are successful.

A difficulty in identifying the risk level of vulnerabilities is that they rarely exist in isolation. For example, there could be several low risk vulnerabilities that present a high risk when combined. Vulnerability scanners are unable to detect vulnerabilities that are revealed only because of potentially unending combinations of attack patterns. Thus, the tool may assign a low risk to each vulnerability, leaving the tester with a false sense of confidence in the security measures in place. A more reliable way of identifying the risk of vulnerabilities in aggregate is through penetration testing, which is discussed in Section 5.2.

Network-based vulnerability scanning has some significant weaknesses. As with network sniffing and discovery, network-based vulnerability scanning only uncovers vulnerabilities for active systems. Generally, they only identify surface vulnerabilities and are unable to address the overall risk level of a scanned network. Although the scan process itself is highly automated, vulnerability scanners can have a high false positive error rate (reporting vulnerabilities when none exist), so an individual with expertise in networking and OS security should interpret the results. Since network-based vulnerability scanning requires more information than port scanning to reliably identify the vulnerabilities on a host, network vulnerability scanning tends to generate significantly more network traffic than port scanning. This may have a negative impact on the hosts or network being scanned or on network segments through which scanning traffic is traversing. Many vulnerability scanners also include network-based tests for DoS attacks that, in the hands of an inexperienced tester, can have a considerable negative impact on scanned hosts.

Another significant limitation of vulnerability scanners is that, like virus scanners and IDSs, the vulnerability scanners rely on a repository of signatures. This requires the testers to update signatures frequently to ensure the scanner recognizes the latest vulnerabilities. Before running any scanner, an organization should install the latest updates to its vulnerability database. Some vulnerability scanner

databases are updated more regularly than others. The frequency of updates should be a major consideration when choosing a vulnerability scanner.

Most vulnerability scanners allow the user to perform different levels of testing in terms of thoroughness. More comprehensive vulnerability scanning is likely to detect more vulnerabilities but slow the overall scanning process. Less comprehensive scanning is likely to take less time but result in identifying only well-known vulnerabilities, not more esoteric ones. It is generally recommended that organizations conduct a thorough vulnerability scan if resources permit.

Vulnerability scanning is a somewhat labor-intensive activity that requires a high degree of human involvement in interpreting the results. It may also disrupt network operations by taking up bandwidth and slowing response times. Nevertheless, vulnerability scanning is extremely important for ensuring that vulnerabilities are mitigated before they are discovered and exploited by adversaries.

As with all pattern-matching and signature-based tools, application vulnerability scanners typically have high false positive rates. The tester should configure and calibrate the scanner to minimize both false positives and false negatives to the greatest possible extent and to meaningfully interpret the results to identify the real vulnerabilities. Scanners also suffer from the high false negative rates that characterize other signature-based tools; vulnerabilities undetected by automated scanners can potentially be detected through the use of multiple vulnerability scanners or other forms of testing. A common practice is to use multiple scanners, which provides the testers a way to compare results.

4.4 Wireless Scanning

Wireless technologies, in the simplest sense, enable one or more devices to communicate without physical connections, such as network cables or peripheral cables. Wireless technologies range from simple technologies, such as wireless keyboards and mice, to complex systems, such as cell phone networks and enterprise wireless local area networks (WLAN). As the number and availability of wireless-enabled devices continue to increase, it is important for organizations to actively test and secure enterprise wireless environments.[17] Wireless scans will help organizations determine corrective actions to mitigate risks posed by wireless-enabled technologies.

The following factors in the organization's environment should be taken into consideration when planning wireless security testing:

- The location of the facility being scanned. The physical proximity of a building to a public area, such as streets and public common areas, or its location in a busy metropolitan area may increase the risk of wireless threats.
- The security level of the data transmitted using wireless technologies
- How often wireless devices connect to and disconnect from the environment, and what the typical traffic levels are for wireless devices (for example, occasional activity or fairly constant activity) because only active wireless devices will be discovered during a wireless scan

[17] For proper measures to secure IEEE 802.11 based WLANs, please refer to NIST SP 800-97, *Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i* and NIST SP 800-48 Revision 1 (DRAFT), *Wireless Network Security for IEEE 802.11a/b/g and Bluetooth*, available at http://csrc.nist.gov/publications/nistpubs/.

- Existing deployments of wireless intrusion detection and prevention systems (WIDPS[18]), which may already collect most of the information that would be gathered by testing.

Wireless scanning should be conducted using a mobile device that has wireless analyzer software installed and configured, such as a laptop, handheld device, or specialty device. The scanning software or tool should allow the operator to configure the device for specific scans and to scan in both passive and active modes. The scanning software should also allow the operator to configure it to identify deviations from the organization's wireless security configuration requirements.

The wireless scanning tool should be capable of scanning all IEEE 802.11a/b/g/n channels (including international frequencies/channels). In some cases, the scanning device should also be fitted with an external antenna to provide an additional level of radio frequency (RF) capturing capability. Support for other wireless technologies, such as Bluetooth, will help evaluate the presence of additional wireless threats and vulnerabilities. Note that devices using nonstandard technology or frequencies outside of the scanning tool's RF range will not be detected or properly recognized by the scanning tool. A tool such as an RF spectrum analyzer will assist organizations in identifying transmissions occurring within the frequency range of the spectrum analyzer. Spectrum analyzers generally analyze a large frequency range (e.g., 3 to 18 GHz). Although these devices do not analyze the traffic, they enable a test team to determine wireless activity within a specific frequency range and tailor additional testing accordingly.

Some devices also support mapping and physical location plotting through the use of a mapping tool, and in some cases, tools may also support Global Positioning System (GPS)-based mapping. Sophisticated wireless scanning tools allow the user to import a floor plan or map to assist in plotting the physical location of discovered devices. It is important to note that GPS has limited capabilities indoors.

An individual with a strong understanding of wireless networking, especially IEEE 802.11a/b/g/n technologies, should operate wireless scanning tools. The operator should be trained on the functionality and capability of the scanning tool and software so that the operator will better understand the information the scanning tool captures and will be more apt to identify potential threats or malicious activity. Individuals with similar skills should also be employed to conduct the analysis of data and results acquired from wireless scans. Scanning tool operators should also be aware of other RF signals authorized for use within the area being scanned.

4.4.1 Passive Wireless Scanning

Passive scanning should be conducted regularly to supplement wireless security measures already in place, such as WIDPSs.[19] Wireless scanning tools used to conduct completely passive scans do not transmit any data, nor do the tools in any way affect the operation of deployed wireless devices. By not transmitting any data, a passive scanning tool remains undetected by malicious users and other devices. This reduces the likelihood of individuals avoiding detection by disconnecting or disabling unauthorized wireless devices.

Passive scanning tools capture wireless traffic being transmitted within the range of the tool's antenna. Most tools provide several key attributes regarding discovered wireless devices, including service set identifier (SSID), device type, channel, media access control (MAC) address, signal strength, and number

[18] For more information, see NIST SP 800-94, *Guide to Intrusion Detection and Prevention Systems (IDPS)*, which is available at http://csrc.nist.gov/publications/nistpubs/.

[19] In some environments, the WIDPS implementation might be performing most of the same functions as passive wireless scanning would. Some WIDPS products offer mobile sensors, similar to the wireless scanning device setup described in Section 4.4. Organizations with WIDPS implementations should use the wireless scanning techniques described in this publication to supplement, not duplicate, the WIDPS functionality.

of packets being transmitted. This information can be used to evaluate the security of the wireless environment and to identify potential rogue devices discovered within range of the scanning device. The wireless scanning tool should also be able to assess the captured packets to determine if there are any operational anomalies or threats.

Wireless scanning tools scan each IEEE 802.11a/b/g/n channel/frequency separately, often scanning each channel for only several hundred milliseconds at a time. All transmissions on a specific channel may not be received by the passive scanning tool. For example, the tool may have been scanning channel 1 at the precise moment when a wireless device transmitted a packet on channel 5. This makes it important to set the dwell time of the tool to a time long enough to capture packets and short enough to efficiently scan each channel. Dwell time configurations will depend on the device or tool used to conduct the wireless scans. In addition, security personnel conducting the scans should slowly move through the area to be scanned to reduce the number of devices that are not detected.

Rogue devices can be identified in several ways through passive scanning, including the following:

- The MAC address of a discovered wireless device indicates the vendor of the device's wireless interface, so if an organization has only deployed wireless interfaces from vendors A and B, the presence of interfaces from any other vendor indicates potential rogue devices.
- If the organization has accurate records of deployed wireless devices, testers can compare the MAC addresses of discovered devices with the MAC addresses of authorized devices. Most scanning tools allow testers to enter a list of authorized devices. Because MAC addresses can be spoofed, testers should not assume that the MAC addresses of discovered devices are accurate. Still, checking MAC addresses can identify rogue devices that are not using spoofing.
- Rogue devices may use SSIDs that are not authorized by the organization.
- Some rogue devices may use SSIDs authorized by the organization, but they may not adhere to the organization's wireless security configuration requirements.

The signal strength of potential rogue devices should be reviewed to determine whether the devices are actually located within the confines of the facility or in the area being scanned. Devices operating outside an organization's confines can be eliminated as rogue devices, but these discovered devices might still pose significant risks, because the organization's devices might inadvertently associate to them.

4.4.2 Active Wireless Scanning

As an advanced step beyond passive wireless scanning, organizations can conduct active scanning. Active scanning builds on information collected during the passive scans and attempts to attach to discovered devices and conduct penetration or vulnerability-related testing. For example, organizations can conduct active wireless scanning on their authorized wireless devices to ensure that they meet wireless security configuration requirements, including authentication mechanisms, data encryption, and administration access, if this information is not already available through other means.

Organizations should be cautious in conducting active scans so that they do not inadvertently scan devices that are owned or operated by neighboring organizations within range. It is important to evaluate the physical location of devices before actively scanning them. Organizations should also be cautious in performing active scans of rogue devices that appear to be operating in the organization's facility. Such devices could belong to a visitor to the organization who inadvertently has wireless access enabled, or to a neighboring organization with a device that is very close to but not within the organization's facility.

Generally, organizations should focus on identifying and locating potential rogue devices, instead of performing active scans of such devices.

Organizations may use active scanning when conducting penetration testing on their own wireless devices. Tools are available that employ scripted attacks and functions, attempt to circumvent implemented security measures, and evaluate the security level of devices. For example, tools used to conduct wireless penetration testing attempt to connect to access points (AP) through various methods to circumvent security configurations. If the tool can gain access to the AP, it can obtain information from the AP and identify the wired networks and wireless devices connected to the AP. Some active tools may also identify vulnerabilities discovered on the wireless client devices or conduct wired network vulnerability tests, as outlined earlier in Section 4.

As active scanning is being performed, the organization's WIDPSs can be monitored to evaluate their capabilities and performance. Depending on the testing goals, the testers conducting the scans may need to inform the WIDPS administrators and wireless network administrators of pending scanning, so that the administrators are prepared for ensuring alarms and alerts. Also, some WIDPSs can be configured to ignore alarms and alerts triggered by a particular device, such as the device performing scanning.

Tools and processes to identify unauthorized devices and vulnerabilities on wired networks can also be used to identify rogue and misconfigured wireless devices. Wired side scanning for wireless devices is another process that can be conducted to discover and possibly locate rogue wireless devices. Sections 3.5 and 4.1 outline wired scanning.

4.4.3 Wireless Device Location Tracking

Security personnel operating the wireless scanning tool should attempt to locate suspicious devices. RF signals propagate in a manner relative to the environment, which makes it important for the operator of the scanning tool to understand how wireless technology operates to support this process. Mapping capabilities are useful during this process, but the main factors in supporting this capability are a knowledgeable operator and appropriate wireless antenna.

If rogue devices are discovered and physically located during the wireless scan, security personnel should ensure that specific policies and processes are followed on how a rogue device is handled, such as shutting it down, reconfiguring it to comply with the organization's policies, or removing it. If a device is to be removed, security personnel should evaluate the activity of the rogue device before confiscating it. This can be done through monitoring transmissions and attempting to access the device.

If discovered wireless devices cannot be located during the scan, security personnel should attempt to use a WIDPS to support the location of discovered devices. This can be done by using the WIDPS to locate a specific MAC address that was discovered during the scan. Properly deployed WIDPSs should have the ability to assist security personnel in locating devices. This usually involves the use of multiple WIDPS sensors to increase location identification granularity. Because the WIDPS will only be able to locate a device within several feet, using a wireless scanning tool may still be required to precisely locate the device.

4.4.4 Bluetooth Scanning

For organizations that want to confirm compliance with their Bluetooth security requirements, passive scanning for Bluetooth-enabled wireless devices should be conducted to evaluate the presence and activity of Bluetooth-enabled devices. Because Bluetooth has a very short range (on average 30 feet, with some devices having ranges as short as 3 feet), scanning for devices can be very difficult and time-

consuming. Testers should take range limitations into consideration when scoping this type of test. Organizations may want to perform scanning only in areas of their facilities that are accessible by the public, to see if attackers could gain access to devices using Bluetooth, or to perform scanning in a sampling of physical locations instead of the entire facility. Because many Bluetooth-enabled devices, such as cell phones and PDAs, are mobile, conducting passive scanning a number of times over a specific period of time may be necessary. Organizations should also test any Bluetooth infrastructure that they deploy, such as Bluetooth access points. If rogue access points are discovered, the organization should handle them according to their established policies and processes.

A number of tools are available for actively testing the security and operation of Bluetooth devices. These tools attempt to connect to discovered devices and perform attacks to surreptitiously gain access and connectivity to Bluetooth-enabled devices. Testers should be extremely cautious of performing active scanning because of the likelihood of inadvertently scanning personal Bluetooth devices, which are prevalent in many environments. Testers should generally use active scanning only when they are certain that the devices belong to the organization. Active scanning can be used to evaluate the security mode in which a Bluetooth device operates and the strength of Bluetooth password identification numbers (PIN). Active scanning can also be used to verify that Bluetooth devices are set to the lowest possible operational power setting to minimize their range. As with IEEE 802.11a/b/g rogue devices, rogue Bluetooth devices should be dealt with according to outlined policies and guidance.

4.5 Application Security Testing

Application security testing helps an organization determine whether its custom application software, such as Web applications, contains vulnerabilities that can be exploited and whether the software behaves and interacts securely with its users, other applications, and its execution environment. Application security testing can be performed in many ways, ranging from review of application source code to penetration testing of the implemented application.[20] Many application security tests subject the application to known attack patterns typical for the application's type. Such attack patterns may directly target the application itself or may attempt to attack the application indirectly by targeting its execution environment or security infrastructure. Examples of attack patterns are information leakage (e.g., reconnaissance, exposure of sensitive information), authentication exploits, session management exploits, subversion (e.g., spoofing, impersonation, command injections), and denial of service attacks.

Application security tests should be integrated into the software development life cycle of the application, so that they are performed early in the life cycle. For example, code reviews can be performed as code is being implemented, instead of waiting until the entire application is ready for testing. Tests should also be performed periodically after an application has gone into production, as well as when significant patches, updates, or other modifications are made to the application, or when the threat environment in which the application operates changes significantly.

There are many application security testing techniques available. They can be divided into white box techniques, which involve direct analysis of the application's source code, and black box techniques, which are performed against the application's binary executable without any knowledge of the source code. Most testing of custom applications is performed with white box techniques, since the source code is available; however, white box techniques cannot detect security defects in interfaces between components, nor can they identify security problems caused during compilation, linking, or installation-time configuration of the application. Still, white box techniques are generally more efficient and cost-effective for finding security defects in custom applications than black box techniques. Black box

[20] Some elements of application security testing, such as penetration testing an application, are target vulnerability validation techniques, not target identification and analysis techniques. Application security testing is discussed only in this section for brevity.

techniques should be used primarily to test the security of individual high-risk components, interactions between components, and interactions between the entire application or application system with its users, other systems, and the external environment. Black box testing should also be used to determine how effectively the application or application system can handle threats.

Because application security testing is such a complex topic, with dozens of techniques commonly used, it is outside the scope of the publication to provide more specific information on test techniques or recommendations for their use.

4.6 Summary

Table 4-1 summarizes the major capabilities of the target identification and analysis techniques discussed in Section 4.

Table 4-1. Target Identification and Analysis Techniques

Type of Test	Capabilities
Network Discovery	• Discovers active devices • Identifies communication paths and facilitates determining network architectures
Network Port and Service Identification	• Discovers active devices • Discovers open ports and associated services and applications
Vulnerability Scanning	• Identifies hosts and open ports • Identifies known vulnerabilities (although with high false positive rates) • Often provides advice on mitigating discovered vulnerabilities
Wireless Scanning	• Identifies unauthorized wireless devices within range of the scanners • Discovers wireless signals outside of an organization's perimeter • Detects potential backdoors and other security violations
Application Security Testing	• Reveals programming, logic, and configuration flaws that could be exploited

There are risks associated with each technique and combination of techniques. To ensure each technique is executed safely and accurately, each member of the test team should have a certain baseline skill set. Table 4-2 provides guidelines for the minimum skill set for each testing technique presented in Section 4.

Table 4-2. Baseline Skill Set for Target Identification and Analysis Techniques

Technique	Baseline Skill Set
Network Discovery	General TCP/IP and networking knowledge; ability to use both passive and active network discovery tools
Network Port and Service Identification	General TCP/IP and networking knowledge; knowledge of ports and protocols for a variety of OSs; ability to use port scanning tools; ability to interpret results from tools
Vulnerability Scanning	General TCP/IP and networking knowledge; knowledge of ports, protocols, services, and vulnerabilities for a variety of OSs; ability to use automated vulnerability scanning tools and interpret and analyze the results
Wireless Scanning	General knowledge of computing and radio transmissions in addition to specific knowledge of wireless protocols, services, and architectures; ability to use automated wireless scanning and sniffing tools
Application Security Testing	Knowledge of specific programming languages and protocols; knowledge of application development and secure coding practices; understanding of vulnerabilities introduced with poor coding practices; ability to use automated software code review and other application security test tools; knowledge of common application vulnerabilities

5. Target Vulnerability Validation Techniques

This section addresses target vulnerability validation techniques, which use the information produced from target identification and analysis to further explore the existence of potential vulnerabilities. The objective is to prove that a vulnerability does exist and demonstrate the security exposures that occur by exploiting the vulnerability. Target vulnerability validation involves the most risk in information security testing, since these testing techniques have a higher potential to impact the target system or network than other testing techniques.

5.1 Password Cracking

Password cracking is the process of recovering secret passwords stored in a computer system or transmitted over networks. Password cracking identifies accounts with weak passwords. Passwords are sometimes stored and transmitted in an encrypted form called a *hash*. When a user enters a password, a hash of the entered password is generated and compared with a stored hash of the user's actual password. If the hashes match, the user is authenticated. Password cracking is performed on hashes that are either intercepted when they are transmitted across a network (using a network sniffer) or retrieved from the targeted system. The latter generally requires administrative-level access on the target system. Once the hashes are obtained, an automated password cracker rapidly generates hashes until a match is found or a tester halts the cracking attempt.

One method for generating hashes is a *dictionary attack*, which uses all words in a dictionary or text file. There are many dictionaries available on the Internet that cover most major and minor languages, names, popular television shows, etc. Another method of cracking is called a *hybrid attack*, which builds on the dictionary method by adding numeric and symbolic characters to dictionary words. Depending on the password cracker being used, this type of attack will try a number of variations, such as using common substitutions of characters and numbers for letters (e.g., p@ssword and h4ckme). Some will also try adding characters and numbers to the beginning and end of dictionary words (e.g., password99, password$%).

Another password-cracking method is called the *brute force* method. Brute force generates all possible passwords up to a certain length and their associated hashes. Since there are so many possibilities, it can take months to crack a password. Although brute force can take a long time, it usually takes far less time than most password policies specify for password changing. Consequently, passwords found during brute force attacks are still too weak. Theoretically, all passwords can be cracked by a brute force attack, given enough time and processing power, although, depending on the algorithm, it could take many years and require serious computing power. Security testers and attackers often have multiple machines to which they can spread the task of cracking passwords, which greatly shortens the time required.

Password crackers should be run on the system with the same frequency as the expiration policy to ensure compliance with the organization's password policy by verifying acceptable password composition. Password cracking that is performed offline produces little or no impact on the system or network. The benefits of password cracking include validating the password policy, verifying policy compliance, and verifying the password filter.

5.2 Penetration Testing

Penetration testing is security testing in which evaluators mimic real-world attacks to attempt to identify methods for circumventing the security features of an application, system, or network. Penetration testing often involves issuing real attacks on real systems and data, using the common tools and techniques used by attackers. Most penetration tests involve looking for combinations of vulnerabilities on a single

system or multiple systems that can be used to gain more access than could be achieved through any single vulnerability. Penetration testing can also be useful for determining the following:

- How well the system tolerates real-world style attack patterns
- The likely level of sophistication an attacker needs to successfully compromise the system
- What additional countermeasures could mitigate the threats against the systems.

Penetration testing can be invaluable, but it is labor-intensive and requires great expertise to minimize the risk to targeted systems. Systems may be damaged or otherwise rendered inoperable in the course of penetration testing, even though the organization benefits in knowing how the system could have been rendered inoperable by an intruder. Although this risk is mitigated by experienced penetration testers, it can never be fully eliminated. Penetration testing should be performed only after careful consideration, notification, and planning.

5.2.1 Penetration Testing Phases

Figure 5-1 represents the four phases of penetration testing. In the planning phase, rules are identified, management approval is finalized and documented, and the testing goals are set. The planning phase sets the groundwork for a successful penetration test. No actual testing occurs in the planning phase.

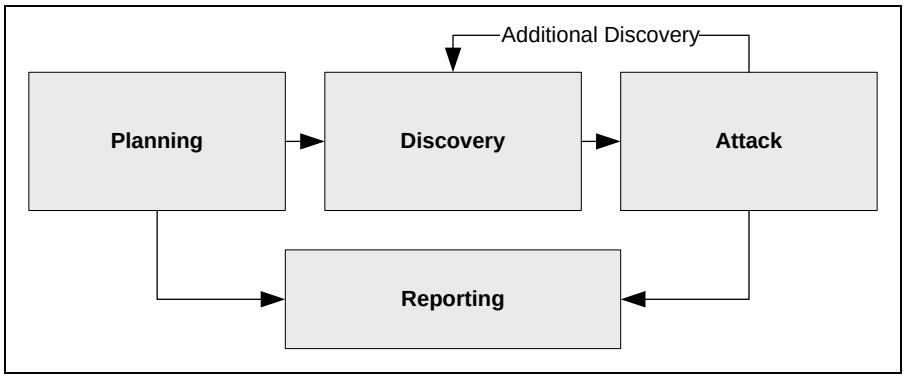

Figure 5-1. Four-Stage Penetration Testing Methodology

The discovery phase includes two parts. The first part begins the actual testing with information gathering and scanning. Network port and service identification, as described in Section 4.2, is conducted to identify potential targets. In addition to port and service identification, other techniques are used to gather the following information on the targeted network:

- **Host name and IP address information** can be gathered through many methods, including DNS interrogation, InterNIC (whois) queries, and network sniffing (generally only during internal tests).
- **Employee names and contact information** can be found by searching the organization's web servers or directory servers.
- **System information such as names and shares** can be found through methods such as NetBIOS enumeration (generally only during internal tests) and Network Information System (NIS) (generally only during internal tests).
- **Application and service information** such as version numbers can be recorded through banner grabbing.

The second part of the discovery phase is vulnerability analysis, which involves comparing services, applications, and OSs of scanned hosts against vulnerability databases (for vulnerability scanners this process is automatic). Human testers can use their own database or public databases such as the National Vulnerability Database (NVD) to identify vulnerabilities manually. Appendix C has more information on publicly available vulnerability databases. Manual processes can identify new or obscure vulnerabilities that automated scanners may miss, but these processes are much slower than an automated scanner.

Executing an attack is at the heart of any penetration test. Figure 5-2 represents the individual steps of the attack phase. This is the process of verifying previously identified potential vulnerabilities by attempting to exploit them. If an attack is successful, the vulnerability is verified and safeguards are identified to mitigate the associated security exposure. Frequently, exploits[21] that are executed do not grant the maximum level of access that can be gained by an attacker. Instead, they may result in the testing team learning more about the targeted network and its potential vulnerabilities, or they may induce a change in the state of the security of the targeted network. Some exploits enable the testers to escalate their privileges on the system or network to gain access to additional resources. Then additional analysis and testing are required to determine the true level of risk for the network, such as identifying what type of information can be gleaned, changed, or removed from the system. In the event an attack on a specific vulnerability proves impossible, the tester should attempt to exploit another discovered vulnerability. If the test team is able to exploit a vulnerability, it can install more tools on the target system or network to facilitate the testing process. These tools are used to gain access to additional systems or resources on the network and to gain access to information about the network or organization. Testing and analysis on multiple systems should be conducted during a penetration test to determine the level of access an adversary could gain. This is represented in the feedback loop in Figure 5-1 between the attack and discovery phase of a penetration test.

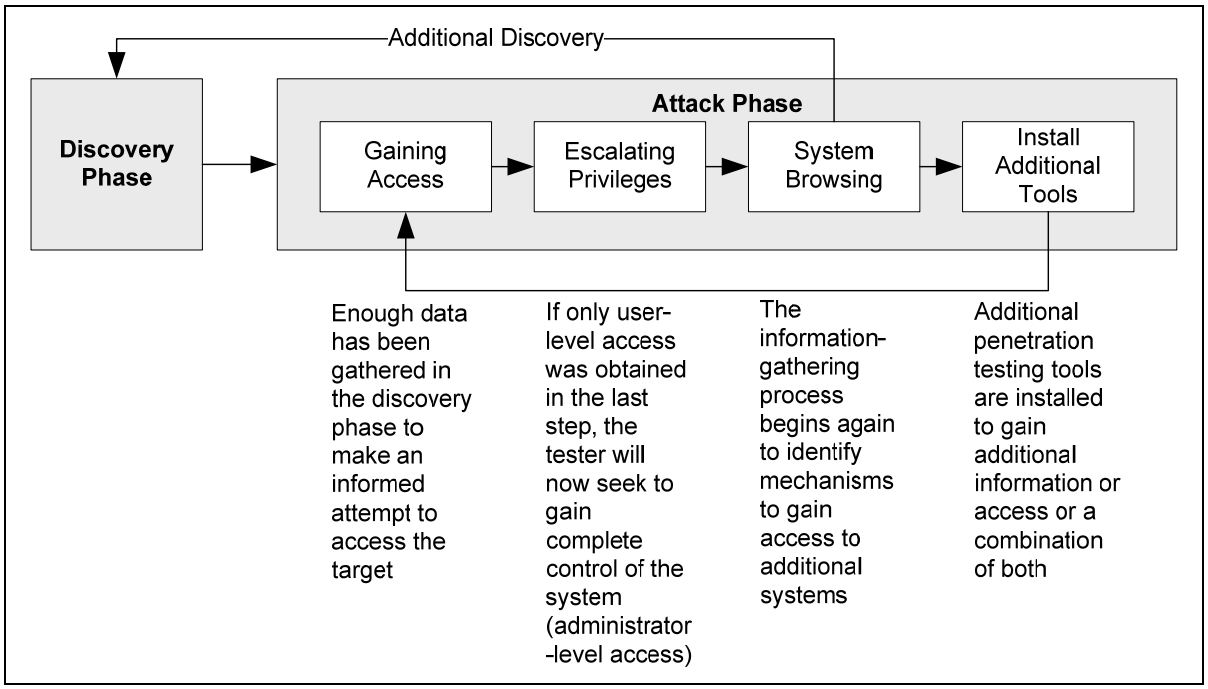

Figure 5-2. Attack Phase Steps with Loopback to Discovery Phase

[21] Exploit programs or scripts are specialized tools for exploiting specific vulnerabilities. The same cautions that apply to freeware tools apply to exploit programs and scripts. Some vulnerability databases, including Bugtraq, available at http://www.securityfocus.com/, provide exploit instructions or code for many identified vulnerabilities.

While vulnerability scanners only check for the possible existence of a vulnerability, the attack phase of a penetration test exploits the vulnerability to confirm its existence. Most vulnerabilities exploited by penetration testing fall into the following categories:

- **Misconfigurations.** Misconfigured security settings, particularly insecure default settings, are usually easily exploitable.
- **Kernel Flaws.** Kernel code is the core of an OS. The kernel code enforces the overall security model for the system, so any security flaw in the kernel puts the entire system in danger.
- **Buffer Overflows.** A buffer overflow occurs when programs do not adequately check input for appropriate length. When this occurs, arbitrary code can be introduced into the system and executed with the privileges of the running program, often administrative-level privileges.
- **Symbolic Links.** A symbolic link (symlink) is a file that points to another file. OSs include programs that will change the permissions granted to a file. If these programs run with privileged permissions, a user could strategically create symlinks to trick these programs into modifying or listing critical system files.
- **File Descriptor Attacks.** File descriptors are numbers the system uses to keep track of files rather than using filenames. Certain file descriptors have implied uses. When a privileged program assigns an inappropriate file descriptor, it exposes that file to compromise.
- **Race Conditions.** Race conditions can occur during the time a program or process has entered into a privileged mode. A user can time an attack to take advantage of the program or process's elevated privileges while the program or process is still in the privileged mode.
- **Incorrect File and Directory Permissions.** File and directory permissions control the access that users and processes have. Poor permissions could allow any number of attacks, including the reading or writing of password files or the addition of hosts to the list of trusted remote hosts.

The reporting phase occurs simultaneously with the other three phases of the penetration test (see Figure 5-1). In the planning phase, the ROE, test plans, and written permission are developed. In the discovery and attack phase, written logs are usually kept, and periodic reports are made to system administrators and/or management. Generally, at the end of the test a testing report is developed to describe the identified vulnerabilities, present a risk rating, and give guidance on the mitigation of the discovered weaknesses. Section 8 discusses in greater detail post-testing activities such as reporting.

5.2.2 Penetration Testing Logistics

Penetration test scenarios should focus on locating and targeting exploitable defects in the design and implementation of the application, system, or network. The tests should reproduce both the most likely and most damaging attack patterns, including worst-case scenarios, such as malicious actions by administrators. A penetration test scenario can be designed to simulate an inside attack, an outside attack, or both. Thus, external and internal security testing methods are considered. If both internal and external testing is to be performed, the external testing usually occurs first.

Outsider scenarios simulate the outsider-attacker who has little or no specific knowledge of the target and who works entirely from assumptions. To simulate an external attack, the testers are not provided with any real information about the target environment other than targeted IP addresses or address ranges. They perform open source research by collecting information on the targets from public web pages, newsgroups, and similar sites. They then use port scanners and vulnerability scanners to identify target hosts. Since the testers' traffic is most likely going through a firewall, the amount of information gleaned

from scanning is far less than what would be obtained if performing the test from an insider perspective. After identifying hosts on the network that can be reached from the outside, testers attempt to compromise one of the hosts. If successful, they may then use this access to compromise other hosts not generally accessible from outside the network. Penetration testing is an iterative process that leverages minimal access to gain greater access.

Insider scenarios simulate the actions of a malicious insider. An internal penetration test is similar to an external test, except that the testers are on the internal network (i.e., behind the firewall) and are granted some level of access to the network or specific systems on the network (generally as a user but sometimes at a higher level). The penetration testers then try to gain a greater level of access to the network and systems on the network through privilege escalation. The testers are provided with the network information that someone with their level of access would normally have. This is generally as a standard employee, although it can also be the information that a system or network administrator might have, depending on the goals of the test.

Penetration testing is important for determining how vulnerable an organization's network is and the level of damage that can occur if the network is compromised. It is important to be aware that depending on the organization's policies, test teams may not be permitted to use certain tools or techniques. Penetration testing also poses a high impact to the organization's networks and systems, because it uses real exploits and attacks against production systems and data. Because of the high cost and potential impact, annual penetration testing of an organization's network and systems may be sufficient. The results of penetration testing should be taken seriously, and discovered vulnerabilities should be mitigated. As soon as results are available, they should be presented to the organization's managers. Organizations should consider conducting less labor-intensive testing activities on a regular basis to ensure that they are maintaining the required security posture. A well-designed program of regularly scheduled network and vulnerability scanning, interspersed with periodic penetration testing, can help prevent many attacks and reduce the potential impact of successful attacks.

5.3 Remote Access Testing

Remote access testing assesses various remote access methods for vulnerabilities. Remote access includes technologies such as terminal servers, virtual private networks (VPN), secure shell (SSH) tunnels, remote desktop applications, and dial-up modems. Remote access testing is intended to discover alternative methods of entry into the network and circumvent perimeter defenses. Remote access testing is often performed as part of penetration testing, but can also be performed on its own to focus solely on remote access implementations. Testing techniques vary, depending on the type of remote access being tested and the specific goals of the test. Examples of commonly used techniques are as follows:

- **Discover unauthorized remote access services.** Port scanning may be used to discover open ports often associated with remote access services. In addition, systems may be manually checked for remote access services by viewing running processes and installed applications.
- **Review rulesets to find unintended remote access paths.** Remote access rulesets, such as those on VPN gateways, should be reviewed for holes or misconfigurations that may permit unwanted access.
- **Test remote access authentication mechanisms.** Each remote access method should require authentication, so testers should first verify that they are required to authenticate before gaining access. Next, testers can try default accounts and default passwords (e.g., guest accounts, maintenance accounts), as well as brute-force attacks. Social engineering can also be used to attempt to get a password reset or to gain access without having an authentication token (e.g., by claiming the token is lost). Testers can also try to gain access through authentication self-service programs that

allow a password reset by answering user-specific questions; this may also involve social engineering.

- **Monitor remote access communications.** Testers can monitor remote access communications with a network sniffer; if the communications are not protected, testers may be able to collect remote access authentication information from them, as well as other data sent and received by remote access users.

Active or intrusive remote access testing should be performed during low demand times to limit potential disruption to employees and the remote access systems.

Another aspect of remote access testing is assessing an organization's phone systems for vulnerabilities that permit unauthorized or unsecured access. NIST SP 800-24, *PBX Vulnerability Analysis*[22], provides information on the elements and approaches to private branch exchange (PBX) security testing. For remote access, the primary target of phone system testing is modems. Although use of modems has decreased because of the wide availability of wired and wireless network access, some successful attacks continue to be launched through unauthorized modems. For example, some users still enable modem access on their work computers for remote access, and some organizations use older technologies, such as building operations controllers and switches, that have maintenance modems enabled. A single compromise via a modem could allow an attacker direct and undetected access to a network, avoiding perimeter security.

Several available software packages allow network administrators, and attackers, to dial large blocks of phone numbers in search of available modems. This process is called *war dialing*. A computer with four modems can dial 10,000 numbers in a matter of days. War dialers provide reports on the numbers discovered to have modems, and some can even attempt some limited automatic attacks when a modem is discovered. To identify unauthorized modems, organization should conduct war dialing at least annually. This test should be conducted after normal business hours to limit potential disruption to employees and the organization's phone system, taking into consideration the possibility that modems may be turned off after hours and might not be detected. In addition to detecting modems, war dialing can also detect faxes. The test should include all numbers that belong to an organization, except those that could be affected negatively by receiving a large number of calls (e.g., 24-hour operation centers and emergency numbers).[23]

5.4 Social Engineering

Social engineering is an attempt to trick someone into revealing information (e.g., a password) that can be used to attack systems or networks. It is used to test the human element and user awareness of security. It can reveal weaknesses in user behavior, such as failing to follow standard procedures. Social engineering can be performed through many means, including analog (e.g., conversations executed in person or over the telephone) and digital (e.g., email, instant messaging). One form of digital social engineering is known as *phishing*, where attackers attempt to steal information such as credit card numbers, social security numbers, user IDs, and passwords. Phishing uses authentic-looking emails to request information or direct users to a fake web site to collect information. Other examples of digital social engineering are crafting fraudulent emails and sending fake attachments that could mimic worm activity.

Security testers using analog social engineering as part of a penetration test typically follow one or more standard approaches. In one approach, the penetration tester poses as a user experiencing difficulty and

[22] See http://csrc.nist.gov/publications/nistpubs/ for additional information on PBX security.
[23] Most war dialing software allows the tester to exempt particular numbers from the calling list.

calls the organization's help desk to gain information on the target network or host, obtain a login ID and credentials, or get a password reset. The second approach involves posing as the help desk and calling a user to get the user to provide user IDs and passwords. Analog social engineering is also often used to gain physical access to an organization. Testers may pose as maintenance technicians, cleaning crew, high profile visitors, etc., to gain access to buildings or secured areas. Testers typically dress in disguise (for example, in maintenance uniforms) and may also have fake badges and identification.

Ideally, social engineering tactics fail, and testers are unable to gain intelligence or access. Realistically, the human element is often the weakest component of an environment. As such, social engineering testing skills include persuasion, a high likeability factor, and the ability to appeal to a user's sympathetic side. Social engineering may be used to target specific individuals or groups in the organization or may have a broad target set. Specific targets may be identified when the organization knows of an existing threat or feels that the loss of information from a person or specific group of persons could have a significant impact on the organization. Individual targeting can lead to embarrassment for those individuals, if the test team successfully elicits information or gains access. It is important that the results of social engineering are used for improving the security of the organization and not to single out individuals. Testers should produce a detailed final report that identifies both the successful and unsuccessful tactics used. This level of detail assists organizations in tailoring their security awareness training programs.

5.5 Physical Security Testing

Information security testing also includes testing physical security controls and procedures. An attacker with physical access to an organization has virtually unlimited access to its networks and systems. An attacker with physical access may be able to connect to the network, steal equipment, capture sensitive information (including installing keylogging devices), or disrupt communications. Physical security testing involves attempts to circumvent physical access controls—such as guards, gates, and card readers—using methods such as fake badges and social engineering. This type of testing evaluates the technical controls, human factors, and policies and procedures. Physical security testing should focus on evaluating the organization's ability to monitor access points with guards or electronic monitoring devices, and evaluating the methods of physical access control, such as badge readers and biometrics, to the organization and secured areas. Caution should be exercised when performing physical security testing; security guards should be made aware of how to verify the validity of the testers' activity, such as a point of contact or type of documentation.

The final report should include information on the activities conducted and the level of success associated with each activity. For instance, if the testers were able to make a fake badge that allowed them access to a facility, the report should reflect that success. Alternatively, if the testers made a fake badge but could not gain access, that too should be included in the report.

5.6 Summary

Each information security testing technique has its own strengths and weaknesses. Table 5-1 compares the testing techniques discussed in Section 5.

Table 5-1. Target Vulnerability Validation Techniques

Type of Test	Capabilities
Password Cracking	• Identifies weak passwords and password policies
Penetration Testing	• Tests security using the same methodologies and tools that attackers employ • Verifies vulnerabilities • Demonstrates how vulnerabilities can be exploited iteratively to gain greater access
Remote Access Testing	• Discovers vulnerabilities and unintended network entry points • Identifies methods that an attacker could use to circumvent perimeter security, including unauthorized and unsecured modems
Social Engineering	• Allows for testing of procedures and the human element (user awareness)
Physical Security Testing	• Evaluates physical monitoring and access controls, including the human element

There are risks associated with each technique and technique combinations. To ensure each technique is executed safely and accurately, each member of the test team should have a certain baseline skill set. Table 5-2 provides guidance on the minimum skill set for each testing technique presented in this guide.

Table 5-2. Security Testing Knowledge, Skills, and Abilities

Technique	Baseline Skill Set
Password Cracking	Knowledge of secure password composition and password storage for OSs; ability to use automated cracking tools
Penetration Testing	Extensive TCP/IP, networking, and OS knowledge; advanced knowledge of network and system vulnerabilities and exploits; knowledge of techniques to evade security detection
Remote Access Testing	TCP/IP and networking knowledge; knowledge of remote access technologies and protocols; knowledge of authentication and access control methods; general knowledge of telecommunications systems and modem and PBX operations; ability to use a variety of scanning and security testing tools, including war-dialing tools
Social Engineering	Ability to influence and persuade people; ability to remain composed under pressure
Physical Security Testing	General knowledge of physical security controls, including monitoring systems and physical access control

6. Information Security Test Planning

Proper planning is critical to a successful security test and is integral to every type of testing presented in this guide. A standard security testing process, presented in Figure 6-1, can be tailored to suit individual organizations; nevertheless, it is recommended that, at a minimum, all security tests follow the high-level steps and document their output.

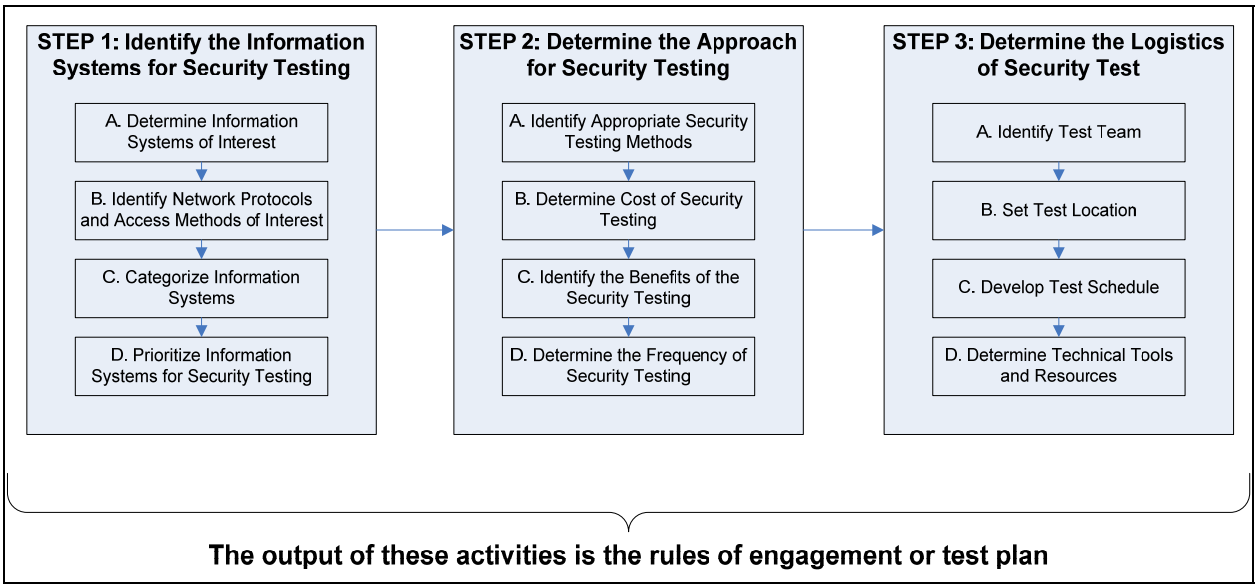

Figure 6-1. Standard Testing Process Example

This section describes each of the steps in the planning process depicted in Figure 6-1, providing guidance on how to choose the information systems to be tested, how to select the appropriate testing approach, and how to address logistical considerations. Additionally, this section provides recommendations for developing the ROE or test plan for conducting the security test. Finally, there is a brief discussion on legal considerations organizations may need to address.

6.1 Identify the Information Systems for Security Testing

Deciding which systems in an organization should undergo security testing is the first step in the planning process. Considerations that come into play include the systems and communication methods of interest, categorization of systems, and prioritization of systems. Each element is discussed below.

6.1.1 Determine Information Systems of Interest

Testing of a number of information systems and network types is required to identify their security posture. The following describes some of the key systems of interest, where a system can refer to everything from an individual information system to an enterprise network, for information security testing:

- **Enterprise Network.** An organization's enterprise network includes all systems connected to it, such as workstations, servers, routers, and perimeter defense systems (e.g., firewalls, intrusion detection and prevention systems [IDPS]). While it is important to ensure individual systems are secure, it is also important to evaluate the security posture of the enterprise as a whole.

- **Development or Test Network.** Development and test networks, whether they are connected to the Internet or self-contained, need to be maintained and evaluated from a security perspective. This helps ensure the development and test systems remain operational and secure with minimized risk.
- **Standalone Systems.** These systems are generally not connected to a network. While at less risk than systems connected to the Internet or other networks, standalone systems are still susceptible to threats such as malicious code transferred from removable media (e.g., USB drives, CDs, floppy disks).
- **Mobile Systems.** These are systems designed to maintain Internet connectivity independently of the system's parent network. Some mobile systems, such as laptops, may also be a component of the enterprise network at times. There are a number of reasons for testing mobile systems, which include determining the likelihood an adversary could exploit a remote system and verifying that mobile systems are not introducing vulnerabilities or malware into the enterprise environment.
- **Industrial Control Systems.** Some organizations rely on industrial control systems such as Supervisory Control and Data Acquisition (SCADA) systems,[24] which apply operational controls over long distances, as well as gathering and processing data. Typical uses include power transmission and distribution and pipeline systems. Since SCADA systems control critical functions, organizations should take special interest in ensuring they maintain a high security posture.
- **Interconnecting Systems.** It is common to see organizations with multiple operating divisions that are connected or multiple agencies connecting to each other to enable information sharing and collaboration. Conducting security tests of the systems and their connections determines the level of risk associated with the interconnection.

6.1.2 Identify Network Protocols and Access Methods of Interest

Systems and networks rely on various network protocols, so testing should verify proper implementation and configuration of the protocols used by an organization. Most testing focuses on traditional TCP/IP communications, including testing devices and applications that use the IPv4, TCP, and UDP protocols, as well as IPv6. Some testing also involves lower level protocols such as the Address Resolution Protocol (ARP). Many organizations are also using wireless communications, such as wireless LAN protocols (e.g., IEEE 802.11a/b/g, IEEE 802.11i) and wireless personal area network (WPAN) protocols (e.g., Bluetooth). Organizations should also identify their access methods, including Internet connection points and remote access telecommunications methods such as modems and digital subscriber line (DSL), so that they can also be subject to evaluation.

6.1.3 Categorize Information Systems

FIPS PUB 199, *Standards for Security Categorization of Federal Information and Information Systems*, provides standards for determining the security category of an organization's information systems, which can be helpful in developing a priority ranking of those systems for testing purposes.[25] FIPS PUB 199 security categories are based on the potential impact on an organization should certain events occur which jeopardize the information systems needed by the organization to accomplish its assigned mission, protect its assets, fulfill its legal responsibilities, maintain its day-to-day functions, and protect individuals. Security categories are to be used in conjunction with vulnerability and threat information in assessing the risk to an organization by operating an information system.[26] Additionally, security categories are a

---

[24] More information on SCADA can be found in NIST SP 800-82 (Second Public Draft), *Guide to Industrial Control Systems (ICS) Security*, available at http://csrc.nist.gov/publications/nistpubs/.

[25] FIPS PUB 199 is available for download from http://csrc.nist.gov/publications/fips/index.html.

[26] NIST SP 800-30, *Risk Management Guide*, provides guidance on conducting a risk assessment. See http://csrc.nist.gov/publications/nistpubs/.

variable to the input component of the conceptual framework for developing assessment procedures for NIST SP 800-53 security controls.[27]

FIPS PUB 199 defines three levels of *potential impact* on organizations or individuals should there be a breach of security (i.e., a loss of confidentiality, integrity, or availability). The FIPS PUB 199 definitions for each category are as follows:

> "The potential impact is **LOW** if the loss of confidentiality, integrity, or availability could be expected to have a **limited** adverse effect on organizational operations, organizational assets, or individuals. A limited adverse effect means that, for example, the loss of confidentiality, integrity, or availability might (i) cause a degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is noticeably reduced; (ii) result in minor damage to organizational assets; (iii) result in minor financial loss; or (iv) result in minor harm to individuals.
>
> The potential impact is **MODERATE** if the loss of confidentiality, integrity, or availability could be expected to have a **serious** adverse effect on organizational operations, organizational assets, or individuals. A serious adverse effect means that, for example, the loss of confidentiality, integrity, or availability might (i) cause a significant degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is significantly reduced; (ii) result in significant damage to organizational assets; (iii) result in significant financial loss; or (iv) result in significant harm to individuals that does not involve loss of life or serious life threatening injuries.
>
> The potential impact is **HIGH** if the loss of confidentiality, integrity, or availability could be expected to have a **severe or catastrophic** adverse effect on organizational operations, organizational assets, or individuals. A severe or catastrophic adverse effect means that, for example, the loss of confidentiality, integrity, or availability might (i) cause a severe degradation in or loss of mission capability to an extent and duration that the organization is not able to perform one or more of its primary functions; (ii) result in major damage to organizational assets; (iii) result in major financial loss; or (iv) result in severe or catastrophic harm to individuals involving loss of life or serious life threatening injuries."

6.1.4 Prioritize Information Systems for Security Testing

Prioritization of information systems for testing is based on their categorization, the expected benefit of the testing, scheduling requirements, and applicable regulations that require testing. The starting point for prioritization is evaluating their categorization and associated requirements for security testing. At this point, an evaluation of the system's impact rating (e.g., high, moderate) and security testing status (e.g., when was the testing last conducted) is necessary to determine the schedule moving forward. For instance, organizations should generally test a high impact system before a moderate impact system; however, a moderate impact system that is overdue may need to be tested before a high impact system

[27] NIST SP 800-53A (Draft), *Guide for Assessing the Security Controls in Federal Information Systems.* See http://csrc.nist.gov/publications/nistpubs/.

whose last security test is still within the acceptable timeframe. As part of continuous monitoring,[28] a number of NIST SP 800-53 security controls must also be constantly tested.[29]

It is recommended that the resources available for security testing be compared with the resources required. If there is a gap between the required and available resources, additional resources should be sought. The benefits realized from security testing should provide quantitative evidence of why more resources are required. After the funding is identified for the high priority systems, the lower priority items may be tested with less frequency and in descending order. The result of this step is a prioritized list of systems that will be tested with associated testing techniques, frequency, and required resources.

6.2 Determine the Approach for Security Testing

There are many factors to consider when determining which techniques to use. Each organization has its own objectives, risks, and challenges when performing security testing. An organization should first determine the testing methods to use (as described in Section 6.2.1). The chosen testing methods will then help determine the actual testing techniques to use. The following is a list of factors organizations should consider when determining the appropriate testing approach:

- **Objectives.** The testing methods and techniques are dependent on the testing objectives. An organization might want to focus on compliance, certification and accreditation (C&A) activities, security architecture, or network vulnerabilities.
- **Threats/Risk.** Testing efforts are influenced by the threats or risks faced by an organization. An organization needs to determine its comfort level with testing techniques that could lead to loss of availability or exposure of sensitive data.
- **Operational Considerations.** There are many operational considerations that influence the selection of testing methods and techniques. For example, current network deployments or migration, system mission, and criticality of the devices under test may drive the testing approach and the acceptable schedule for testing.
- **Budget.** Money drives many testing decisions, such as the types of tools used and the cost of staff to perform the testing. Resource limitations may preclude the use of certain techniques.
- **Time.** If an organization has a short time frame, it may choose to perform less extensive or less time and resource-intensive testing, such as performing vulnerability scanning rather than a penetration test.
- **Skills.** Skills are sometimes a limiting factor in the type and frequency of information security testing. Many organizations may not have security testers on staff with the appropriate skill set to perform certain types of testing. For example, an organization may have staff trained to perform activities such as scanning and vulnerability assessments but not specialized techniques such as penetration testing.

6.2.1 Identify Appropriate Security Testing Methods

The selection of security testing methods or approaches varies depending on the goals and requirements of the testing and the systems and communications of interest. They can range from passive, where the

[28] NIST SP 800-37, *Guide for the Security Certification and Accreditation of Federal Information Systems*, Section 3.4 provides guidance on the continuous monitoring phase of the accreditation process. See http://csrc.nist.gov/publications/nistpubs/.

[29] Continuous monitoring activities include configuration management and control of information system components, security impact analyses of changes to the system, ongoing assessment of security controls, and status reporting. NIST SP 800-53 http://csrc.nist.gov/publications/nistpubs/ provides additional guidance.

test team only reviews documentation such as policies, system security plans, and architecture diagrams, to highly intrusive, where the team exploits vulnerabilities to penetrate the target system. An organization can use a combination of these methods to achieve an in-depth evaluation of its security posture while maintaining an acceptable level of risk to its systems and networks.

Several security testing methods exist to provide organizations with levels of testing and multiple approaches to test different components within their environment. In addition to testing the technical security posture of a given system, network, or component, security testing should be used to test procedures and processes. For example, intrusion detection and incident handling procedures can be evaluated during active security testing.

While the testing techniques presented in Sections 3, 4, and 5 can be executed individually, they should be used together to complement one another to provide the organization a comprehensive picture of its security. Three examples of how an organization may combine security testing techniques are presented below. This is not an exhaustive list of how the techniques may complement one another. Organizations should evaluate each security test to determine the most appropriate combination for their requirements.

- **Example 1**. This example offers one way an organization may identify and validate the technical and nontechnical vulnerabilities in an environment: evaluating system configurations and patch levels and validating the findings by attempting to exploit some or all of the vulnerabilities.
  - **Step 1. Ruleset and Security Configuration Review.** To determine inherent weaknesses in the baseline of the network
  - **Step 2. Network Discovery and Vulnerability Scanning**. To identify all active systems and their known vulnerabilities
  - **Step 3. Penetration Test with Social Engineering and Physical Security-Testing Components.** To validate vulnerabilities in both the technical and the nontechnical elements.
- **Example 2**. This example is specifically geared toward identifying systems in the environment and evaluating weaknesses in the security architecture and system configurations.
  - **Step 1. Documentation Review.** To identify policy and procedure weaknesses and security architecture flaws
  - **Step 2**. **Ruleset and Security Configuration Review.** To identify additional risks. These findings may compound any security architecture weaknesses. For instance, a system may inherit security from other systems based on its placement in the architecture as well as be at greater risk because of its location. A system closer to the perimeter with a security configuration weakness may pose a greater risk than a system under multiple layers of security with the same configuration.
  - **Step 3**. **Wireless Scanning.** To identify any rogue wireless devices and further security architecture weaknesses
  - **Step 4**. **Network Discovery and Vulnerability Scanning.** To identify all active systems and their known vulnerabilities.
- **Example 3**. This example is twofold. First, it determines the likelihood that an external adversary could gain access to the network. Second, it tests the organization's audit capabilities, which facilitate incident handling and forensics.

- **Step 1. External Penetration Testing.** To identify the security posture from an attacker's point of view, including external network discovery, port scanning, and exploitation
- **Step 2**. **Log Review.** To determine effectiveness in capturing attack activities

6.2.2 Determine Cost of Security Testing

To facilitate determining the appropriate testing approach, the cost of conducting each test should be ascertained. The cost depends on a number of factors, which are identified and discussed below:

- **Size of the System to be Tested.** The size of the system in terms of number of components (e.g., single database, all user systems, or entire architecture) and network size (e.g., LAN or Wide Area Network [WAN], number of network locations that a tester will need to physically plug into for testing) will contribute to the time and resources required to complete the testing.
- **Complexity of the System to be Tested.** Testing a network with a heterogeneous operating system environment may be more costly than testing a network with a more homogeneous environment because it may require more diverse skill sets and tools.
- **Test Samples.** The feasibility of using a sample for testing, the sample size, and its makeup should be considered. For example, it may be much more efficient and nearly as effective to port scan sample hosts instead of all hosts on an enterprise network, especially if most hosts are managed.
- **Resources Requirements.** The resources (e.g., hardware/software, time and skills) required to conduct security testing vary for each testing technique. For example, it could take many hours for a skilled tester to review all security documentation.
- **Required Level of Human Interaction.** For instance, if the test team is required to work hand in hand with the IT staff, it may serve as a form of training for the IT staff, but it will likely increase the time necessary to complete the testing compared to the test team and IT staff working independently.

Unless the approach and techniques for testing have been predetermined by a governing entity (e.g., Certifying Authority [CA], Designated Approving Authority [DAA], or government mandate), the costs of conducting each type of test should be quantified to facilitate the decision making process. It may be advantageous to develop a high-level schedule and assign resources appropriately for each type of security testing being considered, taking into account all of the elements outlined above. It is important that individuals with experience conducting the various types of security testing are involved in determining the time and resources required to execute the testing to ensure an accurate and realistic estimate. If an organization plans to use an external test team, a request for quotes may need to be developed and released to allow external testing service providers to respond with competitive quotes.

6.2.3 Identify Benefits of Security Testing

The ability to quantify the benefits of security testing in an organization provides the IT security program a process by which to measure the success of the program and to identify areas for improvement. The following are examples of factors that should be considered in identifying the benefits of testing:

- Learning more about the organization's systems and networks
- Understanding the organization's security posture and how the risks are mitigated
- Decreasing the probability of successful intrusion or mission disruption.

6.2.4 Determine the Frequency of Security Testing

The frequency of security testing is often driven by the organization's requirements to demonstrate compliance with government regulations or policies. For example, FISMA requires periodic testing depending on risk but no less than annually. NIST SPs 800-53 and 800-53A provide organizations with recommendations regarding the frequency of conducting security testing. Since security testing provides a snapshot of systems' and networks' security posture at a given point in time, organizations may choose to set requirements for more frequent testing.

When determining the frequency of testing, organizations should identify the goals of testing, determine how the testing can be performed, and then select the appropriate testing techniques and frequencies. For example, one of an organization's testing goals might be to identify rogue devices on wired networks. This could be accomplished using one or more techniques, such as performing network discovery through passive sniffing or active scanning, or reviewing data collected by network management software, network intrusion detection sensors, or other devices that routinely monitor network activity. If these monitoring devices can generate alerts as soon as a new, potentially rogue device is observed on the network, then there may be little or no need to perform periodic testing for rogue devices because testing is effectively being performed continuously. For testing goals involving specific systems, organizations should consider the relative importance of each system (e.g., FIPS 199 impact level) when determining how frequently each system should be tested.

6.3 Determine Logistics of the Security Test

Addressing logistics for information security testing includes identifying all resources that are required for conducting testing and executing individual security tests, such as team creation and maintenance; an environment from which to test; and a testing schedule and the tools, both hardware and software. These are addressed in the subsections below.

In addition to the standard logistical requirements discussed in the subsections below, it is equally important to identify logistical requirements for each test during the planning phase. Appropriate planning in this area will facilitate a successful test. Depending on the scope and the environment, individual tests may have additional logistical requirements such as sending a visit request for an external test team, shipping equipment to a facility to enable testing, and planning for local or long distance travel. These should be addressed on a case-by-case basis during the planning process.

6.3.1 Identify Test Team

The test team conducts tests on test systems, operational systems, and networks using the methods and techniques described in this guide. Organizations should take care when putting together a test team, because a carefully vetted, skilled, and experienced team will lower the risk of conducting security tests. The test team may also have access to sensitive information regarding an organization's network architecture, security posture, and weaknesses. Some organizations may require background checks or security clearances.

A large range of technical skill sets is required to conduct testing in an effective and efficient manner while ensuring minimal risk. Regardless of the type of testing, testers should have significant security and networking knowledge, including expertise in network security, firewalls, IDSs, OSs, programming, and networking protocols (such as TCP/IP). The test team members should also be skilled in the particular types of security testing being executed. Real world experience is preferred to classroom or laboratory training to demonstrate a true capability to execute technical testing. Allowing inexperienced or untrained staff to conduct such evaluations can negatively affect an organization's systems and networks, which

may hinder its mission and damage the credibility of the security program management office and test team.

The team leader facilitates the testing process, demonstrates an understanding of the organization's environment and requirements, and eases communication between the test team and the organization's security group. The team leader should be selected based on overall technical knowledge and experience with the type of tests being executed and knowledge of the portion of the organization being tested. The team leader should have strong communication, organization, and planning skills.

The skills on the team should be balanced to provide a well-rounded view of the organization's security posture. For example, having an individual that specializes in perimeter defense is helpful to a test team; nevertheless, having a team full of people that specialize in perimeter defense is likely to be redundant unless the sole focus of testing is to determine the security posture of the perimeter. An ideal team is compiled based on the individual requirements of the tests being conducted with a diverse skill set to address all aspects of testing described in the ROE. Therefore, different tests may have different team compositions. This is especially true for security tests that include unique systems or networks. For instance, while SCADA systems may have traditional system and network components, they also have a number of unique components. A traditional security tester may not be familiar with the operations of the unique components, reducing the tester's ability to safely and adequately test the security posture of those systems. In this type of case, the test team may identify one or more subject matter experts to augment the team. In some cases, the subject matter expert may be an experienced security tester and system expert, and in others, the individual or individuals may be skilled strictly in the system being tested. These subject matter experts should be educated on the goals, objectives, approach and process of the test team. When possible, they should also be included in the planning process because they may have critical knowledge to contribute when scoping the effort and determining the approach.

Test teams need to remain abreast of new technology and the latest means by which an adversary may attack that technology. Teams involved in security testing should periodically refresh their knowledge base, reassess their methodology-updating techniques as appropriate, and update their tool kits. For example, attending technical training courses, performing hands-on testing in a test environment, or researching the latest vulnerabilities and exploits are just a few activities in which test teams should regularly engage. Teams should also perform real-world, technical hands-on tests on a regular basis to maintain and improve their skills.

IT staff are frequently cross-trained and capable of assuming the responsibilities of various positions or augmenting different functional teams. It is important to maintain separation of duties between the operations staff and the information security staff to provide an objective view of the security posture and steps necessary to strengthen security in the organization. In addition, those individuals conducting security testing should not be in a position to be influenced by functional requirements or operations during the course of the testing or the reporting period in which they make their recommendations.

6.3.2 Set Test Location

The environment in which the test team operates differs depending on the type of test. The test team can operate either onsite or offsite, with various levels of control and interaction from the organization's security group. Onsite testing is defined as testing executed at the location of the organization. Placing the test team offsite may be desired to make the test more realistic (e.g., when applying the red teaming approach). For tests such as documentation review and site surveys/interviews, the test team generally is located onsite so its members can easily conduct interviews and have access to the organization's security documentation. For external test teams, the organization's security group will need to determine the appropriate level of physical access (e.g., unrestricted, escorted). For technical tests conducted from

within the network such as security configuration reviews and vulnerability scanning, the test team should be provided network access either onsite or through an encrypted VPN tunnel from a trusted environment such as an approved test lab.

Depending on the tools used, test teams may require different levels of access to the network. Some tools require network or domain administrator privileges; if so, organizations should create new administrator accounts for use during testing. Each member of the test team should have his or her own account: administrator accounts should not be shared for any purpose. This approach allows the organization to monitor the specific accounts. Upon conclusion of the test, the accounts should be disabled or deleted.

Technical tests conducted from outside the network's perimeter can be executed following a number of scenarios; the most common are discussed here. The testers' systems can be connected directly to a perimeter device (e.g., border router), which keeps the testers within the organization's logical and physical boundaries. However, using this location does not provide a true evaluation of the organization's security posture from an adversarial viewpoint. External tests can also be executed from a test lab with an Internet connection that is independent from the network of the organization being tested and, if applicable, the organization conducting the testing (e.g., external test teams conducting the tests from their own facility).[30] Organizations conducting external tests may also choose to rent a server and independent Internet connection. These services are provided by a variety of vendors, typically for a monthly fee. If a rented server is used, the test team should securely delete the data on the system and rebuild it before conducting a security test. After testing is complete, the team should follow the guidelines in Section 7.4 for data handling upon completion of testing.

6.3.3 Develop Test Schedule

A schedule for security testing is an important element of planning. The schedule should be updated as modifications occur and disseminated to all stakeholders as appropriate. The following should be taken into consideration when developing the schedule for security testing:

- Known weaknesses in systems or networks
- System or network activities that may impact the functionality of the environment
- System or network activities that may affect the security of the environment
- Availability of resources.

6.3.4 Determine Technical Tools and Resources

The information systems built to execute the security test should meet the requirements of the type of testing and the expected tools. For example, test systems for document review should have applications installed to read the documents, track vulnerabilities, and compose reports. Information systems designed to execute security tests such as vulnerability assessments and penetration testing are more complex in terms of system requirements and software tools. Information systems for technical security tests can be servers, workstations, or laptops. Laptops are generally used by traveling test teams, and servers or workstations may be used if the team is executing the testing from a test lab or onsite location. The team may establish a network from which to execute testing, enabling an environment that supports centralized logging of activities and servers dedicated to activities requiring increased processing power.

[30] Using an independent network is particularly advantageous if red team testing is being conducted. It can make it harder for the security staff to identify the source of the activity (i.e., the IP addresses are not associated with a test team or organization). Also, it prevents an inadvertent denial of service against legitimate users, which could occur if the security staff blocked access from the testers' IP address range in response to the testing activity.

Test system requirements vary. A system that can handle the processing and memory requirements of all the tools, OSs, and virtual machines[31] (VM) should be used, otherwise the system will be more likely to crash during a test. This could cause that component of the test to need to be redone, data to be lost, and test systems to be rebuilt. Processing power and memory requirements will be driven by the tools used and the speed in which the test team wishes to process certain components. For example, password cracking generally requires increased processing power and memory. Test teams may wish to have a dedicated password-cracking server. A dedicated system will allow the team to execute other test objectives during the password-cracking process. Hard drive requirements will depend on the expected amount of data collected during a test. In the event that long-term storage of the data is required, a way to store such data (e.g., independent system or removable media) should also be identified and procured as appropriate.

The tools used by the test team will vary depending on the individual test scope, but the team should have a core set of tools it uses and keeps up to date. Depending on the engagement and organization, the team may use a combination of tools developed in-house, open source tools, and/or commercial or government off-the-shelf (GOTS) tools. Test teams should choose tools from well-established sources. Some organizations may also have tools they require or encourage teams to use; for example, an organization may purchase a license for a product that all its test teams could use. There are many freeware tools available. Appendix A lists common tools and describes each tool's purpose and how to obtain it. Organizations should take care to evaluate each tool before using it in a test. This may range from downloading the tool from a trusted site to conducting an in-depth code review of the tool to ensure that it does not contain malicious code.

Often, the tools will determine the operating system required to execute the testing, including the need for multiple OSs. To accomplish this, systems may be configured a variety of ways, including single OS, single OS with VM images, and dual boot systems. An example of a dual boot system is a system that can be booted to either a version of Microsoft Windows or a version of Linux such as Red Hat, Mandrake, or SuSE. A dual boot system allows a tester to use two OSs from a single machine, but this can be inconvenient because the tester needs to reboot the system to switch between the OS and the tools running on them.

Another more popular and functional option is to use VMs. Many testing tools require a specific operating system. VMs allow testers to use a wider variety of tools more easily because they allow testers to switch OSs without rebooting the system and to run multiple OSs simultaneously. This has several possible benefits, including logging, documentation capabilities, and executing simultaneous tests. Since the system hosting the VM is supporting two or more OSs at one time, test systems running VMs do require greater processing power and memory.

Testers should be knowledgeable, experienced, and comfortable using the OSs on the test system because system modifications are frequently required to operate specific tools or system capabilities successfully. For example, if the test team is using Red Hat Linux to conduct a wireless security test, the team will need to be familiar with installing and configuring wireless network cards because the steps for doing so may not be obvious to a Red Hat Linux novice.

Regardless of the system installation method, organizations conducting security tests should develop and maintain a baseline image from which to conduct the tests. An image provides a standardized toolkit for the team to use and enables rapid deployment of a team. The baseline image should consist of the operating system, drivers, requisite system and security configurations, applications, and tools to conduct

[31] A virtual machine is software that allows a single host to run one or more guest operating systems. The operating systems do not interact and are not aware of each other. A virtual machine monitor is the piece of software that controls communication between the physical hardware and the individual virtual machines.

testing. Full system images are often hardware dependent, so installing an image on another system with different hardware (e.g., video cards) requires the test team to modify the image, which requires specific skills and is time-consuming. VM images are more versatile and do not carry the same hardware restrictions as full system images, making them a more favorable option for test teams. Multifunction teams, such as those with the skills to conduct wireless scans, application testing, vulnerability assessments, and penetration tests, may have one image that contains the tools required to execute all test types or multiple images for various techniques. Having one image is generally preferable to multiple images because retaining multiple images necessitates more maintenance than keeping one image.

The image should be updated periodically to ensure the latest tools and tool versions are being used. During the update period, the team should confirm functionality of the tools and identify, documenting as appropriate, any changes in the tools' functionality or use. Updating tools that discover vulnerabilities (e.g., vulnerability scanners) before each tests helps ensure that recently discovered vulnerabilities are included in the testing. In addition to maintaining the existing toolset, the team should periodically assess its toolkit to identify obsolete tools that should be removed or new tools that should be added.

Before using test systems in a security test, the test team should apply the latest security patches and enable only the services required for connectivity and testing. The organization's security group may validate that test systems are compliant with the organization's security requirements and approved for testing before connecting the test systems to the network. This validation can be done using the same systems used for technical tests such as vulnerability scans. Test systems may not meet all of the organization's security requirements because of the requirements of the tools being used for testing.

Traveling teams should maintain a flyaway kit that includes the systems, images, additional tools, cables, projectors, and other equipment that a team may need when performing testing at other locations. If an organization uses an external test team, the team should not use the organization's resources unless required to do so. If the organization does not authorize external systems connecting to its network, the external test team will need to either install all required tools onto an approved client system or bring a bootable system emulation capability, such as a live CD.[32] Appendix A provides examples of two live CD distributions. If the tools are directly installed onto a client system, the test team should ensure the tools and subsequent files are removed from the system upon completion of the testing.

6.4 Development of Rules of Engagement (ROE)

ROE are detailed guidelines established before the start of an information security test that give the test team authority to conduct the technical and nontechnical activities defined in the ROE without additional permission. In some tests, specifically penetration testing engagements, the ROE will allow certain activities such as port scanning or retrieval of the password file for offline cracking, but forbid other activities that are a greater risk to the system such as installing and using executable files. The authorized point of contact may grant permission for the test team to conduct these activities as the opportunity arises. ROE provide structure and accountability to an information security test. These documents should be developed for every information security test, regardless of scope, level of intrusiveness, or party performing the test (i.e., internal and external test teams). These documents provide the rules and boundaries to which the test team must adhere throughout the security test, thus protecting the organization by reducing the risk of an incident, such as accidental system disruption or inadvertent disclosure of sensitive information. ROE also protect the test team by ensuring the organization's management understands and agrees to the test scope, activities, and limitations. Development of the

[32] A live CD is a fully functioning operating system environment that is contained on a bootable CD. This technology does not require the user to load anything (e.g., software, drivers, etc) onto the system.

ROE should be a collaborative process between the test team and key members of the organization's security group.

The ROE should answer these basic questions:

- What is the scope of the tests?
- Who is authorized to conduct the tests?
- What are the logistics of the tests?
- How should sensitive data be handled?
- What should occur in the event of an incident?

The ROE should identify which systems and networks are authorized to be tested. This can be done by providing the number of systems and the IP addresses or address ranges used by those systems. It should also list specific systems, at a minimum by IP address and preferably also by system name, that are not authorized to be tested. For example, if an organization's payroll database is deemed too mission-critical for a particular type of testing, the system name and IP address should be included in the exclusion list of the ROE.

Upon receiving the list of authorized IP addresses, external teams should verify that all public addresses (i.e., not private, unroutable addresses) are under the organization's purview. Web sites that provide domain name registration information (e.g., WHOIS) can be used to determine owners of address spaces. It is assumed that internal test teams will be able to verify the IP addresses using internal resources; nevertheless, both internal and external security test teams can benefit from confirming the public IP addresses through a third-party source. Test teams should be careful to ensure they are only testing systems within the organization's control. If the organization does not control part or all of its network, such as having organization systems housed on a third party's network, the owner of the other network usually must also consent in writing to the ROE. A similar situation involves systems that are shared by organizations, such as a system using virtual machine technology to provide services to multiple organizations. By signing the ROE, all parties acknowledge and approve of the testing.

In addition to determining which systems are authorized for testing, the ROE should also detail the type and level of testing permitted. For example, if the organization desires a vulnerability assessment, the ROE should provide information on the activities authorized to be performed on the target network, such as port and service identification, vulnerability scanning, security configuration review, and password cracking. Sufficient detail should be included to describe the type of testing, approach, and tools. For example, if password cracking will be used, the method in which the passwords will be obtained (e.g., sniffed off the network or copied from the OS password file) should be included in the ROE. The ROE should also explicitly state any activities that are prohibited—for example, file creation and modification. Ideally, there should be no room for interpretation. If questions regarding scope and level of authorization arise during the course of testing, the team and the organization's identified point of contact should meet to discuss them.

The ROE should address the logistical details of the engagement as well. This includes hours of operations for the test team; clearance or background check level needed for the team members; a call plan with accurate contact information for the test team, network and security operations centers, and the organization's main point of contact for the testing; the physical location from which testing will take place; and the equipment and tools to be used to conduct the tests. Any requirements to inform parent organizations, law enforcement, and a computer incident response team (CIRT) should be identified in the

ROE. In addition, the person responsible for informing the organizations of the pending security test should be identified. For red team or other unannounced testing, the ROE should also define a plan for how test activity detected and reported by the organization's security staff, CIRT, and others should be handled, such as what escalation processes should be followed. The primary purpose for this plan is to ensure that test activity does not trigger reporting of security breaches to external parties, such as external incident response teams.

The IP addresses of the machines from which testing will be conducted should be identified in the ROE so that administrators can differentiate testing activities, such as penetration testing attacks, from actual malicious attacks. (If appropriate for the goals of the test, security administrators can configure intrusion detection systems and other security monitoring devices to ignore activity generated by the IP addresses during testing.)

Data handling requirements should be addressed in the ROE. Specifically, the following should be addressed:

- Storage of organizational data during the test on the test systems, including physical security of the systems as well as passwords for the system and encryption of the data itself
- Storage of the data upon conclusion of the test for long-term storage requirements or vulnerability tracking
- Transmission of the data during or after the test such as across internal or external networks (e.g., the Internet)
- Removal of the data from systems upon conclusion of testing, in particular for third-party tests including references to any specific requirements set forth by the governing organization's policies or procedures.

Finally, the ROE should provide explicit guidance on incident handling in the event the team causes an incident or uncovers an incident during the course of the engagement. The incident handling section in the ROE should define the term *incident* for the organization being tested and provide the threshold for determining if an incident has occurred. The ROE should identify specific primary and alternate points of contact on the test team, generally the team leader and assistant team leader, and the organization's security group. Guidelines should be included in the ROE that clearly delineate the actions both the test team and the organization's security group should execute upon determination that an incident has occurred. For example, if the team discovers an actual intruder or an intruder's footprints within the network, should the team cease testing? If so, when can the team recommence and by whose authority? The ROE should provide explicit instructions on what actions the test team should take in these situations.

Development of and adherence to the ROE are the most critical components of the testing process, as they provide the guidelines, authority, and accountability to ensure a successful security test. Appendix B provides a sample template for the ROE.

For some organizations, the ROE will suffice as a test plan, since it contains much if not all of the information contained in conventional security test plans. Other organizations may require either a separate test plan or a test plan in lieu of ROE. NIST SP 800-53 provides additional information regarding test plans and addresses six distinct steps that testers should consider in developing a security test plan. These steps are: (i) establishing which security controls and control enhancements are to be tested; (ii) selecting the appropriate procedures (and procedural steps) to be used during the testing of the selected security controls and control enhancements; (iii) developing additional test procedures (and procedural statements), if necessary, to address security controls and control enhancements that are not

contained in NIST SP 800-53 or to provide additional testing of security control effectiveness; (iv) optimizing the selected procedures (and procedural steps) to minimize duplication of effort and provide cost-effective test solutions; (v) obtaining results from previous tests and determining the applicability and usefulness of the results; and (vi) finalizing the plan and obtaining the necessary approvals to execute the plan.

In addition to a complete ROE or approved test plan, it may be useful to develop a shorter document (one- or two-page memorandum) the test team can present to parties in the organization (e.g., users or system owners) as authorization to interview them or gain access to a particular system. The document should describe the allowable and unallowable activities, authorized and unauthorized systems, acceptable level of cooperation to be provided by users, and a point of contact in the organization's security group the user can contact for additional information.

6.5 Address Legal Considerations

An evaluation of the potential legal concerns should be addressed before commencing a security test. If an organization authorizes an external entity to conduct a security test, the legal departments of each organization (the organization being tested and the organization conducting the test) may be involved. Legal departments may assist in review of the ROE and provide indemnity or limitation of liability clauses into contracts governing security tests and in particular, those test types deemed intrusive. The legal department may also require external entities to sign nondisclosure agreements to prohibit the test team from disclosing any sensitive, proprietary, or other restricted information to unapproved entities.

The legal department should also address any privacy concerns the organization may have. Most entities have warning banners or signed user agreements that disclose that systems are monitored, and that by using the system, each individual consents to monitoring. Not all organizations have these in place, and the legal department should address potential privacy violations before a test. In addition, captured data may include proprietary data that does not belong to the organization or personal employee data, which may cause privacy concerns. Security personnel should be aware of these risks and conduct packet captures following any requirements set forth by the legal department. The legal department may also determine the data handling requirements to ensure the confidentiality of the data (e.g., vulnerabilities).

Involvement of the legal departments is at the discretion of the organizations; it is recommended that they always be involved for intrusive tests such as penetration testing or red teaming.

6.6 Summary

Information security testing is a complex activity because of organizational requirements, the number and type of systems in the organization, the testing approaches and techniques, and the logistics associated with the testing. Security testing can be simplified and the associated risk reduced through an established, repeatable planning process. Accurate and timely planning of a security test can also ensure all factors pertinent to the success of the test are taken into account.

The core activities involved in planning for an information security test are:

- **Identifying the systems to be tested.** This includes determining the information systems of interest, identifying the network protocols and access methods of interest, and categorizing the information systems (i.e., FIPS 199 categories). The organization can then prioritize the information systems for security testing.

- **Determining the approach for testing the systems.** Organizations need to consider many factors, including the testing objectives, threats against the organization, operational considerations, budget, time, and tester skills. In determining the approach, the organization should identify the appropriate security testing methods, determine the cost of the testing, identify the benefits of the testing, and determine the frequency of security testing.
- **Determining the logistics of the security test.** This includes identifying all required resources, including the test team; selecting the environments and locations from which to test; developing a testing schedule; and acquiring the necessary technical tools.
- **Developing the rules of engagement (ROE).** The ROE are detailed guidelines that provide the rules and boundaries to which the test team must adhere throughout the testing. ROE should be developed for every technical security test. The ROE should identify the systems and networks to be tested, the type and level of testing permitted, logistical details, data handling requirements, and guidance on incident handling.
- **Addressing any legal considerations.** Organizations should evaluate potential legal concerns before commencing a security test, particularly if an external entity is performing the test. Legal departments may review the ROE, address privacy concerns, and perform other functions in support of test planning.

7. Security Testing Execution

During the security testing execution phase, vulnerabilities are identified by the methods and techniques decided upon during the planning phase and identified in the ROE. It is critical for the test team to conduct the security testing in accordance with the signed ROE. The purpose of this section is to highlight some key considerations for the test team throughout the execution phase.

This section discusses proper coordination throughout the security test, which facilitates the testing process and reduces the risk associated with the security tests. Key considerations such as incident handling and challenges organizations face when conducting security testing are presented below. The section also discusses the analysis process and provides recommendations for the collection, storage, transmission, and destruction of data associated with the security test.

7.1 Coordination

Throughout a security test, it is critical to coordinate with various entities in the organization. Coordination requirements are determined by the ROE and should be followed accordingly. Coordination will help ensure—

- Stakeholders are aware of testing schedule, activities, and potential impact the testing may have on the system
- Testing does not occur during upgrades, new technology integration, or other times where the security of the system is being altered (e.g., testing occurs during maintenance windows or periods of low utilization)
- The test team is provided with the required level of access to the facility and systems as appropriate
- Appropriate personnel such as the CIO, CISO, ISSO, or ISSPM are informed of any critical high-impact vulnerabilities as they are discovered
- In the event of an incident, the appropriate individuals are informed (e.g., test team, incident response team, senior management). At that time, it is recommended that activities cease until the incident is addressed and the test team is given approval to recommence testing in accordance with the incident handling section of the ROE. The extent to which test activities should be suspended varies based on the organization and the type of incident, but in many cases the only activities that are suspended are those involving the systems directly involved in the incident.

The level of coordination will be driven primarily by the system and the type of testing. Generally, critical systems require more coordination than noncritical systems to ensure system availability throughout the engagement. Testing techniques have varying levels of risk to the target system during execution. Those techniques that fall in the review category have minimal risk; target identification and analysis category have moderate risk; and a high risk is associated with the target vulnerability validation category. For instance, a critical system undergoing penetration testing generally requires more coordination than a document review of a critical system or a penetration test of a noncritical system. Nevertheless, organizations may encounter circumstances where the reverse is true. In these cases, the level of coordination should be commensurate with the requirements and organizational considerations. Because of this, the test team and other stakeholders such as the system owners should remain vigilant during the execution of the test. The level of access required by the test team will also drive coordination to ensure the team has appropriate physical and system access (e.g., when testing the insider threat).

The test team should be proactive in its communication with the appropriate parties in the organization. Communication can be maintained through periodic status meetings and daily or weekly reports. The attendees of meetings and recipients of reports should be identified in the ROE and may include the test team, ISSPM, ISSO, CISO and the CIO. The frequency of status meetings and reports will be driven by the length and complexity of the security test. For example, for a one-month penetration test, status meetings may be held weekly with daily reports provided during the active testing phase (i.e., the period during which systems are being exploited). Meetings and reports should address the activities to date, rate of success, problems encountered, and critical findings and recommended remediation.

7.2 Testing

As discussed in Section 6.4, the ROE provide the guidelines for conducting the test. The ROE should be followed unless explicit permission to deviate from them is provided, generally in writing, by the original signatory or individual higher in command. It is prudent to ensure all testers read and understand the ROE. Especially for those activities in the target vulnerability validation category, it is recommended that the ROE be reviewed by the test team periodically throughout the engagement.

During security testing, the incident response team of the organization may detect an incident. Two potential causes for the incident are the test team's actions or a true adversary that happens to perform an attack at the same time that testing is occurring. The incident response team or individual that discovers the incident should follow normal escalation procedures. The test team should follow the guidelines set forth by the ROE unless otherwise informed. The test team may also uncover an adversary's presence in the network during the security test. This should be immediately reported to the appropriate individual, and the test team should follow the protocol identified in the ROE. It is recommended that the test team cease testing the systems involved in the incident while the organization responds to the incident.

In addition to encountering a new incident or uncovering an existing incident, the test team may face other technical, operational, and political challenges during security testing, such as the following:

- **Resistance.** There is often resistance to testing from many parties in an organization, including system and network administrators and end users. Reasons for this resistance include fear of losing system or network availability, fear of being reprimanded, inconvenience, and resistance to change. Having upper management approval and support will help resolve problems related to resistance. Incorporating security testing into the organization's overall security policy will also establish a process that does not surprise administrators and users.

- **Lack of Realism.** In preparation for a security test, users and administrators sometimes modify settings to make their systems more secure, more resistant to attack, or bring them into compliance. While these types of changes are generally viewed as positive, changes made under these circumstances are generally only maintained through the duration of the security test. Upon conclusion of the test, the systems are returned to their previous configurations. Not providing notice in advance of all testing to users and administrators helps to address this challenge. Many organizations choose to perform unannounced testing occasionally to supplement announced testing.

- **Time.** Often security testing is incorporated into development or deployment with little notice and narrow timeframes; making security testing a regular part of the development or deployment cycle will allow for adequate testing throughout the project. Time is also a challenge when testing critical systems and networks that are in production. These often need to be tested off-hours in case of accidental loss of availability or other problems. While security testing could continue indefinitely, the test team is often restricted to testing timeframes, while true attackers have no such constraints.

- **Resources.** Security testing faces the continual challenge of obtaining and maintaining adequate resources (e.g., a skilled test team and up-to-date hardware and software). It is suggested that organizations designate security testing equipment, such as laptops and wireless cards, to be used solely for testing.[33] If commercial testing software is used, consider purchasing continuous licenses and support contracts. Testers should schedule time before the testing begins to ensure that all testing software is properly patched and up-to-date. If an in-house test team is not available or desired, it may be a challenge to find dependable and trustworthy external testers. Organizations should seek a firm with an established methodology, proven processes, comparable and sufficient past performance, and experienced personnel. If an organization is using an in-house testing team, continue to recruit and train skilled testers. Offer other challenging opportunities in the organization for testers to become involved to avoid tester burn out.

- **Evolving Technology.** Security testers will need to stay up to date on current tools and testing techniques. Budget for annual training classes and conferences where security testers can update and refresh their skills.

- **Operational Impact.** Although security testing is planned to prevent or limit operational impact, there is always a chance of accidental or unexpected complications. Every test by every tester should be recorded with the timestamp, type of test, tool used, commands used, IP address of testing equipment, etc. It is recommended that a logging script be used to capture all commands and keystrokes during testing. There are both terminal and GUI tools that can record a tester's actions. This type of recording can also assist in handling accusations of security testing impacting operations and system performance. Because of the risk of operational impact, it is recommended to have an established incident response plan in place during testing.

7.3 Analysis

Although some analysis may be performed after testing has been completed (see Section 8.1), most analysis occurs during testing. The primary goals in conducting analysis are to identify false positives, categorize vulnerabilities, and determine the cause of the vulnerabilities.

Security tests conducted with automated tools can produce a significant number of findings, but these findings often need to be validated to identify false positives. The team may validate vulnerabilities by manually examining the vulnerable system or by using a second automated vulnerability assessment tool and comparing the results. Using a second tool is generally fast, but it often produces similar results to the first tool, including the same false positives. Manual examination typically provides more accurate results than using two tools, but manual examination is also potentially quite time-consuming.

Organizations may choose to categorize findings based on the security controls and control families in NIST SP 800-53, which organizes controls into families such as incident response and access control. This categorization may facilitate vulnerability analysis, remediation, and documentation.

While individual vulnerabilities need to be identified and resolved, identifying the root cause of the vulnerabilities is key to improving the organization's overall security posture because a root cause can often be traced to program-level weaknesses. The following is a list of common root causes:

- Insufficient patch management, such as failing to apply patches in a timely fashion or failing to apply patches to all vulnerable systems

[33] Organizations may want to disconnect dedicated test equipment from networks when testing is not occurring.

- Insufficient threat management, such as outdated antivirus signatures, ineffective spam filtering, and firewall rulesets that do not enforce the organization's security policy
- Lack of security baselines, such as inconsistent security configuration settings on similar systems
- Poor integration of security into the system development life cycle, such as missing or unsatisfied security requirements and vulnerabilities in organization-written application code
- Security architecture weaknesses, such as security technologies not being properly integrated into the infrastructure (e.g., poor placement, insufficient coverage, or outdated technologies), or poor placement of systems facilitating their risk of compromise
- Inadequate incident response procedures, such as delayed responses to penetration testing activities
- Inadequate training, both for end users (e.g., failure to recognize social engineering and phishing attacks, deployment of rogue wireless access points) and for network and system administrators (e.g., deployment of weakly secured systems, poor security maintenance)
- Lack of security policies or policy enforcement, such as open ports, active services, unsecured protocols, rogue hosts, and weak passwords.

A useful resource to reference throughout the analysis phase is the NIST National Vulnerability Database (NVD)[34]. NVD is a database that contains information on Common Vulnerabilities and Exposures (CVE), a list of standardized names for known vulnerabilities. The NVD scores vulnerabilities with the Common Vulnerability Scoring System (CVSS) and provides additional information regarding the vulnerability and additional resources to reference for mitigation recommendations (e.g., vendor web sites).

Another goal of analysis is to identify throughout the testing any critical vulnerabilities that the organization needs to address immediately. For instance, if penetration testing exploits a vulnerability that allows the tester to gain administrator rights on a critical system, the test team should notify the person identified in the ROE immediately.

7.4 Data Handling

The method by which an organization's data is handled throughout the security test is critical to ensuring protection of sensitive information, including system architecture, security configurations, and system vulnerabilities. Organizations should ensure proper documentation of the requirements for data handling in the ROE and adhere to their governing policies regarding the handling of system vulnerabilities. This section offers suggested methods for collecting, storing, and transmitting information security test data during an engagement as well as for storing and destroying data upon completion of a security test.

7.4.1 Data Collection

Throughout the security test, the team should collect information relevant to the test. This includes information related to the architecture and configuration of the networks being tested and information regarding the activities of the test team. This data is sensitive, and it is important to handle it appropriately. The following is a description of types of information the test team might collect:

- **Architecture and Configuration Data.** The type of security testing and desired outcome will drive the data collected by the team, which may include but not be limited to system names, IP addresses, OS, physical and logical positions in the network, security configurations, and vulnerabilities.

[34] The NVD website is http://nvd.nist.gov/.

- **Test Team Activities.** The team should keep a log including its test system's information and a step-by-step log of its activities. This provides an audit trail, allowing the organization to distinguish between the actions of the test team and true adversaries. The log of activities can also be useful in developing the security test results report.

The easiest and most effective way to maintain a step-by-step log is to install a keystroke logger on the tester's system or systems that are used in conducting the test.[35] Alternately, for automated tools, the team can maintain the audit logs from each of the tools used. The test team may choose to dump the output of the keystroke logger or tool audit log onto a separate system for a centralized storage and auditing capability. An alternate manual approach is an activities log that tracks each command executed on the network by the test team. This approach is time-consuming for the test team and leaves room for error. If an activities log is used, the log should include, at a minimum, the following information: date and time, tester's name, test system identifier (i.e., IP or MAC), target system identifier (i.e., IP or MAC), tool used, command executed, and comments.

7.4.2 Data Storage

Secure storage of the data collected during the test, including vulnerabilities, analysis results, and mitigation recommendations, is the responsibility of the security test team. Inappropriate release of this information can damage the organization's reputation and increase the likelihood of exploit. At a minimum, the team should store the following information to be used for identifying, analyzing, and reporting on the security posture of the organization and to provide an audit trail of the testing activities:

- ROE
- Test plans
- Results from automated tools
- Documentation on the system security configuration and network architecture
- Findings from interviews, site surveys, etc.
- Test results report
- Corrective action plan or POA&M.

There are a number of options for storing information on discovered vulnerabilities, such as keeping the findings in the format output by the tool used or importing the findings into a database.[36] Most vulnerability scanning tools have report formats that list the system, vulnerabilities and recommended mitigation techniques. This may be an acceptable approach if the test is small in scope (e.g., only uses one tool). For more in-depth security tests, larger organizations, or for tests that use multiple tools or approaches, a more robust and collaborative storage method such as a spreadsheet or database can be developed to store vulnerabilities. Although functionality is limited, a spreadsheet may be appropriate for individual tests, as it is easy to use, generally quick to develop, and a number of tools will output findings into a format able to be incorporated by a spreadsheet. For complex tests where there are multiple testing

[35] A keystroke logger records every keystroke the user of the system makes and puts it in a log. This level of recording provides the assessment team a method to track each action on the network. This capability allows the organization being assessed to see exactly what the assessment team executed on the network, when it occurred, and which system conducted the test. In addition, it provides the assessment team with documentation that a system on the network malfunctioning or being compromised was not caused by the test team.

[36] Storing vulnerability information can also be helpful for performing historical comparisons.

approaches, tests that will recur regularly, or the need to correlate data easily, developing a database may be beneficial.

Organizations should ensure the secure storage of all sensitive testing data such as the ROE, raw vulnerability data, and test report. In the hands of an adversary, information regarding the network architecture, system configuration, security controls, and specific system vulnerabilities would provide a blueprint and roadmap to exploiting the organization's information systems. Organizations may choose to store this data on removable media or an information system to be accessed as needed. The removable media or system designed to store this information should be isolated physically or logically from the day-to-day network resources. Access to this system and the information contained within should be limited to those individuals that require access to fulfill their roles and responsibilities. In addition, it is recommended that the data be encrypted in compliance with FIPS 140-2 to ensure that the data remains secure.

The retention requirements for security testing data vary and may not be explicitly stated for an organization. Maintaining accurate records for an information security test provides organizations with an audit trail of their vulnerabilities and the remediation actions taken to mitigate the identified risks. An audit trail maintained over time may allow organizations to evaluate the effectiveness of their information security program by conducting trend analysis of metrics involving the types of vulnerabilities, frequency of occurrence, mean time to remediation, etc.

Security testing systems, such as servers, laptops, or other mobile devices, should not be left unattended when storing sensitive data without the proper physical and logical security safeguards in place. For example, mobile systems should not be left in unlocked vehicles or in plain sight in locked vehicles, and mobile devices in hotel rooms should be secured by a cable lock, stored in a room safe, or physically secured by other means. In addition to physical safeguards, the test team should also ensure the system is configured in a way that deters adversaries from compromising the system. The test team should take appropriate measures to ensure the integrity and confidentiality of the data contained in the system. The system should be protected at a minimum with a strong password, and it is suggested that organizations consider using two-factor authentication.[37] In addition, all sensitive data on the system should be encrypted,[38] and an authentication mechanism separate from the system authentication should be used to restrict access to the encrypted information. The test team should query the organization undergoing the test to determine if there is an established policy for storing sensitive data, and if such a policy exists, the team should comply with it. If a local policy does not exist, organizations may encrypt individual files or folders on the system or encrypt the entire hard disk. Encrypting the entire disk ensures that all potentially sensitive data is encrypted, whereas addressing individual files or folders increases the risk that sensitive data may be overlooked.

7.4.3 Data Transmission

It may be necessary to transmit test data, such as system configurations and vulnerabilities, over the network or Internet. It is important to ensure the security of the data being transmitted to protect it from compromise. The ROE should address the requirements of and process for transmitting sensitive system information across the network or Internet. Secure data transmission methods include encrypting individual files containing sensitive information, encrypting communication channels (e.g., using VPNs, using the Secure Sockets Layer [SSL] protocol), and providing information through delivered or mailed hard or soft copies.

---

[37] Two-factor authentication provides additional security by requiring two of the following three factors: something you know (e.g., password), something you have (e.g., security token), and something you are (e.g., retinal scan).

[38] Such data should be encrypted in compliance with FIPS 140-2 to ensure that it remains secure.

7.4.4 Data Destruction

When test data is no longer needed, the test systems, hard copy documentation, and media should be appropriately sanitized. NIST SP 800-88, *Guidelines for Media Sanitization*[39] divides media sanitization into four categories:

- **Disposal.** Disposal is the act of discarding media with no other sanitization considerations. This is most often done by paper recycling containing nonconfidential information but may also include other media.
- **Clearing.** Clearing is a level of media sanitization that would protect the confidentiality of information against a robust keyboard attack. Simple deletion of items would not suffice for clearing. Clearing must not allow information to be retrieved by data, disk, or file recovery utilities. It must be resistant to keystroke recovery attempts executed from standard input devices and from data scavenging tools. Overwriting is an example of an acceptable method for clearing media.
- **Purging.** Purging is a media sanitization process that protects the confidentiality of information against a laboratory attack.[40] For some media, clearing media would not suffice for purging. Executing the firmware Secure Erase command (for Advanced Technology Attachment [ATA] drives only) and degaussing[41] are examples of acceptable methods for purging.
- **Destruction.** Physical obliteration of media such that it is no longer usable for its intended purpose and the data it contains is no longer retrievable. Accomplishing physical destruction is possible using a variety of methods, including disintegration, incineration, pulverizing, shredding, and melting.

Organizations should maintain a policy on the sanitization requirements for systems that conduct information security testing. NIST SP 800-88 presents a decision-flow diagram to assist organizations in determining which sanitization method is most applicable for the circumstances.

Third-party test teams should ensure they understand the organization's requirements for sanitization as the policy may differ from organization to organization and possibly among divisions in the same organization. Generally, the test team no longer requires access to test data once the report is submitted. A qualified individual from the organization undergoing the security test should verify that appropriate sanitization measures have been followed.

---

39 NIST SP 800-88 is available at http://csrc.nist.gov/publications/nistpubs/.

40 A laboratory attack would involve an attacker with the resources and knowledge to use nonstandard systems to conduct data recovery attempts on media outside their normal operating environment. This type of attack involves using signal processing equipment and specially trained personnel.

41 Degaussing is exposing the magnetic media to a strong magnetic field to disrupt the recorded magnetic domains.

8. Post-Testing Activities

After the execution phase, which produces findings in terms of vulnerabilities, the organization should take steps to address those vulnerabilities. This section presents ways that organizations can translate findings into actions to improve security. First, final analysis of the findings should be performed and mitigation actions developed. Second, a report should be developed that presents the recommendations. Last, mitigation activities should be performed.

8.1 Mitigation Recommendations

As described in Section 7.3, most analysis occurs during the testing process. Final analysis, such as the development of overall conclusions, usually occurs after all individual testing activities have been completed. The final analysis also involves the development of mitigation recommendations. While identifying and categorizing vulnerabilities is important, a security test is much more valuable if it also causes a mitigation strategy to be developed and implemented. Mitigation recommendations, including the outcome of the root cause analysis, should be developed for each finding. There may be both technical recommendations (e.g., applying a particular patch) and nontechnical recommendations that address the organization's processes (e.g., updating the patch management process). Examples of mitigation actions are policy, process, and procedure modifications; security architecture changes; deployment of new security technologies; and deployment of OS and application patches.

NIST SP 800-53 suggests mitigation recommendations for each security control. Organizations should compare potential mitigation actions against operational requirements to find the actions that best balance functionality and security. Section 8.3 discusses the implementation of mitigation recommendations.

8.2 Reporting

Upon completion of analysis, a report should be generated that identifies system, network, and organization vulnerabilities and the recommended mitigation actions. Security testing results can be used in the following ways:

- As a reference point for corrective action
- In defining mitigation activities to address identified vulnerabilities
- As a benchmark for tracking an organization's progress in meeting security requirements
- To assess the implementation status of system security requirements
- To conduct cost-benefit analysis for improvements to system security
- To enhance other life cycle activities, such as risk assessments, C&A, and process improvement efforts
- To meet reporting requirements such as those of FISMA.

Security testing results should be documented and made available to the appropriate staff, which may include the CIO, CISO, ISSO, and ISSPM as well as the appropriate program managers or system owners. Because the report may have multiple audiences, multiple report formats may be required to ensure all audiences are appropriately addressed. For example, organizations developing reports for FISMA compliance need to address FISMA requirements such as reporting on findings from evaluations, compliance with NIST standards, significant deficiencies, and planned remediation activities. Reports that will remain within the organization can be tailored for the appropriate audiences, such as program

management, information management, security engineers, configuration management, or technical staff. Internal reports should include the test methodology, test results, analysis, and POA&M.[42] A POA&M will ensure that individual vulnerabilities are addressed with specific, measurable, attainable, realistic, and tangible actions.

8.3 Remediation/Mitigation

The POA&M provides a program management office with the details and required actions to appropriately and acceptably mitigate risk. To complement the POA&M, organizations may consider developing a strategy or process for implementing it. Organizations should follow at a minimum the four steps outlined below during their remediation implementation, which will provide consistency and structure to the security personnel and program managers.

The first step in the process is testing the remediation recommendation. Before implementing technical modifications to a production asset, testing should be done on test systems in an environment that replicates the network in which the mitigation action would be implemented. For example, before being pushed to the enterprise, patches should be installed on comparable systems in the test environment to determine if there are any negative implications. Such testing significantly reduces, but does not eliminate, the risk of a system reacting adversely to a technical modification.

Second, the POA&M should be coordinated through the organization's configuration control or configuration management board because the POA&M likely proposes changes to existing systems, networks, policy, or processes. Communicating POA&M changes before deployment and upon completion ensures that the appropriate individuals are aware of the pending changes and their impact on the environment, mission, and operations. At a minimum, the program manager or system owner should be contacted before executing any POA&M actions. The program manager or other appropriate individual should provide approval of the planned mitigation actions before their implementation.

Obtaining management approval can be challenging. It may be beneficial to identify the need for (i.e., policy- or technology-driven) and the positive impact that will be realized with the mitigation action (i.e., increased security posture or compliance). A cost-benefit analysis may also provide managers with a quantitative analysis of the increased savings realized upon implementation of POA&M items. Additional benefits that may be communicated to senior management include decreased exposure, increased control of assets, decreased vulnerabilities, proactive approach to security, and maintenance of compliance.

Third, the mitigation actions are implemented and verified to ensure appropriate and accurate implementation. Verification can take place by conducting an audit of the system, retesting the system and its components, and holding personnel accountable through documentation. A system audit provides technical verification of the changes implemented on the system. The audit can be conducted by onsite security personnel or an external security test team. The audit team may use the mitigation strategy as a checklist for ensuring each of the actions is accomplished. Retesting the system will validate the mitigation actions have been completed. It is important to note that the test team will be able to verify its implementation only if a mirror copy of the original test is performed. As technology evolves, additional vulnerabilities may be uncovered during subsequent security tests. An organization may also choose to verify the implementation of the mitigation strategy through nontechnical means such as documentation. For example, it may be appropriate and cost-effective to hold the security personnel responsible for

[42] NIST SP 800-37 notes that a POA&M "describes the measures that have been implemented or planned: (i) to correct any deficiencies noted during the assessment of the security controls; and (ii) to reduce or eliminate known vulnerabilities in the information system. The plan of actions and milestones document identifies: (i) the tasks needing to be accomplished; (ii) the resources required to accomplish the elements of the plan; (iii) any milestones in meeting the tasks; and (iv) scheduled completion dates for the milestones."

implementing the mitigation strategy accountable by requesting that they sign a document describing all of the accomplished actions. While this method is more cost-effective in the short term for an organization, there are risks with not technically verifying that changes have been implemented.

Last, as part of the implementation strategy, it is important to continuously update POA&Ms to identify those activities that are accomplished, partially accomplished, or pending another individual's or system's action. Ensuring that the POA&M is integrated into the organization's configuration management process will facilitate centralized tracking and management of changes to systems, policy, processes, and procedures, as well as provide a mechanism for oversight to address compliance requirements.

Appendix A—Live CD Distributions for Security Testing

Live distribution CDs focused on security testing are available to the public for free. These distributions provide security testers with a live distribution OS containing tools for security testing. The OS distribution is loaded onto a CD-ROM, USB drive, or other peripheral device. It is not installed onto a system; instead, it is run directly from the device on which it is loaded. Hence, it is termed a "live" distribution. Two such distributions are BackTrack and Knoppix.

BackTrack[43] has a collection of over 300 security tools for network discovery, scanning and sniffing, password cracking, remote access testing, Bluetooth testing, computer forensics, and penetration testing. BackTrack features user modularity, which means the user can customize the distribution to include personal scripts or additional tools. It has tools to analyze Voice over Internet (VoIP) protocols, such as the Session Initiation Protocol (SIP), that are not found in other security-focused live distribution operating environments. In its unique toolset are tools such as Cisco Global Exploiter (CGE) and Cisco Torch that specifically target Cisco systems, and a vulnerability assessment tool, Metasploit. Recognizing the growing importance of application security testing, there are tools such as Peach, Fuzzer, and the java tool, Paros Proxy. Table A-1 provides a sample of the tools available on the BackTrack distribution.

Table A-1. BackTrack Toolkit Sample

Security Testing Technique	Security Testing Tool
Review	
Network Sniffing	Dsniff, Ettercap, Filesnarf, Kismet, Mailsnarf, Msgsnarf, Ntop, Phoss, SinFP, SMB Sniffer, Sshow, and Wireshark
File Integrity Checking	Autopsy, Foremost, RootkitHunter, and Sleuthkit
Target Identification and Analysis	
Application Security Testing	CIRT Fuzzer, Fuzzer 1.2, NetSed, Paros Proxy, and Peach
Network Discovery	Autonomous System Scanner, Ettercap, Firewalk, Netdiscover, Netenum, Netmask, Nmap, P0f, Tctrace, and Umit
Network Port and Service Identification	Amap, AutoScan, Netdiscover, Nmap, P0f, Umit, and UnicomScan
Vulnerability Scanning	Firewalk, GFI LANguard, Hydra, Metasploit, Nmap, Paros Proxy, Snort, and SuperScan
Wireless Scanning	Airsnarf, Airsnort, BdAddr, Bluesnarfer, Btscanner, FakeAP, GFI LANguard, Kismet, and WifiTAP
Target Vulnerability Validation	
Password Cracking	Hydra, John the Ripper, RainbowCrack, Rcrack, SIP Crack, SIP Dump, TFTP-Brute, THC PPTP, VNCrack, and Webcrack
Remote Access Testing	HTTPD/Apache Server, IKE-Probe, IKE-Scan, PSK-Crack, TFTPD, VNC auth Scanner, and VNC Server
Penetration Testing	Driftnet, Dsniff, Ettercap, Kismet, Metasploit, Nmap, Ntop, SinFP, SMB Sniffer, and Wireshark

[43] BackTrack is derived from two separate Linux live security-based distributions, WHAX and the Auditor Security Collection. Both were popular for their abundance of security tools and ease of use. Shortly after the creators of each distribution began to collaborate, they released the first non-beta version, renamed BackTrack, in May 2006. BackTrack quickly became and remains a favorite tool set among security professionals. In October 2006, an updated public beta edition of BackTrack 2.0, containing more tools, was released. The following month, a new beta version 2.6.18.3 was released. BackTrack 2.6.18.3 is the version referred to for this publication.

An older Linux live OS distribution and open source security toolset is Knoppix Security Tools Distribution (STD), which is based on Knoppix Linux. It was created by a security professional to assist with teaching security techniques to others. Knoppix STD was first released in May 2004 as Knoppix-STD 0.1 and has not been updated since. The lack of a newer version is due to its creator leaving the project. Version 0.1 is the version referred to for this publication. Before BackTrack, Knoppix STD was the benchmark security toolset and it remains widely used.

Similar to BackTrack, Knoppix STD enables network discovery, port and service identification, network sniffing, password cracking, forensics, and remote access testing. While there is some overlap between the distributions, there are some differences as well. Knoppix contains some tools that BackTrack does not such as Netcat and Nessus. In addition, it addresses technology areas such as cryptography and offers more tools for computer forensics and sniffing. It does not provide Metasploit and in comparison to BackTrack is weak on wireless security tools. Table A-2 provides a sample of the tools available on the Knoppix STD distribution.

Table A-2. Knoppix STD Toolkit Sample

Security Testing Technique	Security Testing Tool
Review	
Network Sniffing	Dsniff, Ettercap, Ethereal, Filesnarf, Kismet, Mailsnarf, Msgsnarf, Ngrep, Ntop, TCPdump, and Webspy
File Integrity Checking	Autopsy, Biew, Bsed, Coreography, Foremost, Hashdig, Rifiuti, and Sleuthkit
Target Identification and Analysis	
Application Security Testing	NetSed
Network Discovery	Cryptcat, Ettercap, Firewalk, Netcat, Nmap, and P0f
Network Port and Service Identification	Amap, Netcat, Nmap, and P0f
Vulnerability Scanning	Exodus, Firewalk, Nmap, and Snort
Wireless Scanning	Airsnarf, Airsnort, GPSdrive, Kismet, and MACchanger
Target Vulnerability Validation	
Password Cracking	Allwords2, chntpw, Cisilia, Djohn, Hydra, John the Ripper, and Rcrack
Remote Access Testing	Apache Server, IKE-Scan, Net-SNMP, SSHD, TFTPD, and VNC Server
Penetration Testing	Driftnet, Dsniff, Ethereal, Ettercap, Kismet, Nessus, Netcat, Ngrep, Nmap, Ntop, and TCPdump

Appendix B—Rules of Engagement Template

This template provides organizations with a starting point for developing their ROE.[44] Individual organizations may find it necessary to include additional information.

1. Introduction

1.1. Purpose

This section should identify the purpose of the document. It should identify the organization being tested; the group conducting the testing, or if an external entity, the organization conducting the testing; and the purpose of the security test.

1.2. Scope

The scope should identify the boundaries of the information security tests in terms of actions and expected outcomes.

1.3. Assumptions and Limitations

This section should identify any assumptions made by the organization and the test team. Assumptions may relate to any aspect of the test to include the test team, the installation of appropriate safeguards for test systems, etc.

1.4. Risks

There are inherent risks when conducting information security tests, particularly intrusive tests. This section should identify these risks as well as the mitigation techniques and actions the test team will employ to reduce the risks.

1.5. Document Structure

This section should outline the structure of the ROE and describe the content of each section.

2. Logistics

2.1. Personnel

This section should identify by name all personnel assigned to the security testing task as well as the key personnel from the organization being tested. A table should be provided with all points of contact, to include test team, appropriate management personnel, and incident response team. If applicable, security clearance or comparable background-check details should be provided.

2.2. Test Schedule

This section should detail the schedule of testing and include information such as critical tests and milestones. This section should also address the hours during which the testing will take place. For

[44] The structure of this template is intended as illustrative, and organizations should organize their ROE in whatever manner they choose.

example, it may be prudent to conduct technical testing of an operational site during evening hours versus peak business hours.

2.3. Test Site

The location or locations from which testing is authorized should be identified in this section. If testing will occur on the organization's site, access to the building and equipment should be discussed. Physical access should include requirements such as badges, escorts, and security personnel the testers may encounter. Equipment access should address areas such as user or administrator-level access to the systems or network or both, and physical access to computer rooms or specific racks in the computer rooms. If there are areas to which the test team will not have access, this should be identified as well.

If testing will be conducted from a remote location such as a rented server farm or test lab, details of the test site architecture should be described in this section.

2.4. Test Equipment

This section should identify the equipment to be used by the test team to conduct the information security tests. This section should also identify the method of differentiating between the organization's systems and the systems conducting the testing. For example, if the test team's systems are identified by MAC, it would be easy to keep track of the test systems through the use of network discovery software. In addition to the hardware, the tools authorized to be used on the network should be identified. It would be appropriate to include a write-up of each tool in an appendix.

3. Communication Strategy

3.1. General Communication

This section should discuss the frequency and methods of communication. For example, if appropriate, identify the meeting schedule, locations, and conference call information.

3.2. Incident Handling and Response

This section is critical in the event there is an incident on the network while testing is in progress. In this section, criteria for halting the information security testing should be provided. Details should be provided for the test team's course of action in the event one of the test procedures negatively impacts the network or in the event the organization is attacked by an adversary during the course of the testing. In addition, the organization's incident response call tree or chain of command should be provided in a quick-reference format. The process for reinstating the test team for continuation of testing should also be provided in this section.

4. Target System/Network

This section should identify the systems and/or networks to be tested throughout the information security testing. Information should include authorized and unauthorized IP addresses or other distinguishing identifier, if appropriate for the systems (servers, workstations, firewalls, routers, etc.), OSs, and any applications to be tested. It is crucial to identify any system that is not authorized to be tested in the scope of the testing; this is referred to as the exclude list.

5. Testing Execution

This section will be particular to the test type and scope but in general should detail the allowable and unallowable activities. The information security testing methodology should be described in this section. If necessary, a test plan should be developed that complements the ROE; this could be either an appendix or a separate document.

5.1. Nontechnical Test Components

This section should identify the nontechnical test activities that will take place. Information regarding the types of policies, procedures, and other documents to be reviewed should be identified in this section. If interviews or site surveys are being done, guidelines should be established to include approving of the interview list and questions. If the physical security of the information systems is in the scope of the testing, the procedures should be laid out and a form, with the appropriate signature and contact information, generated that the test team may show to any law enforcement or onsite security personnel in the event the team is questioned.

5.2. Technical Test Components

This section should include the type of technical testing to be conducted such as network scanning, discovery, or penetration testing. It should discuss if files are authorized to be installed, created, modified, and/or executed to facilitate testing, and explain the required actions for those files upon completion of testing. Any other information regarding the technical testing of the organization's systems and networks should be included in this section. This section should contain significant detail on what activities will occur on the target network to ensure all parties are aware of what is authorized and to be expected as a result of the testing.

5.3. Data Handling

This section should identify the guidelines for gathering, storing, transmitting, and destroying test data. Requirements for data handling should be unambiguous and detailed. Data results from any type of information security test identify vulnerabilities an adversary could exploit and should be considered, at a minimum, sensitive and proprietary.

6. Reporting

This section should detail the reporting requirements and expected report deliverables to be provided throughout and upon conclusion of the testing. This section will identify the minimum information to be provided in each report (e.g., vulnerabilities and recommended mitigation techniques) and the frequency with which reports will be delivered (e.g., daily status reports). A template may be provided as an appendix to the ROE to demonstrate the report format and content.

7. Signature Page

This page is designed to identify the parties accountable throughout the test and ensure they know and understand their responsibilities throughout the course of the testing. At a minimum, the test team leader and the organization's senior management (CSO, CISO, CIO, etc.) should sign the ROE stating they understand the test scope and boundary.

Appendix C—Resources

This appendix lists a variety of additional resources related to technical security testing. Table C-1 contains a list of NIST documents that complement this guide. Table C-2 provides a list of online resources that organizations may reference for additional information.

Table C-1. Related NIST Documents

NIST Document	URL
SP 800-30, *Risk Management Guide for Information Technology Systems*	http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf
SP 800-40 Version 2.0, *Creating a Patch and Vulnerability Management Program*	http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf
SP 800-53 Revision 1, *Recommended Security Controls for Federal Information Systems*	http://csrc.nist.gov/publications/nistpubs/800-53-Rev1/800-53-rev1-final-clean-sz.pdf
Draft SP 800-53A, *Guide for Assessing the Security Controls in Federal Information Systems*	http://csrc.nist.gov/publications/PubsSPs.html
SP 800-64 Revision 1, *Security Considerations in the Information System Development Life Cycle*	http://csrc.nist.gov/publications/nistpubs/800-64/NIST-SP800-64.pdf
SP 800-84, *Guide to Test, Training, and Exercise Programs for IT Plans and Capabilities*	http://csrc.nist.gov/publications/nistpubs/800-84/SP800-84.pdf
SP 800-92, *Guide to Computer Security Log Management*	http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf
SP 800-94, *Guide to Intrusion Detection and Prevention Systems (IDPS)*	http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf

Table C-2. Online Resources

Resource	URL
Methodologies	
Information Design Assurance Red Team (IDART)	http://www.idart.sandia.gov/
NIST Special Publication 800-53A, *Guide for Assessing the Security Controls in Federal Information Systems (Draft)*	http://csrc.nist.gov/publications/PubsDrafts.html
National Security Agency (NSA) Information Assessment Methodology (IAM)	http://www.iatrp.com/iam.cfm
Open Source Security Testing Methodology Manual (OSSTMM)	http://www.isecom.org/osstmm/
Open Web Application Security Project (OWASP) Testing Project	http://www.owasp.org/index.php/Category:OWASP_Testing_Project
Tools	
BackTrack (Linux live distribution)	http://www.remote-exploit.org/backtrack.html
Extra – Knoppix (Linux live distribution)	http://www.knopper.net/knoppix-mirrors/index-en.html

Resource	URL
F.I.R.E. (Linux live distribution)	http://fire.dmzs.com/
Helix (Linux live distribution)	http://www.e-fense.com/helix/
INSERT Rescue Security Toolkit (Linux live distribution)	http://www.inside-security.de/insert_en.html
Knoppix Security Tools Distribution (STD) (Linux live distribution)	http://s-t-d.org/download.html
L.A.S. Linux (Linux live distribution)	http://www.localareasecurity.com/download
nUbuntu (Linux live distribution)	http://www.nubuntu.org/downloads.php
Operator (Linux live distribution)	http://www.ussysadmin.com/operator/
PHLAX (Linux live distribution)	http://public.planetmirror.com/pub/phlak/?fl=p
Top 100 Network Security Tools	http://sectools.org/
Vulnerability Information	
Common Configuration Enumeration (CCE)	http://cce.mitre.org/
Common Vulnerabilities and Exposures (CVE)	http://cve.mitre.org/
Common Weakness Enumeration (CWE)	http://cwe.mitre.org/
Default Password List	http://www.phenoelit-us.org/dpl/dpl.html
French Security Incident Response Team (FrSIRT)	http://www.frsirt.com/english/
iDefense Lab's Public Advisory List	http://labs.idefense.com/intelligence/vulnerabilities/
milw0rm	http://www.milw0rm.com/
National Vulnerability Database (NVD)	http://nvd.nist.gov/
Neohapsis Archives	http://archives.neohapsis.com/
Open Source Vulnerability Database	http://www.osvdb.org/
Open Web Application Security Project (OWASP) Vulnerabilities	http://www.owasp.org/index.php/Category:Vulnerability
Secunia Advisories	http://secunia.com/advisories/
SecurityFocus Vulnerabilities	http://www.securityfocus.com/vulnerabilities
SecurityTracker	http://www.securitytracker.com/
Secwatch's Vulnerability Archive	http://secwatch.org/advisories/
The Hacker's Choice (THC)	http://freeworld.thc.org/
US-CERT Vulnerability Notes Database	http://www.kb.cert.org/vuls
Wireless Vulnerabilities & Exploits (WVE)	http://www.wirelessve.org/

Appendix D—Glossary

Selected terms used in the publication are defined below.

Active Security Testing: Hands-on security testing of systems and networks to identify their security vulnerabilities.

Banner Grabbing: The process of capturing banner information, such as application type and version, that is transmitted by a remote port when a connection is initiated.

Blue Team: A test team that performs security testing with the knowledge and consent of the organization's IT staff.

External Security Testing: Security testing that is conducted from outside the organization's security perimeter.

False Positive: An alert that incorrectly indicates that malicious activity is occurring.

File Integrity Checking: Software that generates, stores, and compares message digests for files to detect changes to the files.

Information Security Testing: The process of validating the effective implementation of security controls for information systems and networks, based on the organization's security requirements.

Internal Security Testing: Security testing that is conducted from inside the organization's security perimeter.

Network Discovery: The process of discovering active and responding hosts on a network, identifying weaknesses, and learning how the network operates.

Network Sniffing: A passive technique that monitors network communication, decodes protocols, and examines headers and payloads for information of interest. Network sniffing is both a review technique and a target identification and analysis technique.

Operating System (OS) Fingerprinting: Analyzing characteristics of packets sent by a target, such as packet headers or listening ports, to identify the operating system in use on the target.

Passive Security Testing: Nonintrusive security testing primarily involving reviews of documents such as policies, procedures, security requirements, software code, system configurations, and system logs.

Password Cracking: The process of recovering secret passwords stored in a computer system or transmitted over a network.

Penetration Testing: Security testing in which evaluators mimic real-world attacks to attempt to identify methods for circumventing the security features of an application, system, or network. Penetration testing often involves issuing real attacks on real systems and data, using the common tools and techniques used by attackers. Most penetration tests involve looking for combinations of vulnerabilities on a single system or multiple systems that can be used to gain more access than could be achieved through any single vulnerability.

Phishing: A digital form of social engineering that uses authentic-looking but phony emails to request information from users or direct users to a fake web site that requests information.

Plan of Actions and Milestones (POA&M): A document that identifies tasks needing to be accomplished. It details resources required to accomplish the elements of the plan, any milestones in meeting the tasks, and scheduled completion dates for the milestones.

Port Scanner: A program that can remotely determine which ports on a system are open (e.g., whether systems allow connections through those ports).

Red Team: A test team that performs testing using covert methods and without the knowledge of the organization's IT staff, but with full knowledge and permission of upper management. Red team security testing takes an adversarial approach to assessing an organization's security posture.

Review Techniques: Passive information security testing techniques, generally conducted manually, used to evaluate systems, applications, networks, policies, and procedures to discover vulnerabilities. Review techniques include documentation review, log review, ruleset review, system configuration review, network sniffing, and file integrity checking.

Rogue Device: An unauthorized node on a network.

Rules of Engagement (ROE): Detailed guidelines and constraints regarding the execution of information security testing. The ROE is established before the start of a security test. It gives the test team authority to conduct the activities defined in the ROE without additional permission.

Ruleset: A collection of rules or signatures that network traffic or system activity is compared against to determine an action to take, such as forwarding or rejecting a packet, creating an alert, or allowing a system event.

Social Engineering: The process of attempting to trick someone into revealing information (e.g., a password) that can be used to attack systems or networks.

Target Identification and Analysis Techniques: Information security testing techniques, mostly active and generally conducted using automated tools, used to identify systems, ports, services, and potential vulnerabilities. Target identification and analysis techniques include network discovery, network port and service identification, vulnerability scanning, wireless scanning, and application security testing.

Target Vulnerability Validation Techniques: Active information security testing techniques that corroborate the existence of vulnerabilities. Target identification and analysis techniques include password cracking, remote access testing, penetration testing, social engineering, and physical security testing.

Version Scanning: The process of identifying the service application and application version in use.

Virtual Machine: Software that allows a single host to run one or more guest operating systems.

Vulnerability: A weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.

Vulnerability Scanning: A technique used to identify hosts and host attributes, and then identify the associated vulnerabilities.

Appendix E—Acronyms and Abbreviations

Selected acronyms and abbreviations used in the publication are defined below.

AP	Access Point
ARP	Address Resolution Protocol
ATA	Advanced Technology Attachment
C&A	Certification and Accreditation
CA	Certifying Authority
CGE	Cisco Global Exploiter
CIO	Chief Information Officer
CIRT	Computer Incident Response Team
CISO	Chief Information Security Officer
CT&E	Certification Test and Evaluation
CTO	Chief Technology Officer
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DAA	Designated Approving Authority
DNS	Domain Name System
DoS	Denial of Service
DSL	Digital Subscriber Line
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Management Act
FTP	File Transfer Protocol
GOTS	Government Off-the-Shelf
GPS	Global Positioning System
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IAM	Information Assessment Methodology
ICMP	Internet Control Message Protocol
IDART	Information Design Authority Red Team
IDPS	Intrusion Detection and Prevention System
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IIS	Internet Information Server
IP	Internet Protocol
IPS	Intrusion Prevention System
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISAP	Information Security Automation Program
ISN	Initial Sequence Number
ISSO	Information Systems Security Officer
ISSPM	Information Systems Security Program Manager
IT	Information Technology
ITL	Information Technology Laboratory

LAN	Local Area Network
MAC	Media Access Control
NAT	Network Address Translation
NIS	Network Information System
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NSRL	National Software Reference Library
NVD	National Vulnerability Database
OMB	Office of Management and Budget
OS	Operating System
OSSTMM	Open Source Security Testing Methodology Manual
OWASP	Open Web Application Security Project
P2P	Peer-to-Peer
PBX	Private Branch Exchange
PDA	Personal Digital Assistant
PIN	Personal Identification Number
POA&M	Plan of Action and Milestones
POP	Post Office Protocol
RF	Radio Frequency
RFC	Request for Comment
ROE	Rules of Engagement
SAISO	Senior Agency Information Security Officer
SCADA	Supervisory Control and Data Acquisition
SCAP	Security Content Automation Protocol
SHA	Secure Hash Algorithm
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
SP	Special Publication
SSH	Secure Shell
SSID	Service Set Identifier
SSL	Secure Sockets Layer
STD	Security Tool Distribution
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
US-CERT	United States Computer Emergency Readiness Team
USB	Universal Serial Bus
VM	Virtual Machine
VoIP	Voice Over Internet Protocol
VPN	Virtual Private Network

WAN	Wide Area Network
WIDPS	Wireless Intrusion Detection and Prevention System
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
XML	Extensible Markup Language